



Levy Boccato

Novas Propostas e Aplicações de Redes Neurais com Estados de Eco

Campinas

2013



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Levy Boccato

Novas Propostas e Aplicações de Redes Neurais com Estados de Eco

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Prof. Dr. Romis Ribeiro de Faissol Attux

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO LEVY BOCCATO, E ORIENTADA PELO PROF. DR. ROMIS RIBEIRO DE FAISSOL ATTUX

Campinas
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

B63n Boccato, Levy, 1986-
Novas propostas e aplicações de redes neurais com estados de eco / Levy
Boccato. – Campinas, SP : [s.n.], 2013.

Orientador: Romis Ribeiro de Faissol Attux.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de
Engenharia Elétrica e de Computação.

1. Redes neurais artificiais. 2. Processamento de sinais. 3. Aprendizado de
máquina. 4. Mapas auto-organizáveis. 5. Sistemas não lineares. I. Attux, Romis
Ribeiro de Faissol, 1978-. II. Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: New proposals and applications of echo state networks

Palavras-chave em inglês:

Artificial neural networks

Signal processing

Maps, Self-organizing

Machine Learning

Nonlinear systems

Área de concentração: Engenharia de Computação

Titulação: Doutor em Engenharia Elétrica

Banca examinadora:

Romis Ribeiro de Faissol Attux [Orientador]

Gulherme de Alencar Barreto

Ricardo Suyama

Fernando José Von Zuben

Leandro Nunes de Castro Silva

Data de defesa: 04-07-2013

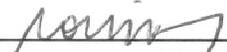
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Levy Boccato

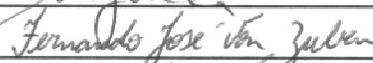
Data da Defesa: 4 de julho de 2013

Título da Tese: "Novas Propostas e Aplicações de Redes Neurais com Estados de Eco"

Prof. Dr. Romis Ribeiro de Faissol Attux (Presidente): 

Prof. Dr. Guilherme de Alencar Barreto: 

Prof. Dr. Ricardo Suyama: 

Prof. Dr. Fernando José Von Zuben: 

Prof. Dr. Leandro Nunes de Castro Silva: 

Agradecimentos

A conclusão desta tese de doutorado marca o fim de um período de muito estudo e trabalho árduo, mas, principalmente, de muitas lembranças alegres que ficarão para sempre em minha memória.

Tenho absoluta certeza que este alvo só foi alcançado pela graça e vontade soberana de Deus, que traçou este caminho para mim desde o princípio. Em todos os momentos, Sua presença em minha vida encheu-me de coragem e felicidade para trabalhar e viver cada dia. Por isso, sou profundamente grato a Ele por sua bondade para comigo. Nas palavras do salmista, “*Graças te dou, visto que por modo assombrosamente maravilhoso me formaste; as tuas obras são admiráveis, e a minha alma o sabe muito bem.*” (Salmo 139.14)

Além disso, também fui ricamente abençoado neste período pelas pessoas com quem pude conviver.

Agradeço aos meus pais, Fernando e Hélia, por todo o carinho, apoio e compreensão que recebi durante toda a minha vida. Tenho muito orgulho em saber que vocês não mediram esforços e contribuíram de maneira decisiva não apenas para minha formação acadêmica, mas, principalmente, para moldar meu caráter e para me orientar na busca por uma vida guiada pelos princípios de Deus.

Em especial, gostaria de agradecer à minha bela Rebeca. Você é a razão da minha maior alegria, fazendo-me crescer como pessoa, como servo de Deus e como seu companheiro. Em você encontro amor, compreensão, sinceridade e incentivo que me abençoam em todos os sentidos. Muito obrigado por seu amor incondicional.

Agradeço também ao meu irmão Esdras, aos meus tios, avós e aos demais familiares, por sempre estarem ao meu lado e também pela ajuda oferecida em várias circunstâncias ao longo deste período.

Ao meu orientador Romis Ribeiro de Faissol Attux, pela dedicação, por todo o auxílio dado

desde a iniciação científica até agora e por ter me encorajado a perseguir este sonho. Seu espírito contagiante sempre me ensinou a ter apreço pela alegria de trabalhar e a preservar uma visão ampla acerca do papel de um docente. Fico muito feliz por saber que desenvolvemos ao longo destes anos uma amizade verdadeira, a qual levarei comigo por toda a vida.

Ao Prof. Fernando José Von Zuben, com quem pude interagir durante todo o doutorado. É uma satisfação para mim poder trabalhar e aprender com você, além de poder contar com sua amizade.

Ao Prof. Amauri Lopes, por ter sido o pioneiro a me guiar na vida acadêmica e pela amizade que desde então cultivamos. Em cada conversa, sempre pude aprender lições valiosas que me mostraram não apenas sua competência, mas também sua vocação para o ensino e orientação, tendo deixado uma marca indelével em minha formação.

Ao Prof. Christiano Lyra Filho e ao colega de doutorado Hugo V. Siqueira, por me darem a oportunidade de trabalhar em um ambiente novo e bastante acolhedor, e também de desenvolver uma agradável amizade em meio às atividades científicas.

A todos os amigos do Laboratório de Processamento de Sinais para Comunicações (DSP-Com), pela companhia e pelos divertidos momentos que pudemos desfrutar juntos.

Aos professores que participaram da banca de defesa, por todas as sugestões e pelas cuidadosas revisões que permitiram o aprimoramento do trabalho.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pela oportunidade concedida e por todo o apoio financeiro.

Resumo

As redes neurais com estados de eco (em inglês, *echo state networks*, ESNs) são estruturas recorrentes capazes de aliar processamento dinâmico a um processo de treinamento relativamente simples, o qual se resume à adaptação dos coeficientes do combinador linear da saída no sentido de mínimo erro quadrático médio (em inglês, *mean squared error*, MSE), enquanto os pesos das conexões no reservatório de dinâmicas são ajustados de maneira antecipada e permanecem fixos.

A presente tese trata dos principais elementos que caracterizam as ESNs e propõe: (i) uma unificação entre as abordagens de computação com reservatórios, como as ESNs e as *liquid state machines* (LSMs), e as *extreme learning machines* (ELMs), sob o termo geral de máquinas desorganizadas, o qual estabelece uma conexão com as pioneiras ideias conexionistas de Alan Mathison Turing; (ii) uma nova arquitetura de ESN, cuja camada de saída é composta por um filtro de Volterra e por um estágio de compressão baseado em Análise de Componentes Principais (em inglês, *Principal Component Analysis*, PCA); (iii) o uso de critérios de aprendizado baseados em teoria da informação e em normas L_p em lugar do critério MSE para a adaptação dos parâmetros da camada de saída de ESNs; e (iv) uma estratégia não-supervisionada de projeto da camada recorrente de ESNs baseada em interações laterais, modeladas segundo a função chapéu mexicano, e na auto-organização dos pesos de entrada.

As propostas elaboradas neste trabalho são analisadas através de simulações no contexto de diferentes problemas de processamento da informação, como equalização de canais de comunicação, separação de fontes e predição de séries temporais.

Abstract

Echo state networks (ESNs) are recurrent structures capable of allying dynamic processing to a relatively simple training process, which amounts to adapting the coefficients of the linear combiner at the output in the minimum mean squared error (MSE) sense, while the connection weights in the dynamical reservoir are adjusted in advance and remain fixed.

The present thesis deals with the main elements that characterize ESNs and proposes: *(i)* a unification between reservoir computing approaches, such as ESNs and liquid state machines (LSMs), and extreme learning machines (ELMs), under the general term of unorganized machines, which establishes a connection with the pioneering connectionist ideas of Alan Mathison Turing; *(ii)* a novel ESN architecture whose output layer is composed of a Volterra filter and of a compression stage based on Principal Component Analysis (PCA); *(iii)* the use of information-theoretic learning criteria and those based on L_p norms instead of the MSE criterion for the adaptation of the parameters of the ESN output layer; and *(iv)* an unsupervised strategy for designing the recurrent layer of ESNs based on lateral interactions, modeled according to the mexican hat function, and on the self-organization of the input weights.

The proposals developed in this work are analyzed through simulations in the context of different information processing problems, such as channel equalization, source separation and time series prediction.

Abreviaturas

AMSE	<i>Average Mean Squared Error</i> – Média do Erro Quadrático Médio
ANN	<i>Artificial Neural Network</i> – Rede Neural Artificial
ASE	<i>Average State Entropy</i> – Entropia Média dos Estados
AWGN	<i>Additive White Gaussian Noise</i> – Ruído Branco Aditivo Gaussiano
AWLN	<i>Additive White Laplace Noise</i> – Ruído Branco Aditivo Laplaciano
BER	<i>Bit Error Rate</i> – Taxa de Erro de Bit
BMU	<i>Best Matching Unit</i>
BPTT	<i>Backpropagation-Through-Time</i>
BSS	<i>Blind Source Separation</i> – Separação Cega de Fontes
EEC	<i>Error Entropy Criterion</i> – Critério de Entropia do Erro
ELM	<i>Extreme Learning Machine</i>
ESN	<i>Echo State Network</i> – Rede Neural com Estados de Eco
ESP	<i>Echo State Property</i> – Propriedade de Estados de Eco
FIR	<i>Finite Impulse Response</i> – Resposta ao Impulso Finita
FNN	<i>Feedforward Neural Network</i> – Rede Neural <i>Feedforward</i>
ICA	<i>Independent Component Analysis</i> – Análise de Componentes Independentes
IIR	<i>Infinite Impulse Response</i> – Resposta ao Impulso Infinita
IIS	<i>Intersymbol Interference</i> – Interferência Inter-Simbólica
IP	<i>Information Potential</i> – Potencial de Informação
IPL	<i>Intrinsic Plasticity</i> – Plasticidade Intrínseca

IRLS	<i>Iteratively Reweighted Least Squares</i>
ITL	<i>Information-Theoretic Learning</i> – Aprendizado Baseado em Teoria da Informação
LASSO	<i>Least Absolute Selection and Shrinkage Operator</i>
LMS	<i>Least Mean Squares</i>
LSM	<i>Liquid State Machine</i>
MAP	<i>Maximum a Posteriori</i>
MCC	<i>Maximum Correntropy Criterion</i> – Critério de Máxima Correntropia
MCC-SIG	<i>Stochastic Information Gradient for Maximum Correntropy Criterion</i>
MED	<i>Maximum Entropy Distribution</i> – Distribuição de Máxima Entropia
MEEC	<i>Minimum Error Entropy Criterion</i> – Critério de Mínima Entropia do Erro
MEE-SIG	<i>Stochastic Information Gradient for Minimum Error Entropy</i>
MLP	<i>Multilayer Perceptron</i> – Perceptron de Múltiplas Camadas
MSE	<i>Mean Squared Error</i> – Erro Quadrático Médio
NARMA	<i>Nonlinear Auto-Regressive Moving Average</i>
NG	<i>Neural Gas</i>
NPEF	<i>Nonlinear Prediction-Error Filter</i> – Filtro de Erro de Predição Não-Linear
PCA	<i>Principal Component Analysis</i> – Análise de Componentes Principais
PDF	<i>Probability Density Function</i> – Função Densidade de Probabilidade
RBF	<i>Radial Basis Function</i> – Função de Base Radial
RC	<i>Reservoir Computing</i> – Computação com Reservatórios
RLS	<i>Recursive Least Squares</i>
RNN	<i>Recurrent Neural Network</i> – Rede Neural Recorrente
RTRL	<i>Real-Time Recurrent Learning</i>
SLFN	<i>Single-Hidden Layer Feedforward Neural Network</i>
SNR	<i>Signal-to-Noise Ratio</i> – Relação Sinal-Ruído
SOM	<i>Self-Organizing Map</i> – Mapa Auto-Organizável
SVM	<i>Support Vector Machine</i> – Máquina de Vetores Suporte
UM	<i>Unorganized Machines</i> – Máquinas Desorganizadas

Sumário

Agradecimentos	vii
Resumo/Abstract	ix
Abreviaturas	xiii
1 Introdução	1
1.1 Objetivos e Organização da Tese	2
2 Redes Neurais com Estados de Eco	7
2.1 Modelo de Neurônio	8
2.2 Redes Neurais Artificiais	10
2.2.1 Redes MLP	11
2.2.2 Redes Recorrentes	14
2.3 Redes Neurais com Estados de Eco	18
2.3.1 Propriedade de Estados de Eco	20
2.3.2 Processo de Treinamento de uma ESN	21
2.4 Conclusão	22
3 Máquinas Desorganizadas de Turing: Conexão Histórica	25
3.1 Introdução	25

3.2	Redes de Turing	26
3.2.1	Rede tipo-A	28
3.2.2	Rede tipo-B	29
3.2.3	Rede tipo-BI	31
3.2.4	Discussão	32
3.3	Computação com Reservatórios	34
3.3.1	Liquid State Machines	34
3.4	Extreme Learning Machines	36
3.5	Conexão entre Turing e modernas máquinas desorganizadas	38
4	Aplicações	43
4.1	Equalização de Canais de Comunicação	44
4.1.1	Fonte de Informação	44
4.1.2	Modelos de Canais de Comunicação	45
4.1.3	Estados do Canal	46
4.1.4	Solução: Projeto de Equalizadores	50
4.1.5	Equalização Supervisionada	52
4.1.6	Equalização Não-Supervisionada	54
4.2	Separação de Fontes	57
4.2.1	Separação de Misturas Convolutivas	58
4.3	Predição de Séries Caóticas	61
4.3.1	Mapa Logístico	62
4.3.2	Sistema de Lorenz	63
4.4	Predição de Séries de Vazões Mensais	64
4.4.1	Séries de Vazões Mensais	65
4.5	Conclusão	67

5	Nova Arquitetura de ESN	69
5.1	Propostas Existentes para a Camada de Saída	70
5.2	Filtro de Volterra	72
5.3	Análise de Componentes Principais	76
5.4	Resultados Experimentais	78
5.4.1	Metodologia	78
5.4.2	Equalização Supervisionada	81
5.4.3	Equalização Não-Supervisionada	87
5.4.4	Separação de Misturas Convolutivas	89
5.4.5	Predição de Séries Caóticas	94
5.4.6	Predição de Séries de Vazões Mensais	100
5.4.7	Resultados Experimentais	101
5.5	Conclusão	105
6	CrITÉRIOS de Adaptação da Camada de Saída	107
6.1	Information-Theoretic Learning	108
6.1.1	Entropia de Rényi	109
6.1.2	Estimador Não-Paramétrico de Entropia	110
6.1.3	CrITÉRIO de Mínima Entropia do Erro	113
6.1.4	CrITÉRIO de Máxima Correntropia	116
6.2	Normas L_p	119
6.3	Resultados Experimentais	126
6.3.1	Metodologia	126
6.3.2	Primeiro Cenário	128
6.3.3	Segundo Cenário	134
6.4	Regularização	138
6.4.1	Ridge Regression	140
6.4.2	LASSO	141

6.4.3	Análise	145
6.5	Conclusão	149
7	Projeto do Reservatório de Dinâmicas	151
7.1	Propostas Existentes para o Reservatório	152
7.2	Proposta: Interação Lateral e Auto-Organização	155
7.3	Estratégias de Auto-Organização	163
7.3.1	Mapas Auto-Organizáveis	164
7.3.2	Neural Gas	165
7.4	Resultados Experimentais	167
7.4.1	Metodologia	168
7.4.2	Equalização Supervisionada	169
7.4.3	Predição de Séries Caóticas	176
7.5	Conclusão	180
8	Conclusões e Perspectivas	183
A	Equalizador Bayesiano	191
A.1	Canal AWGN	193
A.2	Canal AWLN	195
	Referências Bibliográficas	197

Introdução

As redes neurais recorrentes (em inglês, *recurrent neural networks*, RNNs) constituem uma importante classe de ferramentas de neurocomputação devido ao grande potencial que possuem para lidar com problemas de natureza dinâmica (temporal), uma vez que dispõem de laços de realimentação entre diferentes camadas de neurônios. No entanto, os principais algoritmos desenvolvidos para o processo de treinamento deste tipo de rede podem ser relativamente complexos e enfrentar dificuldades em termos de convergência e robustez (Haykin, 1998; Lukosevicius e Jaeger, 2009).

Recentemente, trabalhos pertencentes a uma nova frente de pesquisa, denominada computação com reservatórios (em inglês, *reservoir computing*, RC), abriram uma perspectiva interessante para o emprego de estruturas recorrentes ao introduzirem uma simplificação em seu processo de treinamento. Em particular, Jaeger (2001) propôs as chamadas redes neurais com estados de eco (em inglês, *echo state networks*, ESNs), as quais podem ser vistas como um promissor compromisso entre dois objetivos aparentemente conflitantes: (i) simplicidade do modelo matemático resultante; e (ii) capacidade de expressar uma ampla gama de comportamentos dinâmicos não-lineares.

Ao impor pesos fixos para as conexões recorrentes, a abordagem baseada no conceito de estados de eco evita as dificuldades enfrentadas pelas estratégias de treinamento de RNNs,

mas ainda preserva, até certo ponto, o potencial da estrutura subjacente devido à existência de laços de realimentação no reservatório de dinâmicas. Além disso, o processo de treinamento é relativamente simples, uma vez que consiste em adaptar de maneira supervisionada os parâmetros da camada de saída, a qual usualmente equivale a um combinador linear.

Apesar destes atrativos, as ESNs não são capazes de aproveitar toda a informação estatística referente aos sinais que as atravessam devido ao caráter linear da estrutura empregada na camada de saída e ao emprego do critério de mínimo erro quadrático médio para o ajuste de seus parâmetros. Por isso, uma busca por modificações estruturais que possibilitem uma melhor exploração dos sinais gerados pela camada não-linear recorrente é certamente relevante. Além disto, o projeto do reservatório de dinâmicas, que constitui outro elemento crucial para o uso efetivo das ESNs, requer o desenvolvimento de estratégias que contribuam para uma maior diversidade de comportamentos dinâmicos que ele pode gerar, o que significa que há espaço para novas contribuições.

1.1 Objetivos e Organização da Tese

A presente tese tem por objetivo apresentar novas propostas para as duas partes fundamentais que compõem uma rede neural com estados de eco: 1) a camada de saída e 2) o reservatório de dinâmicas. No primeiro caso, abordaremos tanto a possibilidade de modificar a estrutura de processamento por meio da introdução de não-linearidades, sem que isto comprometa a simplicidade de treinamento, quanto a perspectiva de atingir uma extração mais efetiva da informação através do emprego de critérios mais robustos de adaptação dos parâmetros livres. No segundo caso, a possibilidade de incorporar informações relevantes a respeito da tarefa que a rede deve realizar ao projeto da camada recorrente será explorada. Sendo assim, o conteúdo desta tese está organizado da seguinte maneira.

No Capítulo 2, apresentamos os fundamentos das redes neurais com estados de eco. Inicialmente, revisamos alguns aspectos básicos de redes neurais artificiais, como o modelo

de neurônio e os principais tipos de arquitetura neural, contrastando as redes *feedforward*, como o perceptron de múltiplas camadas (em inglês, *multilayer perceptron*, MLP), e as redes recorrentes. Em seguida, detalhamos os aspectos envolvidos no treinamento de uma RNN, para, então, motivar a abordagem baseada no conceito de estados de eco. Por fim, definimos as chamadas redes neurais com estados de eco bem como a estratégia de projeto e de treinamento que as caracterizam.

Interessantemente, o fato de as ESNs mesclarem elementos não-treinados com componentes totalmente adaptados de acordo com um sinal de referência permite que se estabeleça um elo com as chamadas *extreme learning machines* (ELMs) (Huang, Zhu, e Siew, 2006), as quais, embora sejam estruturas do tipo *feedforward*, também são marcadas pela presença de uma camada intermediária que não está sujeita a treinamento e de uma camada de saída que é efetivamente adaptada. Este elo, na verdade, pode ser visto como um resgate da ideia de desorganização, o que remete ao trabalho pioneiro de Alan Turing (Turing, 1968). Esta conexão histórica entre as contribuições de Alan Turing ao conexionismo e alguns paradigmas recentes de redes neurais artificiais, como as ESNs e as ELMs, é delineada no Capítulo 3 e representa a primeira contribuição original alcançada neste trabalho.

No Capítulo 4, descrevemos os fundamentos dos problemas de tratamento da informação utilizados no decorrer da tese para a avaliação das diferentes propostas de ESNs. Primeiramente, apresentamos os principais conceitos referentes ao problema de equalização de canais de comunicação, tanto em sua formulação supervisionada quanto o caso não-supervisionado, no qual uma abordagem baseada em filtros de erro de predição é explorada (Ferrari et al., 2003; Ferrari, Suyama, Lopes, Attux, e Romano, 2008). Em seguida, passamos para o problema de separação de fontes (Hyvärinen, Karhunen, e Oja, 2001), mais especificamente no âmbito de misturas convolutivas. Por fim, discutimos duas tarefas relevantes de predição de séries temporais associadas a 1) sistemas caóticos (Abarbanel, 1997) e 2) medidas de vazões mensais de rios (Box, Jenkins, e Reinsel, 1994). Estes problemas foram escolhidos não apenas por sua importância teórica/prática mas também pelo fato de exigirem um equilíbrio entre

complexidade computacional e capacidade de processamento, o que condiz com o espírito das ESNs. É importante destacar que, em alguns casos, como equalização cega e separação de fontes, a aplicação de ESNs é inédita.

No Capítulo 5, apresentamos uma nova arquitetura de ESN caracterizada pelo uso de uma estrutura do tipo filtro de Volterra na camada de saída. Esta proposta é motivada pelo fato de o filtro de Volterra não apenas proporcionar a exploração de estatísticas de ordem superior referentes aos sinais gerados pelo reservatório, mas também preservar a simplicidade do processo de treinamento, uma vez que é possível determinar a solução ótima para seus parâmetros que minimiza o erro quadrático médio de maneira fechada (Mathews, 1991; Haykin, 1996). Além disso, evitamos a possibilidade de um crescimento excessivo do número de pesos através da aplicação da técnica baseada em análise de componentes principais (em inglês, *principal component analysis*, PCA) (Jolliffe, 1986; Johnson e Wichern, 2007) antes de transmitir os estados da rede para a camada de saída.

Outra perspectiva capaz de ampliar o aproveitamento estatístico na camada de saída das ESNs, e que ainda não foi abordada na literatura, é discutida no Capítulo 6: em vez de ajustar os parâmetros livres de acordo com o critério de mínimo erro quadrático médio, o qual efetivamente usa as estatísticas de ordem dois dos sinais da rede, propomos estudar as vantagens de se utilizar os critérios derivados do paradigma de aprendizado baseado em teoria da informação (Principe, 2010), mais especificamente, os de mínima entropia do erro e máxima correntropia, e também as opções que visam minimizar normas L_p da medida de erro (Rice e White, 1964). Além disso, também analisamos os possíveis benefícios que o uso de técnicas de regularização no treinamento de ESNs pode trazer em termos da capacidade de generalização da rede.

No Capítulo 7 abordamos a questão do projeto da camada recorrente das ESNs e propomos um novo método não-supervisionado para o ajuste dos pesos do reservatório caracterizado: (i) pela inserção de interações laterais de excitação entre neurônios vizinhos e de estímulos de inibição entre unidades mais distantes, tendo como objetivo a formação de diferentes grupos

de neurônios que são ativados de forma mais intensa por padrões de entrada pertencentes a classes diferentes (Kohonen, 1982), e (ii) pela auto-organização dos pesos de entrada.

Finalmente, no Capítulo 8, as conclusões gerais sobre o trabalho e algumas perspectivas de continuidade são apresentadas.

Redes Neurais com Estados de Eco

Sem dúvida, um dos sistemas mais fascinantes e intrigantes dentre os presentes nas formas de vida mais desenvolvidas, em especial na espécie humana, é o sistema nervoso. Em poucas palavras, podemos dizer que o sistema nervoso é responsável por informar o organismo acerca das condições do ambiente no qual está inserido por meio de entradas sensoriais, por processar a informação coletada, relacionando-a com experiências prévias presentes em uma espécie de memória e por definir ações apropriadas como resposta aos estímulos recebidos (de Castro, 2006).

O cérebro, em particular, tornou-se objeto de estudos e também de apreciação devido a suas peculiaridades: ao mesmo tempo que exibe impressionante habilidade em realizar algumas tarefas, como controle motor, percepção, inferência e reconhecimento de padrões, ele também é impreciso, lento e está sujeito a erros de generalização.

Uma característica muito relevante do sistema nervoso é que este não apenas realiza diversas funções importantes para a manutenção da vida, como também se adapta à medida que experimenta novos desafios. Esta perspectiva de contínuo aprendizado o torna ainda mais atraente não somente aos olhos de biólogos, como também de engenheiros, físicos e matemáticos, os quais vislumbram a possibilidade de reproduzir ou imitar algumas de suas principais habilidades em outros sistemas na esperança de alcançar a solução para os desafios com os quais se deparam.

Neurocomputação é o termo utilizado para definir o ramo de pesquisa dedicado ao estudo e desenvolvimento de modelos e ferramentas computacionais que se inspiram em princípios do sistema nervoso biológico, com o propósito de resolver problemas (de Castro, 2006). As redes neurais artificiais (em inglês, *artificial neural networks*, ANNs), seu principal fruto, são estruturas compostas de unidades de processamento, denominadas neurônios artificiais, interligadas segundo um padrão de conexões específico. Essas redes constituem poderosas ferramentas capazes de se adaptar e desempenhar uma ampla gama de tarefas, como classificação, reconhecimento de padrões, predição de séries temporais, controle e identificação de sistemas, clusterização e aproximação de funções (Haykin, 1998; de Castro, 2006).

2.1 Modelo de Neurônio

A unidade básica de processamento de informação de uma ANN, denominada neurônio artificial, recebe um conjunto de estímulos de entrada provenientes de outros neurônios da rede ou do próprio ambiente, realiza algum tipo de combinação e transformação destes sinais e, por fim, gera um estímulo de saída. Os sinais chegam e partem de cada neurônio por meio de conexões, denominadas sinapses. Cada sinapse possui uma eficiência própria, representada por um peso associado a ela, que corresponde à informação que é efetivamente armazenada no neurônio e, conseqüentemente, na rede. A Figura 2.1 exibe a estrutura do modelo de um neurônio artificial.

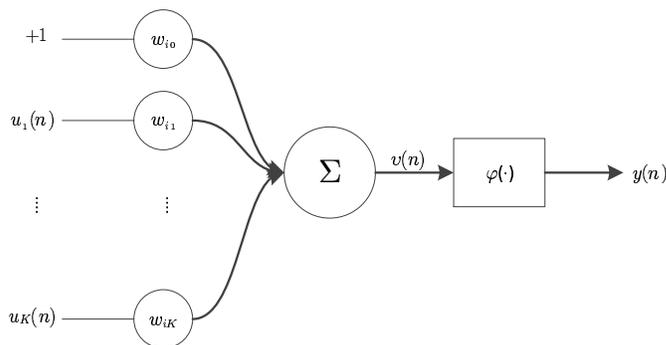


Figura 2.1: Modelo de um neurônio artificial.

Em síntese, os sinais de entrada $u_k(n)$, $k = 1, \dots, K$, junto com um sinal de polarização (*bias*), fixado no valor $+1$, são ponderados linearmente pelos pesos sinápticos w_{ik} , onde i representa o índice do neurônio e k indica o sinal ao qual o peso está associado, e somados, gerando a ativação $v(n)$, a qual passa por uma função de ativação $\varphi(\cdot)$, usualmente de caráter não-linear, gerando a saída $y(n)$. Portanto, em termos matemáticos, a saída do neurônio pode ser expressa da seguinte forma:

$$y(n) = \varphi \left(\sum_{k=1}^K w_{ik} u_k(n) + w_{i0} \right). \quad (2.1)$$

Em 1943, combinando elementos de neurofisiologia e lógica matemática, McCulloch e Pitts propuseram o primeiro modelo algébrico de um neurônio artificial (McCulloch e Pitts, 1943). O neurônio de McCulloch e Pitts é binário, i.e., pode assumir apenas um entre dois estados (0 e 1), e apresenta o seguinte modo de operação: a cada instante de tempo, se não há uma sinapse inibitória ativa, os sinais binários de entrada são somados e o neurônio dispara, i.e., sua saída recebe o valor 1, caso esta soma ultrapasse um limiar θ ; caso contrário, o neurônio permanece inativo, ou seja, responde com um 0 em sua saída.

Observe que o modelo de neurônio mostrado na Figura 2.1 contempla a proposta de McCulloch e Pitts (1943): se todos os pesos w_{ij} forem iguais a 1 e a função de ativação for tal que

$$y(n) = \varphi(v(n)) = \begin{cases} 1 & \text{se } v(n) \geq \theta \\ 0 & \text{caso contrário} \end{cases}, \quad (2.2)$$

chega-se ao modelo proposto por McCulloch e Pitts (1943).

Em 1958, F. Rosenblatt (1958) propôs um modelo de neurônio, junto com uma metodologia de treinamento supervisionado inspirada na ideia de aprendizado de Hebb (Hebb, 1949), denominado *perceptron*¹. Este modelo é bastante parecido com aquele exibido na Figura 2.1,

¹No período entre os trabalhos de McCulloch e Pitts (1943) e F. Rosenblatt (1958), Alan Turing elaborou seu próprio modelo de neurônio artificial, bem como propôs alguns tipos de redes, chamadas de máquinas desorganizadas (Turing, 1968). No Capítulo 3, revisitaremos este trabalho pioneiro a fim de ressaltar os pontos de contato entre as ideias de Turing e alguns paradigmas conexionistas modernos, e.g., as ESNs.

com a restrição de que a função de ativação deve ser similar à empregada por McCulloch e Pitts (1943):

$$y(n) = \varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{caso contrário} \end{cases}, \quad (2.3)$$

onde

$$v = \sum_{k=1}^K w_{ik} u_k(n) + w_{i0}. \quad (2.4)$$

Deste modo, o neurônio atua como um discriminante linear, sendo capaz de classificar corretamente apenas padrões que podem ser separados por um hiperplano no espaço de atributos (Minsky e Papert, 1969).

A ideia de empregar funções de ativação não-lineares se mostra atraente na medida em que permite que a rede gere mapeamentos mais flexíveis, sendo assim capaz de aproximar com maior precisão a resposta desejada quando comparada com modelos lineares (Haykin, 1998). As funções sigmóides, de forma geral, são bastante empregadas em algumas arquiteturas de redes neurais, como as redes MLP, devido a algumas características relevantes, tais como (i) continuidade e diferenciabilidade em todos os pontos, o que viabiliza o uso de algoritmos clássicos de adaptação baseados no gradiente, (ii) saturação do sinal de saída de cada neurônio, o que evita divergência, e (iii) versatilidade para criar mapeamentos mais simples ou complexos, uma vez que são quase lineares em torno da origem e, ao mesmo tempo, possuem forte caráter não-linear próximo da saturação.

2.2 Redes Neurais Artificiais

As redes neurais artificiais são caracterizadas não apenas pelo modelo de neurônio que utilizam, mas também pelo padrão de conectividade entre as unidades - o qual define a arquitetura da rede - e pela estratégia empregada no ajuste dos pesos sinápticos, chamada de algoritmo de aprendizado ou treinamento.

Existem várias propostas de arquiteturas de redes neurais artificiais, dentre as quais men-

cionamos as redes MLP e RBF (*radial basis function*), além dos mapas auto-organizáveis de Kohonen e das redes de Hopfield (Haykin, 1998; Kohonen, 2000; Hopfield, 1982). Mesmo assim, é possível dividir as diferentes arquiteturas em dois grupos básicos: redes *feedforward* (em inglês, *feedforward neural networks*, FNNs) e redes recorrentes (em inglês, *recurrent neural networks*, RNNs). No primeiro caso, os sinais recebidos pela rede são propagados em um único sentido até que as saídas correspondentes sejam determinadas, enquanto, no segundo caso, existem laços de realimentação que transmitem as saídas de neurônios de uma determinada camada para neurônios pertencentes a camadas anteriores.

2.2.1 Redes MLP

Um dos principais expoentes da classe de FNNs é a rede chamada de perceptron de múltiplas camadas (em inglês, *multilayer perceptron*, MLP). A Figura 2.2 exibe a arquitetura geral de uma rede MLP.

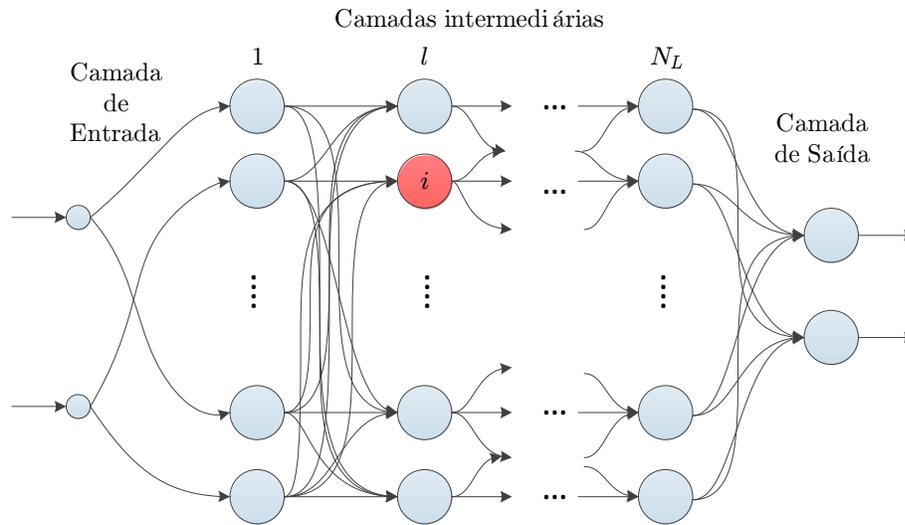


Figura 2.2: Arquitetura de uma rede MLP.

Em síntese, uma rede MLP é composta por diversas unidades de processamento que seguem o modelo apresentado na Figura 2.1 e pode ser dividida em três partes principais: (1) uma camada de entrada, que transmite os sinais externos para a primeira camada intermediária; (2) uma ou mais camadas intermediárias, cuja função é projetar os sinais de entrada

segundo um mapeamento não-linear em espaços de dimensão tipicamente maior que a do espaço original dos dados; e (3) uma camada de saída, que recebe as ativações dos neurônios da última camada intermediária para, através de combinações lineares destes sinais e, eventualmente, uma transformação não-linear, produzir as saídas da ANN.

Considere o i -ésimo neurônio da l -ésima camada intermediária, como destacado na Figura 2.2, sendo $i = 1, \dots, N^l$ e $l = 1, \dots, N_L$, onde N^l denota o número de unidades presentes na camada l e N_L é o número total de camadas intermediárias da MLP. Sua saída $y_i^l(n)$ pode ser determinada por meio da seguinte expressão:

$$y_i^l(n) = \varphi^l \left(\sum_{k=1}^{N^{l-1}} w_{ik}^l y_k^{l-1} + w_{i0}^l \right), \quad (2.5)$$

onde w_{ik}^l representa o peso sináptico da conexão que liga o k -ésimo neurônio da camada $l-1$ ao i -ésimo neurônio da camada l . Para a primeira camada intermediária ($l = 1$), os sinais combinados são as próprias entradas da rede, i.e., $y_k^0 = u_k(n)$, $k = 1, \dots, K$.

O processo de treinamento de uma rede MLP consiste no ajuste de todos os pesos sinápticos w_{ik}^l , tanto das camadas intermediárias quanto da camada de saída, na qual, usualmente, são utilizados somente neurônios lineares, tendo como objetivo encontrar o conjunto de pesos sinápticos que realize o melhor mapeamento possível de entrada-saída, isto é, que forneça valores para as saídas os mais próximos possíveis dos valores desejados. Pode-se, portanto, formular o processo de treinamento como um problema de minimização de uma medida de erro entre as saídas fornecidas pela rede e as saídas desejadas. Desta maneira, é possível empregar técnicas clássicas de otimização não-linear irrestrita para efetuar o treinamento, como, por exemplo, métodos iterativos de primeira e segunda ordens (Haykin, 1998; de Castro, 1998; Luenberger, 2003), entre os quais citamos o conhecido algoritmo *error backpropagation* (Werbos, 1974; Rumelhart, Hinton, e Williams, 1986).

Uma vez fixados os pesos sinápticos da rede, cada padrão de entrada é mapeado em um único - e sempre o mesmo - conjunto de sinais de saída através da propagação das

ativações dos neurônios pelas camadas da rede. Por isso, as redes MLP estão naturalmente habilitadas para lidar com problemas de natureza estática, tais como aproximação de funções e classificação de padrões (Haykin, 1998; Duda, Hart, e Stork, 2001). Além disso, este tipo de rede *feedforward* possui capacidade de aproximação universal, como demonstrado em (Cybenko, 1989) e (Hornik, Stinchcombe, e White, 1989), o que significa que, dada uma função contínua não-constante em um espaço de dimensão finita, existe uma rede MLP que consegue aproximá-la com precisão arbitrária. Entretanto, é importante mencionar que o teorema de aproximação universal não aponta qual é o número de neurônios que precisam estar na camada intermediária, tampouco um método eficiente para o ajuste dos pesos que garantidamente atinja a configuração ótima da rede.

Problemas de natureza dinâmica e temporal, como previsão, identificação e controle de sistemas dinâmicos (Suykens, Vandewalle, e De Moor, 1996; Kuznetsov, Kuznetsov, e Marsden, 1998), filtragem adaptativa (Haykin, 1996) e separação de misturas convolutivas (Hyvärinen et al., 2001) também compõem uma parte muito relevante do repertório de tarefas que pode ser abordado com o auxílio de redes neurais artificiais. No entanto, uma exigência que tais problemas impõem a qualquer modelo que almeje resolvê-los é que este seja capaz de acessar e memorizar o histórico dos sinais de entrada e saída, além dos estados internos do sistema. Neste contexto, ainda que algumas estruturas do tipo *feedforward* apresentem capacidade de aproximação universal, como as redes MLP e RBF (Cybenko, 1989; Park e Sandberg, 1991), e até possam ser aplicadas a estes casos, tal abordagem pode eventualmente ser pouco eficiente, especialmente quando o problema dinâmico apresenta uma profunda dependência de valores passados.

Uma alternativa natural para tratar estes problemas é permitir que o tempo seja representado pelo efeito que tem sobre o processamento, o que significa conferir ao sistema responsável pelo processamento propriedades dinâmicas que respondam às sequências temporais. Isto pode ser feito através da inserção de laços de realimentação entre diferentes camadas e/ou neurônios, de modo que a rede passe a ter uma espécie de memória, sendo,

portanto, capaz de guardar informações úteis a respeito dos sinais para uso posterior.

2.2.2 Redes Recorrentes

As redes neurais recorrentes agregam um grande potencial no que se refere à memorização e à capacidade de acessar o histórico dos sinais presentes no sistema, uma vez que possuem laços de realimentação entre diferentes camadas. Além disto, este tipo de arquitetura também apresenta capacidade de aproximação universal, como demonstrado em (Funahashi e Nakamura, 1993; Schafer e Zimmermann, 2007), e revela pontos de contato com a ideia de máquina de Turing universal (Kilian e Siegelmann, 1996). Todos esses fatos atestam que as RNNs são poderosas ferramentas de processamento de sinais, com boas perspectivas de aplicação em problemas dinâmicos e temporais de aprendizado de máquina.

Não obstante esse caráter promissor, a presença de laços de realimentação transforma uma rede recorrente em um sistema dinâmico não-linear bastante complexo, de maneira que é necessário ajustar os seus parâmetros de forma criteriosa a fim de garantir que ela se comporte adequadamente de modo a atingir o objetivo desejado. No contexto de aprendizado supervisionado, este procedimento é usualmente realizado com o auxílio de métodos iterativos que visam minimizar uma função de erro entre as saídas da rede e as respostas desejadas, tais como o *backpropagation-through-time* (BPTT) (Werbos, 1990), *real-time recurrent learning* (RTRL) (Williams e Zipser, 1989) e algoritmos de segunda ordem (dos Santos e Von Zuben, 2000).

O algoritmo BPTT, proposto por Werbos (1990), estende a ideia de retropropagação do sinal de erro para o âmbito das redes neurais recorrentes. Neste contexto, não apenas é preciso computar o gradiente da função de erro através das sucessivas camadas intermediárias, como ocorre nas redes MLP, como também considerar a influência dos sinais passados que retornam à rede graças às realimentações.

A estratégia proposta para lidar com o efeito das conexões recorrentes consiste em desdobrar a estrutura da rede através do empilhamento de várias cópias da arquitetura original,

uma para cada instante de tempo considerado, de forma a obter uma arquitetura expandida do tipo *feedforward* em que as realimentações são explicitamente representadas como conexões entre estruturas associadas a diferentes instantes de tempo e transmitem sinais pela cadeia até a estrutura no topo. A fim de visualizarmos este processo, exibimos nas Figuras 2.3 e 2.4 um exemplo simples de RNN e a arquitetura obtida após o empilhamento de três cópias da rede.

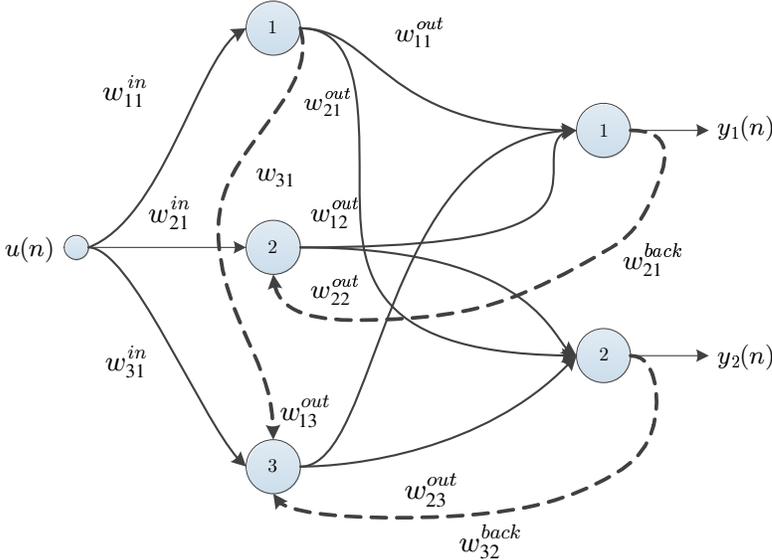


Figura 2.3: Exemplo de uma RNN com uma entrada, três neurônios na camada intermediária e duas saídas. As linhas tracejadas (--) representam os laços de realimentação.

Uma vez construída a arquitetura estendida, pode-se aplicar diretamente o conceito de retropropagação do erro para calcular as derivadas da função de erro em função de todos os pesos sinápticos, já levando em conta a dependência temporal dos sinais: primeiro, os padrões de entrada $u(n), u(n-1), \dots$, são apresentados na entrada das respectivas estruturas e as ativações dos neurônios são propagadas pela cadeia inteira até que se obtenha as saídas no instante atual; em seguida, calcula-se, no sentido inverso, as derivadas da função de erro com respeito a todos os pesos e em todos os instantes de tempo segundo a metodologia do algoritmo *backpropagation*; finalmente, os pesos sinápticos são ajustados combinando a direção do gradiente calculado em cada instante de tempo (Werbos, 1990; Jaeger, 2002b).

Intuitivamente, porém, é possível perceber alguns aspectos críticos nesta abordagem. Em

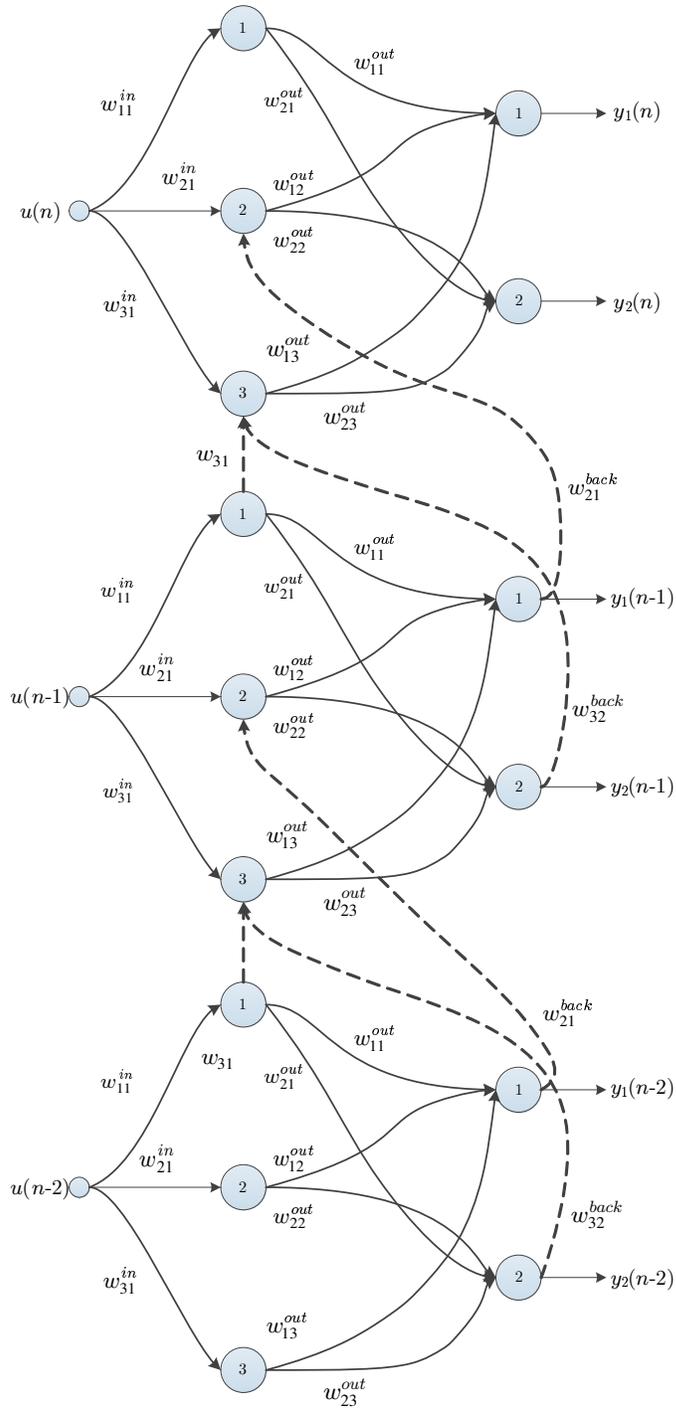


Figura 2.4: Arquitetura estendida de uma RNN.

princípio, seria necessário usar toda a sequência temporal de treinamento, o que significaria criar uma cópia da arquitetura neural para cada um dos padrões de treinamento, criando assim uma estrutura com centenas ou até mesmo milhares de neurônios, o que certamente torna o processo de treinamento bastante custoso.

Uma tentativa de amenizar este problema é truncar o histórico da rede, de modo que no instante n , usamos apenas os dados dentro de uma janela temporal com tamanho suficientemente adequado. Ainda assim, é preciso armazenar as várias cópias da estrutura neural em memória, e a propagação do erro pela cadeia inevitavelmente torna-se uma operação custosa. Além disso, corre-se o risco de perder informações relevantes para o processo de adaptação dos pesos sinápticos ao fixar um valor relativamente pequeno para o tamanho da janela temporal.

Podemos concluir, portanto, que calcular as derivadas da função custo com respeito aos parâmetros que precisam ser ajustados torna-se uma tarefa árdua por causa da presença dos laços de realimentação. Infelizmente, esta dificuldade também é vivenciada nas demais abordagens empregadas no treinamento de RNNs.

Outros obstáculos comumente enfrentados durante a adaptação dos parâmetros de uma RNN estão associados à convergência lenta do algoritmo de aprendizado, à dificuldade de escolher valores adequados para os parâmetros do algoritmo, como o tamanho do passo de adaptação, e ao desvanecimento do gradiente da função de erro à medida que este é propagado no tempo e na topologia da rede (Jaeger, 2002b). Ademais, por causa dos laços de realimentação, pequenas mudanças nos parâmetros, realizadas pelo algoritmo de treinamento, podem levar a dinâmica da rede de pontos fixos estáveis para instáveis, o que causa um súbito salto na medida de erro. Por fim, há a constante ameaça de atingir uma configuração instável da rede. Todos estes fatores acabam por ofuscar, de certa forma, a grande capacidade de processamento intrínseca a estas estruturas.

Neste cenário, as redes neurais com estados de eco (em inglês, *echo state networks*, ESNs) (Jaeger, 2001) se apresentam como uma solução promissora capaz de contornar as dificuldades de treinamento de RNNs. As ESNs são caracterizadas pelo uso de uma RNN, denominada

reservatório de dinâmicas, cujos parâmetros - pesos sinápticos das conexões de entrada e recorrentes - são definidos de maneira prévia e aleatória, sem recorrer a qualquer informação a respeito da tarefa específica a ser desempenhada pela rede, e por uma camada de saída, também chamada de camada de leitura (em inglês, *readout*), que produz as saídas da rede por meio de combinações lineares das ativações do reservatório. Ao manter fixos os pesos das conexões recorrentes presentes na camada interna da rede, o treinamento resume-se ao ajuste dos pesos do combinador linear da camada de saída, o que pode ser realizado por meio de qualquer método que permita a solução de um problema de quadrados mínimos (Jaeger, 2001; Lukosevicius e Jaeger, 2009). Desta forma, as ESNs não apenas exploram, em certa medida, o potencial de processamento de uma estrutura recorrente mas também introduzem uma significativa simplificação no processo de treinamento.

A seguir, apresentamos em detalhes os fundamentos de uma rede neural com estados de eco.

2.3 Redes Neurais com Estados de Eco

A arquitetura básica de uma rede neural com estados de eco, ilustrada na Figura 2.5, consiste de uma camada recorrente de unidades de processamento não-lineares, que dá origem a um repertório de comportamentos dinâmicos, seguida por uma camada de leitura, que combina linearmente os sinais gerados pelos neurônios do reservatório para produzir as saídas da rede.

Os estímulos de entrada, representados pelo vetor $\mathbf{u}(n) = [u_1(n) \dots u_K(n)]^T$, são linearmente combinados segundo os coeficientes especificados na matriz $\mathbf{W}^{in} \in \mathbb{R}^{N \times K}$ e transmitidos para o reservatório de dinâmicas, que é formado por N neurônios não-lineares totalmente conectados cujas ativações, denotadas por $\mathbf{x}(n) = [x_1(n) \dots x_N(n)]^T$, representam os estados da rede e são atualizadas de acordo com a seguinte expressão:

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n)), \quad (2.6)$$

onde $\mathbf{W} \in \mathcal{R}^{N \times N}$ contém os pesos das conexões recorrentes do reservatório e $\mathbf{f}(\cdot) = (f_1(\cdot), \dots, f_N(\cdot))$ denota as funções de ativação das unidades internas.

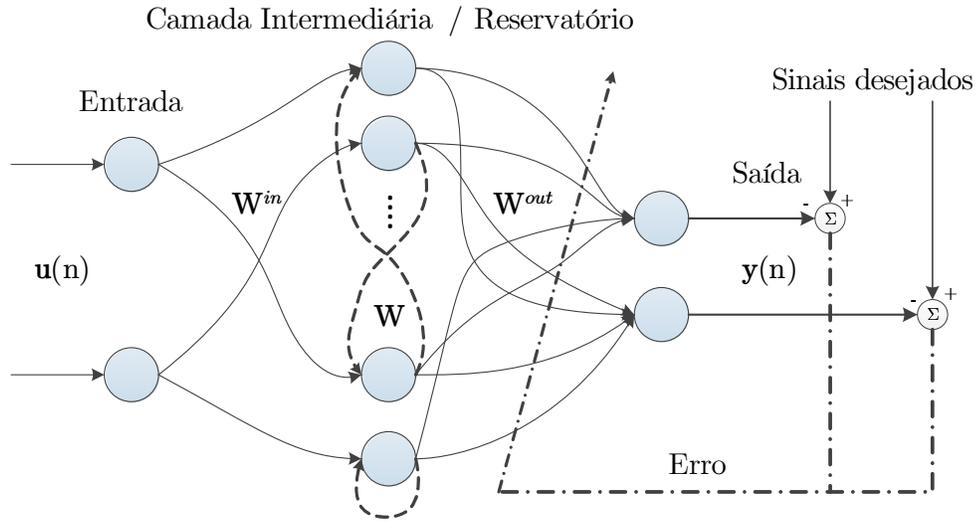


Figura 2.5: Arquitetura básica de ESNs. Apenas os pesos da saída (\mathbf{W}^{out}) são adaptados com base em um sinal de erro.

Finalmente, as saídas da rede, representadas pelo vetor $\mathbf{y}(n) = [y_1(n) \dots y_L(n)]^T$, são determinadas segundo a expressão:

$$\mathbf{y}(n+1) = \mathbf{W}^{out} \mathbf{x}(n+1), \quad (2.7)$$

onde $\mathbf{W}^{out} \in \mathcal{R}^{L \times N}$ é a matriz dos pesos de saída.

A principal ideia subjacente às ESNs é que os parâmetros da camada recorrente podem ser ajustados de maneira antecipada e independente do processo de adaptação da rede, o que significa que apenas os coeficientes da camada de saída são efetivamente ajustados com base em um sinal de referência. Além disso, devido ao caráter linear da camada de saída, os coeficientes ótimos podem ser determinados no sentido de quadrados mínimos com o auxílio de métodos de regressão linear (Jaeger, 2001; Lukosevicius e Jaeger, 2009).

Esta notável simplificação no processo de treinamento de uma estrutura recorrente pode ser trazida à fruição graças à propriedade de estados de eco (em inglês, *echo state property*,

ESP) (Jaeger, 2001), a qual garante que a ativação de cada neurônio do reservatório torna-se uma transformação não-linear do histórico recente do sinal de entrada (por isso o termo *eco*) desde que a matriz de pesos recorrentes \mathbf{W} satisfaça certos requisitos espectrais.

2.3.1 Propriedade de Estados de Eco

Em 2001, Jaeger estudou a dinâmica da arquitetura básica de uma RNN, mostrada na Figura 2.5, verificando que, sob certas circunstâncias, os estados da rede $\mathbf{x}(n)$ tornam-se assintoticamente independentes da condição inicial. Em outras palavras, se a rede é inicializada a partir de dois estados iniciais distintos $\mathbf{x}(0)$ e $\hat{\mathbf{x}}(0)$, e a mesma sequência de sinais de entrada é recebida, então as sequências de estados resultantes $\mathbf{x}(n)$ e $\hat{\mathbf{x}}(n)$ convergem para valores próximos. Quando esta propriedade é satisfeita, o efeito dos estados iniciais desaparece e a dinâmica do reservatório depende exclusivamente do histórico de entrada, de modo que a rede possui estados de eco (Jaeger, 2001).

A definição formal da ESP, feita por Jaeger (2001), baseia-se nas seguintes condições:

- os sinais de entrada são extraídos de um espaço compacto² \mathcal{U} ;
- os estados da rede estão sempre contidos em um conjunto compacto $\mathcal{A} \subset \mathbb{R}^N$ de estados admissíveis, o que significa que a operação de atualização do vetor $\mathbf{x}(n)$, mostrada na Equação (2.6), sempre preserva os estados dentro de \mathcal{A} .

Com estas definições em mente, Jaeger (2001) demonstrou duas condições suficientes com respeito à ESP. A primeira condição mostra que, caso o máximo valor singular da matriz de pesos do reservatório \mathbf{W} , em módulo, esteja dentro do círculo unitário ($|\sigma_{\max}(\mathbf{W})| < 1$), a RNN apresenta estados de eco (Jaeger, 2001, 2002a). É importante ressaltar que esta condição foi demonstrada no contexto de RNNs que não possuem realimentações da saída para o reservatório, e cujos neurônios apresentam funções de ativação do tipo tangente hiperbólica.

²O conceito de espaços compactos vincula-se à área da matemática chamada topologia, e remete a trabalhos como o de Alexandrov e Urysohn (1929).

A segunda condição estabelece a não-existência de estados de eco em função do raio espectral, i.e., do autovalor de maior módulo da matriz de pesos internos \mathbf{W} , denotado por $\rho_s(\mathbf{W})$: se $\rho_s(\mathbf{W}) > 1$, então a rede não possui estados de eco (Jaeger, 2001). Para isto, além das restrições apontadas anteriormente, assume-se que temos entrada nula.

Recentemente, Yildiz, Jaeger, e Kiebel (2012) apresentaram uma nova condição suficiente para a existência de estados de eco que evoca o conceito de estabilidade matricial de Schur, a qual, embora seja equivalente à definição menos restritiva da ESP dada por Buehner e Young (2006), por envolver conceitos bem estudados na literatura, permite a identificação de alguns tipos de matrizes que podem ser usadas para definir os pesos das conexões do reservatório com a garantia de gerar estados de eco.

2.3.2 Processo de Treinamento de uma ESN

Uma vez que a emergência de uma memória dinâmica do histórico dos sinais de entrada pode ser assegurada pela existência dos estados de eco, o que, por sua vez, depende apenas da escolha da matriz de pesos recorrentes (\mathbf{W}), é possível realizar *a priori* o projeto do reservatório, i.e., sem a influência do processo de adaptação da rede (Jaeger, 2001).

É importante destacarmos que o reservatório de dinâmicas tem como função gerar um conjunto de comportamentos dinâmicos que seja o mais diversificado possível. No entanto, ele é gerado sem a informação proveniente do sinal que se deseja aproximar, o que, de certa forma, caracteriza uma perda de capacidade de representação em relação a uma estrutura similar ajustada idealmente a partir de um critério de mínimo erro quadrático médio. Neste contexto, fica evidente o quão essencial é uma definição prévia adequada dos parâmetros da camada recorrente.

Uma maneira simples de preparar um reservatório relativamente rico em dinâmicas, elaborada por Jaeger (Jaeger, 2001), consiste em criar aleatoriamente uma matriz de pesos \mathbf{W} que apresente um certo grau de esparsidade. A intuição subjacente a esta proposta é que um padrão de conexões esparsas tende a favorecer o desacoplamento de grupos de neurônios do

reservatório, o que, por sua vez, contribui para o desenvolvimento de dinâmicas individuais, isto é, pouco correlacionadas.

Desta forma, construindo uma matriz de pesos \mathbf{W} que garanta a existência de estados de eco³ e definindo a matriz dos pesos de entrada \mathbf{W}^{in} de maneira arbitrária, uma vez que não exerce qualquer influência sobre a ESP, o processo de treinamento da ESN resume-se ao problema de determinar os coeficientes do combinador linear na saída que minimizam o erro quadrático médio entre a saída da rede e o sinal desejado, como destacado na Figura 2.5, o que pode ser resolvido com o auxílio de algoritmos de regressão linear. Esta engenhosa estratégia forma a essência das ESNs (Jaeger, 2001; Lukosevicius e Jaeger, 2009).

