UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL (DCA)

REDE NEURAL PARA RECONHECIMENTO ADAPTATIVO DE FONEMAS RUIDOSOS

Candidato: *Martín Graciarena*

Orientador: *Prof. Dr. Márcio Luiz de Andrade Netto*

Tese submetida à Faculdade de Engenharia Elétrica da UNICAMP como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Campinas, 22 de Julho de 1998.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

G753r

Graciarena, Martín
Rede neural para reconhecimento adaptativo de fonemas ruidosos. / Martín Graciarena.--Campinas, SP: [s.n.], 1998.

Orientador: Márcio Luiz de Andrade Netto
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais (Computação). 2. Reconhecimento automático da voz. 3. Kalman, Filtragem de. I. Andrade Netto, Márcio Luiz de. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

RESUMO

No presente trabalho é proposta a incorporação de um mecanismo adaptativo, o filtro de Kalman, ao modelo tradicional de neurônio dando por resultado um modelo que chamamos ***Neurônio de Reconhecimento Adaptativo***, especificamente destinado ao reconhecimento ruidoso de padrões. O objetivo do modelo é a classificação da estimação do padrão limpo realizado pelo filtro de Kalman, a partir de suas observações ruidosas. Se estende naturalmente o modelo proposto a uma rede de neurônios que chamamos ***Rede Neural de Reconhecimento Adaptativo***. Estudaremos também desde o ponto de vista teórico suas propriedades e o aplicaremos à classificação de padrões ruidosos e de séries temporais no problema XOR.

As experiências foram feitas com o objetivo de estudar a robustez do mecanismo proposto frente ao problema de *desemparelhamento de condição*. Este pode resumir-se como os problemas que surgem com os sistemas de reconhecimento quando têm que reconhecer padrões em condições diferentes às presentes nos padrões de treinamento. Em todos os problemas de reconhecimento estudados, o treinamento das redes neurais é feito com exemplos não ruidosos.

A proposta que se apresentará está incluída dentro dos **classificadores robustos**. Isto é, propõe mecanismos para que o classificador seja robusto à distorção ruidosa dos padrões. Isto em contraposição à estratégia clássica de filtrar o ruído na etapa das características (chamadas características invariantes) e evitar que passem ao classificador. Mas o grande problema desta aproximação é que o *classificador não é robusto ao ruído*, portanto, em níveis de distorção onde as características não possam filtrar todo o ruído, existirá um erro no classificador.

Aplicaremos o modelo proposto para a classificação de fonemas ruidosos. Para tal fim a proposta é utilizar três diferentes arquiteturas. Estas diferem entre si na forma de extração do padrão de características do sinal de voz. A primera implica em análise através de um banco de filtros digitais onde os filtros de Kalman estimam os valores médios da energia de saída de cada filtro. A segunda implica no uso de um modelo de predição linear extraído pelo filtro de Kalman diretamente do sinal de voz, onde os padrões a serem reconhecidos são os coeficientes de predição linear. A terceira é uma melhora sobre a anterior, onde se classifica o ângulo dos coeficientes de predição linear. Para esta última arquitetura apresentamos o fundamento teórico de onde foi extraída. Junto com a proposta de dois índices de reconhecimento especificamente destinados a aplicações de voz, as arquiteturas propostas são primeiro comparadas no reconhecimento ruidoso de vogais espanholas afetadas por ruído branco gaussiano em diversas relações sinal - ruído e também no reconhecimento ruidoso de palavras, junto com os modelos Hidden Markov Models (HMM).

A partir dos resultados encontrados no reconhecimento de palavras com a melhor arquitetura, foi proposta uma ***Rede Invariante***. Esta tem a propriedade de ser robusta à compressão e dilatação dos padrões, que é a alteração sofrida pelos coeficientes de predição linear na presença do ruído. Esta é a rede que, juntamente com os coeficientes de predição linear foi a que melhor desempenho teve no reconhecimento ruidoso de palavras.

ÍNDICE

AGRADECIMENTOS

A minha Andréa por ter me acompanhado e apoiado na aventura brasileira. Aos meus pais e irmãos pelo incentivo e carinho.

Principalmente ao Prof. Dr. Márcio Luiz de Andrade Netto, por sua orientação, incentivo e seu contínuo apoio.

Ao Prof. Ing. Luis F. Rocha, por seu estímulo no andamento deste trabalho.

Entre todos os amigos no Brasil, meu agradecimento especial a Andréa e Reginaldo, Perla e Nina. Também a Adrián, Jussara, Henrique, Caio e Ivân. Ao Márcio Leandro pela ajuda por e-mail. Mas principalmente ao Fernando por todo o "suporte estratégico" em Campinas e por sua amizade e bom humor.

Ao Ing. Marcelo Lehmann e aos companheiros do Laboratorio Abierto (LABI) por ter me apoiado na realização dos cursos de pós-graduação na UNICAMP.

Ao grupo de pesquisa em reconhecimento de voz do Instituto de Engenharia Biomédica, Claudio, Pedro e Guillermo, por todas as frutíferas discussões. Também para o resto do pessoal do Instituto, Juan Carlos, Jorge, Patricia e María del Carmen.

A Ilú por ter me ajudado muito na tradução desta tese ao português.

A Universidade Estadual de Campinas, pela oportunidade.

A Universidad de Buenos Aires e a Academia Nacional de Ciencias de Buenos Aires, pelo apoio financiero.

A Andréa, minha esposa maravilhosa.

CAPÍTULO 1

INTRODUÇÃO

1.1- RECONHECIMENTO AUTOMÁTICO DE VOZ

O reconhecimento de voz é uma das aplicações mais importantes do reconhecimento de padrões. A relevância deriva-se de suas possíveis aplicações no benefício da comunidade, tanto em ajuda a deficientes como na automação de tarefas rotineiras realizadas por operários humanos. Algumas de tais aplicações permitem uma comunicação mais simples e direta com as máquinas através da emissão de comandos de voz.

Um grande problema para a efetiva aplicação desta tecnologia é a existência de ruídos nos ambientes onde estes sistemas operam, como por exemplo ruído de escritórios para um reconhecedor de ditados, ruído de jatos em aviões comandados por voz, ou mesmo com ruídos de carros, etc. Fora destes casos específicos, as aplicações típicas do reconhecimento de voz implicam a presença de ruídos de distribuição estatística desconhecida e variável com o tempo.

As amostras de voz utilizadas no treinamento dos sistemas de reconhecimento de voz, são recolhidas, quase sem exceção, em ambientes onde as perturbações estão ausentes. Porém na prática, na etapa de reconhecimento, não se podem garantir estas condições controladas e os padrões podem estar afetados por sinais aleatórios. A falta de robustez frente a esta situação provoca uma importante perda de desempenho de reconhecimento. Este fenômeno é denominado por alguns autores [1] como "Condition Mismatch", ou Desemparelhamento de Condição.

1.2- REDES NEURAIS ARTIFICIAIS

Procurando simular a extraordinária capacidade humana de reconhecimento e memória, ainda que em condições extraordinariamente desfavoráveis, foram criados modelos matemáticos simples do cerébro humano chamados redes neurais artificiais. Estes modelos têm a capacidade de aprendizagem automática e generalização, diretamente a partir de amostras de treinamento. Isto implica que são capazes de descobrir regularidades e relações a partir de exemplos em forma automática e utilizá-las para classificar. Tudo isto permite extrair das redes neurais artificiais um excelente desempenho de reconhecimento.

Uma das principais aplicações das redes neurais artificiais é o reconhecimento de voz. Existem diversas estruturas propostas para esta aplicação. Mas elas estão sujeitas ao problema do desemparelhamento de condição, como todo reconhecedor que é treinado com exemplos limpos e que reconhece em ambientes ruidosos.

1.3- OBJETIVO DA TESE

O objetivo é o desenvolvimento de mecanismos para que os sistemas de reconhecimento, não somente mantenham seu desempenho sob condições emparelhadas mas que, além disso, não sofram uma dramática degradação quando o fenômeno de desemparelhamento de condição ocorre. Neste sentido se propõe a incorporação de mecanismos adaptativos às redes neurais para superar suas limitações como reconhecedores ruidosos.

1.4- ORGANIZAÇÃO DA DISSERTAÇÃO

No **capítulo 2**, apresentaremos uma breve introdução à teoria das redes neurais, principalmente do tipo "*Multilayer Perceptron*". Estudaremos especialmente suas características como ferramenta computacional aplicada ao reconhecimento de padrões. Começaremos com uma breve história da evolução da teoria das redes neurais artificiais, seguiremos com uma apresentação sobre os modelos computacionais básicos e avançados com ênfase no seu treinamento. Finalmente exporemos os fundamentos matemáticos de sua capacidade de aproximação. É importante deixar claro que ao nos referir às redes neurais estamos nos referindo em realidade a "*redes neurais artificiais*", isto é a modelos matemáticos simples do cérebro humano.

A teoria do filtro de Kalman é apresentada no **capítulo 3**. Primeiro formalizaremos a teoria das inovações, que é uma ferramenta básica para a compreensão do filtro de Kalman. Depois apresentaremos brevemente a teoria de modelagem por espaço de estados e depois a teoria do filtro de Kalman. Provaremos suas condições de otimalidade e introduziremos, posteriormente, uma modificação do filtro de Kalman para casos onde o estado a estimar mude com o tempo. Finalmente, apresentaremos exemplos de aplicação do filtro de Kalman normal e de memória finita.

O **capítulo 4** é uma introdução ao reconhecimento de padrões e voz, redes neurais, reconhecimento de voz e reconhecimento de voz ruidosa. Na primeira seção apresentamos a teoria de reconhecimento de padrões com ênfase no critério de Bayes. Depois estudaremos as características especiais do reconhecimento de voz como um problema de reconhecimento de padrões. A seção seguinte apresenta diversas estruturas propostas da aplicação de redes neurais ao reconhecimento de voz. Finalmente, apresentaremos os

conceitos envolvidos no problema de reconhecimento de voz ruidosa e as estratégias para sua solução.

No **capítulo 5** apresentamos o modelo proposto que combina redes neurais e os filtros de Kalman para o reconhecimento de padrões ruidosos, que será chamado neurônio de reconhecimento adaptativo. Estudaremos suas propriedades e demonstraremos as respectivas condições de otimalidade. Estenderemos depois a proposta uma rede neural com neurônios de reconhecimento adaptativo, que chamaremos rede neural de reconhecimento adaptativo. Estudaremos suas propriedades e aprofundaremos nos aspectos de implementação levando em conta seu custo computacional. Finalmente, aplicaremos esta rede na classificação de séries de padrões para o problema XOR

O tema do **capítulo 6** é a aplicação do modelo de rede proposto ao problema do reconhecimento ruidoso de fonemas. São propostas três arquiteturas diferentes, tomando em consideração as características particulares do problema. Estas diferem na forma de extração do padrão de características do sinal de voz. A primera implica o uso de uma análise por um banco de filtros digitais onde os filtros de Kalman estimam os valores médios da energia de saída de cada filtro. A segunda implica o uso de um modelo de predição linear extraído pelo filtro de Kalman diretamente do sinal de voz, onde os padrões a serem reconhecidos são os coeficientes de predição linear. A terceira é uma melhoria sobre a anterior, onde se classifica o ângulo dos coeficientes de predição linear. Junto com a proposta de dois índices de reconhecimento especificamente destinados a aplicações de voz, as arquiteturas propostas são comparadas no reconhecimento ruidoso de vogais espanholas afetadas de ruído branco gaussiano a diversas relações sinal - ruído.

CAPÍTULO 2

INTRODUÇÃO ÀS REDES NEURAIS.

2.1- INTRODUÇÃO

A partir do surgimento dos computadores digitais, com sua grande capacidade de cálculo, tem aumentado o interesse no desenvolvimento de máquinas inteligentes. O objetivo é obter sistemas que sejam capazes de desempenhar tarefas que os seres humanos realizam em forma muito eficiente, como o reconhecimento de padrões, o controle de movimentos, etc. As tentativas de reproduzir esta capacidade humana tem sido dificultosa por que os sistemas artificiais tradicionalmente utilizados não possuem as propriedades imprescindíveis para realizar estas tarefas. Entre estas propriedades pode-se mencionar a capacidade de associação, o conhecimento distribuído, a tolerância a falhas e o processamento paralelo distribuído.

A vantagem potencial de contar com ferramentas deste tipo é, por um lado proporcionar uma melhor compreensão dos mecanismos de processamento da informação do cérebro humano, e por outro lado poder automatizar a resolução de problemas que se caracterizam por um elevado grau de incerteza e variabilidade em suas características. Entre estes problemas podemos mencionar o reconhecimento de voz e imagens, o processamento da linguagem, o controle de sistemas não perfeitamente determinados e outros. Nesta procura surge naturalmente o interesse por obter um modelo de uma "máquina" que realize facilmente as tarefas mencionadas: o cérebro humano.

Neste capítulo apresentaremos uma breve introdução à teoria das redes neurais. Estudaremos especialmente suas características como ferramenta computacional aplicada ao reconhecimento de padrões. Começaremos com uma breve história da evolução da teoria das redes neurais artificiais, seguiremos com uma apresentação sobre os modelos computacionais básicos e avançados e finalmente exporemos os fundamentos matemáticos de sua capacidade de aproximação. É importante clarificar que ao nos referir às redes neurais estamos nos referindo em realidade a "redes neurais artificiais", isto é a modelos matemáticos simples do cérebro humano.

2.2- HISTÓRIA DAS REDES NEURAIS

O interesse pelo desenvolvimento da teoria das redes neurais é interdisciplinar, devido a que os primeiros desenvolvimentos em quanto à modelagem do processamento

realizado pelos neurônios foram realizados por neurofisiologistas. Por outro lado os engenheiros e os cientistas computacionais são os que utilizam estes resultados para o desenvolvimento de máquinas que possam superar a capacidade dos computadores atuais, caracterizados pelo processamento serial de informação e sua baixa tolerância a falhas.

O desenvolvimento da teoria das redes neurais artificiais teve uma evolução muito particular. Acompanhando de perto as primeiras investigações por neurofisiologistas acerca da modelagem do cérebro, os engenheiros começaram a oferecer um fundamento matemático para estes modelos. Desta forma gerou-se uma notável expectativa no potencial destes modelos. A publicação por reconhecidos investigadores da época de uma limitação na capacidade de representação dos modelos iniciais criou um vazio na investigação por muitos anos. Com o aperfeiçoamento dos modelos, a superação da limitação nomeada e o surgimento de novos mecanismos computacionais de treinamento, evidenciou-se um grande crescimento do interesse científico no desenvolvimento dos fundamentos matemáticos e de sua extensão a modelos cada vez mas complexos.

A ideia original de "neurônios" como elementos constituintes do cérebro humano foi introduzida nos trabalhos pioneiros de Ramón y Cajál em 1911 [44].

Em 1943 os investigadores da área da biologia McCullogh & Pitts (neuroanatomista e matemático respectivamente) apresentaram um modelo de neurônio como uma unidade de processamento binário e provaram que estas unidades são capazes de executar muitas das operações que podem ser descritas em termos lógicos. Cabe destacar que esta é a primeira vez que se apresenta um modelo matemático de um dispositivo artificial inspirado na operação do neurônio humano. Este modelo, apesar de ser muito simples trouxe uma grande contribuição sobre a construção dos primeiros computadores digitais, particularmente sobre o de Von Neumann.

Em 1949, Hebb apresentou no seu livro "*The Organization of Behavior*" (A organização do comportamento) pela primeira vez uma regra de aprendizagem fisiológica ao respeito da alteração da sinapse no cérebro em resposta à experiência. Em particular ele sugeriu, seguindo uma sugestão de Ramón y Cajál, que as conexões entre as células que são ativadas ao mesmo tempo tendem a fortalecer, enquanto que as outras conexões tendem a enfraquecer. Esta hipótese passou a influir decisivamente na evolução da teoria de aprendizagem das redes neurais artificiais.

Em 1958, Rosenblatt [9] introduziu uma nova abordagem ao problema de reconhecimento de padrões com o desenvolvimento do "perceptron", cuja representação é

apresentada na figura 2.1. Sua contribuição fundamental foi o "Teorema de Convergência do Perceptron" onde prova a convergência de algoritmo para o ajuste dos pesos do *"perceptron"* quando os padrões são linearmente separáveis. En torno do mesmo período, B. Widrow e sus colaboradores desenvolveram o Adaline (Adaptative Linear Element) e sua correspondente regra de aprendizagem, o algoritmo LMS ou Least Mean Square, para seu uso na área de processamento de sinais. O Adaline apresenta uma estrutura similar ao *"perceptron"* diferindo na regra de atualização das conexões sinápticas.

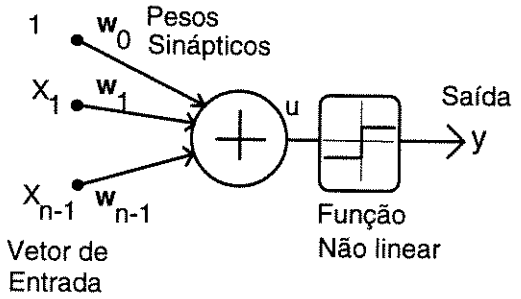


FIGURA 2.1 - "perceptron"

Os desenvolvimentos de Rosenblatt e Widrow criaram grande expectativa a respeito das potencialidades desta linha de investigação, devido ao fato de que a maior parte das investigações nesse tempo eram de natureza heurística. Porém faltavam na época resultados teóricos que justificassem a continuidade do interesse científico pela área, o que trouxe uma redução na produção de novas idéias. Possivelmente o principal problema, de maior significado histórico, foi a exagerada expectativa criada pelos próprios investigadores da área, que não sendo acompanhada por resultados a altura acelerou a diminuição de financiamentos para a investigação. Mas foi após a publicação do livro "*Perceptrons*" em 1969 por Minsky e Papert, que as investigações na área das redes neurais sofreram um atraso significativo. Neste livro, conceitos de matemática moderna como topologias e teoria de grupos são aplicados com o objetivo de analisar as capacidades adaptativas e computacionais dos modelos neurais apresentados. Os autores demonstraram que o "perceptron", apesar de ser capaz de executar as operações booleanas AND e OR, não era capaz de executar outras operações elementares como o XOR (OR-exclusivo). Além disso, estes autores não acreditavam que uma arquitetura multicamada adequada, juntamente com um algoritmo de ajuste de pesos, pudessem ser desenvolvidos com o objetivo de superar esta limitação. Após a publicação destes resultados a maior parte dos investigadores da área de redes neurais passou a buscar alternativas dentro do campo da engenharia e principalmente da lógica matemática.

Apesar deste êxodo generalizado, um pequeno número de investigadores continuou trabalhando com redes neurais nos anos 70, principalmente na área dos mapas

auto-organizados e de aprendizagem competitivo. Os nomes de T. Kohonen, S. Grossberg, B. Widrow, J. Anderson e K. Fukushima estão associados a este período.

Entre os fatores que ajudaram ao resurgimento do interesse pela pesquisa em redes neurais foram os importantes resultados obtidos através da aplicação de conceitos conexionistas ao problema de modelar os materiais paramagnéticos (vidros de spin), realizada pelo físico J. J. Hopfield, publicada em 1982 [10], [11]. Ele considerou um conjunto de neurônios dispostos de forma que suas saídas fossem realimentadas para as entradas. Este modelo, o primeiro a introduzir dinâmica em redes neurais, apresentado na figura 2.2, ajuda a demonstrar como uma rede composta por elementos computacionais simples pode dar por resultado interessantes comportamentos coletivos. A rede neural de Hopfield pode ser considerada como um sistema dinâmico com um número finito de estados de equilíbrio, de forma que o sistema invariavelmente evoluirá para um desses estados a partir de uma condição inicial. É também natural que a localização destes estados de equilíbrio possa ser controlada pela intensidade das conexões (pesos) da rede neural.

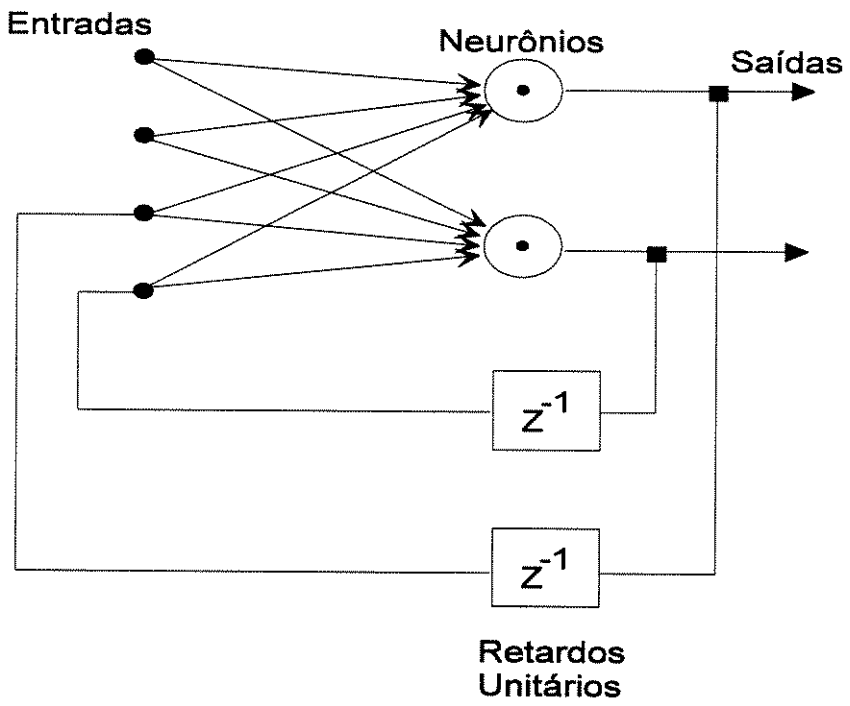


FIGURA 2.2 - Rede de Hopfield

A conclusão importante apresentada por Hopfield é que tais estados de equilíbrio podem ser utilizados como dispositivos de memória, numa forma distinta daquela utilizada pelos computadores convencionais, onde o acesso à informação armazenada acontece através de um endereço; o acesso ao conteúdo de uma memória da rede de Hopfield é possível permitindo que a rede evolua com o tempo para um do seus estados de equilíbrio.

Tais modelos de memória são denominados "*memórias acessíveis por seu conteúdo*". O trabalho de Hopfield com este tipo de rede simétrica recorrente atraiu principalmente matemáticos e engenheiros à investigação nesta área, e as redes de Hopfield foram estudadas como memórias distribuídas e também utilizadas como ferramentas na solução de problemas de otimização restrita.

O fato que efetivamente colocou a área das redes neurais como uma das prioritárias na obtenção de recursos foi o desenvolvimento de um método para ajustar os parâmetros das redes não-recorrentes de múltiplas camadas ou redes multicamada em avanço (feed forward multilayered "perceptrons"). Este método, baseado num algoritmo denominado "backpropagation", tornou-se largamente conhecido logo da publicação, em 1986, do livro "*Parallel Distributed Processing*", editado por J. L. McCleland e D. E. Rumelhart [12], fazendo que os investigadores de diferentes áreas passassem a visualizar interessantes aplicações para redes neurais artificiais. A importância deste método justifica um tratamento mais profundo que será desenvolvido nas próximas seções.

A consolidação das redes neurais como um nova linha de investigação é evidenciada pela organização anual de numerosas conferências internacionais específicas como as ICNN (International Conference on Neural Networks), WCNN (World Conference on Neural Networks), e outras, bem como pela fundação de revistas cientificas dedicadas exclusivamente ao tratamento deste tema, como as IEEE Transactions on Neural Networks, as revistas Neural Networks, Neural Computation e outras.

2.3- ESPECIFICAÇÕES DAS REDES NEURAIS

2.3.1- Antecedente biológico

A partir de conhecer estas limitações os investigadores se voltaram à tarefa de criar modelos muito simplificados do funcionamento do cérebro humano, começando pelo que se acredita é seu fundamento básico, o neurônio.

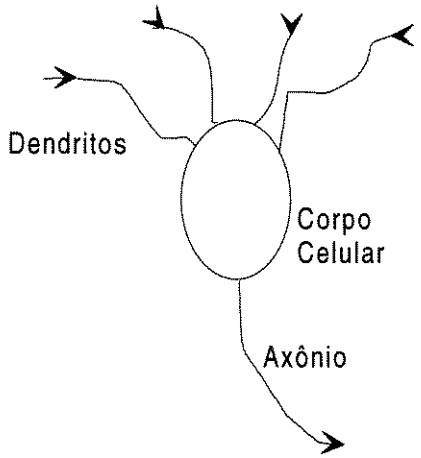


FIGURA 2.3 - Neurônio biológico humano

Apresenta-se na figura 2.3 um diagrama simplificado do neurônio humano. Neste podem distinguir-se as seguintes partes fundamentais: os dendritos que se comunicam com os outros neurônios através da sinapse; o corpo celular que é onde chegam os dendritos; o axônio que é a prolongação maior do neurônio e é quem transporta os impulsos nervosos de saída através de outros neurônios.

2.3.2- Descrição do neurônio básico - variações

O modelo básico do neurônio apresenta-se na figura 2.4. É caracterizado como uma unidade de processamento de informação fundamental para a estrutura do sistema neural.

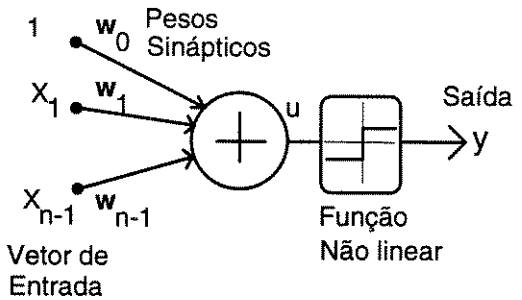


FIGURA 2.4 - Modelo neurônio básico

Podemos identificar três partes elementares:

1.- Um conjunto de sinapsis que é caracterizada por um peso ou força de conexão. A entrada está multiplicada por este peso, de forma que seria reforçada ou atenuada. Mais adiante veremos uma interpretação geométrica destes pesos.

2.- Um somador, ou combinador linear, que soma cada uma das entradas ponderadas pelos respectivos pesos.

3.- Uma função de ativação para limitar os valores de saída do neurônio. Tipicamente se utilizam faixas de [0,1] por exemplo quando se deseja que as saídas representem valores probabilísticos ou de [-1,1].

4.- Um limiar, que permite a translação da reta de separação do espaço de estados fora do origem. Em inglês denomina-se "threshold" ou "bias".

A equação que descreve a saída linear posterior ao somador, também chamada estado de ativação, apresenta-se na primeira parte da equação 2.1 e a equação de saída apresenta-se na segunda parte da mesma equação:

$$u_i = \sum_{j=0}^{p} w_{ij} \cdot X_j ;\ y_i = \varphi(u_i) \tag{2.1}$$

onde na equação 2.1 apresentamos como X_0 ao valor fixo 1 e w_{i0} ao peso correspondente ao limiar.

É importante interpretar a operação matemática realizada pelo conjunto sinapse, somador e limiar. Observados isoladamente os pesos podem ser interpretado como um mecanismo para o reforço ou a atenuação da entrada correspondente. Também o conjunto pesos e somador pode ser interpretado como uma soma ponderada das entradas. Esta operação pode ser apresentada desde um ponto de visto geométrico como um ***produto escalar*** entre o vetor de entrada e o vetor representado pelos pesos. Este último vetor é ortogonal ao hiperplano separador que define o limiar permitindo que este hiperplano não passe pelo origem. Uma saída positiva implica que a projeção do vetor de entrada sobre o vetor de pesos forma um ângulo com este menor de 90°, sendo negativo no caso contrário; ou o vetor de entrada pertence ao subespaço positivo ou ao subespaço negativo. Na figura 2.5 apresentamos um exemplo.