Neste trabalho, o conjunto de coeficientes da camada de leitura que minimiza o erro quadrático médio entre a saída da rede e o sinal desejado é obtido calculando-se a pseudoinversa de Moore-Penrose da matriz de estados $\mathbf{X} \in \mathbb{R}^{N \times T_s}$, onde T_s indica o número de amostras usadas para o treinamento, como mostra a seguinte expressão:

$$\mathbf{W}^{out} = \mathbf{d}\mathbf{X}^\dagger = \mathbf{d}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}, \quad (2.8)$$

onde a matriz $\mathbf{d} \in \mathbb{R}^{L \times T_s}$ contém todas as saídas desejadas para a sequência de treinamento.

2.4 Conclusão

Neste capítulo, apresentamos alguns aspectos fundamentais de redes neurais artificiais, como o modelo de neurônio utilizado, as possíveis arquiteturas - *feedforward* ou recorrente -, dando particular ênfase às abordagens de aprendizado supervisionado. Vimos que enquanto as FNNs estão naturalmente habilitadas para trabalhar com sinais de natureza estática, como no contexto do problema de classificação de padrões (Duda et al., 2001), as RNNs se destacam por sua capacidade de criar e acessar um dispositivo interno de memória dos sinais do sistema,

³É importante mencionar que embora a ESP esteja formulada em termos de σ_{max} , em todos os experimentos que Jaeger realizou, a condição heurística $\rho_s(\mathbf{W}) < 1$ praticamente garantiu a existência de estados de eco e, por isso, é usualmente empregada.

o que pode ser particularmente crucial no tratamento de problemas de caráter temporal e dinâmico, como predição de séries temporais (Box et al., 1994).

Entretanto, as vantagens que as RNNs oferecem são frequentemente acompanhadas por algumas dificuldades em seu processo de treinamento, tais como elevado custo computacional e instabilidade. Neste contexto, as redes neurais com estados de eco emergem como uma alternativa promissora.

O principal atrativo das ESNs está na possibilidade de aliar, em certa medida, a capacidade de processamento de uma estrutura recorrente a um processo de treinamento relativamente simples. Explorando o conceito de estados de eco, é possível abrir mão do ajuste de todos os pesos sinápticos, de modo a resumir o treinamento à adaptação do conjunto de parâmetros do combinador linear da camada de saída, cuja solução ótima pode ser obtida no sentido de mínimo erro quadrático médio por meio de métodos de regressão linear.

Interessantemente, esta abordagem recente traz à baila o espírito de uma rede desorganizada, o que nos remete ao trabalho original de Alan Turing intitulado *Intelligent Machinery*, elaborado no ano de 1948 (Turing, 1968). No próximo capítulo, revisitaremos este trabalho histórico com a finalidade de apontar os pontos de contato entre as ideias de Turing e alguns paradigmas modernos em redes neurais artificiais, como as ESNs.

Máquinas Desorganizadas de Turing:

Conexão Histórica

3.1 Introdução

Alan Mathison Turing (1912-1954) é geralmente lembrado, nos dias de hoje, por dois trabalhos históricos¹ (Copeland, 2004): seu brilhante tratamento do Entscheidungsproblem de David Hilbert (Turing, 1936) - que inclui a associação da ideia de computabilidade à entidade formal conhecida atualmente como “máquina de Turing” e a análise do que veio a ser conhecido como “problema da parada” - e o texto filosófico *sui generis* sobre inteligência artificial (Turing, 1950), no qual, entre outras, coisas, definem-se rigorosamente as bases do chamado “teste de Turing”. Pode-se afirmar com segurança que ambos os trabalhos tiveram um grande impacto no desenvolvimento daquilo que poderíamos chamar de ciência da computação ou, com um pouco mais de audácia, ciência da informação.

Entretanto, uma análise cuidadosa da trajetória científica de Turing revela que o escopo de seus interesses foi muito mais vasto (Hodges, 1992), abrangendo temas pertencentes a campos como criptografia (Copeland, 2004), teoria de probabilidades (Zabell, 1995), proces-

¹Também seria possível afirmar que Turing também é lembrado com frequência por sua contribuição ao esforço de guerra dos aliados, no contexto da análise criptográfica da comunicação alemã pela equipe de Bletchley Park (Hodges, 1992; Copeland, 2004).

samento de sinais (no âmbito do projeto Delilah) (Hodges, 1992), álgebra abstrata, projeto de hardware, projeto de software (Hodges, 1992), xadrez computacional (Copeland, 2004) e morfogênese (Turing, 1952).

Infelizmente, é fato ainda pouco conhecido que, ao elenco exposto, poderiam ser incluídas contribuições muito interessantes à área de redes neurais artificiais (Hodges, 1992; Teuscher, 2001). Em seus esforços no sentido de colaborar com o nascente ramo da inteligência artificial, Turing, tendo por base a relação entre modelos neuronais e lógica clássica (McCulloch e Pitts, 1943), apresentou um conjunto de propostas de redes neurais que se revestem de extrema criatividade e atualidade (Turing, 1968).

Este capítulo dedica-se à análise desses esforços, tendo por base tanto o texto original de Turing (Turing, 1968) quanto o trabalho de Teuscher (Teuscher, 2001), que foi importante para a reavaliação de seu papel junto à linha de desenvolvimento do paradigma conexionista. Primeiramente, buscaremos apresentar de maneira sucinta as arquiteturas propostas por Turing, assim como sua visão acerca da definição de pesos sinápticos, que mostra de modo espantoso sua profunda compreensão daquilo a que atualmente nos referimos como *machine learning*. Depois, buscaremos tecer alguns paralelos entre as propostas de Turing e dois temas de vanguarda em redes neurais artificiais - computação com reservatórios, no qual estão inseridas as ESNs, e *extreme learning machines* - os quais estão organicamente ligados à temática deste trabalho.

3.2 Redes de Turing

Em 1948, Turing entregou a Sir Charles Darwin, diretor do National Physical Laboratory, em Londres, o relatório intitulado *Intelligent Machinery*, descrevendo os principais resultados de sua pesquisa a respeito da possibilidade de máquinas computacionais exibirem comportamentos “inteligentes” de maneira semelhante ao cérebro humano (Copeland, 2004). Em suas próprias palavras:

“I propose to investigate the question as to whether it is possible for machinery to show intelligent behaviour.” (Turing, 1968, p. 1)

Neste trabalho notavelmente original, Turing antecipou diversos conceitos que se tornaram fundamentais na área de inteligência artificial. Por exemplo, o uso de métodos de busca inspirados no mecanismo de evolução natural, como, por exemplo, os algoritmos genéticos (Holland, 1975) e as estratégias evolutivas (Schwefel, 1981), fora vislumbrado por Turing no que ele chamou de “busca evolutiva ou genética” (Turing, 1968). Entretanto, a maior parte de *Intelligent Machinery* é dedicada a uma magnífica discussão sobre aprendizado de máquina, na qual Turing antecipa a abordagem conhecida atualmente como conexionismo (Teuscher, 2001).

É importante destacar que a ideia de usar uma rede composta de unidades simples de processamento interligadas segundo um padrão de conexões já havia sido explorada no trabalho de McCulloch e Pitts (1943), que, curiosamente, não é citado por Turing. No entanto, além das diferenças existentes nos tipos de redes propostas, o trabalho de Turing destaca-se em relação a seu predecessor ao desenvolver de forma pioneira o conceito de que redes desorganizadas, i.e., formadas por neurônios aleatoriamente conectados e inicializados, podem ser treinadas por meio de interferência externa a fim de realizarem uma determinada função desejada. Isto significa que Turing já previa a perspectiva de aprendizado supervisionado de redes neurais (Turing, 1968; Teuscher, 2001).

Infelizmente, *Intelligent Machinery* não recebeu a devida apreciação na época, sendo publicado apenas em 1968, quatorze anos após a morte de Turing. No entanto, graças aos esforços de Copeland (Copeland e Proudfoot, 1999; Copeland, 2004), Teuscher (Teuscher, 2001; Teuscher e Sanchez, 2001) e outros, este trabalho tem começado a ser divulgado de forma mais condizente com suas valiosas contribuições à área de inteligência artificial.

Com o propósito de investigar a emergência de comportamento inteligente em máquinas, algo que ele certamente estava convicto ser possível, Turing propôs o estudo de estruturas criadas de forma arbitrária e aleatória, compostas por certo tipo padrão de componentes, de-

nominadas máquinas (redes) desorganizadas (em inglês, *unorganized machines*, UMs). Particularmente, Turing concebeu dois tipos de redes desorganizadas, denominadas tipo-A e tipo-B, e também uma modificação na rede tipo-B, a qual não recebeu dele um nome, mas que, à semelhança de Teuscher (2001), chamaremos de tipo-BI. O grande atrativo desta última classe de redes desorganizadas reside no fato de que, por meio da existência de sinais de interferência, um agente externo (professor) pode alterar o comportamento de cada conexão presente na estrutura de modo a adaptá-la para a realização de uma função desejada. A seguir, apresentamos os três modelos de redes neurais propostos por Turing.

3.2.1 Rede tipo-A

A primeira - e também a mais simples - rede desorganizada proposta por Turing é formada por um número suficientemente elevado (N) de unidades de processamento (neurônios) similares. Cada neurônio: (i) pode estar em um dentre dois possíveis estados (0 ou 1) em cada instante de tempo; (ii) está vinculado a uma unidade central de sincronização, a partir da qual são emitidos pulsos em intervalos de tempo T ; (iii) possui exatamente dois terminais de entrada, $x_1(t)$ e $x_2(t)$, que estão conectados às saídas de quaisquer dois neurônios da rede, e produz uma única saída $y(t)$, de acordo com a expressão abaixo:

$$y(t + 1) = 1 - x_1(t)x_2(t). \quad (3.1)$$

Logo, cada neurônio computa a função lógica NAND entre as suas entradas, como mostra a Tabela 3.1.

$x_1(t)$	$x_2(t)$	$y(t + 1)$
0	0	1
0	1	1
1	0	1
1	1	0

Tabela 3.1: Atualização do estado de um neurônio em função de suas entradas - porta NAND.

A opção de Turing pelo uso de portas NAND como unidade básica das redes desorganiza-

das foi certamente motivada pelo fato de que qualquer função lógica pode ser implementada por meio de arranjos formados estritamente por este elemento de processamento (Ercegovac, Lang, e Moreno, 1998; Teuscher, 2001). Como consequência desta propriedade, é possível afirmar que qualquer máquina de Turing universal com uma fita de memória finita pode ser simulada através de uma rede tipo-A (Teuscher, 2001).

No entanto, a fim de que uma rede tipo-A apresente um comportamento desejado, é necessário projetar cuidadosamente sua topologia e escolher o número de neurônios, bem como seus respectivos estados iniciais, o que pode ser uma tarefa bastante árdua. Além disto, é importante destacar que as redes tipo-A trabalham de maneira determinística a partir de sua configuração inicial, não permitindo adaptações de seus parâmetros durante sua operação. A Figura 3.1 ilustra uma possível topologia de rede desorganizada tipo-A com cinco neurônios.

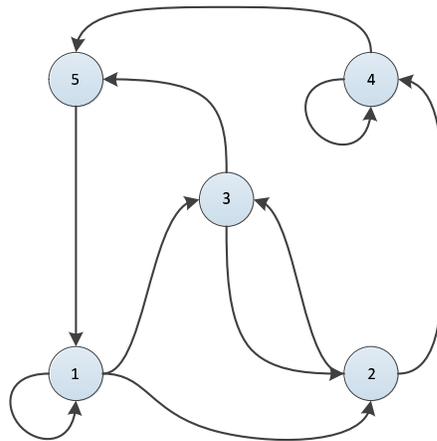


Figura 3.1: Exemplo de uma topologia típica de uma rede desorganizada tipo-A.

3.2.2 Rede tipo-B

A característica distintiva das redes tipo-B está no fato de que cada conexão entre suas unidades de processamento possui um nó de interferência que apresenta diferentes modos de operação, o que abre a possibilidade de reforçar as ligações úteis e eliminar as inúteis. Cada nó de interferência consiste de uma rede tipo-A com apenas três neurônios, a qual,

dependendo dos estados iniciais, pode: (i) inverter o sinal de entrada, (ii) interromper a transmissão de qualquer informação, fixando a saída no valor 1, ou (iii) alternadamente exibir os comportamentos (i) e (ii). Na Figura 3.2, apresentamos a representação utilizada por Turing para uma conexão tipo-B, bem como a rede tipo-A que forma o nó de interferência, enquanto a Figura 3.3 exibe a rede tipo-B correspondente à tipo-A mostrada anteriormente.

À primeira vista, as redes tipo-B não apresentam mudanças significativas em relação às máquinas tipo-A. Porém, a motivação de Turing era caminhar em direção a estruturas mais flexíveis que permitissem uma espécie de treinamento para a realização de tarefas específicas.

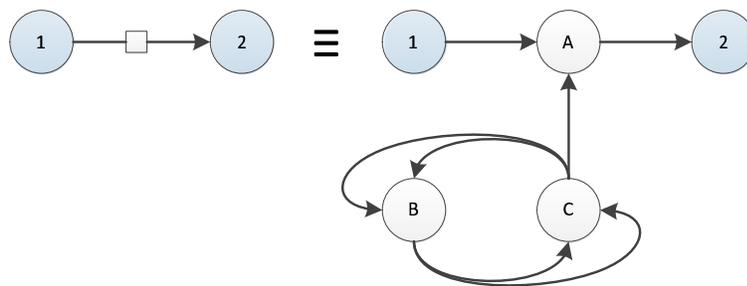


Figura 3.2: Conexão de uma rede desorganizada tipo-B.

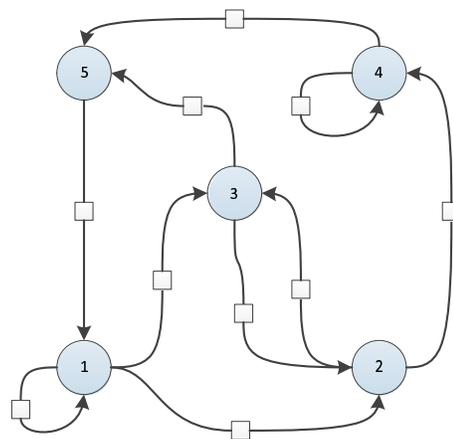


Figura 3.3: Exemplo de uma topologia típica de uma rede desorganizada tipo-B.

Neste contexto, as redes tipo-B constituem um passo intermediário: por um lado, a estruturação das conexões ainda deve ser elaborada no momento de concepção da rede, não podendo ser modificada após sua inicialização; por outro lado, surge a possibilidade de se

alterar o comportamento da rede atuando diretamente nas conexões entre os neurônios.

3.2.3 Rede tipo-BI

Finalmente, Turing propôs uma modificação na estrutura das conexões existentes nas redes tipo-B através da inserção de duas entradas de interferência, como mostrado na Figura 3.4.

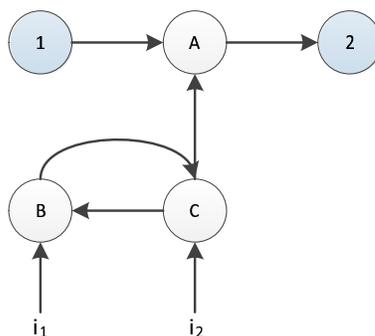


Figura 3.4: Conexão de uma rede tipo-BI.

Esta nova classe de rede desorganizada, denominada tipo-BI (Teuscher, 2001), oferece a possibilidade de ajustar o modo de operação de cada conexão da rede por meio da escolha dos valores das entradas externas. Agora, fica mais claro qual era a intenção de Turing ao introduzir as redes tipo-B: seu propósito era construir uma espécie de *switch* (nó de interferência) formado pelo mesmo tipo de unidade básica de processamento da rede e que, no final, permitisse a um agente externo (professor) adaptar seu modo de operação a fim de que a rede atingisse um comportamento desejado.

Na visão de Turing, *“the cortex of an infant is an unorganised machine, which can be organised by suitable interfering training”* (Turing, 1968, p. 15). Neste sentido, as redes tipo-BI, uma vez que trazem entradas de interferência que podem ser acessadas e modificadas por um agente externo, contemplam, em certa medida, o processo de educação (treinamento) que, de acordo com Turing, era o responsável por organizar a estrutura cerebral do ser humano em desenvolvimento, de modo a permitir o aprendizado de tarefas desejadas.

3.2.4 Discussão

Apesar de sua simplicidade em termos de formulação e arquitetura, as máquinas desorganizadas propostas por Turing são capazes de produzir comportamentos bastantes complexos. De fato, como destacado na Seção 3.2.1, as redes tipo-A podem ser utilizadas para implementar qualquer função lógica (Teuscher, 2001). Turing acreditava que, assim como ocorre com as redes tipo-A, as redes tipo-B e tipo-BI também conseguiriam reproduzir qualquer função lógica, podendo, assim, implementar uma máquina universal com uma fita de memória finita: *“In particular with a B-type unorganised machine with sufficient units one can find initial conditions which will make it into a universal machine with a given storage capacity.”* (Turing, 1968, p. 13).

No entanto, como evidenciado em (Copeland e Proudfoot, 1996), (Teuscher, 2001) e (Teuscher e Sanchez, 2001), esta afirmação não está correta. Na verdade, existem funções lógicas simples que não podem ser implementadas pelas redes tipo-B, como, por exemplo, a operação conhecida como OU-exclusivo (em inglês, *exclusive disjunction* (XOR)).

Felizmente, por meio de modificações bastante simples na estrutura da conexão tipo-B, é possível resolver este problema. Como discutido em (Copeland e Proudfoot, 1996), a alegação de Turing a respeito do poder computacional das redes tipo-B torna-se verdadeira se os possíveis modos de operação do nó de interferência forem interromper (*ii*) e transmitir, em vez do modo inverter (*i*). Então, os autores sugeriram a adição de um outro nó de interferência em cada conexão, de modo que a nova sinapse pode simplesmente passar o sinal de entrada ou interromper a transmissão. A Figura 3.5 exhibe o modelo de conexão proposto por Copeland e Proudfoot (1996).

Uma solução similar foi introduzida por Teuscher (2001): inverter o sinal de entrada antes de passá-lo ao nó de interferência. Isto foi realizado através da inserção de um outro neurônio tipo-A na conexão, como mostrado na Figura 3.6, o qual é funcionalmente equivalente à proposta anterior, porém mais simples.

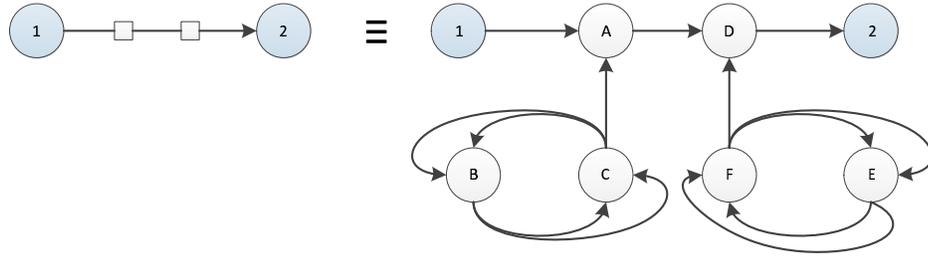


Figura 3.5: Proposta de Copeland e Proudfoot.

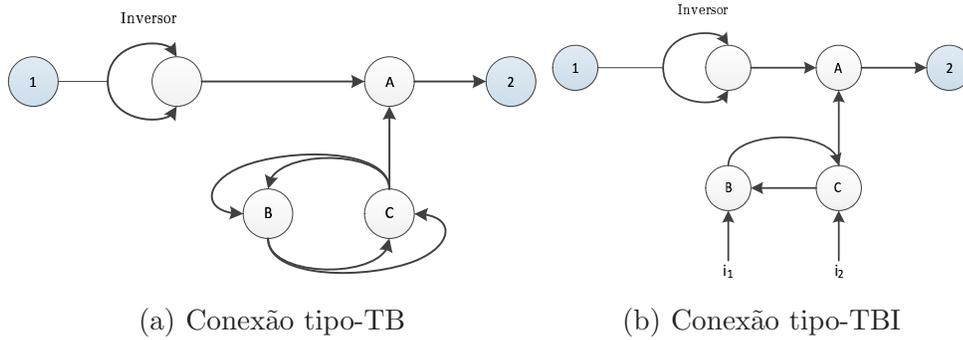


Figura 3.6: Proposta de Teuscher.

Por último, mas não menos importante, *Intelligent Machinery* trouxe outra contribuição inovadora: com o propósito de exemplificar o processo de organização de máquinas desorganizadas, Turing propôs as chamadas máquinas tipo-P, as quais possuem uma estrutura similar a uma máquina de Turing sem a fita de memória, cuja descrição inicial é incompleta e aleatória, e que recebem estímulos de prazer e dor em resposta às decisões tomadas. Aqui, pode-se perceber uma interessante relação conceitual com a abordagem de aprendizado por reforço (em inglês, *reinforcement learning*) (Sutton e Barto, 1998).

Sem dúvidas, *Intelligent Machinery* representa um marco importante no estudo de aprendizado de máquina, discutindo e desenvolvendo conceitos fundamentais nas áreas de inteligência artificial, neurociência e filosofia da mente que estavam muito à frente de seu tempo. Iremos confirmar esta constatação seguindo um caminho diferente dos esforços de Copeland, Proudfoot, Teuscher e outros, na medida em que nossa ênfase estará nos pontos de contato entre as ideias de Turing e modernas máquinas desorganizadas, em particular, as *extreme learning machines* (ELMs) e a abordagem conhecida como computação com reservatórios.

3.3 Computação com Reservatórios

As redes neurais com estados de eco, propostas por Jaeger (2001), e as *liquid state machines* (LSMs), propostas por Maass, Natschläger, e Markram (2002), oferecem uma abordagem criativa para a adaptação de estruturas neurais com realimentação e estabeleceram um novo paradigma de treinamento de RNNs conhecido como computação com reservatórios (em inglês, *reservoir computing* (RC)) (Verstraeten, Schrauwen, D’Haene, e Stroobandt, 2007; Lukosevicius e Jaeger, 2009).

Como destacado no Capítulo 2, a arquitetura padrão em RC é composta por: (i) uma camada recorrente densamente conectada de elementos de processamento não-lineares, a qual é capaz de preservar informações a respeito do histórico dos sinais de entrada; (ii) uma camada de saída adaptativa, que é responsável por combinar os sinais do reservatório (ou líquido, na nomenclatura das LSMs) com o objetivo de aproximar as saídas desejadas.

O diferencial de RC reside na possibilidade de preservar, até certo ponto, a capacidade de processamento inerente a uma estrutura recorrente sem incorrer nas desvantagens associadas às abordagens convencionais de treinamento de RNNs. Estes objetivos são alcançados de maneira elegante ao se definirem os parâmetros da camada interna sem que se faça uso de qualquer informação referente à tarefa desejada, fixando-se seus valores de acordo com algum critério pré-definido. Deste modo, o processo de treinamento consiste em adaptar somente os coeficientes da camada de saída (Lukosevicius e Jaeger, 2009). A Figura 3.7 exhibe a arquitetura básica em RC, além de destacar o fato de que o processo de adaptação com base em um sinal de erro está confinado à camada de saída.

3.3.1 Liquid State Machines

Independente e simultaneamente ao desenvolvimento das ESNs, um paradigma similar de computação com reservatório, chamado *liquid state machine* (LSM), surgiu dentro de uma abordagem conceitual próxima à neurociência, visando estudar algumas habilidades

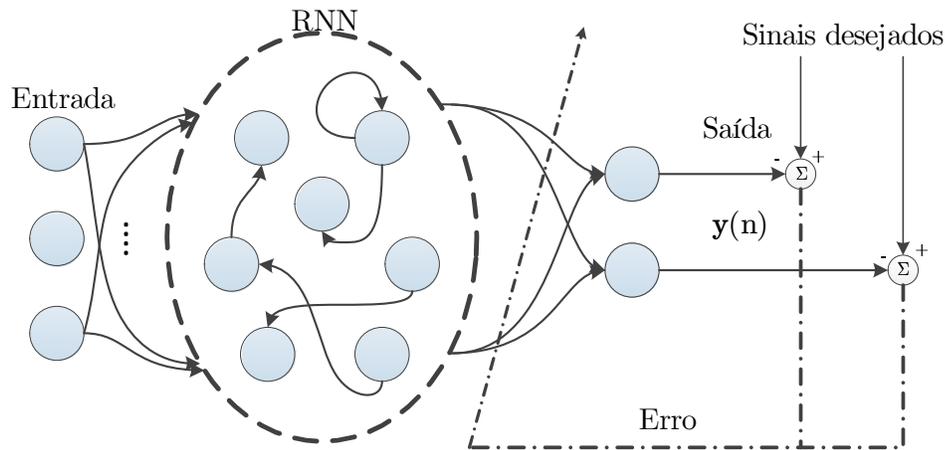


Figura 3.7: Arquitetura utilizada em RC.

computacionais do cérebro. Esta arquitetura, originalmente proposta por Maass et al. (2002), baseia-se em um modelo computacional que mapeia funções temporais de entrada, dadas por sequências de números ou bits, em sequências de saída (Maass, 2007) - o que é realizado por um filtro especial dotado de uma memória que desvanece com o tempo - e um estágio de leitura, segundo uma arquitetura similar à mostrada na Figura 3.7.

O filtro está relacionado a sistemas dinâmicos que definem o reservatório e o termo *líquido* justifica-se pelos padrões dinâmicos associados às respostas desta camada a estímulos externos. Diferentemente das ESNs, o reservatório de uma LSM é baseado em modelos neuronais mais realistas, o que abre a possibilidade de gerar espaços de estado multidimensionais para adequadamente realizar tarefas genéricas de computação. Naturalmente, o poder computacional de uma LSM depende do grau de compatibilidade entre as propriedades do líquido e o problema computacional a ser resolvido, algo que foi provado em rigorosos termos matemáticos em Maass et al. (2002), dando origem ao que Maass (2007) chamou de propriedades de separação e aproximação para LSMs. Em consonância com o espírito de RC, o líquido também exhibe um padrão esparsamente conectado entre os neurônios (Maass, Natschläger, e Markram, 2004), o que resgata alguns elementos das formulações conexionistas de Turing, como descrito na Seção 3.2. Além disso, esta abordagem estabelece uma importante conexão entre a teoria de máquinas desorganizadas e alguns aspectos estruturais de redes biológicas,

como aqueles identificados por Yamazaki e Tanaka (2007) no contexto das propriedades de organização e computação do cerebelo.

3.4 Extreme Learning Machines

ELMs são redes neurais *feedforward* com uma única camada intermediária (em inglês, *single-hidden layer feedforward neural networks* (SLFNs)) que apresentam uma abordagem de treinamento distintiva: os parâmetros dos neurônios da camada intermediária, *viz.*, os pesos de entrada e as polarizações (*biases*), são definidos de maneira aleatória e independente. Ao abrir mão de adaptar os parâmetros da camada intermediária, o processo de treinamento da rede se resume a encontrar os coeficientes ótimos do combinador linear presente na camada de saída, o que essencialmente envolve uma solução de um problema de regressão linear (Huang, Zhu, e Siew, 2004, 2006). Desta forma, as ELMs evitam a retropropagação de um sinal de erro e o uso de algoritmos iterativos (Haykin, 1996), assim como a possibilidade de convergência para mínimos locais da função de erro, o que representa um interessante progresso em termos de simplicidade e eficiência de treinamento (Huang, Zhu, e Siew, 2006).

Esta abordagem é sustentada por dois importantes resultados teóricos: (i) uma demonstração rigorosa de que existe uma SLFN cujos parâmetros da camada intermediária são aleatoriamente escolhidos, segundo qualquer distribuição de probabilidades contínua, que consegue aproximar com precisão arbitrária o sinal desejado se as funções de ativação dos neurônios desta camada forem infinitamente diferenciáveis (Huang, Zhu, e Siew, 2006); (ii) um elegante procedimento construtivo que demonstra a capacidade de aproximação universal das ELMs - o erro de aproximação do sinal desejado pode ser sempre reduzido com a inserção de um novo neurônio na camada intermediária através da escolha cuidadosa do novo peso de saída (Huang, Chen, e Siew, 2006).

Além disto, existem evidências interessantes que revelam um potencial aumento da capacidade de generalização da rede quando o vetor de parâmetros possui norma mínima (Bartlett,

1998). Neste contexto, o operador generalizado de Moore-Penrose para inversão matricial se credencia como uma solução capaz de atender ambos os requisitos (Huang, Zhu, e Siew, 2006).

A Figura 3.8 exibe a arquitetura básica de uma *extreme learning machine*, ressaltando também o fato de que somente os coeficientes da camada de saída é que são ajustados tendo como base a informação de uma medida de erro entre a saída da rede e o sinal desejado.

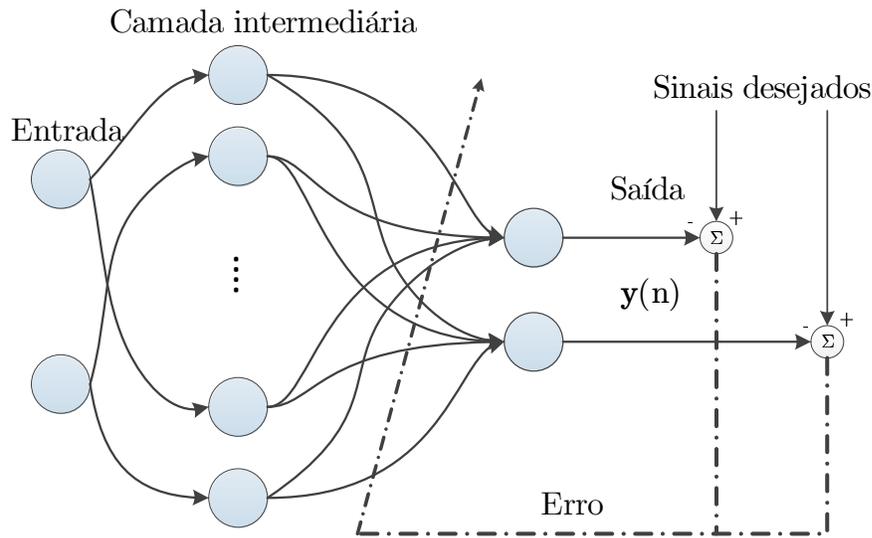


Figura 3.8: Arquitetura de uma *extreme learning machine*.

Em termos matemáticos, as saídas dos neurônios da camada intermediária podem ser expressas segundo a expressão:

$$\mathbf{x}_h(n) = \mathbf{f}^h(\mathbf{W}^h \mathbf{u}(n) + \mathbf{b}), \quad (3.2)$$

onde $\mathbf{W}^h \in \mathbb{R}^{N_h \times K}$ contém os pesos de entrada aleatoriamente definidos, sendo N_h o número de neurônios na camada intermediária e K o número de entradas da rede, e o vetor $\mathbf{b} = [b_0 \cdots b_{N_h}]^T$ especifica as polarizações de cada neurônio. Com isto, as saídas da ELM são obtidas por meio de combinações lineares das ativações dos neurônios da camada intermediária:

$$\mathbf{y}(n) = \boldsymbol{\beta} \mathbf{x}_h(n), \quad (3.3)$$

onde $\beta = [\beta_1 \dots \beta_{N_h}]$ especifica os coeficientes destas combinações. Como podemos observar, a simplicidade do treinamento é preservada uma vez que o problema de obter os valores ótimos de β_i é linear com respeito aos parâmetros.

3.5 Reflexões sobre os pontos de contato entre as redes de Turing e as modernas máquinas desorganizadas

A exposição realizada nas Seções 3.3 e 3.4 revela as similaridades conceituais e de motivação entre as abordagens de computação com reservatórios e as *extreme learning machines*. Ambas as propostas são caracterizadas pelo uso de camadas de neurônios que não são treinadas, i.e., que não têm seus parâmetros adaptados, e cujas ativações são combinadas por um *readout* linear, o que leva a significativas vantagens em termos de tratabilidade e simplicidade do processo de treinamento.

Curiosamente, estas abordagens também convergem na medida em que evocam a ideia de “desorganização”, um aspecto biológico que foi levado em consideração em trabalhos anteriores sobre redes neurais, e.g., (F. Rosenblatt, 1958) e, como discutiremos com mais detalhes, (Turing, 1968), sob o disfarce de estruturas como reservatórios (camadas intermediárias) aleatoriamente construídos. De fato, uma vez que o reservatório (ou camada intermediária) não é treinado, sempre há uma parte da estrutura das ESNs, LSMs e ELMs que permanece essencialmente desorganizada.

Há, portanto, uma notável semelhança entre a estrutura das redes de Turing e os métodos baseados em reservatório. Esta similaridade se manifesta especialmente em dois pontos: (i) na existência de uma camada recorrente com neurônios cujos padrões de conexão não são definidos segundo uma metodologia supervisionada e (ii) na presença de elementos específicos que são projetados para incorporar a informação trazida por um sinal de referência.

É preciso ressaltar que, apesar desta similaridade, a motivação por trás destes dois conjuntos de propostas é fundamentalmente diferente. Turing, como discutido na Seção 3.2, conce-

beu uma arquitetura desorganizada ao refletir sobre o processo de desenvolvimento do sistema nervoso humano, desde a embriogênese até a educação de uma criança. Por outro lado, os paradigmas relacionados à RC e às ELMs são derivados a partir de considerações de natureza mais prática, as quais surgem a partir de uma visão das redes neurais como aproximadores de funções / dinâmicas e de uma busca por um compromisso adequado no tocante ao processo de treinamento.

A propósito, convém destacar que o último ponto de vista tem sido dominante na área de redes neurais artificiais como consequência do renascimento deste campo de pesquisa nos anos 80, o que naturalmente levou a uma certa ênfase em otimização, seleção de modelos e estatística. Os trabalhos de Turing, neste contexto, tornam-se cruciais na medida em que oferecem a oportunidade de o programa de pesquisa em redes neurais harmonizar-se mais com a perspectiva de computação natural a partir da qual ele surgiu.

Com relação à necessidade de se converter a informação processada pelas redes desorganizadas, diferentes caminhos aparecem no trabalho de Turing e nas máquinas modernas. Isto se deve à própria natureza dos sinais envolvidos: Turing trabalhou com sequências booleanas, enquanto ESNs e ELMs podem recorrer a métodos clássicos de quadrados mínimos que se adequam muito bem aos números reais (e complexos). É sempre preciso ter em mente que vivemos em um mundo no qual os recursos computacionais são bastante vastos e amplamente disponíveis, o que oferece a vantagem de se poder desenvolver, com o auxílio de simulações, abordagens estocásticas de treinamento que Turing nunca teve a chance de testar. Este fato torna suas contribuições ainda mais impressionantes, uma vez que a dualidade fundamental entre desorganização e supervisão foi completamente descoberta por ele e, além disso, suas soluções para lidar com ela foram bem engenhosas.

Um ponto para potenciais sinergias entre estas duas vertentes parece ser a perspectiva de modificar uma rede de Turing para gerar uma versão booleana de uma rede neural com estados de eco e, talvez, para encontrar um equivalente à propriedade de existência de estados de eco (Jaeger, 2001). Isto poderia ser de imenso valor para simplificar os sistemas que operam sobre

alfabetos finitos, os quais tipicamente dão origem a problemas de otimização combinatória. Esta perspectiva também daria uma importante contribuição à área de processamento de sinais no contexto de campos finitos (Gutch, Gruber, Yeredor, e Theis, 2012), um ramo de pesquisa que pode trazer contribuições relevantes em aplicações que vão desde codificação de canal até análise de dados genômicos.

Como complemento ao que tem sido discutido até o momento, é importante destacar que certas descobertas recentes na teoria de sistemas complexos e neurociência também apresentam interessantes similaridades com os aspectos essenciais das propostas conexionistas de Turing. Por exemplo, a recente transposição da teoria de grafos para o estudo anatômico da conectividade cerebral (Scannell, Burns, Hilgetag, O’Neil, e Young, 1999) realçou características como: 1) regiões segregadas de processamento de informação trabalhando em paralelo; 2) uma densa estrutura de conectividade entre os neurônios em cada região de processamento; 3) um número relativamente pequeno de conexões esparsas entre diferentes regiões de processamento a fim de construir um aparato distribuído de processamento. Estas características parecem indicar a possibilidade de que grupos de neurônios conectados de acordo com padrões complexos são, em certo sentido, responsáveis por entregar informações pré-processadas a uma camada mais controlada e dedicada à fusão e processamento de dados em um nível superior, o que nos remete às estruturas básicas discutidas neste e no capítulo anterior.

Tendo em vista esta relevante conexão histórica entre os modelos conexionistas de Turing e certos paradigmas recentes de redes neurais, tomamos emprestado de Turing a designação *máquinas desorganizadas* como um termo geral que captura a essência destas propostas e unifica as abordagens de RC - ESNs e LSMs - e as ELMs (Boccatto, Soares, Fernandes, Soriano, e Attux, 2011).

É nossa mais sincera convicção que o fato de as visões de Turing, formadas ainda no período de infância das redes neurais, mostrarem estas íntimas conexões com aspectos fundamentais de modernas máquinas desorganizadas, é um verdadeiro tributo a sua habilidade de perceber os fatores essenciais subjacentes ao processamento de informação, assim como

um convite para os pesquisadores da área revisitarem com um espírito renovado o legado de fundadores como Turing.

Uma vez estabelecida esta interessante conexão com as ideias de Turing, passaremos à apresentação das demais contribuições alcançadas neste trabalho, as quais se desenvolveram no contexto das redes neurais com estados de eco. Antes, porém, é conveniente descrevermos os aspectos fundamentais dos diferentes problemas de tratamento de informação que serão empregados na análise das propostas de ESNs.

Aplicações

A fim de analisarmos de maneira adequada o potencial de estruturas recorrentes, como as ESNs, é importante realizar tal análise no âmbito de problemas que apresentem uma íntima relação com a estrutura temporal dos sinais de informação envolvidos, e para os quais a aplicação de uma abordagem baseada em modelos não-lineares seja particularmente vantajosa.

Por isso, neste trabalho serão consideradas duas tarefas de grande importância em processamento de sinais que naturalmente possuem ambos requisitos: equalização de canais de comunicação e separação de fontes, particularmente no contexto de misturas convolutivas (Haykin, 1996; Romano, Attux, Cavalcante, e Suyama, 2011). Além destas tarefas, exploraremos também dois problemas de predição de séries temporais que impõem diferentes e significativos desafios a qualquer modelo de predição em virtude da natureza dos sinais envolvidos: no primeiro caso, as séries estão associadas aos estados de um sistema dinâmico não-linear operando em regime caótico (Abarbanel, 1997), enquanto, no segundo caso, as séries referem-se a medidas de vazões mensais de diferentes bacias hidrográficas brasileiras (Ballini, 2000; Sacchi, Ozturk, Principe, Carneiro, e da Silva, 2007).

A seguir, descrevemos os fundamentos e as particularidades de cada um dos problemas tratados neste trabalho.

4.1 Equalização de Canais de Comunicação

Um sistema de comunicação é projetado a fim de permitir que informações de interesse sejam enviadas de maneira eficiente a partir de um transmissor para um receptor, usando, para tanto, um canal disponível. Na prática, comunicações bem-sucedidas inevitavelmente requerem que os dados que chegam no receptor sejam adequadamente processados, uma vez que o canal sempre introduz algum tipo de distorção sobre o sinal que ele transmite. Usualmente, esta distorção pode ser modelada em termos de um sistema cuja entrada e saída são, respectivamente, o sinal transmitido $s(n)$ e o sinal recebido $r(n)$, como mostra a Figura 4.1.



Figura 4.1: Modelo de um canal de comunicação como um sistema entrada-saída.

4.1.1 Fonte de Informação

A fonte de informação é responsável por gerar o sinal discreto que contém as informações a serem transmitidas. Uma hipótese amplamente utilizada é a de que a sequência de símbolos gerada, denotada por $s(n)$, consiste de variáveis aleatórias discretas, independentes e identicamente distribuídas (i.i.d.) cujos valores restringem-se a um alfabeto finito $\mathbb{A} = \{s_i, i = 1, \dots, S\}$, onde S é o número de símbolos do alfabeto.

O tipo de modulação empregada na transmissão define o alfabeto (Barry, Lee, e Messerschmitt, 2003; Proakis e Salehi, 2007)¹: por exemplo, uma modulação 2-PAM está associada ao alfabeto $\{+1, -1\}$, enquanto a modulação 4-QAM produz o alfabeto complexo $\{-\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j, -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j, \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j, \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j\}$, onde $j = \sqrt{-1}$.

¹Consideramos o modelo em banda base.

4.1.2 Modelos de Canais de Comunicação

A natureza e a intensidade das modificações sofridas pelos dados enviados estão essencialmente relacionadas às características do modelo do canal, o qual pode ser tanto linear quanto não-linear, e também possuir diferentes níveis de aleatoriedade. Neste trabalho, vamos considerar que os efeitos do canal sobre o sinal transmitido podem ser:

- a) Interferência inter-simbólica (IIS): provocada pelo espalhamento temporal que ocorre no meio de transmissão, que gera uma sobreposição de dados associados a diferentes instantes de amostragem.
- b) Ruído: corresponde a perturbações aleatórias no sinal recebido devido a fatores como agitação térmica e interferências eletromagnéticas.
- c) Distorções não-lineares: surgem devido à presença de não-linearidades em componentes elétricos e no próprio meio de transmissão, como ocorre, por exemplo, nas fibras ópticas (Agrawal, 2002).

Classicamente, o modelo linear de canal é utilizado para representar um sistema de comunicação (Haykin, 1996). Como a própria nomenclatura indica, este modelo não considera efeitos não-lineares, de modo que as distorções que afetam o sinal transmitido são do tipo IIS e ruído. A IIS é modelada através de uma combinação linear dos símbolos transmitidos, de acordo com a equação a diferenças²:

$$\begin{aligned} s'(n) &= h_0^* s(n) + h_1^* s(n-1) + \dots + h_{n_c-1}^* s(n-n_c+1) \\ &= [h_0^* \ h_1^* \ \dots \ h_{n_c-1}^*] \cdot \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-n_c+1) \end{bmatrix} \\ &= \mathbf{h}^H \mathbf{s}(n), \end{aligned} \tag{4.1}$$

²Note que estamos supondo um canal de memória finita. Caso contrário, a combinação envolveria, virtualmente, infinitas amostras passadas do sinal da fonte, sendo melhor representada por um filtro de resposta ao impulso infinita (em inglês, *infinite impulse response* (IIR)).

onde os parâmetros h_j^* são chamados de coeficientes do canal, n_c é o comprimento do canal, e $(\cdot)^H$ denota o operador complexo conjugado transposto. A Equação (4.1) revela que a IIS pode ser representada por meio de um filtro linear de resposta ao impulso finita (em inglês, *finite impulse response* (FIR)), cuja função de transferência pode ser escrita da forma:

$$H(z) = \sum_{j=0}^{n_c-1} h_j^* z^{-j}. \quad (4.2)$$

Por sua vez, o ruído é matematicamente modelado como um processo estocástico estacionário, branco e com distribuição gaussiana de média zero e variância σ_η^2 , que é adicionado ao sinal resultante da interferência inter-simbólica. Deste modo, o sinal efetivamente recebido é dado por

$$r(n) = s'(n) + \eta(n), \quad (4.3)$$

onde $\eta(n)$ denota o valor do ruído no instante n .

4.1.3 Estados do Canal

Por definição, os estados do canal correspondem aos possíveis valores que o sinal recebido pode assumir na ausência de ruído. No caso de um canal linear, definido pelo conjunto de coeficientes $h_0^*, \dots, h_{n_c-1}^*$, os estados do canal de dimensão m , denotados por $\mathbf{c}_i \in \mathbb{C}^m$, são obtidos exclusivamente a partir do efeito da IIS, sendo, portanto, equivalentes aos próprios vetores $\mathbf{s}'(n) = [s'(n) \ \dots \ s'(n - m + 1)]^T$.

Usando a Equação (4.1), podemos escrever:

$$\mathbf{c}_i = \mathbf{s}'(n) = \begin{bmatrix} h_0^* s(n) + h_1^* s(n-1) + \dots + h_{n_c-1}^* s(n-n_c+1) \\ h_0^* s(n-1) + h_1^* s(n-2) + \dots + h_{n_c-1}^* s(n-n_c) \\ \vdots \\ h_0^* s(n-m+1) + h_1^* s(n-m) + \dots + h_{n_c-1}^* s(n-n_c-m+2) \end{bmatrix}. \quad (4.4)$$

Definindo a matriz de convolução

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n_c-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & h_0 & \cdots & h_{n_c-2} & h_{n_c-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & h_0 & \cdots & h_{n_c-1} \end{bmatrix}, \mathbf{H} \in \mathbb{C}^{m \times (n_c+m-1)} \quad (4.5)$$

e o vetor

$$\mathbf{s}_i(n) = \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-n_c+1) \\ s(n-n_c) \\ \vdots \\ s(n-m+1) \\ \vdots \\ s(n-n_c-m+2) \end{bmatrix} \quad (4.6)$$

contendo uma possível sequência de $(n_c + m - 1)$ símbolos da fonte, cada estado do canal pode ser determinado de acordo com a seguinte expressão:

$$\mathbf{c}_i = \mathbf{H}^* \mathbf{s}_i(n). \quad (4.7)$$

Uma vez que o sinal transmitido pertence a um alfabeto finito, o número de estados também é finito e corresponde ao número de combinações de símbolos que compõem a sequência $\mathbf{s}_i(n)$. Assim, o número total de estados de dimensão m é dado por $N_{\text{estados}} = S^{n_c+m-1}$.

Finalmente, construindo a matriz $\mathbf{S} = [\mathbf{s}_1(n) \ \dots \ \mathbf{s}_{N_{\text{estados}}}(n)]$, $\mathbf{S} \in \mathbb{A}^{(n_c+m-1) \times N_{\text{estados}}}$, contendo todas as possíveis sequências de $(n_c + m - 1)$ símbolos em suas colunas, é possível

determinar todos os estados do canal através de uma única operação:

$$\mathbf{C} = \mathbf{H}^* \mathbf{S}, \quad (4.8)$$

onde as colunas de $\mathbf{C} \in \mathbb{C}^{m \times N_{\text{estados}}}$ correspondem aos estados do canal.

EXEMPLO 4.1. Canal de Fase Mínima

Considere um sinal de informação pertencente ao alfabeto $\mathbb{A} = \{+1, -1\}$ (modulação 2-PAM), o qual é transmitido através do canal linear cuja função de transferência é dada por $H(z) = 1,6 + z^{-1}$. Vamos determinar os estados do canal de dimensão $m = 2$. Neste caso, o comprimento do canal é $n_c = 2$ e o número de símbolos possíveis da fonte é $S = 2$. Logo, temos $N_{\text{estados}} = 2^{2+2-1} = 8$ estados, os quais podem ser obtidos a partir da Equação (4.8):

$$\mathbf{C} = \underbrace{\begin{bmatrix} 1,6 & 1 & 0 \\ 0 & 1,6 & 1 \end{bmatrix}}_{\mathbf{H}^*} \cdot \underbrace{\begin{bmatrix} s(n) & s(n-1) & s(n-2) \\ +1 & +1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix}}_{\mathbf{S}}^T = \begin{bmatrix} r(n) & r(n-1) \\ 2,6 & 2,6 \\ 2,6 & 0,6 \\ 0,6 & -0,6 \\ 0,6 & -2,6 \\ -0,6 & 2,6 \\ -0,6 & 0,6 \\ -2,6 & -0,6 \\ -2,6 & -2,6 \end{bmatrix}^T \quad (4.9)$$

A Figura 4.2 exhibe os estados do canal no espaço de dimensão $m = 2$. É possível perceber que os estados associados ao símbolo $s(n) = +1$ podem ser facilmente separados dos estados referentes a $s(n) = -1$ por meio de uma reta. Isto indica que um classificador linear (i.e., um filtro FIR), cuja fronteira de decisão é definida por uma reta, é capaz de corretamente identificar o símbolo atual emitido pela fonte ($s(n)$) a partir da observação do estado do

canal.

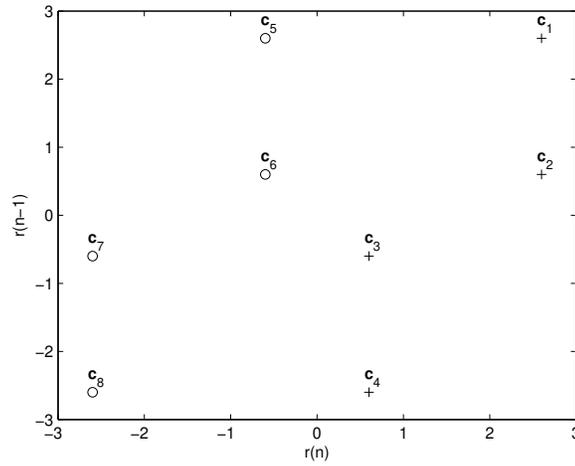


Figura 4.2: Estados de dimensão $m = 2$ do canal $H(z) = 1,6 + z^{-1}$. Os estados associados a $s(n) = +1$ e $s(n) = -1$ correspondem aos símbolos + e o, respectivamente.

EXEMPLO 4.2. Canal com Estados Coincidentes

Considere os mesmos parâmetros do cenário anterior, mas agora a função de sistema do canal é dada por $H(z) = 1 + z^{-1}$. Repetindo o mesmo procedimento, obtemos os estados de dimensão $m = 2$ deste canal:

$$\mathbf{C} = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}}_{\mathbf{H}^*} \cdot \underbrace{\begin{bmatrix} s(n) & s(n-1) & s(n-2) \\ +1 & +1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix}}_{\mathbf{S}}^T = \begin{bmatrix} r(n) & r(n-1) \\ 2 & 2 \\ 2 & 0 \\ 0 & 0 \\ 0 & -2 \\ 0 & 2 \\ 0 & 0 \\ -2 & 0 \\ -2 & -2 \end{bmatrix}^T \quad (4.10)$$

Podemos observar na Equação (4.10) que diferentes sequências de símbolos da fonte foram mapeadas no mesmo estado. Isto fica bastante evidente ao observarmos a Figura 4.3, que apresenta os estados do canal no espaço de dimensão $m = 2$. Quando este fenômeno acontece, diz-se que o canal possui estados coincidentes.

Canais que apresentam esta característica peculiar impõem significativos obstáculos a qualquer estratégia que procure recuperar o sinal de informação original, como destacado por Montalvão, Dorizzi, e Mota (1999). De fato, estruturas do tipo *feedforward*, tanto lineares quanto não-lineares, apresentam dificuldades em lidar com canais desta natureza pelo fato de a fronteira de decisão gerada, por mais flexível que seja, não conseguir distinguir os estados que foram mapeados no mesmo ponto do espaço. Neste contexto, uma estrutura dotada de realimentações surge como uma opção bastante promissora para tratar esta classe de canais de comunicação.

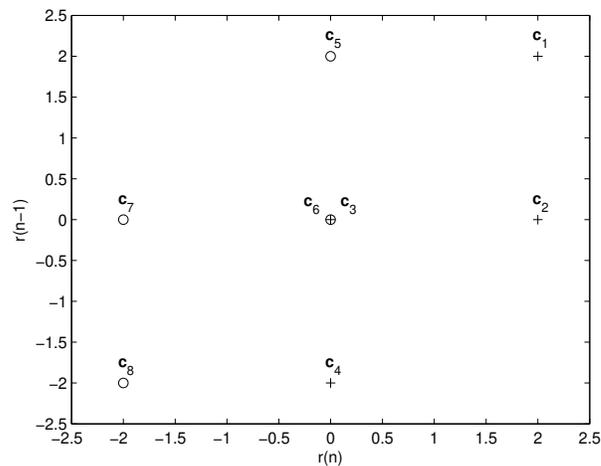


Figura 4.3: Estados de dimensão $m = 2$ do canal $H(z) = 1 + z^{-1}$. Os estados associados a $s(n) = +1$ e $s(n) = -1$ correspondem aos símbolos + e o, respectivamente.

4.1.4 Solução: Projeto de Equalizadores

Uma estratégia bem estabelecida para contrabalançar os efeitos do canal consiste em empregar um filtro especialmente projetado - um equalizador - no receptor. O papel deste

filtro deve ser o inverso daquele desempenhado pelo canal e, idealmente, sua saída deve corresponder a uma versão livre de distorções do sinal transmitido, i.e.,

$$y(n) = ks(n - d), \quad (4.11)$$

onde $y(n)$ é a saída do equalizador, k é um ganho constante e d denota o atraso de equalização. Esta condição é comumente chamada de *zero-forcing* (Haykin, 1996).

A recuperação do sinal transmitido pode ser realizada de acordo com duas abordagens: estimação de sequência ou símbolo a símbolo. No primeiro caso, um bloco de amostras recebidas é utilizado para se obter, através do critério de máxima verossimilhança, os símbolos transmitidos (Forney, 1972). Este tipo de equalizador pode ser implementado com o auxílio do algoritmo de Viterbi (Viterbi, 1967; Forney, 1973). Por outro lado, equalizadores símbolo a símbolo usam um número fixo de amostras para estimar um único símbolo a cada instante de tempo, permitindo, assim, que seus parâmetros sejam adaptados à medida que a transmissão acontece, com a finalidade de se adequar às características do canal (Haykin, 1996). Neste trabalho, consideraremos a abordagem de equalização símbolo a símbolo.

O projeto de um equalizador engloba as escolhas de: 1) uma estrutura de filtragem adequada; 2) um critério de equalização apropriado; e 3) uma abordagem eficiente de otimização para ajustar os parâmetros do filtro.

A primeira escolha é absolutamente vital, pois ela determina um limite inexorável para o desempenho do equalizador. O caráter linear ou não-linear do canal de comunicação, o grau de influência das amostras passadas sobre a saída atual do canal (i.e., a memória do canal), a natureza e intensidade do ruído - um fator inerentemente estocástico -: cada um destes fatores cria dificuldades com as quais a estrutura de filtragem deve lidar.

Um canal linear pode, em alguns casos, ser adequadamente tratado por meio de um equalizador linear, como visto no Exemplo 4.1. Contudo, contrariando o que poderíamos imaginar à primeira vista, uma estrutura não-linear pode ser decisiva para resolver certos problemas lineares de equalização (Adali, 1999). A memória do canal exerce uma profunda

influência sobre a memória requerida para o equalizador, tanto para os modelos lineares, quanto para os não-lineares, e, neste contexto, o uso de uma estrutura recorrente emerge de maneira natural. Finalmente, a estrutura do ruído existente no canal também pode exigir o uso de um equalizador não-linear - como consequência, por exemplo, de um caráter não-gaussiano bastante pronunciado (Erdogmus e Principe, 2006) - ou até mesmo de uma estrutura recorrente, caso a dependência temporal seja o elemento chave.

Por outro lado, as duas escolhas restantes gravitam em torno de um aspecto crucial: a quantidade de informação *a priori* disponível a respeito do sinal de interesse. Quando amostras deste sinal podem ser acessadas pelo algoritmo de aprendizado do equalizador, falamos de equalização supervisionada. Caso contrário, temos o cenário não-supervisionado (ou cego) de equalização (Romano et al., 2011).

4.1.5 Equalização Supervisionada

A segunda escolha equivale essencialmente a traduzir o objetivo da tarefa de equalização, representada na Equação (4.11), em termos matemáticos. Uma opção clássica tem sido empregar o critério baseado no erro quadrático médio (em inglês, *mean squared error* (MSE)), também conhecido como critério de Wiener, o qual leva à seguinte função custo:

$$J_{MSE} = E \{ [s(n-d) - y(n)]^2 \}, \quad (4.12)$$

onde $E\{\cdot\}$ corresponde ao operador estatístico de esperança. Alternativas supervisionadas para o MSE foram propostas no contexto do campo de pesquisa chamado *information-theoretic learning* (ITL) (Erdogmus e Principe, 2006). Também é importante mencionar a existência de métodos para equalização não-supervisionada (cega) (Hyvärinen et al., 2001), os quais baseiam-se em um procedimento estatístico centrado, diretamente ou indiretamente, em momentos de ordem superior a dois.

Finalmente, uma vez definidos a estrutura de filtragem e o critério de equalização, a ta-

refa restante consiste em utilizar uma abordagem de otimização para encontrar uma solução com respeito aos parâmetros livres do filtro. No contexto da formulação baseada no MSE, se a estrutura do filtro é do tipo *feedforward*, i.e., sem laços de realimentação, e linear com respeito aos seus parâmetros ajustáveis, então a função custo construída possui um único mínimo, a chamada solução de Wiener, a qual pode ser obtida por meio de um procedimento de estimação baseada em mínimos quadrados ou através de métodos iterativos como os algoritmos LMS (*least mean squares*) e RLS (*recursive least squares*) (Haykin, 1996). Por outro lado, se a estrutura de filtragem é não-linear com respeito aos parâmetros e/ou recorrente, a superfície da função custo MSE pode apresentar múltiplos ótimos locais, o que estabelece um problema de busca bem mais complexo.

Toda esta discussão nos leva a algumas considerações bastante pertinentes. Sem dúvida, é possível afirmar que, de um ponto de vista estritamente estrutural, o dispositivo de equalização mais desejável é uma estrutura não-linear e recorrente, uma vez que, neste caso, o equalizador seria capaz de lidar tanto com os requisitos de memória quanto com as exigências associadas à flexibilidade de gerar mapeamentos de entrada-saída. Todavia, da perspectiva de otimização, encontrar a solução ótima MSE para este equalizador pode ser uma tarefa bastante árdua devido tanto ao caráter multimodal da função custo quanto à necessidade de definir apropriadamente as direções e os tamanhos dos passos de ajuste nos algoritmos iterativos.

Curiosamente, como destacado anteriormente neste trabalho, a abordagem baseada nas redes neurais com estados de eco emerge como uma solução atrativa para equalização, uma vez que estas redes são não-lineares e recorrentes, tendo como diferencial o fato de não exigirem um processo complexo de otimização de seus parâmetros.

A Figura 4.4 resume os principais elementos que compõem o problema de equalização de canais de comunicação em sua versão supervisionada.