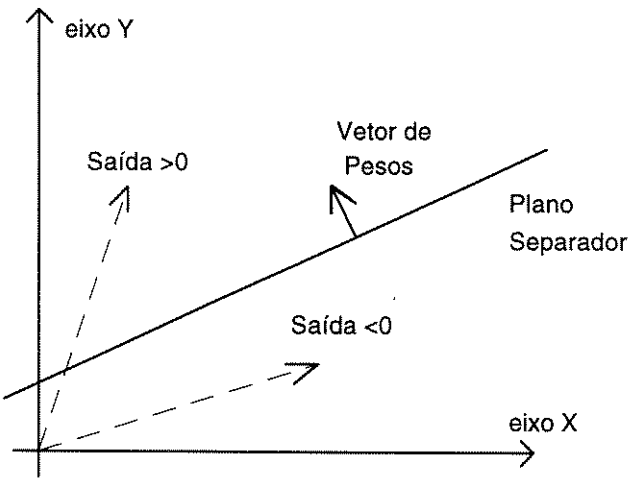


FIGURA 2.5 - Exemplo vetor e hiperplano

Em termos vetoriais podemos representar o vetor de entrada como $X = [1\ x_1\ x_2\ ...\ x_n]$ e o vetor de pesos $W = [w_0\ w_1\ ...\ w_n]$, então a operação implementada pelo neurônio da figura 2.4 será (onde $(\bullet)^t$ denota transposição):

$$y=\varphi(W^t \cdot X) \tag{2.2}$$

Como função de ativação $\varphi(\bullet)$, as mais usadas são a função sinal, a função linear por partes, a função logística e a função sigmóide. Em geral as funções são monotonas crescentes e apresentam algum tipo de descontinuidadade ou não linearidade e são saturadas. O tipo de função de ativação a ser utilizado dependerá do tipo de estado de ativação, seja este discreto ou contínuo. A seguir são apresentadas algumas funções de ativação freqüentemente utilizadas na literatura:

Função sinal:

É a função mais utilizada para casos discretos, sendo definida da forma:

$$y(u)=\begin{cases}0 & \text{se } u<0\\ [0,1] & \text{se } u=0\\ 1 & \text{se } u>0\end{cases} \tag{2.3}$$

Com relação ao caso u=0 é conveniente arbitrar o valor de y(u) a um dos dois extremos do intervalo [0,1]. Então a saída da função sinal será binária.

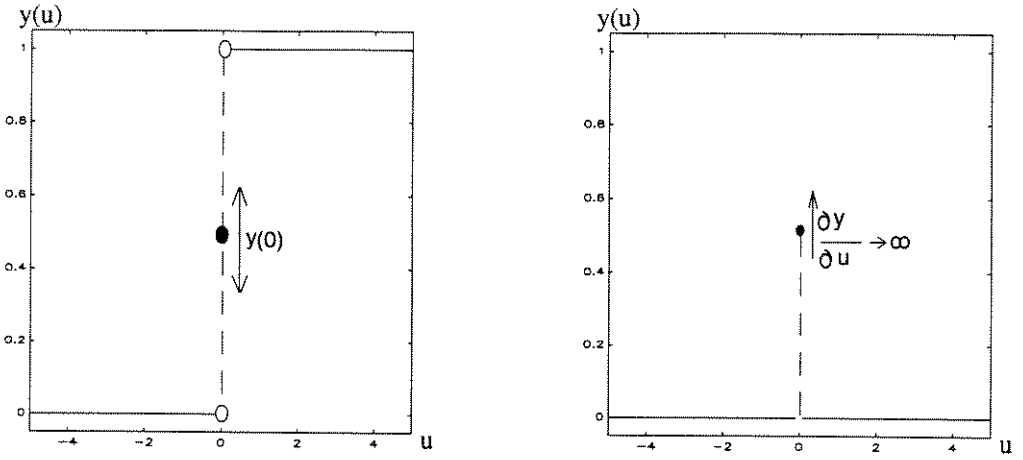


FIGURA 2.6 - Função sinal e sua derivada

Observamos também que a função sinal é linear por partes e apresenta uma descontinuidade na origem, onde sua derivada assume um valor infinito.

Função linear por partes:

Apesar do intenso uso da função sinal nos inícios do desenvolvimento da teoria de redes neurais, a descontinuidade que apresenta na origem é um fator indesejável para os algoritmos que usam a derivada da função na atualização dos pesos, já que esta pode tomar valores infinitos. Para limitar a derivada a valores finitos usa-se a função linear por partes, que apresenta a forma: (onde p é uma constante positiva).

$$y(u) = \begin{cases} 0 & \text{se } p < 0 \\ pu & \text{se } 0 < p < 1 \\ 1 & \text{se } p > 1 \end{cases} \tag{2.4}$$

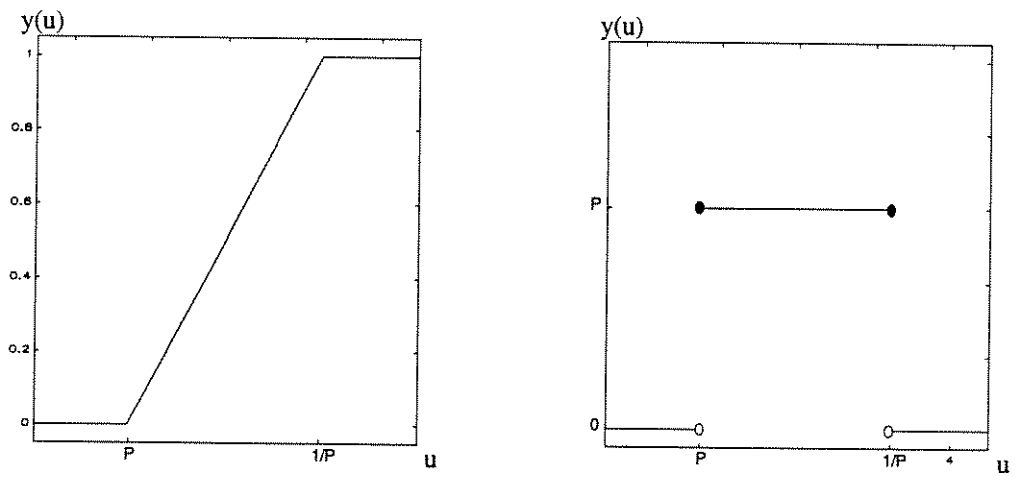


FIGURA 2.7 - Função linear por partes e sua derivada

Observe-se que se o valor de p tende a infinito, a função linear por partes transforma-se na função sinal.

Função logística:

É uma das duas funções de ativação mais usadas no caso contínuo, sendo definida da forma: (onde p é uma constante positiva).

$$y(u) = \frac{e^{pu}}{e^{pu}+1} = \frac{1}{1+e^{-pu}} \tag{2.5}$$

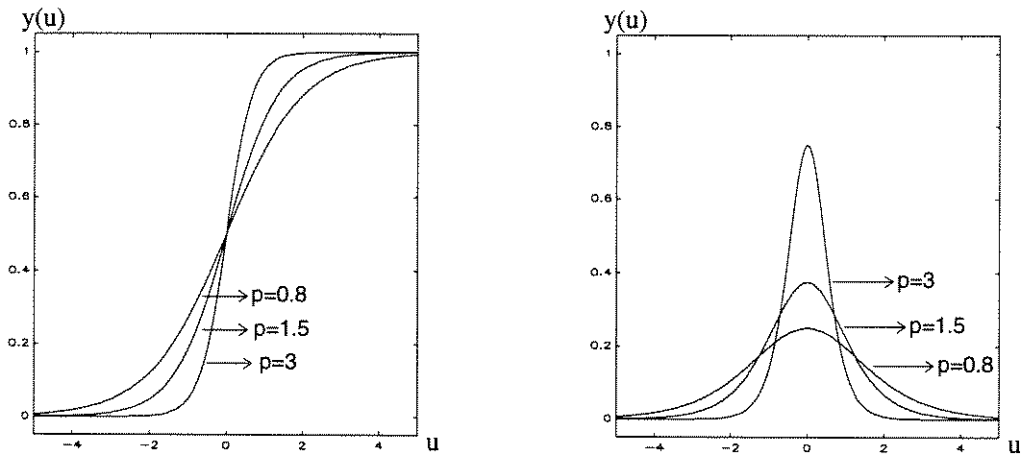


FIGURA 2.8 - Função logística e sua derivada

Uma informação importante é o valor da derivada parcial em relação à entrada interna, dada na forma:

$$\frac{\partial y}{\partial u} = py(1-y) \tag{2.6}$$

observe-se que para valores de $p \to \infty$ a função logística aproxima-se à função sinal. Na realidade, a origem deste tipo de função também está vinculada à preocupação em limitar o intervalo de variação da derivada da função sinal, mas ao contrário da função linear por partes, sua derivada também é uma função contínua.

Função tangente hiperbólica:

Pelo fato de apresentar a função logística valores de saída limitados entre a faixa $]0,1[$, em muitos casos esta é substituída pela função tangente hiperbólica, que preserva a

forma sigmoidal da função logística mas assume valores positivos e negativos na faixa]-1,1[. A função tangente hiperbólica e sua derivada estão definidas da seguinte forma:

$$y(u) = \tanh(pu) = \frac{e^{pu} - e^{-pu}}{e^{pu} + e^{-pu}} \tag{2.7}$$

$$\frac{\partial y}{\partial u} = p(1 - y^2) \tag{2.8}$$

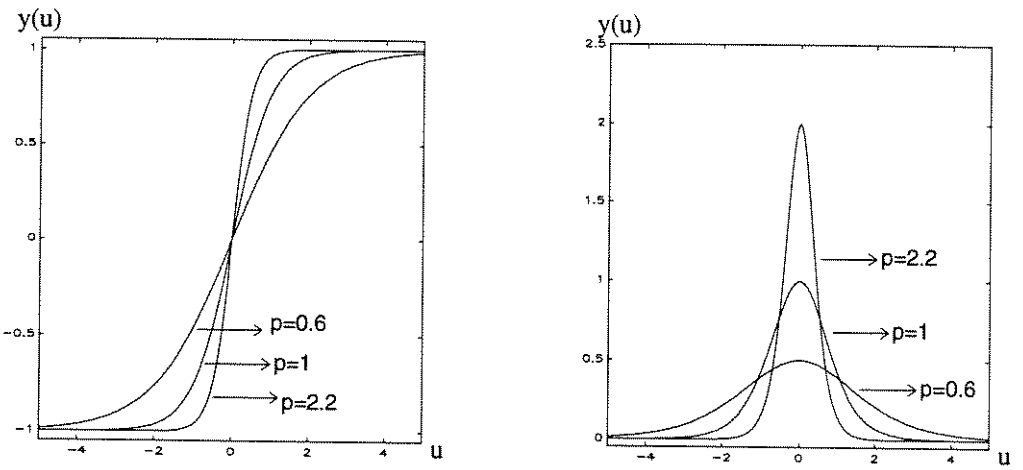


FIGURA 2.9 - Função tangente hiperbólica e sua derivada

A partir de agora sempre que a saída do neurônio seja exclusivamente binária, utilizaremos como função de ativação a função sinal e, nos casos onde as saídas sejam contínuas, utilizaremos a função tangente hiperbólica com p=1, que também chamaremos função sigmóide.

2.3.3- Topologias

É possível identificar três tipos de topologias de conexão dos neurônios para formar uma rede neural artificial:

1) Rede totalmente interconectada

2) Redes não recorrentes em camadas

3) Redes recorrentes em camadas

O primeiro tipo representa as topologias onde cada unidade se conecta consigo mesma e com todas as outras unidades da rede. O segundo caso engloba as ligações entre as saídas das unidades de um nível inferior em direção às entradas de todas as unidades de níveis superiores, não existindo conexões entre os elementos de um mesmo nível. O terceiro tem a mesma estrutura do segundo tipo, mas com a possibilidade de saídas de unidades de um nível superior estabelecerem conexões com entradas de todas as unidades de um nível inferior a aquela, representando uma realimentação interna de informação, geralmente realizada com o auxílio de operadores de atraso.

Estes três tipos de topologias permitem a construção de redes neurais extremamente complexas, cobrindo a quase totalidade dos sistemas neurais conhecidos e experimentados até hoje. Neste trabalho serão utilizados apenas arquiteturas de redes neurais em camadas sendo importante distinguir nestas, três tipos de unidades processadoras:

Unidades de entrada: *podem receber sinais do mundo exterior*

Unidades de saída: *enviam sinais ao mundo exterior*

Unidades escondidas: *seus sinais de entrada e saída são internos da rede.*

2.3.4- Redes neurais multicamada

A rede que utilizaremos neste trabalho, do tipo *"perceptron"* multicamada, merece um detalhado mais específico. Em geral a quantidade de entradas utilizadas é proporcional a dimensão do padrão de características, sendo estas do tipo freqüencial e/ou do tipo temporal. A quantidade de saídas, quando a rede neural é utilizada em reconhecimento de padrões, é proporcional à quantidade de classes, sendo que um estado de ativação alto implica a pertinência à uma classe. A quantidade de neurônios intermediáriarios não está relacionada por nenhuma limitação externa, porém seu valor é fundamental para que a rede possa separar eficientemente o espaço de estados. Se este valor é muito pequeno a rede não contará com suficientes graus de liberdade para separar o espaço de estados numa quantidade de classes desejada, pelo contrário se for muito grande vai a implicar que a rede "sobreajustará" os valores de entrada e não conseguirá uma boa generalização da base de dados.

2.4- APRENDIZAGEM DE REDES NEURAIS

2.4.1- Introdução

A aquisição de conhecimento pela rede neural artificial se dá pela aplicação de métodos de aprendizagem baseados na modificação das intensidades das conexões entre neurônios. O desenvolvimento de métodos de aprendizagem para a rede neural multicamada apresentam processamento predominantemente numérico e se relacionam com conceitos muito conhecidos na aplicação de métodos de estimação de parâmetros, procedimento fundamental em áreas como classificação de padrões, processamento de sinais e controle adaptativo.

É importante qualificar as potencialidades deste processo. Por exemplo poderíamos admitir que uma rede possa modificar sua topologias ou modificar as funções das unidades processadoras ou simplesmente modificar a intensidade das conexões. A grande diversidade de alternativas possíveis para o processo de aprendizagem e o evidente crescimento da complexidade para situações mais abrangentes impõem fortes restrições em quanto à capacidade da rede de ajustar um conjunto de seus parâmetros e funções. Portanto não devem ser criadas expectativas exageradas quanto à abrangência do processo de aprendizagem em redes neurais artificiais, pelo menos no estado atual de desenvolvimento.

Neste trabalho, serão aplicados processos de aprendizagem capazes de alterar apenas a intensidade das conexões da rede neural.

2.4.2- Paradigmas - Supervisionada ou Não Supervisionada

Em princípio é possível distinguir dois grandes paradigmas de aprendizagem:

Aprendizagem supervisionada ou associativa

Aprendizagem não-supervisionada ou auto-organizada.

Os qualificativos supervisionada e não-supervisionada são procedentes da teoria de reconhecimento de padrões. A distinção entre os dois paradigmas de aprendizagem está baseada na disponibilidade ou não, para o algoritmo de aprendizagem das respostas corretas e os exemplos de entrada apresentados. Chama-se caso supervisionada quando se apresenta à rede neural as saídas desejadas para cada exemplo da base de dados e caso contrário,

quando não se apresenta a saída desejada e trata-se que a rede encontre uma estrutura própria da base de dados.

No caso dos algoritmos de aprendizagem supervisionada aplicados a problemas de reconhecimento de padrões, a resposta desejada em geral se especifica como um valor alto que indica pertinência a uma classe e baixo para a não pertinência. Desta maneira é possível detectar erros de classificação que podem comandar o processo de aprendizagem. Um procedimento comum é aplicar uma política de recompensa para as classificações corretas e punição para as classificações incorretas. Em termos matemáticos, os erros de classificação geral dos sinais de erro numérico que determinam uma direção de ajuste para a intensidade das interconexões da rede neural.

No caso de algoritmos de aprendizagem não supervisionada aplicados a problemas de reconhecimento de padrões, é realizado basicamente um agrupamento dos padrões a fim de estabelecer classes de decisão. Neste caso, não existe uma definição de erro, sendo que o processo de aprendizagem é conduzido pela definição de políticas de competição entre as unidades processadoras. Como competição se entende uma interação entre as unidades de forma de estabelecer núcleos de ativação quando se apresenta um determinado padrão na entrada. Asim sendo, de acordo com as características deste núcleo de ativação (posição, intensidade, dimensão, etc), os padrões de entrada podem ser adequadamente classificados.

Os dois paradigmas de aprendizagem utilizam o conceito de informação local para realizar a aprendizagem. No caso supervisionada, o erro deverá ser definido localmente, enquanto que no caso não supervisionada, a competição se estabelece entre unidades processadoras que obedecem a algum critério de proximidade. Desta forma, um aspecto importante da aprendizagem em redes neurais é fazer que a rede apresente um determinado comportamento global através da implementação local da aprendizagem.

Os algoritmos de aprendizagem não supervisionada geralmente exigem uma carga de processamento menor, mas apresentam uma precisão numérica inferior ao caso supervisionada. O critério de eleição entre os dois paradigmas de aprendizado vai depender da aplicação. No presente trabalho utilizaremos unicamente algoritmos de aprendizagem supervisionadas.

2.4.3- Políticas de apresentação de padrões

Nos casos de reconhecimento de padrões o treinamento é de tipo "off-line", no sentido que inicialmente cria-se uma base de dados com padrões representativos do

conjunto que se deseja reconhecer. Porém existem várias políticas a respeito da apresentação dos padrões de treinamento permitindo o estabelecimento de ciclos de treinamento distintos. Cada política procura privilegiar algum critério e a eleição da política mas adequada dependerá das especificações de cada aplicação. Entre as políticas mas divulgadas podemos citar:

Política 1: Aplicação do treinamento somente depois da apresentação de todos os padrões à rede neural. Esta política privilegia o desempenho global do treinamento em detrimento do desempenho específico respeito de cada padrão da base de dados. Analisando em termos de evolução do erro entre as saídas da rede e as saídas desejadas, é possível afirmar que esta política privilegia a média do erro em detrimento da variância.

Política 2: Aplicação do treinamento somente nos padrões que produzem um desempenho abaixo de um determinado nível, esta política é exatamente o oposto da apresentada anteriormente, privilegiando a variância em detrimento da média dos erros existentes entre os padrões desejados e aqueles produzidos pela rede neural.

Política 3: Apresentação aleatória dos padrões de treinamento; é uma alternativa intermediária com relação às duas políticas anteriores atribuindo igual importância para a média e a variância do erro.

Política 4: Apresentação seqüencial dos padrões de treinamento, ordenação dos padrões disponíveis para o treinamento e apresentação seqüencial à rede neural. Esta ordenação pode seguir uma série de critérios, como por exemplo níveis de complexidade, similaridade e correlação.

2.4.4- Definição de erros

Para a utilização do paradigma supervisionada no treinamento de redes neurais é imperioso definir um critério de erro entre as saídas da rede e os valores desejados. Em geral este critério de erro é de tipo escalar, apesar de ser extraído de relações vetoriais.

Existem diversas alternativas entre as quais eleger, e na continuação apresentamos as mais utilizadas:

Erro Quadrático Médio:

É calculado como a soma do quadrado das diferenças entre o vetor saída da rede e o vetor da saída desejada divididos pelo número de fatores. A interpretação geométrica desta definição de erro é o quadrado da distância entre os dois extremos dos vetores correspondentes.

Se as saídas da rede são definidas pelo vetor y_i, com i=0,..,n-1; e os valores desejados para a saída por d_i, com i=0,..,n-1, o erro quadrático médio é :

$$EQM=\sum_{i=0}^{n}(y_i-d_i)^2 \tag{2.9}$$

Esta medida de erro é muito utilizada em problemas de classificação onde se utiliza um vetor de valores desejados como "*um de todos*", isto é, um valor alto para a saída que indica a classe e valor 0 ou -1 para as outras. No que segue deste trabalho, esta medida de erro será a utilizada.

Mínima Entropia Relativa:

A origem desta medida de erro provém da teoria de informação. A idéia por trás desta proposta é a minimização da entropia relativa entre a distribuição de probabilidade do vetor de saídas da rede e a distribuição de probabilidade do vetor de saídas desejadas (tomadas como distribuições aproximadas).

Esta medida de erro é utilizada quando se deseja aproximar via aprendizagem , as distribuições de probabilidade discretas. Sua equação é:

$$MER=\sum_{i=0}^{n}\{d_i\cdot\ln(\frac{d_i}{y_i})+(1-d_i)\cdot\ln(\frac{1-d_i}{1-y_i})\} \tag{2.10}$$

2.4.5- Influência do estado inicial:

Outra propriedade do processo de treinamento supervisionada é a influência exercida pelo estado inicial da rede no resultado do treinamento, ou seja, mesmo no caso da garantia de minimização do erro (função custo), o mínimo encontrado pode não ser o mínimo global. Considere o mapa apresentado na figura 2.10 relacionando o estado da rede

com a função custo. O processo de minimização da função custo produzirá resultados diferentes para os estados iniciais dados por A e B, sendo que o mínimo global somente é obtido partindo do estado inicial dado por B.

Também é importante destacar a existência de descontinuidades no processo de ajuste de pesos, provocando alterações significativas no procedimento de convergência. Por exemplo mesmo com o auxilio da figura 2.10 pode-se verificar que para estados da rede muito próximos como é o caso dos estados C e D, o estado final de convergência é totalmente diferente.

FIGURA 2.10 - Mínimos locais e globais

2.4.6- Treinamento do neurônio - regra *"perceptron"* e prova de Convergência.

Em 1957, Frank Rosenblatt [9], trabalhando na Universidade de Cornell, criou uma das primeiras redes neurais artificiais com a habilidade para aprender. Seu trabalho se fundamenta no modelo apresentado por McCullogh & Pitts, em 1943. Seu modelo de estudo chamado "***perceptron***", e no contexto do seu trabalho, o "perceptron" refere-se exclusivamente ao modelo neuronal com função não linear do tipo sinal. Ele desenvolveu um algoritmo supervisionada de treinamento e testou a convergência dos pesos sinápticos para problemas linearmente separáveis. Ao longo deste trabalho generalizaremos o termo *"perceptron"* a modelos de neurônios com função não linear do tipo sigmóide.

Na figura 2.11 seguinte apresentaremos o modelo de *"perceptron"* utilizado por Rosenblatt:

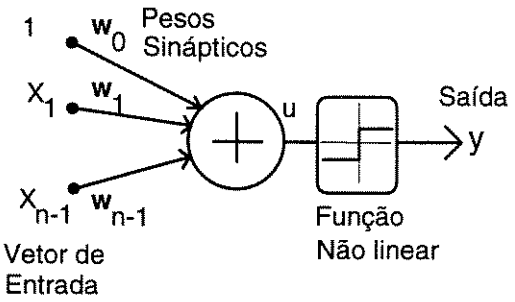


FIGURA 2.11 - "perceptron"

A equação que implementa o *"perceptron"* é um produto escalar entre o vetor de entrada e o vetor de pesos sinápticos, passado por uma função sinal. Para que o hiperplano separador possa passar fora do origem fazemos $\mathbf{X}_0$=+1. Matematicamente podemos escrever da seguinte forma:

$$y = S(w^t X) = \begin{cases} 1, \text{ se } w^t X \geq 0 \\ 0, \text{ se } w^t X < 0 \end{cases} \tag{2.11}$$

Desenvolveremos a seguir a **regra de aprendizagem** do "perceptron". Somente consideraremos problemas linearmente separáveis e buscaremos pesos para conseguir a separação. Um procedimento simples é apresentar cada padrão e ver se ele produz a saída desejada. Se efetivamente a produz então ficará inalterado, se a saída não é a desejada, somaremos um termo que resulte proporcional ao produto entre a entrada e a saída desejada. Então podemos escrever o algoritmo numa forma simplificada:

$$\mathbf{w}_{ij} = \mathbf{w}_{ij} + \Delta \mathbf{w}_{ij} \tag{2.12}$$

onde

$$\Delta w_{ij} = \begin{cases} 2\eta y_i^u X_j^u & \text{se } y_i^u \neq d_i^u \\ 0 & \text{c.c.} \end{cases} \tag{2.13}$$

onde u é o número de padrões. Admitindo-se que as saídas desejadas são valores binários, $d_i = \pm 1$, então a equação 2.13, pode-se escrever de uma forma análoga:

$$\Delta \mathbf{w}_{ij} = \eta(1 - y_i^u d_i^u) y_i^u \mathbf{X}_j^u \tag{2.14}$$

que é o mesmo:

$$\Delta \mathbf{w}_{ij} = \eta(y_i^u - d_i^u)\mathbf{X}_j^u \tag{2.15}$$

onde o parâmetro η é o coeficiente de aprendizagem. Podemos interpretar a equação 2.15 como constituída por dois parâmetros, um associado ao aprendizado do padrão correto e outro ao "desaprendizado" do padrão incorreto.

Se além de pedir que o sinal da saída seja o mesmo que o do valor desejado, é aconselhável também pedir que a saída linear h seja maior que um determinado limiar, de modo a assegurar que o limiar de separação esteja um pouco separado do padrão mais aproximado. Então pedimos que:

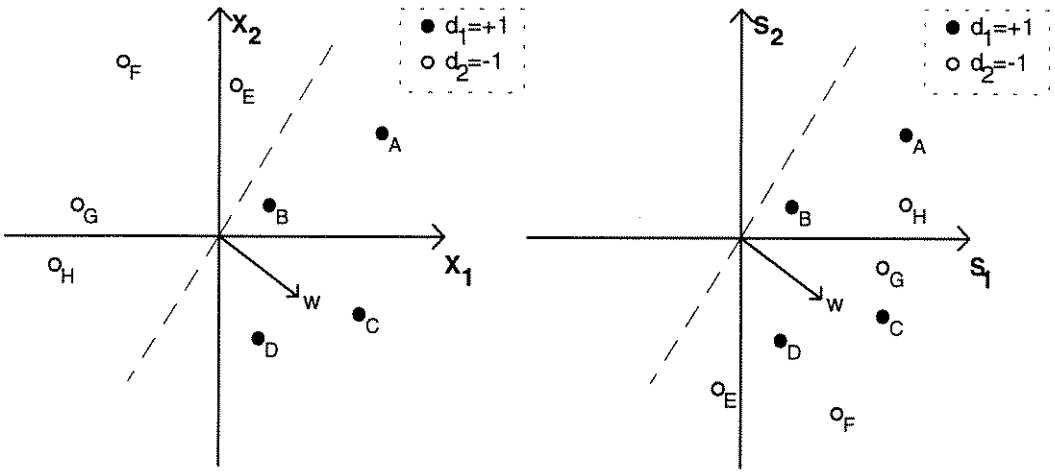
$$y_i^u h_i^u = y_i^u \sum_k \mathbf{w}_{ik}\mathbf{X}_k^u > Nk \tag{2.16}$$

Finalmente na equação 2.14 em função do novo requerimento obtemos:

$$\Delta \mathbf{w}_{ij} = \eta \mathbf{S}(Nk - y_i^u h_i^u) y_i^u \mathbf{X}_j^u \tag{2.17}$$

onde $\mathbf{S}(\bullet)$ é a função sinal. A equação 2.17 é a chamada **regra de aprendizagem do "perceptron"**, desenvolvida por Rosenblatt em 1962.