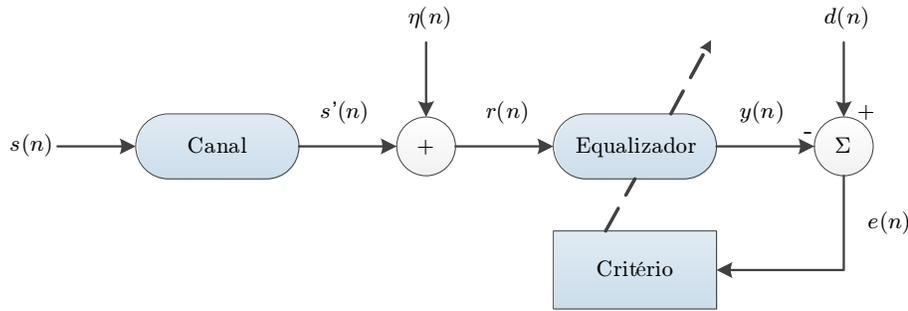


Figura 4.4: Diagrama de blocos do problema de equalização supervisionada.

4.1.6 Equalização Não-Supervisionada

O problema de elaborar um critério eficaz de equalização torna-se mais complicado quando não é possível ter acesso a um sinal de referência. Uma teoria bastante rica foi desenvolvida para lidar com este problema quando tanto o canal quanto o equalizador são sistemas lineares, recorrendo essencialmente a informações contidas em estatísticas de ordem superior do sinal recebido (Haykin, 1996; Romano et al., 2011). Esta possibilidade é motivada de maneira substancial pelos trabalhos de Benveniste, Goursat, e Ruget (1980) e Shalvi e Weinstein (1990).

O trabalho de Benveniste et al. (1980) demonstrou que se as funções densidade de probabilidade (em inglês, *probability density functions*, PDFs) do sinal transmitido e da saída do equalizador forem iguais - desde que não sejam gaussianas -, então pode-se dizer que o efeito do canal foi eliminado, i.e., a condição *zero-forcing*, exibida na Equação (4.11), foi satisfeita. Por sua vez, Shalvi e Weinstein (1990) mostraram não ser necessário formular um critério de equalização não-supervisionada com base nas PDFs dos sinais envolvidos: basta garantir que os momentos de segunda ordem do sinal transmitido e da saída do equalizador, bem como um cumulante não-nulo de ordem superior a dois, e.g., a *kurtosis* (Shalvi e Weinstein, 1990; Papoulis, 1991), sejam iguais. Inspirados por estes teoremas, vários algoritmos foram desenvolvidos para o ajuste cego dos parâmetros de equalizadores lineares explorando de maneira implícita ou explícita informações estatísticas de ordem superior, como, por exemplo, o algoritmo de módulo constante (em inglês, *constant modulus algorithm*, CMA) (Godard,

1980).

Entretanto, esta abordagem teórica não é diretamente extensível para lidar com o caso geral de um equalizador não-linear, pois é possível que as condições dos teoremas de (Benveniste et al., 1980) e (Shalvi e Weinstein, 1990) sejam atendidas sem que necessariamente o sinal recuperado seja igual ao transmitido (Ferrari, 2005; Romano et al., 2011).

O nó górdio foi cortado com a elegante estratégia proposta por Cavalcante, Montalvão, Dorizzi, e Mota (2000): usar filtros de erro de predição não-lineares (em inglês, *nonlinear prediction-error filters*, NPEFs) como equalizadores não-supervisionados. Esta ideia, que foi posteriormente desenvolvida e analisada em (Ferrari et al., 2003) e (Ferrari et al., 2008), é aplicável toda vez que o sinal desejado for composto de amostras i.i.d e o canal for linear.

Assumindo que o canal é linear e que sua resposta ao impulso contém D termos, o sinal recebido pode ser expresso da seguinte forma:

$$r(n) = h_0^*s(n) + h_1^*s(n-1) + \dots + h_{D-1}^*s(n-D+1) + \eta(n). \quad (4.13)$$

Suponha que desejemos construir um preditor genérico não-linear para estimar $r(n)$ a partir das amostras contidas em $\mathbf{r}_P(n) = [r(n-1), r(n-2), \dots, r(n-N_P)]^T$. A saída do preditor, $\hat{r}(n)$, será dada por

$$\begin{aligned} \hat{r}(n) &= F\{\mathbf{r}_P(n)\} \\ &= F\{r(n-1), r(n-2), \dots, r(n-N_P)\}, \end{aligned} \quad (4.14)$$

onde $F\{\cdot\}$ representa o mapeamento não-linear produzido pelo preditor, de modo que o erro de predição $e(n) = r(n) - \hat{r}(n)$ é dado por

$$\begin{aligned} e(n) &= h_0^*s(n) + h_1^*s(n-1) + \dots + h_{D-1}^*s(n-D+1) - F\{r(n-1), r(n-2), \dots, r(n-N_P)\} \\ &\quad + \eta(n). \end{aligned} \quad (4.15)$$

Usando a Equação (4.13), podemos escrever³

$$e(n) = h_0^*s(n) + h_1^*s(n-1) + \dots + h_{D-1}^*s(n-D+1) - P\{s(n-1), \dots, s(n-D-N_P+1)\} + \eta(n). \quad (4.16)$$

Considere que o preditor, caracterizado pelo mapeamento $P\{\cdot\}$, será projetado com o objetivo de atingir uma condição de mínimo erro quadrático médio de predição. Dada a informação disponível em $r(n-1), \dots, r(n-N_P)$, isto é, $s(n-1), \dots, s(n-N_P-D+1)$, a melhor estimativa que o preditor pode fazer é

$$\hat{r}(n) = \sum_{j=1}^{D-1} h_j^*s(n-j). \quad (4.17)$$

Neste caso, o erro de predição será necessariamente igual a

$$e^{\text{ótimo}}(n) = h_0^*s(n) + \eta(n), \quad (4.18)$$

visto que $s(n)$ não pode fazer parte de $\hat{r}(n)$, já que as amostras $s(n)$, $s(n-1)$, $s(n-2)$ etc. são, por hipótese, estatisticamente independentes. Portanto, o erro de predição ótimo oferece uma estimativa de $s(n)$, a menos de um ruído aditivo e um fator de escala / desvio de fase devido a h_0 , o que significa que o filtro de erro de predição atua como um equalizador não-supervisionado (Ferrari et al., 2008).

Por isso, o projeto de preditores eficientes é crucial para a construção de equalizadores cegos não-lineares. Neste contexto, as ESNs, que foram testadas de maneira bem-sucedida em tarefas de predição em trabalhos como (Sacchi et al., 2007), e que oferecem uma solução de compromisso entre capacidade de processamento e complexidade computacional, credenciam-se como opções promissoras. A Figura 4.5 ilustra a abordagem preditiva para equalização cega.

³Observe que ao explicitamente mostrarmos a ação do preditor em função das amostras da fonte ($s(n-1), \dots, s(n-D-N_P+1)$), optamos por mudar a representação do mapeamento para $P\{\cdot\}$.

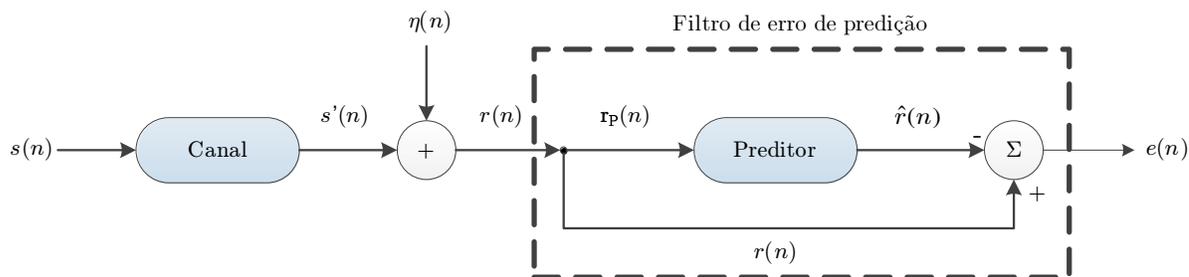


Figura 4.5: Esquema de equalização não-supervisionada baseada em previsão.

4.2 Separação de Fontes

Separação cega de fontes (em inglês, *blind source separation*, BSS) representa um problema relevante associado ao conceito de extração de informação. Em essência, BSS consiste em recuperar um conjunto de sinais de interesse (fontes) a partir de uma coleção de misturas destes sinais. O termo “cego” implica em usar uma quantidade mínima de informação a respeito da natureza das fontes e do sistema que gera as misturas. Esta tarefa de recuperar as fontes originais a partir de um conjunto de misturas tem se mostrado bastante útil e necessária em vários contextos diferentes, como processamento de sinais biomédicos, reconhecimento de padrões e comunicações digitais (Hyvärinen et al., 2001).

Curiosamente, o cérebro humano notabiliza-se por sua habilidade de isolar uma fonte de informação desejada em meio a uma mistura de sinais de voz vindos de diversas conversas e ruído de fundo, como ilustrado no famoso *cocktail party problem* (Hyvärinen et al., 2001). Entretanto, uma abordagem sistemática para o problema de BSS não estava disponível até o trabalho pioneiro de Héault, Jutten, e Ans (1985), a partir do qual uma cornucópia de diferentes estratégias para separação de fontes tem sido desenvolvida.

Apesar disto, uma importante característica se faz frequentemente presente nos trabalhos de BSS: o processo de formação das misturas é modelado segundo um sistema linear sem memória e sem ruído. Neste caso, o modelo dos dados pode ser completamente definido em termos de uma matriz de mistura $\mathbf{A} \in \mathbb{R}^{M \times F}$, de modo que o vetor observado $\mathbf{r}(n) \in \mathbb{R}^{M \times T_s}$

é dado por:

$$\mathbf{r}(n) = \mathbf{A}\mathbf{s}(n), \quad (4.19)$$

onde $\mathbf{s}(n) \in \mathbb{R}^{F \times T_s}$ contém os valores instantâneos das fontes, M indica o número de misturas, i.e., de sinais observados, F é o número de fontes de informação e T_s é o número de amostras. Ou seja, cada amostra de uma mistura em particular resulta de uma combinação linear das amostras instantâneas de todas as fontes. Usualmente, trabalha-se com o caso em que $F = M$, i.e., temos o mesmo número de fontes e de misturas.

Adicionalmente, com respeito às propriedades das fontes, uma suposição comum é que elas são sinais aleatórios estatisticamente independentes. Esta hipótese levou ao desenvolvimento de vários métodos baseados na abordagem conhecida como Análise de Componentes Independentes (em inglês, *Independent Component Analysis*, ICA) (Hyvärinen, 1999; Hyvärinen et al., 2001). Neste contexto, o objetivo principal é encontrar uma matriz \mathbf{B} que gere componentes do vetor $\mathbf{z}(n) = \mathbf{B}\mathbf{r}(n)$ tão independentes quanto possível.

Todavia, alguns exemplos naturais de sistemas misturadores inevitavelmente apresentam um caráter mais complexo. Com efeito, existem cenários práticos nos quais os sinais observados devem ser interpretados como sendo formados por uma combinação de diferentes fontes e de suas versões atrasadas. Por exemplo, um conjunto de microfones detetando diferentes fontes em um ambiente reverberante caracteriza uma situação na qual o efeito das amostras passadas não pode ser simplesmente desprezado. Conseqüentemente, o modelo linear apresentado anteriormente, embora de grande utilidade em muitos contextos, não captura fielmente as características particulares deste cenário.

4.2.1 Separação de Misturas Convolutivas

Esta limitação motivou o desenvolvimento de um modelo diferente, o qual pode ser descrito através da seguinte expressão:

$$\mathbf{r}(n) = \sum_{k=0}^{D-1} \mathbf{A}_k \mathbf{s}(n-k), \quad (4.20)$$

onde D é a memória do sistema misturador, i.e., o número de amostras atrasadas que efetivamente tem influência na composição das misturas. Assim, cada sinal observado $r_i(n)$ pode ser escrito como

$$r_i(n) = \sum_{k=0}^{D-1} \sum_{l=1}^M \mathbf{A}_k^{(i,l)} s_l(n-k), \quad (4.21)$$

onde $\mathbf{A}_k^{(i,l)}$ denota o elemento (i, l) (linha, coluna) da matriz de mistura \mathbf{A}_k . Como podemos observar, $r_i(n)$ pode ser interpretado como o resultado de uma convolução. Por isso, os sinais recebidos são usualmente chamados de misturas convolutivas.

A abordagem baseada em ICA ainda pode ser empregada neste caso, mas a hipótese de independência precisa ser expandida com suposições relacionadas à estrutura temporal das versões atrasadas da fonte. Além disso, em adição às ambiguidades de permutação e escala verificadas no modelo linear (Hyvärinen et al., 2001), também pode haver uma ambiguidade de filtragem (Haykin, 2000; Cichocki e Amari, 2002). Estas dificuldades revelam os obstáculos adicionais presentes no problema de separação de misturas convolutivas.

Apesar disto, é possível amenizar estas dificuldades ao considerarmos a abordagem baseada em predição apresentada no contexto de equalização não-supervisionada. Como discutido na Seção 4.1.6, um filtro de erro de predição suficientemente flexível é capaz de remover o caráter convolutivo do sinal recebido $r(n)$, i.e., de eliminar o efeito do canal associado às amostras atrasadas da fonte, oferecendo uma estimativa da amostra instantânea da fonte original $s(n)$, que é exatamente a única informação à qual o preditor não tem acesso.

Esta interessante propriedade é o elemento chave a ser explorado no contexto de BSS, criando um procedimento alternativo para a separação de misturas convolutivas.

Seja $y_i(n)$ a saída do preditor com N_P entradas associadas à i -ésima mistura convolutiva

$r_i(n)$, dada por

$$\begin{aligned} y_i(n) &= F\{r_i(n-1), \dots, r_i(n-N_P)\} \\ &= P\{\mathbf{s}(n-1), \dots, \mathbf{s}(n-N_P-D+1)\}. \end{aligned} \quad (4.22)$$

Neste caso, o erro de predição pode ser determinado segundo a expressão:

$$\begin{aligned} e_i(n) &= r_i(n) - y_i(n) \\ &= \Phi\{\mathbf{s}(n), \mathbf{s}(n-1), \dots, \mathbf{s}(n-N_P-D+2)\} \\ &\quad - P\{\mathbf{s}(n-1), \mathbf{s}(n-2), \dots, \mathbf{s}(n-N_P-D+1)\}, \end{aligned} \quad (4.23)$$

e, em consonância com o espírito da discussão realizada na Seção 4.1.6, é perfeitamente seguro dizer que usando uma estrutura de predição suficientemente flexível, o erro de predição tenderia a ser igual a

$$e_i^{\text{ótimo}}(n) = \tilde{x}_i(n) = \sum_{l=1}^M \mathbf{A}_0^{(i,l)} s_l(n), \quad (4.24)$$

o qual envolve precisamente os elementos que o preditor não pode usar, ou equivalentemente,

$$\tilde{\mathbf{x}}(n) = \mathbf{A}_0 \mathbf{s}(n), \quad (4.25)$$

onde $\tilde{\mathbf{x}}(n) = [\tilde{x}_1(n) \dots \tilde{x}_M(n)]^T$.

Note a forte semelhança entre a Equação (4.25) e o modelo linear sem memória apresentado na Equação (4.19). Isto significa que a influência das amostras passadas das fontes, i.e., o caráter convolutivo das misturas, pode ser virtualmente eliminado através do projeto de um banco de filtros de erro de predição não-lineares. Em outras palavras, este estágio de pré-processamento produz um conjunto de sinais $\tilde{x}_i(n)$ formados exclusivamente por combinações lineares das amostras instantâneas das fontes originais. Logo, métodos clássicos baseados em ICA podem ser prontamente utilizados para concluir a separação das fontes.

Esta abordagem baseada em predição, ilustrada na Figura 4.6, foi originalmente proposta

em (Ferrari, 2005; Suyama et al., 2007), onde a estrutura de filtros *fuzzy* foi empregada de maneira bem-sucedida para implementar os filtros de erro de predição. É importante mencionar que, em certas situações desafiadoras, pode ser pertinente inserir entradas adicionais nos NPEFs contendo amostras das outras misturas. Esta versão estendida da abordagem preditiva será útil nos cenários que vamos considerar.

Neste trabalho, as redes neurais com estados de eco farão o papel da estrutura de predição empregada para gerar os filtros de erro de predição. A motivação para o uso de ESNs reside na hipótese que estas redes sejam capazes de preservar, até certa medida, a capacidade de processamento de uma RNN para, então, poderem implementar uma estrutura de predição suficientemente flexível.

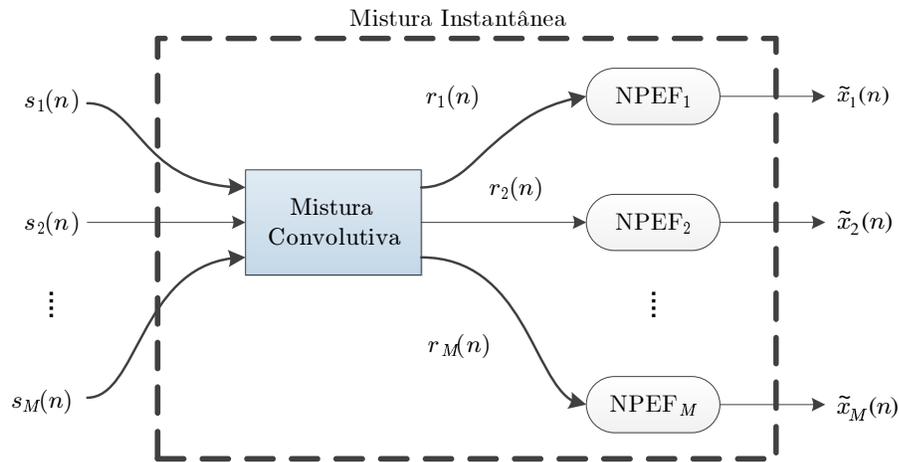


Figura 4.6: A abordagem preditiva para converter um problema de misturas convolutivas em um de misturas instantâneas.

4.3 Predição de Séries Caóticas

Sistemas dinâmicos não-lineares são caracterizados por um conjunto de equações não-lineares determinísticas (usualmente, equações diferenciais ou a diferenças), e por serem capazes de exibir uma vasta gama de comportamentos e fenômenos dinâmicos, tais como múltiplos pontos de equilíbrio, ciclos limite e caos (Strogatz, 2000).

Este último comportamento, que pode ser encontrado em vários sistemas biológicos, mecânicos, elétricos e físicos (Strogatz, 2000), está associado a soluções aperiódicas com forte sensibilidade à condição inicial, de maneira que o estado de um sistema caótico parece evoluir segundo uma trajetória aleatória a despeito do fato de que ele é essencialmente determinístico. Devido a estas características peculiares, a predição de sequências de amostras geradas por sistemas operando em regime caótico constitui uma tarefa desafiadora no âmbito de análise de séries temporais (Abarbanel, 1997; Kantz e Schreiber, 2004).

Entre as técnicas desenvolvidas para lidar com a predição de séries caóticas (Farmer e Sidorowich, 1987; Kantz e Schreiber, 2004), redes neurais recorrentes têm recebido uma especial atenção devido a sua flexibilidade, capacidade de memória e estrutura não-linear (Mandic e Chambers, 2001). Por esta razão, tal tarefa também é comumente empregada na análise de desempenho de ESNs (Jaeger, 2001; Ozturk, Xu, e Principe, 2007), e fará parte do escopo de aplicações envolvidas na análise das propostas deste trabalho.

As séries caóticas que consideraremos estão associadas a dois modelos caóticos clássicos: o mapa logístico (May, 1976) e o sistema de Lorenz (Lorenz, 1963).

4.3.1 Mapa Logístico

O primeiro modelo corresponde a um sistema dinâmico a tempo discreto cujo estado $l(n)$ evolui de acordo com a seguinte expressão:

$$l(n+1) = \mu l(n) [1 - l(n)]. \quad (4.26)$$

O parâmetro μ é uma constante positiva que exerce profunda influência sobre o comportamento do sistema. Esta influência pode ser percebida com o auxílio do diagrama de bifurcação do sistema, apresentado na Figura 4.7(a), que mostra os possíveis valores que o estado pode assumir em regime permanente em função do valor de μ .

Algumas observações interessantes podem ser extraídas da Figura 4.7(a). Por exemplo,

no intervalo $2 \leq \mu \leq 3$, o estado do sistema é atraído para um ponto de equilíbrio estável. Por outro lado, à medida que μ aumenta, o sistema produz oscilações permanentes entre dois, quatro, oito, dezesseis, etc. possíveis valores. Acima de um limiar, para determinados valores, como $\mu = 4$, o sistema exibe comportamento caótico para quase todas as condições iniciais⁴, gerando uma série temporal aperiódica com uma função de autocorrelação quase impulsiva, como podemos ver na Figura 4.7(b).

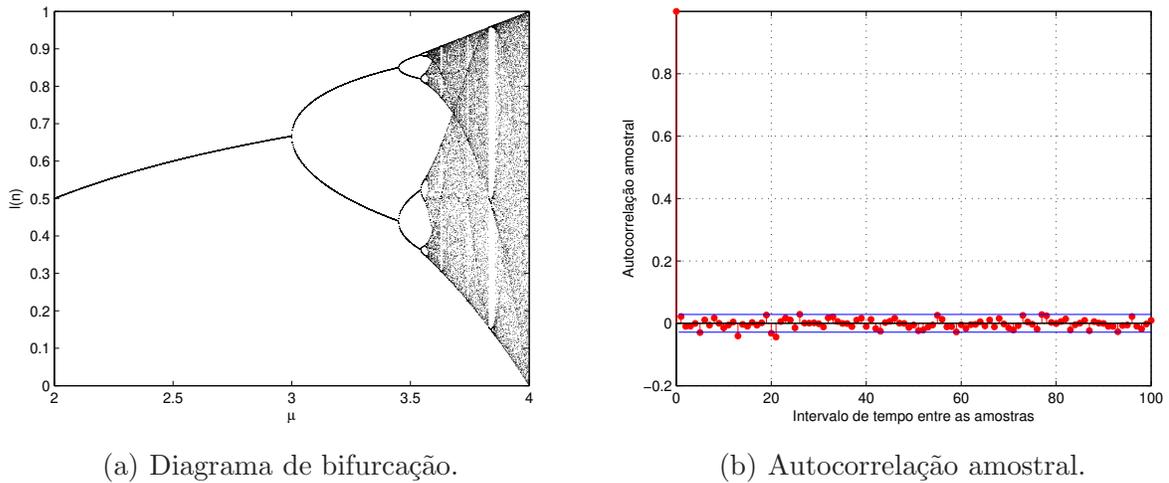


Figura 4.7: Características do mapa logístico para $\mu = 4$ e condição inicial $l(0) = 0,49$.

4.3.2 Sistema de Lorenz

O segundo modelo envolve um sistema dinâmico não-linear, tridimensional e a tempo contínuo, conhecido como oscilador de Lorenz, o qual é governado pelas seguintes equações:

$$\begin{aligned} \frac{dl_1}{dt} &= \sigma(l_2 - l_1) \\ \frac{dl_2}{dt} &= l_1(\rho - l_3) - l_2 \\ \frac{dl_3}{dt} &= l_1l_2 - \beta l_3, \end{aligned} \tag{4.27}$$

⁴E.g., $l(0) = 0,49$.

onde σ , ρ e β são constantes positivas. Para os valores $\sigma = 10$, $\beta = 8/3$ e $\rho = 28$, o sistema dinâmico exibe comportamento caótico. Na Figura 4.8, exibimos a trajetória do estado do sistema em regime caótico considerando uma taxa de amostragem de 0,025 segundo e a condição inicial $(l_1(0); l_2(0); l_3(0)) = (-0,2028; 1,815; 22,646)$, bem como a função de autocorrelação amostral da série temporal associada à primeira coordenada $(l_1(t))$. Como podemos observar, o decaimento da correlação entre as amostras é relativamente lento à medida que aumentamos o espaçamento entre elas.

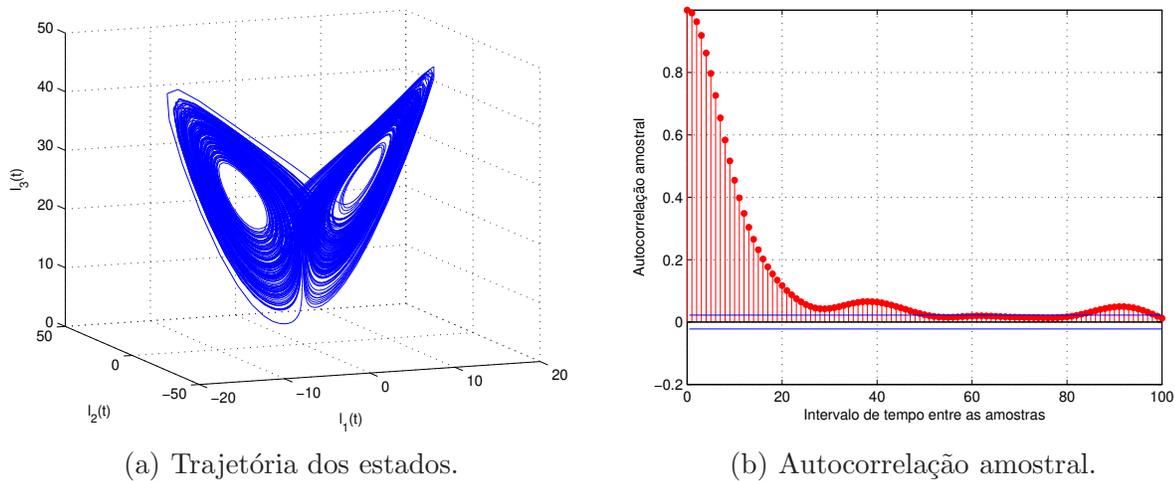


Figura 4.8: Características do sistema de Lorenz para $\sigma = 10$, $\beta = 8/3$ e $\rho = 28$.

Neste trabalho, em vez de realizarmos a predição das três coordenadas do estado do sistema, apenas abordaremos o problema de predição da primeira coordenada $(l_1(t))$. Além disso, a série temporal correspondente é normalizada para ter média nula e variância unitária, e as medidas de desempenho na predição serão computadas neste domínio.

4.4 Predição de Séries de Vazões Mensais

A predição de séries de vazões mensais constitui um importante problema no contexto de países que dependem de usinas hidroelétricas para geração de energia elétrica. Este é o caso do Brasil: mais de 80% de toda energia elétrica gerada provém deste tipo de fonte. Sob estas

circunstâncias, decisões cruciais sobre planejamento de energia, assim como estratégias de preço/cobrança, são profundamente dependentes de previsões confiáveis destas séries (Luna e Ballini, 2011; Siqueira, Boccato, Attux, e Lyra Filho, 2011). Entretanto, o caráter não-estacionário e sazonal destas séries representa um significativo obstáculo com o qual qualquer estratégia de previsão precisa lidar. Além disso, é essencial que a estrutura escolhida para a previsão seja capaz de explorar as relações existentes entre as amostras da série (Box et al., 1994).

Neste contexto, as redes neurais recorrentes surgem como alternativas promissoras graças à presença de realimentações internas que permitem a formação de um elemento de memória potencialmente benéfico em termos de qualidade de previsão. Contudo, algumas dificuldades associadas ao processo de treinamento destas redes, como discutido no Capítulo 2, complicam, até certo ponto, a aplicação prática destas ferramentas.

Neste cenário, as redes neurais com estados de eco revelam-se como opções interessantes para lidar com o problema de previsão de séries de vazões devido a sua simplicidade e potencial de processamento dinâmico. Esta perspectiva foi inicialmente explorada por Sacchi et al. (2007), mas existem muitos aspectos envolvendo o uso de ESNs neste problema que ainda precisam ser estudados: por exemplo, as potenciais vantagens de se empregar um estágio de pré-processamento dos dados ainda não foram totalmente analisadas; além disso, diferentes propostas de ESNs, tanto em termos de estratégias de projeto do reservatório de dinâmicas quanto em relação à estrutura de processamento utilizada na camada de saída, precisam ser consideradas.

4.4.1 Séries de Vazões Mensais

Séries de vazões mensais são essencialmente não-estacionárias e apresentam componentes sazonais que refletem os períodos de chuva e seca associados aos rios afluentes que dependem das estações do ano, e que acabam tendo um impacto indesejável sobre o desempenho de qualquer preditor, seja ele linear ou não-linear (Luna e Ballini, 2011). Por causa disto, uma

transformação matemática é usualmente aplicada para modificar o comportamento estatístico da série: as componentes sazonais são removidas e re-inseridas apenas ao final do processo de predição. Este procedimento de dessazonalização está descrito na Equação (4.28):

$$g'_{i,m} = \frac{g_{i,m} - \hat{\mu}_m}{\hat{\sigma}_m}, \quad (4.28)$$

onde $g_{i,m}$ denota as amostras da série de vazão original $\mathbf{g}(n)$, a qual é transformada em uma nova e dessazonalizada série $\mathbf{g}'(n)$ de média zero e variância unitária. A média e o desvio padrão associados a cada mês m são estimados segundo as expressões abaixo:

$$\hat{\mu}_m = \frac{1}{N_{\text{ano}}} \sum_{i=1}^{N_{\text{ano}}} g_{i,m} \quad (4.29)$$

$$\hat{\sigma}_m = \sqrt{\frac{1}{N_{\text{ano}}} \sum_{i=1}^{N_{\text{ano}}} (g_{i,m} - \hat{\mu}_m)^2}, \quad (4.30)$$

onde $g_{i,m}$ denota a vazão no ano $i = 1, 2, \dots, N_{\text{ano}}$, e no mês $m = 1, 2, \dots, 12$.

As sequências de medidas mensais de vazões associadas a diferentes localidades podem apresentar características bem distintas devido à diversidade dos respectivos comportamentos hidrológicos. A fim de ilustrarmos este fato, exibimos na Tabela 4.1 os valores da média e do desvio padrão (m^3/s) das séries de vazões dos postos de Furnas, Emborcação, Água Vermelha e Sobradinho, referentes ao período de 1931 a 1990. Os valores de vazões mensais dos principais postos brasileiros estão disponíveis no sítio eletrônico do Operador Nacional do Sistema Elétrico - http://www.ons.org.br/operacao/vazoes_naturais.aspx.

Série	Média ($\hat{\mu}$)	Desvio Padrão ($\hat{\sigma}$)
Furnas	942,0444	620,2873
Emborcação	495,6861	366,4537
Água Vermelha	$2,0944 \times 10^3$	$1,3198 \times 10^3$
Sobradinho	$2,7937 \times 10^3$	$2,0139 \times 10^3$

Tabela 4.1: Média e desvio padrão do histórico de séries de vazões mensais.

Como podemos observar na Tabela 4.1, as séries são bem diferentes tanto em suas medidas médias quanto nas respectivas flutuações das vazões ao longo do período 1931-1990.

4.5 Conclusão

Neste Capítulo, apresentamos os conceitos fundamentais dos problemas de extração e processamento de informação que compõem o repertório de tarefas que serão abordadas com auxílio das redes neurais com estados de eco ao longo deste trabalho: equalização de canais de comunicação, tanto no contexto supervisionado quanto no caso cego, separação de fontes, predição de séries caóticas e predição de séries de vazões mensais.

Em todos os casos, o uso de ESNs se mostra interessante em virtude do compromisso que estas redes estabelecem entre capacidade de gerar mapeamentos flexíveis de entrada-saída, inclusive com o auxílio de um dispositivo de memória, e custo computacional de adaptação dos parâmetros livres da arquitetura neural.

Nos capítulos seguintes, serão descritas e analisadas as novas propostas envolvendo três aspectos fundamentais de uma ESN: (1) a estrutura da camada de leitura, (2) o critério de adaptação dos parâmetros da camada de saída e (3) a estratégia de projeto do reservatório.

Nova Arquitetura de ESN

De acordo com a estratégia de treinamento que caracteriza as redes neurais com estados de eco, em vez de ajustarmos todo o conjunto de pesos sinápticos, somente os coeficientes do combinador linear da camada de saída é que precisam ser adaptados, o que configura uma vantagem marcante. Além disso, a ideia de manter fixos os pesos das conexões do reservatório confere agilidade ao processo de treinamento. Por outro lado, isto acaba por reduzir a capacidade de processamento da rede em comparação com uma estrutura similar idealmente ajustada. Portanto, como discutido no Capítulo 2, as ESNs podem ser vistas como uma solução de compromisso entre desempenho e simplicidade.

Entretanto, o caráter linear da camada de saída, aliado ao emprego do critério de mínimo erro quadrático médio para a adaptação dos parâmetros livres, limita a capacidade de a estrutura explorar de maneira efetiva a informação estatística dos sinais provenientes das dinâmicas não-lineares geradas no reservatório. Esta constatação motiva o desenvolvimento de uma linha de pesquisa dedicada à camada de saída das ESNs, tendo como objetivo investigar e propôr novas maneiras de combinar os sinais do reservatório a fim de aproximar com maior precisão o sinal desejado.

Uma possibilidade interessante para contornar esta limitação é o emprego de estruturas não-lineares na camada de saída. Todavia, é crucial que a estrutura escolhida guarde uma

dependência linear com respeito aos parâmetros livres, pois, neste caso, é possível encontrar uma solução fechada no sentido de quadrados mínimos (ou Wiener) (Haykin, 1996), de modo a preservar a simplicidade do treinamento da rede. Levando em consideração estes fatos, propomos o uso de uma estrutura do tipo filtro de Volterra (Mathews, 1991) na camada de saída de uma ESN.

Antes de procedermos à descrição detalhada da nova proposta, apresentaremos um breve resumo das principais ideias existentes na literatura para a camada de saída de uma ESN.

5.1 Propostas Existentes para a Camada de Saída

A camada de saída - ou *readout* - de uma rede neural com estados de eco tem como função básica mapear os sinais criados no reservatório ($\mathbf{x}(n)$) nos sinais de saída representados em $\mathbf{y}(n)$, visando aproximar um conjunto de respostas desejadas ($\mathbf{d}(n)$), o que, conceitualmente, corresponde a uma tarefa supervisionada bastante frequente em aprendizado de máquina (Bishop, 2006).

A proposta inicial para a camada de saída e, sem dúvida, a mais utilizada, consiste em empregar um combinador linear (Jaeger, 2001), como destacado na Seção 2.3. O uso de uma estrutura linear tem como atrativo sua eficiência e simplicidade de treinamento, uma vez que a solução ótima para os coeficientes da combinação linear pode ser obtida analiticamente. Vale ressaltar que Lukosevicius e Jaeger (2009) discutem algumas formas diferentes de expressar a solução analítica, tendo em vista a preocupação com a estabilidade numérica das operações envolvidas em cada uma delas. Neste trabalho, a obtenção dos pesos de saída é feita através da Equação (2.8).

Outras opções interessantes surgem no contexto de uma rede neural com estados de eco adaptativa: neste caso, os pesos do combinador linear na camada de saída devem ser ajustados de maneira *online*, isto é, à medida em que novos sinais de entrada são coletados. Para isto, uma primeira abordagem consistiria em minimizar o erro quadrático médio através de

ajustes no vetor de coeficientes segundo o gradiente estocástico da função de erro, o que nos remete ao algoritmo LMS, um método clássico em filtragem adaptativa (Haykin, 1996). Como alternativas ao LMS, temos o algoritmo de mínimos quadrados recursivo (em inglês, *recursive least squares* (RLS)) (Haykin, 1996) e o algoritmo *backpropagation decorrelation* (Steil, 2004).

Seguindo uma filosofia diferente, Shi e Han (2007) propõem o emprego de uma metodologia baseada em *Support Vector Machines* (SVM) para o treinamento de uma ESN, e a utilizam no contexto de predição de séries temporais. Basicamente, a ideia é enxergar o reservatório de dinâmicas como o *kernel* temporal para a SVM, de modo que a estrutura linear de saída pode ser treinada usando as mesmas funções de perda e regularização existentes no âmbito de SVM.

Com o objetivo de ampliar a capacidade de processamento da camada de saída, o uso de uma estrutura não-linear emerge como uma possibilidade interessante a ser investigada. Em (Maass et al., 2002), (Bush e Anderson, 2005) e (Babinec e Pospíchal, 2006), uma rede MLP foi empregada como *readout* de LSMs e ESNs, respectivamente. Contudo, embora as redes MLP possuam uma maior flexibilidade para gerar o mapeamento dos sinais do reservatório na saída desejada, seu treinamento traz significativas complicações quando comparado ao de um combinador linear, ao ponto de, em algumas aplicações, levar a um desempenho inferior em relação à ESN original.

Interessantemente, como destacado no Capítulo 3, as ELMs são capazes de contornar estes aspectos indesejáveis de treinamento de uma rede *feedforward* e ainda oferecer flexibilidade para gerar o mapeamento desejado. Logo, o uso combinado das duas redes desorganizadas, o que significa construir uma arquitetura híbrida na qual uma ELM desempenha a função de *readout* de uma ESN, emerge como uma perspectiva promissora. Esta ideia foi originalmente proposta por (Butcher, Verstraeten, Schrauwen, Day, e Haycock, 2010) e (Butcher, Verstraeten, Schrauwen, Day, e Haycock, 2013)¹.

¹Mais precisamente, a arquitetura híbrida proposta por Butcher et al. (2010, 2013) usa duas ELMs na camada de saída, as quais recebem, respectivamente, as entradas da ESN e os estados de eco.

É possível também realizar o treinamento de uma ESN de maneira não-supervisionada, i.e., sem utilizar explicitamente um sinal desejado, tendo em mãos apenas uma informação, ou medida, acerca do desempenho atual do modelo, o que caracteriza um problema de treinamento vinculado ao paradigma denominado aprendizado por reforço (Lukosevicius e Jaeger, 2009). Alguns exemplos de aplicações desta metodologia para o treinamento de ESNs em alguns problemas, principalmente no contexto de controle de sistemas, podem ser encontrados em (Xu, Lan, e Príncipe, 2005), (Devert, Bredeche, e Schoenauer, 2008), (Jiang, Berry, e Schoenauer, 2008a) e (Jiang, Berry, e Schoenauer, 2008b).

Finalmente, outras perspectivas dignas de menção envolvem: a combinação de diferentes *readouts*, como feito por Bush e Anderson (2006), e a formação de uma hierarquia de *readouts*, na qual as saídas de um nível superior na cadeia servem de coeficientes para combinar as saídas de um nível inferior, como proposto em Jaeger (2007).

A seguir, apresentamos a proposta de uma nova arquitetura de ESN, na qual o combinador linear da saída é substituído pela estrutura de um filtro de Volterra.

5.2 Filtro de Volterra

Seja $\mathbf{x}(n)$ o vetor de estados da rede, o qual determina as entradas do filtro de Volterra. A l -ésima saída da rede $y_l(n)$ é obtida por meio da combinação de expansões polinomiais dos sinais de entrada, como revela a expressão abaixo (Mathews, 1991):

$$\begin{aligned}
 y_l(n) = & v_0^{(l)} + \sum_{i=1}^N v_1^{(l)}(i)x_i(n) + \sum_{i=1}^N \sum_{j=1}^N v_2^{(l)}(i,j)x_i(n)x_j(n) \\
 & + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N v_3^{(l)}(i,j,k)x_i(n)x_j(n)x_k(n) + \dots,
 \end{aligned} \tag{5.1}$$

onde $v_o^{(l)}(k_1, \dots, k_o)$ denota os coeficientes do combinador não-linear, denominados Volterra *kernels*, com $o = 1, \dots, M_o$ representando a ordem dos termos polinomiais correspondentes. M_o assume um valor finito quando a expansão polinomial é truncada. Podemos dizer que a

série de Volterra representa uma extensão da ideia da série de Taylor, uma vez que permite a modelagem de respostas dotadas de memória. Conseqüentemente, o filtro de Volterra é uma ferramenta que pode ser utilizada para modelar uma ampla classe de sistemas não-lineares, assim como a série de Taylor se mostra útil para expressar uma variedade de funções não-lineares.

É possível utilizar uma notação vetorial para simplificar a Equação (5.1), de modo a obter uma expressão para a saída do filtro bastante familiar à que encontramos no cenário de filtragem linear. Seja $\mathbf{x}_v(n) \in \mathbb{R}^{N_{ker} \times 1}$ o vetor contendo os termos polinomiais referentes aos produtos cruzados entre as entradas “lineares” presentes em $\mathbf{x}(n) = [x_1(n) \dots x_N(n)]^T$ até a M_o -ésima ordem, e $\mathbf{V} \in \mathbb{R}^{N_{ker} \times L}$,

$$\mathbf{V} = \begin{bmatrix} v_0^{(1)} & v_0^{(2)} & \dots & v_0^{(L)} \\ v_1(1)^{(1)} & v_1(1)^{(2)} & \dots & v_1(1)^{(L)} \\ \vdots & \vdots & & \vdots \\ v_1(N)^{(1)} & v_1(N)^{(2)} & \dots & v_1(N)^{(L)} \\ v_2(1,1)^{(1)} & v_2(1,1)^{(2)} & \dots & v_2(1,1)^{(L)} \\ \vdots & \vdots & & \vdots \\ v_2(1,N)^{(1)} & v_2(1,N)^{(2)} & \dots & v_2(1,N)^{(L)} \\ \vdots & \vdots & & \vdots \\ v_2(N,N)^{(1)} & v_2(N,N)^{(2)} & \dots & v_2(N,N)^{(L)} \\ v_3(1,1,1)^{(1)} & v_3(1,1,1)^{(2)} & \dots & v_3(1,1,1)^{(L)} \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (5.2)$$

a matriz contendo os *kernels* do filtro de Volterra, onde a l -ésima coluna contém o conjunto de coeficientes $v_o^{(l)}(k_1, \dots, k_o)$, $o = 1, \dots, M_o$, associados à l -ésima saída da rede. Logo, as saídas do filtro de Volterra podem ser determinadas segundo a expressão abaixo:²

$$\mathbf{y}(n) = \mathbf{V}^T \mathbf{x}_v(n). \quad (5.3)$$

²A Equação (5.3) também pode ser utilizada para computar de maneira simultânea as saídas do filtro de Volterra para múltiplos instantes de tempo. Neste caso, basta construir a matriz $\mathbf{X}_v = [\mathbf{x}_v(0) \mathbf{x}_v(1) \dots \mathbf{x}_v(T_s - 1)]$, onde T_s é o número de amostras.

Com base na Equação (5.3), podemos afirmar que o filtro de Volterra, embora seja uma estrutura não-linear, apresenta uma dependência linear com respeito aos parâmetros ajustáveis $v_o^{(l)}(k_1, \dots, k_o), o = 1, \dots, M_o$. Portanto, é possível obter uma solução fechada para o conjunto de parâmetros que minimiza a esperança do erro quadrático entre as saídas do filtro e os valores desejados. Conseqüentemente, todo o arcabouço conceitual de filtragem ótima e o conceito de solução de Wiener podem ser estendidos para o filtro de Volterra, de modo que seus coeficientes ótimos são dados por

$$\mathbf{V} = \mathbf{R}_{\mathbf{x}_v}^{-1} \mathbf{p}_{\mathbf{x}_v \mathbf{d}}, \quad (5.4)$$

onde $\mathbf{R}_{\mathbf{x}_v} = E\{\mathbf{X}_v \mathbf{X}_v^T\} \cong \frac{1}{T_s} \sum_{n=1}^{T_s} \mathbf{x}_v(n) \mathbf{x}_v^T(n)$ e $\mathbf{p}_{\mathbf{x}_v \mathbf{d}} = E\{\mathbf{X}_v \mathbf{d}^T\} \cong \frac{1}{T_s} \sum_{n=1}^{T_s} \mathbf{x}_v(n) \mathbf{d}^T(n)$, sendo $\mathbf{X}_v = [\mathbf{x}_v(0) \ \mathbf{x}_v(1) \ \dots \ \mathbf{x}_v(T_s - 1)]$, $\mathbf{d} = [\mathbf{d}(0) \ \dots \ \mathbf{d}(T_s - 1)]$ a matriz com os sinais que desejamos obter nas saídas do filtro e T_s o número de amostras de treinamento.

Contudo, não podemos confundir esta solução de Wiener e os termos nela envolvidos com aquela existente para o caso de um filtro linear. No cenário linear, apenas as estatísticas de segunda ordem do vetor de entrada são utilizadas na matriz de autocorrelação, que possui uma estrutura do tipo Toeplitz (Haykin, 1996). Já no caso do filtro de Volterra, a matriz $\mathbf{R}_{\mathbf{x}_v}$ possui os seguintes elementos:

$$\mathbf{R}_{\mathbf{x}_v} = \begin{bmatrix} E\{x_1^2(n)\} & \dots & E\{x_1(n)x_N(n)\} & | & E\{x_1^3(n)\} & \dots & E\{x_1(n)x_N^2(n)\} & \dots \\ E\{x_2(n)x_1(n)\} & \dots & E\{x_2(n)x_N(n)\} & | & \vdots & & \vdots & \vdots \\ \vdots & & \vdots & | & & & & \\ E\{x_N(n)x_1(n)\} & \dots & E\{x_N^2(n)\} & | & E\{x_N(n)x_1^2(n)\} & \dots & E\{x_N^3(n)\} & \dots \\ E\{x_1^3(n)\} & \dots & E\{x_1^2(n)x_N(n)\} & | & E\{x_1^4(n)\} & \dots & E\{x_1^2(n)x_N^2(n)\} & \dots \\ \vdots & & \vdots & | & \vdots & & \vdots & \vdots \\ E\{x_N^2(n)x_1(n)\} & \dots & E\{x_N^3(n)\} & | & E\{x_N^2(n)x_1^2(n)\} & \dots & E\{x_N^4(n)\} & \dots \\ \vdots & & \vdots & | & \vdots & & \vdots & \vdots \end{bmatrix}.$$

Observe que o primeiro quadrante destacado na matriz $\mathbf{R}_{\mathbf{x}_v}$ corresponde precisamente à matriz de autocorrelação do caso linear, que contém as estatísticas de segunda ordem do sinal de entrada. Além disso, podemos perceber que surgem estatísticas de ordem superior, o que significa que o filtro de Volterra, ao considerar os termos polinomiais, consegue explorar com maior profundidade as características estatísticas do vetor de estados da rede. Com efeito, utilizando um filtro de Volterra de ordem $M = 2$, passamos a considerar estatísticas de $\mathbf{x}(n)$ até a 4ª ordem; de ordem $M = 3$, usamos as estatísticas de $\mathbf{x}(n)$ até a 6ª ordem; para uma ordem M qualquer, surgem estatísticas de ordem $2M$.

Esta evidência, aliada ao fato já mencionado de que a relação não-linear estabelecida entre a entrada e a saída, por meio da série truncada de Volterra, é linear com respeito aos parâmetros ajustáveis $v_o^{(l)}(k_1, \dots, k_o)$, encoraja o uso do filtro de Volterra como uma alternativa interessante às estruturas lineares para a camada de saída de uma rede neural com estados de eco.

No entanto, a estrutura de Volterra traz uma nova preocupação: à medida que o número de estados de eco (N) aumenta, o número de coeficientes tende a crescer dramaticamente. Com efeito, isto se torna evidente ao observarmos a expressão para o número de *kernels* não-ambíguos N_{ker} (e.g., usamos somente $x_1(n)x_2(n)$, descartando $x_2(n)x_1(n)$):

$$N_{ker} = 1 + \underbrace{N}_{1^{\text{a}} \text{ ordem}} + \underbrace{\frac{N(N+1)}{2}}_{2^{\text{a}} \text{ ordem}} + \underbrace{\frac{N(N+1)(N+2)}{6}}_{3^{\text{a}} \text{ ordem}} + \dots \quad (5.5)$$

Como podemos notar, com o aumento da ordem máxima dos *kernels* que pretendemos utilizar, o respectivo número de coeficientes que precisam ser ajustados tende a ser severamente ampliado. Esta possibilidade de rápida expansão do número de parâmetros traz constantes preocupações referentes à aplicação prática do filtro de Volterra junto a ESNs.

Com o propósito de amenizar este problema, decidimos utilizar uma estratégia capaz de reduzir o número de sinais que efetivamente são transmitidos para a camada de saída, a

qual se baseia em uma técnica clássica de compressão denominada Análise de Componentes Principais (em inglês, *Principal Component Analysis*, PCA) (Kendall, 1975; Jolliffe, 1986; Hyvärinen, 1999; Hyvärinen et al., 2001). Assim, em vez de transmitirmos todos os estados de eco, utilizamos apenas um número relativamente pequeno de componentes principais, o que diminui significativamente a quantidade de pesos do filtro de Volterra que precisam ser ajustados sem que isto necessariamente acarrete uma perda excessiva de informação.

5.3 Análise de Componentes Principais

Seja $\mathbf{X} \in \mathbb{R}^{N \times T_s}$ a matriz contendo as ativações das unidades internas do reservatório considerando T_s amostras de treinamento. Assumindo que a média do vetor de estados de eco é zero, a matriz de covariância dos estados pode ser estimada por $\hat{\mathbf{C}} = \mathbf{X}\mathbf{X}^T/T_s$.

A partir da decomposição em autovalores e autovetores de $\hat{\mathbf{C}}$, é possível construir a matriz $\mathbf{Q} \in \mathbb{C}^{N \times N_{pc}}$ que contém os autovetores associados aos N_{pc} maiores autovalores de $\hat{\mathbf{C}}$, os quais representam as direções onde a energia dos dados está mais concentrada. Assim, as componentes principais $\mathbf{q}_i(n)$, $n = 1, \dots, T_s$ são obtidas através da projeção dos estados da rede nas direções especificadas em \mathbf{Q} , como mostra a expressão a seguir (Kendall, 1975; Jolliffe, 1986) ³:

$$\mathbf{q}_i(n) = \mathbf{Q}_i^T \mathbf{x}(n), \quad i = 1, \dots, N_{pc}. \quad (5.6)$$

A perspectiva do uso de PCA é motivada pela observação de que existe um grau não desprezível de redundância linear entre os estados da rede. Com efeito, o estudo conduzido por Ozturk et al. (2007) já evidenciou a ocorrência de uma forte correlação entre tais sinais, o que sugere que podemos reter um número relativamente pequeno de componentes principais de $\mathbf{x}(n)$ para representar a porção mais significativa das dinâmicas geradas pelo reservatório.

Neste contexto, os autovalores de $\hat{\mathbf{C}}$ oferecem uma valiosa assistência para a escolha de quantas componentes principais são necessárias: uma vez que a soma dos N_{pc} maiores

³É importante mencionar que, embora tenhamos utilizado uma estimativa *offline* da matriz de covariância, poderíamos ter feito o PCA baseado em uma estimação *online* desta matriz (Haykin, 1996).

autovalores de $\hat{\mathbf{C}}$, dividida pela soma total de todos os autovalores, representa a porcentagem da energia capturada nas N_{pc} direções correspondentes, deve-se escolher o valor de N_{pc} que leve a um compromisso apropriado entre uma efetiva redução de dimensionalidade e um aceitável erro quadrático médio de compressão.

Finalmente, a aplicação de PCA também se mostra pertinente na medida em que reduz a complexidade das operações subsequentes, além de poder diminuir o ruído, uma vez que os dados deixados de lado, i.e., aqueles não aproveitados nas direções principais escolhidas, provavelmente portam certo grau de energia associado a este fator aleatório. Em suma, nossa proposta é utilizar a técnica de PCA e o filtro de Volterra a fim de alcançar, por um lado, compressão e uma conseqüente redução no número de coeficientes que precisam ser ajustados, e, por outro lado, uma exploração mais efetiva da informação presente nos estados de eco, com o propósito de aprimorar a aproximação do sinal desejado. A Figura 5.1 mostra a nova proposta de arquitetura de rede neural com estados de eco.

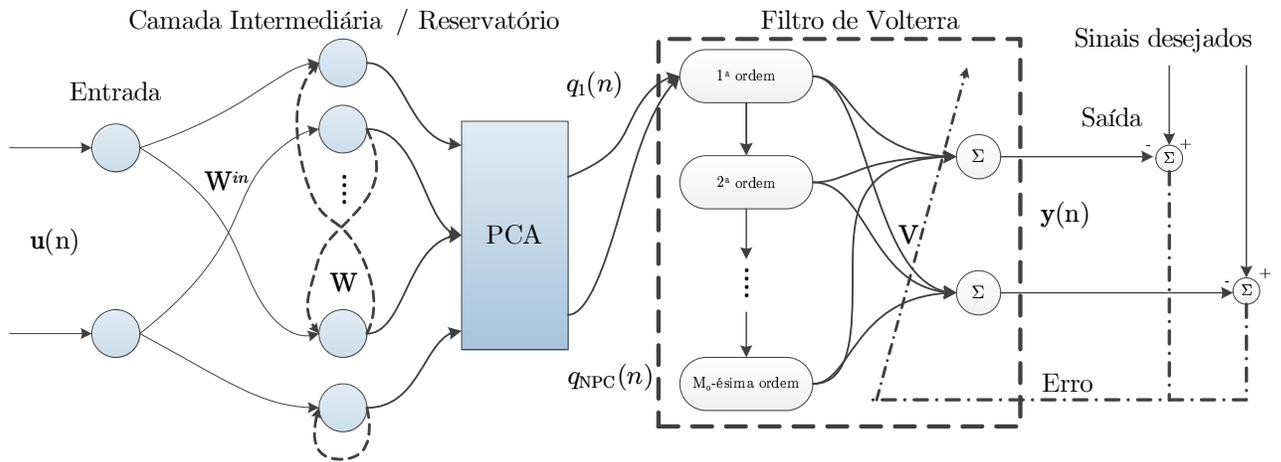


Figura 5.1: Nova arquitetura de ESN: camada de saída formada por um estágio de PCA e por um filtro de Volterra.

Outro aspecto importante a ser destacado é que a arquitetura proposta é flexível com respeito a diferentes estratégias de projeto do reservatório, i.e., pode ser utilizada em conjunto com qualquer método de criação das matrizes de pesos associadas à camada recorrente, i.e., W^{in} e W .

5.4 Resultados Experimentais

Nesta seção, analisaremos as potenciais vantagens adquiridas com o uso da nova camada de saída não-linear, em comparação com a arquitetura original de uma ESN, no âmbito dos problemas apresentados no Capítulo 4.

5.4.1 Metodologia

O desempenho obtido pelas diferentes ESNs é avaliado em termos do erro quadrático médio entre o sinal desejado ($d(n)$) e a saída oferecida pelas redes, $y_{ESN}(n)$, o qual é computado segundo a expressão:

$$\text{MSE} = \frac{1}{T_s} \sum_{i=1}^{T_s} [d(i) - y_{ESN}(i)]^2, \quad (5.7)$$

onde T_s denota o número total de amostras utilizadas na etapa de treinamento e/ou teste.

A fim de obter uma medida consistente de desempenho no que se refere à qualidade de aproximação do sinal desejado, são realizados N_{exp} experimentos independentes para cada modelo estudado e a média dos valores de MSE (em inglês, *average mean squared error* (AMSE)) é determinada:

$$\begin{aligned} \text{AMSE} &= \frac{1}{N_{\text{exp}}} \sum_{k=1}^{N_{\text{exp}}} \text{MSE}_k \\ &= \frac{1}{N_{\text{exp}}} \frac{1}{T_s} \sum_{k=1}^{N_{\text{exp}}} \sum_{i=1}^{T_s} [d(i) - y_{ESN}(i)]_k^2, \end{aligned} \quad (5.8)$$

onde MSE_k denota o erro quadrático médio obtido no k -ésimo experimento. Adicionalmente, o desvio padrão do conjunto de valores de MSE obtidos nestes experimentos também é determinado, o qual oferece uma noção da variabilidade do desempenho alcançado por cada rede.

Com respeito ao reservatório de dinâmicas, consideramos duas possibilidades: a ideia original de Jaeger (2001) e a estratégia proposta por Ozturk et al. (2007). Como apontado no Capítulo 2, a proposta de Jaeger (2001) é criar uma matriz de conectividade aleatória e

esparsa que satisfaça a propriedade de estados de eco. Um exemplo particular oferecido em (Jaeger, 2001), e que exploramos neste trabalho, define os valores dos pesos das conexões recorrentes que compõem a matriz \mathbf{W} de acordo com a seguinte regra:

$$w_{ij} = \begin{cases} 0,4 & \text{com probabilidade } 0,025 \\ -0,4 & \text{com probabilidade } 0,025 \\ 0 & \text{com probabilidade } 0,95 \end{cases} \quad (5.9)$$

Mais recentemente, Ozturk et al. (2007) propuseram uma maneira não-supervisionada e analítica de gerar o reservatório de uma ESN. Primeiramente, os autores concentraram suas atenções na versão linearizada da equação de atualização da rede (2.6) em torno do estado atual $\mathbf{x}(n)$ a cada instante de tempo n :

$$\mathbf{x}(n+1) = \mathbf{W}^{in} \mathbf{u}(n+1) + \mathbf{W} \mathbf{x}(n). \quad (5.10)$$

A função de transferência deste sistema é dada por:

$$\frac{X(z)}{U(z)} = (z\mathbf{I} - \mathbf{W})^{-1} \mathbf{W}^{in} = \frac{\text{adj}(z\mathbf{I} - \mathbf{W})}{\det(z\mathbf{I} - \mathbf{W})} \mathbf{W}^{in}, \quad (5.11)$$

de modo que os polos são obtidos resolvendo $\det(z\mathbf{I} - \mathbf{W}) = 0$. A solução, portanto, equivale aos autovalores da matriz de pesos do reservatório \mathbf{W} (Ozturk et al., 2007).

Tendo esta perspectiva em mente, a proposta de Ozturk et al. (2007) é projetar o reservatório de uma ESN de modo que o sistema linearizado, mostrado na Equação (5.10), apresente uma distribuição uniforme de seus pólos dentro do círculo unitário no plano complexo. A finalidade disto é permitir que a dinâmica do sistema explore uniformemente diferentes constantes de tempo, o que também, em certo sentido, tende a descorrelacionar os estados da rede, os quais formam as funções base a serem combinadas na camada de saída na tentativa de aproximar o sinal desejado.

Em outras palavras, a ideia que motiva esta abordagem é que, na ausência de qualquer informação a respeito da saída desejada, o procedimento mais prudente é espalhar uniformemente os pólos do sistema linearizado, na tentativa de prover boas aproximações para mapeamentos arbitrários, o que evoca o arcabouço conceitual subjacente aos chamados filtros de Kautz (Ozturk et al., 2007).