Desenvolveremos a seguir a **prova de convergência** da regra de aprendizagem do "perceptron". Determinaremos primeiro a condição que deve cumprir a direção dos pesos que resolve o problema. Em todo problema, do tipo linearmente separável, temos duas (ou mais) classes. Faremos a demonstração para um único "perceptron", já que a estensão da mesma é trivial e, portanto trabalharemos com problemas de somente duas classes. Neste caso atribuiremos saída positiva a uma classe, e saída negativa à outra. *O vetor de pesos w que resolva o problema será tal que o hiperplano por ele definido, separe as duas classes no plano cartesiano e deixe a um dos dois lados todos os padrões no plano* $\mathbf{X}^t d$. Onde $\mathbf{X}$ são as coordenadas do vetor e d são os valores desejados para as saídas (d_1=+1, d_2=-1). De acordo com esta definição plano $\mathbf{X}^t d$, os padrões da classe 1 não mudam de lugar e os padrões da classe 2 terão novas posições especulares respeito das antigas. Na figura 2.12 se apresenta um exemplo bidimensional para clarificar este conceito.



FIGURAS 2.12 e 2.13 - Condições do hiperplano de separação em planos X e S= $X^t d$

Em função da figura 2.13 observamos que para quaisquer vetor padrão, a direção do vetor **w** na qual todas a projeções são positivas não solucionará o problema. Dependendo do conjunto de padrões do problema, existirá um conjunto grande de tais direções ou um pequeno. Então poderemos usar esta observação para medir a dificuldade de resolver um dado problema em particular. Definiremos a continuação uma direção que depende unicamente da pior das projeções:

$$D(\mathbf{w}) = \frac{1}{|\mathbf{w}|} \min_{u} \mathbf{w}\mathbf{X}^u \tag{2.18}$$

Esta é simplesmente a distância da pior das projeções $\mathbf{X}^u$ sobre o plano perpendicular a **w**. O fator divisor o transforma somente em função da direção de **w.** Se D(**w**) é positivo então todos os padrões estarão no lado correto e o problema será solucionável.

Então para a prova de convergência assumiremos que o problema tem solução e provaremos que a regra de aprendizagem do *"perceptron"* alcançará a solução em um número finito de passos. Tudo o que devemos assumir é que existe um **w*** tal que D(**w***)> 0. Esta prova foi extraída da referência [32].

A essência da demonstração é encontrar limites para $|\mathbf{w}|$ e para projeção $\mathbf{w}\mathbf{w}^*$ sobre o vetor desejado **w***. Induziremos a conclusão que $\mathbf{w}\mathbf{w}^*/|\mathbf{w}|$ cresce indefinidamente se o número de atualizações M dos pesos do *"perceptron"* continua crescendo. Mas isto é impossível já que **w*** é fixo, de forma que a adaptação deve parar em algum momento.

Se definimos M^u como o número de vezes que o padrão foi usado para atualizar os pesos, então obtemos: (assumindo que os pesos iniciais são todos zero)

$$\mathbf{w} = \eta \sum_u M^u \mathbf{X}^u \tag{2.19}$$

Consideremos primeiro $\mathbf{w}\mathbf{w}^*$:

$$\mathbf{w}\mathbf{w}^* = \eta \sum_u M^u \mathbf{X}^u \mathbf{w}^* \geq \eta M \min_u \mathbf{X}^u \mathbf{w}^* = \eta M \, D(\mathbf{w}^*) |\mathbf{w}^*| \tag{2.20}$$

Então vemos que o produto escalar $\mathbf{w}\mathbf{w}^*$ cresce no máximo com M. Agora para encontrar um limite inferior para $|\mathbf{w}|$, consideramos a mudança na magnitude de $\mathbf{w}$ numa única atualização de um padrão α:

$$\Delta|\mathbf{w}|^2 = (\mathbf{w} + \eta \mathbf{X}^\alpha)^2 - \mathbf{w}^2 \tag{2.21}$$

$$= \eta^2 (\mathbf{X}^\alpha)^2 + 2\eta \mathbf{w} \mathbf{X}^\alpha \tag{2.22}$$

$$\leq \eta^2 N + 2\eta N k \tag{2.23}$$

$$= N\eta(\eta + 2k) \tag{2.24}$$

A inequação vem diretamente da condição $Nk \geq \mathbf{w}\mathbf{X}^\alpha$ para realizar uma atualização com o padrão α. Como usamos $\mathbf{X}^\alpha = \pm 1$ então $(\mathbf{X}^\alpha)^2 = N$. somando todos os incrementos de $|\mathbf{w}|^2$ para os M passos obtemos:

$$|\mathbf{w}|^2 \leq MN\eta(\eta + 2k) \tag{2.25}$$

Então $|\mathbf{w}|$ não cresce mais rápido do que $\sqrt{M}$, portanto o quociente $\mathbf{w}\mathbf{w}^*/|\mathbf{w}|$ não cresce mais rápido do que $\sqrt{M}$. Porém, isto é impossível já que o vetor $\mathbf{w}^*$ é fixo e o quociente $\mathbf{w}\mathbf{w}^*/|\mathbf{w}|$ representa uma projeção sobre este, então M deve deixar de crescer em algum momento. Computaremos em seguida o produto escalar normalizado:

$$\Phi = (\mathbf{w}\mathbf{w}^*)^2 \Big/ |\mathbf{w}|^2 |\mathbf{w}^*|^2 \tag{2.26}$$

que é o quadrado do co-seno do ângulo entre os vetores **w** e **w***. Como é um co-seno ele é obviamente menor ou igual a 1, então deve cumprir que:

$$1 \geq \Phi \geq MD(\mathbf{w}^*)^2 \eta / N(\eta + 2k) \tag{2.27}$$

Portanto da equação 2.27 poderemos derivar uma expressão para o limite superior de M, se utilizamos o menor **w*** possível, o que daria $D=D_{max}$:

$$M \leq N(1 + 2k/\eta)/D_{max}^2 \tag{2.28}$$

A equação 2.28 define o limite para o número de iterações M para que o vetor de pesos possa convergir ao valor ótimo.

2.4.7- Algoritmo LMS.

2.4.7.1- Superfície de erro quadrático médio e equação de Wiener-Hopf:

Nesta seção demonstramos que a superfície do erro quadrático médio de um combinador linear é uma função quadrática dos pesos, desta forma é facilmente percorrida pelo método do gradiente descendente.

Seja o padrão de entrada $\mathbf{X_k}$ e sua saída associada $\mathbf{d_k}$ extraídos de um processo estatisticamente estacionário. Durante a adaptação, o vetor de pesos varia de forma que ainda com entradas estacionárias, a saída $S_\mathbf{k}$ e o erro $e_\mathbf{k}$ serão geralmente não estacionários. Deve-se tomar muito cuidado com a definição da função de erro já que esta é variável com o tempo, de forma que tomaremos o que se define como uma "média no conjunto", isto é uma média sobre todas as realizações de um processo estocástico. Derivando a expressão do erro do combinador linear obtemos:

$$e_k^2 = (d_k - \mathbf{X}_k^T \mathbf{W}_k)^2 \tag{2.29}$$

$$= d_k^2 - 2d_k \mathbf{X}_k^T \mathbf{W}_k + \mathbf{W}_k^T \mathbf{X}_k \mathbf{X}_k^T \mathbf{W}_k \tag{2.30}$$

Assumindo que um conjunto de combinadores lineares como uma união com o mesmo conjunto de pesos $W_\mathbf{k}$ na mesma iteração; estes têm as mesmas entradas $X_\mathbf{k}$ e as mesmas saídas desejadas $d_\mathbf{k}$ derivadas de processos ergódicos estacionários. Cada

combinador linear produzirá um sinal de erro e_k individual e, tomando-se a média sobre cada uma destas saídas obteremos:

$$\mathbf{E}[e_k^2] = \mathbf{E}[d_k^2] - 2\mathbf{E}[d_k X_k^T]W_k + W_k^T \mathbf{E}[X_k X_k^T]W_k \tag{2.31}$$

Definindo o vetor **P** como a correlação cruzada entre a saída desejada d_k e o vetor X, de forma tal que:

$$\mathbf{P^T} = \mathbf{E}[d_k X_k^T] \tag{2.32}$$

onde E[•] é a esperança matemática e definindo a matriz **R** como a matriz de autocorrelação do vetor de entrada X_k:

$$\mathbf{R} = \mathbf{E}[X_k X_k^T] \tag{2.33}$$

$$= \mathbf{E}\begin{bmatrix} 1 & x_{1k} & \cdots & x_{nk} \\ x_{1k} & x_{1k}x_{1k} & \cdots & x_{1k}x_{nk} \\ \vdots & \vdots & & \vdots \\ x_{nk} & x_{nk}x_{1k} & \cdots & x_{nk}x_{nk} \end{bmatrix} \tag{2.34}$$

Esta, como toda matriz de autocorrelação, é uma matriz real, simétrica e semi-definida positiva. Então de acordo com as definições realizadas nas equações 2.32 e 2.33, a equação 2.31 fica:

$$\mathbf{E}[e_k^2] = \mathbf{E}[d_k^2] - 2\mathbf{P^T}W_k + W_k^T \mathbf{R} W_k \tag{2.35}$$

Vemos então que a equação da superfície é uma função quadrática dos pesos ou um parabolóide hiperbólico. Mas a propriedade mais importante que tem esta superfície é a convexidade (uma reta que une dois pontos quaisquer da superfície está sempre acima da mesma) o que garante a existência de um mínimo global.

Representaremos a seguir um gráfico com a função de erro quadrático médio para um combinador linear de dois pesos. Nos eixos XY estão os pesos do combinador linear e no eixo Z está a função de erro. Então um ponto quaisquer da grade representa o erro quadrático médio sobre o conjunto de treinamento quando os pesos da rede estão fixos nos valores associados com um ponto da grade. O ajuste dos pesos pelo método do gradiente implica a modificação dos pesos de forma tal a evoluir sobre superfície para o mínimo erro possível.

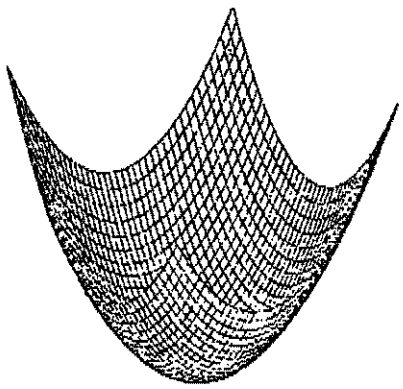


FIGURA 2.14- Função erro quadrático médio linear (extraída de [7])

O gradiente da função de erro no vetor de pesos ótimo, que chamaremos **W***, obtém-se quando o gradiente se anula. O gradiente da função erro é obtido diferenciando a equação 2.35 num ponto qualquer:

$$\nabla_k = \frac{\partial E[e_k^2]}{\partial \mathbf{W}_k} = -2\mathbf{P} + 2\mathbf{R}\mathbf{W}_k \tag{2.36}$$

Igualando a equação 2.36 a zero obtemos o vetor **W***, que resulta ser o ótimo:

$$\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P} \tag{2.37}$$

A equação 2.37, também chamada ***equação de Wiener-Hopf***, expressa que o vetor de pesos ótimo **W***, no sentido de minimizar o erro quadrático médio, é obtido através do produto entre a inversa da matriz de correlação do vetor de entrada e o vetor de correlação cruzada entre a saída desejada e a entrada. No caso que o faixa da matriz **R** seja menor que sua dimensão o vetor de pesos ótimo **W*** será um subespaço do espaço de estados dos pesos. O filtro cujos coeficientes sejam solução da equação de Wiener-Hopf é chamado ***filtro de Wiener***. A equação 2.37 em forma de sistema de equacões também chama-se equação de Yule-Walker.

2.4.7.2-Método de descenso de maior inclinação ("steepest descent"):

Observando a estrutura da equação 2.37, vemos que para obter o vetor de pesos ótimo **W*** devemos inverter uma matriz de NxN, onde N é o número de "taps" do filtro. Do ponto de vista do cálculo numérico sabe-se que o cálculo da inversa de uma matriz por um lado, requer o cálculo de operações de ordem de N^3, e por outro lado dependendo do número de condição da matriz e da precisão utilizada, o resultado pode estar longe do

verdadeiro valor. Para evitar o uso da inversão de matriz a fim de resolver a equação 2.37 prefere-se o uso de algoritmos de tipo iterativo, mais concretamente o algoritmo de descenso de maior inclinação.

Neste algoritmo uma estimação inicial dos pesos é atualizada de forma a aproximar-se do ponto ótimo descrito na equação 2.37. A direção de atualização é a do descenso de maior inclinação na superfície de erro e esta é de direção oposta ao do vetor gradiente nesse ponto.

É importante destacar que ter uma direção de ajuste não implica conhecer o tamanho do passo a realizar-se. A área da matemática que trata este tipo de problemas , chama-se otimização de sistemas, proveu algoritmos iterativos de busca de um passo que minimiza algum critério dada uma direção de ajuste. São conhecidos os métodos de Fibonacci, da seção áurea e outros. Mais freqüentemente utiliza-se um passo fixo denotado por η denominado coeficiente de aprendizagem.

$$\nabla \mathbf{W}_k = -\eta \nabla_k \tag{2.38}$$

Então o mecanismo de atualização pelo descenso de maior inclinação em forma matricial dá por resultado:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \nabla \mathbf{W}_k \tag{2.39}$$

$$= \mathbf{W}_k - \eta \nabla_k \tag{2.40}$$

A equação 2.39 permite formalizar o método do descenso de maior inclinação como o método que atualiza uma estimação do vetor de pesos de um filtro de Wiener como uma soma do valor anterior, e um termo de correção proporcional ao negativo do gradiente da função superfície de erro quadrático médio.

O método do descenso de maior inclinação é exato no sentido que não se utilizaram aproximações na sua derivação. Por um lado a função de custo minimizada é uma "média no conjunto" sobre múltiplas realizações de um processo estocástico e por outro lado o algoritmo assume um conhecimento exato das matrizes de correlação **R** e **P**.

2.4.7.3-Algoritmo LMS:

As propriedades do método do descenso de maior inclinação mostradas no parágrafo anterior fazem que sua aplicação seja muito limitada. As limitações surgem de dois fatores, por um lado não é possível conhecer a estatística de conjunto já que na prática somente se conhece um único erro, mas o maior problema é o conhecimento exato das matrizes **R** e **P**. Especificamente para o cálculo do negativo do gradiente é preciso conhecer exatamente as matrizes de correlação **R** e **P**, o qual na prática é difícil que ocorra, sobretudo ainda mais em ambientes onde a estatística dos sinais muda com o tempo.

O algoritmo LMS que apresentaremos a continuação realiza um descenso aproximado de maior inclinação via uma *estimação instantânea do verdadeiro gradiente*. Esta estimação é obtida computando a derivada do erro quadrático instantâneo com relação aos pesos.

$$\hat{\nabla}_k^T = \frac{\partial e_k^2}{\partial \mathbf{W}_k} = \left[\frac{\partial e_k^2}{\partial W_{0k}} \; \frac{\partial e_k^2}{\partial W_{1k}} \cdots \frac{\partial e_k^2}{\partial W_{nk}} \right] \tag{2.41}$$

O LMS utiliza esta estimação do gradiente na equação de atualização dos pesos sinápticos do filtro:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta(-\hat{\nabla}_\mathbf{k}) \tag{2.42}$$

$$= \mathbf{W}_k - \eta \frac{\partial e_k^2}{\partial \mathbf{W}_k} \tag{2.43}$$

Desenvolvendo a equação 2.43 obtemos:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - 2\eta \mathbf{e}_\mathbf{k} \frac{\partial \mathbf{e}_\mathbf{k}}{\partial \mathbf{W}_\mathbf{k}} \tag{2.44}$$

$$= \mathbf{W}_k - 2\eta e_k \frac{\partial (d_k - \mathbf{W}_k^T \mathbf{X}_k)}{\partial \mathbf{W}_k} \tag{2.45}$$

Como a saída desejada d_k é independente dos pesos W_k obtemos finalmente a equação de atualização do algoritmo LMS:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\eta e_k \mathbf{X}_k \tag{2.46}$$

Na atualização dos pesos usa-se uma estimação do verdadeiro gradiente através do cômputo da derivada do único erro disponível. Como isto é uma estimação da direção de descenso não corresponderá à do gradiente verdadeiro e será de tipo aleatória. Por esta razão também chama-se ao algoritmo LMS como *algoritmo de descenso estocástico.*

Demonstra-se que a estimação do gradiente utilizada é uma estimação não polarizada do verdadeiro gradiente. Isto é, o valor médio da estimação e o verdadeiro valor do gradiente coincidem. Isto significa que em média a estimação do gradiente convergirá ao verdadeiro valor e que o vetor de pesos converge em média ao valor ótimo **W***.

O valor da constante η é importante para a evolução do algoritmo já que determina a estabilidade e a convergência do mesmo. Para padrões independentes no tempo demonstra-se que a convergência em média e variância dos pesos está garantida para a maior parte dos casos práticos no caso que:

$$0 < \eta < \frac{1}{\mathrm{tr}\,[R]} \tag{2.47}$$

onde tr[•] significa traço e é a somatoria dos elementos da diagonal da matriz de autocorrelação **R**. Outra expressão do traço de **R** é $E[\mathbf{X}\mathbf{X}^T]$.

2.4.8- Treinamento de redes neurais - "backpropagation"

2.4.8.1- Inicialização do algoritmo

O algoritmo "backpropagation" é uma extensão do algoritmo LMS para o treinamento de redes neurais multicamada. Devido a que ele é um algoritmo iterativo é preciso indicar uma condição inicial dos parâmetros livres da rede, como os pesos sinápticos e os limiares. A escolha incorreta podería levar o sistema rede-algoritmo ao fenômeno de ***saturação prematura***. Podemos descrever este fenômeno da seguinte forma: supondo que por efeito da inicialização o estado de ativação do neurônio é alto e de sinal positiva, a saída da função sigmóide será +1. Se a saída desse neurônio deve ser negativa, o erro será grande, mas como o estado de ativação é alto, a derivada da sigmóide será de valor muito pequeno o que implicará um valor de correção dos pesos muito baixo.

A forma usual de inicialização, quando não existe informação "a priori", é a atribuição de pesos de acordo com números aleatórios gerados por uma distribuição uniforme de valor baixo. Em Lee *et al.* (1991) apresenta-se uma fórmula para a

minimização da probabilidade de saturação prematura. Esta regra pode-se resumir nas seguintes premissas:

1) A saturação incorreta é evitada quando se escolhem pesos através de uma escolha de valores iniciais provenientes de uma variável aleatória de distribuição uniforme sobre uma pequena faixa de valores.

2) A saturação incorreta é de ocorrência menos provável, se o número de neurônios é mantido baixo de acordo com a correta operação da rede neural.

3) A saturação incorreta raramente ocorre se os neurônios operam na sua região linear.

2.4.8.2- Desenvolvimento do algoritmo

Normalmente o objetivo da adaptação é reduzir o erro de classificação sobre todo o conjunto de treinamento. A função de erro mais freqüentemente usada é o Erro Quadrático Médio, em inglês, Mean Squared Error (MSE). Esta já foi definida na equação 2.35. O método do gradiente tem por propósito objetivo os pesos da rede neural durante a apresentação de cada padrão por gradiente descendente com o objetivo de reduzir o MSE sobre todos os padrões. Existem métodos mais sofisticados de minimização como o Quase-Newton ou de Gradiente Conjugado, que oferecem melhores propriedades de convergência, porém com um custo computacional mais elevado. A discussão que segue restringe-se à minimização do MSE pelo método do gradiente.

A adaptação da rede pelo método do gradiente começa com um valor inicial arbitrário $\mathbf{W}_0$ para os pesos do sistema, escolhido a fim de reduzir a probabilidade de saturação prematura. O gradiente da função MSE é calculado e o vetor de pesos é alterado na direção correspondente ao negativo do gradiente calculado. Este procedimento se repete provocando que o MSE seja sucessivamente reduzido em média e causando que o vetor de pesos se aproxime a um valor ótimo local.

O método do gradiente descedente pode ser descrito pela seguinte equação:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\nabla_k) \tag{2.48}$$

onde μ é o coeficiente de aprendizagem, cujo valor é crítico para controlar a estabilidade e a convergência do algoritmo, e ∇_k é o valor do gradiente no ponto da superfície do MSE correspondente a $\mathbf{W} = \mathbf{W}_k$.

Desenvolveremos a continuação o **algoritmo de aprendizagem** para um neurônio com função não linear, também chamada SAE por Single Adaptative Element. O objetivo é a minimização do $E[e(k)]^2$ sobre todo o conjunto de treinamento através de uma eleição apropriada de $\mathbf{W}$, onde $E[\bullet]$ é o valor esperado de $(\bullet)$. Para tal, derivaremos o algoritmo de "backpropagation" para o SAE. Uma estimação instantânea do gradiente é obtida com cada apresentação de um vetor de entrada e o método do gradiente descendente é usado para minimizar o erro. A estimação instantânea do gradiente é obtida durante a apresentação do k-ésimo vetor $\mathbf{X}_k$ está dada por:

$$\hat{\nabla}_k = \frac{\partial (e_k)^2}{\partial \mathbf{W}_k} = 2e_k \frac{\partial e_k}{\partial \mathbf{W}_k} \tag{2.49}$$

onde o erro no tempo k é definido como:

$$e_k = d_k - y_k = d_k - \mathrm{sgm}(s_k) \tag{2.50}$$

diferenciando a equação anterior respeito de $\mathbf{W}_k$:

$$\frac{\partial e_k}{\partial \mathbf{W}_k} = -\frac{\partial \mathrm{sgm}(s_k)}{\partial \mathbf{W}_k} = -\mathrm{sgmd}(s_k) \frac{\partial s_k}{\partial \mathbf{W}_k} \tag{2.51}$$

Como a saída linear é: $s_k = \mathbf{X}_k^t \mathbf{W}_k$, então:

$$\frac{\partial s_k}{\partial \mathbf{W}_k} = \mathbf{X}_k \tag{2.52}$$

a expressão final para a equação 2.51 é:

$$\frac{\partial e_k}{\partial \mathbf{W}_k} = -\mathrm{sgmd}(s_k)\mathbf{X}_k \tag{2.53}$$

logo inserindo a equação 2.53 na 2.49 obtemos:

$$\hat{\nabla}_k = -2e_k \mathrm{sgmd}(s_k)\mathbf{X}_k \tag{2.54}$$

Usando esta estimação do gradiente com o método do gradiente descendente temos um método para a minimização do erro quadrático médio ainda depois que a saída linear passou através da função sigmóide não linear. Então o algoritmo da por resultado:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\hat{\nabla}_k) \tag{2.55}$$

$$= \mathbf{W}_k + 2\mu e_k \mathrm{sgmd}(s_k)\mathbf{X}_k \tag{2.56}$$

A equação 2.56 é a equação do "backpropagation" para o SAE. O nome "backpropagation" tem mais sentido quando o algoritmo é utilizado num *"perceptron"* multicamada, que será estudado a continuação.

Se a eleição da função sigmóide é a função tangente hiperbólica, definida como:

$$y_k = \tanh(s_k) = \left(\frac{1-e^{-2s_k}}{1+e^{-2s_k}}\right) \tag{2.57}$$

então a derivada da sigmóide é:

$$\mathrm{sgmd}(s_k) = \frac{\partial \tanh(s_k)}{\partial s_k} \tag{2.58}$$

$$= 1-(\tanh(s_k))^2 = 1-y_k^2 \tag{2.59}$$

Finalmente obtemos a equação do "backpropagation" para o SAE com função sigmóide:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k (1-y_k^2)\mathbf{X}_k \tag{2.60}$$

Desenvolveremos o **algoritmo "backpropagation"** para o ***"perceptron"* multicamada.** A publicação do algoritmo de "backpropagation", conhecido através do livro publicado por Rumelhart *et al.* [12], foi sem dúvida o desenvolvimento de maior inflûencia no campo das redes neurais durante a década passada. Mas isto ocorreu somente nesta época já que os investigadores de redes neurais utilizava como funções não lineares as funções do tipo sinal e esta é incompatível com a técnica mencionada.

Os conceitos básicos do "backpropagation" são simples, mas desafortunadamente estas idéias são obscurecidas por uma notação relativamente complicada. Existem

numerosas fontes que apresentam a seu modo o algoritmo de "backpropagation", entre as mais conhecidas podemos citar [16] e [20]. Porém, para a apresentação do algoritmo, escolhimos a forma apresentada no livro de Hertz *et al* [32], porque aparece como a mais simples e direta.

Neste caso o erro quadrático médio a minimizar é a soma dos erros quadráticos entre as saídas da rede e seus correspondentes valores desejados, isto é:

$$e_k^2 = \sum_{i=0}^{n-1} e_{ik}^2 = \sum_{i=0}^{n-1} (d_i - y_i)_k^2 \qquad (2.61)$$

Em sua forma mais simples o treinamento por "backpropagation" começa pela apresentação de um padrão de entrada **X** à rede, recorrendo a mesma na forma de gerar um vetor de resposta **E** na saída, e computar os erros para cada saída. O próximo passo consiste em enviar os efeitos dos erros para atrás através da rede, na forma de associar o "***erro quadrático derivativo***" δ com cada "perceptron", computar o gradiente de cada δ e finalmente atualizar os pesos de cada *"perceptron"* baseados em sua correspondente estimação do gradiente. Logo se apresenta um novo padrão e o processo é repetido. Devemos passar várias vezes pelo conjunto de treinamento para minimizar o erro quadrático médio.

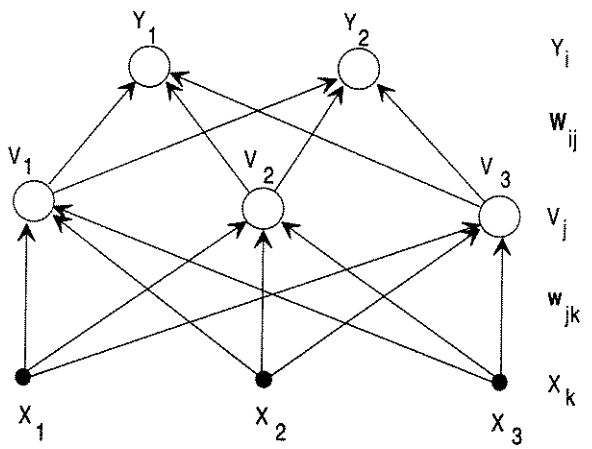


FIGURA 2.15 - Rede neural de duas camadas

Consideraremos um *"perceptron"* de duas camadas como o apresentado na figura 2.15. As notações utilizadas são: as unidades de saída são denotadas por Y_i, as unidades ocultas por V_j e os terminais de entrada por X_k. Os pesos sinápticos entre as entradas e as unidades ocultas são denotados por w_{jk} e os pesos entre as unidades intermediárias e as saídas por W_{ij}. Notar que o índice **i** sempre se refere a uma unidade de saída, **j** a uma

unidade oculta e **k** a um terminal de entrada. Utilizaremos a notação u como supraíndice para diferenciar os padrões, onde a entrada k alimentada pelo padrão u será indicada por X_k^u. Os valores desejados para as saídas serão denotados por d_i^u. As entradas podem ser binárias ou apresentar valores contínuos. Utilizaremos **N** para o número de unidades de entrada (k=1,2,...,N), **P** para o número de padrões de entrada (u=1,2,...,P), **M** para o número de saídas da rede (i=1,2,...,M) e **H** para o número de neurônios da camada oculta (j=1,2,...,H).

Dado um padrão u, a unidade oculta j recebe uma entrada:

$$h_j^u = \sum_k \mathbf{w}_{jk} X_k^u \tag{2.62}$$

e produz uma saída:

$$V_j^u = \mathrm{sgm}(h_j^u) = \mathrm{sgm}(\sum_k \mathbf{w}_{jk} X_k^u) \tag{2.63}$$

A unidade de saída i então recebe:

$$h_i^u = \sum_j \mathbf{W}_{ij}\, V_j^u = \sum_j \mathbf{W}_{ij}\, \mathrm{sgm}(\sum_k \mathbf{w}_{jk} X_k^u) \tag{2.64}$$

e produz uma saída externa:

$$Y_i^u = \mathrm{sgm}(h_i^u) = \mathrm{sgm}(\sum_j \mathbf{W}_{ij} V_j^u) = \mathrm{sgm}(\sum_j \mathbf{W}_{ij}\, \mathrm{sgm}(\sum_k \mathbf{W}_{jk} X_k^u)) \tag{2.65}$$

Então para ser conseqüentes com as definições que tomamos, redefiniremos a função de erro:

$$E[w] = \frac{1}{2} \sum_{ui} (d_i^u - Y_i^u)^2 \tag{2.66}$$

que resulta ser

$$E[w] = \frac{1}{2} \sum_{ui} (d_i^u - \mathrm{sgm}(\sum_j \mathbf{W}_{ij}\, \mathrm{sgm}(\sum_k \mathbf{w}_{jk} X_k^u)))^2 \tag{2.67}$$

Esta é claramente uma função contínua e diferenciável de cada peso, de forma que podemos usar o algoritmo de gradiente descendente para convergir para os pesos apropriados. Ainda devemos diferenciar os pesos da camada de saída dos da camada oculta. A continuação desenvolveremos a atualização para cada um destes conjuntos.

Para os pesos $\mathbf{w}_{jk}$ entre a camada oculta e a camada de saída, a regra de gradiente descendente da por resultado:

$$\Delta \mathbf{W}_{ij} = -\eta \frac{\partial E}{\partial \mathbf{W}_{ij}} = \eta \sum_{u} (d_i^u - \mathbf{Y}_i^u)\ \mathrm{sgmd}(h_i^u) V_j^u \tag{2.68}$$

$$= \eta \sum_{u} \delta_i^u V_j^u \tag{2.69}$$

onde definimos:

$$\delta_i^u = \mathrm{sgm}(h_i^u)\ (d_i^u - Y_i^u) \tag{2.70}$$

Para uma única camada não faz muito sentido realizar esta definição, mas se verá que será muito útil para as redes multicamada. Estes são os "***erros quadráticos derivativos***" apresentados anteriormente.

Para os pesos $\mathbf{W}_{ij}$ entre os terminais de entrada e a camada oculta, deveremos aplicar a regra de gradiente descendente diferenciando respeito dos pesos $\mathbf{w}_{jk}$ que estão mais profundamente inseridos na equação 2.67. Então utilizando a regra da cadeia obtemos:

$$\Delta \mathbf{w}_{jk} = -\eta \frac{\partial E}{\partial \mathbf{w}_{jk}} = -\eta \sum_{u} \frac{\partial E}{\partial V_j^u} \frac{V_j^u}{\partial \mathbf{w}_{jk}} \tag{2.71}$$

$$= \eta \sum_{ui} (d_i^u - Y_i^u)\ \mathrm{sgmd}(h_i^u) \mathbf{W}_{ij}\ \mathrm{sgmd}(h_j^u) X_k^u \tag{2.72}$$

$$= \eta \sum_{ui} \delta_i^u\ \mathbf{W}_{ij}\ \mathrm{sgmd}(h_j^u)\ X_k^u \tag{2.73}$$

$$= \eta \sum_{u} \delta_j^u\ X_k^u \tag{2.74}$$

onde

$$\delta_j^u = \text{sgmd}(h_j^u) \sum_i W_{ij}\ \delta_i^u \tag{2.75}$$

Notar que a equação 2.75 tem a mesma estrutura que a equação 2.70, já que ambas são definições de erros quadráticos derivativos, mas para diferentes camadas. A diferença origina-se na equação 2.75, então o que se propaga para atrás não é o erro como na equação 2.70 mas se propagaram os erros derivativos δ_i^u da primeira camada ponderados por seu correspondente peso de conexão W_{ij}. Este é justamente o mecanismo da retropropagação do erro que da nome ao algoritmo ***"backpropagation"***. Então podemos utilizar a mesma rede da propagação, numa versão bidirecional da mesma para a atualização dos pesos.

Na prática este algoritmo é utilizado em forma incremental: um padrão é apresentado na entrada e logo todos os pesos são atualizados antes de que se apresente outro padrão. Na realidade veremos mais adiante, que existem outras políticas de atualização dos pesos da rede neural.

A continuação apresentaremos um diagrama passo a passo do algoritmo de "backpropagation":

1) Inicializar os pesos da rede com pequenos valores aleatórios.

2) Apresentar à entrada da rede um padrão de entrada X_k^u, com k=1,2,...,N.

3) Propagar o padrão de entrada através da rede, na forma de calcular as saídas produzidas Y_i^u, com i=1,2,...,M, utilizando a equação 2.65:

$$Y_i^u = \text{sgm}\left(\sum_j W_{ij}\ \text{sgm}\left(\sum_k w_{jk} X_k^u\right)\right)$$

4) Computar os erros derivativos da última camada, utilizando a equação 2.70:

$$\delta_i^u = \text{sgm}(h_i^u)\ (d_i^u - Y_i^u)$$

com i=1,2,...,M, pela comparação entre os valores atuais das saídas Y_i^u com os valores desejados para as mesmas d_i^u para o padrão atual μ.

5) Computar os erros derivativos das camadas intermediárias pela retropropagação dos erros derivativos da camada de saída, via equação 2.75

$$\delta_j^u = \text{sgmd}(h_j^u) \sum_i W_{ij}\ \delta_i^u$$

com j=1,2,...,H. Desta forma calculamos todos os erros derivativos para todos os neurônios da rede.

6) Com os erros derivativos calculados, calculamos os deltas para cada peso da rede:

$$\Delta W_{ij} = \eta\, \delta_i V_j$$

$$\Delta w_{jk} = \eta\, \delta_j X_k$$

Logo atualizam-se os pesos da rede seguindo a fórmula:

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

$$w_{jk} = w_{jk} + \Delta w_{jk}$$

com i=1,2,...,M, j=1,2,...,H, e k=1,2,..,N.

7) Voltar ao passo 2, e repetir para todos o padrões, isto é μ=1,2,...,P.

2.4.9- Problemas e melhoria de velocidade do "backpropagation"

Podemos por um lado destacar os **problemas** com os quais deve enfrentar-se o algoritmo de "backpropagation" desenvolvido:

- Um dos principais problemas de convergência em quanto a velocidade e a mínimos locais apresenta-se pelo tipo de função de erro utilizada e a não linearidade do neurônio. Devido a não linearidade perde-se a condição de convexidade da função de erro e portanto aparecem os mínimos locais, onde o algoritmo podería não convergir.

Apresentamos na figura 2.16 um gráfico da função de erro para uma rede de dois pesos sinápticos extraída do toolbox de redes neurais do programa Matlab 4.0 [28].

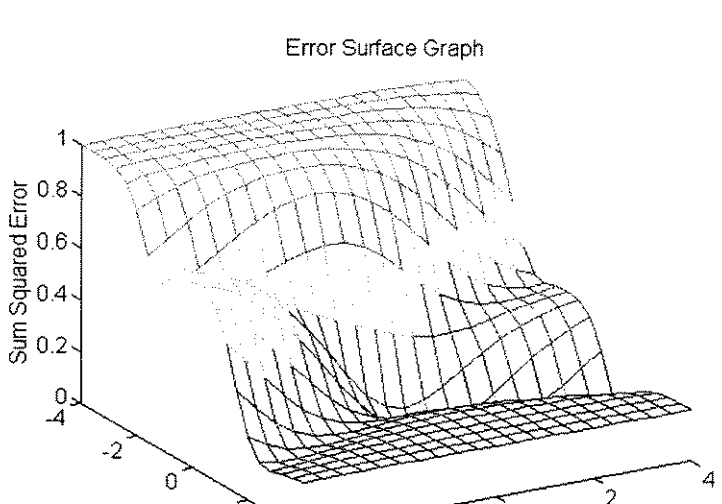


FIGURA 2.16 - Função de erro para dois pesos.

• Outro problema deve-se a que o erro sobre o qual deverão aprender os neurônios da camada intermediária, é um erro flutuante. Isto se explica por sua dependência do erro derivativo do peso da última camada, o qual se modifica na medida que progressa o aprendizado.

Por outro lado destacaremos **possíveis melhorias** do algoritmo em função dos problemas apresentados, em quanto à velocidade de convergência e a evitar mínimos locais.

• Incorporamos à regra de aprendizagem um termo que é uma combinação linear da correção anterior. Esta combinação leva uma constante β que regra a sua magnitude. Demonstramos que esta constante deve estar na seguinte faixa: $0 \leq |\beta| < 1$, para assegurar sua convergência. Pode-se interpretar o efeito deste agregado como uma "inércia" à evolução dos pesos no espaço de estados. Uma vantagem imédiata é que em caso de encontrar-se com um mínimo local, onde a primeira derivada se anula, se a inércia é suficientemente grande, poderá ser superado.

$$\nabla W_{ij}(n) = \alpha \nabla W_{ij}(n-1) + \beta \delta_j(n) y_i(n) \tag{2.76}$$

Se o sinal do gradiente e do momento coincidem, o câmbio atual do peso sináptico será maior, portanto tem um efeito de *aceleração* em lugares de descenso pronunciado. Se pelo contrário os sinais não coincidem, o câmbio de peso diminuirá, o qual daria um efeito

estabilizador nas direções onde os sinais oscilam em sinal. Também pode interpretar-se como uma tentativa de agregar informações de segunda ordem na evolução dos pesos sinápticos.

• É importante que os valores desejados para as saídas não estejam na faixa de saturação dos neurônios, para evitar o fenômeno de saturação prematura apresentado anteriormente. Então deve-se escolher os valores desejados de valor absoluto inferiores a saturação da função sigmóide. Uma boa eleição para a função sigmóide na prática é +0.9 e -0.9.

• Os valores iniciais dos neurônios devem ser provistos por uma variável aleatória de distribuição uniforme sobre uma pequena faixa de valores. Mas esta faixa não deve ser muito pequena, caso contrário os gradientes serão pequenos e a rede demorará em convergir.

• Dentro do possível a ordem de apresentação dos padrões à rede deveria ser aleatório. Este é um fator crítico para melhorar a convergência da rede, pois poderia evitar a queda em mínimos locais da função de custo.

• Para diminuir a quantidade de cálculos pode-se detectar os casos onde o gradiente é abaixo de um limiar e evitar que se realize a correção.

• Em próximos capítulos apresentaremos um teorema que expressa a capacidade de aproximação universal das redes neurais multicamada. Em princípio este teorema não implica que a camada de saída deva ter funções não lineares limitadoras da faixa de saída. Porém em algumas aplicações serão imprescindíveis como por exemplo a aproximação de funções de probabilidade. Então tem em geral uma melhoria significativa da velocidade se a aplicação o permite, a eliminação das funções não lineares da camada de saída.

2.4.10 Aplicação a um problema de classificação.

Realizaremos diversos exemplos de treinamento com o algoritmo "backpropagation" de uma rede neural com uma camada de neurônios ocultas, com o objetivo de destacar as características do mesmo. Realizaremos também diversas experiências variando parâmetros ou modificando políticas de apresentação de padrões, para evidenciar experimentalmente o comportamento do algoritmo e sua sensibilidade aos parâmetros determinados pelo usuário.

O problema de treinamento proposto é a classificação de um padrão bidimensional pertencente a duas classes igualmente prováveis. Cada uma das classes está definida por uma função de densidade de probabilidade gaussiana de valor médio μ da matriz de covariância Σ. Os valores utilizados nas experiências são:

CLASSE 1 : $\mu = [0.3\ 0.5]$

$$\Sigma = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.4 \end{bmatrix}$$

CLASSE 2 : $\mu = [-0.3\ -0.4]$

$$\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.4 \end{bmatrix}$$

Gerou-se uma realização do sistema como base de dados para o treinamento do classificador. O mecanismo foi o seguinte: de acordo com o resultado de uma variável aleatória discreta com dois estados igualmente prováveis, se gera uma realização de cada uma das classes com as características apresentadas. Foram escolhidas variâncias altas com o objetivo de que a confusão entre classes fosse elevada de modo que o aprendizado seria uma tarefa complicada. Porém a curva de aprendizagem foi razoavelmente rápida, como veremos nos gráficos.

Na seguinte figura se apresenta um diagrama de 20 realizações de cada uma das classes, onde denotamos com "**o**" as pertencentes à classe 1 e com "**+**" as pertencentes à classe 2.

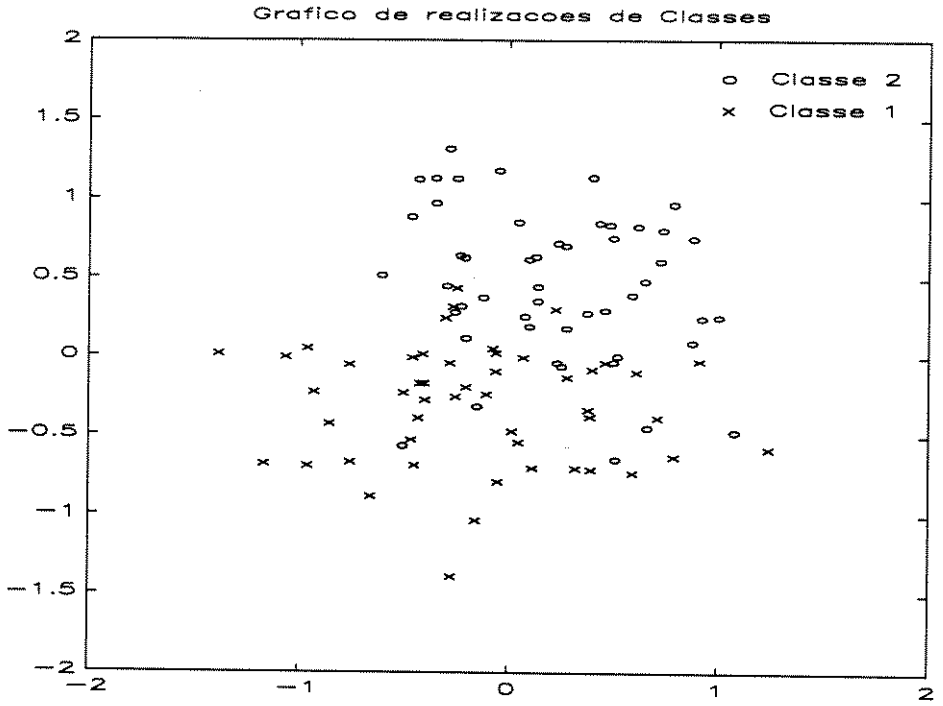


FIGURA 2.17 - Realizações das classes

EXPERIMENTO N°1: Para estudar a dependência das características de aprendizagem do algoritmo com relação aos parâmetros tais como o coeficiente de aprendizagem e o coeficiente de momentum. Repetiremos diversos experimentos com um conjunto fixo de valores de pesos e de pontos iniciais. O objetivo é fixar regras empíricas para uma adequada seleção destes parâmetros, já que não existem fórmulas para sua determinação exata. Do único que se dispõe é de um conjunto de cotas para estes valores como foram apresentadas oportunamente. A política de apresentação de padrões é apresentar um exemplo de cada classe e logo com o gradiente calculado para ambas apresentações, se produz uma atualização dos pesos.

Para 5 conjuntos diferentes de pesos sinápticos e de pontos iniciais. Implementaremos diferentes combinações de valores do coeficiente de aprendizagem e momentum. Os valores utilizados são:

α	λ
0.1	0.0
0.1	0.5
0.1	0.9
0.5	0.0
0.5	0.5
0.5	0.9

0.9	0.0
0.9	0.5
0.9	0.9

TABELA 2.1 - Realizações das classes

Apresentamos a continuação, os gráficos dos resultados médios de 5 repetições. O primeiro gráfico é para comparar os diferentes coeficientes de aprendizagem sem o termo momentum. Logo se apresentam os gráficos para α=0.1 variando o coeficiente de momentum. Logo para α=0.5 e finalmente para α=0.9.

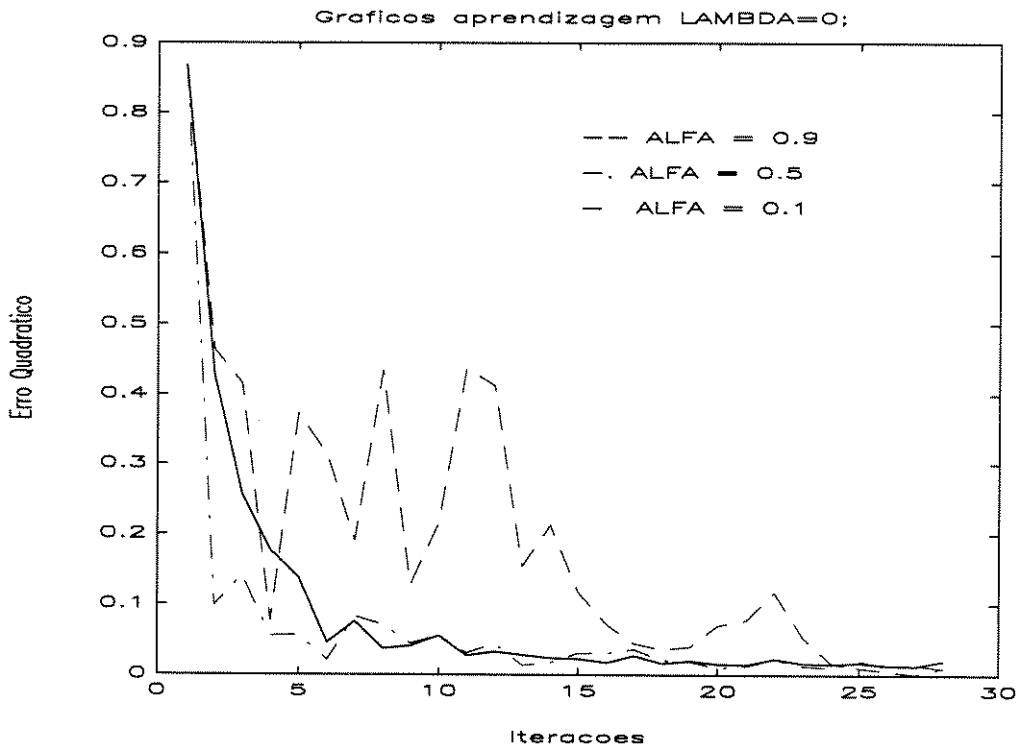


FIGURA 2.18 - Diferentes alfas sem momentum

Evidencia-se na figura 2.18 que, ainda para este problema trivial, a eleição do coeficiente de aprendizagem é crítica para a velocidade do treinamento. O valor de 0.1 para o coeficientes de aprendizagem é muito reduzido e o descenso do erro médio é lento. O valor de 0.5 é o que melhor combina velocidade de descenso e erro médio mínimo. O valor de 0.9 revela que para valores altos se produzem oscilações significativas no treinamento. Isto deve-se a que, em cada ponto se calcula o gradiente e em superfícies de nível não circulares este gradiente não aponta diretamente ao mínimo, um valor alto do coeficiente de aprendizagem provocará a passagem para outra curva de nível de similar altura, retardando o treinamento e produzindo esas oscilações típicas.

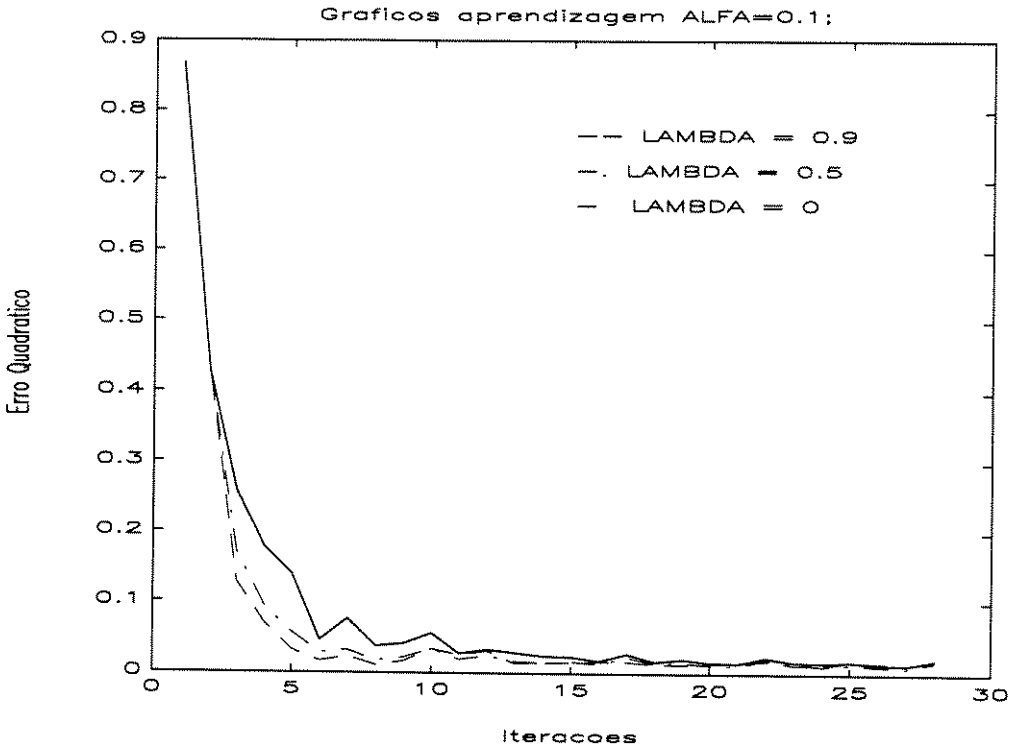


FIGURA 2.19 - Alfa = 0.1 e diferentes momentum

Podemos concluir que para baixos valores do coeficiente de aprendizagem, como ser é 0.1, o efeito do momentum é aumentar a velocidade de convergência, já que a curva com al=0.9 proporciona valores de erro menores em menor tempo, e estabiliza o aprendizado, já que as oscilações são menores quanto maior seja o momentum. Este último efeito é a "inércia" respeito dos câmbios nos pesos.

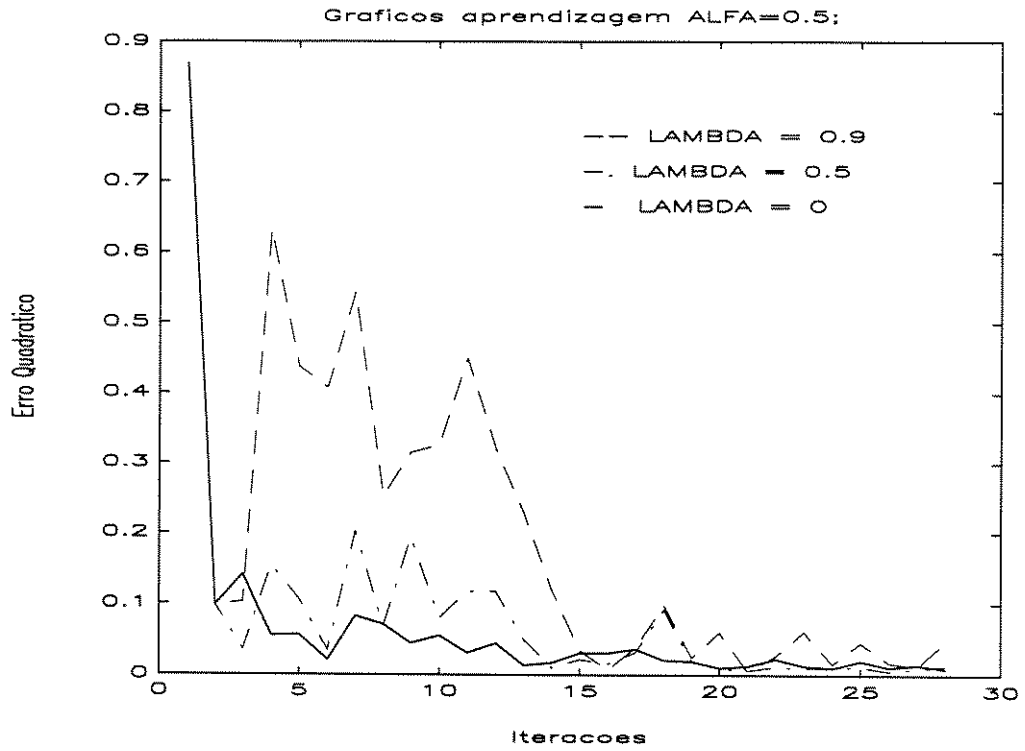


FIGURA 2.20 - Alfa = 0.5 e diferentes momentum

Para valores do coeficiente de aprendizagem de 0.5, o efeito do coeficiente de momentum não se aprecia significativamente, possivelmente isto é devido as particularidades do problema.

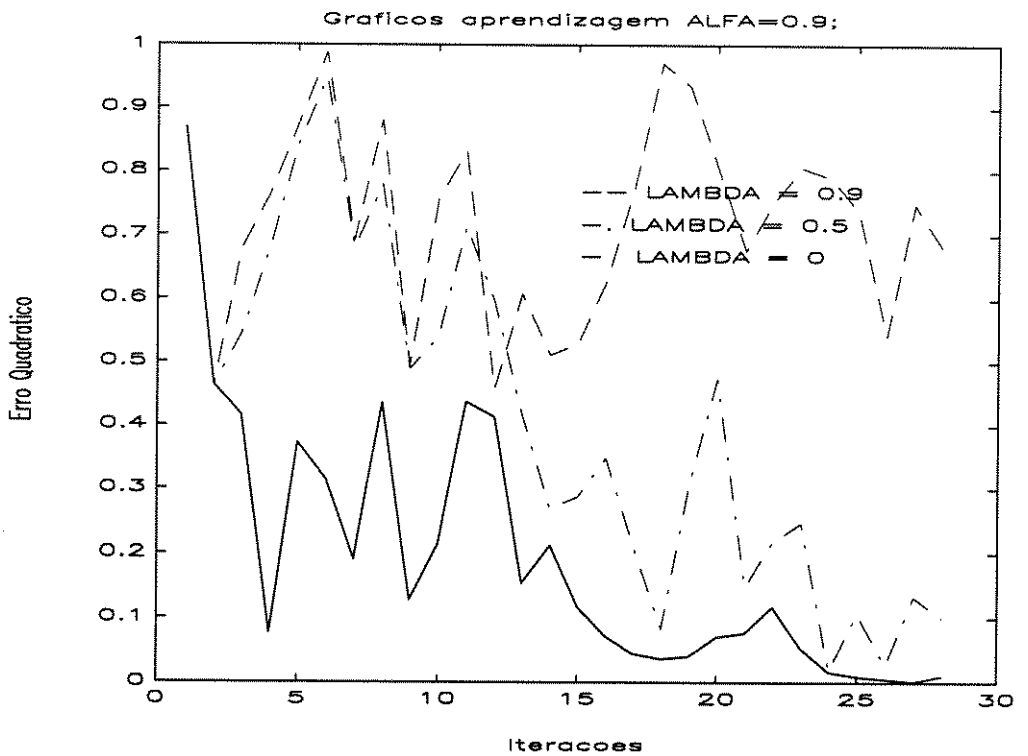


FIGURA 2.21 - Alfa = 0.9 e diferentes momentum

Evidencia-se pelas curvas que o valor do coeficiente de aprendizagem de 0.9 é grande demais para trazer benefícios no aprendizado. Neste caso o efeito do coeficiente de momentum é justamente o contrário que o evidenciado anteriormente. Para um valor elevado do coeficiente de aprendizagem, um valor elevado do coeficiente de momentum introduz maior instabilidade no treinamento, sendo que em casos limites (=0.9) pode provocar à oscilação do treinamento.