Adicionalmente, os autores também introduziram o uso da entropia média dos estados (em inglês, *average state entropy*, ASE) como medida da riqueza do reservatório de uma ESN. Utilizando esta nova métrica, foi verificado por eles que a matriz de pesos do reservatório \mathbf{W} , projetada com a finalidade de espalhar os polos do sistema linearizado uniformemente, tende a produzir um vetor de estados $\mathbf{x}(n)$ com entropia média maior que a obtida com a estratégia original de Jaeger (2001).

Logo, juntando as duas principais ideias do trabalho de Ozturk et al. (2007), temos uma estratégia alternativa para criar o reservatório de dinâmicas de uma ESN. Em termos matemáticos, a matriz \mathbf{W} deve assumir a forma canônica:

$$\mathbf{W} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{N-1} & -a_N \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad (5.12)$$

de maneira que o polinômio característico de \mathbf{W} seja:

$$\begin{aligned} I(s) &= \det(s\mathbf{I} - \mathbf{W}) = s^N + a_1 s^{N-1} + \dots + a_N \\ &= (s - p_1)(s - p_2) \dots (s - p_N), \end{aligned} \quad (5.13)$$

onde p_i 's denotam os autovalores de \mathbf{W} , que correspondem aos pólos do sistema linearizado, e a_i 's são os coeficientes do polinômio característico de \mathbf{W} . Desta forma, ao definir as posições

dos pólos p_i , os quais devem ser distribuídos de forma uniforme em uma circunferência de raio $\rho_s < 1$, podemos usar a Equação (5.13) para encontrar os coeficientes a_i , a fim de, finalmente, definir os pesos do reservatório segundo (5.12). Nos experimentos realizados, utilizamos um raio espectral igual a $\rho_s = 0,8$.

Deste ponto em diante, usaremos as abreviações R-ESN, ASE-ESN, R-PVESN e ASE-PVESN como referência à rede original de Jaeger (2001), ao modelo proposto por Ozturk et al. (2007), e à nova arquitetura com os reservatórios projetados segundo a ideia de esparsidade (R-PVESN), e de acordo com o critério baseado na entropia média dos estados da rede (ASE-PVESN), respectivamente.

Outro aspecto relevante é que, na arquitetura proposta, a camada de leitura consiste de um filtro de Volterra de terceira ordem, sem, contudo, considerar os termos quadráticos (2ª ordem). Esta escolha é motivada pelo fato de tais elementos afetarem a simetria ímpar do sinal de entrada, particularmente no contexto do problema de equalização. Finalmente, é importante mencionar que a técnica de PCA é aplicada tanto no treinamento quanto no teste da rede, com a diferença que, no último caso, em vez de determinar as N_{pc} direções preferenciais e projetar os estados da rede nelas, a mesma matriz de direções \mathbf{Q} determinada na fase de treinamento é utilizada. Deste modo, assim como as matrizes de pesos sinápticos, a matriz de direções principais \mathbf{Q} constitui um elemento imutável da rede uma vez que ela é treinada.

5.4.2 Equalização Supervisionada

O processo de treinamento das ESNs é realizado com base em $T_s = 1100$ símbolos da fonte de informação $s(n)$, a qual é composta de amostras i.i.d pertencentes ao alfabeto $\{+1, -1\}$ (modulação 2-PAM), sendo que as 100 primeiras amostras servem apenas para inicializar a rede, de modo a remover possíveis efeitos transitórios e não são consideradas no cálculo do MSE. O mesmo número de símbolos é empregado na etapa de teste das ESNs.

Em todos os casos considerados, cada ESN procura estimar a informação original $s(n)$ a

partir de uma única amostra recebida $r(n)$, o que significa que não há atraso de equalização ($d = 0$), e que o número de entradas e saídas das redes é igual a um ($K = L = 1$), o que caracteriza um cenário desafiador do ponto de vista de equalização (Romano et al., 2011). Além disso, não consideramos a presença de ruído, de maneira que a entrada da ESN $u(n)$ é precisamente o sinal $s'(n)$ resultante do efeito da IIS, como mostrado na Equação (4.1).

Em todos os experimentos envolvendo a arquitetura proposta, o número de neurônios no reservatório permanece fixo e igual a $N = 40$, enquanto o número de componentes principais pode variar.

Primeiro Cenário

O primeiro canal que iremos considerar é caracterizado pela função de sistema $H(z) = 0,5 + z^{-1}$. Embora este canal pareça ser relativamente simples, ele não pode ser equalizado com o auxílio de filtros lineares quando o atraso de equalização é nulo, pois, neste caso, os estados correspondentes não podem ser linearmente separados.

A fim de ilustrar as peculiaridades deste cenário, mostramos na Figura 5.2 os estados do canal de dimensão $m = 2$ junto com a fronteira de decisão do equalizador ótimo bayesiano, cuja descrição pode ser encontrada no Apêndice A, a qual possui nitidamente um forte caráter não-linear, reforçando a necessidade do uso de uma estrutura não-linear de equalização.

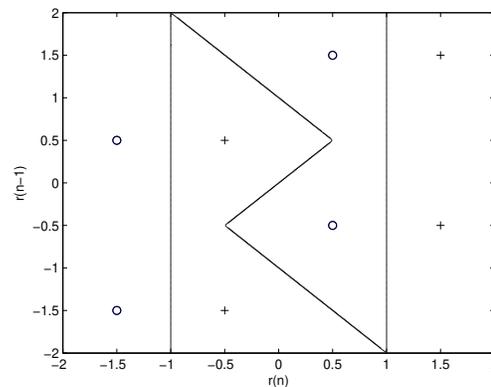


Figura 5.2: Estados do canal $H(z) = 0,5 + z^{-1}$ e a fronteira de decisão do equalizador ótimo bayesiano. Os estados associados a $s(n) = +1$ e $s(n) = -1$ correspondem, respectivamente, aos símbolos $+$ e o .

As redes neurais com estados de eco foram treinadas e testadas de acordo com a metodologia descrita na seção anterior e o desempenho obtido por cada arquitetura é apresentado na Tabela 5.1 considerando um conjunto de $N_{\text{exp}} = 20$ experimentos independentes. Os valores mostrados entre parênteses correspondem aos desvios padrão.

$H(z) = 0,5 + z^{-1}$		AMSE	
Rede	Parâmetro	Treinamento	Teste
R-ESN	$N = 10$	$2,66(\pm 3,22) \cdot 10^{-1}$	$2,72(\pm 3,27) \cdot 10^{-1}$
	$N = 40$	$1,87(\pm 2,65) \cdot 10^{-5}$	$2,07(\pm 3,03) \cdot 10^{-5}$
	$N = 60$	$2,37(\pm 2,19) \cdot 10^{-5}$	$2,74(\pm 2,58) \cdot 10^{-5}$
ASE-ESN	$N = 10$	$1,24(\pm 0,60) \cdot 10^{-1}$	$1,28(\pm 0,59) \cdot 10^{-1}$
	$N = 40$	$1,65(\pm 0,97) \cdot 10^{-2}$	$2,54(\pm 3,27) \cdot 10^{-2}$
	$N = 60$	$9,56(\pm 7,08) \cdot 10^{-3}$	$1,72(\pm 2,88) \cdot 10^{-2}$
R-PVESN	$N_{pc} = 3$	$4,26(\pm 4,34) \cdot 10^{-3}$	$4,36(\pm 4,36) \cdot 10^{-3}$
	$N_{pc} = 5$	$6,28(\pm 9,72) \cdot 10^{-5}$	$7,38(\pm 11,2) \cdot 10^{-5}$
	$N_{pc} = 6$	$2,56(\pm 3,80) \cdot 10^{-6}$	$3,06(\pm 4,60) \cdot 10^{-6}$
	$N_{pc} = 10$	$3,34(\pm 4,02) \cdot 10^{-10}$	$2,06(\pm 4,86) \cdot 10^{-8}$
ASE-PVESN	$N_{pc} = 3$	$1,02(\pm 0,78) \cdot 10^{-2}$	$1,06(\pm 0,78) \cdot 10^{-2}$
	$N_{pc} = 5$	$1,09(\pm 0,81) \cdot 10^{-3}$	$1,58(\pm 1,78) \cdot 10^{-3}$
	$N_{pc} = 6$	$4,09(\pm 4,04) \cdot 10^{-4}$	$8,61(\pm 17,7) \cdot 10^{-4}$
	$N_{pc} = 10$	$4,71(\pm 2,47) \cdot 10^{-6}$	$6,63(\pm 18,2) \cdot 10^{-4}$

Tabela 5.1: Valores AMSE obtidos com cada ESN considerando o canal $H(z) = 0,5 + z^{-1}$.

Algumas conclusões interessantes podem ser extraídas da Tabela 5.1. Primeiramente, é possível observar que o desempenho das ESNs é aprimorado quando o número de neurônios presentes no reservatório, ou o número de componentes principais, aumenta.

Em segundo lugar, comparando os valores AMSE obtidos pela ASE-ESN com aqueles alcançados pela ASE-PVESN, ficam evidentes os benefícios conquistados através do uso de uma camada de saída não-linear. De fato, considere o caso em que $N = 60$ para a primeira rede, e $N_{pc} = 6$ para a rede proposta, o que significa que o número de pesos de saída que precisam ser ajustados é aproximadamente igual (60 e 62 pesos, respectivamente): nesta situação, a rede proposta leva a um desempenho significativamente superior.

Além disso, é possível observar na Tabela 5.1 que há uma disparidade entre os valores

AMSE obtidos pela R-ESN e pela ASE-ESN, o que sugere que, para este cenário em particular, projetar a matriz de pesos do reservatório segundo a proposta de Jaeger (2001) conduz a um desempenho melhor. Esta observação é corroborada quando comparamos os valores AMSE associados a cada versão da nova arquitetura proposta: como mostrado na Tabela 5.1, o modelo proposto alcança o melhor desempenho quando o reservatório é projetado de forma similar à R-ESN, o que, mais uma vez, indica o potencial da arquitetura proposta.

Um último comentário pertinente está relacionado à variação dos valores MSE obtidos com cada rede: é possível inferir, a partir da Tabela 5.1, com o auxílio dos valores dos desvios padrão, que o modelo proposto, mesmo nos piores casos, oferece um aumento significativo de desempenho, tanto para a sequência de treinamento quanto para a sequência de teste, quando comparado com as redes R-ESN e ASE-ESN.

Segundo Cenário

Seja $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$ a função de sistema que descreve o canal. De acordo com Proakis e Salehi (2007), este é o canal linear com três coeficientes que introduz as mais severas distorções sobre o sinal transmitido. Além disso, no caso em que a dimensão dos estados do canal é $m = 1$, ocorre a superposição de estados, i.e., existem estados coincidentes, os quais, como discutido no Exemplo 4.2, comprometem de forma significativa o desempenho de equalizadores *feedforward*. Neste contexto, o emprego de estruturas recorrentes, como as ESNs, torna-se uma alternativa bastante interessante.

Os resultados médios obtidos com cada ESN a partir de um conjunto de $N_{\text{exp}} = 20$ experimentos independentes são apresentados na Tabela 5.2.

Os valores AMSE exibidos na Tabela 5.2 revelam que o uso de uma camada de saída não-linear foi relevante do ponto de vista de desempenho. De fato, ao estabelecermos uma comparação entre as redes R-ESN e R-PVESN, ou então entre a ASE-ESN e a ASE-PVESN, verificamos que a nova arquitetura sempre alcança os melhores resultados. Particularmente, os valores AMSE obtidos pela R-PVESN são muito menores que os demais, o que indica que

a combinação do reservatório esparsos (Jaeger, 2001) com a estrutura proposta para a camada de leitura - PCA seguido por um filtro de Volterra - é a mais adequada para este cenário.

$H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$		AMSE	
Rede	Parâmetro	Treinamento	Teste
R-ESN	$N = 10$	$3,03(\pm 1,07) \cdot 10^{-1}$	$3,08(\pm 1,13) \cdot 10^{-1}$
	$N = 40$	$4,73(\pm 1,51) \cdot 10^{-2}$	$5,18(\pm 1,61) \cdot 10^{-2}$
	$N = 60$	$1,72(\pm 0,74) \cdot 10^{-2}$	$2,01(\pm 0,68) \cdot 10^{-2}$
ASE-ESN	$N = 10$	$2,55(\pm 0,95) \cdot 10^{-1}$	$2,59(\pm 0,98) \cdot 10^{-1}$
	$N = 40$	$9,64(\pm 4,74) \cdot 10^{-2}$	$1,05(\pm 0,43) \cdot 10^{-1}$
	$N = 60$	$6,24(\pm 2,86) \cdot 10^{-2}$	$7,70(\pm 3,42) \cdot 10^{-2}$
R-PVESN	$N_{pc} = 3$	$7,19(\pm 1,96) \cdot 10^{-2}$	$7,53(\pm 2,01) \cdot 10^{-2}$
	$N_{pc} = 5$	$2,30(\pm 1,82) \cdot 10^{-3}$	$2,46(\pm 1,93) \cdot 10^{-3}$
	$N_{pc} = 6$	$5,56(\pm 5,95) \cdot 10^{-4}$	$6,60(\pm 7,20) \cdot 10^{-4}$
	$N_{pc} = 10$	$9,00(\pm 8,40) \cdot 10^{-7}$	$1,01(\pm 1,88) \cdot 10^{-5}$
ASE-PVESN	$N_{pc} = 3$	$1,53(\pm 0,48) \cdot 10^{-1}$	$1,57(\pm 0,51) \cdot 10^{-1}$
	$N_{pc} = 5$	$4,37(\pm 2,02) \cdot 10^{-2}$	$4,83(\pm 2,27) \cdot 10^{-2}$
	$N_{pc} = 6$	$1,90(\pm 0,88) \cdot 10^{-2}$	$2,30(\pm 1,13) \cdot 10^{-2}$
	$N_{pc} = 10$	$6,36(\pm 2,53) \cdot 10^{-4}$	$1,17(\pm 1,98) \cdot 10^{-2}$

Tabela 5.2: Valores AMSE obtidos com cada ESN para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$.

Terceiro Cenário

Por fim, consideraremos um caso em que o canal não apenas é responsável por misturar amostras atrasadas do sinal de informação (i.e., IIS), mas também por introduzir distorções não-lineares sobre o sinal transmitido. Em particular, vamos usar um modelo de canal em que a IIS e a parte não-linear do canal estão separadas: primeiro, o sinal transmitido atravessa um filtro FIR cuja função de sistema é dada por $H(z) = 0,5 + z^{-1}$, para então sofrer uma transformação não-linear descrita pela seguinte expressão:

$$y_{\text{canal}}(n) = y_{\text{lin}}(n) - 0,8y_{\text{lin}}^2(n) - 0,3y_{\text{lin}}^3(n), \quad (5.14)$$

onde $y_{\text{lin}}(k)$ denota a saída da parte linear do canal.

Este canal pode ser visto como uma extensão daquele empregado no primeiro cenário, e,

para qualquer valor de atraso de equalização, os respectivos estados não podem ser linearmente separados, o que ressalta a necessidade de uma estrutura não-linear para o equalizador.

Apresentamos na Tabela 5.3 os valores AMSE obtidos por cada ESN considerando $N_{\text{exp}} = 20$ experimentos independentes. Porém, diferentemente dos cenários anteriores, utilizamos somente $N = 20$ neurônios na camada recorrente da arquitetura proposta.

Canal não-linear		AMSE	
Rede	Parâmetro	Treinamento	Teste
R-ESN	$N = 10$	$4,54(\pm 2,19) \cdot 10^{-1}$	$4,60(\pm 2,20) \cdot 10^{-1}$
	$N = 40$	$5,24(\pm 5,52) \cdot 10^{-3}$	$5,35(\pm 5,34) \cdot 10^{-3}$
	$N = 60$	$1,09(\pm 0,80) \cdot 10^{-3}$	$1,66(\pm 1,54) \cdot 10^{-3}$
	$N = 100$	$4,40(\pm 4,50) \cdot 10^{-4}$	$8,26(\pm 6,64) \cdot 10^{-4}$
ASE-ESN	$N = 10$	$1,77(\pm 0,88) \cdot 10^{-1}$	$1,80(\pm 0,86) \cdot 10^{-1}$
	$N = 40$	$9,83(\pm 4,31) \cdot 10^{-3}$	$1,75(\pm 2,25) \cdot 10^{-2}$
	$N = 60$	$3,98(\pm 2,36) \cdot 10^{-3}$	$1,27(\pm 2,76) \cdot 10^{-2}$
	$N = 100$	$1,49(\pm 0,81) \cdot 10^{-3}$	$5,29(\pm 3,13) \cdot 10^{-3}$
R-PVESN	$N_{pc} = 3$	$9,05(\pm 6,43) \cdot 10^{-2}$	$9,50(\pm 7,02) \cdot 10^{-2}$
	$N_{pc} = 5$	$5,19(\pm 8,38) \cdot 10^{-5}$	$5,96(\pm 9,56) \cdot 10^{-5}$
	$N_{pc} = 8$	$4,49(\pm 8,85) \cdot 10^{-9}$	$1,11(\pm 2,11) \cdot 10^{-8}$
	$N_{pc} = 10$	$2,08(\pm 5,51) \cdot 10^{-11}$	$5,95(\pm 21,3) \cdot 10^{-9}$
ASE-PVESN	$N_{pc} = 3$	$2,31(\pm 0,45) \cdot 10^{-1}$	$2,29(\pm 0,48) \cdot 10^{-1}$
	$N_{pc} = 5$	$1,32(\pm 0,69) \cdot 10^{-2}$	$1,42(\pm 0,72) \cdot 10^{-2}$
	$N_{pc} = 8$	$4,72(\pm 4,31) \cdot 10^{-5}$	$3,53(\pm 10,0) \cdot 10^{-3}$
	$N_{pc} = 10$	$1,32(\pm 1,52) \cdot 10^{-6}$	$2,43(\pm 4,47) \cdot 10^{-3}$

Tabela 5.3: Valores AMSE alcançados por cada ESN para o canal não-linear.

É possível observar na Tabela 5.3 que os valores AMSE obtidos com a R-PVESN são algumas ordens de grandeza menores que aqueles associados às redes R-ESN e ASE-ESN, o que ilustra novamente a potencial vantagem advinda do uso de uma camada de leitura mais flexível.

Quando o reservatório é projetado segundo a estratégia proposta por Ozturk et al. (2007), há uma significativa redução do erro de aproximação do sinal desejado na etapa de treinamento com a arquitetura proposta. Contudo, na etapa de teste, o ganho de desempenho é relativamente pequeno. Este fato, também verificado no cenário anterior, se deve à maior

variação do desempenho da ASE-PESN na etapa de teste.

5.4.3 Equalização Não-Supervisionada

As condições de treinamento das redes neurais com estados de eco são equivalentes às empregadas no contexto de equalização supervisionada (Seção 5.4.2). Porém, uma vez que o sinal de informação $s(n)$ pode ser estimado a partir do erro de predição do sinal recebido, como mostrado na Seção 4.1.6, os valores MSE mostrados a seguir referem-se ao erro quadrático médio entre o erro de predição associado a cada ESN e o erro ideal $e^{\text{ótimo}}(n)$, definido na Equação (4.18).

Vamos considerar os mesmos canais lineares estudados na seção anterior: $H(z) = 0,5 + z^{-1}$ e $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$.

Para o primeiro canal, verificamos, em experimentos preliminares, que o melhor desempenho com a arquitetura proposta, considerando diferentes quantidades de neurônios no reservatório, foi obtido quando $N_{pc} = 4$ componentes principais eram utilizadas. Por outro lado, no caso do segundo canal, o melhor desempenho foi obtido para $N_{pc} = 6$. Restringiremos, portanto, a apresentação dos resultados com a nova arquitetura para estes casos.

Os valores AMSE associados a cada ESN foram determinados com base em $N_{exp} = 20$ experimentos independentes envolvendo cada canal e são exibidos nas Tabelas 5.4 e 5.5, respectivamente.

É possível observar na Tabela 5.4 que a nova arquitetura de ESN alcançou um desempenho melhor para todos os tamanhos de reservatório quando comparada com as redes R-ESN e ASE-ESN. Com respeito à estratégia de projeto do reservatório, embora os resultados obtidos pela R-PVESN e pela ASE-PVESN sejam bastante similares, há uma pequena vantagem em usar uma matriz de pesos aleatória e esparsa (Jaeger, 2001) em vista do fato de não ser preciso utilizar um número elevado de neurônios no reservatório.

Outra observação interessante pode ser feita através da comparação dos valores AMSE exibidos na Tabela 5.4 com aqueles obtidos pelas ESNs no contexto de equalização super-

visionada deste canal, mostrados na Tabela 5.1: notamos que as ESNs conseguem atingir desempenhos melhores quando o processo de treinamento conta com amostras de referência do sinal original, o que evidencia a maior dificuldade da tarefa de equalização no contexto não-supervisionado.

$H(z) = 0,5 + z^{-1}$		AMSE	
Rede	Parâmetro	Treinamento	Teste
R-ESN	$N = 10$	$1,63(\pm 2,12) \cdot 10^{-1}$	$1,67(\pm 2,19) \cdot 10^{-1}$
	$N = 50$	$1,18(\pm 0,25) \cdot 10^{-2}$	$1,23(\pm 0,31) \cdot 10^{-2}$
	$N = 100$	$2,43(\pm 0,27) \cdot 10^{-2}$	$2,86(\pm 0,53) \cdot 10^{-2}$
ASE-ESN	$N = 10$	$1,33(\pm 0,66) \cdot 10^{-1}$	$1,37(\pm 0,66) \cdot 10^{-1}$
	$N = 50$	$2,81(\pm 1,18) \cdot 10^{-2}$	$3,88(\pm 2,40) \cdot 10^{-2}$
	$N = 100$	$3,16(\pm 0,66) \cdot 10^{-2}$	$5,20(\pm 2,12) \cdot 10^{-2}$
R-PVESN ($N_{pc} = 4$)	$N = 10$	$6,01(\pm 2,24) \cdot 10^{-3}$	$6,06(\pm 2,41) \cdot 10^{-3}$
	$N = 50$	$6,98(\pm 1,73) \cdot 10^{-3}$	$7,04(\pm 1,91) \cdot 10^{-3}$
	$N = 100$	$8,43(\pm 2,57) \cdot 10^{-3}$	$8,73(\pm 2,73) \cdot 10^{-3}$
ASE-PVESN ($N_{pc} = 4$)	$N = 10$	$1,36(\pm 0,64) \cdot 10^{-2}$	$1,42(\pm 0,66) \cdot 10^{-2}$
	$N = 50$	$8,75(\pm 2,55) \cdot 10^{-3}$	$9,06(\pm 2,62) \cdot 10^{-3}$
	$N = 100$	$7,41(\pm 2,03) \cdot 10^{-3}$	$7,59(\pm 2,00) \cdot 10^{-3}$

Tabela 5.4: Valores AMSE associados a cada ESN para o canal $H(z) = 0,5 + z^{-1}$.

$H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$		AMSE	
Rede	Parâmetro	Treinamento	Teste
R-ESN	$N = 10$	$9,72(\pm 2,28) \cdot 10^{-2}$	$9,66(\pm 2,39) \cdot 10^{-2}$
	$N = 50$	$2,01(\pm 0,52) \cdot 10^{-2}$	$2,17(\pm 0,55) \cdot 10^{-2}$
	$N = 100$	$2,82(\pm 0,32) \cdot 10^{-2}$	$3,58(\pm 0,85) \cdot 10^{-2}$
ASE-ESN	$N = 10$	$6,69(\pm 2,47) \cdot 10^{-2}$	$7,08(\pm 2,98) \cdot 10^{-2}$
	$N = 50$	$3,18(\pm 1,00) \cdot 10^{-2}$	$3,68(\pm 0,90) \cdot 10^{-2}$
	$N = 100$	$3,27(\pm 0,43) \cdot 10^{-2}$	$5,65(\pm 2,44) \cdot 10^{-2}$
R-PVESN ($N_{pc} = 6$)	$N = 10$	$1,13(\pm 0,39) \cdot 10^{-2}$	$1,21(\pm 0,40) \cdot 10^{-2}$
	$N = 50$	$1,54(\pm 0,19) \cdot 10^{-2}$	$1,64(\pm 0,25) \cdot 10^{-2}$
	$N = 100$	$1,70(\pm 0,27) \cdot 10^{-2}$	$1,84(\pm 0,34) \cdot 10^{-2}$
ASE-PVESN ($N_{pc} = 6$)	$N = 10$	$2,30(\pm 0,40) \cdot 10^{-2}$	$3,07(\pm 1,54) \cdot 10^{-2}$
	$N = 50$	$2,21(\pm 0,32) \cdot 10^{-2}$	$2,59(\pm 0,57) \cdot 10^{-2}$
	$N = 100$	$2,13(\pm 0,40) \cdot 10^{-2}$	$2,45(\pm 0,56) \cdot 10^{-2}$

Tabela 5.5: Valores AMSE associados a cada ESN para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$.

Estas observações também podem ser verificadas na Tabela 5.5, com a ressalva de que a diferença entre os valores AMSE associados a cada ESN se torna relativamente pequena para este canal.

Em certo sentido, portanto, as principais conclusões extraídas no contexto de equalização supervisionada foram mantidas no caso não-supervisionado. Além disso, o desempenho obtido pelas ESNs, especialmente pela nova arquitetura, pode ser considerado promissor.

5.4.4 Separação de Misturas Convolutivas

No problema de separação de misturas convolutivas, para cada mistura utilizamos uma rede neural com estados de eco no papel de um filtro de erro de predição com o propósito de remover superposições no tempo.

Em vez de computarmos o erro entre as fontes originais e suas versões recuperadas por meio de métodos de separação - e.g. técnicas baseadas em ICA -, vamos nos concentrar apenas neste estágio de pré-processamento, uma vez que nosso objetivo é analisar a capacidade de as ESNs transformarem misturas convolutivas em instantâneas. Assim, cada valor MSE corresponde ao erro quadrático médio entre o erro de predição ideal, apresentado na Equação (4.24), e o erro de predição da ESN. No total, temos M valores AMSE, um para cada mistura.

Como já destacado na Seção 4.2.1, vamos considerar a versão estendida da abordagem preditiva, na qual as amostras de todas as misturas são oferecidas aos filtros de erro de predição.

Os cenários considerados neste trabalho envolvem a separação de duas fontes a partir de um sistema misturador com memória $D = 2$. Em todos os casos, $T_s = 5000$ amostras são efetivamente utilizadas no treinamento e teste das ESNs. A seguir, descrevemos brevemente cada cenário, apresentando as especificações referentes aos parâmetros das ESNs em cada caso.

Primeiro Cenário

O primeiro caso é um exemplo de um canal paraunitário com as seguintes matrizes de mistura:

$$\mathbf{A}_0 = \begin{bmatrix} 0,79 & -0,55 \\ 0,21 & -0,15 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} -0,15 & -0,21 \\ 0,55 & 0,79 \end{bmatrix}. \quad (5.15)$$

Um sistema deste tipo, como demonstrado por (Inouye e Liu, 2002), limita o emprego de estruturas lineares junto com um critério baseado apenas em estatísticas de segunda ordem. Sendo assim, uma estrutura não-linear pode ser bastante pertinente pelo fato de implicitamente explorar informações de ordem superior.

Com base em experimentos preliminares envolvendo a arquitetura proposta, adotamos $N = 75$ para o número de neurônios na camada recorrente. Além disso, o número de amostras passadas de cada mistura, as quais correspondem às entradas do respectivo preditor, é igual a um ($N_P = 1$), e, particularmente neste caso, apenas uma ESN implementa a versão estendida do NPEF, o que significa que as amostras das duas misturas são entradas desta ESN em particular, enquanto a outra ESN se concentra em uma única mistura.

Segundo Cenário

O segundo caso envolve o sistema misturador descrito pelas seguintes matrizes:

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0,5 \\ 0,8 & 0,6 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 0,6 & -1,2 \\ 0,3 & 0,9 \end{bmatrix}. \quad (5.16)$$

Substituindo estas matrizes na Equação (4.21), cada sinal observado $r_i(n)$ pode ser visto como sendo fruto de uma combinação de transmissões SISO (*single-input single-output*) através de canais de fase mínima e máxima.

Os experimentos associados à arquitetura proposta foram realizados com $N = 75$ neurô-

nios no reservatório. Com respeito aos filtros de erro de predição, duas amostras passadas são usadas na predição, e as duas ESNs implementam a versão estendida destes filtros, i.e., recebem como entradas os sinais $r_1(n)$ e $r_2(n)$.

Terceiro Cenário

Finalmente, consideramos as seguintes matrizes de mistura:

$$\mathbf{A}_0 = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 3 & 0 \end{bmatrix}. \quad (5.17)$$

A característica peculiar deste sistema misturador é que ele não pode ser invertido com estruturas lineares. Neste cenário, ambas versões da arquitetura proposta (R-PVESN e ASE-PVESN) têm $N = 60$ neurônios dentro do reservatório. Mais uma vez, as ESNs recebem os sinais de ambas misturas como entradas, com a diferença que uma amostra adicional referente à primeira mistura é utilizada na predição (i.e., $N_P = 2$ somente para a primeira mistura).

Discussão

Tendo estes aspectos em mente, apresentamos o conjunto de resultados obtidos com as ESNs no problema de separação de misturas convolutivas. A Tabela 5.6 mostra os valores AMSE associados com cada ESN considerando os três sistemas misturadores descritos anteriormente.

No primeiro cenário, os benefícios trazidos pela arquitetura proposta tornam-se mais evidentes ao observarmos seu desempenho junto à segunda mistura. Como podemos ver na Tabela 5.6, enquanto a R-ESN e a ASE-ESN não conseguem eliminar adequadamente o caráter convolutivo da segunda mistura, a rede proposta, especialmente a versão R-PVESN, leva a um significativo progresso em termos de AMSE. É pertinente mencionar que o melhor resultado para a primeira mistura também é obtido com o auxílio do modelo de ESN pro-

Rede	Parâmetro	Primeiro Cenário		Segundo Cenário		Terceiro Cenário	
		AMSE ₁	AMSE ₂	AMSE ₁	AMSE ₂	AMSE ₁	AMSE ₂
R-ESN	$N = 10$	$5,30(\pm 0,45) \cdot 10^{-2}$	$8,96(\pm 0,32) \cdot 10^{-1}$	$8,25(\pm 1,95) \cdot 10^{-1}$	$3,88(\pm 0,67) \cdot 10^{-1}$	$6,68(\pm 0,82) \cdot 10^{-1}$	$5,74(\pm 2,46) \cdot 10^{-1}$
	$N = 20$	$4,29(\pm 0,54) \cdot 10^{-2}$	$8,10(\pm 0,44) \cdot 10^{-1}$	$6,27(\pm 0,94) \cdot 10^{-1}$	$3,17(\pm 0,36) \cdot 10^{-1}$	$5,04(\pm 0,67) \cdot 10^{-1}$	$3,80(\pm 0,43) \cdot 10^{-1}$
	$N = 40$	$3,12(\pm 0,23) \cdot 10^{-2}$	$6,28(\pm 0,59) \cdot 10^{-1}$	$4,50(\pm 0,30) \cdot 10^{-1}$	$2,41(\pm 0,20) \cdot 10^{-1}$	$3,19(\pm 0,66) \cdot 10^{-1}$	$2,80(\pm 0,20) \cdot 10^{-1}$
	$N = 50$	$3,20(\pm 0,34) \cdot 10^{-2}$	$5,92(\pm 0,39) \cdot 10^{-1}$	$4,29(\pm 0,31) \cdot 10^{-1}$	$2,22(\pm 0,15) \cdot 10^{-1}$	$2,55(\pm 0,42) \cdot 10^{-1}$	$2,55(\pm 0,24) \cdot 10^{-1}$
	$N = 60$	$3,21(\pm 0,27) \cdot 10^{-2}$	$5,76(\pm 0,50) \cdot 10^{-1}$	$3,87(\pm 0,36) \cdot 10^{-1}$	$2,20(\pm 0,19) \cdot 10^{-1}$	$2,35(\pm 0,47) \cdot 10^{-1}$	$2,43(\pm 0,27) \cdot 10^{-1}$
	$N = 100$	$3,98(\pm 0,26) \cdot 10^{-2}$	$5,36(\pm 0,41) \cdot 10^{-1}$	$3,20(\pm 0,30) \cdot 10^{-1}$	$1,80(\pm 0,14) \cdot 10^{-1}$	$2,34(\pm 0,20) \cdot 10^{-1}$	$1,93(\pm 0,25) \cdot 10^{-1}$
ASE-ESN	$N = 10$	$5,40(\pm 0,27) \cdot 10^{-2}$	$9,04(\pm 0,22) \cdot 10^{-1}$	$8,92(\pm 1,68) \cdot 10^{-1}$	$4,99(\pm 0,96) \cdot 10^{-1}$	$6,51(\pm 0,53) \cdot 10^{-1}$	$6,19(\pm 2,73) \cdot 10^{-1}$
	$N = 20$	$5,16(\pm 0,17) \cdot 10^{-2}$	$8,88(\pm 0,28) \cdot 10^{-1}$	$6,75(\pm 1,01) \cdot 10^{-1}$	$3,67(\pm 0,37) \cdot 10^{-1}$	$4,96(\pm 0,54) \cdot 10^{-1}$	$4,19(\pm 1,30) \cdot 10^{-1}$
	$N = 40$	$5,30(\pm 0,20) \cdot 10^{-2}$	$8,79(\pm 0,25) \cdot 10^{-1}$	$5,29(\pm 0,44) \cdot 10^{-1}$	$2,88(\pm 0,30) \cdot 10^{-1}$	$3,83(\pm 0,34) \cdot 10^{-1}$	$2,92(\pm 0,32) \cdot 10^{-1}$
	$N = 50$	$5,46(\pm 0,24) \cdot 10^{-2}$	$8,74(\pm 0,27) \cdot 10^{-1}$	$5,00(\pm 0,39) \cdot 10^{-1}$	$2,66(\pm 0,20) \cdot 10^{-1}$	$3,66(\pm 0,38) \cdot 10^{-1}$	$2,79(\pm 0,25) \cdot 10^{-1}$
	$N = 60$	$5,61(\pm 0,29) \cdot 10^{-2}$	$8,60(\pm 0,35) \cdot 10^{-1}$	$4,36(\pm 0,33) \cdot 10^{-1}$	$2,55(\pm 0,18) \cdot 10^{-1}$	$3,51(\pm 0,38) \cdot 10^{-1}$	$2,65(\pm 0,20) \cdot 10^{-1}$
	$N = 100$	$6,37(\pm 0,31) \cdot 10^{-2}$	$8,54(\pm 0,26) \cdot 10^{-1}$	$3,70(\pm 0,20) \cdot 10^{-1}$	$2,04(\pm 0,14) \cdot 10^{-1}$	$3,32(\pm 0,33) \cdot 10^{-1}$	$2,20(\pm 0,21) \cdot 10^{-1}$
R-PVESN	$N_{pc} = 3$	$3,68(\pm 0,26) \cdot 10^{-2}$	$2,19(\pm 0,91) \cdot 10^{-1}$	$1,00(\pm 0,09) \cdot 10^{-2}$	$5,82(\pm 0,57) \cdot 10^{-1}$	$3,35(\pm 0,43) \cdot 10^{-1}$	$4,68(\pm 2,30) \cdot 10^{-3}$
	$N_{pc} = 5$	$2,75(\pm 0,19) \cdot 10^{-2}$	$4,84(\pm 2,00) \cdot 10^{-2}$	$4,57(\pm 1,07) \cdot 10^{-1}$	$2,23(\pm 0,40) \cdot 10^{-1}$	$1,86(\pm 0,26) \cdot 10^{-1}$	$8,34(\pm 2,19) \cdot 10^{-3}$
	$N_{pc} = 6$	$2,74(\pm 0,29) \cdot 10^{-2}$	$2,75(\pm 0,98) \cdot 10^{-2}$	$2,32(\pm 0,64) \cdot 10^{-1}$	$1,29(\pm 0,35) \cdot 10^{-1}$	$1,57(\pm 0,24) \cdot 10^{-1}$	$1,23(\pm 0,23) \cdot 10^{-2}$
	$N_{pc} = 8$	$3,13(\pm 0,30) \cdot 10^{-2}$	$1,13(\pm 0,31) \cdot 10^{-2}$	$1,10(\pm 0,17) \cdot 10^{-1}$	$6,95(\pm 1,41) \cdot 10^{-2}$	$2,24(\pm 0,20) \cdot 10^{-1}$	$2,52(\pm 0,33) \cdot 10^{-2}$
	$N_{pc} = 10$	$4,68(\pm 0,40) \cdot 10^{-2}$	$6,78(\pm 1,16) \cdot 10^{-3}$	$8,82(\pm 0,91) \cdot 10^{-2}$	$6,51(\pm 0,90) \cdot 10^{-2}$	$4,02(\pm 0,19) \cdot 10^{-1}$	$4,62(\pm 0,35) \cdot 10^{-2}$
ASE-PVESN	$N_{pc} = 3$	$3,98(\pm 0,14) \cdot 10^{-2}$	$3,11(\pm 0,60) \cdot 10^{-1}$	$1,14(\pm 0,04) \cdot 10^{-2}$	$6,28(\pm 0,28) \cdot 10^{-1}$	$2,67(\pm 0,46) \cdot 10^{-1}$	$6,77(\pm 2,12) \cdot 10^{-3}$
	$N_{pc} = 5$	$3,49(\pm 0,31) \cdot 10^{-2}$	$1,10(\pm 0,41) \cdot 10^{-1}$	$5,78(\pm 1,06) \cdot 10^{-1}$	$3,09(\pm 0,49) \cdot 10^{-1}$	$1,72(\pm 0,35) \cdot 10^{-1}$	$8,70(\pm 1,57) \cdot 10^{-3}$
	$N_{pc} = 6$	$3,48(\pm 0,36) \cdot 10^{-2}$	$5,77(\pm 1,14) \cdot 10^{-2}$	$3,97(\pm 0,88) \cdot 10^{-1}$	$2,19(\pm 0,42) \cdot 10^{-1}$	$1,49(\pm 0,27) \cdot 10^{-1}$	$1,25(\pm 0,23) \cdot 10^{-2}$
	$N_{pc} = 8$	$3,99(\pm 0,31) \cdot 10^{-2}$	$2,27(\pm 0,54) \cdot 10^{-2}$	$1,86(\pm 0,20) \cdot 10^{-1}$	$1,17(\pm 0,17) \cdot 10^{-1}$	$2,21(\pm 0,26) \cdot 10^{-1}$	$2,53(\pm 0,30) \cdot 10^{-2}$
	$N_{pc} = 10$	$5,38(\pm 0,39) \cdot 10^{-2}$	$1,28(\pm 0,32) \cdot 10^{-2}$	$1,38(\pm 0,13) \cdot 10^{-1}$	$9,15(\pm 1,23) \cdot 10^{-2}$	$3,99(\pm 0,30) \cdot 10^{-1}$	$4,60(\pm 0,33) \cdot 10^{-2}$

Tabela 5.6: Valores AMSE obtidos com cada ESN no problema de separação de duas misturas convolutivas.

posto, embora a diferença de desempenho neste caso não seja tão pronunciada quanto aquela verificada para a segunda mistura.

Com respeito ao segundo cenário, é possível observar que a flexibilidade adicional de um *readout* não-linear permitiu que a nova arquitetura alcançasse desempenhos superiores para ambas as misturas, especialmente para os casos em que $N_{pc} = 8$ e $N_{pc} = 10$.

As observações apontadas até o momento são válidas também para o terceiro cenário. Como podemos perceber na Tabela 5.6: (1) apenas a arquitetura proposta realiza uma predição suficientemente boa e, portanto, alcança valores menores de AMSE para a segunda mistura quando comparada às redes R-ESN e ASE-ESN; (2) para a primeira mistura, os melhores resultados também são obtidos com o modelo proposto, embora, neste caso, o progresso de desempenho não seja tão expressivo quanto aquele verificado para a segunda mistura.

Outro aspecto importante a ser comentado é que atingir o melhor desempenho para uma determinada mistura não necessariamente implica em um nível de desempenho similar com respeito à outra mistura. Por exemplo, considere o conjunto de resultados obtidos com a R-PVESN no terceiro cenário: quando $N_{pc} = 3$, o menor valor de AMSE é atingido para a segunda mistura; contudo, o erro mínimo para a primeira mistura ocorre quando $N_{pc} = 6$. Portanto, em aplicações práticas, pode ser necessário adotar uma solução de compromisso a fim de alcançar níveis de desempenho satisfatórios para ambas as misturas.

Baseado nestas evidências, é possível afirmar que o uso de uma estrutura mais flexível na camada de saída de uma ESN, como um filtro de Volterra, certamente foi um elemento chave para o bom desempenho alcançado pela arquitetura proposta. Além disso, por meio da técnica de PCA, as melhorias no desempenho foram atingidas sem um significativo aumento da complexidade do processo de treinamento, o que constitui outro atrativo desta proposta (Bocato, Lopes, Attux, e Von Zuben, 2011, 2012).

Passaremos agora à análise de desempenho da nova arquitetura de ESN no âmbito dos demais problemas de predição considerados neste trabalho.

5.4.5 Predição de Séries Caóticas

Como apontado na Seção 4.3, as ESNs serão empregadas na predição de dois sistemas caóticos: o mapa logístico e o sistema de Lorenz. Em ambos os casos, o erro na predição realizada por cada ESN será monitorado à medida que o número de amostras passadas disponíveis na entrada da rede for aumentado. Além disso, também vamos avaliar como a qualidade da predição é afetada quando o horizonte de predição h aumenta. Na etapa de treinamento, a partir da condição inicial, as $T_s = 1100$ amostras subsequentes do estado destes sistema são utilizadas, mas as primeiras 100 amostras não são consideradas no cálculo das medidas de MSE. O conjunto de teste, por sua vez, é composto das $T_s = 1100$ amostras seguintes.

Ao elenco de arquiteturas de redes neurais com estados de eco que temos utilizado até o momento, vamos acrescentar a estrutura híbrida proposta por Butcher et al. (2013), a qual é caracterizada pelo uso de ELMs na camada de saída de uma ESN, como destacado na Seção 5.1, e, por isso, será identificada pela abreviatura ESN/ELM.

Uma vez que os melhores resultados nos problemas de equalização e separação foram obtidos pela arquitetura proposta combinada com o reservatório de Jaeger (2001), e que o mesmo comportamento foi observado em experimentos preliminares de predição caótica, vamos empregar somente esta estratégia de projeto da camada recorrente para as duas arquiteturas com camadas de saída não-lineares, a saber, PVESN e ESN/ELM.

Mapa Logístico

Primeiramente, vamos examinar a influência do número de amostras de entrada na predição do estado no próximo instante, i.e., $l(n+1)$. Em outras palavras, consideramos o caso em que o horizonte de predição é $h = 1$, enquanto n_s varia. A Tabela 5.7 apresenta os valores AMSE obtidos com cada ESN em um conjunto de $N_{\text{exp}} = 20$ experimentos independentes considerando os seguintes parâmetros: $N = 100$ para as redes R-ESN e ASE-ESN, $N = 20$ e $N_{pc} = 10$ para a R-PVESN, e, finalmente, $N = 5$ e $N_h = 100$ para a R-ESN/ELM.

		AMSE	
Rede	Entradas	Treinamento	Teste
R-ESN	$n_s = 1$	$4,88(\pm 2,56).10^{-3}$	$6,63(\pm 3,08).10^{-3}$
	$n_s = 2$	$2,11(\pm 0,90).10^{-3}$	$2,94(\pm 1,12).10^{-3}$
	$n_s = 3$	$3,00(\pm 1,51).10^{-3}$	$4,20(\pm 1,88).10^{-3}$
	$n_s = 4$	$6,97(\pm 2,15).10^{-3}$	$1,07(\pm 0,36).10^{-2}$
	$n_s = 5$	$2,12(\pm 0,55).10^{-2}$	$2,79(\pm 0,57).10^{-2}$
ASE-ESN	$n_s = 1$	$6,39(\pm 2,32).10^{-3}$	$9,97(\pm 3,04).10^{-3}$
	$n_s = 2$	$4,91(\pm 2,56).10^{-3}$	$8,46(\pm 2,88).10^{-3}$
	$n_s = 3$	$6,29(\pm 2,64).10^{-3}$	$1,26(\pm 0,50).10^{-2}$
	$n_s = 4$	$1,20(\pm 0,43).10^{-2}$	$1,82(\pm 0,66).10^{-2}$
	$n_s = 5$	$2,89(\pm 0,93).10^{-2}$	$4,26(\pm 1,14).10^{-2}$
R-PVESN	$n_s = 1$	$1,39(\pm 4,88).10^{-7}$	$4,16(\pm 12,6).10^{-7}$
	$n_s = 2$	$1,40(\pm 4,59).10^{-7}$	$9,73(\pm 22,0).10^{-7}$
	$n_s = 3$	$5,08(\pm 8,41).10^{-7}$	$3,83(\pm 7,56).10^{-6}$
	$n_s = 4$	$1,52(\pm 1,33).10^{-6}$	$9,52(\pm 10,2).10^{-6}$
	$n_s = 5$	$3,05(\pm 3,58).10^{-5}$	$1,08(\pm 0,89).10^{-4}$
R-ESN/ELM	$n_s = 1$	$4,84(\pm 13,4).10^{-10}$	$7,07(\pm 19,8).10^{-10}$
	$n_s = 2$	$1,64(\pm 6,57).10^{-3}$	$1,76(\pm 6,85).10^{-3}$
	$n_s = 3$	$9,54(\pm 31,6).10^{-5}$	$1,35(\pm 0,44).10^{-4}$
	$n_s = 4$	$1,62(\pm 3,97).10^{-3}$	$2,48(\pm 4,96).10^{-3}$
	$n_s = 5$	$6,87(\pm 6,52).10^{-3}$	$1,97(\pm 3,86).10^{-2}$

Tabela 5.7: Valores AMSE obtidos pelas ESNs na predição do estado do mapa logístico considerando $h = 1$ e diferentes valores de n_s .

É possível observar na Tabela 5.7 que o desempenho das ESNs é deteriorado à medida que entradas adicionais estão disponíveis. Este comportamento provavelmente se deve ao fato que, tendo em vista o perfil de correlação quase impulsivo da série temporal em questão, como mostrado na Seção 4.3.1, as entradas adicionais, que não são particularmente úteis mesmo em termos ideais, acabam interferindo no comportamento dinâmico do reservatório, agindo na prática como uma espécie de “ruído” externo.

Também podemos notar na Tabela 5.7 que as arquiteturas de ESN caracterizadas pelo uso de *readouts* não-lineares alcançaram resultados significativamente melhores. De fato, os valores AMSE obtidos com estas arquiteturas, especialmente com a R-ESN/ELM, são algumas ordens de grandeza menores que aqueles obtidos com as redes R-ESN e ASE-ESN. Curiosamente, o desempenho da R-ESN/ELM é o mais afetado quando n_s aumenta. Final-

mente, é importante mencionar que conclusões semelhantes com relação ao efeito do número de entradas podem ser feitas quando outros horizontes de predição são considerados.

Agora, investigaremos o impacto do horizonte de predição no desempenho de cada ESN quando uma única amostra de entrada está disponível ($n_s = 1$). Devido ao perfil impulsivo da função de autocorrelação da variável de estado, o comportamento esperado é que, à medida que h aumenta, o erro de predição tenda a deteriorar-se de maneira significativa. Exibimos na Tabela 5.8 os valores AMSE obtidos com cada ESN em um conjunto de $N_{\text{exp}} = 20$ experimentos independentes. É importante ressaltar que nos casos para $h > 1$, utilizamos uma abordagem de predição direta, i.e., o sinal desejado na etapa de treinamento era $l(n+h)$.

Rede	Horizonte	AMSE	
		Treinamento	Teste
R-ESN	$h = 1$	$4,88(\pm 2,56) \cdot 10^{-3}$	$6,63(\pm 3,08) \cdot 10^{-3}$
	$h = 2$	$1,07(\pm 0,03) \cdot 10^{-1}$	$1,33(\pm 0,04) \cdot 10^{-1}$
	$h = 3$	$1,16(\pm 0,01) \cdot 10^{-1}$	$1,42(\pm 0,03) \cdot 10^{-1}$
ASE-ESN	$h = 1$	$6,39(\pm 2,32) \cdot 10^{-3}$	$9,97(\pm 3,04) \cdot 10^{-3}$
	$h = 2$	$1,14(\pm 0,01) \cdot 10^{-1}$	$1,43(\pm 0,04) \cdot 10^{-1}$
	$h = 3$	$1,16(\pm 0,01) \cdot 10^{-1}$	$1,42(\pm 0,03) \cdot 10^{-1}$
R-PVESN	$h = 1$	$1,39(\pm 4,88) \cdot 10^{-7}$	$4,16(\pm 12,6) \cdot 10^{-7}$
	$h = 2$	$7,71(\pm 24,0) \cdot 10^{-5}$	$2,39(\pm 5,12) \cdot 10^{-4}$
	$h = 3$	$2,94(\pm 3,00) \cdot 10^{-2}$	$3,34(\pm 12,2) \cdot 10^{-1}$
R-ESN/ELM	$h = 1$	$4,84(\pm 13,4) \cdot 10^{-10}$	$7,07(\pm 19,8) \cdot 10^{-10}$
	$h = 2$	$6,74(\pm 20,0) \cdot 10^{-7}$	$9,95(\pm 29,0) \cdot 10^{-7}$
	$h = 3$	$3,18(\pm 6,46) \cdot 10^{-3}$	$4,18(\pm 8,33) \cdot 10^{-3}$

Tabela 5.8: Valores AMSE obtidos pelas ESNs na predição do estado do mapa logístico considerando $n_s = 1$ e diferentes horizontes de predição.

Como esperado, o erro de predição associado a cada ESN aumenta à medida que o horizonte de predição cresce. Porém, podemos observar na Tabela 5.8 que, enquanto as redes R-ESN e ASE-ESN não conseguem prever o estado do sistema para $h > 1$, as arquiteturas com camadas de saída não-lineares alcançam resultados satisfatórios nestes casos, especialmente a R-ESN/ELM, que é a única arquitetura com bom desempenho para $h = 3$.

Para concluir a análise neste cenário, mostramos na Figura 5.3 a sequência correta do estado do mapa logístico, bem como os valores estimados por cada ESN, junto com os histo-

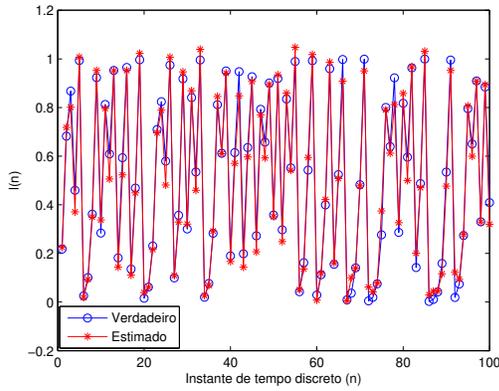
gramas dos resíduos (erros). Visto que os valores AMSE obtidos pelas arquiteturas R-PVESN e R-ESN/ELM foram muito pequenos, não haveria distinção visual entre as respectivas séries preditas. Por isso, mostramos apenas a série associada à arquitetura PV-ESN. É possível perceber com clareza na Figura 5.3 o avanço em termos de precisão na estimação do estado futuro do mapa logístico obtido com uma camada de saída não-linear.

Sistema de Lorenz

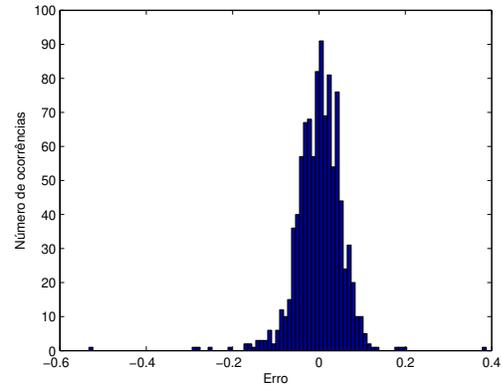
Em todos os experimentos, os seguintes valores para os parâmetros das redes foram adotados: $N = 100$, para a R-ESN e ASE-ESN, $N = 20$ e $N_{pc} = 10$ para a R-PVESN, e, finalmente, $N = 20$ e $N_h = 100$ para a R-ESN/ELM. Primeiramente, mostramos na Tabela 5.9 os valores AMSE associados a cada ESN à medida que o número de entradas aumenta, considerando a predição da amostra seguinte do estado do sistema, ou, equivalentemente, para um horizonte igual a $h = 0,045$ segundos, que corresponde ao período de amostragem.

Rede	Entradas	AMSE	
		Treinamento	Teste
R-ESN	$n_s = 1$	$1,02(\pm 0,60) \cdot 10^{-4}$	$1,75(\pm 1,38) \cdot 10^{-4}$
	$n_s = 3$	$2,30(\pm 1,12) \cdot 10^{-5}$	$3,61(\pm 1,69) \cdot 10^{-5}$
	$n_s = 4$	$1,87(\pm 0,47) \cdot 10^{-5}$	$3,85(\pm 1,20) \cdot 10^{-5}$
	$n_s = 6$	$2,19(\pm 0,95) \cdot 10^{-5}$	$3,81(\pm 1,47) \cdot 10^{-5}$
ASE-ESN	$n_s = 1$	$2,02(\pm 0,32) \cdot 10^{-3}$	$3,98(\pm 1,64) \cdot 10^{-3}$
	$n_s = 3$	$1,60(\pm 0,45) \cdot 10^{-4}$	$2,69(\pm 0,87) \cdot 10^{-4}$
	$n_s = 4$	$4,60(\pm 0,84) \cdot 10^{-5}$	$1,02(\pm 0,19) \cdot 10^{-4}$
	$n_s = 6$	$3,92(\pm 1,22) \cdot 10^{-5}$	$8,09(\pm 2,54) \cdot 10^{-5}$
R-PVESN	$n_s = 1$	$4,52(\pm 14,0) \cdot 10^{-5}$	$3,23(\pm 3,19) \cdot 10^{-4}$
	$n_s = 3$	$1,55(\pm 2,32) \cdot 10^{-6}$	$5,87(\pm 14,6) \cdot 10^{-5}$
	$n_s = 4$	$1,74(\pm 1,59) \cdot 10^{-7}$	$3,63(\pm 3,93) \cdot 10^{-6}$
	$n_s = 6$	$8,36(\pm 7,46) \cdot 10^{-7}$	$9,93(\pm 9,84) \cdot 10^{-6}$
R-ESN/ELM	$n_s = 1$	$7,28(\pm 4,00) \cdot 10^{-4}$	$1,78(\pm 1,13) \cdot 10^{-3}$
	$n_s = 3$	$2,72(\pm 2,13) \cdot 10^{-4}$	$5,83(\pm 4,83) \cdot 10^{-4}$
	$n_s = 4$	$5,43(\pm 2,91) \cdot 10^{-5}$	$1,40(\pm 0,71) \cdot 10^{-4}$
	$n_s = 6$	$2,47(\pm 1,64) \cdot 10^{-4}$	$5,65(\pm 4,28) \cdot 10^{-4}$

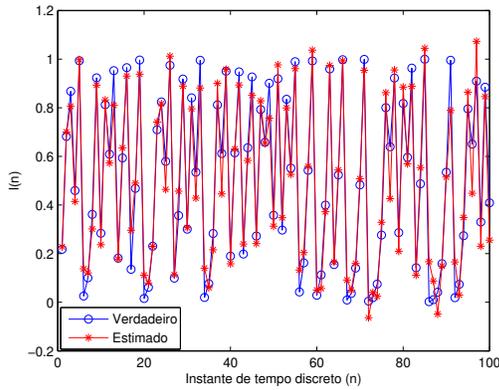
Tabela 5.9: Valores AMSE obtidos com cada ESN na predição do estado do sistema de Lorenz considerando $h = 0,045$ s.



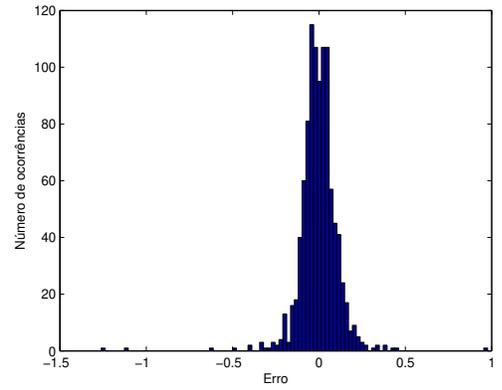
(a) R-ESN



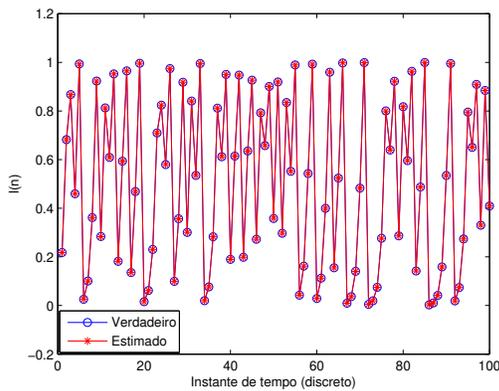
(b) R-ESN



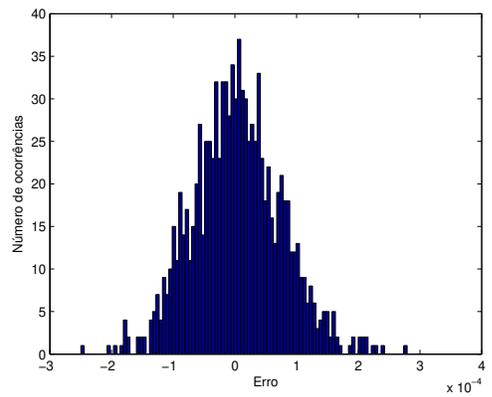
(c) ASE-ESN



(d) ASE-ESN



(e) R-PVESN



(f) R-PVESN

Figura 5.3: Curvas de predição do estado do mapa logístico e histogramas dos resíduos associados a cada ESN considerando $h = 1$ e $n_s = 1$.

Algumas observações interessantes podem ser extraídas da Tabela 5.9: (i) o uso de entradas adicionais pode melhorar o desempenho das redes na predição, o que provavelmente está relacionado ao lento decaimento da função de autocorrelação da série de Lorenz, como mostrado na Seção 4.3.2: de fato, o melhor desempenho para quase todas as arquiteturas foi obtido com $n_s = 4$; (ii) o emprego de uma ELM na camada de saída não ofereceu qualquer progresso quando comparado ao tradicional combinador linear; (iii) por outro lado, os melhores resultados foram obtidos com a R-PVESN, embora a diferença com relação aos valores AMSE atingidos pela R-ESN e pela ASE-ESN tenha sido relativamente pequena.