EXPERIMENTO Nº2: Experimentamos a política de apresentação de padrões a fim de estudar as modificações no aprendizado frente a diferentes políticas de apresentação dos padrões. Já que o objetivo do aprendizado é a redução do erro médio sobre o conjunto de treinamento, é importante estudar formas de chegar ao mínimo erro na forma mais rápida possível.

Como primeira política tomamos a atualização a cada apresentação de um padrão. Como segunda política tomamos a atualização logo da apresentação de um exemplo de cada classe. Este serve para casos como este, onde o número de classes é reduzido. Finalmente tomamos a atualização após a apresentação de todos os exemplos da classe. Repetiremos diversos experimentos com um conjunto fixo de valores de pesos e de pontos iniciais.

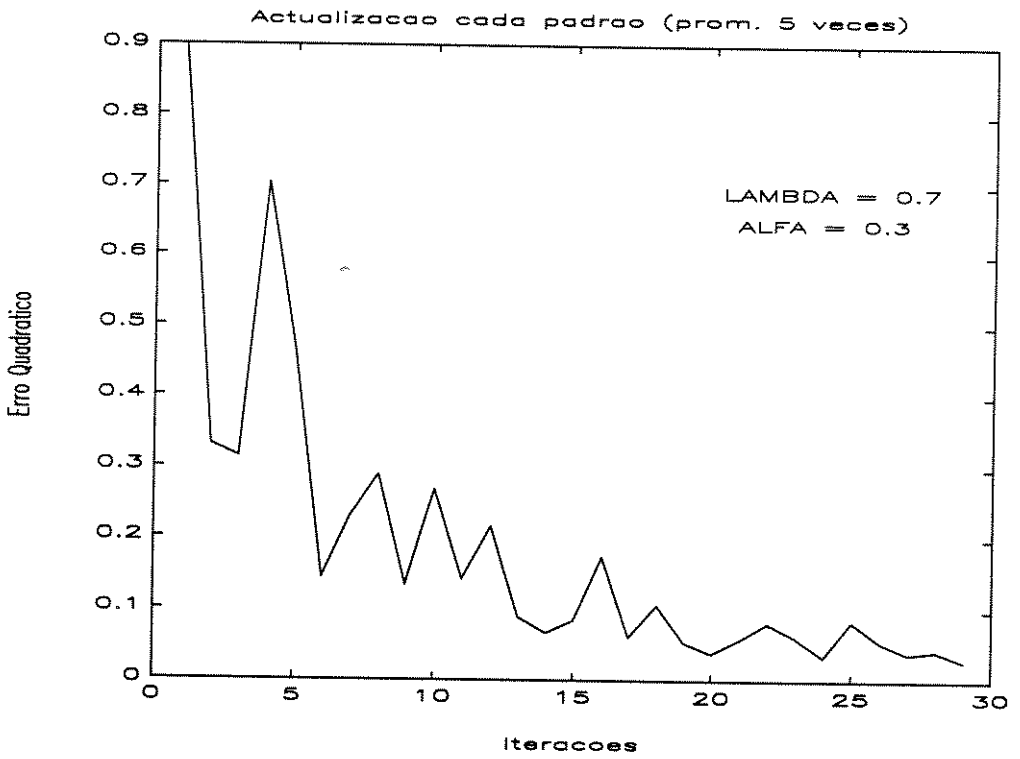


FIGURA 2.22 - Atualização a cada padrão

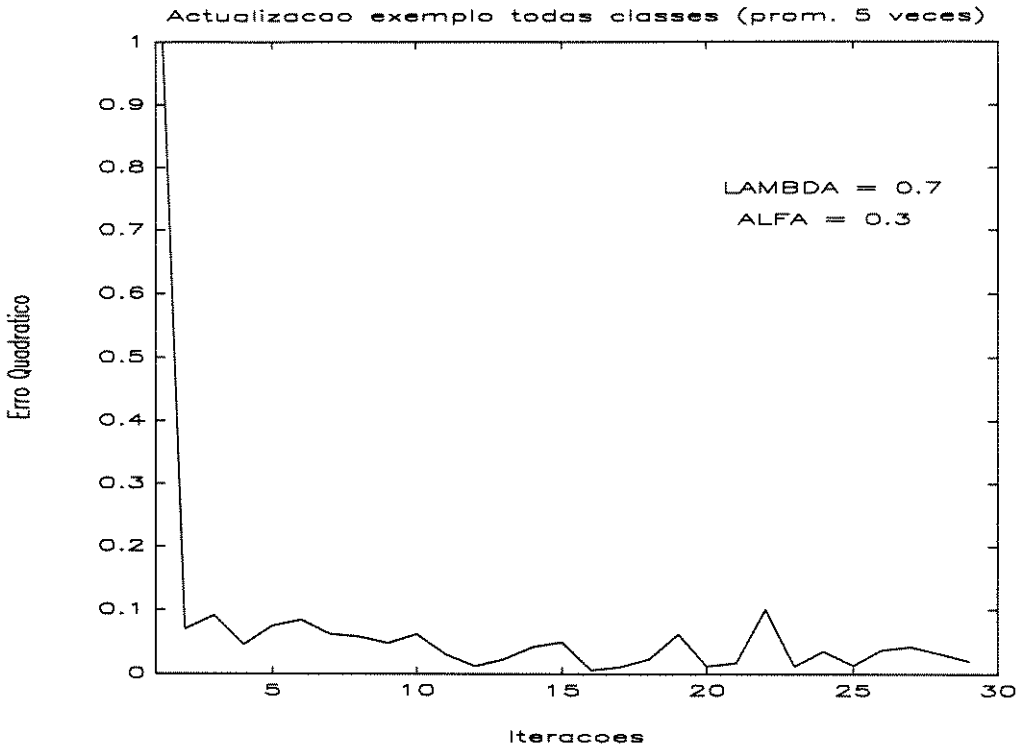


FIGURA 2.23 - Atualização logo de exemplo de cada classe

EXPERIMENTO Nº3: Experimentamos com o número de neurônios da camada intermediária respeito da velocidade de aprendizagem. Começamos com um número reduzido de neurônios capazes de resolver o problema, isto é 2, já que se deve contar com a entrada de bias. Posteriormente aumentou-se a 5 neurônios na camada intermediária e finalmente foram utilizados 10 neurônios na camada intermediária.

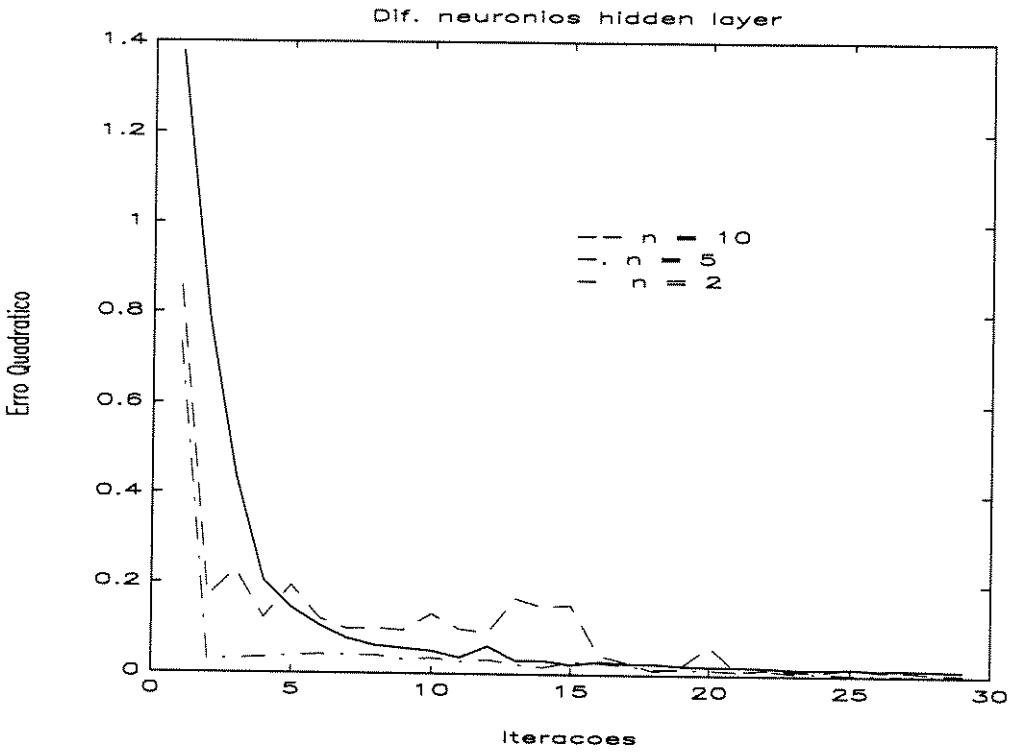


FIGURA 2.24 - Diferentes neurônios camada intermediária

Se evidencia que no começo do experimento as curvas de 2 e 5 neurônios são aproximadamente iguais e que a curva de 10 é mais lenta. O desenvolvimento geral da curva mostra que finalmente as três configurações resolvem o problema, sendo que quantos mais neurônios é maior a dimensionalidade do espaço o que possivelmente dificulte e atrase o aprendizado neste caso.

Porém, para avaliar corretamente o resultado de treinamento ao aumentar o número de neurônios deveríamos estudar como melhora a capacidade de representação do problema. Isto é, como melhorou a generalização da rede respeito de padrões não presentes na base de dados. Para isto deve-se computar os índices de reconhecimento das redes treinadas respeito de um conjunto de amostras não apresentes na base de dados de treinamento. É importante notar que a porcentagem de erros será elevada devido ao significativo solapamento entre as classes.

2.5- CAPACIDADE DE REPRESENTAÇÃO

2.5.1- Capacidade do neurônio- limitação da rede unicamada

Como apresentamos anteriormente, uma rede unicamada está limitada a separar o espaço de estados por hiperplanos separadores e não pode resolver problemas que não

sejam linearmente separáveis. Neste sentido pode resolver os problemas lógicos AND e OR, mas não pode resolver um problema como o XOR. Este último será tratado a continuação por sua importância histórica.

2.5.2- Análise do problema XOR

O estudo do problema OR-exclusivo ou XOR é muito importante por duas razões; por um lado é precisamente este problema o que produz um atraso importante no desenvolvimento das redes neurais e por outro lado é um problema elementar de separação que apresenta características de não-separabilidade linear e portanto revela as limitações dos "perceptrons" unicamada.

Considere os pontos (0,0), (0,1), (1,0), (1,1) no plano cartesiano $\mathcal{R}^2$, tal como se apresenta na figura 2.25. O objetivo é determinar uma rede com duas entradas $x_i \in \{0,1\}$ e $i=\{1,2\}$ e uma saída $e \in \{0,1\}$, tal que:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

TABELA 2.2- Tabela Saídas e entradas XOR

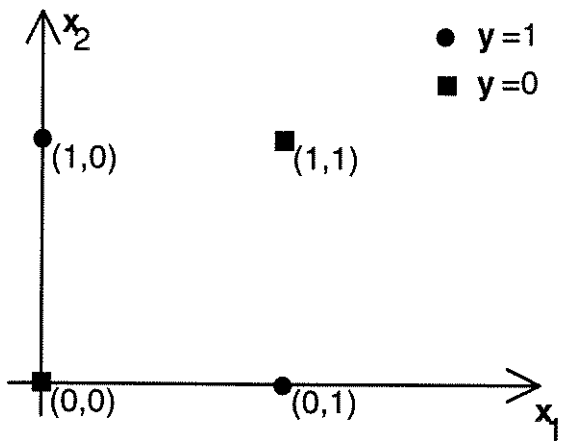


FIGURA 2.25 - Representação gráfica problema XOR

Demonstramos anteriormente que um *"perceptron"* classifica a entrada de acordo com a posição no espaço de estados e de acordo com um hiperplano de separação. Se

consideramos um *"perceptron"* com duas entradas e uma de bias, teremos então três pesos sinápticos w_1, w_2 e w_3. A equação que define este hiperplano de separação é e= f(X_1w_1+ X_2w_2+X_3w_3), onde f() é a função sinal. Observando a figura 2.25 vemos que não é possível resolver o problema XOR, isto é separar com um hiperplano o espaço em duas classes onde uma tenha saída e=1 e a outra tenha saída e=0. Quer dizer que não se pode resolver o problema XOR somente com uma camada de "perceptrons".

Esta é justamente a afirmação realizada por Minsky e Papert em seu famoso livro "Perceptrons" de 1969. Esse foi um resultado importante em sua época já que demonstrou uma limitação fundamental dos "perceptrons". Porém os autores afirmam que não havia razão para supor que redes multicamada poderiam resolver o problema. Eles deixam esta afirmação como uma hipótese, no sentido que deveria ser confirmada ou rejeitada no futuro.

Veremos a continuação como uma rede de duas camadas com dois neurônios na camada intermediária e uma neurônio na camada de saída pode resolver o problema XOR, como a apresentada na figura 2.26:

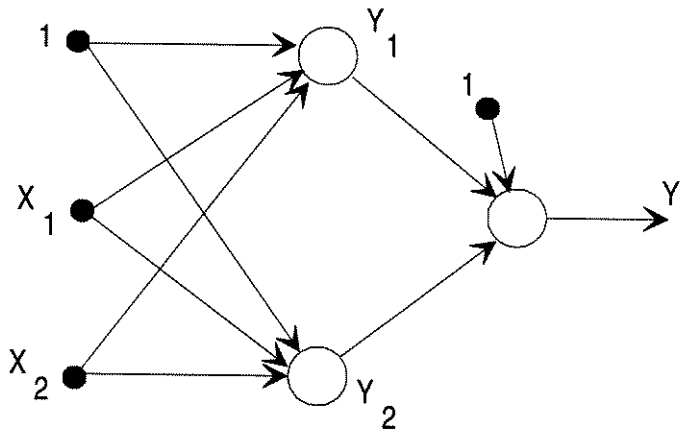


FIGURA 2.26 - Representação gráfico da rede neural de duas camadas para problema XOR

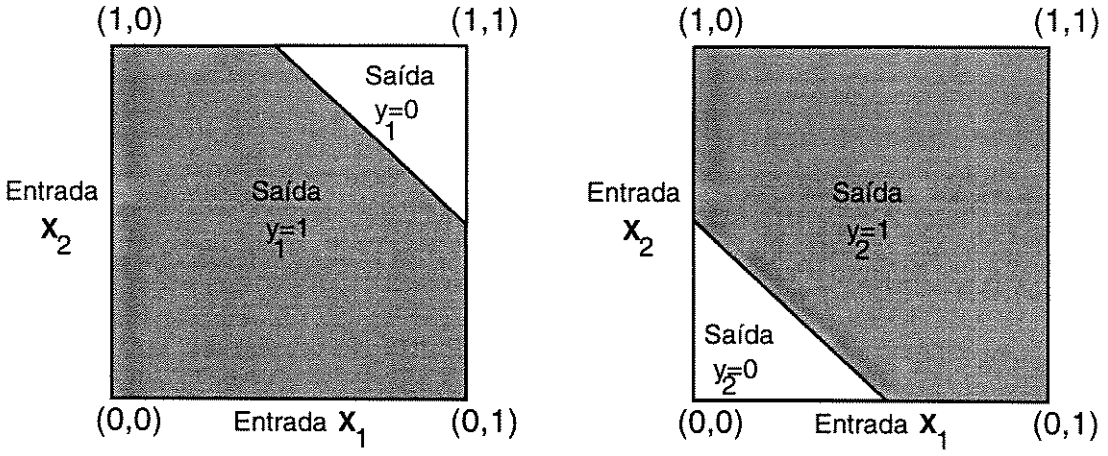


FIGURA 2.27 - Partição do espaço pelos neurônios 1 e 2.

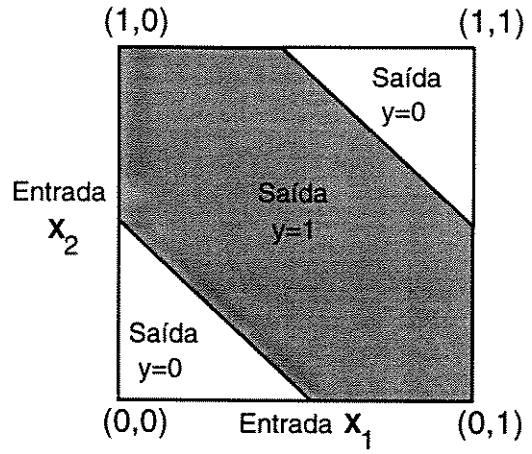


FIGURA 2.28 - Partição do espaço pelo neurônio 3.

Vemos que a separação do espaço de entrada realizado pelos dois neurônios da camada intermediária, como se apresenta nas figuras 2.27 e 2.28. Observamos que o neurônio 1 da saída inibitória à entrada (1,1) e exitatória para o resto, e que o neurônio 2 da saída inibitória à entrada (0,0) e exitatória para o resto. A função do neurônio 3 da camada de saída é realizar uma combinação linear das saídas dos neurônios da camada intermediária. De acordo com isto, da saída inibitória para o padrão (0,0) quando o neurônio 1 tem saída inibitória, e o neurônio 2 tem saída exitatória e o padrão (1,1) quando o neurônio 1 tem saída exitatória e o neurônio 2 tem saída inibitória. O resultado final se apresenta na figura 2.28, onde se revela que efetivamente o resultado final é uma correta separação do espaço para resolver o problema XOR.

2.5.3- Aproximação de regiões e número de camadas

Consideramos agora um problema mais generalizado de classificação de padrões num espaço de dimensão finita. Consideramos um mapeamento tal que o espaço esteja divido em duas classes e a classe interna seja formada pelo espaço interno à interseção de três retas no espaço.

Em função da explicação do problema XOR, podemos apreciar que este problema de classificação pode ser resolvido por uma rede de duas camadas com três entradas correspondentes a os valores 1, X1 e X2, duas saídas correspondendo a cada uma das classes e três neurônios na camada intermediária com saídas lineares, tal que cada uma realiza uma das fronteiras da classe A. Tomando a saída booleana AND, destas saídas se logra a rede neural,

Estudaremos a continuação a capacidade de representação das redes neurais desde o ponto de vista da classificação de padrões. É importante notar que esta capacidade provém das funções não lineares dos neurônios. A demonstração deste postulado é simples na medida em que se suponha que a função de transferência dos neurônios foram lineares, uma rede de várias camadas seria exatamente substituida por outra de uma camada, já que a composição de uma transferência linear sobre outra transferência linear pode ser reduzida a uma única transferência linear.

As capacidades de representação de redes neurais de um, dois ou mais neurônios na camada intermediária, com funções não lineares de tipo sinal é apresentada na tabela 2.3 (adaptada do famoso artigo de Lippmann [2]):

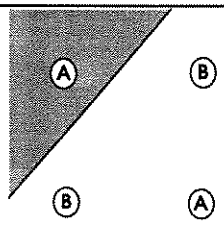
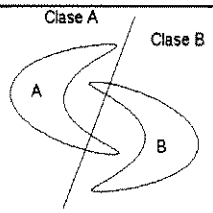
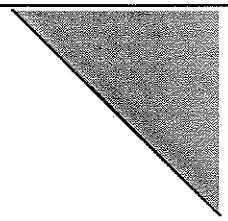
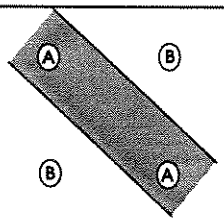
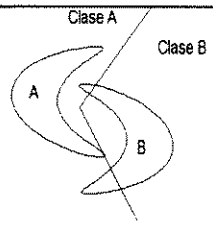
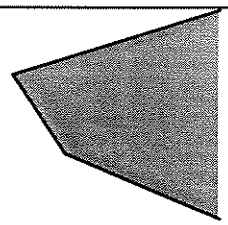
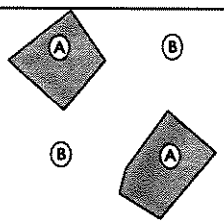
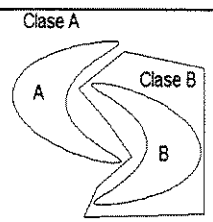
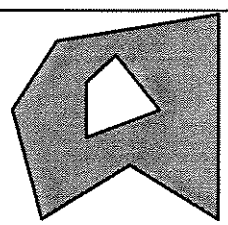
ESTRUTURA	TIPO DE REGIÕES DE DECISÃO	PROBLEMA OR - EXCLUSIVO	CLASSES COM REGIÕES SUPERPOSTAS	REGIÃO MAIS GERAL COM A ESTRUTURA
UMA CAMADA	MEIO PLANO LIMITADO POR UM HIPERPLANO			
DUAS CAMADAS	ABERTA CONVEXA OU REGIÃO FECHADA			
TRÊS CAMADAS	ARBITRÁRIA (Somente limitada pelo número de nodos)			

TABELA 2.3 - Capacidade de representação de redes neurais com função sinal

A segunda coluna apresenta as capacidades de representação, na terceira se apresenta a solução ao problema OR-exclusivo, na quarta se apresenta uma solução para um problema onde as classes estão entrelaçadas e finalmente se apresenta um diagrama com a forma mais geral possível.

Como se observa da tabela 2.3 um ***"perceptron"* de uma camada** realiza uma partição do espaço por um hiperplano cuja equação está definida pelos pesos sinápticos.

Um ***"perceptron"* de duas camadas** realiza figuras de tipo polígono convexo. A propriedade de convexidade se refere a que uma reta que uma dois pontos quaisquer da superfície deve estar totalmente incluída na mesma. Estas regiões são produto das interseções das regiões semi-plano formadas por cada nodo da primeira camada da rede. Cada nodo da primeira camada se comporta como um *"perceptron"* unicamada e apresenta uma saída alta em um dos lados do semi-plano. Se todos os pesos entre os N nodos da primeira camada são unitários e o offset é N-ε, onde $0<\varepsilon<1$, então o nodo de saída apresentará um estado de ativação alto somente se os estados de ativação de todos os neurônios da primeira camada fossem altos. Isto corresponde a uma operação lógica AND.

Então as regiões de saída alta serão as produzidos pelas interseções dos semiplanos dos neurônios da primeira camada. Estas regiões são convexas com tantos lados como nodos houver na primeira camada. Este tipo de região permite a uma rede de duas camadas resolver o problema XOR, mas não separar classes com regiões superpostas.

Um ***"perceptron"*** **de três camadas** pode formar regiões de decisão arbitrariamente complexas, não convexas e inconexas e pode separar classes com regiões superpostas. Isto pode ser provado por construção. Podemos supor que cada M neurônios da primeira camada forma uma região convexa elementar e que cada neurônio da segunda camada ao qual estão conectadas forma uma AND dessa região. Logo se cada região conexa não CONVEXA for aproximada por N regiões elementares e cada região não conexa for aproximada por O regiões elementares precisaremos uma camada a mais para implementar uma função OR de todas as saídas da segunda camada.

É importante destacar que as conclusões obtidas foram baseadas em "perceptrons" com função de transferência não linear de tipo escalão. Exibem um comportamento similar com os "perceptrons" que utilizam outras funções não lineares. Quando se utilizam funções de transferência não linear de tipo sigmóide a característica principal das fronteiras de decisão entre as classes é que são limitadas por curvas suaves em vez de linhas retas. Porém o análise da capacidade de representação destas redes se torna muito dificil.

2.5.4- Redes neurais e aproximadores não lineares.

Aprofundaremos a seguir o estudo da rede neural como um aproximador não linear, com ênfase na função de transferência.

Já destacamos que o papel da função não linear na camada de saída é diferente quando está na camada intermediária. Na camada de saída simula a aproximação de uma função lógica entre -1 e 1 e portanto deve aproximar uma função limiar. Na camada intermediária sua função é a de gerar ***momentos de ordem superior*** do vetor de entrada. Se explica esta propriedade a partir de considerar a função não linear por sua aproximação em série.

Para detalhar esta propriedade trabalharemos com uma rede neural de entradas binárias. Utilizaremos a seguinte propriedade: se o vetor de entrada **X** tem componentes x(i) binárias, onde i=1:N, então $(x(i))^k = x(i)\ \forall k>0$.

A função de transferência sigmóide se denota através da seguinte equação:

$$F(s) = \frac{1}{1+\exp(-s)} \tag{2.77}$$

onde $s=W^T X$, onde ^{T}denota transposição. W são os pesos sinápticos e X é o vetor de entrada. Então podemos reescrever a equação 2.77 calculando a saída do neurônio j, em função da expansão matricial:

$$F_j(s) = \frac{1}{1+\exp(-\sum_{i=0}^{N} w_{ji} x_i)} \tag{2.78}$$

usando a expansão de Taylor na equação 2.78:

$$F_j(s) = \sum_{k=0}^{\infty} a_k \left(\sum_{i=0}^{N} w_{ji} x \right)^k \tag{2.79}$$

utilizando a propriedade do vetor binário podemos expressar a equação 2.79 como:

$$F_j(s) = \gamma_0 + \sum_n \gamma_n i_n + \sum_n \sum_m \gamma_{nm} i_n i_m + \ldots \tag{2.80}$$

Então para cada componente j do conjunto de neurônios de saída, a função não linear gera combinações lineares de todas as 2^N possíveis combinações dos produtos cruzados das N entradas binárias, isto é gera pares, triplos, ... e n-tuplas. O coeficiente γ de cada produto cruzado depende da matriz de pesos W a qual é treinada na forma de selecionar os produtos cruzados relevantes, isto é, os que são ***típicos dos padrões de treinamento e insensíveis ao ruído***. Na etapa de reconhecimento, uma entrada de prova excitaria alguns produtos cruzados e a decisão final se derivará desta função total. Se o padrão de entrada contém ruído, alguns produtos cruzados que deveriam ser excitados não o serão e vice-versa. Porém se os dados de prova são consistentes com os da base de dados é de se esperar que os produtos cruzados excitados e não excitados não sejam relevantes. Conseqüentemente, uma correta decisão podería ocorrer através desta classificação discriminante.

2.5.5- Prova de aproximação universal

Existe um teorema de Kolmogorov-Sprecher onde se demonstra que uma estrutura não linear do tipo rede neural em camadas pode aproximar quaisquer função real contínua.

Teorema de Kolmogorov-Sprecher: Para cada número inteiro n≥2, existe uma função real e monotónicamente crescente $\psi(x)$, $\psi([0,1])=[0,1]$, dependente de n e com a seguinte propriedade:

Para cada número $\delta>0$ dado, há um número racional ε, $0<\varepsilon<\delta$, tal que quaisquer função real continúa de n variáveis, $\phi(\mathbf{x})$, definida em I^n, possa ser representada exatamente por

$$\phi(\mathbf{x}) = \sum_{j=1}^{2n+1} \chi[\sum_{i=1}^{n} \lambda i \psi(x_i + \varepsilon(j-1)) + j - 1] \qquad (2.81)$$

onde χ é uma função real e contínua deinclinação de ϕ, e λ é uma constante independente de ϕ. ■

Se observamos detidamente esta expansão não linear é uma equação que representa um "perceptron" multicamada, onde os fatores de aproximação são os pesos da rede. Este teorema fornece um fundamento teórico essencial para a aplicação das redes neurais à aproximação de funções de classificação de padrões.

Agora, o teorema demonstra o que a rede neural multicamada pode fazer mas não revela como fazê-lo. Tampouco revela a quantidade necessaria de coeficientes em cada uma das camadas. Então é um problema sumamente importante encontrar um mecanismo ótimo para calcular estes coeficientes, na forma de lograr a aproximação desejada.

CAPITULO 3

FILTRO DE KALMAN

3.1.- TEORIA DAS INOVAÇÕES

3.1.1- Introdução

Neste capítulo formalizaremos primeiro a teoria das inovações, que é uma ferramenta básica para a compreensão do filtro de Kalman. Depois apresentaremos brevemente a teoria de modelização por espaço de estados e a teoria do filtro de Kalman. Provaremos suas condições de optimalidade. Introduziremos posteriormente uma modificação do filtro de Kalman para casos onde o estado a estimar mude com o tempo. Finalmente apresentaremos exemplos de aplicação do filtro de Kalman normal e de memória finita.

Dado um processo estocástico discreto e estacionário, sem componentes periódicas, podemos produzir um processo associado chamado processo de inovações pela aplicação de uma transformação linear que é causal e causalmente invertível. O requerimento de causalidade implica que a transformação linear pode ser implementada em tempo real. O requerimento de invertibilidade implica que o processo estocástico original é recuperável de sua versão transformada que é o processo de inovações. A característica que faz à representação de um processo estocástico por inovações uma ferramenta analítica muito poderosa é o fato de que o processo de inovações é um processo muito mais simples de trabalhar que o processo estocástico original. O importante é que ambos processos possuem a mesma informação estatística. Isto implica que não há perda de informação como resultado da transformação.

3.1.2- Definição

Dado um processo estocástico $\{x(n)\}$, sem componentes periódicas, definimos um **processo de inovações** como $\{\nu(n)\}$ um ***processo ruído branco***, tal que o processo $\{x(n)\}$ pode ser determinado do processo $\{\nu(n)\}$ por uma transformação causal e causalmente invertível. A causalidade define-se como a anulação da resposta ao impulso para tempos negativos, tomando como zero o momento da aplicação do impulso. Um ruído branco é definido, de acordo com a teoria dos processos estocásticos, como um processo onde as amostras de dois tempos diferentes estão descorrelacionadas. Uma transformação é causalmente invertível quando é possível construir outro filtro cuja resposta em freqüência é

igual à inversa da resposta em freqüência do filtro original, e este segundo filtro é causal. A exclusão de componentes periódicas implica que a transferência no plano z do sistema gerador não tem pólos no círculo unitário. Esta é uma condição que satisfaz a todos os sinais reais.

Seja {h(n)} a resposta ao impulso de um filtro usado para transformar o processo estocástico {x(n)} no processo de inovações {ν(n)}. Para que o filtro seja causal requere-se que:

$$h(n) = 0; \quad n<0 \tag{3.1}$$

Seja $H(e^{jw})$ a resposta em freqüência do filtro, definido como a transformada discreta de Fourier da resposta ao impulso {h(n)}. Para que o filtro transforme o sinal de entrada {x(n)} de densidade espectral de potência $S_X(w)$ em ruído branco, requere-se que:

$$\left|H(e^{jw})\right|^2 = \frac{\sigma_\nu^2}{S_X(w)} \tag{3.2}$$

Devido a que o filtro recebe um processo estocástico qualquer e fornece como saída um processo estocástico do tipo ruído branco, este filtro é chamado "***filtro branqueador***" ou "***filtro de análise***".

Seja {g(n)} a resposta ao impulso de um filtro usado para reconstruir o processo estocástico original {x(n)} a partir do processo de inovações {ν(n)}. Para que este segundo filtro seja causal requere-se que:

$$g(n) = 0; \quad n<0 \tag{3.3}$$

Seja $G(e^{jw})$ a resposta em freqüência do segundo filtro. Para que a saída do filtro, produzido em resposta à entrada {ν(n)}, seja equivalente ao processo original {x(n)} até segundo ordem, requere-se que a saída do filtro e {x(n)} tenham a mesma densidade espectral de potência, isto implica:

$$S_X(w) = \sigma_\nu^2 \left|G(e^{jw})\right|^2 \tag{3.4}$$

As equações (3.2) e (3.4) são satisfeitas se a função de transferência dos dois filtros estão relacionadas por:

$$\left|G(e^{jw})\right| = \frac{1}{\left|H(e^{jw})\right|} \tag{3.5}$$

Existem infinitas transformações lineares que satisfazem estas relações. Isto deve-se a que somente se exige uma relação de amplitudes e não de fases. Esta é uma característica típica do processamento por espectro de potência, que por ser de segundo ordem apresenta somente amplitudes e não fases.

De acordo com isto, chamaremos o segundo filtro como o ***"filtro inverso"*** *ou* ***"filtro de síntese"***. Desta forma a equação (3.1) certifica que a transformação de {x(n)} a {v(n)} seja causal e as equações (3.3) e (3.5) certificam que a transformação seja causalmente invertível.

Na figura seguinte apresentam-se ambos filtros:

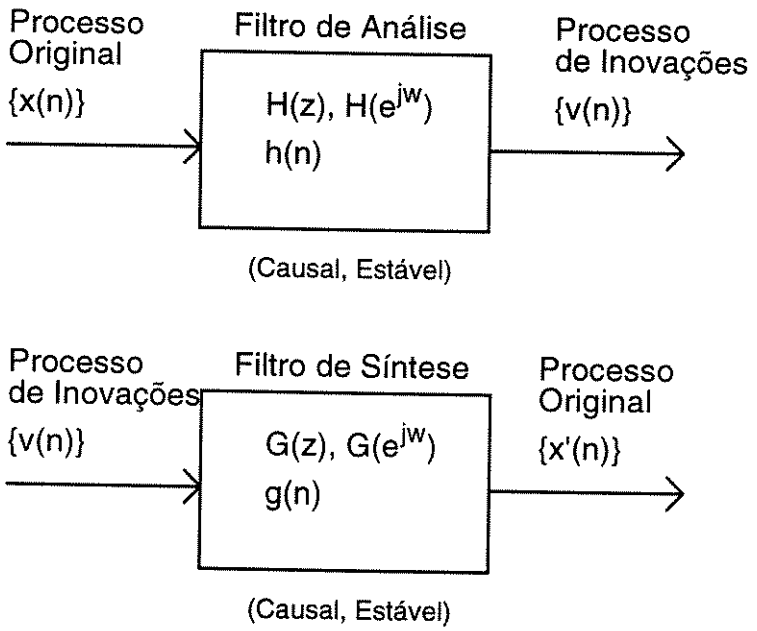


FIGURA 3.1 - Filtros branqueador e inverso ou de análise e síntese.

É importante destacar que o processo original de saída do filtro de síntese é outra realização do processo estocástico original, e que não será igual ponto a ponto com a seqüência original, salvo no caso que o processo de inovações de saída do filtro branqueador seja exatamente o mesmo que alimenta ao filtro inverso. Por esta razão denota-se como {x'(n)} a saída do filtro de síntese.

As características fundamentais desta transformação são:

1) O processo estocástico $\{x(n)\}$ e o processo de inovações $\{v(n)\}$ tem a mesma "informação estatística" já que podemos ir de um processo ao outro em tempo real.

2) O processo de inovações $\{v(n)\}$ é um processo estatístico muito mais simples de trabalhar que o processo estocástico original $\{x(n)\}$. As amostras de $\{v(n)\}$ em diferentes instantes estão descorrelacionadas enquanto que as amostras $\{x(n)\}$, em geral, estão correlacionadas.

Um conceito fundamental desta teoria, que a transforma em suporte de muitas teorias do processamento de sinais, particularmente da estimação espectral, é que a informação redundante do sinal de entrada se armazena nos coeficientes da transformação linear, já o ruído branco de saída não leva informação por ter correlação nula entre as amostras de diferentes tempos.

3.1.3 - Representação canônica

Seja um processo estocástico $\{x(n)\}$ estacionário no sentido amplo, sem componentes periódicas e seja $S_X(w)$ sua densidade espectral de potência, que deve ser uma função contínua da freqüência angular w. Seja $S_X(z)$ a transformada z da função de autocorrelação do processo denotada por $\{r_X(n)\}$.

Supondo que a função $S_X(z)$ e que $\ln(S_X(z))$ são analíticas, e com derivadas contínuas, na região do plano z definida por $\rho<|z|<1/\rho$, onde $0<\rho<1$. Isto implica que $S_X(z)$ não tem pólos nem zeros nesta região.

Demonstra-se (ver [19]) que $S_X(z)$ pode ser fatorizado da seguinte forma:

$$S_X(z) = \sigma_v^2 \, G(z) \, G(1/z) \tag{3.6}$$

onde:

$$\sigma_v^2 = e^{c0} \tag{3.7}$$

$$G(z) = \exp\left(\sum_{k=1}^{\infty} c_k z^{-k} \right), \quad \rho<|z| \tag{3.8}$$

$$G(1/z) = \exp\left(\sum_{k=-\infty}^{-1} c_k z^{-k} \right), \quad |z|<1/\rho \tag{3.9}$$

com c_k como os coeficientes de Fourier da função $\ln(S_X(z))$, para $k=]-\infty;+\infty[$.

Para pontos do círculo unitário, onde $z=e^{jw}$, podemos reescrever a equação (3.6), da seguinte forma:

$$S_X(w)=\sigma_V^2\left|G(e^{jw})\right|^2 \tag{3.10}$$

A equação (3.10) é a chamada ***fatorização canônica da densidade espectral de potência***.

Tanto G(z) como ln(G(z)) são analíticas em $\rho<|z|$, com $\rho<1$, o que implica que não tem pólos nem zeros nesta região, então este é um filtro de ***fase mínima***. Isto implica então que G(n) é causal e estável. Como os zeros de G(z) são os pólos de 1/G(z), então 1/G(z) também é de fase mínima.

Tanto H(1/z) como ln(H(1/z)) são analíticas em $|z|<1/\rho$, com $\rho<1$, o que implica que não tem pólos nem zeros nesta região, Então este é um filtro ***anti-causal***.

3.1.4 - Espectros de potência racional

3.1.4.1 - Modelo AR

Considere um processo estocástico {x(n)} para o qual a função $S_X(z)$ consiste somente de pólos:

$$S_X(z)=\frac{\sigma^2}{\prod_{i=1}^{M}(1-\alpha_i z^{-1})(1-\alpha_i^* z)} \tag{3.11}$$

onde $|\alpha_i|<1$ para todo i. Correspondentemente a função de transferência do filtro inverso definida pela função G(z) consiste somente de pólos (já que este é o filtro gerador do processo):

$$G(z)=\frac{1}{\prod_{i=1}^{M}(1-\alpha_i z^{-1})} \tag{3.12}$$

Expressando o polinômio denominador em forma expandida:

$$G(z) = \frac{1}{1 + \sum_{i=1}^{M} a_i z^{-i}} \tag{3.13}$$

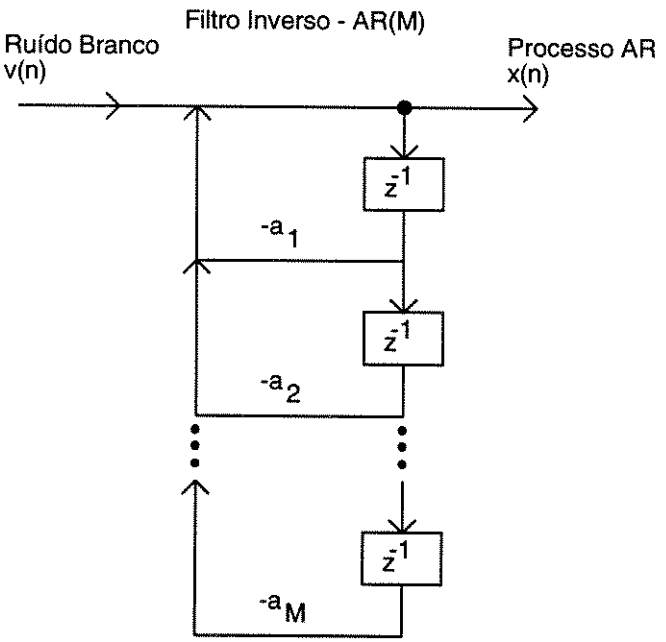
onde os coeficientes a_i estão unívocamente relacionados com os pólos de G(z), isto é, os coeficientes α_i. Isto implica que a resposta ao impulso do filtro inverso {g(n)} tem duração infinita. Na literatura este tipo de filtro denomina-se **IIR** (Infinite Impulse Response). Pela classe de transferência este filtro é do tipo somente pólos.

Como H(z)=1/G(z), a função de transferência do filtro branqueador será:

$$H(z) = 1 + \sum_{i=1}^{M} a_i z^{-i} \tag{3.14}$$

Então a resposta ao impulso do filtro branqueador é de duração finita.

Nas seguintes figuras apresentam-se as estruturas dos filtros inverso e branqueador de um processo AR.



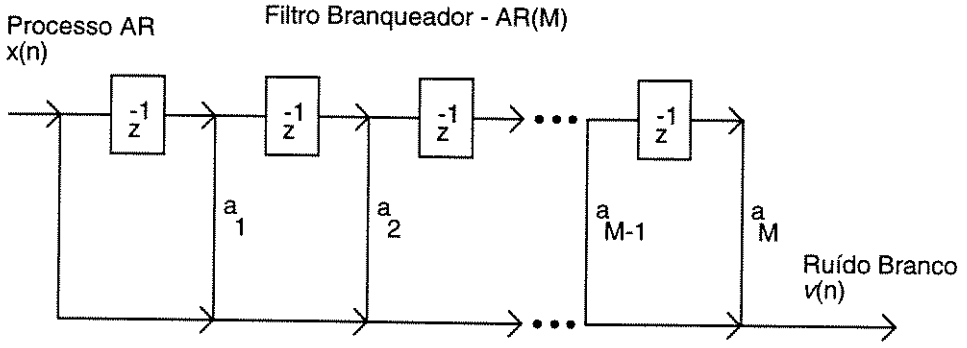


FIGURA 3.2 - Estruturas dos filtros inverso e branqueador de um processo AR.

No domínio do tempo a equação a diferenças que representa a ambos filtros é:

$$x(n) + a_1x(n-1) + a_2x(n-2) + \ldots a_Mx(n-M) = v(n) \tag{3.15}$$

ou também:

$$x(n) = -a_1x(n-1) + -a_2x(n-2) + \ldots -a_Mx(n-M) + v(n) \tag{3.16}$$

Um processo deste tipo é denominado **processo autorregresivo (AR)** de ordem M. Este nome deve-se ao fato que a saída atual do filtro $v(n)$ é composta por valores de entrada x(n) em diferentes tempos, isto é, uma combinação linear de valores regressivos de x(n). Os coeficientes a_i são denominados coeficientes autorregresivos do processo x(n).

3.1.4.2 - Modelo MA e ARMA

Considere um processo estocástico {x(n)} para o qual a função $S_X(z)$ consiste somente de zeros:

$$S_X(z) = \sigma^2 \prod_{i=1}^{N} (1-\beta_i z^{-1})(1-\beta_i^* z) \tag{3.17}$$

onde $|\beta_i| < 1$ para todo i. Correspondentemente a função G(z) que define a função de transferência do filtro inverso consiste somente de zeros:

$$G(z) = \prod_{i=1}^{N} (1-\beta_i z^{-1}) \tag{3.18}$$

Expressando o polinômio denominador em forma expandida:

$$G(z) = 1 + \sum_{i=1}^{N} b_i z^{-i} \tag{3.19}$$

onde os coeficientes b_i estão unívocamente relacionados com os zeros de G(z), isto é, os coeficientes β_i. Isto implica que a resposta ao impulso do filtro inverso $\{g(n)\}$ tem duração finita. Na literatura este tipo de filtro denomina-se **FIR** (Finite Impulse Response). Pelo tipo de transferência este filtro é do tipo somente zeros.

Como H(z)=1/G(z), a função de transferência do filtro branqueador será:

$$H(z) = \frac{1}{1 + \sum_{i=1}^{N} b_i z^{-i}} \tag{3.20}$$

Então a resposta ao impulso do filtro branqueador é de duração infinita.

No domínio do tempo a equação a diferenças que representa a ambos filtros é:

$$v(n) + b_1 v(n-1) + b_2 v(n-2) + \ldots b_N v(n-N) = x(n) \tag{3.21}$$

Um processo deste tipo é denominado **processo de média móvel (MA** por **moving average)** de ordem N. Este nome deve-se a que a saída atual do filtro x(n) é uma combinação linear de valores presentes e passados do processo de inovações $v(n)$. Os coeficientes b_i são denominados coeficientes de média móveis do processo x(n).

Na literatura este tipo de filtro denomina-se **FIR** (Finite Impulse Response). Pelo tipo de transferência este filtro é do tipo somente zeros.

Também poderíamos definir um processo chamado **processo autorregresivo-média móvel (ARMA)** de ordem (M,N). Sua equação a diferenças no domínio do tempo será:

$$x(n) + \sum_{i=1}^{M} a_i x(n-i) = v(n) + \sum_{i=1}^{N} b_i v(n-i) \tag{3.22}$$

A função de transferência deste sistema consiste de pólos e de zeros.

3.2- O FILTRO DE KALMAN

3.2.1- Introdução

O filtro de Wiener assume que os processos estocásticos que representam a entrada e a saída são mutuamente estacionários. Esta suposição limita a utilidade do filtro de Wiener. O filtro de Kalman supera esta limitação e fornece a solução a uma classe de problemas de estimação recursiva que minimizam o erro quadrático médio. O filtro de Kalman inclui ao filtro de Wiener como caso especial.

O filtro de Kalman é formulado usando a aproximação de espaço de estados no qual um sistema dinâmico é descrito por um conjunto de variáveis que representa o estado. Este conjunto de variáveis contém toda a informação necessária acerca do comportamento do sistema, de forma tal que dados os valores presentes e passados da entrada e do sistema podemos computar a saída e o estado futuro do mesmo. A principal função do filtro de Kalman é fornecer uma estimação do estado de um sistema.

O filtro de Kalman é ideal para sua implementação em computador devido a que seu esquema recursivo implica somente o armazenamento do último valor do estado.

3.2.2 - Equações de estado

Seja um vetor M-dimensional X(n) o *estado* de um sistema dinâmico linear, de tempo discreto, seja um vetor N-dimensional Y(n) a *saída observada* do sistema, ambos medidos no tempo n. Estes vetores são do tipo variáveis aleatórias vetoriais. O sistema *modelo* pode ser descrito por duas equações:

1) Equação de Processo

$$X(n+1) = \Phi(n+1,n)X(n) + v(n) \tag{3.23}$$

onde $\Phi(n+1,n)$ é a *matriz de transição de estados* de tamanho M por M. O vetor M por 1 $\mathbf{v}(n)$ é o *ruído do processo*. A equação de processo modela a evolução ruidosa do estado. O vetor $\mathbf{v}(n)$ é modelado como um processo de ruído branco de valor médio nulo, cuja matriz de correlação é definida como :

$$\mathbf{E}[\mathbf{v}(n)\mathbf{v}^T(k)] = \begin{cases} \mathbf{Q}(n) & k = n \\ 0 & k \neq n \end{cases} \tag{3.24}$$

2) Equação de Medição

$$Y(n) = C(n)X(n) + w(n) \tag{3.25}$$

onde $\mathbf{C}(n)$ é a *matriz de observação* de tamanho N por M. O vetor M por 1 $\mathbf{w}(n)$ é o *ruído de medição*. A equação de medição modela a dependência ruidosa da observação respeito do estado. O vetor $\mathbf{w}(n)$ é modelado como um processo de ruído branco de valor médio nulo, cuja matriz de correlação é definida como :

$$\mathbf{E}[\mathbf{w}(n)\mathbf{w}^T(k)] = \begin{cases} \mathbf{R}(n) & k = n \\ 0 & k \neq n \end{cases} \tag{3.26}$$

A seguir apresenta-se um diagrama de fluxo do sistema modelo:

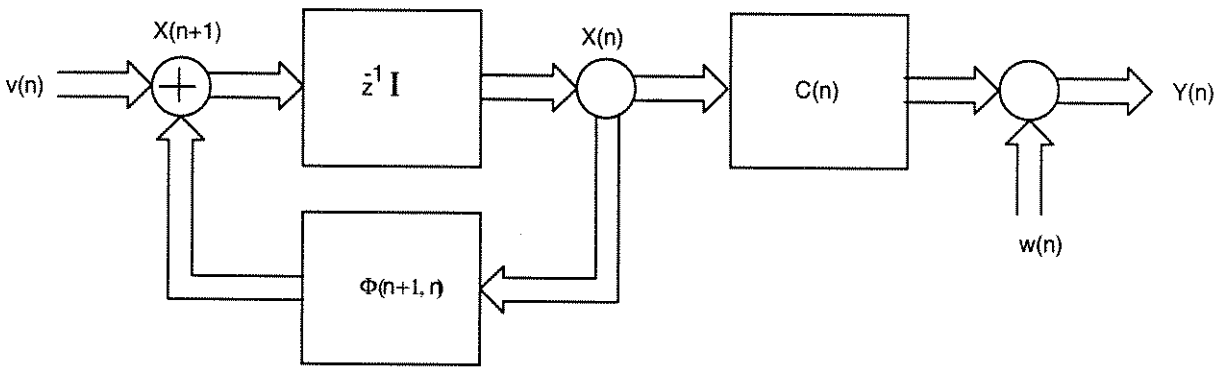


FIGURA 3.3 - Diagrama de fluxo do sistema modelo

Os vetores de ruído v(n) e w(n) são estatisticamente independendentes, isto é:

$$\mathbf{E}[\mathbf{v}(n)\mathbf{w}^T(k)] = 0 \qquad \forall k, n \tag{3.27}$$

A matriz de transição de estados $\Phi(n+1,n)$ e a matriz de medição $\mathbf{C}(n)$ são conhecidas, assim como também são conhecidos os valores da observação Y(1), Y(2), ...,Y(n). O problema a resolver é encontrar para cada tempo $n \geq 1$ a estimação mínima no sentido do erro quadrático médio de X(i). Se i=n, chama-se *filtragem*, se i>n chama-se *predição* e chama-se *alisado* se $1 \leq i < n$.

3.2.3 - O Processo de Inovações referido ao problema da filtragem

Seja o vetor $\hat{Y}(n)$ a estimação do vetor de observações Y(n) que minimiza o erro quadrático médio, dados os valores passados do vetor de observação começando de n=1, até n-1. Estes valores passados são representados por Y(1), Y(2), ...,Y(n-1) que conformam o espaço vetorial $\mathbf{Y_{n-1}}$. Definiremos o processo de inovações como:

$$\alpha(n) = Y(n) - \hat{Y}(n) \qquad n = 1,2,... \tag{3.28}$$

Para o propósito de estimar o estado X(n), a informação contida nos dados observados até, e incluindo o tempo n-1, está totalmente incluída na estimação predita $\hat{Y}(n)$. Isto significa que o vetor M por 1 $\alpha(n)$ representa a nova informação no vetor de observação Y(n), por isso o nome do processo.

O processo de inovação tem as seguintes propriedades:

1) O processo de inovações $\alpha(n)$ é ortogonal a todas as observações passadas Y(1), Y(2), ...,Y(n-1), tal que:

$$\mathbf{E}[\alpha(n)y^{T}(k)] = 0 \qquad 1 \leq k \leq n-1 \tag{3.29}$$

Isto é simplesmente uma generalização do princípio de ortogonalidade, que implica que o erro de estimação é ortogonal às entradas usadas para construir a estimação que minimiza o erro quadrático médio.

2) O processo de inovações consiste em uma seqüência de variáveis aleatórias vetoriais que são ortogonais entre si, tal que:

$$\mathbf{E}[\alpha(n)\alpha^{T}(k)] = 0 \qquad 1 \leq k \leq n-1 \tag{3.30}$$

Esta propriedade, que segue à propriedade 1, é a uma generalização do corolário do princípio de ortogonalidade que diz que o erro de estimação é ortogonal à estimação que minimiza o erro quadrático médio.

3) Há uma correspondência um a um entre a seqüência de dados observados {E(k), k=1,2,...} e o processo de inovações {$\alpha(k)$, k=1,2,...}. Em particular cada uma das seqüencias pode ser obtida a partir da outra por uma transformação linear sem perda de informação.

3.2.4 - Equação geral de estimadores recursivos

O objetivo é obter uma estimação $\hat{X}(n)$ do estado X(n) no tempo n, a partir da informação das amostras desde o tempo 1, isto é desde o começo da amostragem. Esta estimação será obtida através de uma combinação linear dos erros de estimação ou da seqüência de inovações. Devido à propriedade de ortogonalidade entre os mesmos, a combinação linear é ótima já que esta referida a um conjunto de vetores ortogonais.

Desenvolveremos a seguir a equação geral dos estimadores recursivos; primeiro para o caso escalar e depois para o caso matricial. Expressaremos a combinação linear cujos coeficientes serão as incógnitas a encontrar:

$$\hat{X}(n) = \sum_{i=1}^{n} a(i)\alpha(i) \tag{3.31}$$

A fórmula para a eleição dos coeficientes da combinação linear será desenvolvida a partir da minimização do seguinte critério de erro quadrático médio:

$$J = E[X(n) - \hat{X}(n)]^2 = E[X(n) - \sum_{i=1}^{n} a(i)\alpha(i)]^2 \tag{3.32}$$

Por ser este critério quadrático nos coeficientes a(i) podemos encontrar o mínimo simplesmente computando a derivada da função respeito aos parâmetros e igualando a zero.

$$\frac{\partial J}{\partial a(n)} = 0 \Rightarrow -2E[X(n) - \sum_{i=1}^{n} a(i)\alpha(i)]\alpha(j) = 0 \tag{3.33}$$

levando em conta que $E[\alpha(i)\alpha(j)] = 0;\ \forall i \neq j \Rightarrow E[X(n)\alpha(i)] - a(i)E[\alpha(i)\alpha(i)] = 0$; a expressão dos coeficientes é:

$$a(i) = E[X(n)\alpha(i)] \Big/ E[\alpha(i)\alpha(i)] \tag{3.34}$$

Então introduzindo a equação (3.34) na (3.31), obtemos:

$$\hat{X}(n) = \sum_{i=1}^{n} E[X(n)\alpha(i)] \left[E[\alpha(i)\alpha(i)]\right]^{-1} \alpha(n) \tag{3.35}$$

A equação (3.35) é para o caso escalar, a seguir a apresentamos para o caso matricial:

$$\hat{X}(n) = \sum_{i=1}^{n} E[X(n)\alpha(i)^T]\left[E[\alpha(i)\alpha(i)^T]\right]^{-1}\alpha(n) \tag{3.36}$$

Se decompomos a equação (3.36) entre os tempos 1 até k-1 e outro término até k, temos:

$$\hat{X}(n) = \hat{X}(n-1) + E[X(n)\alpha(n)^T]\left[E[\alpha(n)\alpha(n)^T]\right]^{-1}\alpha(n) \tag{3.37}$$

A equação (3.37) é a chamada equação básica dos estimadores recursivos.