Estas observações atestam que o uso de uma camada de saída não-linear pode trazer aprimoramentos no desempenho, mas também indicam que nem todas as estruturas não-lineares serão bem-sucedidas dependendo da aplicação envolvida.

		AMSE	
Rede	Horizonte	Treinamento	Teste
R-ESN	1	$1,22(\pm 0,60).10^{-5}$	$2,57(\pm 1,65).10^{-5}$
	3	$9,90(\pm 2,70).10^{-5}$	$2,26(\pm 0,70).10^{-4}$
	5	$3,61(\pm 0,60).10^{-4}$	$1,34(\pm 0,34).10^{-3}$
	7	$3,34(\pm 1,06).10^{-3}$	$1,96(\pm 0,54).10^{-2}$
	9	$3,15(\pm 0,81).10^{-2}$	$1,15(\pm 0,21).10^{-1}$
ASE-ESN	1	$2,68(\pm 0,77).10^{-5}$	$6,69(\pm 2,79).10^{-5}$
	3	$1,70(\pm 0,48).10^{-4}$	$4,52(\pm 1,71).10^{-4}$
	5	$6,17(\pm 1,05).10^{-4}$	$2,77(\pm 1,19).10^{-3}$
	7	$5,03(\pm 1,33).10^{-3}$	$2,82(\pm 0,54).10^{-2}$
	9	$4,08(\pm 0,79).10^{-2}$	$1,47(\pm 0,30).10^{-1}$
R-PVESN	1	$3,76(\pm 3,04).10^{-7}$	$5,39(\pm 6,37).10^{-6}$
	3	$7,07(\pm 7,68).10^{-7}$	$9,65(\pm 13,1).10^{-6}$
	5	$2,46(\pm 1,70).10^{-6}$	$2,88(\pm 2,42).10^{-5}$
	7	$2,33(\pm 1,88).10^{-4}$	$2,72(\pm 1,15).10^{-3}$
	9	$6,10(\pm 3,14).10^{-3}$	$6,53(\pm 1,47).10^{-2}$
R-ESN/ELM	1	$1,18(\pm 0,85).10^{-4}$	$3,03(\pm 2,87).10^{-4}$
	3	$4,32(\pm 4,47).10^{-4}$	$9,41(\pm 10,3).10^{-4}$
	5	$9,27(\pm 7,27).10^{-4}$	$2,17(\pm 1,87).10^{-3}$
	7	$3,07(\pm 1,81).10^{-3}$	$7,81(\pm 4,22).10^{-3}$
	9	$2,33(\pm 0,96).10^{-2}$	$1,04(\pm 0,41).10^{-1}$

Tabela 5.10: Valores AMSE obtidos com cada ESN na predição do estado do sistema de Lorenz considerando $n_s = 4$.

Finalmente, considerando o melhor caso em termos do número de entradas ($n_s = 4$), monitoramos o desempenho de cada ESN conforme o horizonte de predição foi aumentado. Na Tabela 5.10, os valores AMSE obtidos com as ESNs são apresentados, e a coluna associada ao horizonte de predição fornece o número inteiro que multiplica o intervalo de tempo padrão $h = 0,045s$.

Como podemos observar, para todos os valores de horizonte de predição, o melhor desempenho é obtido com a nova arquitetura caracterizada pelo uso da estrutura do filtro de Volterra na camada de leitura. Entretanto, mais uma vez, uma *extreme learning machine* não foi capaz atingir um desempenho melhor quando comparada com um combinador linear na camada de saída da ESN na predição do estado do sistema de Lorenz.

5.4.6 Predição de Séries de Vazões Mensais

Os cenários de predição considerados nesta seção estão associados a três períodos da série de vazão mensal da hidroelétrica de Furnas, localizada no Rio Grande, Brasil. Os conjuntos de teste são formados pelas amostras dos seguintes períodos: (1) 1952 a 1956 - (seco), cuja vazão média é igual a $656,41 \text{ m}^3/s$; (2) 1972 a 1976 - (médio), cuja vazão média é igual a $882,63 \text{ m}^3/s$; (3) 1981 a 1985, (úmido), cuja vazão média é igual a $942,04 \text{ m}^3/s$. Todos os períodos de teste, portanto, compreendem 5 anos, ou 60 amostras de vazões mensais, e são comumente utilizados neste tipo de estudo (Siqueira et al., 2011) devido a suas características diversificadas. Em contrapartida, o conjunto de treinamento é composto de todas as demais amostras da série de vazão no período de 1931 a 1990.

Todas as arquiteturas de rede neural com estados de eco foram treinadas com duas entradas, i.e., usando as amostras associadas ao mês atual e anterior. Esta escolha baseou-se em experimentos preliminares e buscou atingir um compromisso entre desempenho e parcimônia. O horizonte de predição, por sua vez, foi sempre igual a um.

Também a partir de testes preliminares, apenas duas componentes principais foram efetivamente utilizadas na camada de saída da arquitetura proposta. Com respeito ao modelo de

Butcher et al. (2013), o número de neurônios na camada intermediária foi também definido a partir de testes desse tipo.

5.4.7 Resultados Experimentais

As Tabelas 5.11 a 5.13 exibem o desempenho obtido com as arquiteturas de ESN estudadas neste trabalho, medido através do erro quadrático médio, considerando uma média sobre 20 experimentos independentes, com respeito aos conjuntos de treinamento e de teste, tanto no domínio dessazonalizado quanto no domínio real das séries de vazões. O parâmetro N denota o número de neurônios do reservatório, enquanto N_h é o número de neurônios na camada intermediária da ELM empregada na arquitetura híbrida.

Rede	N/N_h	Treinamento		Teste	
		Real	Dessas.	Real	Dessas.
R-ESN	15	$10,9855.10^4$	0,6020	$7,3308.10^4$	0,2985
ASE-ESN	25	$10,8298.10^4$	0,5692	$7,2037.10^4$	0,2823
R-PVESN	30	$11,1337.10^4$	0,5831	$5,4968.10^4$	0,2491
ASE-PVESN	80	$10,8662.10^4$	0,5632	$5,5751.10^4$	0,2471
R-ESN/ELM	10 / 40	$10,1403.10^4$	0,5362	$4,4674.10^4$	0,2214
ASE-ESN/ELM	7 / 50	$10,4620.10^4$	0,5356	$4,8742.10^4$	0,2139

Tabela 5.11: Valores MSE para a série de Furnas - 1952/1956.

Rede	N/N_h	Treinamento		Teste	
		Real	Dessas.	Real	Dessas.
R-ESN	15	$11,1632.10^4$	0,5761	$6,6845.10^4$	0,4068
ASE-ESN	25	$10,3911.10^4$	0,5643	$7,5629.10^4$	0,4330
R-PVESN	30	$11,1091.10^4$	0,5803	$7,0115.10^4$	0,3838
ASE-PVESN	80	$10,7798.10^4$	0,5602	$6,5607.10^4$	0,3603
R-ESN/ELM	5 / 50	$10,2798.10^4$	0,5178	$5,0143.10^4$	0,3758
ASE-ESN/ELM	7 / 60	$10,1119.10^4$	0,5333	$6,0904.10^4$	0,4293

Tabela 5.12: Valores MSE para a série de Furnas - 1972/1976.

O teste estatístico chamado ANOVA Friedman foi utilizado para checar se os métodos de predição oferecem resultados significativamente diferentes (Luna e Ballini, 2011). Os valores p alcançados foram: $6,52336.10^{-9}$, para o período 1952/1956; $2,88366.10^{-7}$ para 1972/1976 e

Rede	N/N_h	Treinamento		Teste	
		Real	Dessas.	Real	Dessas.
R-ESN	110	$6,6726 \cdot 10^4$	0,3082	$25,3856 \cdot 10^4$	1,6505
ASE-ESN	100	$7,2630 \cdot 10^4$	0,3294	$23,1586 \cdot 10^4$	1,6114
R-PVESN	120	$8,5571 \cdot 10^4$	0,3815	$22,8347 \cdot 10^4$	1,6265
ASE-ESN	60	$8,8430 \cdot 10^4$	0,3893	$23,1160 \cdot 10^4$	1,6342
R-ESN/ELM	100 / 60	$7,7406 \cdot 10^4$	0,3662	$22,4578 \cdot 10^4$	1,7028
ASE-ESN/ELM	10 / 40	$8,2486 \cdot 10^4$	0,3801	$24,1913 \cdot 10^4$	1,7842

Tabela 5.13: Valores MSE para a série de Furnas - 1981/1985.

0,0975 para 1981/1985. Isto indica que os desempenhos obtidos na predição das respectivas séries de vazões são realmente distintos ou, em outras palavras, que as estruturas empregadas na predição afetam diretamente os resultados finais.

Os resultados apresentados nas Tabelas 5.11 a 5.13 nos permitem extrair algumas importantes observações. Primeiramente, é possível notar que atingir o melhor desempenho no domínio dessazonalizado não necessariamente significa que o mesmo nível de desempenho é alcançado no domínio original da série. Por exemplo, a R-ESN/ELM obteve o menor valor de MSE de treinamento no domínio real da série de Furnas na Tabela 5.13, porém o valor de MSE no domínio dessazonalizado foi o segundo pior em comparação com as demais redes.

Este problema é inerente à estratégia de predição adotada e ocorre porque o processo de dessazonalização trata igualmente os meses com diferentes valores de desvio padrão. Em segundo lugar, os resultados obtidos não sugerem uma preferência clara entre os métodos de projeto do reservatório de dinâmicas das ESNs.

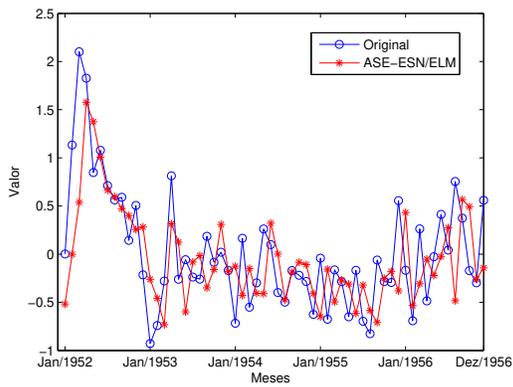
Os modelos de ESN que utilizam uma camada de saída linear (R-ESN e ASE-ESN) apresentaram, em geral, desempenhos superiores frente ao conjunto de treinamento quando comparados com a arquitetura proposta (R-PVESN e ASE-PVESN), e com a arquitetura híbrida de Butcher et al. (2013) no período de teste de 1981/1985. Porém, esta observação não se mantém com respeito aos conjuntos de teste. Foi verificada a possibilidade de sobre-treinamento (*overfitting*), mas uma redução no número de neurônios do reservatório tende a deteriorar o desempenho das ESNs clássicas.

Por outro lado, podemos observar nas Tabelas 5.11 a 5.13 que a introdução de uma camada de saída não-linear - filtro de Volterra junto com PCA ou uma ELM - levou a uma significativa redução do erro de predição nos conjuntos de teste: em todos os períodos, no domínio real, e em dois dos três períodos no domínio dessazonalizado. Isto significa que, embora os melhores resultados no treinamento estejam associados às ESNs clássicas, as ESNs com camadas de leitura não-lineares são capazes de absorver de uma maneira balanceada as características das séries, extraíndo as informações essenciais do conjunto de treinamento sem, contudo, comprometer sua capacidade de generalização.

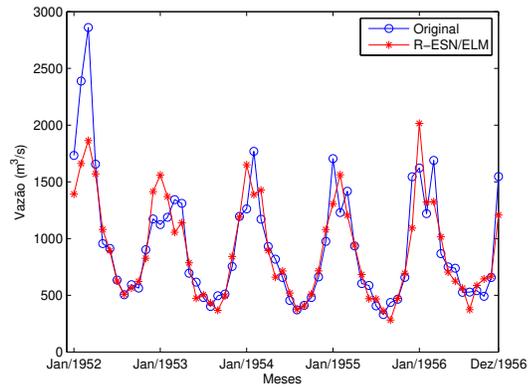
É pertinente também ressaltar que a arquitetura baseada em filtro de Volterra exigiu um número maior de neurônios no reservatório de dinâmicas, o que sugere que é necessário ter mais estados de eco a fim de que, na etapa de compressão, não se perca uma parcela significativa de informação referente ao sinal de entrada. Por fim, verificamos que a arquitetura híbrida de Butcher et al. (2013) alcançou os melhores resultados no geral e, na maioria dos casos, foram utilizados mais neurônios na camada intermediária da ELM do que no reservatório da ESN.

A partir de todas estas observações, podemos afirmar que os resultados obtidos mostram os benefícios alcançados através da introdução de uma camada de saída mais flexível e não-linear para a predição de séries de vazões. Para concluir esta análise, exibimos nas Figuras 5.4 a 5.6 as séries de vazões originais associadas aos três períodos de teste junto com as melhores estimativas, tanto no domínio real quanto no domínio dessazonalizado.

Por fim, é pertinente destacar que a aplicação de máquinas desorganizadas - ESNs e ELMs - ao problema de predição de séries de vazões mensais tem sido investigada de maneira mais detalhada em trabalhos recentes, como (Siqueira, Boccato, Attux, e Lyra Filho, 2012a), (Siqueira, Boccato, Attux, e Lyra Filho, 2012b) e (Siqueira, Boccato, Attux, e Lyra Filho, 2012c).

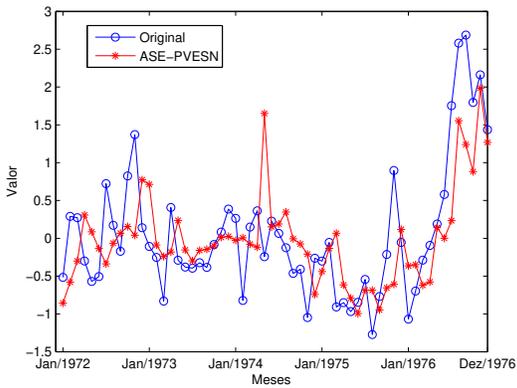


(a) Dessazonalizado

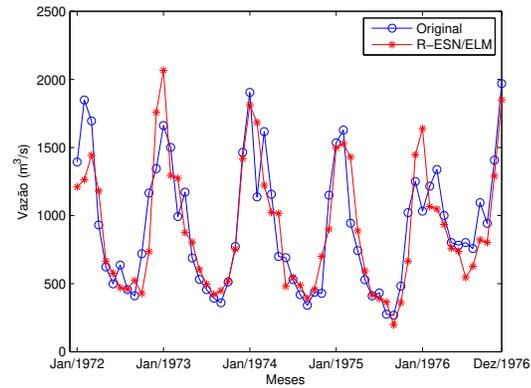


(b) Real

Figura 5.4: Melhores previsões da série Furnas 1952/1956.

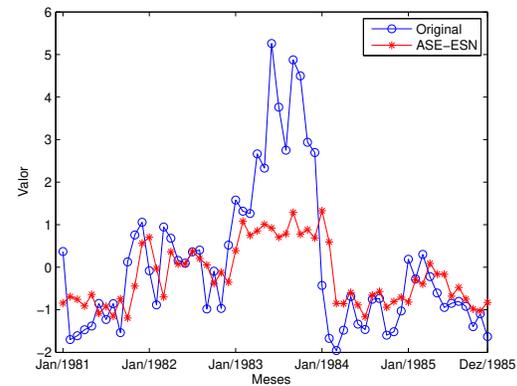


(a) Dessazonalizado

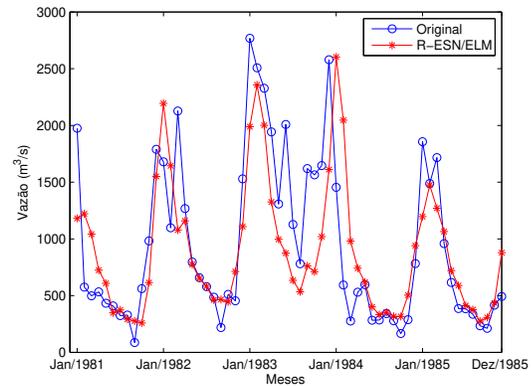


(b) Real

Figura 5.5: Melhores previsões da série Furnas 1972/1976.



(a) Dessazonalizado



(b) Real

Figura 5.6: Melhores previsões da série Furnas 1981/1985.

5.5 Conclusão

Neste capítulo, apresentamos uma nova arquitetura de rede neural com estados de eco caracterizada pelo uso de uma estrutura do tipo filtro de Volterra na camada de saída. O principal atrativo desta ideia é que, com o auxílio de uma estrutura não-linear de filtragem, é possível explorar as estatísticas de ordem superior dos sinais do reservatório e ainda preservar a simplicidade do processo de treinamento, uma vez que a solução ótima para os parâmetros do filtro de Volterra podem ser determinados no sentido de mínimo erro quadrático médio segundo a mesma metodologia de um combinador / filtro linear. Adicionalmente, a técnica de compressão baseada em PCA é aplicada sobre os sinais do reservatório antes de serem transmitidos para a camada de leitura, de modo que apenas um número reduzido de componentes principais é efetivamente usado pelo filtro de Volterra, evitando, assim, um crescimento rápido do número de parâmetros a serem adaptados.

Os benefícios das extensões propostas para o desempenho da ESN foram analisados em vários cenários diferentes dos problemas apresentados no Capítulo 4. Os resultados obtidos neste trabalho não apenas destacam o ganho de desempenho trazido pela camada de saída proposta, especialmente nos cenários mais desafiadores, mas também indicam a viabilidade da abordagem baseada em ESNs nos problemas estudados, caracterizando-as como ferramentas promissoras de processamento de informação.

Até o momento, temos explorado o paradigma de treinamento que visa minimizar o erro quadrático médio entre um sinal de referência e a saída oferecida pela ESN. Entretanto, existe a possibilidade de alcançarmos um aproveitamento mais efetivo do conteúdo estatístico dos sinais existentes em uma ESN por meio do uso de outros critérios para o ajuste dos coeficientes do combinador linear que compõe o *readout*, em vez de modificarmos a própria estrutura da camada de saída. Esta perspectiva será investigada no próximo capítulo.

Critérios de Adaptação da Camada de Saída

As redes neurais com estados de eco abrem um caminho interessante para o uso efetivo de estruturas recorrentes na medida em que simplificam o processo de treinamento de uma RNN ao restringi-lo à adaptação dos parâmetros da camada de saída, enquanto a camada recorrente pode ser pré-definida e permanecer fixa. Porém, as ESNs não são capazes de explorar de maneira completa o conteúdo estatístico referente aos sinais gerados pelo reservatório e também ao sinal desejado. Esta limitação, como destacado no Capítulo 5, é fruto de dois aspectos principais: (1) o caráter linear da estrutura da camada de saída e (2) a adoção do critério de mínimo erro quadrático médio para a adaptação dos parâmetros livres.

O primeiro destes fatores tem motivado a busca por estruturas alternativas para a camada de saída, como, por exemplo, a nova arquitetura de ESN, apresentada no Capítulo 5, cuja camada de saída é formada por um filtro de Volterra (Boccatto et al., 2012). Interessantemente, há uma outra perspectiva capaz de ampliar o aproveitamento estatístico da camada de saída de uma ESN: substituir o critério de adaptação dos parâmetros do *readout*.

A abordagem clássica de treinamento de ESNs, que envolve o uso de um combinador linear adaptado segundo o critério MSE, revela-se ótima quando a distribuição do sinal de erro é gaussiana (Wiener, 1958), pois leva em conta apenas as estatísticas de segunda ordem dos

sinais provenientes do reservatório. Quando esta hipótese não é válida, o emprego de critérios que explorem de maneira mais completa as estatísticas dos sinais presentes no sistema pode ser pertinente. Até o momento, esta possibilidade ainda não foi investigada na literatura.

Neste trabalho, vamos estudar os principais critérios derivados do paradigma de aprendizado baseado na teoria da informação (em inglês, *information-theoretic learning*, ITL) (Erdogmus e Principe, 2006; Principe, 2010) e também algumas opções que surgem com o emprego de normas L_p da medida de erro (Rice e White, 1964).

6.1 Information-Theoretic Learning

O campo de pesquisa conhecido como aprendizado baseado na teoria da informação oferece um conjunto de critérios e algoritmos de adaptação capazes de prover uma extração mais efetiva do conteúdo estatístico disponível ao explorarem grandezas como entropia e informação mútua (Shannon, 1948; Cover e Thomas, 2006; Principe, 2010). Um exemplo emblemático é fornecido pelo critério de entropia do erro (em inglês, *error entropy criterion*, EEC).

Em termos simples, o conceito de entropia, amplamente conhecido a partir do trabalho de Shannon (1948), visa quantificar a incerteza associada a uma variável aleatória, oferecendo uma medida da informação que se pode extrair de um conjunto de observações tendo o conhecimento das probabilidades de ocorrência dos eventos, ou, equivalentemente, conhecendo a função densidade de probabilidade (em inglês, *probability density function*, PDF) que descreve o comportamento desta variável (Cover e Thomas, 2006).

No entanto, apesar de seu sólido fundamento teórico, a definição de Shannon (1948) não é facilmente explorada em um cenário no qual o processo de aprendizado se dá a partir de um conjunto de amostras de um determinado sinal aleatório, o qual caracteriza todos os problemas de extração de informação discutidos no Capítulo 4.

6.1.1 Entropia de Rényi

Neste contexto, a definição que Alfred Rényi dá ao conceito de entropia (Rényi, 1961) é particularmente útil em virtude da possibilidade de obter estimadores não-paramétricos - com o auxílio de métodos de estimação de densidade baseados em funções *kernel*, como o método da janela de Parzen (M. Rosenblatt, 1956; Parzen, 1962) - para aproximar a PDF do sinal de erro a partir de um conjunto de amostras desta variável.

Seja $p_X(x)$ a função densidade de probabilidade de uma variável aleatória contínua X . A entropia de Rényi para a variável X , denotada por $H_\alpha(X)$, é definida como

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int p_X^\alpha(x) dx, \quad (6.1)$$

onde $\alpha \geq 0$. No caso limite em que $\alpha \rightarrow 1$, a entropia de Rényi é equivalente à entropia de Shannon (Principe, 2010).

O argumento do logaritmo em (6.1), chamado de potencial de informação (em inglês, *information potential*, IP) de ordem α , pode ser escrito como

$$V_\alpha(X) = \int p_X^\alpha(x) dx = \int p_X^{\alpha-1}(x) p_X(x) dx = E_X\{p_X^{\alpha-1}(X)\}. \quad (6.2)$$

Em especial, vamos trabalhar com a entropia quadrática de Rényi, i.e., com $\alpha = 2$:

$$H_2(X) = -\log \int p_X^2(x) dx, \quad (6.3)$$

de modo que, neste caso, o potencial de informação corresponde ao valor esperado da PDF de X :

$$V_2(X) = \int p_X^2(x) dx = E_X\{p_X(X)\}. \quad (6.4)$$

6.1.2 Estimador Não-Paramétrico de Entropia

Como destacado anteriormente, a dificuldade fundamental para o uso efetivo do conceito de entropia como critério de adaptação está na necessidade de calcularmos esta grandeza a partir de um conjunto de amostras da variável sob observação. Em outras palavras, é preciso determinar a entropia da variável sem ter conhecimento prévio a respeito de sua PDF.

Neste contexto, uma abordagem de estimação não-paramétrica de PDFs baseada na ideia de funções *kernel*, conhecida como método da janela de Parzen (M. Rosenblatt, 1956; Parzen, 1962), oferece uma solução elegante para este dilema. Suponha que temos T_s amostras independentes e identicamente distribuídas (i.i.d) $\{x_1, \dots, x_{T_s}\}$ de uma variável aleatória X . A estimativa de Parzen da PDF de X usando uma função *kernel* arbitrária $\kappa_{\sigma_k}(\cdot)$ é dada por:

$$\hat{p}_X(x) = \frac{1}{T_s \sigma_k} \sum_{i=1}^{T_s} \kappa\left(\frac{x - x_i}{\sigma_k}\right), \quad (6.5)$$

onde σ_k define a largura do *kernel*. Esta função *kernel* deve obedecer às seguintes propriedades (Silverman, 1986):

1. $\kappa(x) \geq 0$.
2. $\int_{\mathcal{R}} \kappa(x) dx = 1$.
3. $\lim_{x \rightarrow \infty} |x \kappa(x)| = 0$.

Um exemplo de função matemática que satisfaz as propriedades acima, sendo, portanto, uma função *kernel* candidata, é a gaussiana, denotada por $G_{\sigma_k}(\cdot)$:

$$G_{\sigma_k}(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{\|x\|^2}{2\sigma_k^2}}. \quad (6.6)$$

Assim, utilizando *kernels* gaussianos e substituindo a estimativa de PDF, dada na Equação

(6.5), na definição da entropia quadrática de Rényi em (6.3), obtemos:

$$\begin{aligned}
\hat{H}_2(X) &= -\log \int_{-\infty}^{\infty} \left(\frac{1}{T_s \sigma_k} \sum_{i=1}^{T_s} G_{\sigma_k}(x - x_i) \right)^2 dx \\
&= -\log \frac{1}{T_s^2} \int_{-\infty}^{\infty} \left(\sum_{i=1}^{T_s} \sum_{j=1}^{T_s} G_{\sigma_k}(x - x_j) \cdot G_{\sigma_k}(x - x_i) \right) dx \\
&= -\log \frac{1}{T_s^2} \sum_{i=1}^{T_s} \sum_{j=1}^{T_s} \int_{-\infty}^{\infty} G_{\sigma_k}(x - x_j) \cdot G_{\sigma_k}(x - x_i) dx \tag{6.7}
\end{aligned}$$

A Equação (6.7) pode ser simplificada graças a uma propriedade da função gaussiana: o resultado da integral do produto entre duas gaussianas corresponde ao valor da gaussiana calculada na diferença entre os argumentos e cuja variância é a soma das variâncias originais. Com isto, obtemos o estimador da entropia quadrática de Rényi (Principe, 2010):

$$\hat{H}_2(X) = -\log \left(\frac{1}{T_s^2} \sum_{i=1}^{T_s} \sum_{j=1}^{T_s} G_{\sigma_k \sqrt{2}}(x_j - x_i) \right). \tag{6.8}$$

Observe em (6.8) que o potencial de informação pode ser estimado diretamente a partir dos dados segundo a expressão:

$$\hat{V}_2(X) = \frac{1}{T_s^2} \sum_{i=1}^{T_s} \sum_{j=1}^{T_s} G_{\sigma_k \sqrt{2}}(x_j - x_i). \tag{6.9}$$

A Equação (6.9) apresenta um dos resultados essenciais em ITL: o potencial de informação, que é um escalar, pode ser estimado diretamente a partir das amostras com uma avaliação exata de uma integral sobre a variável aleatória para *kernels* gaussianos. Assim, em vez de depender do formato da PDF, o IP é uma função de pares de amostras, à semelhança dos estimadores de média e variância, que também trabalham sobre os dados disponíveis (Erdogmus e Principe, 2002b; Principe, 2010).

Isto significa que é possível evitar o cálculo explícito da entropia de Rényi, o qual demandaria a estimação da PDF e uma operação de integração numérica no domínio da variável

aleatória, através da estimação do IP. Desta forma, por meio de uma operação algébrica (logaritmo), obtém-se a estimativa da entropia quadrática.

Observamos também na Equação (6.9) que a variância da gaussiana é um parâmetro livre que precisa ser ajustado. Por isso, os valores estimados de entropia são influenciados pela escolha da largura da função *kernel*, o que é um problema crucial em estimação de densidades de probabilidade (Silverman, 1986). Além disso, percebemos que é necessário calcular o valor da gaussiana sobre todos os pares de amostras, o que significa que o custo do estimador é proporcional a $O(T_s^2)$ (Principe, 2010).

O estimador mostrado na Equação (6.8) apresenta algumas propriedades interessantes (Erdogmus e Principe, 2002b; Principe, 2010):

- 1) invariância à média da densidade subjacente às amostras. Por causa desta característica, quando utilizamos uma função custo baseada em entropia para aprendizado supervisionado, a média do sinal de erro não é necessariamente nula. Uma forma de contornar este problema consiste em adicionar um termo de *bias* à saída do sistema para forçar a média do erro para zero.
- 2) o valor mínimo de (6.8) ocorre quando todas as amostras da variável aleatória são iguais.
- 3) o mínimo global de (6.8) é suave, i.e., tem gradiente nulo e matriz hessiana semi-definida positiva.

No âmbito de aprendizado supervisionado, é imperativo que a função custo atinja o mínimo global quando todas as amostras de erro são iguais e nulas. Combinando as duas primeiras propriedades, esta condição pode ser atingida pelo estimador de entropia. Além disso, é necessário que o mínimo global seja suave, o que está assegurado pela terceira propriedade. Portanto, à luz destas características, bem como da possibilidade de implicitamente explorar o conteúdo estatístico da própria PDF, podemos dizer que o estimador da entropia de Rényi para o sinal de erro se credencia como um critério alternativo de adaptação.

6.1.3 Critério de Mínima Entropia do Erro

Considere que desejemos ajustar os parâmetros $\mathbf{p} = [p_1 \dots p_Z]$ de um sistema adaptativo linear segundo uma metodologia de aprendizado supervisionado, como ilustrado na Figura 6.1.

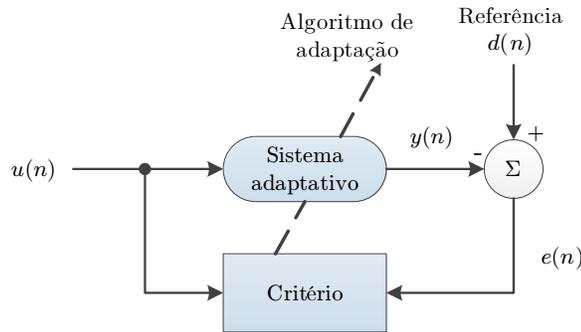


Figura 6.1: Esquema genérico de um problema de aprendizado supervisionado.

Explorando os conceitos de entropia de Rényi e estimação de PDF baseada em funções *kernel*, o critério de mínima entropia do erro (em inglês, *minimum error entropy criterion*, MEEC) é definido como (Erdogmus e Principe, 2002a; Principe, 2010)

$$\mathbf{MEEC}: \min_{\mathbf{p}} \hat{H}(e(n)) \quad (6.10)$$

$$\text{s.a. } e(n) = d(n) - y(n) \text{ e } E\{e(n)\} = 0.$$

O objetivo do MEEC é remover o máximo possível de incerteza do sinal de erro. Portanto, a PDF do erro seria, idealmente, uma função delta de Dirac ($\delta(\cdot)$), o que significaria que a incerteza sobre $e(n)$ foi completamente eliminada ou, equivalentemente, que toda a informação contida nos pares de dados $(u(n), d(n))$ foi assimilada pelo sistema adaptativo em seus parâmetros livres.

Ora, o estimador de entropia mostrado na Equação (6.8) é uma função monotonicamente decrescente com o IP. Logo, minimizar a entropia do sinal de erro equivale a maximizar o

potencial de informação:

$$\min_{\mathbf{p}} \hat{H}(e(n)) = \max_{\mathbf{p}} \hat{V}(e(n)). \quad (6.11)$$

Sendo assim, a solução ótima segundo o MEEC é obtida fazendo

$$\frac{\partial \hat{H}_2(e)}{\partial p_k} \rightarrow \frac{\partial \hat{V}_2(e)}{\partial p_k} = 0, \quad (6.12)$$

onde $k = 1, \dots, Z$.

Infelizmente, ao contrário do que ocorre no caso do critério MSE, não é possível obter uma solução ótima fechada para o problema em (6.10). Contudo, o estimador de IP, apresentado na Equação (6.9), é contínuo e diferenciável com respeito aos parâmetros \mathbf{p} . Estas características são bastante desejáveis, pois permitem a derivação de algoritmos de aprendizado do tipo gradiente (Erdogmus e Principe, 2002b; Principe, 2010).

Em especial, vamos nos concentrar em um cenário de adaptação *online* dos coeficientes \mathbf{p} de um filtro FIR e no cálculo do gradiente estocástico para o potencial de informação. Neste contexto, usaremos a mesma estratégia que Widrow e Stearns (1985) propuseram no famoso algoritmo LMS: aproximar a esperança estatística de uma variável pelo seu valor instantâneo.

Retirando o operador de esperança estatística da expressão do IP - $V_\alpha(X) = E_X\{p_X^{\alpha-1}(X)\}$ - e substituindo a PDF pela estimativa de Parzen calculada sobre as L_j amostras mais recentes no instante n , o potencial de informação estocástico é dado por

$$\hat{V}_\alpha(e(n)) \approx \left(\frac{1}{L_j} \sum_{i=n-L_j}^{i=n-1} \kappa_{\sigma_k}(e(n) - e(i)) \right)^{\alpha-1}, \quad (6.13)$$

onde as amostras de erro $e(n)$ são dadas por

$$\begin{aligned} e(n) &= d(n) - y(n) = d(n) - \mathbf{p}^T \mathbf{u}(n) \\ &= d(n) - \sum_{i=1}^Z p_i u_i(n). \end{aligned} \quad (6.14)$$

Derivando o potencial de informação com respeito a um coeficiente p_k do filtro, obtemos

$$\begin{aligned} \frac{\partial \hat{V}_\alpha(e(n))}{\partial p_k} &= -\frac{\alpha - 1}{L_j^{\alpha-1}} \left(\sum_{i=n-L_j}^{i=n-1} \kappa_{\sigma_k}(e(n) - e(i)) \right)^{\alpha-2} \\ &\times \left[\sum_{i=n-L_j}^{i=n-1} \kappa'_{\sigma_k}(e(n) - e(i))(u_k(n) - u_k(i)) \right], \end{aligned} \quad (6.15)$$

onde $\kappa'_{\sigma_k}(\cdot)$ representa a derivada da função *kernel* com relação ao seu argumento.

Esta é a expressão geral do gradiente estocástico do potencial de informação de ordem α para uma função *kernel* arbitrária, a qual caracteriza o algoritmo denominado *stochastic information gradient for minimum error entropy* (MEE-SIG). No caso particular do IP quadrático ($\alpha = 2$), o primeiro somatório na Equação (6.15) desaparece. Assim, considerando *kernels gaussianos*, o gradiente estocástico é dado por:

$$\frac{\partial \hat{V}_2(e(n))}{\partial p_k} = \frac{1}{\sigma_k^2 L_j} \left[\sum_{i=n-L_j}^{i=n-1} G_{\sigma_k \sqrt{2}}(e(n) - e(i))(e(n) - e(i))(u_k(n) - u_k(i)) \right]. \quad (6.16)$$

Com isto, a cada nova amostra, os coeficientes do filtro são atualizados na direção definida pelo gradiente:

$$p_k(n+1) = p_k(n) + \epsilon \frac{\partial \hat{V}_2(e(n))}{\partial p_k}, \quad (6.17)$$

onde $k = 1, \dots, Z$ e ϵ define o tamanho do passo de adaptação.

É importante destacar que embora tenhamos feito a dedução do MEE-SIG para o caso de um filtro FIR, esta abordagem pode ser diretamente aplicada ao treinamento da camada de saída das máquinas desorganizadas - ESNs e ELMs: a única diferença é que os sinais de entrada $u_k(n)$ da Equação (6.16) correspondem às saídas dos neurônios da camada intermediária destas redes.

6.1.4 Critério de Máxima Correntropia

Outro critério que possui qualidades semelhantes ao MEEC, mas que envolve operações mais simples, remete ao conceito de correntropia (Liu, Pohkarel, e Principe, 2007; Principe, 2010). Por definição, a correntropia cruzada entre duas variáveis aleatórias X e Y é

$$v(X, Y) = E_{XY}\{G_{\sigma_k}(X - Y)\} = \int \int G_{\sigma_k}(x - y)p_{XY}(x, y)dxdy, \quad (6.18)$$

onde $p_{XY}(x, y)$ representa a função densidade de probabilidade conjunta das variáveis X e Y .

Substituindo o operador de esperança estatística pela média amostral, a correntropia cruzada pode ser estimada a partir de um conjunto de amostras $(x_i, y_i), i = 1, \dots, T_s$, de acordo com a seguinte expressão:

$$\hat{v}(X, Y) = \frac{1}{T_s} \sum_{i=1}^{T_s} G_{\sigma_k}(x_i - y_i). \quad (6.19)$$

Com base na Equação (6.19), a correntropia cruzada pode ser interpretada como uma medida de similaridade entre variáveis aleatórias, assim como o MSE. As Figuras 6.2 e 6.3 mostram as superfícies das medidas de MSE e de correntropia em função dos valores das variáveis X e Y .

Há, porém, uma diferença fundamental entre estas medidas: enquanto o MSE pondera de forma igual as amostras próximas e distantes da reta $X = Y$, a correntropia atenua exponencialmente a contribuição dos pontos distantes devido à forma da função *kernel*, o que lhe dá maior imunidade à presença de *outliers* (Principe, 2010). Esta característica motiva a definição de um critério de adaptação alternativo baseado em correntropia.

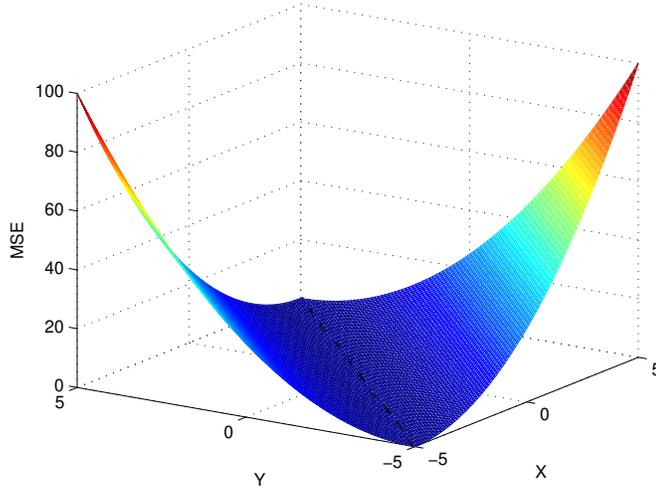


Figura 6.2: Superfície da função custo MSE.

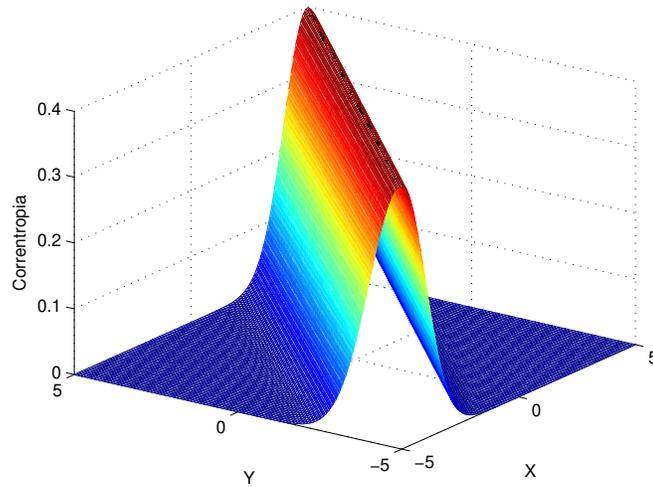


Figura 6.3: Superfície da função de correntropia.

Seja $E = X - Y$ a variável aleatória de erro. Sua PDF pode ser estimada com o auxílio do método da janela de Parzen:

$$\hat{p}_E(e) = \frac{1}{T_s} \sum_{i=1}^{T_s} G_{\sigma_k}(e - e_i). \quad (6.20)$$

Calculando o valor da PDF de E na origem ($e = 0$) e explorando a simetria da função

gaussiana, *viz.*, $G_{\sigma_k}(e) = G_{\sigma_k}(-e)$, obtemos:

$$\hat{p}_E(0) = \frac{1}{T_s} \sum_{i=1}^{T_s} G_{\sigma_k}(e_i). \quad (6.21)$$

Comparando as Equações (6.19) e (6.21), podemos concluir que a correntropia entre X e Y é equivalente ao valor estimado da PDF da variável $E = X - Y$ na origem, *i.e.*, $\hat{v}(X, Y) = \hat{p}_E(0)$ (Principe, 2010). Logo, surge o critério de máxima correntropia do erro (em inglês, *maximum correntropy criterion*, MCC), definido na Equação (6.22), cujo objetivo é maximizar o valor da PDF do sinal de erro na origem, o que significa maximizar o número de amostras com pequenos desvios entre a saída do sistema adaptativo e o sinal desejado no âmbito de aprendizado supervisionado, ou seja,

$$\text{MCC: } \max_{\mathbf{p}} \hat{v}(d(n), y(n)) = \max_{\mathbf{p}} \hat{p}_E(0). \quad (6.22)$$

Embora não seja possível determinar uma solução fechada que maximize a correntropia cruzada, uma abordagem de adaptação *online* baseada no gradiente estocástico pode ser empregada para ajustar os parâmetros de um sistema adaptativo de maneira supervisionada. À semelhança do procedimento adotado na Seção 6.1.3, faremos a dedução do algoritmo MCC-SIG (*stochastic information gradient for maximum correntropy criterion*) considerando um filtro FIR com coeficientes $p_k, k = 1, \dots, Z$.

Portanto, o objetivo é maximizar a correntropia entre a saída do filtro $y(n) = \mathbf{p}^T \mathbf{u}(n)$ e o sinal desejado $d(n)$. Substituindo a PDF do sinal de erro pela estimativa de Parzen tomada sobre as L_j amostras mais recentes, a função objetivo do MCC pode ser escrita como

$$\hat{v}(d(n), y(n)) = \frac{1}{L_j} \sum_{i=n-L_j+1}^n G_{\sigma_k}(e(i)), \quad (6.23)$$

onde $e(i) = d(i) - y(i)$.

Derivando a Equação (6.23) com respeito a um coeficiente p_k do filtro, obtemos a direção

do ajuste a ser realizado:

$$\frac{\partial \hat{v}(d(n), y(n))}{\partial p_k} = \frac{1}{\sigma_k^2 L_j} \sum_{i=n-L_j+1}^n G_{\sigma_k}(e(i)) e(i) u_k(i). \quad (6.24)$$

Comparando as Equações (6.16) e (6.24), é possível perceber que, no caso do MCC, o produto entre amostras do erro e da entrada é ponderado pela função *kernel* calculada sobre a amostra de erro, enquanto no MEE todos os termos envolvem diferenças entre pares de valores de erro e/ou entrada. Por isso, o custo computacional do MCC-SIG é inferior ao MEE-SIG (Principe, 2010).

6.2 Normas L_p

Além dos critérios derivados do paradigma de ITL, outras opções relevantes surgem no contexto de aprendizado supervisionado quando consideramos diferentes normas do sinal de erro (Rice e White, 1964; Gonin e Money, 1989; Debeye e Van Riel, 1990).

A norma L_p de um vetor $\mathbf{e} \in \mathbb{R}^{T_s \times 1}$ é definida como:

$$\|\mathbf{e}\|_p = \sqrt[p]{\sum_{i=1}^{T_s} |e_i|^p}, \quad p = 1, \dots, \infty, \quad (6.25)$$

sendo que

$$\|\mathbf{e}\|_\infty = \lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=1}^{T_s} |e_i|^p} = \max\{|e_i|, i = 1, \dots, T_s\}. \quad (6.26)$$

Sem dúvida, o critério mais conhecido e utilizado em problemas de aprendizado supervisionado, como regressão linear e filtragem adaptativa, é o de minimização da norma L_2 (euclidiana) do erro entre a saída do sistema adaptativo e a resposta desejada:

$$\|\mathbf{e}\|_2 = \sqrt{\sum_{i=1}^{T_s} |e_i|^2}, \quad (6.27)$$

Observe que minimizar a norma L_2 do vetor de amostras de erro leva à mesma solução ótima obtida no sentido dos quadrados mínimos. Além disso, tendo um número suficientemente elevado de amostras, estas soluções convergem para a famosa solução de Wiener, a qual minimiza a esperança do MSE, como destacado na Seção 4.1.5 (Haykin, 1996).

No entanto, como mencionado na Seção 6.1, existem situações em que a solução MSE não é a ideal. Nestes casos, pode ser interessante substituir a norma L_2 por outra, e.g., a norma L_1 . A fim de ilustrar as diferentes soluções que podem ser obtidas com diferentes normas L_p , bem como ressaltar o espírito que motiva cada uma delas, vamos estudar um exemplo simples de ajuste dos coeficientes de um estimador linear.

EXEMPLO 6.1. Identificação de sistema linear

Considere que desejemos realizar a identificação de um sistema desconhecido, cujo comportamento é observado com base nas respostas y_s que ele fornece para o sinal de entrada x . Usaremos um estimador linear de dois coeficientes, que recebe o mesmo sinal de entrada e gera uma saída segundo a regra $y_e = \alpha x + \beta$, para tentar reproduzir a operação do sistema estudado. A Figura 6.4 exibe os elementos que compõem esta tarefa de identificação.

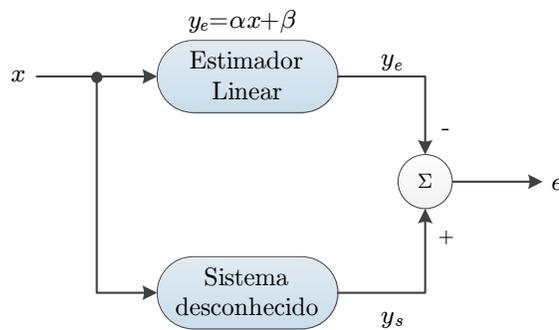


Figura 6.4: Esquema do problema de identificação de um sistema desconhecido usando um estimador linear.

O objetivo é determinar os coeficientes do estimador linear que façam com que sua saída se aproxime ao máximo da resposta fornecida pelo sistema desconhecido. Em termos matemáticos, podemos traduzir este objetivo como um problema de minimização do erro entre a

saída do sistema (y_s) e a resposta do estimador linear (y_e). Em particular, vamos considerar três critérios associados a diferentes normas do sinal de erro, a saber, as normas L_1 , L_2 e L_∞ .

O sinal de entrada x é definido como uma variável aleatória contínua com distribuição uniforme no intervalo $[-5, 5]$. O sistema que desejamos identificar é definido pela seguinte relação: $y_s = 0,25x - 0,5 + \eta$, onde η representa um ruído interno e é modelado por uma variável aleatória gaussiana de média nula e variância σ_η^2 ajustada para uma relação sinal-ruído (em inglês, *signal-to-noise ratio*, SNR) de 8 dB. Os dados disponíveis, portanto, são os pares $\{x(i), y_s(i)\}_{i=1}^{T_s}$, onde T_s indica o número total de amostras. Neste exemplo, consideramos $T_s = 100$.

As equações abaixo mostram as funções custo associadas às normas L_1 , L_2 e L_∞ , respectivamente, onde $\mathbf{e} = [e(1) \dots e(T_s)]^T$. Note que, na realidade, vamos minimizar o quadrado da norma L_2 , o que não afeta a solução ótima, mas facilita as deduções matemáticas.

$$\text{Critério } \mathbf{L}_1 : \min J_{L_1} = \min \|\mathbf{e}\|_1 = \min \sum_{i=1}^{T_s} |e(i)|. \quad (6.28)$$

$$\text{Critério } \mathbf{L}_2 : \min J_{L_2} = \min \|\mathbf{e}\|_2^2 = \min \sum_{i=1}^{T_s} e(i)^2. \quad (6.29)$$

$$\text{Critério } \mathbf{L}_\infty : \min J_{L_\infty} = \min \|\mathbf{e}\|_\infty = \min \max\{|e(1)|, \dots, |e(T_s)|\}. \quad (6.30)$$

A solução ótima associada ao critério de mínima norma L_2 pode ser obtida de forma fechada igualando a derivada da função custo J_{L_2} , mostrada na Equação (6.29), com respeito aos parâmetros α e β a zero e resolvendo o sistema linear obtido. Seguindo este procedimento, a solução ótima L_2 é dada por:

$$\alpha_{L_2} = \frac{\sum_{i=1}^{T_s} x(i)y_s(i) - \frac{\sum_{i=1}^{T_s} y_s(i) \sum_{i=1}^{T_s} x(i)}{T_s}}{\sum_{i=1}^{T_s} x(i)^2 - \frac{\{\sum_{i=1}^{T_s} x(i)\}^2}{T_s}} \quad (6.31)$$

e

$$\beta_{L_2} = \frac{\sum_{i=1}^{T_s} y_s(i) - \alpha \sum_{i=1}^{T_s} x(i)}{T_s}. \quad (6.32)$$

No caso da norma L_1 , não é possível determinar uma solução fechada para os coeficientes do estimador linear. No entanto, ao observarmos as expressões das derivadas de J_{L_1} com relação a estes parâmetros, as quais são mostradas nas Equações (6.33) e (6.34), é possível compreender o que a solução ótima L_1 tenta fazer: estabelecer um equilíbrio entre o número de amostras que contribuem com erros positivos (i.e., que estão acima da reta definida pelo estimador linear) e o número de amostras com erros negativos (i.e., que estão abaixo da reta).

$$\frac{\partial J_{L_1}}{\partial \alpha} = \frac{1}{T_s} \sum_{i=1}^{T_s} \text{sign}(y_s - (\alpha x(i) + \beta)) x_i = 0 \quad (6.33)$$

$$\frac{\partial J_{L_1}}{\partial \beta} = \frac{1}{T_s} \sum_{i=1}^{T_s} \text{sign}(y_s - (\alpha x(i) + \beta)) = 0 \quad (6.34)$$

Por fim, analisando a Equação (6.30), podemos concluir que a solução ótima L_∞ procura minimizar a distância máxima em módulo entre a reta definida pelo estimador linear e o ponto mais distante dela. Mas, à semelhança do que ocorre com a norma L_1 , não é possível encontrar uma solução ótima fechada para (6.30).

Por isso, utilizamos uma rotina de busca em grade para encontrar, aproximadamente, a solução ótima associada às normas L_1 e L_∞ . Esta abordagem só é possível pelo fato de o número de coeficientes ser pequeno neste exemplo.

Sendo assim, foram determinadas as soluções ótimas associadas às normas L_1 , L_2 e L_∞ , as quais são apresentadas na Tabela 6.1. Além disso, exibimos na Figura 6.5 as retas definidas pelos estimadores lineares cujos coeficientes ótimos minimizam, respectivamente, as normas L_1 , L_2 e L_∞ do sinal de erro.

Observando a Figura 6.5 e a Tabela 6.1, fica evidente que o emprego de diferentes normas do sinal de erro como critério de adaptação não somente traz uma motivação diferente para o processo de escolha dos parâmetros ótimos, mas também pode levar a soluções ótimas

distintas.

Critério	Coeficientes ótimos	
	α	β
L_1	0,23	-0,50
L_2	0,23319	-0,5582
L_∞	0,26	-0,57

Tabela 6.1: Coeficientes ótimos do estimador linear obtidos com os critérios de mínima norma L_1 , L_2 e L_∞ .

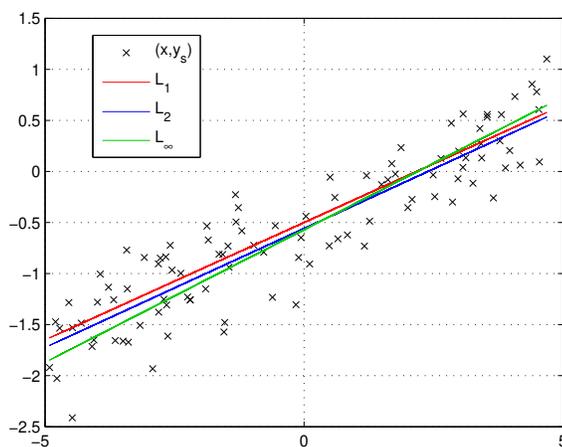


Figura 6.5: Soluções ótimas associadas às normas L_1 , L_2 e L_∞ .

EXEMPLO 6.2. Presença de *outliers*

Considere o mesmo problema de identificação de sistema tratado no exemplo anterior, mas, agora, o sistema em questão sofre com a presença de um ruído espúrio que, na maioria das vezes, introduz uma pequena variação em relação à reta $a_0x + b_0$, mas, em algumas raras ocasiões, pode introduzir uma variação bastante elevada. Para percebermos com maior nitidez os efeitos deste ruído sobre os critérios de normas L_p , utilizamos apenas $T_s = 20$ amostras.

A Figura 6.6 exibe as retas criadas pelos estimadores lineares definidos pelos coeficientes ótimos que minimizam as normas L_1 , L_2 e L_∞ , enquanto a Tabela 6.2 apresenta os valores destes coeficientes.

Critério	Coeficientes ótimos	
	α	β
L_1	0,25	-0,55
L_2	0,1707	-0,3531
L_∞	-0,08	-0,44

Tabela 6.2: Coeficientes ótimos do estimador linear obtidos com os critérios de mínima norma L_1 , L_2 e L_∞ .

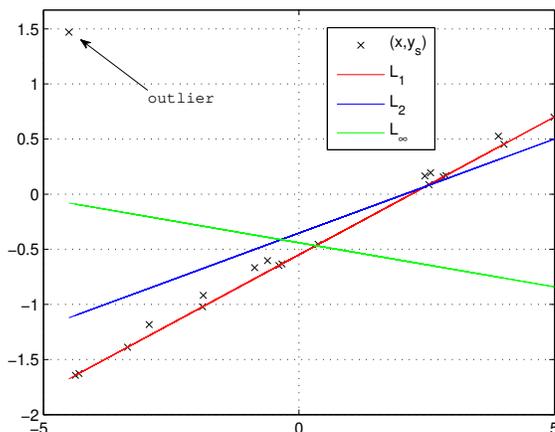


Figura 6.6: Soluções ótimas associadas às normas L_1 , L_2 e L_∞ na presença de *outliers*.

Observando os resultados obtidos, percebemos uma clara diferença entre as normas L_p no tocante ao tratamento de amostras espúrias (*outliers*). A norma L_∞ , uma vez que procura minimizar o desvio máximo, é fortemente atraída para o *outlier*, de modo que sua solução ótima muda drasticamente em relação ao exemplo anterior, e o estimador linear se afasta bastante da reta que originalmente gerou os dados. A norma L_2 , por sua vez, também leva em consideração o *outlier*, ponderando sua contribuição junto com os demais desvios, de modo que a reta associada estimador linear acaba sendo deslocada na direção do *outlier*.

Por outro lado, a norma L_1 , uma vez que se concentra na distribuição dos pontos em torno da reta, e não propriamente na magnitude dos desvios, praticamente despreza o *outlier*: note que a solução ótima para os coeficientes α e β não é muito diferente daquela obtida no exemplo anterior. Portanto, a norma L_1 se mostra mais robusta por ser insensível a grandes

resíduos, i.e., grandes diferenças entre os dados observados e os dados ajustados ou preditos.

Estes dois exemplos no contexto de estimação linear serviram para ressaltar algumas características interessantes de cada norma L_p . Além disso, vimos que, dependendo da natureza do problema e dos sinais envolvidos, o uso de uma determinada norma do sinal de erro pode ser particularmente vantajoso em relação à abordagem convencional, *viz.*, o critério MSE ou, equivalentemente, a norma L_2 . Por exemplo, se o objetivo em questão é desconsiderar um possível ruído elevado, a norma L_1 é a mais adequada. Se, contudo, desejamos projetar um sistema para que ele se comporte bem justamente nas situações extremas (e.g., detecção de anomalias), i.e., em que ocorrem *outliers*, então a norma L_∞ é a melhor opção.

Por isso, com base nestes exemplos, temos a expectativa de que o uso de diferentes normas L_p para a adaptação dos parâmetros da camada de saída de uma ESN possa trazer ganhos de desempenho no sentido de alcançar uma melhor aproximação do sinal desejado dependendo das características dos problemas e sinais envolvidos.

Neste trabalho, vamos considerar os critérios de minimização das normas L_1 e L_4 . Esta última norma foi escolhida para ser uma aproximação, até certa medida, da norma L_∞ .

Apesar de não ser possível determinar uma solução fechada para os critérios de mínima norma L_1 e L_4 , podemos obter a solução ótima de forma iterativa segundo uma metodologia de adaptação *online* baseada no gradiente estocástico. No caso da norma L_2 , o processo de atualização dos coeficientes de um filtro FIR é definido pela famosa expressão do algoritmo LMS (Widrow e Stearns, 1985; Haykin, 1996):

$$p_k(n+1) = p_k + \epsilon(d(n) - y(n))u_k(n). \quad (6.35)$$

As Equações 6.36 e 6.37 apresentam as expressões de atualização dos parâmetros do filtro

FIR segundo os gradientes das normas L_1 e L_4 , respectivamente¹.

$$p_k(n+1) = p_k + \epsilon \operatorname{sign}(d(n) - y(n))u_k(n). \quad (6.36)$$

$$p_k(n+1) = p_k + \epsilon(d(n) - y(n))^3 u_k(n). \quad (6.37)$$

É importante enfatizar que, embora tenhamos tratado da adaptação dos coeficientes de um filtro FIR, as deduções podem ser estendidas de forma natural para o treinamento da camada de saída das redes neurais com estados de eco.

6.3 Resultados Experimentais

A análise comparativa entre os diferentes critérios de adaptação para a camada de saída de uma ESN será feita no âmbito do problema de equalização supervisionada de canais de comunicação, definido no Capítulo 4.

6.3.1 Metodologia

Uma vez que estamos interessados em estabelecer uma comparação adequada entre o critério MSE e os critérios baseados em ITL e normas L_p , em vez de utilizarmos uma métrica de desempenho baseada no erro quadrático médio de aproximação do sinal desejado, como realizado no Capítulo 5, a qual, em certo sentido, tende a favorecer os modelos treinados de acordo com o critério MSE, vamos avaliar o desempenho das ESNs através de duas perspectivas diferentes: (*i*) a da taxa de erro de bit (em inglês, *bit error rate* (BER)), determinada segundo a expressão

$$\text{BER} = \frac{1}{2T_s} \sum_{i=1}^{T_s} |\operatorname{sign}(y(i)) - d(i)|, \quad (6.38)$$

¹As constantes multiplicativas que surgem no cálculo do gradiente foram incorporadas ao passo de adaptação ϵ nas Equações (6.35) a (6.37).

onde $y(i)$ e $d(i)$ representam a saída da rede e o sinal desejado, respectivamente, e (ii) a da função densidade de probabilidade do sinal de erro, a qual oferece uma visão acerca do grau de proximidade entre a PDF do erro referente a cada critério e a função delta, que representa a PDF ideal, como mencionado na Seção 6.1.3.

Em todos os experimentos, para cada critério de adaptação e para cada valor de relação sinal-ruído (em inglês, *signal-to-noise ratio* (SNR)), definida como

$$\text{SNR} = 10 \log \frac{E\{s'(n)^2\}}{\sigma_\eta^2} \text{ (dB)}, \quad (6.39)$$

o valor de BER é obtido transmitindo-se símbolos até que 600 erros sejam detectados ou um valor máximo de 10^6 símbolos seja atingido, sendo a fonte de sinal $s(n)$ composta por amostras i.i.d pertencentes ao alfabeto $\{+1, -1\}$ (modulação 2-PAM). A fim de obter uma medida suficientemente confiável, as curvas de BER versus SNR resultam de uma média baseada em $N_{\text{exp}} = 100$ experimentos independentes.

Como discutido no Capítulo 4, o sinal recebido $r(n)$ resulta do efeito da IIS e da presença de ruído. Com relação à IIS, vamos considerar dois cenários distintos em termos das características do canal. No primeiro cenário, o canal é um sistema de fase máxima descrito pela função de transferência $H(z) = 0,5 + z^{-1}$, o qual, como enfatizado na Seção 5.4.2, exige uma estrutura não-linear de equalização quando se deseja estimar o símbolo da fonte de informação sem qualquer atraso. O segundo cenário, por sua vez, está associado ao canal com estados coincidentes abordado no Exemplo 4.2, cuja função de transferência é dada por $H(z) = 1 + z^{-1}$.

Além disso, vamos considerar dois modelos para $\eta(n)$: ruído branco aditivo gaussiano (em inglês, *additive white Gaussian noise* (AWGN)) e ruído branco aditivo laplaciano (em inglês, *additive white Laplace noise* (AWLN)). Devido ao fato de a distribuição laplaciana ter uma cauda mais longa que a gaussiana, mas um pico mais estreito em torno do valor zero, o modelo AWLN tende a gerar um ruído esparsos, com muitas amostras concentradas na região próxima a zero e algumas amostras com amplitude mais elevada que no caso AWGN.

Assim como adotado no Capítulo 5, as ESNs recebem em sua entrada uma única amostra do sinal recebido $r(n)$ ($K = 1$) e procuram estimar a informação original $s(n)$, de modo que não há atraso de equalização. O número de neurônios no reservatório de dinâmicas das ESNs permanece fixo e igual a $N = 100$, sendo que os parâmetros que definem a camada recorrente - os pesos de entrada e das realimentações - foram ajustados segundo a proposta de Jaeger (2001), descrita na Seção 5.4.1.

Tendo por base testes preliminares envolvendo o primeiro canal e para o ruído AWGN na SNR de 15 dB, adotamos os seguintes valores para os parâmetros dos algoritmos de adaptação associados a cada critério:

Quadro 1 Valores atribuídos aos parâmetros dos algoritmos de adaptação.

MSE: $\epsilon = 0,005$.

MEEC: $L_j = 10$, $\sigma_k = 5$ e $\epsilon = 5$.

MCC: $L_j = 10$, $\sigma_k = 5$ e $\epsilon = 5$.

Norma L_1 : $\epsilon = 0,002$.

Norma L_4 : $\epsilon = 0,0005$.

6.3.2 Primeiro Cenário

AWGN

Tendo como base a metodologia descrita na seção anterior, foram obtidas as curvas de BER em função da SNR para as ESNs treinadas com os critérios MSE, MEEC e MCC, e com os critérios que minimizam as normas L_1 e L_4 , no caso do canal $H(z) = 0,5 + z^{-1}$ e ruído AWGN, as quais são exibidas na Figura 6.7. Além disso, apresentamos também a curva de BER versus SNR obtida pelo equalizador bayesiano, descrito no Apêndice A, considerando duas entradas.

Em primeiro lugar, podemos observar, na Figura 6.7, que os critérios baseados em teoria da informação foram capazes de trazer um ganho de desempenho em termos de BER em relação ao baseado no MSE. Em particular, a ESN treinada com o MEEC obteve o melhor desempenho no geral, o que ressalta a capacidade de exploração da informação estatística dos sinais do reservatório que o MEEC possui.

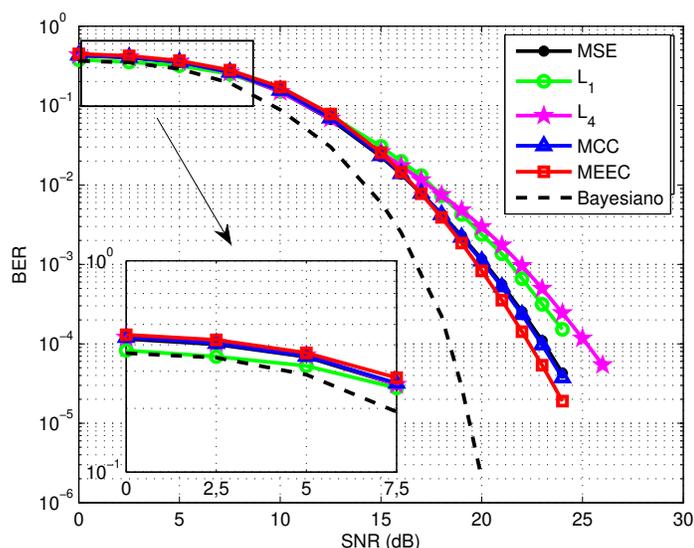


Figura 6.7: Curvas de BER versus SNR referentes a cada critério de adaptação considerando o canal $H(z) = 0,5 + z^{-1}$ e ruído AWGN.

Porém, quando o nível de ruído se torna maior, i.e., à medida que a SNR diminui, os valores de BER associados ao MEEC tornam-se um pouco maiores que aqueles obtidos pela ESN treinada com o critério MSE. Esta pequena queda de desempenho pode estar relacionada ao fato de que os valores dos parâmetros do MEE-SIG - e.g., σ_k - envolvidos na estimação da PDF do erro foram ajustados considerando uma SNR de 15 dB, e, portanto, podem não ser os ideais para as situações em que a potência do ruído torna-se dominante.

Por sua vez, a ESN treinada com o MCC também obteve um desempenho melhor que o da rede ajustada segundo o critério MSE, embora o ganho não seja tão pronunciado quanto aquele obtido com o MEEC. Este comportamento decorre do fato de que o MCC compartilha algumas das características do MEEC, mas não possui a mesma robustez e flexibilidade uma vez que se concentra apenas no valor da PDF do erro na origem. Não obstante, é importante lembrar que o custo computacional do MCC-SIG é inferior ao do MEE-SIG.

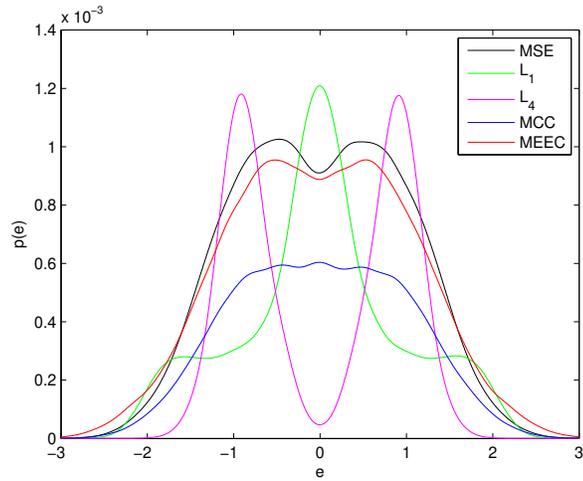
Por outro lado, as opções associadas às normas L_p não foram capazes de melhorar o desempenho da ESN em relação ao critério MSE. De fato, o uso da norma L_4 para a adaptação da camada de leitura da ESN acabou levando a um desempenho pior que o obtido pela ESN

treinada com o critério MSE em toda a faixa de valores de SNR, como podemos notar na Figura 6.7, o que pode estar relacionado ao fato de que a solução de mínima norma L_4 , em certa medida, reflete o espírito da norma L_∞ no sentido de tentar minimizar o erro máximo cometido, e não diretamente todo o conjunto de amostras de erro.

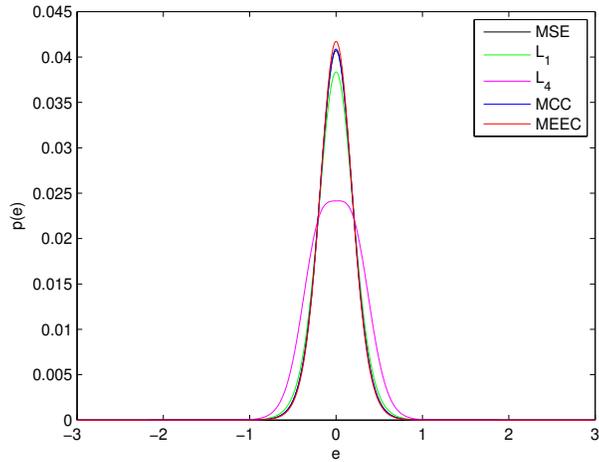
Como discutido no Exemplo 6.1, o objetivo da solução que minimiza a norma L_1 é balancear a distribuição das amostras de erro com sinal positivo e as amostras de erro com sinal negativo. Em outras palavras, ela não leva em consideração a magnitude dos erros, somente seu sinal. Por isso, na situação em que o ruído presente nas amostras é pequeno (SNR alta), o uso da norma L_1 não trouxe vantagem quando comparado ao MSE (norma L_2). Entretanto, quando o ruído torna-se elevado (região de SNR baixa ampliada na Figura 6.7), a ESN adaptada no sentido de mínima norma L_1 atinge o melhor desempenho, o que, provavelmente, se deve à capacidade de a solução ótima L_1 não ser atraída na direção das amostras mais distantes da média, bem como de desprezar eventuais *outliers*, como verificado no Exemplo 6.1.

Este conjunto de observações é complementado pela análise das PDFs do sinal de erro entre a sequência de símbolos desejada e a sequência da saída oferecida pelas ESNs treinadas com cada critério, as quais são apresentadas na Figura 6.8, para as SNRs de 5 dB e 24 dB, respectivamente. Estas PDFs foram estimadas com o auxílio do método da janela de Parzen considerando *kernels* gaussianos e $\sigma_k = 0,1$.

Podemos perceber que para a SNR de 5 dB, a PDF do erro associada ao critério MEEC apresenta uma cauda mais longa do que a PDF associada ao MSE, e as amplitudes da PDF na faixa em torno do valor zero são menores que aquelas associadas ao MSE. Isto explica o desempenho ligeiramente inferior da ESN treinada com o MEEC, verificado na Figura 6.7, nos valores baixos de SNR, como, e.g., 5 dB. Além disso, percebemos que a PDF do erro referente à norma L_1 é a única que preserva um pico destacado centrado na origem, o que contribui para que seu desempenho seja o melhor em termos de BER nesta condição.



(a) SNR = 5 dB



(b) SNR = 24 dB

Figura 6.8: PDFs do sinal de erro associadas a cada ESN/critério de adaptação considerando o canal $H(z) = 0,5 + z^{-1}$ e ruído AWGN.

No caso em que a SNR é 24 dB, a diferença de desempenho entre as ESNs treinadas com cada critério está diretamente ligada à amplitude do pico centrado no valor zero e ao decaimento da PDF do erro à medida que o módulo da amostra de erro aumenta. De fato, quanto mais estreito e elevado for o pico, melhor o desempenho. Por esta razão, a ESN treinada com o MEEC atingiu os menores valores de BER para as SNRs altas, enquanto as redes ajustadas segundo as normas L_1 e L_4 obtiveram desempenhos inferiores quando comparadas àquela adaptada com o critério MSE.

Por fim, é possível constatar na Figura 6.7 que todas as abordagens utilizadas para o treinamento da camada de saída das ESNs levaram a desempenhos inferiores quando comparados com o do equalizador bayesiano com duas entradas, o que, em certo sentido, era esperado, uma vez que o equalizador MAP possui conhecimento completo sobre as características da fonte, do canal e do ruído, e é explicitamente formulado para minimizar a probabilidade de erro de decisão. Além disso, a condição fundamental para a correta recuperação do símbolo transmitido através deste canal é que a estrutura de equalização seja não-linear, o que é plenamente atendido pelo equalizador bayesiano.

AWLN

A Figura 6.9 apresenta as curvas de BER em função da SNR obtidas com as ESNs treinadas de acordo com os critérios MEEC, MSE, MCC e as normas L_1 e L_4 , bem como a curva associada ao equalizador bayesiano com duas entradas, considerando o canal $H(z) = 0,5 + z^{-1}$ e ruído AWLN.

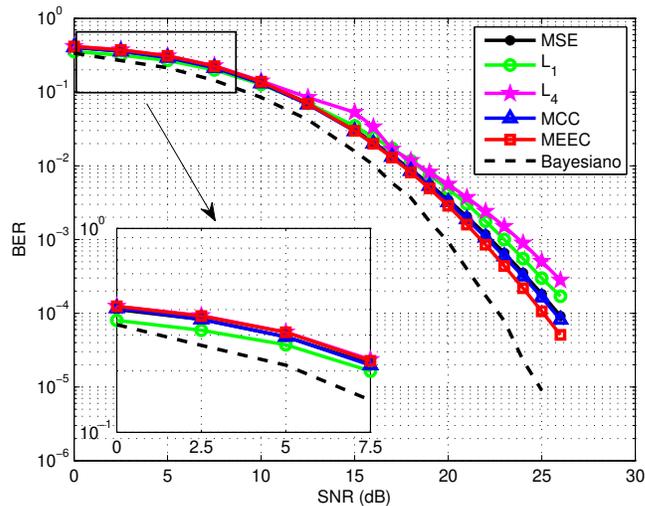
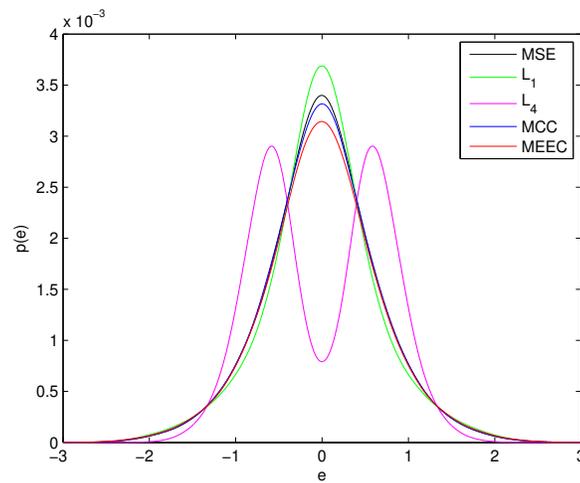


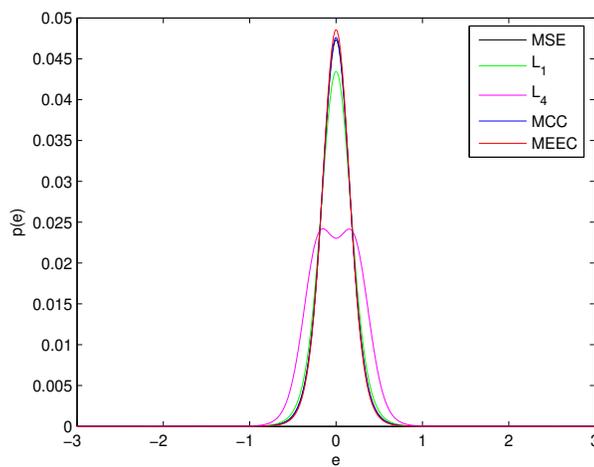
Figura 6.9: Curvas de BER versus SNR referentes a cada critério de adaptação considerando o canal $H(z) = 0,5 + z^{-1}$ e ruído AWLN.

Podemos extrair da Figura 6.9 algumas observações bastante parecidas com aquelas reportadas no caso AWGN: (i) a ESN treinada com o MEEC atinge o melhor desempenho,

superando a rede treinada com o MSE, embora ocorra uma perda de desempenho à medida que a SNR diminui; (ii) o desempenho da rede treinada com o MCC apresenta um comportamento semelhante àquela associada ao MEEC, mas a diferença dos valores de BER em relação aos obtidos com o MSE é pequena; (iii) o uso das normas L_p não traz vantagens em termos de taxa de erro de bit, exceto nos casos em que a potência do ruído torna-se elevada, quando a rede treinada com a norma L_1 do erro alcança o melhor desempenho; (iv) as ESNs não conseguem atingir os valores de BER oferecidos pelo equalizador bayesiano.



(a) SNR = 10 dB



(b) SNR = 26 dB

Figura 6.10: PDFs do sinal de erro associadas a cada ESN/critério de adaptação considerando o canal $H(z) = 0,5 + z^{-1}$ e ruído AWLN.

Com relação às PDFs do sinal de erro, exibidas na Figura 6.10 para as SNRs de 10 dB e 26 dB, as conclusões são semelhantes às aquelas verificadas no cenário anterior: quando a SNR é alta, a PDF do sinal de erro referente ao MEEC é a mais estreita e com o pico mais elevado na origem, enquanto no caso de uma SNR menor, a PDF do erro associado à norma L_1 é a que mais preserva estas características, de modo que o melhor desempenho da ESN é obtido quando os parâmetros de sua camada de saída são adaptados segundo este critério.

6.3.3 Segundo Cenário

Considerando o canal $H(z) = 1 + z^{-1}$ e ruídos AWGN e AWLN, foram obtidas as curvas de BER em função da SNR para as ESNs treinadas com os critérios MSE, MEEC e MCC, e com os critérios que minimizam as normas L_1 e L_4 , as quais são mostradas nas Figuras 6.11 e 6.12, respectivamente. Além disso, mostramos a curva de BER versus SNR associada ao equalizador bayesiano, definido no Apêndice A, com duas entradas.

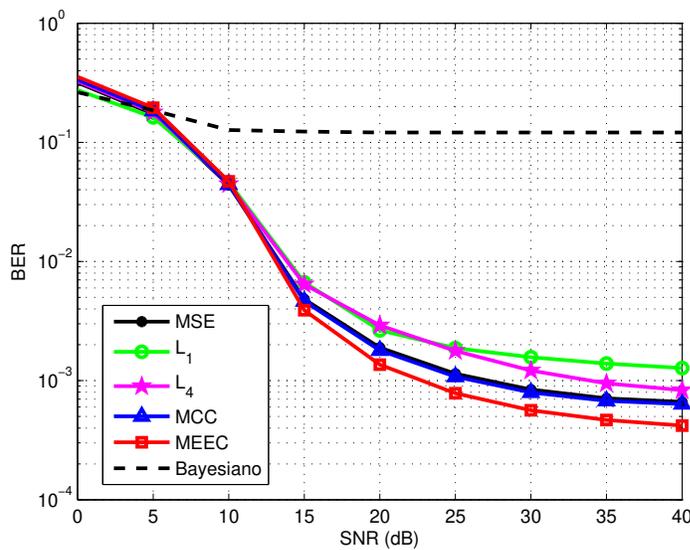


Figura 6.11: Curvas de BER versus SNR referentes a cada critério de adaptação considerando o canal $H(z) = 1 + z^{-1}$ e ruído AWGN.

Os resultados apresentados nas Figuras 6.11 e 6.12 reforçam as principais observações

feitas no contexto do canal anterior com relação aos critérios de adaptação²: o melhor desempenho da ESN é atingido quando os parâmetros do combinador linear da saída são ajustados de acordo com o MEEC. Por outro lado, as ESNs treinadas de acordo com as normas L_1 e L_4 não conseguem alcançar o mesmo desempenho da rede ajustada segundo o critério MSE, exceto quando a potência do ruído se torna mais pronunciada.

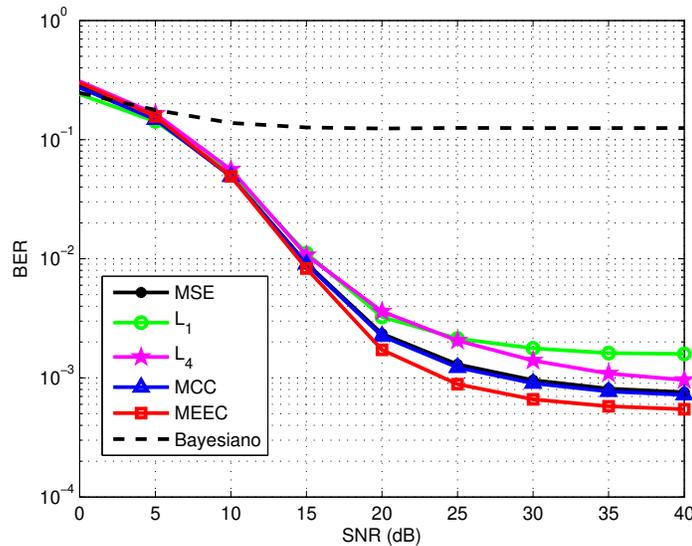


Figura 6.12: Curvas de BER versus SNR referentes a cada critério de adaptação considerando o canal $H(z) = 1 + z^{-1}$ e ruído AWLN.

Interessantemente, percebemos nas Figuras 6.11 e 6.12 que as ESNs, em especial, aquela treinada com o auxílio do algoritmo MEE-SIG, ultrapassam de maneira significativa o desempenho do equalizador bayesiano de duas entradas, o qual não consegue superar uma taxa de erro de 12,5%. Este fato evidencia as vantagens de a estrutura de equalização ser dotada de laços de realimentação e está intimamente ligado à natureza do canal $H(z) = 1 + z^{-1}$.

Como discutido no Exemplo 4.2, este canal possui estados coincidentes, o que significa que diferentes sequências de símbolos da fonte são mapeadas no mesmo estado. Por causa disto, um equalizador do tipo *feedforward*, seja ele linear ou não-linear, visto que cria uma

²Por este motivo, decidimos não apresentar as PDFs estimadas do sinal do erro associadas a cada critério, uma vez que elas apenas confirmam o que já foi observado no primeiro cenário, tanto no caso AWGN, quanto no caso AWLN.

fronteira de decisão estática no espaço dos estados, não é capaz de corretamente distinguir entre aqueles que foram mapeados no mesmo ponto (Montalvão et al., 1999).

No caso do equalizador bayesiano, toda vez que ocorre o estado coincidente, sua função de decisão atinge o valor zero, de maneira que a escolha entre os símbolos $s_{\text{MAP}}(n) = +1$ e $s_{\text{MAP}}(n) = -1$ é feita de maneira arbitrária. Sendo assim, o equalizador comete um erro de decisão em metade das vezes que o estado coincidente aparece em sua entrada e, por isso, a taxa de erro mínima que ele pode atingir está diretamente ligada à probabilidade de ocorrência do estado coincidente.

Seja m a dimensão do vetor de sinal recebido $\mathbf{r}(n)$ e $n_c = 2$ o comprimento do canal $H(z) = 1 + z^{-1}$. O número de estados possíveis de dimensão m é dado por $N_{\text{estados}} = 2^{m+n_c-1} = 2^{m+1}$. Uma vez que, para todo m , apenas duas sequências de símbolos são mapeadas no mesmo estado, a probabilidade de ocorrência do estado coincidente de $H(z) = 1 + z^{-1}$ é dada por:

$$P\{\text{observar o estado coincidente}\} = \frac{2}{N_{\text{estados}}} = \frac{1}{2^m}. \quad (6.40)$$

Portanto, a taxa de erro mínima que o equalizador bayesiano pode alcançar em função da dimensão do estado é

$$\text{BER}_{\text{Mínima}} = \frac{P\{\text{observar o estado coincidente}\}}{2} = \frac{1}{2^{m+1}}. \quad (6.41)$$

Por outro lado, as ESNs, uma vez que criam uma memória dinâmica do sinal de entrada em sua camada recorrente, se mostram capazes de superar o limite de desempenho do equalizador bayesiano e recuperar o sinal transmitido $s(n)$ mesmo quando o estado coincidente é observado.

Esta questão fica mais evidente quando observamos a Figura 6.13, na qual são mostradas as curvas de BER em função da SNR associadas ao equalizador bayesiano considerando diferentes valores de m , bem como a curva de BER versus SNR associada à ESN treinada com o MEEC.

É possível observar na Figura 6.13 que o desempenho do equalizador bayesiano se aproxima daquele obtido pela ESN treinada segundo o MEEC na condição em que o ruído é praticamente desprezível (SNR próxima de 40 dB) quando $m = 10$ amostras do sinal recebido são utilizadas. No entanto, o custo computacional desta abordagem se torna bastante elevado para este caso: uma vez que a função de decisão do equalizador bayesiano, deduzida no Apêndice A, consiste da combinação de funções gaussianas posicionadas sobre os N_{estados} estados do canal, temos $2^{m+n_c-1} = 2048$ elementos não-lineares a serem combinados para a decisão de cada um dos símbolos da fonte.

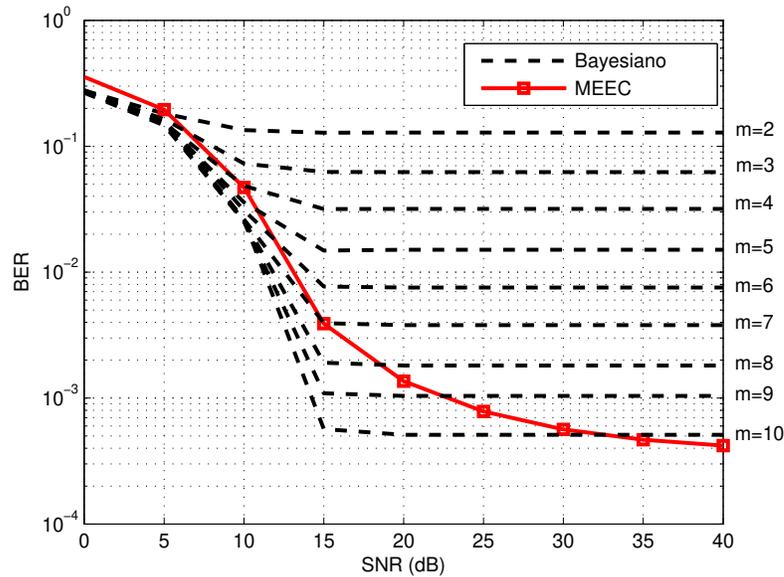


Figura 6.13: Curvas de BER versus SNR referentes ao equalizador bayesiano com várias entradas e à ESN treinada com o MEEC considerando o canal $H(z) = 1 + z^{-1}$ e ruído AWGN.

As ESNs, em contrapartida, fazem a estimação/decisão do símbolo transmitido utilizando um número muito menor de componentes, i.e., de elementos não-lineares, os quais correspondem às ativações dos $N = 100$ neurônios presentes no reservatório, para atingir o mesmo valor de BER. Além disso, elas realizam esta tarefa utilizando uma única amostra do sinal recebido, o que, neste cenário, caracteriza a pior situação em termos da frequência de ocorrência do estado coincidente.

Estas observações, portanto, ressaltam as vantagens adquiridas com o emprego de uma estrutura recorrente no papel de equalizador. É importante destacar que os equalizadores com *feedback* de decisão (em inglês, *decision-feedback equalizers*, DFEs) (Austin, 1967; Belfiore e Park Jr., 1979) também oferecem vantagens similares para o tratamento de canais com memória longa e/ou com estados coincidentes. Contudo, ao realimentarem decisões passadas, podem surgir alguns efeitos indesejáveis devido à propagação de erros de decisão, algo que não acontece nas ESNs. Por isso, temos nas ESNs um exemplo bastante promissor que alia capacidade de processamento dinâmico, que contribui de maneira crucial para o sucesso na tarefa de equalização no âmbito de canais com estados coincidentes, a um processo de treinamento relativamente simples. E, finalmente, dentre as opções analisadas para o ajuste dos parâmetros das ESNs, o critério de mínima entropia do erro leva a um ganho de desempenho em relação à abordagem convencional baseada no MSE graças a sua maior flexibilidade e ao fato de explorar a própria PDF do sinal de erro, em vez de somente as estatísticas de segunda ordem.

6.4 Regularização

Com relação ao processo de treinamento da camada de saída de redes neurais com estados de eco, foram exploradas duas perspectivas: (i) o emprego de uma estrutura de processamento não-linear em lugar do combinador linear, como proposto no Capítulo 5, e (ii) a adoção de diferentes critérios de otimalidade para o processo de adaptação dos parâmetros livres, visando uma extração mais efetiva do conteúdo estatístico dos sinais da rede.

Em ambos os casos, a solução ótima para os coeficientes \mathbf{W}^{out} foi obtida de maneira irrestrita. Ou seja, o processo de treinamento foi formulado segundo um problema de minimização/maximização de uma função custo/objetivo associada a uma medida de erro em relação ao sinal de referência que não impunha restrições sobre os coeficientes de \mathbf{W}^{out} .

A ausência deste tipo de restrição acaba abrindo a possibilidade para a obtenção de

valores bastante elevados para os coeficientes da camada de saída, o que faz com que o modelo de aproximação - no caso, a ESN - se torne, em certa medida, um modelo com alta variância, especialmente quando tem de lidar com dados de entrada que não foram usados no treinamento.

No âmbito de aprendizado supervisionado, ilustrado na Figura 6.1, estas preocupações decorrem do fato de o problema de aproximação de um determinado mapeamento, tendo como base um conjunto de exemplos ou amostras de entrada-saída, ser inevitavelmente mal-formulado ou mal-comportado, visto que a informação contida no conjunto de exemplos não é suficiente para a reconstrução única do mapeamento em regiões onde não existem dados disponíveis (Poggio e Girosi, 1990; Girosi, Jones, e Poggio, 1995).

É justamente no contexto deste dilema de extração de informação / aprendizado que entram em cena as chamadas técnicas de regularização. Em suma, regularização é o procedimento de obtenção de um modelo de aproximação bem-comportado a partir de um mal-comportado através da incorporação de informações adicionais ao processo de ajuste do modelo na forma de penalizações e/ou restrições de suavidade junto ao modelo, bem como de limitantes para a norma do vetor de parâmetros (Tikhonov, 1963). A imposição de regularidade na função a ser aproximada se mostra atraente na medida em que pode reduzir a possibilidade de sobre-treinamento (em inglês, *overfitting*) da rede, melhorando o processo de generalização.

Neste trabalho, vamos analisar como a capacidade de generalização das ESNs pode ser aprimorada quando o critério de adaptação dos parâmetros da camada de saída explora a ideia de regularização. Dentre as várias técnicas existentes na literatura (Hastie, Tibshirani, e Friedman, 2001), escolhemos duas abordagens que trabalham com restrições envolvendo diferentes normas L_p do vetor de parâmetros. A seguir, descrevemos brevemente as técnicas denominadas *ridge regression* (Tikhonov, 1963) e *least absolute selection and shrinkage operator* (LASSO) (Tibshirani, 1996), bem como os procedimentos adotados para a obtenção das respectivas soluções.

6.4.1 Ridge Regression

Seja \mathbf{W}^{out} o vetor de coeficientes da camada de saída de uma ESN³ e \mathbf{d} o vetor contendo as respostas desejadas para um conjunto de T_s amostras de treinamento.

O método conhecido como *ridge regression* (Tikhonov, 1963) define a adaptação dos coeficientes \mathbf{W}^{out} como um problema de minimização da norma L_2 do vetor de erro $\mathbf{e} = \mathbf{d} - \mathbf{W}^{out}\mathbf{X}$ com a adição de um termo de regularização associado à norma L_2 do vetor de parâmetros a ser ajustado. Portanto, o problema de minimização a ser resolvido pode ser enunciado da seguinte forma:

$$\min_{\mathbf{W}^{out}} \|\mathbf{d} - \mathbf{W}^{out}\mathbf{X}\|_2^2 + \lambda \|\mathbf{W}^{out}\|_2^2, \quad (6.42)$$

onde $\mathbf{X} = [\mathbf{x}(1) \dots \mathbf{x}(T_s)]$ contém os vetores de estados da ESN para todas as amostras de entrada.

Pode-se mostrar que resolver (6.42) é equivalente a encontrar a solução para o seguinte problema de otimização com restrições (Tikhonov, 1963):

$$\begin{aligned} \min_{\mathbf{W}^{out}} \|\mathbf{d} - \mathbf{W}^{out}\mathbf{X}\|_2^2 \\ \text{s.a. } \|\mathbf{W}^{out}\|_2 \leq \chi, \end{aligned} \quad (6.43)$$

onde χ restringe a magnitude dos coeficientes da combinação linear e é inversamente proporcional ao parâmetro λ de (6.42).

A solução ótima para (6.42) e (6.43) é dada pela seguinte expressão (Tikhonov, 1963; Lukosevicius e Jaeger, 2009)⁴:

$$\mathbf{W}^{out} = \mathbf{d}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}. \quad (6.44)$$

³Por simplicidade, a apresentação das técnicas de regularização será feita para o caso de uma única saída ($L = 1$), mas os conceitos abordados podem ser facilmente estendidos para uma rede com múltiplas saídas: neste caso, basta tratar cada linha da matriz \mathbf{W}^{out} separadamente.

⁴Esta expressão é válida para o caso em que o número de linhas da matriz \mathbf{X} , que corresponde ao número de neurônios no reservatório (N), é menor que o número de amostras de treinamento (T_s). Caso contrário ($N > T_s$), a solução seria dada por $\mathbf{W}^{out} = \mathbf{d}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$.

Variando o valor de χ (ou λ), alteramos o peso do termo de regularização no processo de adaptação dos coeficientes do combinador linear: quando $\chi = \infty$, retornamos ao problema de otimização irrestrito e à solução definida na Equação (2.8); porém, à medida que χ diminui, os módulos dos coeficientes w_{ij}^{out} são progressivamente reduzidos, até que todos atingem o valor zero na condição de $\chi = 0$.

6.4.2 LASSO

A adição de uma restrição sobre a norma L_2 dos coeficientes é um meio eficiente de atingir estabilidade numérica para a solução ótima dos coeficientes da combinação linear e melhorar o desempenho na aproximação do sinal desejado, especialmente na etapa de teste do modelo, quando este deve lidar com dados que não foram usados durante seu treinamento. Entretanto, esta estratégia não produz uma solução parcimoniosa para o modelo, uma vez que a magnitude dos coeficientes cai lentamente sem, contudo, atingir o valor zero mesmo quando a restrição sobre a norma do vetor de coeficientes é bastante forte. Este fato também dificulta a interpretação dos valores dos coeficientes em termos da importância de cada dado de entrada para a composição da saída.

Por outro lado, a técnica denominada LASSO consegue atender a estas características de maneira elegante ao introduzir uma restrição sobre a norma L_1 dos coeficientes da combinação linear (Tibshirani, 1996). Assim, o problema a ser resolvido pode ser enunciado da seguinte forma:

$$\min_{\mathbf{W}^{out}} \|\mathbf{d} - \mathbf{W}^{out}\mathbf{X}\|_2^2 + \lambda \|\mathbf{W}^{out}\|_1, \quad (6.45)$$

ou, equivalentemente, na forma com restrições:

$$\begin{aligned} \min_{\mathbf{W}^{out}} \|\mathbf{d} - \mathbf{W}^{out}\mathbf{X}\|_2^2 \\ \text{s.a. } \|\mathbf{W}^{out}\|_1 \leq \chi. \end{aligned} \quad (6.46)$$

A adição da restrição sobre a norma L_1 do vetor de parâmetros impossibilita a obtenção

de uma solução fechada para (6.46). Contudo, existem algumas abordagens iterativas que conseguem aproximar a solução ótima para o LASSO.

Algoritmo de Tibshirani

A ideia original de Tibshirani (1996) consiste em resolver uma sequência de problemas de programação quadrática com restrições de desigualdade lineares até obter uma solução que satisfaça a condição sobre a norma L_1 . Uma maneira simples de representar as restrições em (6.46) como um conjunto de restrições lineares é criar uma desigualdade para cada uma das possíveis combinações dos sinais dos elementos de \mathbf{W}^{out} . Por exemplo, no caso de apenas três parâmetros, teríamos o seguinte conjunto de restrições:

$$\begin{aligned}
& +w_1^{out} + w_2^{out} + w_3^{out} \leq \chi \quad , \quad +w_1^{out} + w_2^{out} - w_3^{out} \leq \chi \\
& +w_1^{out} - w_2^{out} + w_3^{out} \leq \chi \quad , \quad +w_1^{out} - w_2^{out} - w_3^{out} \leq \chi \\
& -w_1^{out} + w_2^{out} + w_3^{out} \leq \chi \quad , \quad -w_1^{out} + w_2^{out} - w_3^{out} \leq \chi \\
& -w_1^{out} - w_2^{out} + w_3^{out} \leq \chi \quad , \quad -w_1^{out} - w_2^{out} - w_3^{out} \leq \chi
\end{aligned}
\tag{6.47}$$

Seja $g(\mathbf{W}^{out}) = \|\mathbf{d} - \mathbf{W}^{out}\mathbf{X}\|_2^2$ a função de erro quadrático e $\boldsymbol{\delta}_i, i = 1, \dots, 2^N$ as N -tuplas na forma $(\pm 1, \pm 1, \dots, \pm 1)$. Portanto, a condição $\sum |W_j^{out}| \leq \chi$ é equivalente a $\boldsymbol{\delta}_i^T \mathbf{W}^{out} \leq \chi$ para todo i .

Para um vetor arbitrário de parâmetros \mathbf{W}^{out} , o conjunto $\mathcal{E} = \{i : \boldsymbol{\delta}_i^T \mathbf{W}^{out} = \chi\}$ indica as restrições que são satisfeitas na condição de igualdade, enquanto $\mathcal{S} = \{i : \boldsymbol{\delta}_i^T \mathbf{W}^{out} < \chi\}$ aponta as restrições para as quais a igualdade ainda não foi atingida. Denotamos por $\mathbf{G}_{\mathcal{E}}$ a matriz cujas linhas são as tuplas $\boldsymbol{\delta}_i$ para $i \in \mathcal{E}$.

O algoritmo de Tibshirani (1996) começa com $\mathcal{E} = i_0$, onde $\boldsymbol{\delta}_{i_0} = \text{sign}(\mathbf{W}_0^{out})$, sendo \mathbf{W}_0^{out} a solução irrestrita para o problema de minimização do erro quadrático, a qual é obtida pela Equação (2.8). Em seguida, obtém-se a solução que minimiza $g(\mathbf{W}^{out})$ sujeita a $\boldsymbol{\delta}_{i_0}^T \mathbf{W}^{out} \leq \chi$,

e a condição de norma L_1 é verificada. Se for satisfeita, o algoritmo termina; caso contrário, a restrição linear que foi violada é adicionada ao conjunto \mathcal{E} e o processo continua até que $\sum |W_j^{out}| \leq \chi$. O Quadro 2 esboça o procedimento proposto por Tibshirani (1996) para aproximar a solução de (6.46).

Quadro 2 Algoritmo de Tibshirani para obter a solução LASSO.

Comece com $\mathcal{E} = i_0$, onde $\delta_{i_0} = \text{sign}(\mathbf{W}_0^{out})$.

Faça

Determine a solução $\hat{\mathbf{W}}^{out}$ que minimiza $g(\mathbf{W}^{out})$ sujeita a $\mathbf{G}_{\mathcal{E}} \mathbf{W}^{out} \leq \chi \mathbf{1}$.

Adicione i ao conjunto \mathcal{E} , onde $\delta_i = \text{sign}(\hat{\mathbf{W}}^{out})$.

Enquanto $\sum_{j=1}^N |\hat{\mathbf{W}}^{out}| > \chi$

Algoritmo IRLS

Outra possibilidade interessante para encontrarmos a solução ótima do problema LASSO, apresentado na Equação (6.45), está associada ao algoritmo denominado *iteratively reweighted least squares* (IRLS) (Lawson, 1961; Rice e Usow, 1968; Green, 1984).

Para compreendermos o funcionamento deste método, primeiro vamos derivar a solução ótima para o seguinte problema:

$$\min_{\mathbf{W}^{out}} J_{\text{IRLS}} = \min_{\mathbf{W}^{out}} \|\mathbf{e}\|_2^2 + \lambda \|\mathbf{W}^{out} \mathbf{Q}\|_2^2, \quad (6.48)$$

onde $\mathbf{e} = \mathbf{y} - \mathbf{d}$, sendo $\mathbf{y} \in \mathbb{R}^{1 \times T_s}$ o vetor com todas as saídas da ESN, i.e., $\mathbf{y} = \mathbf{W}^{out} \mathbf{X}$, $\mathbf{X} \in \mathbb{R}^{N \times T_s}$, $\mathbf{W}^{out} \in \mathbb{R}^{1 \times N}$, $\mathbf{d} \in \mathbb{R}^{1 \times T_s}$ e $\mathbf{Q} \in \mathbb{R}^{N \times N}$ é uma matriz diagonal que pondera os coeficientes de \mathbf{W}^{out} .

Observe que (6.48) é um problema de minimização do erro quadrático entre \mathbf{d} e \mathbf{y} com a adição de um termo de penalização proporcional à norma L_2 do vetor de coeficientes \mathbf{W}^{out} ponderados pelos elementos de \mathbf{Q} .

Derivando a função J_{IRLS} com respeito ao vetor de parâmetros \mathbf{W}^{out} , obtemos:

$$\begin{aligned}\frac{\partial J_{\text{IRLS}}}{\partial \mathbf{W}^{out}} &= \frac{\partial(\mathbf{W}^{out}\mathbf{X} - \mathbf{d})(\mathbf{W}^{out}\mathbf{X} - \mathbf{d})^T}{\partial \mathbf{W}^{out}} + \frac{\partial\lambda(\mathbf{W}^{out}\mathbf{Q})(\mathbf{W}^{out}\mathbf{Q})^T}{\partial \mathbf{W}^{out}} \\ &= 2\mathbf{W}^{out}\mathbf{X}\mathbf{X}^T - 2\mathbf{d}\mathbf{X}^T + 2\lambda\mathbf{W}^{out}\mathbf{Q}\mathbf{Q}^T\end{aligned}\quad (6.49)$$

Igualando (6.49) a zero, obtemos a solução ótima:

$$\mathbf{W}^{out} = \mathbf{d}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{Q}\mathbf{Q}^T)^{-1}.\quad (6.50)$$

Interessantemente, é possível expressar o problema do LASSO, definido na Equação (6.45), na forma do problema em (6.48) ao escrevermos a norma L_1 como uma função da norma L_2 . Com efeito, qualquer norma L_p do vetor \mathbf{W}^{out} pode ser escrita como uma norma L_2 ponderada:

$$\|\mathbf{W}^{out}\|_p^p = \|\mathbf{W}^{out}\mathbf{Q}\|_2^2,\quad (6.51)$$

sendo que os elementos da diagonal de \mathbf{Q} são definidos como

$$q_i = |W_i^{out}|^{\frac{p-2}{2}}, \quad i = 1, \dots, N.\quad (6.52)$$

Com o auxílio desta relação, podemos aproximar a norma L_1 por meio de uma norma L_2 ponderada, computando os coeficientes q_i através da Equação (6.52) para $p = 1$ e, deste modo, iterativamente computar a solução do problema LASSO por meio da Equação (6.50). Esta é a estratégia básica do algoritmo IRLS, cujos passos principais estão resumidos no Quadro 3.

Quadro 3 Algoritmo IRLS para aproximar a solução LASSO.

Comece com $\mathbf{W}^{out} = \mathbf{d}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$ ($\lambda = 0$).

Faça

Monte a matriz \mathbf{Q} cujos elementos da diagonal são $q_i = |W_i^{out}|^{-1/2}$, $i = 1, \dots, N$.

Atualize os coeficientes de \mathbf{W}^{out} com a Equação (6.50).

Enquanto Convergência não for atingida

Entretanto, o uso efetivo deste método traz uma preocupação: à medida que W_i^{out} tende a zero, o cálculo de q_i torna-se numericamente instável. Uma maneira simples de amenizar este problema consiste em adicionar uma pequena constante positiva a cada coeficiente W_i^{out} no cálculo dos elementos $q_i, i = 1, \dots, N$.

6.4.3 Análise

As potenciais vantagens de se explorar a ideia de regularização durante o processo de treinamento de ESNs serão analisadas no contexto do problema de predição da série de vazões mensais associada ao posto de Furnas. O conjunto de treinamento é formado por todas as amostras mensais do período 1931-1990, exceto por aquelas referentes ao período 1972-1976, as quais compõem o conjunto de teste, com o qual efetivamente avaliaremos a capacidade de generalização das redes.

Para garantir uma comparação justa entre as diferentes técnicas de regularização, são realizados $N_{\text{exp}} = 50$ experimentos independentes, sendo que o mesmo conjunto de reservatórios é empregado, de maneira que as diferentes soluções para os coeficientes do combinador linear da saída são computadas com base nos mesmos sinais de entrada.

Em cada experimento, a camada recorrente da ESN recebe $K = 3$ entradas - a amostra atual e as duas anteriores - e é formada por $N = 10$ neurônios, sendo que os pesos de entrada podem assumir os valores $+1$ e -1 com igual probabilidade, e os pesos das conexões recorrentes são ajustados segundo a proposta de Jaeger (2001), apresentada na Seção 5.4.1.

As técnicas de regularização consideradas neste trabalho possuem um parâmetro que define o grau de importância/preferência dada a soluções com coeficientes de magnitude baixa. Para cada valor atribuído ao fator de regularização λ ou, equivalentemente, à magnitude da restrição χ imposta à norma do vetor de coeficientes, obtém-se uma solução diferente para o vetor de coeficientes. Por isso, fizemos uma varredura no domínio de λ (ou χ), calculando a solução para \mathbf{W}^{out} de acordo com as abordagens *ridge regression* e LASSO, e testamos o desempenho das respectivas redes no conjunto de teste. Ao final deste processo, adotamos

como solução ótima de cada experimento o vetor de parâmetros associado ao valor de λ (ou χ) que levou ao menor erro de aproximação na etapa de teste.

A Tabela 6.3 apresenta a média dos valores de MSE obtidos pelas ESNs, treinadas segundo as diferentes técnicas de regularização, na predição de um passo à frente da vazão mensal da série de Furnas no período de 1972 a 1976. Além disso, para termos uma noção mais precisa a respeito da variação do desempenho destas soluções, mostramos também o valor máximo e o valor mínimo do MSE, bem como o desvio padrão, associados ao conjunto de experimentos realizados. Por fim, a Tabela 6.3 exibe a média do número de coeficientes aproximadamente nulos em cada solução, denotado por N_{zeros} .

O teste estatístico denominado ANOVA Friedman foi empregado tendo como base o conjunto de valores de erro quadrático médio associado à cada estratégia de ajuste dos parâmetros da camada de saída da ESN para os N_{exp} experimentos independentes. O valor p obtido indicou a ocorrência de mudanças significativas, do ponto de vista estatístico, no desempenho da rede devido à escolha do método de ajuste da camada de saída.

Solução	AMSE	min MSE	max MSE	Desvio padrão	N_{zeros}
MSE irrestrito	0,41411	0,36712	0,50511	0,02844	6,320
<i>Ridge</i>	0,40271	0,36712	0,49615	0,02339	5,280
LASSO	0,39670	0,36712	0,48778	0,02267	7,520
LASSO/IRLS	0,39730	0,36712	0,48233	0,02146	6,680

Tabela 6.3: Desempenho médio das ESNs treinadas com o auxílio de técnicas de regularização na predição da série de vazões de Furnas no período 1972-1976.

Algumas observações interessantes podem ser extraídas da Tabela 6.3: (i) o desempenho da ESN no melhor caso foi o mesmo para todas as abordagens de treinamento, o que significa que, neste experimento específico, a adição de restrições à norma do vetor de coeficientes não trouxe benefício em termos do erro de predição; (ii) por outro lado, percebemos uma melhora do desempenho da ESN no pior caso com o uso de regularização, em particular, no caso da solução LASSO obtida com o auxílio do algoritmo IRLS; e (iii) o desvio padrão dos valores de MSE obtidos no conjunto de experimentos foi reduzido, em especial, com a inserção de

uma restrição sobre a norma L_1 do vetor de parâmetros (LASSO).

Estas evidências ressaltam um dos benefícios que motivam o uso de técnicas de regularização para o treinamento de ESNs: a rede passa a produzir uma superfície de aproximação com menor variância (ou, em outras palavras, mais suavizada), o que contribui para um melhor desempenho na etapa de teste. Além disso, como podemos perceber na Tabela 6.3, esta característica está acompanhada de um progresso de desempenho da rede na predição das vazões mensais do período 1972-1976: os valores AMSE são reduzidos com o uso da abordagem de *ridge regression* e, um pouco mais nitidamente, com o LASSO. É importante destacar que, embora o ganho de desempenho possa parecer pequeno, ele é compatível, até certo ponto, com os resultados mostrados na Seção 5.4.6, quando estruturas não-lineares foram utilizadas na camada de leitura das ESNs. Portanto, podemos dizer que a ideia de regularização representa um elemento importante a ser explorado no treinamento de redes neurais com estados de eco.

Com respeito ao número de coeficientes nulos, constatamos na Tabela 6.3 que o método LASSO tende a alcançar soluções mais parcimoniosas para o vetor de coeficientes quando comparado à solução irrestrita e ao *ridge regression*.

Outra maneira interessante de comparar as diferentes técnicas de regularização, bem como de visualizar suas características particulares, é analisar a evolução dos valores dos coeficientes da solução ótima de cada abordagem em função do valor do fator de regularização λ , ou, equivalentemente, da restrição χ imposta à norma do vetor de coeficientes. Estas curvas são exibidas nas Figuras 6.14 a 6.16, as quais também mostram a evolução do MSE referente ao conjunto de teste para cada abordagem de regularização.

Podemos perceber nas Figuras 6.14(b) a 6.16(b) que, começando na solução MSE irrestrita, o desempenho da rede pode ser melhorado quando a restrição sobre a norma do vetor de coeficientes aumenta, até que o valor ótimo é atingido, a partir do qual os métodos de regularização não são capazes de aprimorar o desempenho da rede na predição dos valores de vazões mensais.

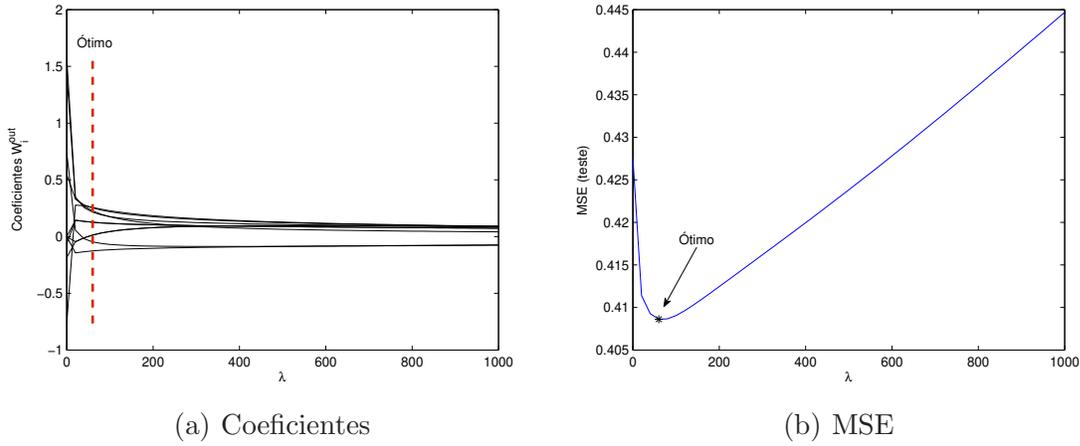


Figura 6.14: Curvas de evolução dos coeficientes do combinador linear e do MSE de teste em função de λ para o *ridge regression*.

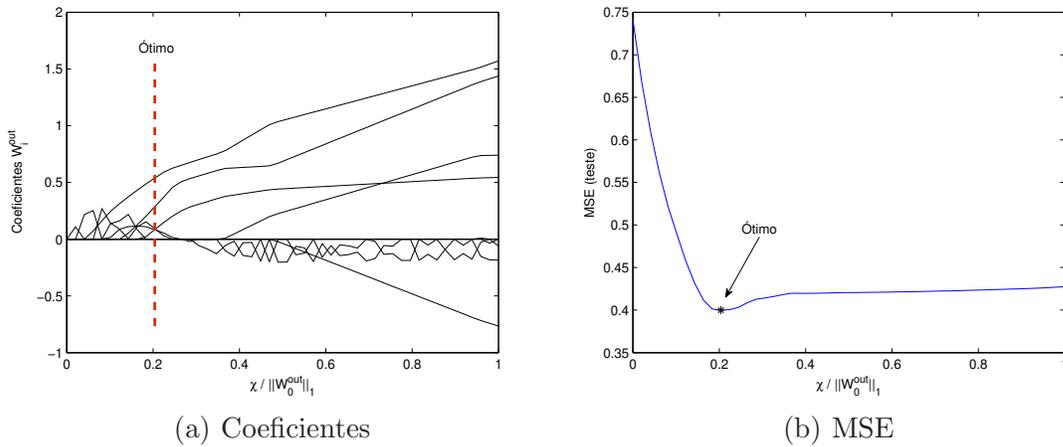


Figura 6.15: Curvas de evolução dos coeficientes do combinador linear e do MSE de teste em função da razão $\frac{\lambda}{\|W_0^{out}\|_1}$ para o LASSO - algoritmo de Tibshirani.

Com respeito ao *ridge regression*, podemos observar na Figura 6.14(a) que a magnitude dos coeficientes cai de maneira suave e, até certo ponto, lenta à medida que o fator de regularização λ aumenta. No caso do LASSO, vemos nas Figuras 6.15(a) e 6.16(a) que a redução do módulo dos coeficientes é mais rápida conforme a restrição sobre norma L_1 de W^{out} se torna mais forte. Além disso, como destacado na Seção 6.4.2, o LASSO consegue fixar um número maior de coeficientes no valor zero, oferecendo uma solução mais parcimoniosa quando comparada a do *ridge regression*.

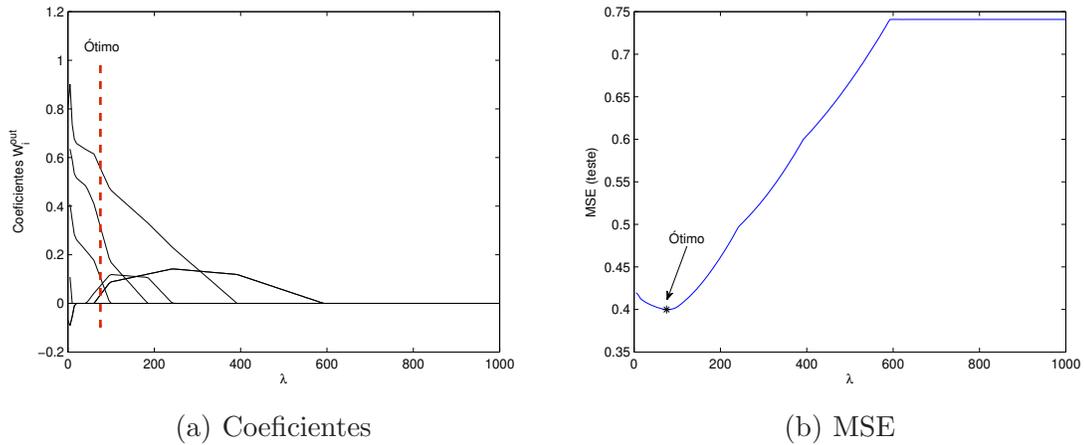


Figura 6.16: Curvas de evolução dos coeficientes do combinador linear e do MSE de teste em função de λ para o LASSO - IRLS.

Por fim, com relação às opções que computam a solução do LASSO - o algoritmo de Tibshirani (1996) e o IRLS -, não há muita diferença no desempenho da rede treinada com cada um destes métodos. Porém, devido a sua maior simplicidade operacional, o IRLS pode ser considerado uma forma adequada para implementar o LASSO.

6.5 Conclusão

Este capítulo destinou-se a uma investigação relativa ao uso de critérios diferentes daquele baseado no erro quadrático médio para a adaptação dos coeficientes do combinador linear que forma a camada de saída de redes neurais com estados de eco, tendo como objetivo alcançar uma extração mais efetiva do conteúdo estatístico dos sinais presentes na rede e, assim, aproximar o sinal desejado com maior precisão.

Neste contexto, analisamos duas alternativas vinculadas ao paradigma de aprendizado baseado em teoria de informação: os critérios de mínima entropia do erro e de máxima correntropia. Com o auxílio de métodos de estimação de PDF baseados em funções *kernel*, o MEEC oferece maior flexibilidade e robustez quando comparado ao MSE, sendo capaz de ajustar os parâmetros do sistema adaptativo enquanto modifica o formato da própria PDF do sinal de erro. O MCC, por sua vez, concentra-se apenas no valor da PDF do erro na

origem e procura maximizar o número de amostras com pequenos desvios entre o sinal de referência e a saída do sistema, com a vantagem de apresentar maior imunidade a *outliers* que o MSE.

Além disso, avaliamos também a possibilidade de explorar critérios que minimizam normas L_p diferentes da euclidiana (norma L_2). Em particular, consideramos as opções associadas às normas L_1 e L_4 do sinal de erro em virtude das características distintas das respectivas soluções ótimas.

A análise comparativa entre os critérios de adaptação foi realizada no âmbito do problema de equalização supervisionada de canais de comunicação. Verificamos que os critérios de ITL e, mais destacadamente, o MEEC, levaram a ganhos de desempenho em termos de taxa de erro de bit quando comparados ao MSE. Por outro lado, o uso das normas L_1 e L_4 não trouxe vantagens nos cenários escolhidos, exceto quando o valor da SNR era suficientemente baixo.

Adicionalmente, no contexto de um canal com estados coincidentes, as vantagens de utilizar uma estrutura recorrente para equalização foram percebidas de maneira emblemática: o desempenho das ESNs foi bastante superior ao do equalizador Bayesiano contendo várias entradas. Ademais, no momento em que as duas abordagens se equivalem em termos de BER, as ESNs requerem um número significativamente menor de combinações de elementos não-lineares que o equalizador Bayesiano, o que favorece sua aplicação e implementação.

Por fim, analisamos os benefícios de incorporar técnicas de regularização, como a *ridge regression* e o LASSO, ao processo de treinamento das ESNs. A partir dos resultados obtidos na predição da série de vazões de Furnas, constatamos que o uso de tais técnicas pode ser interessante não somente para obter uma solução mais parcimoniosa para os coeficientes da camada de saída, mas também para aprimorar a capacidade da rede responder de maneira adequada a estímulos não vistos durante a etapa de treinamento.