3.2.5 - Formulação do filtro de Kalman

Desenvolveremos a seguir o filtro de Kalman, levando em consideração as hipóteses formuladas ao respeito do ruído de entrada e saída nas equações (3.24), (3.26) e (3.27).

A matriz de autocorrelação do processo de inovação será:

$$\Sigma(n) = E[\alpha(n)\alpha(n)^T] \tag{3.38}$$

Definiremos a seguir uma nova variável como a diferença entre o valor verdadeiro e o estimado, que será o *erro de estimação de estado*:

$$\tilde{X}(n) = X(n) - \hat{X}(n) \tag{3.39}$$

Tomando as equações (3.28) e a equação de medição (3.25) temos:

$$\alpha(n) = Y(n) - \hat{Y}(n) = Y(n) - C(n)\hat{X}(n) \tag{3.40}$$

Aplicando a equação de medição (3.25) à equação (3.40) e utilizando a equação (3.39), temos:

$$\begin{aligned} &= C(n)(X(n) - \hat{X}(n)) + w(n) \\ &= C(n)\tilde{X}(n) + w(n) \end{aligned} \tag{3.41}$$

Se calculamos agora a expressão da matriz de autocorrelação do processo de inovações levando em conta as equações (3.38), (3.24), (3.29) e (3.41):

$$\Sigma(n) = E[\alpha(n)\alpha(n)^T] = C(n)E[\tilde{X}(n)\tilde{X}^T(n)]C^T(n) + R(n) \tag{3.42}$$

Definimos a *matriz de autocorrelação do erro de estimação de estado* como:

$$P(n) = E[\tilde{X}(n)\tilde{X}^T(n)] \tag{3.43}$$

Introduzindo a equação (3.43) na equação (3.42) obtemos:

$$\Sigma(n) = C(n)P(n)C^T(n) + R(n) \tag{3.44}$$

3.2.6 - Predição de um passo

O próximo passo que desenvolveremos será a predição que minimiza o erro quadrático médio do estado atual $\mathbf{X}(n)$ dadas as observações Y(1), Y(2) até o tempo Y(n). Como há uma correspondência unívoca entre os vetores de observação $\mathbf{Y}$ e o processo de inovações α, podemos expressar esta estimação como:

$$\hat{X}(i / Y(n)) = \sum_{k=1}^{n} A_i(k)\alpha(k) \tag{3.45}$$

onde o conjunto de matrizes $\{\mathbf{A}_i(k)\}$ são matrizes a serem determinadas.

Partindo da definição do princípio de ortogonalidade temos que o erro de estimação é ortogonal ao processo de inovações. Isto é:

$$E[\tilde{X}(i / Y(n))\alpha^T(m)] = 0, \quad 1 \leq m \leq n \tag{3.46}$$

Substituindo pela definição do erro de estimação de estado em (3.46), obtemos:

$$E[(X(i) - \hat{X}(i / Y(n))\alpha^T(m)] = 0, \quad 1 \leq m \leq n \tag{3.47}$$

Desenvolvendo a equação (3.47):

$$E[X(i)\alpha^T(m)] = E[\hat{X}(i / Y(n))\alpha^T(m)], \quad 1 \leq m \leq n \tag{3.48}$$

Substituindo em (3.48) a equação (3.45),

$$\begin{aligned} E[X(i)\alpha^T(m)] &= E[\sum_{k=1}^{n} A_i(k)\alpha(k)\alpha^T(m)] \\ &= \sum_{k=1}^{n} A_i(k)E[\alpha(k)\alpha^T(m)] \\ &= A_i(m)E[\alpha(m)\alpha^T(m)] \\ &= A_i(m)\Sigma(m) \end{aligned} \tag{3.49}$$

onde em (3.49) utilizamos primeiro a propriedade que a matriz **A** é constante e pode sair do operador esperança. Finalmente utilizamos a definição da matriz de autocorrelação do processo de inovações.

Então se despejamos a matriz **A** da última fila da equação (3.49), pós-multiplicando pela inversa da matriz de autocorrelação do processo de inovações $\Sigma(m)$:

$$A_i(m) = E[X(i)\alpha^T(m)]\Sigma^{-1}(m) \tag{3.50}$$

Substituindo a equação (3.50) na equação (3.45)

$$\hat{X}(i/Y(n)) = \sum_{k=1}^{n} E[X(i)\alpha^T(k)]\boldsymbol{\Sigma}^{-1}(k)\alpha(k) \tag{3.51}$$

Dado que o objetivo é que o estimador seja recursivo, vamos separar a equação (3.51) numa parte até o tempo n-1 e outra no tempo n. Para isso primeiro substituiremos o índice i por n+1:

$$\hat{X}(n+1/Y(n)) = \sum_{k=1}^{n} E[X(n+1)\alpha^T(k)]\boldsymbol{\Sigma}^{-1}(k)\alpha(k) \tag{3.52}$$

Logo desenvolveremos os térmos dentro da somatória:

$$\begin{aligned} E[X(n+1)\alpha^T(k)] &= \Phi(n+1,n)E[X(n)\alpha^T(k)] + E[v(n)\alpha^T(k)] \\ &= \Phi(n+1,n)E[X(n)\alpha^T(k)], \qquad 0 \subseteq k \subseteq n \end{aligned} \tag{3.53}$$

onde na equação (3.53) fizemos uso do fato que o processo de inovações e o ruído de observações são decorrelacionados. Então se utilizamos a equação (3.53) na equação (3.52) obtemos:

$$\hat{X}(n+1/Y(n)) = \Phi(n+1,n)\sum_{k=1}^{n} E[X(n)\alpha^T(k)]\Sigma^{-1}(k)\alpha(k) \tag{3.54}$$

Então desenvolveremos a equação (3.54) separando os térmos até n-1 dos que dependem de n:

$$\begin{aligned}\hat{X}(n+1/Y(n)) = \Phi(n+1,n)\sum_{k=1}^{n-1} E[X(n)\alpha^T(k)]\Sigma^{-1}(k)\alpha(k) \\ + \Phi(n+1,n)E[X(n)\alpha^T(n)]\Sigma^{-1}(n)\alpha(n)\end{aligned} \tag{3.55}$$

onde na equação (3.55), o primeiro térmo é identificado de acordo à equação (3.51):

$$\hat{X}(n/Y(n-1)) = \sum_{k=1}^{n-1} E[X(n)\alpha^T(k)]\Sigma^{-1}(k)\alpha(k) \tag{3.56}$$

Potanto substituindo a equação (3.56) na (3.55) e definindo a matriz **G**(n) de ganho do preditor:

$$G(n) = \Phi(n+1,n)E[X(n)\alpha^T(n)]\Sigma^{-1}(n) \tag{3.57}$$

Finalmente podemos escrever a equação (3.55) como:

$$\hat{X}(n+1/Y(n)) = \Phi(n+1,n)\hat{X}(n/Y(n-1)) + G(n)\alpha(n) \tag{3.58}$$

A equação (3.58) revela que podemos computar a predição do estado no tempo n, somando-se à predição no tempo n-1, pré-multiplicada pela matriz de transição, um térmo de correção igual ao processo de inovações pré-multiplicado por uma matriz de ganho.

Matriz de Riccati:

A equação (3.57) apresenta a matriz de ganho de predição, porém devemos modificar esta fórmula para que seja conveniente para a computação. Para começar desenvolveremos a correlação $X(n)\alpha^T(n)$, utilizando a equação (3.41):

$$\begin{aligned}E[X(n)\alpha^T(n)] &= E[X(n)\tilde{X}^T(n)]C^T(n) + E[X(n)w^T(n)] \\ &= E[X(n)\tilde{X}^T(n)]C^T(n)\end{aligned} \tag{3.59}$$

onde na equação (3.59) utilizamos o fato que por definição, a correlação entre o estado e o ruído de observação é nulo. Além disso pela definição do princípio de ortogonalidade o erro de estimação é ortogonal à predição, isto é:

$$E[\hat{X}(n/Y(n-1))\tilde{X}^T(n)]=0 \tag{3.60}$$

Então podemos usar a equação (3.39) e a (3.60), para modificar a equação (3.43):

$$\begin{aligned} P(n) &= E[X(n)\tilde{X}^T(n)] - E[\hat{X}(n)\tilde{X}^T(n)] \\ &= E[X(n)\tilde{X}^T(n)] \end{aligned} \tag{3.61}$$

com o resultado da equação (3.61), podemos simplificar a equação (3.59):

$$E[X(n)\alpha^T(n)]=P(n)C^T(n) \tag{3.62}$$

Logo a equação da matriz de ganho de predição (3.57) fica:

$$G(n)=\Phi(n+1,n)P(n)C^T(n)\Sigma^{-1}(n) \tag{3.63}$$

A fórmula (3.63) requer que seja conhecida a matriz P(n), de forma que deve-se encontrar uma expresão para seu cômputo recursivo. Extrairemos esta fórmula a partir de sua definição:

$$\tilde{X}(n+1)=X(n+1)-\hat{X}(n+1/Y(n)) \tag{3.64}$$

substituindo na equação (3.64) a definição da equação de estado e a equação (3.58)

$$\begin{aligned} \tilde{X}(n+1) &= \Phi(n+1,n)[X(n)-\hat{X}(n/Y(n-1))]+v(n) \\ &\quad -G(n)[C(n)\tilde{X}(n)+w(n)] \end{aligned} \tag{3.65}$$

reorganizando os térmos da equação (3.65) obtemos:

$$\begin{aligned} \tilde{X}(n+1) &= [\Phi(n+1,n)-G(n)C(n)]\tilde{X}(n) \\ &\quad +v(n)-G(n)w(n) \end{aligned} \tag{3.66}$$

Como a definição da matriz de autocorrelação do erro de estimação de estado é:

$$P(n+1)=E[\tilde{X}(n+1)\tilde{X}^T(n+1)] \tag{3.67}$$

Então substituíndo a equação (3.66) na equação (3.67) e reorganizando os térmos levando em conta que os vetores de ruído são mutuamente ortogonais, teremos finalmente a equação recursiva da matriz de estimação do erro de estado, como:

$$\begin{aligned} P(n+1)=&[\Phi(n+1,n)-G(n)C(n)]P(n)[\Phi(n+1,n)-G(n)C(n)]^T \\ &+Q(n)+G(n)R(n)G^T(n) \end{aligned} \tag{3.68}$$

onde Q(n) e R(n) são as matrizes de autocorrelação de v(n) e w(n), respectivamente. Podemos reorganizar a parte direita da equação (3.68), de forma que seja mais tratável para sua computação. Desenvolveremos a equação da matriz de ganho e utilizaremos a propriedade que uma matriz multiplicada por sua transposta da como resultado a matriz identidade.

$$\begin{aligned} P^+(n)&=P(n-1)-\Phi(n,n-1)G(n)C(n)P(n-1) \\ P(n+1)&=\Phi(n+1,n)P^+(n)\Phi^T(n+1,n)+Q(n) \end{aligned} \tag{3.69}$$

portanto, as equacões (3.69) são chamadas **equacões diferenciais de Riccati.**

3.2.7 Algoritmo de filtragem

Na filtragem o requerimento é computar uma estimação do estado $\hat{X}(n/Y(n))$. Quer dizer, obter uma estimação do estado no mesmo tempo de predição. O algoritmo apresentado à continuação é o filtro de Kalman para o caso da filtragem:

$$\Sigma(n)=C(n)P(n-1)C^T(n)+R(n)$$

$$H(n)=P(n-1)C^T(n)\Sigma^{-1}(n)$$

$$\alpha(n)=y(n)-C(n)\Phi(n,n-1)\hat{X}(n-1/Y(n-1))$$

$$\hat{X}(n/Y(n))=\Phi(n,n-1)\hat{X}(n-1/Y(n-1))+H(n)\alpha(n)$$

$$P^+(n)=P(n-1)-H(n)C(n)P(n-1)$$

$$P(n+1)=\Phi(n+1,n)P^+(n)\Phi^T(n+1,n)+Q(n)$$

onde a matriz H(n) chamada, matriz de ganho do filtro, está relacionada com a matriz G(n) através de: $H(n) = \Phi(n,n-1)C(n)$.

3.2.8 Condicões iniciais e estimacões não polarizadas:

Dado que o algoritmo de estimação de estado do filtro de Kalman é recursivo, ele precisa de valores iniciais para a estimação de estado $\hat{X}(0/Y(0))$ e P(0). No caso de falta de informação deverão escolher-se valores para estas variáveis da seguinte forma:

$$\hat{X}(0/Y(0)) = E[X(0)] \tag{3.70}$$

$$P(0) = E[X(0)X^T(0)] \tag{3.71}$$

Esta eleição é crítica em quanto a que condiciona a otimalidade do algoritmo, isto é a minimização do traço da matriz de autocorrelação do erro de estimação, como veremos a seguir.

Provaremos também que a estimação de estado calculada com o filtro de kalman é não polarizada. Para provar isso, primeiro recordaremos que isto será certo ***se e somente se*** verifica-se que:

$$E[\hat{X}(n/Y(n))] = E[X(n)] \tag{3.72}$$

Provaremos isto utilizando as equações (3.23), (3.25), (3.40) e (3.58)

$$\begin{aligned} E[\hat{X}(n/Y(n))] &= \Phi(n,n-1)E[\hat{X}(n-1/Y(n-1))] \\ &+ H(n)C(n)\Phi(n,n-1)\{E[X(n-1)] - E[\hat{X}(n-1/Y(n-1))]\} \end{aligned} \tag{3.73}$$

Então suponnha que para n=1 a equação (3.73) fica:

$$\begin{aligned} E[\hat{X}(1/Y(1))] &= \Phi(1,0)E[\hat{X}(0/Y(0))] \\ &+ H(1)C(1)\Phi(1,0)\{E[X(0)] - E[\hat{X}(0/Y(0))]\} \end{aligned} \tag{3.74}$$

como $\hat{X}(0)$ debe ser especificada podemos usar:

$$E[\hat{X}(0/Y(0))] = \hat{X}(0/Y(0))] \tag{3.75}$$

Então se escolhemos a $\hat{X}(0)$ de acordo à equação (3.70), Então logo de simplificar, a equação (3.74) obtemos:

$$E[\hat{X}(1/Y(1))]=\Phi(1,0)E[X(0)] \tag{3.76}$$

Como a partir da equação de estados tomamos o valor esperado a ambos lados e recordando que o ruído de estados é de valor médio nulo, obtemos que:

$$E[X(1)]=\Phi(1,0)E[X(0)] \tag{3.77}$$

comparando as equações (3.77) e (3.76), obtemos:

$$E[\hat{X}(1/Y(1))]=E[X(1)] \tag{3.78}$$

Por indução, a partir das equações (3.75) e (3.78), finalmente provamos que a estimação é não polarizada:

$$E[\hat{X}(n/Y(n))]=E[X(n)] \tag{3.79}$$

A derivação do filtro de Kalman baseia-se na premissa que o erro de estimação de estado:

$$\tilde{X}(n)=X(n)-\hat{X}(n) \tag{3.80}$$

é ortogonal ao espaço gerado pelos vetores observados Y(1),...,Y(n). Esta ortogonalização implica que a predição é a mínima possível. Então a estimação do estado satisfaz a condição:

$$E[\tilde{X}^T(n)\tilde{X}(n)]=\text{mínimo} \tag{3.81}$$

Isto é equivalente a dizer que é uma estimação de mínima variância. Então podemos dizer que a estimação de estado obtida pelo filtro de Kalman é uma estimação ***linear, não polarizada e de mínima variância.***

3.3 - Exemplo: Filtro de Kalman para estimação de uma constante

Apresentaremos a continuação um exemplo de aplicação do filtro de Kalman escalar para a estimação de uma constante. Para isto modelaremos o estado como um valor

constante e a observação como uma observação ruidosa. No gráfico seguinte (figura 3.4) apresenta-se com linha cheia a convergencia da estimação do estado até o verdadeiro valor -0.5.

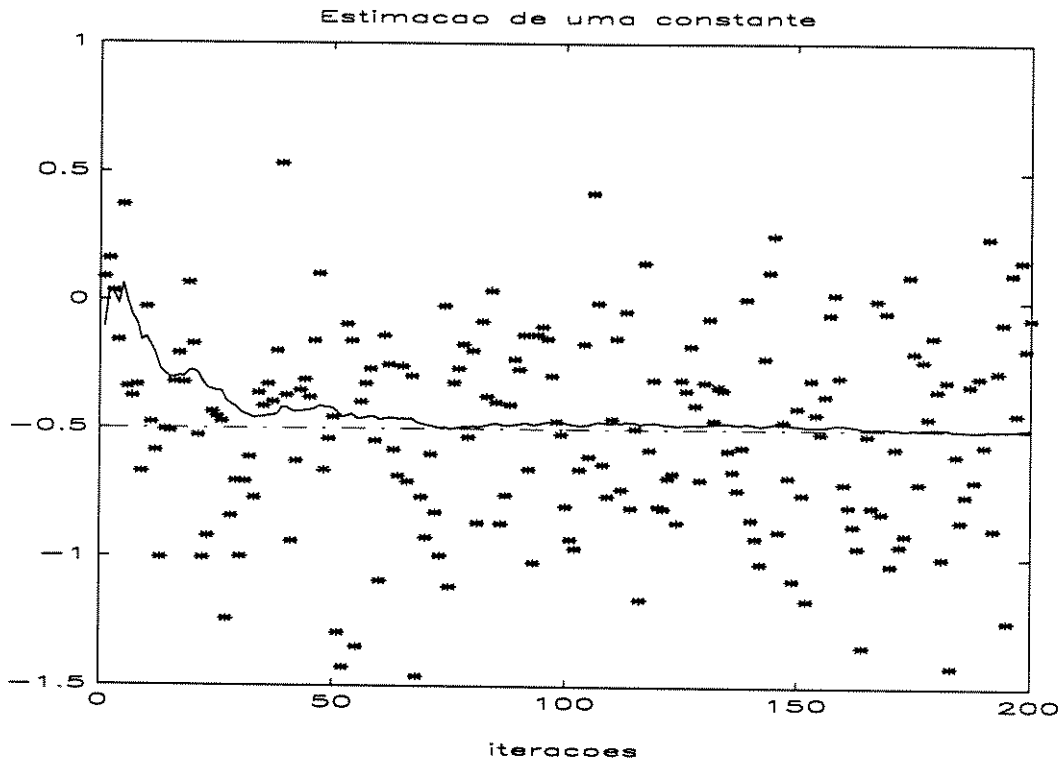


FIGURA 3.4 - Estimação de uma constante (-0.5)

Podemos observar na figura 3.4, que o filtro de Kalman, representado em linha cheia, mesmo partindo de uma estimação inicial grosseira, converge rápidamente ao verdadeiro valor, em linha pontilhada, levando em conta que as observações são sumamente ruidosas, pontos '*'. Sobre o final do experimento a estimação é práticamente igual ao verdadeiro valor e além disso é de variância quase nula. Este efeito se obtém porque utiliza-se toda a informação desde o princípio do experimento para a estimação.

3.4 - Exemplo: Filtro de Kalman para estimação escalar com memória finita

Um dos mais importantes problemas na hora de implementar um filtro de Kalman para resolver um problema do mundo real, é que este suponha que o modelo de estados não varie com o tempo. Pelo contrário os problemas do mundo real em geral caracterizam-se por sua variação com o tempo. Um exemplo disto é o controle de um aeroplano ou um foguete, na medida que este evoluciona, ao gastar seu combustível faz como que o modelo

tenha que mudar. Para poder enfrentar estes problemas, deve-se descartar dados antigos que tennhan pouca relação com a situação no momento da estimação.

Uma das técnicas mais utilizadas para resolver este objetivo é utilizar um filtro do tipo memória "desvanêscente" [8]. Isto é um filtro onde os dados usados para a estimação do estado atual sejam os de uma janela de tempo. Desta forma descartam-se os dados do passado distante de pouca relação com o estado atual. O mecanismo mais freqüentemente usado é o pesado de acordo à distância em tempo com os dados presentes. Isto se obtém *aumentando* a *covariância da matriz do erro de medição* P. Uma forma de obter isto é através da siguiente equação.

$$P_k^* = s^{(j-k)} P_k \quad s \supseteq 1;\ k = j,\ j-1,\ j-2,... \tag{3.82}$$

onde P* é a nova matriz de covariância do erro de medição, j é um número maior ou igual a 1 e s é um número maior ou igual a 1 que controla a profundidade da memória. Para s=1, verifica-se que P*=P, para s~1, a memória é muito profunda e para s>1 e s~0, a memória é de curto prazo.

Uma das vantagens da equação da nova matriz de covariância do erro de estimação pode-se insertar muito fácilmente nas equacões do filtro de Kalman, que ficam da siguiente maneira:

$$\Sigma(n) = C(n)P(n-1)C^T(n) + R(n)$$

$$H(n) = P(n-1)C^T(n)\Sigma^{-1}(n)$$

$$\alpha(n) = y(n) - C(n)\Phi(n,n-1)\hat{X}(n-1/Y(n-1))$$

$$\hat{X}(n/Y(n)) = \Phi(n,n-1)\hat{X}(n-1/Y(n-1)) + H(n)\alpha(n)$$

$$P^+(n) = P(n-1) - H(n)C(n)P(n-1)$$

$$P(n+1) = s\Phi(n+1,n)P^+(n)\Phi^T(n+1,n) + Q(n)$$

onde a única modificação realizada é na última equação de atualização da matriz de covariância do erro de estimação.

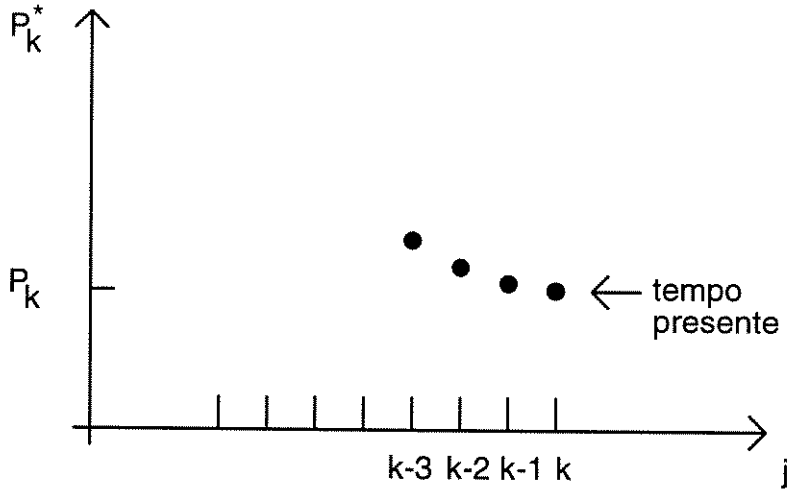


FIGURA 3.5 - Evolução da matriz de covariância do erro de estimação

Este exemplo é muito similar ao anterior mas com a característica que o estado muda com o tempo. Foi proposto com o objetivo de destacar as características do filtro de Kalman adaptativo.

Esa variação é proposta do seguinte modo. O verdadero estado mantem-se constante entre as iteracões 0 até 70 e 130 até 200, tomando os valores -0.5 e 0.5 respectivamente. A transição entre as iteraciões 70 até 130 é linear. Esta evolução visualiza as diferentes etapas da estimação do estado. No início observa-se na zona constante o começo da evolução do filtro, a zona linear permite visualizar o "traking" feito pelo filtro e finalmente a zona constante permite ver como realiza-se a estimação logo da transição.

Para ver a dependência do filtro com a constante s que regra a profundidade da memória, apresentam-se varias figuras com diferentes valores de s. A figura 3.6 apresenta a estimação com s=1.01. Como vêmos este caso é de memória muito profunda e por tanto não pode seguir a evolução da reta. Assim mesmo, devido ao mesmo efeito, nas iterações 50 até 70, realiza-se uma precisa estimação do verdadero valor da variável.

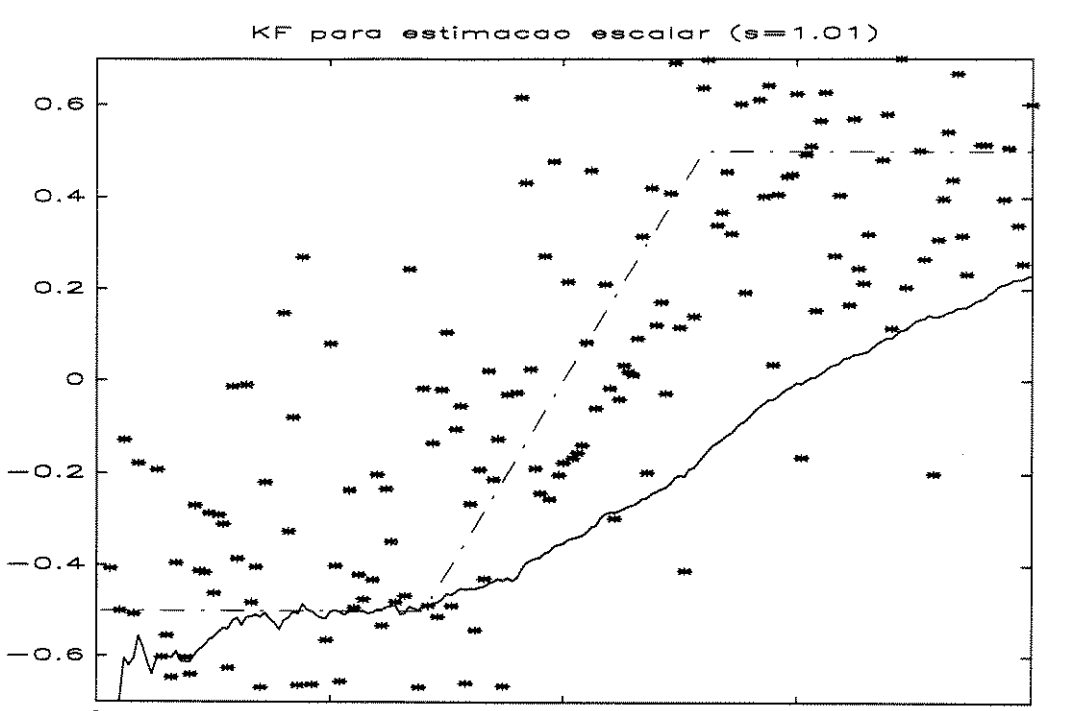


FIGURA 3.6 - Estimação escalar com s = 1.01

A figura 3.7 apresenta a estimação com s=1.1. Este caso é o de memória curta e por tanto pode seguir a evolução da reta, em inclinação, mas com um pequeno retardo. Por ter memória de curto prazo, a estimação nas zonas constantes é mais ruidosa que na figura 3.6.