Projeto do Reservatório de Dinâmicas

A marca distintiva das redes neurais com estados de eco reside no fato de o processo de treinamento da estrutura se concentrar apenas nos parâmetros da camada de saída, enquanto os demais elementos da rede podem ser ajustados de maneira antecipada e, a princípio, sem recorrer a qualquer informação referente à tarefa que a rede deve realizar, como destacado no Capítulo 2.

Esta estratégia divide a arquitetura neural em duas partes que servem a propósitos diferentes: (i) a camada recorrente, denominada reservatório de dinâmicas, cuja função é criar um repertório diversificado de comportamentos dinâmicos que guardem alguma informação a respeito do conjunto de sinais de entrada, e (ii) a camada de leitura, que produz as saídas da rede por meio de combinações dos sinais gerados pelo reservatório.

O sucesso de uma ESN em uma determinada aplicação depende, portanto, da sinergia entre a camada recorrente e a camada de saída. Por isso, duas linhas de pesquisa distintas têm sido desenvolvidas pela comunidade de RC. A primeira se dedica à camada de saída e procura propor estruturas alternativas para o *readout* que sejam capazes de aproximar o sinal de referência com maior precisão. As propostas da nova arquitetura de ESN caracterizada pelo uso de um filtro de Volterra na camada de saída, apresentada no Capítulo 5, e do uso de critérios baseados em ITL e normas L_p em lugar do MSE para o ajuste dos parâmetros,

discutida no Capítulo 6, estão inseridas nesta vertente.

Por outro lado, a segunda linha de pesquisa busca compreender os efeitos das características do reservatório sobre o desempenho alcançado em uma determinada tarefa e também desenvolver métodos adequados para o projeto e adaptação da camada recorrente das ESNs. Neste contexto, o dilema é criar um repertório suficientemente diversificado de comportamentos dinâmicos sem violar o espírito de simplicidade inerente às ESNs.

A ideia de criar uma matriz de pesos aleatória e esparsa satisfaz o requisito de baixo custo computacional e permite a formação de uma memória interna associada ao histórico recente do sinal de entrada graças à ESP (Jaeger, 2001). Entretanto, parece intuitivo que uma estratégia específica de projeto do reservatório para uma determinada tarefa, i.e., que leve em consideração elementos e particularidades da tarefa em questão, produza resultados melhores que um procedimento geral e aleatório. Por isso, trabalhos recentes, entre os quais citamos (Triesch, 2005), (Schrauwen, Wardermann, Verstraeten, Steil, e Stroobandt, 2008) e (Boedecker, Obst, Mayer, e Asada, 2009), têm investigado a possibilidade de incorporar informações relevantes a respeito do sinal de entrada ao processo de elaboração do reservatório.

Neste trabalho, propomos um método não-supervisionado de projeto do reservatório de dinâmicas de ESNs caracterizado pela (i) introdução de realimentação lateral positiva entre unidades vizinhas, bem como de estímulos inibitórios entre unidades mais distantes, e (ii) pela auto-organização dos pesos de entrada.

Antes de avançarmos à descrição detalhada da nova proposta, apresentaremos um breve resumo das principais estratégias para o projeto do reservatório existentes na literatura.

7.1 Propostas Existentes para o Reservatório

Recentemente, Lukosevicius e Jaeger (2009) fizeram uma ampla revisão na área de computação com reservatórios na tentativa de reportar o desenvolvimento deste campo de pesquisa desde o seu surgimento. Neste trabalho, os autores propuseram uma classificação em três ca-

tegorias das variadas estratégias de projeto do reservatório de dinâmicas de uma rede neural com estados de eco:

- Métodos genéricos, os quais não levam em consideração qualquer informação a respeito da tarefa que a rede deve realizar.
- Pré-treinamento não-supervisionado do reservatório utilizando o sinal de entrada $\mathbf{u}(n)$, mas não o sinal desejado $\mathbf{d}(n)$.
- Pré-treinamento supervisionado do reservatório, no qual tanto o sinal de entrada $\mathbf{u}(n)$ quanto a saída desejada $\mathbf{d}(n)$ estão disponíveis.

A abordagem clássica, proposta por Jaeger (2001), que consiste em gerar aleatoriamente uma matriz de pesos que satisfaça a condição de existência de estados de eco, está inserida na primeira categoria supracitada. Apesar de sua simplicidade, uma potencial desvantagem desta estratégia é que, embora os neurônios do reservatório estejam conectados de forma esparsa, suas ativações podem ainda estar fortemente acopladas (Ozturk et al., 2007).

Com base na definição recente da ESP, a qual remete ao conceito de estabilidade matricial de Schur, Yildiz et al. (2012) ofereceram uma lista de matrizes conhecidas que são diagonalmente estáveis e que, portanto, constituem potenciais candidatas para atuarem como matriz de pesos do reservatório. Por exemplo, qualquer matriz com elementos não-negativos e com raio espectral inferior à unidade garante a existência de estados de eco.

Entretanto, o uso destas abordagens envolve a seleção de valores adequados para os pesos do reservatório e para o raio espectral, o que usualmente depende de experiência prática e/ou soluções heurísticas. Além disso, o desempenho da rede pode variar significativamente quando diferentes reservatórios construídos com o mesmo raio espectral são empregados (Ozturk et al., 2007), o que em geral não é desejável. Neste contexto, é importante mencionar que Verstraeten, Dambre, Dutoit, e Schrauwen (2010) analisaram os papéis potencialmente interdependentes de dois parâmetros - o ganho de entrada, que multiplica todos os pesos da matriz \mathbf{W}^{in} , e o raio espectral - sobre o compromisso entre capacidade de memória e habili-

dade de criar mapeamentos não-lineares, na tentativa de elucidar alguns aspectos a respeito do modo de operação do reservatório e de oferecer recomendações práticas para a escolha de seus valores dependendo da necessidade de memória/não-linearidade na tarefa envolvida.

Por causa destes fatores, a ideia de recorrer ao sinal de entrada durante o processo de adaptação do reservatório, o qual pode trazer informações relevantes a respeito da tarefa que se deseja realizar, emerge como uma possibilidade interessante e corresponde à segunda categoria supracitada, a qual pode ser subdividida em duas vertentes: 1) métodos locais e 2) globais. O termo “local” aqui significa que os parâmetros associados a um neurônio i são adaptados tendo como base nenhuma outra informação senão as ativações das unidades diretamente conectadas ao neurônio i . Com efeito, os métodos locais são quase que exclusivamente não-supervisionados, uma vez que a informação sobre o desempenho na saída é inacessível ao reservatório.

Nesta vertente, um mecanismo biológico de ajuste da excitabilidade dos neurônios conhecido como plasticidade intrínseca (em inglês, *intrinsic plasticity* (IPL)), tem atraído a atenção da comunidade de RC. Em termos simples, IPL envolve a adaptação de parâmetros intrínsecos aos neurônios tendo como objetivo maximizar a informação a respeito do sinal de entrada que cada unidade consegue transmitir para sua saída, considerando um conjunto de restrições de energia e/ou faixa de excursão da amplitude do sinal na saída (Schrauwen et al., 2008; Lukosevicius e Jaeger, 2009). Como discutido em (Triesch, 2005), este processo é equivalente a maximizar a entropia na saída de cada neurônio, o que evoca a noção de distribuições de máxima entropia (em inglês, *maximum entropy distributions* (MEDs)). Por esta razão, as regras de aprendizado baseadas em IPL tem por objetivo aproximar na saída do neurônio uma certa distribuição de probabilidade desejada, e.g., minimizando a distância de Kullback-Leibler entre a distribuição observada e a MED desejada (Schrauwen et al., 2008). Interessantemente, IPL evoca a essência da abordagem de maximização de informação (infoMax), introduzida por Bell e Sejnowski (1995), no contexto de separação de fontes e deconvolução cega.

Em (Verstraeten, Schrauwen, e Stroobandt, 2007) e (Schrauwen et al., 2008), uma regra IPL para neurônios com função de ativação do tipo tangente hiperbólica ($f : \mathbb{R} \rightarrow (-1,1)$) foi derivada, a qual leva a uma distribuição gaussiana de média nula para o vetor de estados de eco. Nestes trabalhos, mostrou-se que o uso de IPL pode melhorar, ainda que modestamente, o desempenho das ESNs em algumas aplicações, como, por exemplo, a predição de um sinal proveniente de um sistema NARMA (*nonlinear auto-regressive moving average*).

Por outro lado, o estudo conduzido por Boedecker et al. (2009) revelou que a aplicação de IPL para neurônios com funções de ativação do tipo tangente hiperbólica é limitada no que se refere às distribuições de saída que ela pode atingir. Por exemplo, no caso de um neurônio com função de ativação $\tanh(\cdot)$ e auto-realimentação, e um sinal de entrada uniformemente distribuído no intervalo $[-1, 1]$, não é possível obter uma distribuição laplaciana - a qual, a propósito, poderia ser interessante em virtude da esparsidade que tende a gerar.

Ainda na segunda categoria supracitada de estratégias de projeto do reservatório de dinâmicas, a proposta de Ozturk et al. (2007), discutida no Capítulo 5, constitui uma abordagem não-supervisionada de caráter global, e oferece uma solução algébrica para a matriz de pesos do reservatório que evoca os princípios referentes aos filtros de Kautz.

Há, por fim, abordagens que tratam o projeto do reservatório da perspectiva de aprendizado supervisionado, recorrendo a ferramentas de otimização evolutiva para ajustar as conexões recorrentes, como, por exemplo, o trabalho de Schmidhuber, Wierstra, Gagliolo, e Gomez (2007), no qual o método denominado Evolino foi utilizado.

7.2 Proposta: Interação Lateral e Auto-Organização

Dentre as categorias de estratégias de projeto destacadas na Seção 7.1, a abordagem não-supervisionada nos parece ser a mais promissora e, talvez, a mais plausível do ponto de vista biológico. No entanto, em vez de seguirmos o caminho de maximização da informação transmitida por um neurônio, que, por sua vez, leva ao problema de aproximação de

uma determinada distribuição de probabilidade desejada, sugerimos um objetivo diferente: a formação automática de grupos de neurônios que se especializem em responder a diferentes classes de padrões de entrada. Em outras palavras, propomos projetar o reservatório de estados de eco de tal modo que suas respostas, i.e., as ativações das unidades internas, passem a refletir algumas características topológicas do sinal de entrada.

Esta motivação nos fez revisitar o trabalho pioneiro de Teuvo Kohonen intitulado *Self-Organized Formation of Topologically Correct Feature Maps* (Kohonen, 1982), no qual um processo auto-organizável foi implementado por meio de um sistema relativamente simples cujas respostas adquirem a mesma ordem topológica presente nos estímulos de entrada.

O modelo considerado consiste de um arranjo uni ou bidimensional contendo N_u unidades de processamento. A i -ésima unidade está associada a um vetor de pesos $\boldsymbol{\mu}_i = [\mu_{i1} \dots \mu_{in}]^T$, $\|\boldsymbol{\mu}_i\| = 1$, $i = 1, \dots, N_u$, recebe o conjunto de sinais de entrada $\boldsymbol{\xi} = [\xi_1 \dots \xi_n]^T$ e computa a função discriminante $\eta_i = \boldsymbol{\mu}_i^T \boldsymbol{\xi} = \sum_{j=1}^n \mu_{ij} \xi_j$. Então, a unidade que produz o maior valor discriminante é selecionada e passa por um processo de adaptação, juntamente com seus vizinhos mais próximos, sendo a vizinhança aqui definida pela proximidade no arranjo escolhido. Os parâmetros destas unidades são ajustados a fim de aumentar a intensidade de suas respostas ao padrão de entrada atual, como mostra a seguinte expressão:

$$\boldsymbol{\mu}_i(t+1) = \frac{\boldsymbol{\mu}_i(t) + \alpha \boldsymbol{\xi}}{\|\boldsymbol{\mu}_i(t) + \alpha \boldsymbol{\xi}\|}, \quad (7.1)$$

onde t é um índice temporal e α é o tamanho do passo.

Note que a regra de adaptação em (7.1) rotaciona os vetores de pesos das unidades selecionadas na direção do padrão de entrada, reduzindo o ângulo entre eles, o que significa que estas unidades tendem a responder com maior intensidade a esta entrada em particular, i.e., seus valores discriminantes se tornam maiores.

Como resultado deste processo, diferentes regiões do arranjo são ativadas por estímulos de entrada distintos, de modo que é possível reconhecer algumas propriedades do espaço original dos dados e inferir similaridades entre diferentes padrões de entrada através da ativação do

mesmo grupo de unidades do arranjo. Portanto, com o auxílio de um sistema de dimensão reduzida, torna-se possível extrair algumas informação a respeito dos dados de entrada que não poderiam ser facilmente observadas no espaço original - usualmente multidimensional - dos dados (Kohonen, 2000).

O processo de auto-organização discutido até o momento pode ser descrito em termos de duas fases principais: (i) formação de um agrupamento (*cluster*) de atividade no arranjo em torno da unidade cuja ativação é máxima, e (ii) adaptação dos pesos de entrada das unidades pertencentes à região em que a atividade ficou confinada. Interessantemente, em adição ao modelo anteriormente mencionado, Kohonen (1982) também analisou um sistema neural capaz de manifestar um processo auto-organizável similar, como estudado em (von der Malsburg, 1973) e (Amari, 1980), o qual é caracterizado pela presença de interação lateral entre as unidades do sistema que é modelada pela função matemática ilustrada na Figura 7.1.

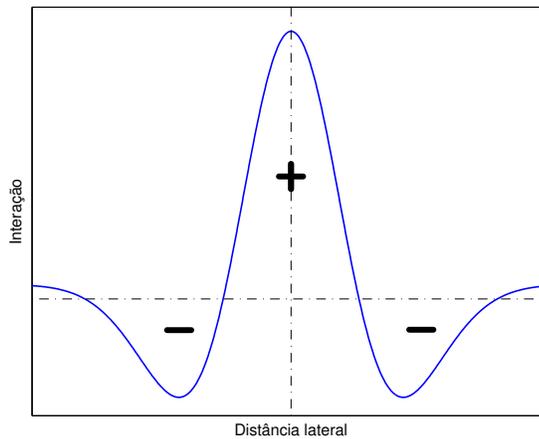


Figura 7.1: Interação lateral como função da distância.

O perfil mostrado na Figura 7.1, comumente chamado de função chapéu mexicano, indica a existência de uma região de excitação em torno da unidade de referência cuja intensidade cai à medida que a distância aumenta. Entretanto, assim que um limiar é ultrapassado, as unidades mais distantes passam a ser inibidas pelo elemento de referência. Este tipo de

interação, que recebe suporte de evidências anatômicas e fisiológicas, pode ser implementado de acordo com um modelo de rede unidimensional, como exibido na Figura 7.2, onde os círculos vazios e cheios denotam as sinapses de excitação e inibição, respectivamente.

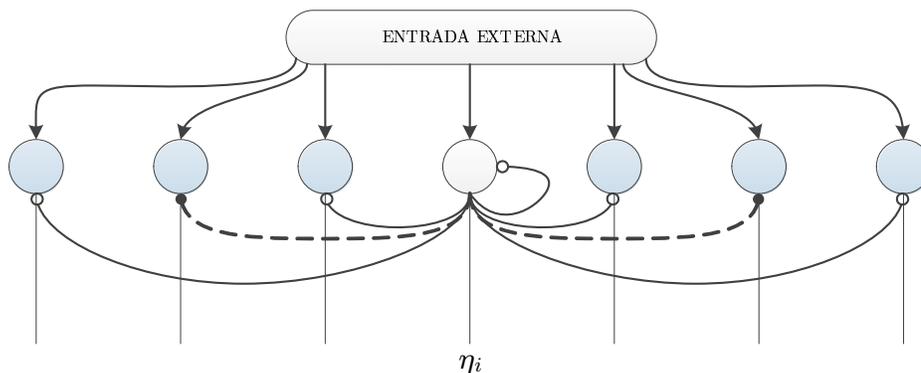


Figura 7.2: Conectividade neural que implementa o modelo matemático de interação lateral.

Portanto, a ativação de cada unidade da rede é influenciada não apenas pelo sinal de entrada, mas também pela ativação das unidades vizinhas, as quais podem tanto reforçar ou impedir que ela responda àquele padrão de entrada específico dependendo do tipo de interação lateral e da distância entre elas. Em termos matemáticos, a ativação η_i da i -ésima unidade pode ser escrita como:

$$\eta_i(t) = f\left\{\varphi_i + \sum_{k \in S_i} \gamma_k \eta_k(t - \Delta t)\right\}, \quad (7.2)$$

onde S_i indica as unidades conectadas a i , φ_i é a ativação integrada associada aos estímulos de entrada, Δt denota o atraso das ativações laterais e γ_k define a eficiência sináptica bem como o tipo de interação lateral entre i e cada um de seus vizinhos.

Este modelo particular de rede neural foi utilizado por Kohonen (1982) para demonstrar, primeiramente, que a atividade de neurônios vizinhos fica agrupada (clusterizada) na vizinhança da unidade mais intensamente ativada graças ao efeito das interações laterais. Vamos ilustrar este fenômeno através de um pequeno exemplo.

EXEMPLO 7.1. Formação de um *cluster* de atividade.

Considere um arranjo unidimensional com $N_u = 50$ unidades cujas ativações integradas associadas à entrada são determinadas por $\varphi_i = 2 \sin(\pi i/50), i = 1, \dots, N_u$, e cujas funções de ativação são dadas por:

$$f\{x\} = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{se } 0 \leq x \leq A \\ A & \text{se } x > A \end{cases} \quad (7.3)$$

Os coeficientes γ_k são definidos de acordo com uma aproximação discreta da função chapéu mexicano, mostrada na Figura 7.3, com $a = 5$ e $b = 0,1$. Então, a ativação de cada unidade é computada segundo a Equação (7.2) durante $T = 50$ instantes de tempo.

A Figura 7.4 permite algumas observações interessantes a respeito da evolução da atividade das unidades da rede ao longo do tempo de duas formas complementares: a Figura 7.4(a) mostra a ativação η_i em função da posição i de cada unidade no arranjo, enquanto a Figura 7.4(b) consiste de uma grade de N_u linhas e T colunas em que a cor de cada célula indica se a ativação daquela unidade naquele instante de tempo foi alta (cor vermelha) ou baixa (cor azul).

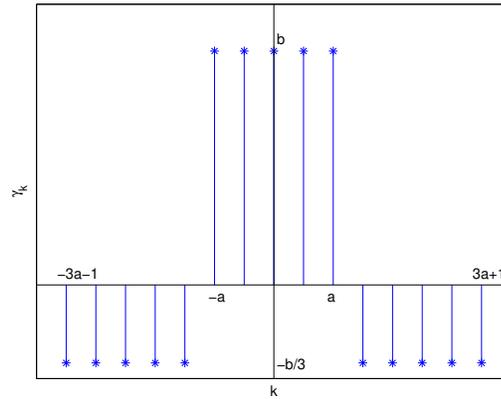
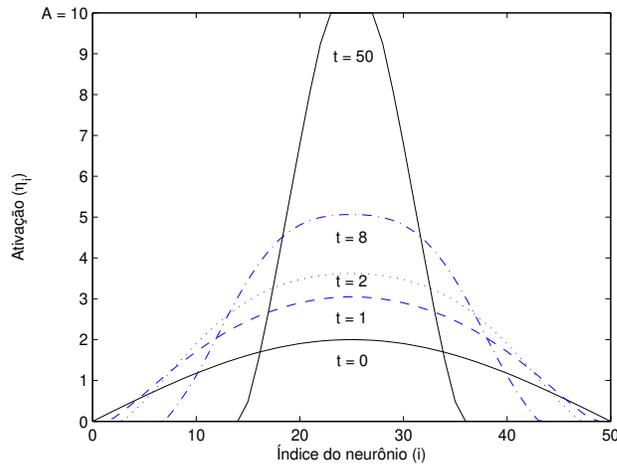


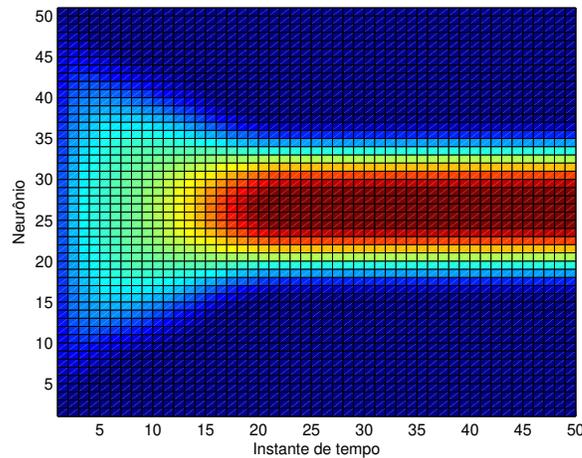
Figura 7.3: Definição dos coeficientes γ_k .

Como podemos notar na Figura 7.4, a presença de interação lateral leva à formação de

um *cluster* de atividade em torno da unidade que teve a máxima ativação inicial: enquanto as unidades situadas nas extremidades do arranjo são progressivamente inibidas e param de responder, os vizinhos mais próximos tem suas ativações aumentadas ao longo do tempo, de modo que o *cluster* se torna mais estreito.



(a) η_i versus i .



(b) Mapa de cores da atividade da rede.

Figura 7.4: Formação de um *cluster* de atividade via interação lateral.

O processo completo de auto-organização, contudo, envolve a adaptação dos parâmetros das unidades do arranjo a fim de aumentar a atividade de unidades vizinhas e, em última análise, criar múltiplos grupos que são fortemente ativados por diferentes padrões de entrada. Interessantemente, no exemplo anterior, este aspecto foi incorporado na forma da ativação

integrada φ_i . De fato, uma vez que φ_i já define um grupo inicial de unidades em torno do centro do arranjo que são mais fortemente ativadas, ele pode ser interpretado como resultando do produto escalar entre vetores de pesos bem ajustados associados a cada unidade e um certo vetor de entrada.

Por isso, após analisar os efeitos da interação lateral na atividade da rede, Kohonen (1982) introduziu uma estratégia de adaptação dos pesos de entrada do arranjo na qual as modificações são proporcionais às ativações pré-sinápticas e pós-sinápticas, o que evoca a ideia de aprendizado baseado na regra de Hebb (Hebb, 1949), e verificou a emergência de um processo de auto-organização similar ao que foi observado com o modelo anterior.

Neste trabalho, estamos particularmente interessados no modelo neural proposto por Kohonen (1982), especialmente em virtude das analogias que podemos estabelecer com as ESNs. Comparando as Equações (2.6) e (7.2), é possível reconhecer que as ativações das unidades do arranjo e dos neurônios do reservatório de uma ESN são determinadas de uma maneira muito semelhante: em ambos os casos, as ativações passadas dos elementos vizinhos, bem como o sinal de entrada atual, afetam a resposta de cada elemento.

Motivados por esta evidência, propomos projetar o reservatório de redes neurais com estados de eco tendo como objetivo auto-organizar a atividade das unidades internas de tal forma que diferentes grupos de neurônios tornem-se especializados em responder a diferentes classes de padrões de entrada. Para isto, (i) interpretamos as conexões recorrentes dentro do reservatório como promovendo interações laterais entre as unidades internas, (ii) ajustamos os pesos destas conexões segundo o perfil da função chapéu mexicano, de maneira que cada neurônio ativado estimula seus vizinhos mais próximos enquanto inibe as unidades mais distantes de responderem ao padrão de entrada, e (iii) adaptamos os pesos de entrada das unidades do reservatório a fim de obter um mapa topológico.

A abordagem proposta encoraja a formação de grupos de neurônios dentro do reservatório que são mais intensamente ativados por uma certa classe de padrões de entrada, o que, implicitamente, preserva um certo grau de diversidade de comportamentos dinâmicos. Adi-

cionalmente, ela é baseada em um ajuste prévio da matriz de pesos do reservatório \mathbf{W} , a qual permanece fixa durante a adaptação dos pesos de entrada, bem como durante o treinamento da camada de saída. Finalmente, a proposta exhibe uma forte inspiração biológica, à semelhança do trabalho original de Kohonen (1982), e pode ser empregado no caso de operação *online* da ESN, i.e., à medida que novos dados são coletados.

Portanto, o processo de adaptação de uma ESN usando o método proposto pode ser resumido da seguinte forma: 1) escolha a estrutura topológica do reservatório, a qual estabelece as relações de vizinhança entre os neurônios; 2) defina os pesos das conexões recorrentes, i.e., os elementos da matriz \mathbf{W} , de modo que cada neurônio interaja com seus vizinhos de acordo com o perfil da função chapéu mexicano; 3) ajuste os pesos de entrada para que cada neurônio, junto com seus vizinhos mais próximos no arranjo, sejam ativados pelo estímulo de entrada que esteja mais próximo de seus vetores de pesos, e 4) determine a solução ótima dos coeficientes da camada de saída. Neste capítulo, vamos restringir o estudo do método proposto a uma ESN cuja camada de saída consiste de um combinador linear cujos parâmetros são ajustados no sentido dos quadrados mínimos. Contudo, deve ficar claro que a estratégia proposta para o projeto do reservatório pode ser utilizada em conjunto com as estruturas de *readout* e critérios de adaptação discutidos nos Capítulos 5 e 6.

A estrutura de reservatório considerada neste trabalho será a de um arranjo unidimensional (Figura 7.2). Com respeito à segunda etapa supracitada, adotaremos a aproximação discreta da função chapéu mexicano mostrada na Figura 7.3. Entretanto, uma vez que a matriz de pesos do reservatório é um elemento vital para a existência de estados de eco e para a emergência de comportamentos dinâmicos, é necessário realizar alguns ajustes em \mathbf{W} .

Por exemplo, se todos os neurônios do reservatório tiverem a mesma conectividade lateral, a matriz de pesos recorrentes correspondente será simétrica e, portanto, terá somente autovalores reais, o que, à luz do que foi discutido em (Ozturk et al., 2007), pode não ser vantajoso. Uma solução simples pode ser obtida designando diferentes valores para as larguras das regiões de excitação e inibição em torno de cada neurônio, os quais são selecionados

de maneira aleatória segundo uma distribuição uniforme no intervalo $[0, a_{\max}]$. Em outras palavras, cada neurônio passa a ter seu próprio perfil de conectividade lateral.

Além disso, a fim de satisfazer as condições da ESP (Yildiz et al., 2012), aplicamos um fator de escala apropriado sobre \mathbf{W}^1 . Como consequência disto, as amplitudes iniciais das sinapses laterais, as quais dependem do parâmetro b , não influenciam a configuração final de \mathbf{W} e, em última análise, o próprio desempenho da rede. Neste caso, apenas o raio espectral (ρ_s) afeta o comportamento dinâmico da ESN².

Na terceira etapa supracitada, a adaptação dos pesos de entrada do reservatório será realizada com o auxílio de dois métodos diferentes - mapas auto-organizáveis (em inglês, *self-organizing maps* (SOMs)) (Kohonen, 2000) e a *neural gas network* (NG) -, os quais serão apresentados na próxima seção.

Finalmente, é importante enfatizar que a ideia central do método proposto é a existência de interações laterais entre as unidades do reservatório, as quais são modeladas de acordo com a função chapéu mexicano. Ademais, a perspectiva de pré-ajustar os pesos de entrada por meio de técnicas de auto-organização emerge como um complemento natural que contribui para a formação dos *clusters* de atividade. Estas características distinguem a estratégia proposta de trabalhos anteriores que analisaram o uso de métodos auto-organizáveis no contexto de RNNs e de modelos de RC, tais como (Lazar, Pipa, e Triesch, 2009), (Lukoševičius, 2010) e (Basterrech, Fyfe, e Rubino, 2011).

7.3 Estratégias de Auto-Organização

Nesta seção, descrevemos as estratégias de auto-organização utilizadas neste trabalho para adaptar os pesos de entrada das unidades da camada recorrente. Como destacado na Seção

¹Na verdade, o fator de escala é aplicado sobre $\mathbf{W}^+ = |\mathbf{W}|$, onde $w_{ij}^+ = |w_{ij}|$, de modo que \mathbf{W} se torna diagonalmente estável, segundo a definição de Schur, e a rede tem estados de eco.

²A razão entre as amplitudes dos pesos sinápticos de excitação e de inibição também poderia ser vista como um parâmetro do método proposto. Neste trabalho, adotamos a razão considerada por Kohonen (1982), como mostrado na Figura 7.3, uma vez que o uso de valores menores tenderia a remover o efeito das sinapses de inibição.

7.2, o propósito deste estágio é que as atividades das unidades do reservatório adquiram, em certo sentido, as propriedades topológicas do espaço de entrada, de maneira que unidades vizinhas tendam a ser ativadas conjuntamente por estímulos de entrada que apresentem um certo grau de similaridade.

7.3.1 Mapas Auto-Organizáveis

Seja N_u o número de unidades contidas em um arranjo unidimensional. Cada unidade está associada a um vetor de pesos $\mathbf{W}_i^{in} \in \mathbb{R}^{K \times 1}$, $i = 1, \dots, N_u$ e possui dois vizinhos - esquerdo e direito -, exceto as unidades situadas nas extremidades do arranjo, as quais possuem apenas um vizinho.

O processo de adaptação dos mapas auto-organizáveis é baseado no conceito de aprendizado competitivo (Kohonen, 2000): para cada padrão de entrada $\mathbf{p}_j(t) \in \mathbb{R}^{K \times 1}$, $j = 1, \dots, T_s$, a unidade cujo vetor de pesos tem a menor distância (euclidiana) com respeito ao padrão de entrada é considerada a vencedora e é chamada de *best matching unit* (BMU). Então, o vetor de pesos da BMU, bem como os de seus vizinhos, são ajustados na direção do padrão de entrada, como mostram as seguintes expressões:

$$\mathbf{W}_{\text{BMU}}^{in}(t+1) = \mathbf{W}_{\text{BMU}}^{in}(t) + \alpha(\mathbf{p}_j(t) - \mathbf{W}_{\text{BMU}}^{in}(t)) \quad (7.4)$$

e

$$\mathbf{W}_k^{in}(t+1) = \mathbf{W}_k^{in}(t) + e^{\frac{-1}{2\rho^2}} \alpha(\mathbf{p}_j(t) - \mathbf{W}_k^{in}(t)), \quad (7.5)$$

onde α denota a taxa de aprendizado, ρ controla o raio da função de vizinhança gaussiana e k representa o índice dos vizinhos da BMU.

A fim de construir de maneira bem-sucedida um mapa topológico associado aos eventos de entrada, é importante que a influência da unidade vencedora sobre seus vizinhos seja progressivamente reduzida durante o processo de adaptação. Além disso, a taxa de aprendizado também deve ser reduzida à medida que o processo de adaptação acontece, a fim de que as

unidades selecionadas sejam mais fortemente atraídas na direção dos padrões de entrada no início, enquanto ajustes finos são realizados em seus vetores de pesos no final do processo. Por isso, depois que todos os pesos de entrada são apresentados, i.e., assim que uma época de treinamento se completa, o raio ρ e o tamanho do passo α decaem segundo as expressões:

$$\rho(t + 1) = \rho_{\text{ini}} \exp(-t/N_T) \quad (7.6)$$

e

$$\alpha(t + 1) = \alpha_{\text{ini}} \exp(-t/N_T), \quad (7.7)$$

onde ρ_{ini} e α_{ini} representam os valores iniciais de α e ρ , respectivamente, t é o índice da época e N_T indica o número máximo de épocas de treinamento.

Finalmente, o processo completo de adaptação pode ser repetido até que um número máximo de épocas (N_T) seja atingido e/ou a taxa de aprendizado atinja um valor mínimo pré-estabelecido (α_{min}).

7.3.2 Neural Gas

O segundo método de auto-organização, denominado *neural gas network* (NG) (Martinetz e Schulten, 1991; Fritzke, 1995), difere dos SOMs em três aspectos fundamentais: (i) em vez de usar um arranjo topológico fixo de unidades neurais, a NG automaticamente cria e atualiza uma matriz de conectividade entre as unidades que espelha, até certo ponto, potenciais similaridades existentes no espaço dos dados de entrada - o termo “*gas*” vem desta propriedade; (ii) cada padrão de entrada apresentado à rede faz com que todos os vetores de pesos sejam ajustados; e (iii) a extensão de tais modificações não é determinada por uma geometria de arranjo, mas pelas distâncias relativas entre as unidades neurais no espaço de entrada.

Seja $\mathbf{p}(t)$ o padrão atualmente apresentado na entrada da rede. Para cada unidade i , o número k_i de unidades cujos vetores de pesos estão mais próximos do padrão de entrada que

o vetor de pesos da i -ésima unidade é determinado. Em termos matemáticos, $k_i = |\mathcal{A}|$, onde $|\cdot|$ é a cardinalidade de um conjunto e $\mathcal{A} = \{m \in \mathbb{N} : \|\mathbf{p}(t) - \mathbf{W}_m^{in}\| < \|\mathbf{p}(t) - \mathbf{W}_i^{in}\|\}$. De maneira similar, i_l identifica a unidade da rede para a qual existem l unidades cujos vetores de pesos estão mais próximos de $\mathbf{p}(t)$ quando comparados a $\mathbf{W}_{i_l}^{in}$. Conseqüentemente, $k_{i_l} = l$.

Então, cada vetor de pesos é adaptado de acordo com a seguinte expressão:

$$\mathbf{W}_i^{in}(t+1) = \mathbf{W}_i^{in}(t) + \epsilon e^{-k_i/\lambda}(\mathbf{p}(t) - \mathbf{W}_i^{in}(t)), \quad (7.8)$$

onde ϵ é o tamanho do passo e λ determina o número de unidades cujos vetores de pesos são significativamente modificados em cada iteração.

A NG é capaz de aprender possíveis relações de vizinhança entre vetores de pesos através da construção e atualização de uma matriz de conectividade $\mathbf{C} \in \mathbb{R}^{N_u \times N_u}$: assim que um padrão de entrada é apresentado e todos os vetores de pesos são apropriadamente modificados, uma conexão entre a unidade i_0 , cujo vetor de pesos tem a menor distância em relação ao padrão de entrada, e a unidade i_1 , cujo vetor de pesos é o segundo mais próximo a $\mathbf{p}(t)$, é criada trocando o valor de $C_{i_0 i_1}$ de zero para um.

Além disso, cada conexão (i, j) possui uma idade $t_{(i, j)}$ que pode ser alterada de três formas diferentes: (i) quando a idade de uma conexão (i, j) excede um tempo de vida máximo (T_{life}), a conexão é removida, i.e., C_{ij} recebe o valor zero; (ii) se a conexão $C_{i_0 i_1}$ que o algoritmo está tentando criar já existe, a idade correspondente $t_{(i_0, i_1)}$ é reinicializada para zero; (iii) a idade de todas as conexões que a unidade i_0 possui é incrementada após a apresentação de cada padrão.

É conveniente reduzir de maneira progressiva o parâmetro λ e o tamanho do passo ϵ à medida que as épocas de treinamento são completadas a fim de realizar um ajuste fino envolvendo apenas um pequeno conjunto de vetores de pesos mais próximos ao padrão de entrada no final do processo. Em contrapartida, aumentando o valor de T_{life} , o algoritmo inicialmente esquece com maior rapidez as conexões, enquanto, no fim do processo, as conexões são mantidas por um período mais longo uma vez que os vetores de pesos já devem ter atingido uma

configuração adequada no espaço de entrada. Por isso, após todos os padrões de entrada serem apresentados para a rede, estes parâmetros são ajustados de acordo com a seguinte expressão:

$$q(t) = q_i(q_f/q_i)^{t/N_T}, \quad (7.9)$$

onde $q \in \{\lambda, \epsilon, T_{\text{life}}\}$, q_i e q_f são os valores inicial e final do parâmetro, respectivamente, e N_T representa o número de épocas de treinamento.

Como discutido na Seção 7.2, a NG é utilizada para ajustar os pesos de entrada dos neurônios presentes no reservatório de uma ESN. Entretanto, uma vez que a interação lateral, modelada pela função chapéu mexicano, é implementada em um arranjo unidimensional, as relações de vizinhança determinadas pelo NG devem ser mapeadas em um arranjo unidimensional a fim de efetivamente observarmos a formação dos agrupamentos de atividade. Isto pode ser feito através da reordenação das unidades do reservatório, i.e., das colunas da matriz \mathbf{W} , de tal modo que neurônios adjacentes no arranjo (colunas adjacentes em \mathbf{W}) sejam, de fato, vizinhos como especificado na matriz de conectividade \mathbf{C} da NG.

Curiosamente, o fato de a NG automaticamente definir uma matriz de conectividade entre as unidades do reservatório pode ser explorado como um método alternativo para projetar a camada recorrente da ESN: a própria matriz \mathbf{C} pode ser utilizada como matriz de pesos das conexões recorrentes se aplicarmos um fator de escala apropriado para que seu maior autovalor, em módulo, seja inferior a um e a rede possua estados de eco (Yildiz et al., 2012). Esta possibilidade, bem como aquela usando a função chapéu mexicano e o esquema de reordenação das colunas de \mathbf{W} , serão analisadas neste trabalho.

7.4 Resultados Experimentais

A fim de analisarmos as potenciais vantagens da estratégia auto-organizável de projeto do reservatório, as ESNs serão aplicadas a dois problemas de processamento de informação:

1) equalização supervisionada de canais de comunicação, mais especificamente no contexto

de comunicações digitais, no qual o sinal de entrada está intrinsecamente distribuído em classes separadas associadas aos estados do canal (Haykin, 1996), como visto na Seção 4.1.3, e 2) predição de séries caóticas, onde a existência de *clusters* no perfil dos dados de entrada geralmente não é evidente (Abarbanel, 1997).

7.4.1 Metodologia

O desempenho das ESNs em ambas as tarefas será analisado em termos da média do erro quadrático médio, i.e., do valor AMSE entre o sinal desejado $d(n)$ e a saída oferecida pelas redes ($y_{\text{ESN}}(n)$) considerando um conjunto de N_{exp} experimentos independentes, o qual é determinado de acordo com a Equação (5.8). Além disso, o desvio padrão dos valores MSE obtidos em tais experimentos também é determinado a fim de termos uma medida da variação do desempenho alcançado com cada ESN.

As abreviaturas J-ESN, ASE-ESN e Rand-ESN indicam o método utilizado para o projeto do reservatório: ‘J’ refere-se à estratégia original de Jaeger (2001), descrita na Seção 5.4.1; ‘ASE’ está associada à proposta de Ozturk et al. (2007), apresentada na Seção 5.4.1, para a qual consideramos um raio espectral igual a 0,9; ‘Rand’ representa um procedimento usual em RC - os pesos do reservatório são gerados segundo uma distribuição uniforme no intervalo (0,1) e a matriz \mathbf{W} é multiplicada por uma constante a fim de que seu raio espectral seja ajustado para o valor 0,9. Em todas estas abordagens, os pesos de entrada do reservatório (W_{ij}^{in}) recebem os valores +1 e -1 com igual probabilidade.

Com respeito ao método proposto, consideramos $a_{\text{max}} = 5$ e um raio espectral igual a $\rho_s = 0,9$. Uma análise mais detalhada acerca da influência do número de neurônios no reservatório (N) e da largura máxima das regiões de excitação e inibição (a_{max}) sobre o desempenho do reservatório proposto será realizada na Seção 7.4.2 no contexto de um cenário de equalização.

As abreviaturas SOM-PESN e NG-PESN indicam se os pesos de entrada são adaptados por meio de mapas auto-organizáveis ou pela *neural gas network*. Como discutido na Seção

7.3.2, duas possibilidades diferentes podem ser exploradas quando a NG é empregada: 1) a matriz de conectividade criada pela NG pode ser usada como matriz de pesos do reservatório se aplicarmos um fator de escala de maneira que seu raio espectral atinja um valor pré-determinado (mais especificamente, $\rho_s = 0,9$); ou 2) \mathbf{W} é construída com base na função chapéu mexicano e também é multiplicada por um fator de escala para garantir a existência de estados de eco (Yildiz et al., 2012). Estas possibilidades são identificadas pelos sufixos ‘C’ e ‘MX’, respectivamente. Portanto, SOM-PESN e NG-PESN-MX diferem apenas na estratégia auto-organizável utilizada para adaptar os pesos de entrada, uma vez que o perfil da função chapéu mexicano define os pesos recorrentes em ambos os casos.

Independentemente do método de projeto do reservatório, sempre consideramos $N = 100$ neurônios não-lineares na camada recorrente e a presença de um combinador linear na camada de saída, cujos coeficientes ótimos são obtidos no sentido MSE com o auxílio de uma operação de pseudo-inversão matricial, de acordo com a Equação (2.8).

Com base em experimentos preliminares, foram adotados os seguintes valores para os parâmetros do SOM: $\rho_{\text{ini}} = 5$, $\alpha_{\text{ini}} = 0,5$ and $\alpha_{\text{min}} = 0,02$. De maneira similar, os parâmetros da NG foram definidos como: $\epsilon_i = 0,5$, $\epsilon_f = 0,05$, $\lambda_i = 30$, $\lambda_f = 0,1$, $T_{\text{life}_i} = 10$ and $T_{\text{life}_f} = 40$. Em ambos os casos, $N_u = N$ e $N_T = 100$ épocas de treinamento foram realizadas.

7.4.2 Equalização Supervisionada

O processo de treinamento das ESNs é realizado com $T_s = 1100$ amostras do sinal de informação $s(n)$, o qual é composto por amostras i.i.d pertencentes ao alfabeto binário $\{+1, -1\}$ (modulação 2-PAM) (Proakis e Salehi, 2007), mas as primeiras cem amostras servem apenas para inicializar a rede e eliminar possíveis efeitos residuais da condição inicial. O mesmo número de amostras é utilizado na etapa de teste das redes.

Em todos os casos considerados, o sinal recebido $r(n)$ é distorcido pela ação de um sistema linear e por ruído AWGN cuja variância leva a uma relação sinal-ruído de 20 dB. Devido ao caráter discreto da fonte, as amostras do sinal recebido estão naturalmente distribuídas em

diferentes *clusters* ou nuvens de dados em torno dos estados do canal.

As ESNs recebem em sua entrada um conjunto de K amostras do sinal recebido - $r(n), \dots, r(n - K + 1)$ - e fazem a estimação de $s(n)$ sem qualquer atraso de equalização. A escolha particular pelos valores $K = 1$ e $K = 2$ é motivada pelo fato de as características dos canais serem afetadas pela dimensão do vetor de sinal recebido.

Primeiro Cenário

O primeiro canal é definido pela função de transferência $H(z) = 0,5 + z^{-1}$, o qual, como destacado nos Capítulos 5 e 6, exige que a estrutura de equalização seja não-linear.

As ESNs foram treinadas e testadas segundo a metodologia descrita na seção anterior, e a Tabela 7.1 apresenta os valores AMSE obtidos com cada ESN. Os valores mostrados entre parênteses referem-se ao desvio padrão.

Rede	$K = 1$	$K = 2$
J-ESN	0,0276($\pm 0,0041$)	0,0508($\pm 0,0066$)
ASE-ESN	0,1207($\pm 0,0929$)	0,0859($\pm 0,0080$)
Rand-ESN	0,0214($\pm 0,0026$)	0,0262($\pm 0,0037$)
SOM-PESN	0,0037 ($\pm 0,00069$)	0,0064 ($\pm 0,0012$)
NG-PESN-C	0,1493($\pm 0,4544$)	0,0080($\pm 0,0012$)
NG-PESN-MX	0,0039 ($\pm 0,00070$)	0,0071 ($\pm 0,0012$)

Tabela 7.1: Valores AMSE obtidos com cada ESN em função do número de entradas considerando $H(z) = 0,5 + z^{-1}$.

É possível observar na Tabela 7.1 que o uso de interação lateral dentro do reservatório, modelada pela função chapéu mexicano, junto com a auto-organização dos pesos de entrada, foi capaz de melhorar o desempenho em termos de AMSE em, aproximadamente, uma ordem de grandeza em comparação com as redes J-ESN, ASE-ESN e Rand-ESN. Além disso, tal desempenho pode ser considerado robusto em virtude dos pequenos valores do desvio padrão, especialmente no caso em que $K = 1$, o que caracteriza outra vantagem do método proposto.

Interessantemente, podemos perceber que a NG-PESN-C tem um desempenho bastante ruim no caso em que $K = 1$, mas atinge um valor AMSE comparável aos das redes SOM-

PESN e NG-PESN-MX quando $K = 2$. Este fato pode estar relacionado às diferentes relações de vizinhança que a NG consegue criar nestas situações. Quando $K = 1$, cada neurônio pode ter, no máximo, dois vizinhos (direito e esquerdo) e a matriz de pesos recorrentes gerada é esparsa, mas cada conexão tem a mesma eficiência sináptica. Por isso, a interação lateral definida por \mathbf{W} se mostra mais limitada quando comparada àquela produzida pela função chapéu mexicano. Por outro lado, quando $K = 2$, relações de vizinhança bem definidas são estabelecidas entre cada unidade e entre os diferentes *clusters*. Este fenômeno pode ser visto na Figura 7.5, a qual mostra a configuração dos pesos de entrada do reservatório no espaço dos dados obtida com cada estratégia auto-organizável.

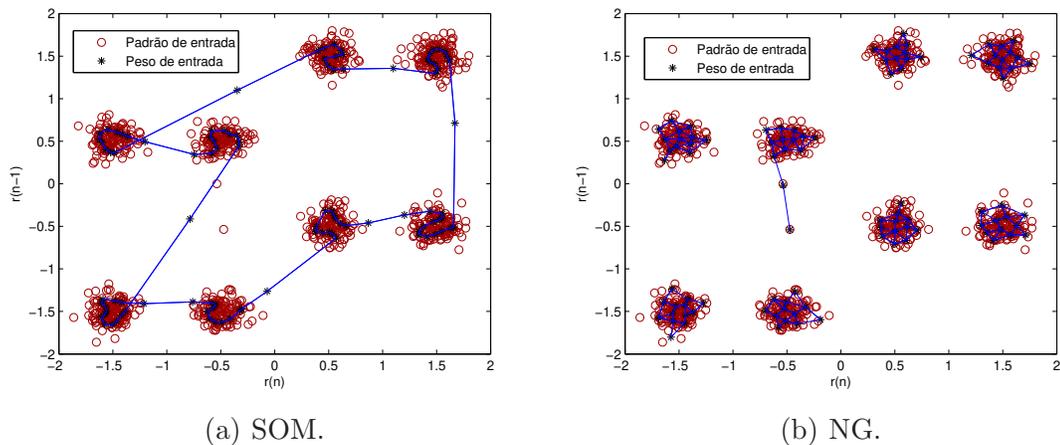


Figura 7.5: Distribuição dos pesos de entrada do reservatório após auto-organização.

Como podemos notar, ambas estratégias são capazes de identificar as classes de entrada e de apropriadamente espalhar os vetores de pesos $\mathbf{W}_i^{in}, i = 1, \dots, N$, no espaço dos dados e nos *clusters* correspondentes. Adicionalmente, é possível perceber que a NG criou um padrão de conectividade que é caracterizado pela presença de várias conexões entre unidades situadas no mesmo *cluster* (conexões *intra-cluster*), mas poucas conexões *inter-clusters*. Neste caso, somente neurônios cujos vetores de pesos estão próximos é que são efetivamente estimulados, à semelhança do que ocorre na abordagem baseada na função chapéu mexicano. Por isso, o desempenho da NG-PESN-C foi bastante parecido com aqueles obtidos com a SOM-PESN e a NG-PESN-MX para $K = 2$.

Outra característica interessante do método proposto está associada à formação de *clusters* de atividade dentro do reservatório. A Figura 7.6 exibe os mapas de cores de atividade, definidos na Seção 7.2, observados quando dois padrões diferentes são mantidos fixos na entrada da SOM-PESN durante $T = 100$ instantes de tempo: a primeira está associada ao estado do canal localizado em $(-1,5; -1,5)$, enquanto o segundo padrão vem do *cluster* situado em torno do estado $(0,5; -0,5)$.

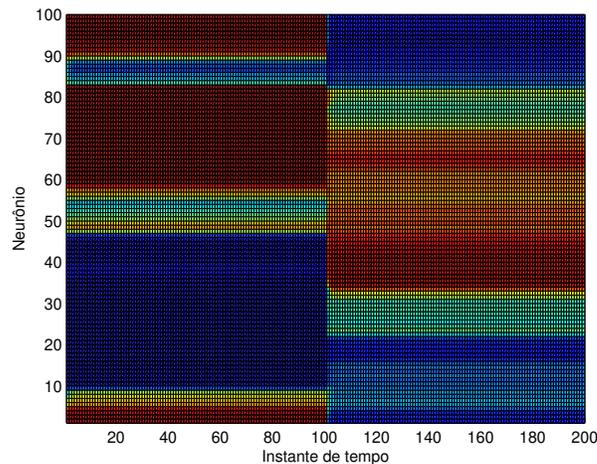


Figura 7.6: Mapas de cores de atividade obtidos com a SOM-PESN quando os estados $(-1,5; -1,5)$ e $(0,5; -0,5)$ são apresentados na entrada da rede.

Com base na Figura 7.6, é possível afirmar que, devido ao efeito combinado da interação lateral com a adaptação dos pesos de entrada, diferentes grupos de neurônios são ativados de maneira pronunciada por padrões de entrada pertencentes a classes distintas. Em outras palavras, diferentes *clusters* de atividade são adequadamente formados³.

Análise de Sensibilidade

Antes de prosseguirmos para o próximo cenário de equalização, vamos analisar a sensibilidade do método proposto em relação a dois parâmetros importantes: o número de unidades do reservatório (N) e a largura máxima das regiões de excitação e de inibição em torno de

³É pertinente mencionar que *clusters* de atividade similares aos mostrados na Figura 7.6 podem ser obtidos com a NG-PESN-MX.

cada unidade, definida por a_{\max} .

Uma vez que o número máximo de conexões laterais é dado por N e a largura da região de excitação é igual a $2a_{\max} + 1$, de acordo com a aproximação discreta da função chapéu mexicano mostrada na Figura 7.3, a_{\max} pode ser qualquer valor inteiro entre 0 e $\lceil \frac{N}{2} \rceil - 1$: no primeiro caso, a conectividade lateral é bastante estreita, uma vez que a ativação do neurônio é transmitida para sua própria entrada por meio de realimentação positiva e para apenas dois vizinhos via sinapses de inibição; por outro lado, no segundo caso, o neurônio está conectado a todas as demais unidades da rede.

Por isso, considerando diferentes tamanhos de reservatório, examinamos o desempenho da rede na tarefa de equalização descrita na seção anterior em termos do AMSE para todos os possíveis valores de a_{\max} . A Figura 7.7 exibe os valores AMSE obtidos em função de a_{\max} para cada valor de N considerando $N_{exp} = 250$ experimentos independentes.

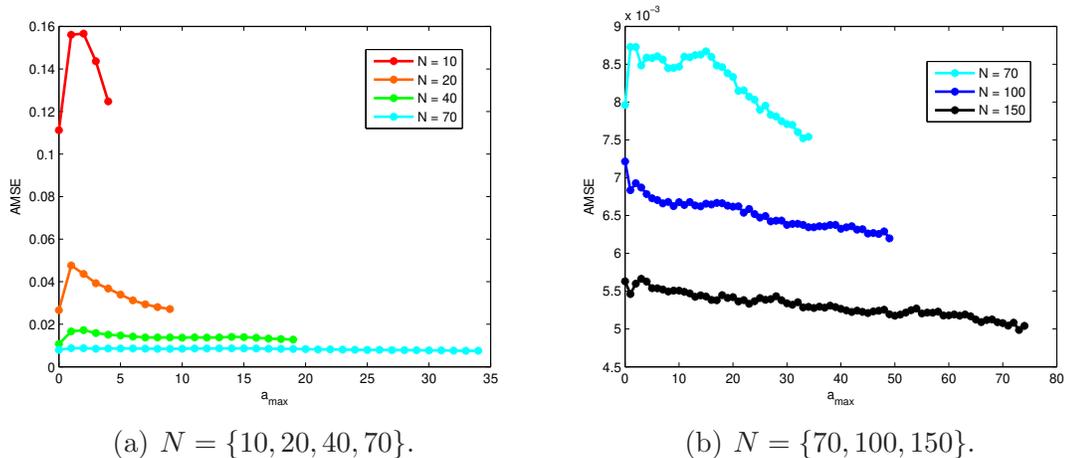


Figura 7.7: Valores AMSE em função de a_{\max} para a ESN proposta.

É possível observar na Figura 7.7 que o desempenho da ESN proposta é aprimorado à medida que o número de neurônios na camada intermediária aumenta, o que está em consonância com o comportamento geral da maioria dos modelos de RC (Lukosevicius e Jaeger, 2009). Adicionalmente, os resultados obtidos indicam que pode ser vantajoso adotar valores elevados para o parâmetro a_{\max} , especialmente quando temos muitos neurônios no reservatório, o que encoraja a emergência de neurônios cuja conectividade lateral é ampla e,

ao mesmo tempo, de unidades que interagem com apenas alguns poucos vizinhos. Apesar disto, o impacto da escolha de a_{\max} sobre o desempenho da rede tende a diminuir à medida que cresce a dimensão do reservatório: a diferença percentual entre o pior e o melhor valor AMSE associado a cada N cai de 40% ($N = 10$) para 10% ($N = 150$), aproximadamente.

Portanto, a fim de alcançar um desempenho adequado com o reservatório proposto, é aconselhável utilizar um número suficientemente elevado de unidades na camada recorrente e adotar valores altos para a máxima largura das regiões de excitação e inibição, o que tende a aumentar a diversidade de possíveis conectividades laterais dos neurônios.

Segundo Cenário

O segundo canal, caracterizado pela função de transferência $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$, introduz severas distorções sobre o sinal transmitido (Proakis e Salehi, 2007), as quais variam de acordo com o número de amostras às quais o equalizador tem acesso: quando $K = 1$, o canal apresenta estados coincidentes, o que significa que a presença de recorrência pode ser decisiva para adequadamente distinguir entre os estados mapeados no mesmo ponto, como verificado na Seção 6.3.3; por outro lado, quando $K = 2$, os estados coincidentes desaparecem, mas o uso de uma estrutura não-linear de equalização ainda é essencial.

Os valores AMSE associados a cada ESN em função dos valores de K são apresentados na Tabela 7.2.

Rede	$K = 1$	$K = 2$
J-ESN	0,0885($\pm 0,0126$)	0,1101($\pm 0,0116$)
ASE-ESN	0,1501($\pm 0,0257$)	0,1440($\pm 0,0154$)
Rand-ESN	0,0610($\pm 0,0074$)	0,0914($\pm 0,0117$)
SOM-PESN	0,0405 ($\pm 0,0053$)	0,0471 ($\pm 0,0058$)
NG-PESN-C	0,8702($\pm 3,0823$)	0,0895($\pm 0,0219$)
NG-PESN-MX	0,0394 ($\pm 0,0055$)	0,0484 ($\pm 0,0068$)

Tabela 7.2: Valores AMSE obtidos com cada ESN em função do número de entradas considerando $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$.

É possível observar que o método proposto para o projeto do reservatório oferece um

ganho de desempenho em ambos os casos - $K = 1$ e $K = 2$ -, embora as diferenças entre os valores AMSE da SOM-PESN, NG-PESN-MX e Rand-ESN não sejam tão destacadas quanto aquelas verificadas no cenário anterior. Curiosamente, enquanto o desempenho da Rand-ESN piorou quando uma amostra adicional de entrada foi disponibilizada à rede ($K = 2$), as redes SOM-PESN e NG-PESN-MX foram capazes de manter o mesmo nível de desempenho em termos de AMSE.

Além disso, podemos perceber que a NG-PESN-C não foi capaz de adequadamente equalizar o canal quando $K = 1$. Ao analisarmos o histograma dos valores MSE associados à NG-PESN-C, o qual é apresentado na Figura 7.8, é possível notar que em apenas alguns poucos experimentos é que os valores MSE obtidos com esta rede foram bastante elevados. A possibilidade de ocorrer períodos de instabilidade na saída da rede, o que resulta em valores altos de MSE, representa uma preocupação frequente no contexto de redes neurais recorrentes, mas, aqui, este efeito parece ser agravado pela ausência de fortes interações laterais entre as unidades do reservatório, como aquelas presentes nas redes NG-PESN-MX e SOM-PESN e modeladas pela função chapéu mexicano, as quais podem criar uma espécie de efeito estabilizador.

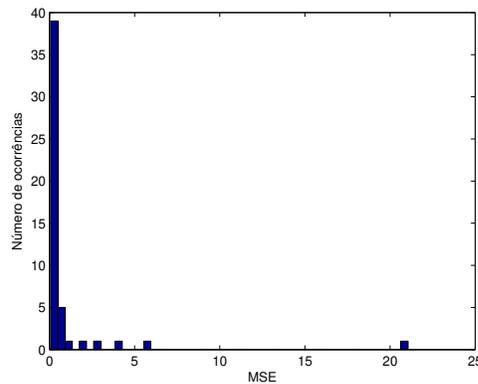


Figura 7.8: Histograma dos valores MSE obtidos pela NG-PESN-C com $K = 1$.

Em contrapartida, quando $K = 2$, uma vez que a conectividade estabelecida pela NG se torna mais intrincada, como confirmado na Figura 7.5, o desempenho da NG-PESN-C pode ser considerado competitivo. Estes fatos, portanto, enfatizam a relevância das interações

laterais dentro do reservatório de dinâmicas.

7.4.3 Predição de Séries Caóticas

Como apontado na Seção 4.3, as ESNs serão empregadas na predição de dois sistemas caóticos: o mapa logístico e o sistema de Lorenz. A partir da condição inicial, as $T_s = 1100$ amostras subsequentes do estado do sistema em questão são utilizadas na etapa de treinamento, mas as primeiras cem amostras não fazem parte das medidas de MSE. O conjunto de teste, por sua vez, é composto das $T_s = 1100$ amostras seguintes.

Mapa Logístico

Antes de apresentarmos o desempenho obtido com cada ESN na predição do estado do mapa logístico, é pertinente analisarmos o comportamento do método proposto em termos da formação de *clusters* de atividade no contexto de uma tarefa para a qual o sinal de entrada não é facilmente separado em classes distintas. Isto pode ser verificado na Figura 7.9, a qual mostra a distribuição dos pesos de entrada do reservatório ajustados com as técnicas SOM e NG⁴, junto com os padrões de treinamento referentes ao estado do mapa logístico. Como podemos observar, não é possível separar os padrões de treinamento em diferentes grupos. Mesmo assim, os pesos de entrada foram adequadamente distribuídos pelo espaço dos dados de entrada por meio das estratégias de auto-organização utilizadas neste trabalho.

Conseqüentemente, é de se esperar que a ideia de *clusters* de atividade, claramente observada no contexto do problema de equalização de canais, se torne, em certo sentido, menos evidente aqui. A Figura 7.10 exhibe os mapas de cores de atividade, definidos na Seção 7.2, obtidos quando três padrões de entrada diferentes são apresentados à rede SOM-PESN: $\mathbf{u} = [0,0612 \ 0,0155]^T$, $\mathbf{u} = [0,9627 \ 0,5965]^T$ e $\mathbf{u} = [0,0078 \ 0,9980]^T$, os quais foram escolhidos pelo fato de estarem bem separados no espaço de entrada.

⁴Devido à forte semelhança entre as configurações finais obtidas com os SOMs e a NG, mostramos apenas os resultados para o caso dos SOMs.

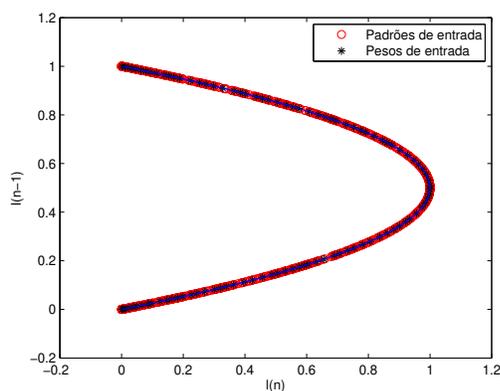


Figura 7.9: Distribuição dos pesos de entrada após auto-organização via SOM/NG.

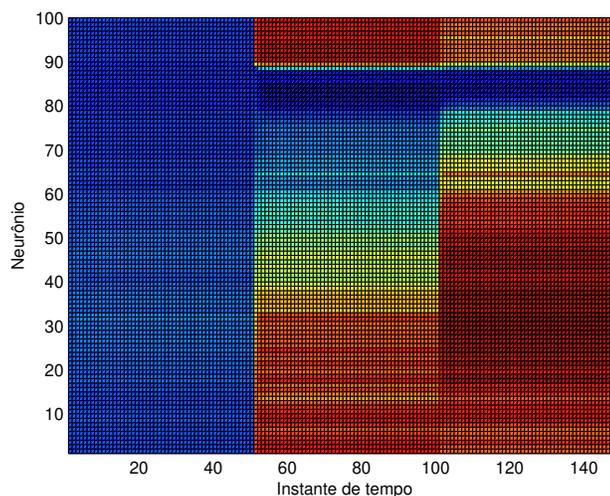


Figura 7.10: Mapas de cores de atividade referentes à SOM-PESN para três padrões de entrada diferentes associados ao mapa logístico.

O caráter não-clusterizado dos dados envolvidos tem o efeito de reduzir a distinção entre as respostas de cada neurônio a diferentes padrões de entrada, como podemos observar na Figura 7.10. Esta observação se torna mais evidente se compararmos estes mapas de atividade com aqueles observados na Figura 7.6 no contexto de equalização de canais.

Passaremos agora à análise de desempenho das diferentes ESNs na predição de um passo à frente do estado do mapa logístico. A Tabela 7.3 mostra os valores AMSE, bem como os desvios padrão, obtidos por cada ESN.

Rede	$K = 1$	$K = 2$
J-ESN	$5,90(\pm 2,90).10^{-3}$	$3,30(\pm 1,90).10^{-3}$
ASE-ESN	$7,00(\pm 3,10).10^{-3}$	$6,00(\pm 2,60).10^{-3}$
Rand-ESN	$2,41(\pm 1,32).10^{-5}$	$1,75(\pm 0,97).10^{-5}$
SOM-PESN	$1,19(\pm 0,53).10^{-6}$	$8,41(\pm 3,55).10^{-7}$
NG-PESN-C	$6,23(\pm 2,03).10^{-6}$	$8,93(\pm 13,0).10^{-4}$
NG-PESN-MX	$2,65(\pm 1,32).10^{-6}$	$1,99(\pm 0,83).10^{-6}$

Tabela 7.3: Valores AMSE associados a cada ESN em função do número de entradas na predição do estado do mapa logístico.

Como podemos observar, as redes SOM-PESN e NG-PESN-MX, as quais são caracterizadas por produzirem interações laterais dentro do reservatório que seguem o perfil da função chapéu mexicano, alcançaram os melhores desempenhos. De fato, os respectivos valores AMSE são, aproximadamente, uma ordem de grandeza menores que aqueles obtidos pela Rand-ESN para $K = 1$ e $K = 2$, e três ordens de grandeza menores que os valores AMSE associados à J-ESN e à ASE-ESN.

Interessantemente, ao contrário do que foi observado no caso de equalização de canais, a NG-PESN-C obteve um desempenho adequado com $K = 1$ - o valor AMSE é levemente superior ao da SOM-PESN -, mas seu desempenho é deteriorado quando uma amostra adicional é utilizada na predição ($K = 2$). Este comportamento pode estar relacionado ao fato que, nesta tarefa, a ideia de diferentes classes de padrões de entrada não está presente, de modo que a passagem de $K = 1$ para $K = 2$, que permitiu à NG criar uma vizinhança suficientemente adequada no âmbito de equalização, não tem o mesmo impacto aqui.

Sistema de Lorenz

Para o sistema de Lorenz, definido na Seção 4.3.2, consideramos um horizonte de predição de $h = 0,045$ segundos, à semelhança do procedimento adotado na Seção 5.4.5. A Tabela 7.4 apresenta os valores AMSE obtidos com cada ESN em função do número de entradas disponíveis.

Os resultados mostrados na Tabela 7.4 revelam que o desempenho das ESNs é significa-

tivamente aprimorado quando o reservatório proposto é empregado: para $K = 1$, os valores AMSE obtidos com a SOM-PESN são, aproximadamente, quatro ordens de grandeza menores que aqueles relacionados às redes J-ESN, ASE-ESN e Rand-ESN, enquanto, para $K = 2$, esta diferença é reduzida para uma ordem de grandeza, mas ainda pode ser considerada um progresso relevante em termos de precisão na predição. Para ambos os valores de K , o segundo melhor desempenho foi obtido com a NG-PESN-MX, o que enfatiza os benefícios trazidos pela estratégia proposta para o projeto da camada recorrente.

Rede	$K = 1$	$K = 2$
J-ESN	$1,96(\pm 1,86).10^{-4}$	$3,05(\pm 1,08).10^{-5}$
ASE-ESN	$3,90(\pm 1,40).10^{-3}$	$1,82(\pm 1,39).10^{-4}$
Rand-ESN	$8,24(\pm 2,60).10^{-4}$	$1,23(\pm 0,38).10^{-5}$
SOM-PESN	$4,50(\pm 2,78).10^{-8}$	$1,12(\pm 0,60).10^{-6}$
NG-PESN-C	$1,08(\pm 0,50).10^{-5}$	$1,90(\pm 1,30).10^{-5}$
NG-PESN-MX	$1,49(\pm 1,47).10^{-7}$	$2,55(\pm 1,71).10^{-6}$

Tabela 7.4: Valores AMSE associados a cada ESN em função do número de entradas na predição do estado do sistema de Lorenz.

O fato de os métodos baseados em auto-organização alcançarem melhores resultados quando $K = 1$ pode estar relacionado à relativa simplicidade da tarefa de clusterização envolvida neste caso. Adicionalmente, é importante lembrar que, uma vez que o sistema de Lorenz é descrito por três variáveis de estado interdependentes, a análise de uma única componente, como estamos considerando aqui, é inevitavelmente influenciada pelas variáveis restantes, as quais podem exercer um efeito parecido com um ruído sobre a série temporal associada à variável analisada. Desta forma, a presença de uma amostra de entrada adicional ($K = 2$) não apenas influencia a dificuldade da tarefa de clusterização mas pode também reforçar a interferência externa das outras variáveis de estado, o que parece ser menos pronunciado no caso em que $K = 1$.

A NG-PESN-C, por outro lado, não foi capaz de alcançar a mesma qualidade de predição da SOM-PESN e da NG-PESN-MX, e os valores AMSE obtidos por esta rede ficaram próximos àqueles associados às redes J-ESN, ASE-ESN e Rand-ESN, especialmente quando temos

$K = 2$. Esta evidência, junto com as observações discutidas nos cenários anteriores, indicam que a possibilidade de usar a matriz de conectividade criada pela NG como a própria matriz de pesos do reservatório nem sempre representa uma alternativa interessante.

7.5 Conclusão

Neste capítulo, apresentamos um novo método não-supervisionado para projetar o reservatório de redes neurais com estados de eco, no qual as conexões recorrentes são interpretadas como mecanismos que produzem interações laterais entre os neurônios, e são modeladas de acordo com o perfil da função chapéu mexicano, de maneira que cada neurônio ativado estimula positivamente seus vizinhos mais próximos enquanto envia estímulos de inibição às unidades mais distantes. Esta ideia, inspirada no trabalho pioneiro de Kohonen (1982), contribui para a formação de grupos de neurônios, denominados *clusters* de atividade, que se tornam especializados em responder a classes específicas e distintas de dados de entrada. Neste contexto, a possibilidade de auto-organizar os pesos de entrada do reservatório a fim de capturar propriedades topológicas dos dados de entrada surge como uma estratégia natural que coopera com a interação lateral para criar os *clusters* de atividade de maneira estável. Para esta tarefa, duas técnicas diferentes foram consideradas: SOMs e a NG.

O modelo proposto foi analisado no contexto de duas importantes tarefas de processamento de informação: equalização de canais e predição de séries caóticas. Estes problemas foram selecionados não apenas devido à sua relevância e às suas características desafiadoras, mas também por causa do caráter distinto dos sinais que as ESNs manipulam em cada caso: no primeiro, o sinal de entrada da ESN claramente está distribuído em diferentes classes situadas na vizinhança dos estados do canal, enquanto o sinal envolvido na segunda tarefa não apresenta esta propriedade.

Os resultados obtidos neste capítulo não apenas destacam o ganho de desempenho alcançado com o reservatório proposto quando comparado a algumas estratégias usuais em RC

para a definição da camada recorrente, como também indicam que a atividade dentro do reservatório tende a ficar clusterizada, i.e, grupos de neurônios tendem a ser fortemente ativados por padrões associados a diferentes classes, embora este fenômeno não seja tão evidente no contexto de predição caótica devido ao caráter não-clusterizado do sinal associado à série temporal.

Outra variante analisada neste trabalho, a NG-PESN-C, usa a própria matriz de conectividade criada pela NG para definir os pesos das conexões recorrentes no reservatório, e foi capaz de realizar as tarefas desejadas com, aproximadamente, a mesma qualidade que as redes propostas - SOM-PESN e NG-PESN-MX - em alguns cenários. Entretanto, seu desempenho se mostrou bastante influenciado pelo caráter dos dados envolvidos no tocante à presença de agrupamentos distintos e também pela dimensão do espaço de entrada, o que, em certo sentido, constitui uma potencial desvantagem desta abordagem em termos de robustez.

Conclusões e Perspectivas

O presente trabalho dedicou-se ao estudo das chamadas redes neurais com estados de eco e ao desenvolvimento de novas propostas para a camada de saída e para o reservatório de dinâmicas, bem como à aplicação de ESNs e dos modelos propostos a problemas relevantes de processamento de informação.

No Capítulo 2, foram apresentados os principais conceitos referentes às redes neurais com estados de eco. O principal atrativo das ESNs reside na possibilidade de resumir o processo de treinamento à adaptação dos parâmetros do combinador linear da saída, enquanto a camada recorrente, denominada reservatório de dinâmicas, pode ser pré-ajustada sem levar em consideração qualquer informação a respeito dos sinais de entrada e desejados. Esta engenhosa abordagem é possível graças à propriedade de estados de eco, a qual garante a emergência de uma memória dinâmica dentro do reservatório associada ao histórico recente do sinal de entrada.

Esta mescla singular entre aleatoriedade e adaptação supervisionada, ou, em outras palavras, entre organização e desorganização, levou-nos a reconhecer um vínculo conceitual entre as ESNs e as *extreme learning machines*, e, ainda mais importante, a enxergar nestes paradigmas recentes um renascimento das ideias conexionistas de Alan Turing (Turing, 1968). No Capítulo 3, as máquinas desorganizadas propostas por Alan Turing foram apresentadas

e os pontos de contato com os modelos de computação com reservatórios, e.g., as ESNs, e as ELMs, foram discutidos. Esta conexão histórica não somente permite que se unifiquem estas abordagens sob o termo geral de *máquinas desorganizadas*, como também presta um tributo às contribuições de Turing ao campo de redes neurais artificiais, e representa a primeira contribuição original deste trabalho.

O Capítulo 4 pode ser visto como uma espécie de interlúdio entre as contribuições deste trabalho, no qual descrevemos os fundamentos dos problemas de tratamento de informação empregados na análise de desempenho das ESNs e dos modelos propostos, a saber: equalização de canais de comunicação - tanto supervisionada quanto não-supervisionada -, separação cega de fontes, particularmente no contexto de misturas convolutivas, predição de séries caóticas e predição de séries de vazões mensais.

No Capítulo 5, propusemos uma nova arquitetura de ESN na qual o combinador linear da saída é substituído pela estrutura de um filtro de Volterra, o qual oferece a possibilidade de explorarmos um conjunto de estatísticas de ordem superior dos sinais gerados no reservatório e, ao mesmo tempo, preserva a simplicidade do processo de treinamento. A proposta é complementada pelo uso da técnica de PCA para reduzir o número de sinais efetivamente transmitidos para a camada de saída, evitando assim uma rápida expansão do número de coeficientes do filtro que precisam ser ajustados. Os resultados obtidos neste trabalho destacam o significativo ganho de desempenho conquistado com a nova camada de saída em comparação com a ESN original, especialmente nos cenários mais desafiadores, e também indicam a viabilidade da abordagem baseada nas ESNs para os problemas estudados, caracterizando-as como ferramentas promissoras para processamento de sinais.

Outra contribuição original deste trabalho referente à camada de saída de ESNs envolve a aplicação de critérios de adaptação baseados em métricas de teoria da informação - entropia e correntropia - e em normas L_p do sinal de erro como alternativas ao critério MSE. Esta ideia é motivada por algumas limitações do critério MSE em cenários de aprendizado nos quais há: 1) a presença de sinais com distribuições de cauda longa, 2) ocorrem perturbações

aleatórias de amplitude pronunciada (*outliers*) e 3) forte caráter não-linear. No âmbito do problema de equalização supervisionada, verificamos que os critérios de ITL, em especial, o de mínima entropia do erro, trazem um ganho de desempenho em termos de taxa de erro de bit quando comparados à ESN treinada de acordo com o critério MSE. Por outro lado, observamos que o uso das normas L_1 e L_4 do sinal de erro como critério de adaptação não se mostrou capaz de melhorar o desempenho da ESN em relação àquela treinada com o critério MSE. Apesar disto, a aplicação das normas L_p não deve ser totalmente descartada. O fato de a ESN treinada com a norma L_1 ter obtido o melhor desempenho quando a SNR é significativamente reduzida sugere que em cenários nos quais o ruído se torna dominante - e.g., quando a PDF do ruído pode ser modelada como uma função quase impulsiva -, esta opção pode ser vantajosa.

Neste mesmo capítulo, as vantagens de a estrutura de equalização ser dotada de realimentações foram evidenciadas: as ESNs superam de maneira notória o desempenho do equalizador ótimo do tipo *feedforward* (Bayesiano) quando o canal apresenta estados coincidentes. Ademais, mostramos que através do aumento da dimensão do vetor de sinal recebido, a BER associada ao equalizador Bayesiano consegue se aproximar daquela obtida pela ESN treinada com o MEEC. Contudo, o custo computacional da abordagem bayesiana se torna significativamente maior, o que ressalta os benefícios da presença de recorrência e, particularmente, dos modelos de computação com reservatórios, que aliam esta característica a um processo de treinamento relativamente simples.

Ainda no Capítulo 6, investigamos os efeitos que a inserção de restrições/penalizações associadas à magnitude dos parâmetros do combinador linear da saída tem sobre o desempenho da ESN. Esta ideia, conhecida como regularização, visa reduzir a possibilidade de sobre-treinamento do modelo e evitar que o desempenho do modelo esteja sujeito à variância alta. Dentre as diferentes estratégias de regularização, foram consideradas aquelas que impõem restrições à norma L_2 - *ridge regression* - e à norma L_1 - o operador LASSO - do vetor de coeficientes ajustáveis. Os resultados obtidos na predição da série de vazões referente ao

posto de Furnas indicam que estas estratégias, em especial, o LASSO, podem alcançar uma solução mais parcimoniosa para o conjunto de coeficientes da camada de saída e aprimorar a capacidade de generalização da ESN.

Por fim, no Capítulo 7, voltamos as atenções para a camada recorrente das ESNs e apresentamos uma nova estratégia de projeto do reservatório de dinâmicas. A proposta, inspirada no trabalho pioneiro de Kohonen (1982), interpreta as conexões recorrentes como um mecanismo de interação lateral entre os neurônios e ajusta os pesos destas sinapses de maneira a reproduzir o perfil da função chapéu mexicano. Assim, cada neurônio envia sinais de excitação a seus vizinhos mais próximos e sinais de inibição às unidades mais distantes, tendo como objetivo a formação de grupos de neurônios que se especializem em responder com maior intensidade a padrões de entrada pertencentes a diferentes classes, i.e., que apresentam características topológicas distintas.

Esta ideia é complementada pela auto-organização dos pesos de entrada do reservatório, os quais são espalhados no espaço dos dados de tal forma que relações de similaridade existentes nos padrões de entrada sejam mapeadas, até certo ponto, nas relações de vizinhança dos neurônios, o que contribui para a formação de diferentes *clusters* de atividade dentro do reservatório. Para esta etapa, foram consideradas duas estratégias auto-organizáveis, a saber, os mapas auto-organizáveis e a *neural gas network*. Os resultados obtidos no contexto dos problemas de equalização de canais e de predição de séries caóticas mostram que o novo reservatório pode trazer ganhos de desempenho para a ESN em ambas as tarefas quando comparado com algumas estratégias usuais de projeto da camada recorrente. Além disso, verificamos que a atividade neural dentro do reservatório tende a ficar agrupada em torno de diferentes unidades, embora este comportamento seja menos evidente quando os sinais de entrada não podem ser adequadamente separados em classes distintas.

As contribuições alcançadas nesta tese abrem várias possibilidades interessantes de continuidade do trabalho. Por exemplo, a partir da conexão estabelecida entre as máquinas desorganizadas de Turing e as ESNs, surge a ideia de estender o conceito de estados de eco

para o domínio discreto ou, em outras palavras, de modificar uma rede de Turing para gerar uma versão Booleana de uma ESN. Esta perspectiva pode ser particularmente útil para simplificar os sistemas que operam sobre alfabetos finitos, os quais tipicamente dão origem a problemas de otimização combinatória. Além disso, tal modelo poderia ser aplicado como uma ferramenta de processamento de sinais no contexto de campos finitos (Gutch et al., 2012), um ramo de pesquisa que pode trazer contribuições relevantes em aplicações como codificação de canal e análise de dados genômicos.

Esta unificação entre as ESNs e as ELMs motiva: 1) a extensão do estudo envolvendo o uso dos critérios baseados em ITL e em normas L_p para outras máquinas desorganizadas, como as ELMs, e para o cenário de equalização não-supervisionada; e 2) a possibilidade de utilizar a nova camada de saída no âmbito das *extreme learning machines*.

Outra perspectiva de prosseguimento deste trabalho consiste em combinar as propostas apresentadas nos Capítulos 5 a 7 para a camada de saída e para a camada recorrente de uma ESN. Nesta vertente, é pertinente investigar o uso de critérios baseados em ITL para o treinamento da nova arquitetura de ESN, cuja camada de saída é formada por um filtro de Volterra, bem como o emprego do reservatório proposto em conjunto com uma camada de saída não-linear, como a arquitetura proposta ou o modelo híbrido de Butcher et al. (2013).

Por fim, com relação ao reservatório proposto, podemos apontar os seguintes aspectos como metas futuras: 1) explorar diferentes estruturas de vizinhança no reservatório; e 2) elaborar estratégias que definam de maneira inteligente o perfil de conectividade lateral de cada neurônio, e.g., recorrendo à noção de redes complexas (Strogatz, 2001).

Trabalhos publicados durante o doutorado

Listamos abaixo os trabalhos publicados e aceitos para publicação durante o período do doutorado.

Artigos publicados em periódicos:

1. Levy Boccato, Rafael Krummenauer, Romis Attux e Amauri Lopes, *Improving the Efficiency of Natural Computing Algorithms in DOA Estimation Using a Noise Filtering Approach*, Circuits, Systems, and Signal Processing, 2013 (Aceito).
2. Levy Boccato, Amauri Lopes, Romis Attux e Fernando J. Von Zuben, *An extended echo state network using Volterra filtering and principal component analysis*, Neural Networks, vol. 32, págs. 292-302, 2012.
3. Levy Boccato, Rafael Krummenauer, Romis Attux e Amauri Lopes, *Application of natural computing algorithms to maximum likelihood estimation of direction of arrival*, Signal Processing, vol. 92, págs. 1338-1352, 2012.
4. Hugo V. Siqueira, Levy Boccato, Romis Attux e Christiano Lyra Filho, *Echo State Networks in Seasonal Streamflow Series Prediction*, Learning and Nonlinear Models, vol. 10, n° 3, págs. 181-191, 2012.
5. Levy Boccato, Everton S. Soares, Marcos M. L. P. Fernandes, Diogo C. Soriano e Romis Attux, *Unorganized Machines: From Turing's Ideas to Modern Connectionist Approaches*, International Journal of Natural Computing Research, vol. 2, n° 4, págs. 1-16, 2011.

Capítulos de livro:

6. Hugo V. Siqueira, Levy Boccato, Romis Attux e Christiano Lyra Filho, *Echo State Networks for Seasonal Streamflow Series Forecasting*, em *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, Lecture Notes in Computer Science, vol. 7435, págs. 226-236, 2012.
7. Hugo V. Siqueira, Levy Boccato, Romis Attux e Christiano Lyra Filho, *Echo State Networks and Extreme Learning Machines: a Comparative Study on Seasonal Streamflow Series Prediction*, em *Neural Information Processing*, Lecture Notes in Computer Science, vol. 7664, págs. 281-290, 2012.

Trabalhos completos publicados em anais de congressos:

8. André Gonçalves, Levy Boccato, Romis Attux e Fernando José Von Zuben, *A multi-Gaussian component EDA with restarting applied to direction of arrival tracking*, IEEE Congress on Evolutionary Computation (CEC 2013), Cancún, 2013. (Aceito)
9. Levy Boccato, Daniel G. Silva, Denis Fantinato, Kenji Nose Filho, Rafael Ferrari, Romis Attux, Aline Neves, Jugurta Montalvão e João M. T. Romano, *Error Entropy Criterion in Echo State Network Training*, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, págs. 35-40, 2013.
10. Guilherme. P. Coelho, Celso. C. Barbante, Levy Boccato, Romis Attux, José R. de Oliveira e Fernando J. Von Zuben, *Automatic feature selection for BCI: An analysis using the Davies-Bouldin index and extreme learning machines*, International Joint Conference on Neural Networks (IJCNN 2012), págs. 985-992, 2012.
11. Levy Boccato, Diogo C. Soriano, Romis Attux e Fernando J. Von Zuben, *Performance analysis of nonlinear echo state network readouts in signal processing tasks*, International Joint Conference on Neural Networks (IJCNN 2012), págs. 2439-2446, 2012.
12. Everton Z. Nadalin, Levy Boccato, Romis Attux, Leonardo T. Duarte, Amauri Lopes, João M. T. Romano e R. Suyama, *Multimodal optimization in the context of Sparse Component Analysis*, IEEE Symposium Series on Computational Intelligence (SSCI 2011), págs. 85-91, 2011.
13. Levy Boccato, Amauri Lopes, Romis Attux e Fernando J. Von Zuben, *An echo state network architecture based on volterra filtering and PCA with application to the channel equalization problem*, International Joint Conference on Neural Networks (IJCNN 2011), págs. 580-587, 2011.
14. Hugo V. Siqueira, Levy Boccato, Romis Attux e Christiano Lyra Filho, *Previsão de Séries de Vazões com Redes Neurais de Estados de Eco*, X Congresso Brasileiro de Inteligência Computacional - CBIC 2011, págs. 1-8, 2011.
15. Paulo S. Prampero, Levy Boccato e Romis Attux, *Multimodal Optimization as a Clustering Task: General Formulation and Application in the Context of Particle Swarm*, X Congresso Brasileiro de Inteligência Computacional - CBIC 2011, págs. 1-7, 2011.

Resumos expandidos publicados em anais de congressos:

16. Levy Boccato, Everton S. Soares, Marcos M. L. P. Fernandes, Diogo C. Soriano e Romis Attux, *Uma Discussão acerca das Contribuições de Alan Turing à Área de Redes Neurais Artificiais - Parte I: Aspectos Introdutórios*, V Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial (V EADCA), págs. 30-33, 2012.
17. Levy Boccato, Everton S. Soares, Marcos M. L. P. Fernandes, Diogo C. Soriano e Romis Attux, *Uma Discussão acerca das Contribuições de Alan Turing à Área de Redes Neurais Artificiais - Parte II: Desorganização Revisitada*, V Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial (V EADCA), págs. 34-37, 2012.
18. Marcos M. L. P. Fernandes, Levy Boccato, Hugo V. Siqueira, Romis Attux, Fernando J. Von Zuben e Christiano Lyra Filho, *Comparação Preliminar de Desempenho entre Extreme Learning Machines e Perceptrons de Múltiplas Camadas*, V Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial (V EADCA), págs. 47-50, 2012.
19. Paulo S. Prampero, Levy Boccato e Romis Attux, *Otimização Multimodal Baseada em Clusterização.*, V Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial (V EADCA), págs. 22-25, 2012.

Resumos publicados em anais de congressos:

20. Hugo V. Siqueira, Levy Boccato, Romis Attux e Christiano Lyra Filho, *Neural networks for streamflow series forecasting: a comparative study between echo state networks and MLPs*, 25th European Conference on Operational Research (EURO), 2012.

Equalizador Bayesiano

O equalizador ótimo de memória finita (*feedforward*) no sentido de mínima probabilidade de erro de símbolo é determinado por meio do critério MAP (*maximum a posteriori*) e é denominado equalizador bayesiano (Chen, Mulgrew, e McLaughlin, 1993; Ferrari, 2005).

Qualquer equalizador de memória finita pode ser interpretado como um decisor que divide o espaço ao qual pertence o vetor de sinal recebido $\mathbf{r}(n)$ em partições associadas a cada um dos possíveis valores do símbolo transmitido ($s(k-d)$). O lugar geométrico que delimita estas partições é chamado de fronteira de decisão. Desta forma, o projeto do equalizador bayesiano consiste em definir a fronteira de decisão que leve à maior probabilidade de acerto do símbolo detetado no receptor.

Para isto, utiliza-se o critério MAP, de maneira que a decisão é definida com base na probabilidade de o símbolo original ter sido $s(n-d) = s$ dado que o vetor de sinal recebido foi $\mathbf{r}(n)$. Ou seja, a partir da observação feita, de acordo com o critério MAP, o equalizador deve decidir pelo símbolo s mais provável. Isto equivale a decidir pelo símbolo que maximiza a probabilidade *a posteriori*, i.e., a probabilidade condicional $P(s(n-d) = s | \mathbf{r}(n))$ (Papoulis, 1991; Leon-Garcia, 2008):

$$s_{\text{MAP}} = \arg \max_s P(s(n-d) = s | \mathbf{r}(n)). \quad (\text{A.1})$$

De acordo com a regra de Bayes, a probabilidade *a posteriori* do símbolo transmitido pode ser expressa da seguinte forma:

$$P(s(n-d) = s | \mathbf{r}(n)) = \frac{p(\mathbf{r}(n) | s(n-d) = s) \cdot P(s(n-d) = s)}{p(\mathbf{r}(n))}, \quad (\text{A.2})$$

onde $P(s(n-d) = s)$ é a probabilidade *a priori* do símbolo transmitido e $p(\mathbf{r}(n) | s(n-d) = s)$ é a PDF condicional do vetor de sinal recebido dado o conhecimento do símbolo transmitido.

Como apresentado no Capítulo 4, cada amostra de sinal recebido resulta da ação do canal sobre o símbolo transmitido, i.e., da IIS, e da presença de ruído. Logo, o vetor de sinal recebido pode ser escrito como

$$\mathbf{r}(n) = \mathbf{c} + \boldsymbol{\eta}(n), \quad (\text{A.3})$$

onde \mathbf{c} representa um estado do canal.

Uma vez que cada estado do canal está associado a um dos possíveis símbolos transmitidos, como vimos na Seção 4.1.3, um erro de decisão do equalizador ocorre quando o ruído é suficientemente grande a ponto de fazer com que o vetor de sinal recebido cruze a fronteira de decisão, saindo da região correspondente ao estado verdadeiro e passando para a região associada a outro estado, o qual, por sua vez, está relacionado a um símbolo $s(n-d)$ diferente do que foi transmitido. Entretanto, se o ruído gerar uma observação $\mathbf{r}(n)$ que ocupe uma posição dentro da região associada a outro estado, mas que esteja relacionado ao mesmo símbolo que aquele que foi transmitido, a decisão do equalizador será correta, embora a distorção introduzida pelo canal tenha sido significativa.

Usando a lei da probabilidade total (Leon-Garcia, 2008), a PDF condicional $p(\mathbf{r}(n) | s(n-d) = s)$ pode ser escrita como:

$$p(\mathbf{r}(n) | s(n-d) = s) = \sum_{\mathbf{c}_j \in \mathbf{C}_d^s} p(\mathbf{r}(n) | \mathbf{c}_j) P(\mathbf{c}_j), \quad (\text{A.4})$$

onde \mathbf{C}_d^s é o conjunto de estados para os quais $s(n-d) = s$ e $P(\mathbf{c}_j)$ é a probabilidade de

ocorrência destes estados.

Usando a Equação (A.3), a PDF $p(\mathbf{r}(n)|\mathbf{c}_j)$ pode ser escrita como:

$$p(\mathbf{r}(n)|\mathbf{c}_j) = p_{\boldsymbol{\eta}}(\mathbf{r}(n) - \mathbf{c}_j), \quad (\text{A.5})$$

onde $p_{\boldsymbol{\eta}}(\cdot)$ denota a PDF do vetor de ruído.

Neste trabalho, vamos determinar o mapeamento entrada-saída ou a função de decisão do equalizador bayesiano para o caso em que o sinal transmitido pertence a um alfabeto binário $\mathbb{A} = \{+1, -1\}$ (modulação 2-PAM) e o vetor de entrada do equalizador é real, i.e., $\mathbf{r}(n) \in \mathbb{R}^m$. Além disso, consideraremos dois modelos para o vetor $\boldsymbol{\eta}(n)$: ruído branco aditivo gaussiano (em inglês, *additive white Gaussian noise* (AWGN)) e ruído branco aditivo laplaciano (em inglês, *additive white Laplace noise* (AWLN)).

A.1 Canal AWGN

Uma vez que temos apenas dois símbolos possíveis para o sinal $s(n-d)$, o critério MAP, apresentado na Equação (A.1), se resume a comparar $P(s(n-d) = +1|\mathbf{r}(n))$ com $P(s(n-d) = -1|\mathbf{r}(n))$. Portanto, a função de decisão do equalizador bayesiano pode ser escrita como

$$f_{\text{Bayes}}(\mathbf{r}(n)) = P(s(n-d) = +1|\mathbf{r}(n)) - P(s(n-d) = -1|\mathbf{r}(n)), \quad (\text{A.6})$$

e a decisão é dada por:

$$P(s(n-d) = +1|\mathbf{r}(n)) \underset{s_{\text{MAP}}=-1}{\overset{s_{\text{MAP}}=+1}{\gtrless}} P(s(n-d) = -1|\mathbf{r}(n)), \quad (\text{A.7})$$

ou, equivalentemente,

$$s_{\text{MAP}} = \text{sign}(f_{\text{Bayes}}(\mathbf{r}(n))). \quad (\text{A.8})$$

Para o canal AWGN, a Equação (A.5) é dada por:

$$p(\mathbf{r}(n)|\mathbf{c}_j) = (2\pi\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right). \quad (\text{A.9})$$

Como $P(s(n-d) = +1) = P(s(n-d) = -1) = \frac{1}{2}$, cada estado do canal tem a mesma probabilidade de ocorrência, i.e., $P(\mathbf{c}_j) = \frac{1}{N_{\text{estados}}}$. Assim, usando as Equações (A.4) e (A.9), as PDFs $p(\mathbf{r}(n)|s(n-d) = +1)$ e $p(\mathbf{r}(n)|s(n-d) = -1)$ equivalem a:

$$p(\mathbf{r}(n)|s(n-d) = +1) = \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right), \quad (\text{A.10})$$

onde \mathbf{C}_d^+ é o conjunto dos estados para os quais $s(n-d) = +1$, e

$$p(\mathbf{r}(n)|s(n-d) = -1) = \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right), \quad (\text{A.11})$$

onde \mathbf{C}_d^- é o conjunto dos estados para os quais $s(n-d) = -1$, respectivamente.

Substituindo a Equação (A.2) em (A.6), obtemos:

$$f_{\text{bayes}}(\mathbf{r}(n)) = \frac{1}{2p(\mathbf{r}(n))} (p(\mathbf{r}(n)|s(n-d) = +1) - p(\mathbf{r}(n)|s(n-d) = -1)). \quad (\text{A.12})$$

Como o termo $\frac{1}{2p(\mathbf{r}(n))}$ é um fator de escala positivo, (A.12) pode ser reescrita como

$$f'_{\text{bayes}}(\mathbf{r}(n)) = p(\mathbf{r}(n)|s(n-d) = +1) - p(\mathbf{r}(n)|s(n-d) = -1). \quad (\text{A.13})$$

Substituindo as Equações (A.10) e (A.11) em (A.13), obtemos:

$$\begin{aligned}
f'_{\text{bayes}}(\mathbf{r}(n)) &= \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} p(\mathbf{r}(n)|\mathbf{c}_j) - \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} p(\mathbf{r}(n)|\mathbf{c}_i) \\
&= \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right) \\
&\quad - \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_i\|_2^2}{2\sigma_\eta^2}\right) \\
&= \left\{ \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_i\|_2^2}{2\sigma_\eta^2}\right) \right\} \\
&\quad \times \frac{1}{N_{\text{estados}}} (2\pi\sigma_\eta^2)^{-m/2}.
\end{aligned} \tag{A.14}$$

Uma vez que o fator $\frac{1}{N_{\text{estados}}} (2\pi\sigma_\eta^2)^{-m/2}$ é sempre positivo, ele não afeta o sinal da função de decisão e, por isso, pode ser desconsiderado. Logo, a função de decisão do equalizador bayesiano pode ser escrita como:

$$\begin{aligned}
f_{\text{Bayesiano}}^{\text{AWGN}}(\mathbf{r}(n)) &= \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_i\|_2^2}{2\sigma_\eta^2}\right) \\
&= \sum_{j=1}^{N_{\text{estados}}} \zeta_j \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_2^2}{2\sigma_\eta^2}\right),
\end{aligned} \tag{A.15}$$

onde $\zeta_j = +1$ se $\mathbf{c}_j \in \mathbf{C}_d^+$ e $\zeta_j = -1$ se $\mathbf{c}_j \in \mathbf{C}_d^-$.

Como podemos observar na Equação (A.15), a função de decisão do equalizador bayesiano apresenta um caráter não-linear e é completamente definida com base nos estados do canal e nas estatísticas do ruído.

A.2 Canal AWLN

Para o canal AWLN, a Equação (A.5) é dada por:

$$p(\mathbf{r}(n)|\mathbf{c}_i) = \left(\frac{1}{2b}\right)^m \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_i\|_1}{b}\right), \quad (\text{A.16})$$

onde $b = \sigma_\eta \frac{\sqrt{2}}{2}$.

Usando o mesmo raciocínio da seção anterior, podemos escrever as PDFs condicionais $p(\mathbf{r}(n)|s(n-d) = +1)$ e $p(\mathbf{r}(n)|s(n-d) = -1)$ como:

$$p(\mathbf{r}(n)|s(n-d) = +1) = \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_1}{\sqrt{\sigma_\eta^2/2}}\right), \quad (\text{A.17})$$

onde \mathbf{C}_d^+ é o conjunto dos estados para os quais $s(n-d) = +1$, e

$$p(\mathbf{r}(n)|s(n-d) = -1) = \frac{1}{N_{\text{estados}}} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} (2\sigma_\eta^2)^{-m/2} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_1}{\sqrt{\sigma_\eta^2/2}}\right), \quad (\text{A.18})$$

onde \mathbf{C}_d^- é o conjunto dos estados para os quais $s(n-d) = -1$, respectivamente.

Substituindo as Equações (A.17) e (A.18) em (A.6) e eliminando os fatores de escala positivos, a função de decisão do equalizador bayesiano para o caso de canal AWLN pode ser definida de acordo com a seguinte expressão:

$$f_{\text{Bayesiano}}^{\text{AWLN}}(\mathbf{r}(n)) = \sum_{j=1}^{N_{\text{estados}}} \zeta_j \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{c}_j\|_1}{\sqrt{\sigma_\eta^2/2}}\right), \quad (\text{A.19})$$

onde $\zeta_j = +1$ se $\mathbf{c}_j \in \mathbf{C}_d^+$ e $\zeta_j = -1$ se $\mathbf{c}_j \in \mathbf{C}_d^-$.

Referências Bibliográficas

- Abarbanel, H. D. I. (1997). *Analysis of observed chaotic data*. Springer.
- Adali, T. (1999). Why a nonlinear solution for a linear problem? Em *Proceedings of IEEE Workshop on Neural Networks for Signal Processing* (págs. 157–165).
- Agrawal, G. (2002). *Fiber-optic communication system* (3ª ed.). Wiley-Interscience.
- Alexandrov, P., e Urysohn, P. (1929). *Mémoire sur les espaces topologiques compacts*. Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam, Proceedings of the Section of Mathematical Sciences.
- Amari, S. (1980). Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42, 339–364.
- Austin, M. (1967). *Decision feedback equalization for digital communications over dispersive channels* (Relat. Técnico N° 461). MIT Research Laboratory of Electronics.
- Babinec, S., e Pospíchal, J. (2006). Merging echo state and feedforward neural networks for time series forecasting. Em *Proceedings of the 16th International Conference on Neural Networks (ICANN 2006)* (Vol. 4131, págs. 367-375).
- Ballini, R. (2000). *Análise e previsão de vazões utilizando séries temporais, redes neurais e redes neurais nebulosas*. Tese de doutorado, Faculdade de Engenharia Elétrica e Computação (FEEC) - UNICAMP.
- Barry, J. R., Lee, E. A., e Messerschmitt, D. G. (2003). *Digital communication* (3ª ed.). Springer.

- Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2), 525–536.
- Basterrech, S., Fyfe, C., e Rubino, G. (2011). Self-organizing maps and scale-invariant maps in echo state networks. Em *11th International Conference on intelligent Systems Design and Applications (ISDA)* (págs. 94–99).
- Belfiore, C. A., e Park Jr., J. H. (1979). Decision feedback equalization. *Proceedings of the IEEE*, 67(8), 1143–1156.
- Bell, A. J., e Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7, 1129–1159.
- Benveniste, A., Goursat, M., e Ruget, G. (1980). Robust identification of a non-minimum phase system: blind adjustment of a linear equalizer in data communications. *IEEE Transactions on Automatic Control*, 25(3), 385–399.
- Bishop, C. M. (2006). *Pattern recognition and machine learning* (1ª ed.). Springer.
- Boccatto, L., Lopes, A., Attux, R., e Von Zuben, F. J. (2011). An echo state network architecture based on Volterra filtering and PCA with application to the channel equalization problem. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2011)* (págs. 580–587).
- Boccatto, L., Lopes, A., Attux, R., e Von Zuben, F. J. (2012). An extended echo state network using volterra filtering and principal component analysis. *Neural Networks*, 32, 292–302.
- Boccatto, L., Soares, E. S., Fernandes, M. M. L. P., Soriano, D. C., e Attux, R. (2011). Unorganized machines: from Turing’s ideas to modern connectionist approaches. *International Journal of Natural Computing Research*, 2(4), 1–16.
- Boedecker, J., Obst, O., Mayer, N. M., e Asada, M. (2009). Initialization and self-organized optimization of recurrent neural network connectivity. *HFSP Journal*, 3(5), 340–349.
- Box, G., Jenkins, G., e Reinsel, G. C. (1994). *Time series analysis, forecasting and control*

(3^a ed.). Holden Day.

- Buehner, M., e Young, P. (2006). A tighter bound for the echo state property. *IEEE Transactions on Neural Networks*, 17(3), 820–824.
- Bush, K. A., e Anderson, C. W. (2005). Modeling reward functions for incomplete state representations via echo state networks. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2005)* (Vol. 5, págs. 2995–3000).
- Bush, K. A., e Anderson, C. W. (2006). Exploiting iso-error pathways in the N, k-plane to improve echo state network performance. Em *Proceedings of the 2006 Conference of Neural Information Processing Systems (NIPS 2006)*.
- Butcher, J. B., Verstraeten, D., Schrauwen, B., Day, C., e Haycock, P. (2010). Extending reservoir computing with random static projections: a hybrid between extreme learning and RC. Em *Proceedings of ESANN'2010 - European Symposium on Artificial Neural Networks* (págs. 303–308).
- Butcher, J. B., Verstraeten, D., Schrauwen, B., Day, C. R., e Haycock, P. W. (2013). Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks*, 38, 76–89.
- Cavalcante, C. C., Montalvão, J. R., Dorizzi, B., e Mota, J. C. M. (2000). A neural predictor for blind equalization in digital communications. Em *Adaptive Systems for Signal Processing, Communications and Control (AS-SPCC' 00)* (págs. 347–351).
- Chen, S., Mulgrew, B., e McLaughlin, S. (1993). Adaptive Bayesian equalizer with decision feedback. *IEEE Transactions on Signal Processing*, 41, 2918–2927.
- Cichocki, A., e Amari, S. (2002). *Adaptive blind signal and image processing: Learning algorithms and applications*. John Wiley & Sons.
- Copeland, B. J. (2004). *The essential Turing*. Oxford University Press.
- Copeland, B. J., e Proudfoot, D. (1996). On Alan Turing's anticipation of connectionism. *Synthese*, 108, 361–377.
- Copeland, B. J., e Proudfoot, D. (1999). Alan Turing's forgotten ideas in computer science.

Scientific American, 280(4), 77–81.

- Cover, T. M., e Thomas, J. A. (2006). *Elements of information theory* (2ª ed.). Wiley-Interscience.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals and Systems*, 2, 303–314.
- de Castro, L. N. (1998). *Análise e síntese de estratégias de aprendizado para redes neurais artificiais*. Dissertação de mestrado, Faculdade de Engenharia Elétrica e Computação - Universidade Estadual de Campinas.
- de Castro, L. N. (2006). *Fundamentals of natural computing: basic concepts, algorithms and applications*. Chapman & Hall/CRC.
- Debye, H. W. J., e Van Riel, P. (1990). L_p -norm deconvolution. *Geophysical Prospecting*, 38, 381–403.
- Devert, A., Bredeche, N., e Schoenauer, M. (2008). Unsupervised learning of echo state networks: a case study in artificial embryogeny. Em *Proceedings of the 8th International Conference on Artificial Evolution (EA 2007)* (Vol. 4926, págs. 278–290).
- dos Santos, E. P., e Von Zuben, F. J. (2000). Efficient second-order learning algorithms for discrete-time recurrent neural networks. Em *Recurrent neural networks: design and applications* (págs. 47–75). CRC Press.
- Duda, R. O., Hart, P. E., e Stork, D. G. (2001). *Pattern classification* (2ª ed.). John Wiley & Sons.
- Ercegovac, M. D., Lang, T., e Moreno, J. H. (1998). *Introduction to digital systems*. Wiley.
- Erdogmus, D., e Principe, J. C. (2002a). An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems. *IEEE Transactions on Signal Processing*, 50(7), 1780–1786.
- Erdogmus, D., e Principe, J. C. (2002b). Generalized information potential for adaptive systems training. *IEEE Transactions on Neural Networks*, 13(5), 1035–1044.
- Erdogmus, D., e Principe, J. C. (2006). From linear adaptive filtering to nonlinear information

- processing - the design and analysis of information processing systems. *Signal Processing Magazine, IEEE*, 23, 14–33.
- Farmer, J. D., e Sidorowich, J. J. (1987). Predicting chaotic time series. *Physical Review Letters*, 59, 845-848.
- Ferrari, R. (2005). *Equalização de canais de comunicação digital baseada em filtros fuzzy*. Dissertação de mestrado, Faculdade de Engenharia Elétrica e Computação - Universidade Estadual de Campinas.
- Ferrari, R., Panazio, C. M., Attux, R. R. F., Cavalcante, C. C., de Castro, L. N., Von Zuben, F. J., et al. (2003). Unsupervised channel equalization using fuzzy prediction-error filters. Em *Neural Networks for Signal Processing, 2003. NNSP'03. 2003 IEEE 13th Workshop on* (págs. 869–878).
- Ferrari, R., Suyama, R., Lopes, R. R., Attux, R. R. F., e Romano, J. M. T. (2008). An optimal MMSE fuzzy predictor for SISO and MIMO blind equalization. Em (págs. 86–91). First IAPR Workshop on Cognitive Information Processing.
- Forney, G. D. (1972). Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory*, 18, 363–378.
- Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3), 268–278.
- Fritzke, B. (1995). A growing neural gas network learns topologies. Em *Advances in neural information processing systems 7* (págs. 625–632). MIT Press.
- Funahashi, K.-I., e Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6, 801-806.
- Girosi, F., Jones, M., e Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 7(2), 219–269.
- Godard, D. N. (1980). Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Transactions on Communications*, 28(11), 1867–1875.

- Gonin, R., e Money, A. H. (1989). *Nonlinear L_p -norm estimation*. CRC Press.
- Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B*, 46(2), 149–192.
- Gutch, H. W., Gruber, P., Yeredor, A., e Theis, F. J. (2012). ICA over finite fields - separability and algorithms. *Signal Processing*, 92(8), 1796–1808.
- Hastie, T., Tibshirani, R., e Friedman, J. H. (2001). *The elements of statistical learning*. Springer.
- Haykin, S. (1996). *Adaptive filter theory* (3^a ed.). NJ: Prentice Hall.
- Haykin, S. (1998). *Neural networks: A comprehensive foundation* (2^a ed.). Prentice Hall.
- Haykin, S. (Ed.). (2000). *Unsupervised adaptive filtering, vol. I: blind source separation*. John Wiley & Sons.
- Hebb, D. O. (1949). *The organization of behavior*. Wiley & Sons.
- Hérault, J., Jutten, C., e Ans, B. (1985). Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. Em *Actes du X-ème Colloque GRETSI* (págs. 1017–1022).
- Hodges, A. (1992). *Alan Turing: The Enigma*. Vintage.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational properties. *Proceedings of the National Academy of Sciences of the USA*, 79(8), 2554-2558.
- Hornik, K., Stinchcombe, M., e White, H. (1989). Multilayer feedforward neural networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- Huang, G.-B., Chen, L., e Siew, C.-K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4), 879–892.

- Huang, G.-B., Zhu, Q.-Y., e Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004)* (págs. 985–990).
- Huang, G.-B., Zhu, Q.-Y., e Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70, 489–501.
- Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, 2, 94 – 128.
- Hyvärinen, A., Karhunen, J., e Oja, E. (2001). *Independent component analysis*. John Wiley & Sons.
- Inouye, Y., e Liu, R. (2002). A system-theoretic foundation for blind equalization of an fir mimo channel system. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(4), 425–436.
- Jaeger, H. (2001). *The echo state approach to analyzing and training recurrent neural networks* (Relat. Técnico N° 148). German National Research Center for Information Technology.
- Jaeger, H. (2002a). *Short term memory in echo state networks* (Relat. Técnico N° 152). Bremen: German National Research Center for Information Technology.
- Jaeger, H. (2002b). *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach* (Relat. Técnico N° 159). Bremen: German National Research Center for Information Technology.
- Jaeger, H. (2007). *Discovering multiscale dynamical features with hierarchical echo state networks* (Relat. Técnico N° 9). Jacobs University Bremen.
- Jiang, F., Berry, H., e Schoenauer, M. (2008a). Supervised and evolutionary learning of echo state networks. Em *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN 2008)* (Vol. 5199, págs. 215-224).
- Jiang, F., Berry, H., e Schoenauer, M. (2008b). Unsupervised learning of echo state networks: balancing the double pole. Em *Proceedings of the 10th Genetic and Evolutionary*

- Computation Conference (GECCO 2008)* (págs. 869-870).
- Johnson, R. A., e Wichern, D. W. (2007). *Applied multivariate statistical analysis* (6ª ed.). Prentice Hall.
- Jolliffe, I. T. (1986). *Principal component analysis*. Springer-Verlag.
- Kantz, H., e Schreiber, T. (2004). *Nonlinear time series analysis* (2ª ed.). Cambridge University Press.
- Kendall, M. (1975). *Multivariate analysis*. Charles Griffin & Company.
- Kilian, J., e Siegelmann, H. (1996). The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128, 48-56.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1), 59-69.
- Kohonen, T. (2000). *Self-organizing maps* (3ª ed.). Springer.
- Kuznetsov, Y., Kuznetsov, L., e Marsden, J. (1998). *Elements of applied bifurcation theory* (2ª ed.). Springer-Verlag.
- Lawson, C. L. (1961). *Contribution to the theory of linear least maximum approximations*. Tese de doutorado, University of California.
- Lazar, A., Pipa, G., e Triesch, J. (2009). SORN: a self-organizing recurrent neural network. *Frontiers in Computational Neuroscience*, 3(23).
- Leon-Garcia, A. (2008). *Probability, statistics and random processes for electrical engineering* (3ª ed.). Prentice Hall.
- Liu, W., Pohkarel, P., e Principe, J. C. (2007). Correntropy: properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11), 5286-5298.
- Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2), 130-141.
- Luenberger, D. G. (2003). *Linear and nonlinear programming* (2ª ed.). Springer.
- Lukoševičius, M. (2010). *On self-organizing reservoirs and their hierarchies* (Relat. Técnico

- Nº 25). Jacobs University Bremen.
- Lukosevicius, M., e Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3, 127–149.
- Luna, I., e Ballini, R. (2011). Top-down strategies based on adaptive fuzzy rule-based systems for daily time series forecasting. *International Journal of Forecasting*, 27(3), 1-17.
- Maass, W. (2007). Liquid computing. Em B. L. S. B. Cooper e A. Sorbi (Eds.), *Lecture Notes in Computer Science* (Vol. 4497, págs. 507–516).
- Maass, W., Natschläger, T., e Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531-2560.
- Maass, W., Natschläger, T., e Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal Physiology - Paris*, 98(4–6), 315–330.
- Mandic, D. P., e Chambers, J. A. (2001). *Recurrent neural networks for prediction*. Wiley-Interscience.
- Martinetz, T., e Schulten, K. (1991). A “neural gas” network learns topologies. Em *Artificial neural networks, vol. I* (págs. 397–402). Elsevier.
- Mathews, V. J. (1991). Adaptive polynomial filters. *IEEE Signal Processing Magazine*, 8, 10–26.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261, 459–467.
- McCulloch, W., e Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4), 115–133.
- Minsky, M. L., e Papert, S. A. (1969). *Perceptrons*. MIT Press.
- Montalvão, J., Dorizzi, B., e Mota, J. C. M. (1999). Some theoretical limits of efficiency of linear and nonlinear equalizers. *Journal of Communications and Information Systems*, 14, 85–92.
- Ozturk, M. C., Xu, D., e Principe, J. C. (2007). Analysis and design of echo state networks.

- Neural Computation*, 19, 111-138.
- Papoulis, A. (1991). *Probability, random variables, and stochastic processes*. McGraw-Hill International Editions.
- Park, J., e Sandberg, I. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2), 246-257.
- Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1065–1076.
- Poggio, T., e Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), 1481–1497.
- Principe, J. C. (2010). *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer.
- Proakis, J. G., e Salehi, M. (2007). *Digital communications* (5^a ed.). McGraw-Hill Science.
- Rice, J. R., e Usow, K. H. (1968). The Lawson algorithm and extensions. *Math. Comp.*, 22, 118–127.
- Rice, J. R., e White, J. S. (1964). Norms for smoothing and estimation. *SIAM Review*, 6, 243–255.
- Rényi, A. (1961). On measures of information and entropy. Em *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960* (págs. 547–561).
- Romano, J. M. T., Attux, R. R. d. F., Cavalcante, C. C., e Suyama, R. (2011). *Unsupervised signal processing: channel equalization and source separation*. CRC Press.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
- Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27, 832–837.
- Rumelhart, D. E., Hinton, G. E., e Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536.
- Sacchi, R., Ozturk, M., Principe, J., Carneiro, A., e da Silva, I. (2007). Water inflow

- forecasting using the echo state network: a Brazilian case study. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2007)* (págs. 2403–2408).
- Scannell, J. W., Burns, G. A., Hilgetag, C. C., O’Neil, M. A., e Young, M. P. (1999). The connectional organization of the cortico-thalamic system of the cat. *Cereb Cortex*, *9*(3), 277–299.
- Schafer, A. M., e Zimmermann, H. G. (2007). Recurrent neural networks are universal approximators. *International Journal of Neural Systems*, *17*(5), 253–263.
- Schmidhuber, J., Wierstra, D., Gagliolo, M., e Gomez, F. J. (2007). Training recurrent networks by Evolino. *Neural Computation*, *19*(3), 757–779.
- Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J. J., e Stroobandt, D. (2008). Improving reservoirs using intrinsic plasticity. *Neurocomputing*, *71*, 1159–1171.
- Schwefel, H.-P. (1981). *Numerical optimization of computer models*. John Wiley & Sons.
- Shalvi, O., e Weinstein, E. (1990). New criteria for blind deconvolution of non-minimum phase systems (channels). *IEEE Transactions on Information Theory*, *36*(2), 312–321.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423.
- Shi, Z., e Han, M. (2007). Support vector echo-state machine for chaotic time-series prediction. *IEEE Transactions on Neural Networks*, *18*(2), 359–372.
- Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall.
- Siqueira, H. V., Boccatto, L., Attux, R., e Lyra Filho, C. (2011). Predição de séries de vazões com redes neurais de estados de eco. Em *X Congresso Brasileiro de Inteligência Computacional* (págs. 1–7).
- Siqueira, H. V., Boccatto, L., Attux, R., e Lyra Filho, C. (2012a). Echo state networks and extreme learning machines: a comparative study on seasonal streamflow series prediction. *Lecture Notes in Computer Science*, *7664*, 491–500.

- Siqueira, H. V., Boccato, L., Attux, R., e Lyra Filho, C. (2012b). Echo state networks for seasonal streamflow series forecasting. *Lecture Notes in Computer Science*, 7435, 226–236.
- Siqueira, H. V., Boccato, L., Attux, R., e Lyra Filho, C. (2012c). Echo state networks in seasonal streamflow series prediction. *Learning and Nonlinear Models*, 10(3), 181–191.
- Steil, J. (2004). Backpropagation-decorrelation: online recurrent learning with $O(N)$ complexity. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004)* (Vol. 1, págs. 843-848).
- Strogatz, S. H. (2000). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press.
- Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410, 268–276.
- Sutton, R., e Barto, A. (1998). *Reinforcement learning*. MIT Press.
- Suyama, R., Duarte, L. T., Ferrari, R., Rangel, L. E. P., Attux, R., Cavalcante, C. C., et al. (2007). A nonlinear prediction approach to the blind separation of convolutive mixtures. *EURASIP J. Appl. Signal Process.*, 2007, 84–84.
- Suykens, J., Vandewalle, J., e De Moor, B. (1996). *Artificial neural networks for modeling and control of non-linear systems*. Springer.
- Teuscher, C. (2001). *Turing’s connectionism: an investigation of neural networks architectures*. Springer.
- Teuscher, C., e Sanchez, E. (2001). A revival of Turing’s forgotten connectionist ideas: Exploring unorganized machines. Em R. M. French e J. J. Sourné (Eds.), *Connectionist Models of Learning, Development and Evolution, Proceedings of the Sixth Neural Computation and Psychology Workshop, NCPW6* (págs. 153–162). Springer-Verlag.
- Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B*, 58(1), 267–288.
- Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4, 1035–1038.

- Triesch, J. (2005). A gradient rule for the plasticity of a neuron's intrinsic excitability. Em *Artificial neural networks: Biological inspirations - ICANN 2005* (págs. 65–70). Springer.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, 230–265.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460.
- Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London, Series B*, 237, 37–72.
- Turing, A. M. (1968). Intelligent machinery. Em C. R. Evans e A. D. Robertson (Eds.), *Cybernetics: key papers*. Baltimore Md. and Manchester: University Park Press.
- Verstraeten, D., Dambre, J., Dutoit, X., e Schrauwen, B. (2010). Memory versus non-linearity in reservoirs. Em *Proceedings of the WCCI 2010 IEEE World Congress on Computational Intelligence* (págs. 2669–2676).
- Verstraeten, D., Schrauwen, B., D'Haene, M., e Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20(3), 391–403.
- Verstraeten, D., Schrauwen, B., e Stroobandt, D. (2007). Adapting reservoirs to get Gaussian distributions. Em *Proceedings of ESANN'2007 - European Symposium on Artificial Neural Networks* (págs. 495–500).
- Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2), 260–269.
- von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14, 85–100.
- Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Tese de doutorado, Harvard University.
- Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. Em *Proceedings of the IEEE* (Vol. 78, págs. 1550–1560).
- Widrow, B., e Stearns, S. (1985). *Adaptive signal processing*. Prentice Hall.

- Wiener, N. (1958). *Nonlinear problems in random theory*. MIT.
- Williams, R., e Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1, 270-280.
- Xu, D., Lan, J., e Príncipe, J. C. (2005). Direct adaptive control: an echo state network and genetic algorithm approach. Em *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2005)* (Vol. 3, págs. 1483–1486).
- Yamazaki, T., e Tanaka, S. (2007). The cerebellum as a liquid state machine. *Neural Networks*, 20(3), 290–297.
- Yildiz, I. B., Jaeger, H., e Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, 35, 1–9.
- Zabell, S. L. (1995). Alan Turing and the central limit theorem. *The American Mathematical Monthly*, 102(6), 483–494.