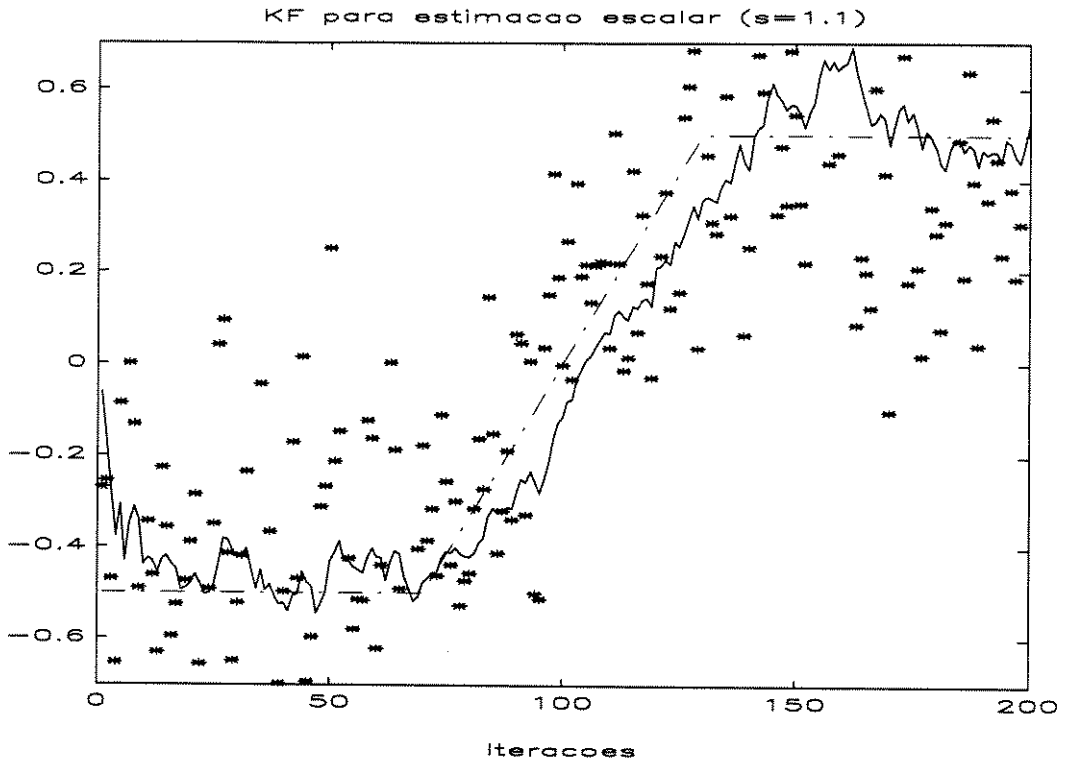


FIGURA 3.7 - Estimação escalar com s = 1.1

A figura 3.8 apresenta a estimação com s=1.15. Este caso é o de memória de "muito curto prazo" e por tanto pode seguir a evolução da reta, muito melhor que nas figuras anteriores. Por ter memória de muito curto prazo, a estimação nas zonas constantes é também mais ruidosa que nas figuras anteriores.

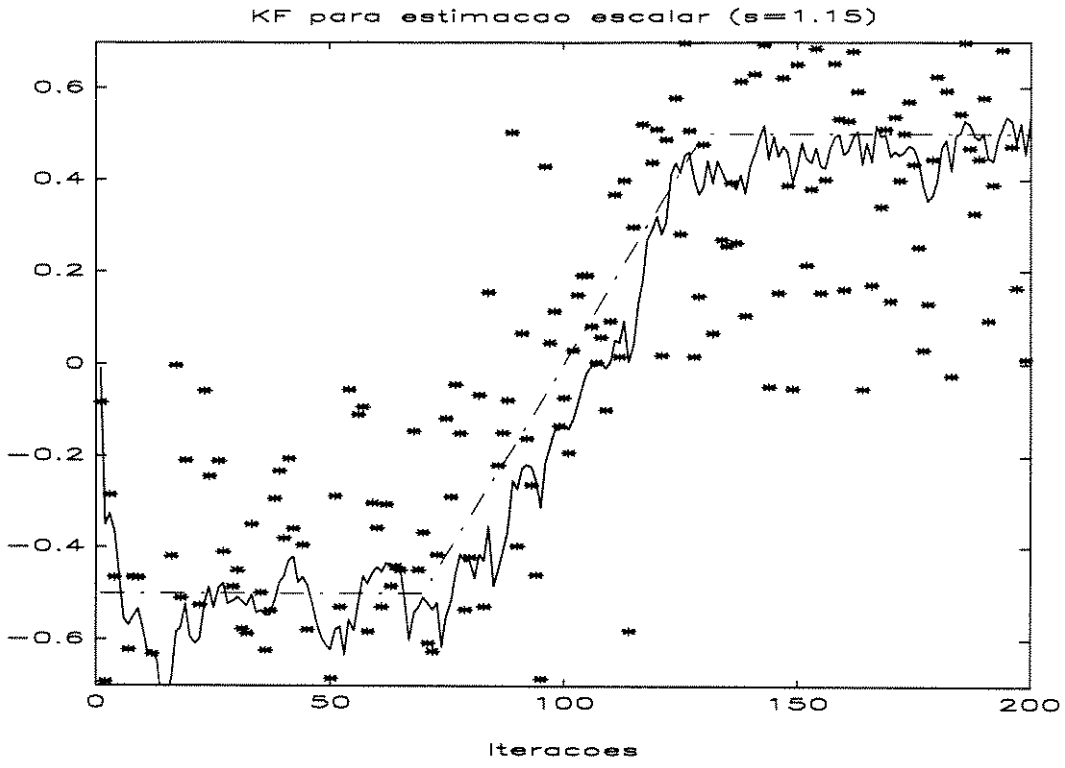


FIGURA 3.8 - Estimação escalar com s = 1.15

Conclui-se que existe uma *relação de compromiso entre a profundidade da memória e a variância da estimação*. Se a memória é de longo prazo disponhe-se de muitas amostras para realizar a estimação e cada novo valor pesa relativamente menos que o resto. Se a memória é de curto prazo, a variância da estimação aumenta debido a maior dependência com os valores atuais de medição, mas ao mesmo tempo a capacidade de reação à mudança do estado é maior.

3.5 - Exemplo: Comparação do Filtro de Kalman e LMS para equalização adaptativa

Neste exemplo compara-se a desempenho do filtro de Kalman e do algoritmo LMS para a equalização adaptativa de um canal telefônico. Para a eficiente transmição de dados a alta velocidade, precisa-se uma equalização do canal. O equalizador mais usado é um filtro linear do tipo MA cujos coeficientes são ajustados em forma adaptativa a partir do sinal de entrada e um sinal desejado, gerado em sincronismo com o receptor. No modo treinamento realiza-se o ajuste indicado e no modo reconhecimento os pesos de filtro são fixos. Este tipo de equalização é muito efetiva na prática já que permite a adaptação frente a mudança de condições do canal. Porém o período de treinamento do equalizador retarda a transmisão de dados. Desta forma a menor tempo de adaptação, menor tempo de retardo na transmisão. O exemplo que será apresentado foi extraído de [19].

Seja $\mathbf{u}(n)$ o vetor de entrada ao filtro equalizador em tempo n, seja $\mathbf{h}(n)$ o vetor de coeficientes do filtro em tempo n. Quando $\mathbf{h}(n)$ encontra-se em seu mínimo global, $\mathbf{h_o}(n)$, o erro quadrático medio asume seu valor ε_{min}. Seja $e_o(n)$ o sinal de erro para o valor ótimo de h(n), Então assumindo estacionariedade, as equações de estado são:

$$\mathbf{h_o}(n+1)=\mathbf{h_o}(n) \tag{3.83}$$

$$d(n)=\mathbf{u}^T(n)\mathbf{h_o}(n)+\mathbf{e_o}(n) \tag{3.84}$$

onde d(n) é o sinal desejado. A equação (3.82) é a equação de estado, onde expressa-se que o estado ótimo não muda com o tempo, de forma que a matriz de transição de estado é a matriz identidade e o ruído de estado é nulo. A equação (3.83) é a equação de medição, que expresa que o sinal desejado é o produto escalar entre o vetor de entrada $\mathbf{u}(n)$ e o vetor de coeficientes ótimos $\mathbf{h_o}(n)$ mais um erro de medida $e_o(n)$. Para poder aplicar a teoría do filtro de Kalman, assumimos que $\{e_o(n)\}$ é um processo de ruído branco de média zero e variância ε_{min}. Esta é uma aproximação razoável já que normalmente o mínimo erro quadrático médio é pequeno, tal que a correlação entre amostras sucesivas do sinal de erro $e_o(n)$ possam ser descartadas.

Levando em conta as equações de estado apresentadas e os nomes das variáveis utilizadas, o filtro de Kalman para a atualização dos coeficientes do filtro equalizador fica:

$$\Sigma(n)=\mathbf{u}^T(n)\mathbf{K}(n-1)\mathbf{u}(n)+\varepsilon_{min} \tag{3.85}$$

$$\mathbf{G}(n)=\mathbf{K}(n-1)\mathbf{u}(n)\Sigma^{-1}(n) \tag{3.86}$$

$$\alpha(n)=d(n)-\mathbf{u}^T(n)\hat{\mathbf{h}}(n) \tag{3.87}$$

$$\hat{\mathbf{h}}(n)=\hat{\mathbf{h}}(n-1)+\mathbf{G}(n)\alpha(n) \tag{3.88}$$

$$\mathbf{K}(n)=\mathbf{K}(n-1)-\mathbf{G}(n)\mathbf{u}^T(n)\mathbf{K}(n-1) \tag{3.89}$$

Com as condições iniciais:

$$\begin{aligned}\hat{\mathbf{h}}(0)&=0\\ \mathbf{K}(0)&=c\mathbf{I} \quad ;c>0\end{aligned} \tag{3.90}$$

Onde o valor de ε_{min} não se pode conhecer á priori, de modo que se escolhe um valor arbitrário. Típicamente 0.01 ó 0.001 são valores usados na prática.

Na próxima figura apresentaremos as curvas de erro de estimação do filtro equalizador utilizando o filtro de Kalman e o algoritmo LMS. O vetor de coeficientes de filtro original foi: [0.1 0.2 0.3 0.4]'

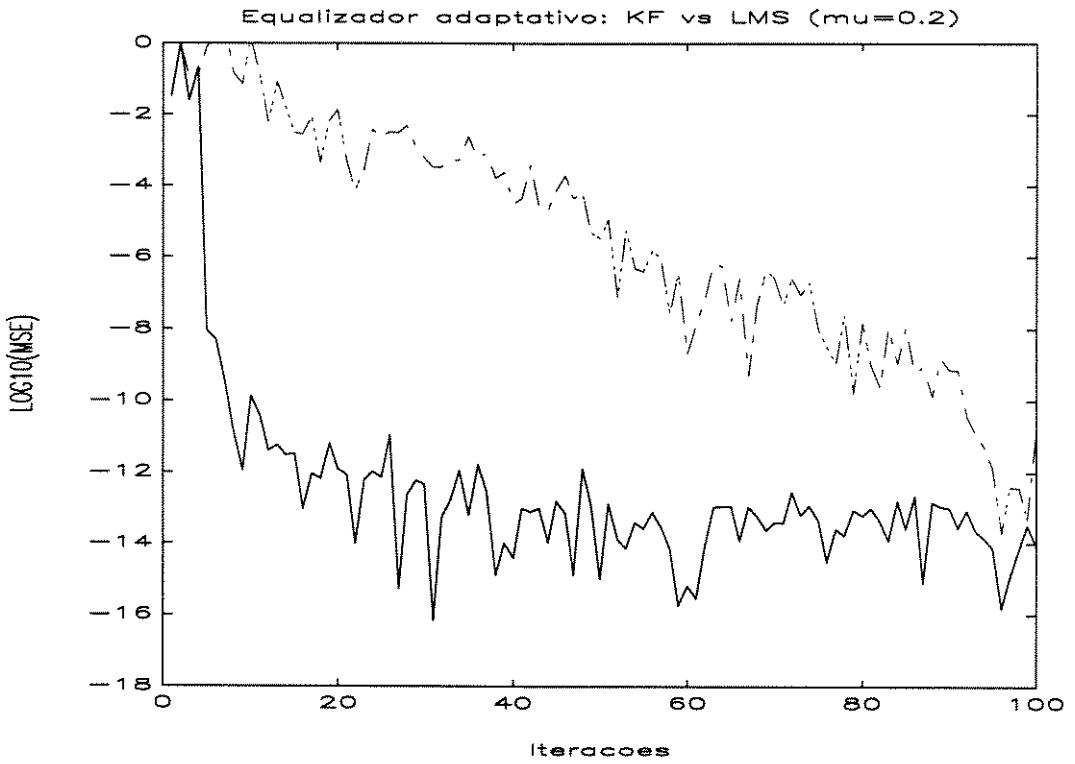


FIGURA 3.9 - Erro de estimação do filtro equalizador

Observamos a significativa diferença na velocidade de convergência entre o filtro de Kalman e o filtro LMS. O filtro de Kalman nas primeiras 10 iterações diminui em mais de um ordem de magnitude o erro de estimação, enquanto que para obter aproximadamente o mesmo descenso o algoritmo LMS precisa de aproximadamente umas 50 iterações.

Na próxima figura apresentaremos a evolução dos coeficientes até os valores originais utilizados: [0.1 0.2 0.3 0.4]'

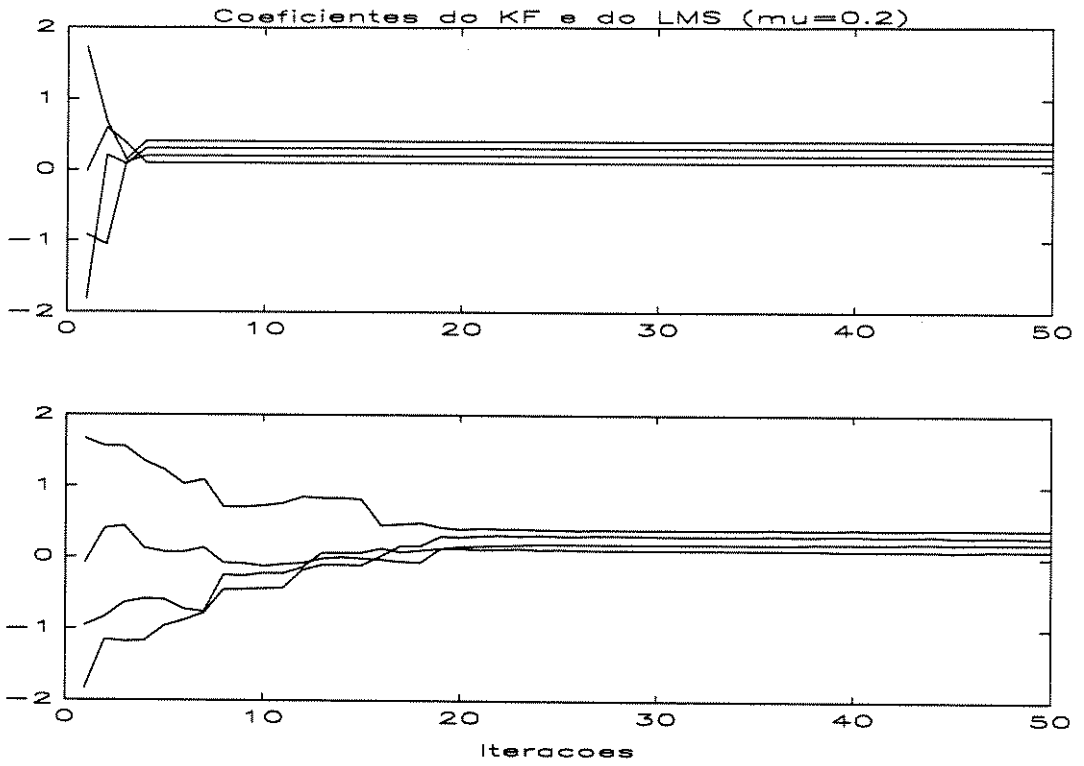


FIGURA 3.10 - Evolução dos coeficientes

Nestes gráficos pode-se evidenciar que já na iteração 5, os coeficientes convergem ao seu valor verdadeiro, sendo que nas iterações seguientes realizam-se ajustes infinitesimais. Pelo contrário a evolução do algoritmo LMS é muito mais lenta.

CAPÍTULO 4

REDES NEURAIS E RECONHECIMENTO DE VOZ RUIDOSA.

4.1- RECONHECIMENTO DE PADRÕES - CRITÉRIO DE BAYES

4.1.1 - Descrição geral

Em forma geral, o reconhecimento de padrões (RP) é a ciência que concerne à descrição ou classificação de medidas [16]. Pode-se definir uma classificação como um mapeamento entre o sinal de entrada e uma ou mais classes pré-especificadas através da extração de características significativas ou atributos, e o processamento destes atributos. Então o RP é a habilidade de classificar. As técnicas do RP são uma parte importante dos sistemas inteligentes, sendo usados tanto no processamento de dados como na etapa de toma de decisão. Há pouca dúvida que o RP é uma importante tecnologia de rápido desenvolvimento com interesse e participação interdisciplinar. O RP não está limitado a uma aproximação mas a um extenso conjunto de técnicas e conhecimentos. Entre suas principais aplicações pode-se mencionar :

- Pré-processamento, segmentação e análise de imagens.
- Visão computacional.
- Análise sísmico.
- Classificação e análise de sinais de radar.
- Reconhecimento de voz.
- Análise de sinais eletrocardiográficas.
- Diagnose médico.

Entre as principais áreas com as quais o RP está relacionado, pode-se mencionar:

- Processamento adaptativo de sinais e sistemas.
- Inteligência artificial.
- Modelamento neural.

- Teorias de estimação e otimização.
- Linguagens formais.
- Teoria de autômatas.
- Lógica fuzzy.

4.1.2 - Mapeamento de padrões

O RP pode ser caracterizado como um processo de ***mapeamento de informação***, redução ou rotulado de informação. Uma visão abstrata do problema de classificação /descrição do RP é apresentada na figura 4.1.

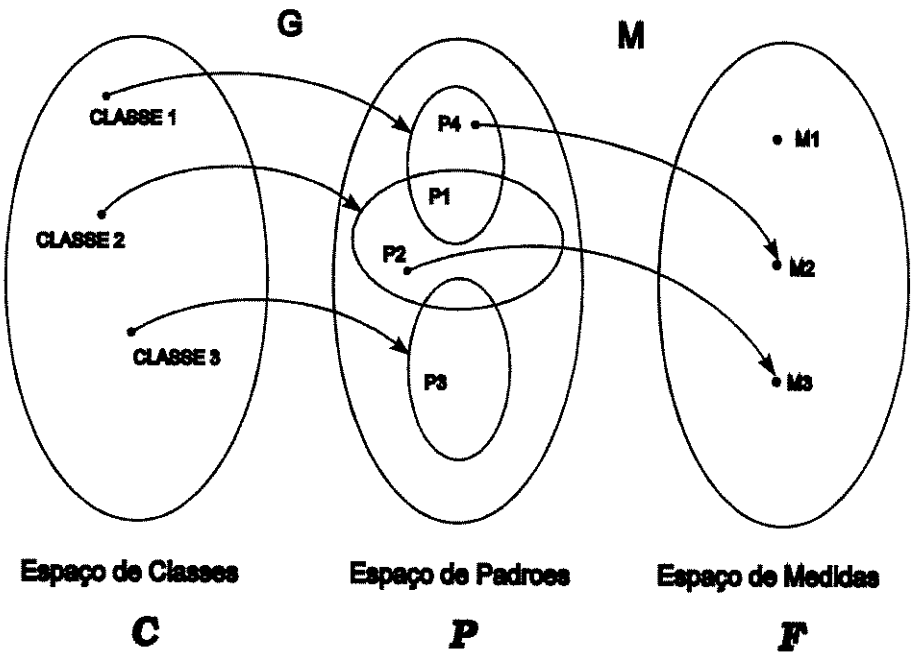


FIGURA 4.1 - Mapeamento na representação do processo de RP

Nesta figura apresentam-se dois mapeamentos diferentes. O primeiro é um mapeamento entre o espaço das classes C (onde as diferentes classes são declaradas) e o espaço dos padrões P (onde os diferentes padrões são agrupados, segundo a classe que os gero). O segundo é um mapeamento entre o espaço dos padrões P e o espaço das características ou medidas F (a F provém de "feature" que significa característica em inglês) (onde as diferentes medidas são geradas por cada padrão). A relação que une os dois primeiros mapas denota-se por G, que pode ser probabilística, onde cada classe gera um

subconjunto de padrões no espaço dos padrões ***P***. Os subconjuntos do espaço podem-se sobrepor o que implica que compartem atributos. A relação que une os outros dois mapas denota-se por M, onde cada padrão dos subconjuntos de ***P*** gera uma observação ou característica no espaço ***F***.

Então usando estes conceitos, o objetivo de muitos problemas de RP é: dada uma medida m_i, pertencente ao espaço das medidas ***F***, deseja-se um mecanismo para ***identificar*** e ***inverter*** os mapeamentos dados por G e M, de forma tal de associar a cada medida uma classe no espaço original ***C***. O principal problema é que estes mapeamentos geralmente não são, funções, e mesmo se fossem, rara vez são invertíveis. A dificuldade principal é a superposição das medidas ou características. Então a eleição do sistema de extração de características e portanto da função M é clave para obter, se possível, um mapeamento invertível. O desenho de um bom sistema de medida é um aspecto importante do desenho de um bom sistema de RP. Uma boa eleição da função favorece a discriminatividade, isto é a capacidade de discriminar as classes através de suas medidas.

Outro aspecto importante é que ainda que dois padrões estejam contíguos ou próximos no espaço ***P***, já que foram gerados pela mesma classe w_1, não necessariamente suas medidas vão estar próximas no espaço ***F***. Observa-se que os padrões P_1 e P_4 que estão no mesmo subconjunto (portanto perto em quanto à distância entre ambos), de ***P***, geram duas medidas m_1 e m_3, que não necessariamente estão próximas no espaço ***F***. Isto é um aspecto importante no caso que se utilizem mecanismos de "clustering" como medida de proximidade.

4.1.3 - Estrutura típica de um sistema de RP

Na figura 4.2 se apresenta um diagrama geral de um sistema de RP.

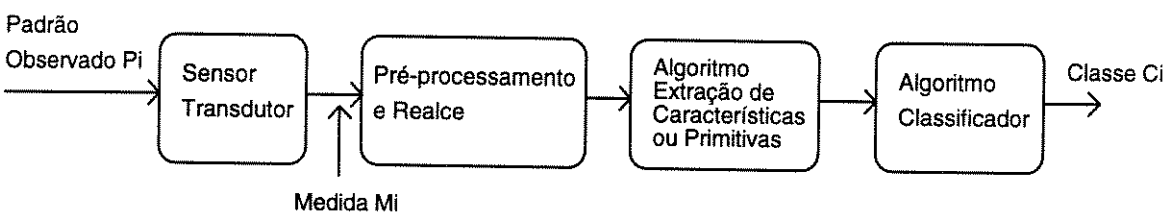


FIGURA 4.2 - Diagrama geral de um sistema de RP

O padrão observado P_i pode ser qualquer objeto físico disponível a serem medido. Possíveis exemplos podem ser: pressão sanguínea, voz, uma imagem, etc. A partir deste,

um sensor ou transdutor encarrega-se de transformar este sinal numa forma factível de ser processada, em geral por meios eléctricos ou eletrônicos, como pode ser um microfone, sensores de eletroencefalografía ou uma câmara de vídeo que transforma sua entrada em sinais elétricas. A partir da medida m_i passa-se por um pré-processamento ou realce que implica uma transformação dos dados para ajudar na computação e na extração de características ou na filtragem para eliminar ruído. Logo, através de um algoritmo apropriado, extraem-se as características ou primitivas. Estas são posteriormente classificadas por um algoritmo ou classificador adequado, que fornece a classe c_i do padrão observado.

Em forma não rigorosa, as ***características*** podem ser definidas como ***qualquer medida possível de ser extraída***. Estas podem ser de baixo nível como intensidades de sinal. Podem ser numéricas, como peso (em quilos) e/ou simbólicas como por exemplo as côres. As características podem ser também produto de um algoritmo de extração de características aplicado ao conjunto de dados disponíveis. Isto poderia implicar um alto custo computacional se as características extraídas são resultado de um processamento de alto nível. A eleição do tipo de características a serem extraídos depende 1) de seu custo computacional, 2) de sua redução da dimensionalidade do problema, sem descartar informação vital, ou de sua compactação da informação 3) de sua discriminatividade, o que redundará em baixos erros de classificação.

4.1.4 - Distorção de padrões - um problema fundamental

Um aspecto muito importante a levar em conta é que as características extraídas podem ter erros ou ruído. Isto implica que o mecanismo de extração é suscetível de ter erros, que poderiam ser do tipo computacional ou outros. Também uma medição com ruído poderia afetar este mecanismo de forma tal que as características extraídas do padrão observado não correspondam as do verdadeiro padrão.

Em geral busca-se uma classificação, reconhecimento ou descrição de padrões que seja ***invariante*** a algumas mudanças conhecidas nos padrões respeito do caso "ideal" [16]. Estas desviações podem incluir um conjunto muito grande de casos, como as perturbações aleatórias. Esta é uma tarefa muito difícil de lograr devido a que ainda um conjunto de padrões de uma mesma classe pode exibir grandes variações. Entre estas variações podem-se mencionar as amplificações ou atenuações, as translações e as rotações em caso de imagens. Um caso muito importante são as perturbações próprias da natureza de um processo estocástico. Cada exemplar é uma realização do processo, e um conjunto de

exemplares do mesmo processo exibe grandes diferenças na sua evolução temporal e seu único fator comum é sua estatística.

Uma aproximação muito usada na prática e apresentada em muitos livros de RP como [16], é a procura de ***características invariantes*** de forma que as medidas extraídas dos padrões não mudem ante possíveis distorções como as mencionadas no parágrafo anterior. No caso do processamento de imagens, estas perturbações, do tipo *determinístico*, são bem definidas como RST (de Rotação, eScala e Translação), então as características a utilizar deseja-se que sejam invariantes ante este tipo de perturbações. Porém *não sempre é possível obter características invariantes*, particularmente no caso da presença de distorções do tipo de ruído aleatório, isto é *quando processam-se realizações de um processo estocástico*. Para obter uma invariância total de um processo estocástico deve-se dispor de informação infinita [19]; isto implica que na prática vai-se a trabalhar com *estimações* dos verdadeiros valores. Então a aproximação anterior leva um risco grande já que se esta invariância não é total, em algum nível de perturbação as características vão a serem distorcidas. Neste caso o classificador incorrerá num erro de classificação.

4.1.5 - O espaço e o vetor de características, regiões de decisão

Em geral é muito útil desenvolver um ponto de vista geométrico das características. Estas são organizadas num ***vetor de características*** d-dimensional e o conjunto possível de todas as características define um espaço multidimensional de características, também chamado ***espaço de características***. Cada característica define um número real ou um ponto neste espaço $\mathbf{R^d}$.

Então com estes conceitos, visualiza-se o RP como uma partição do espaço de características em diferentes regiões, correspondendo cada uma com uma classe. Um classificador particiona (por algum meio como veremos mais adiante) o espaço de características em regiões com as seguintes condições: 1) Estas regiões devem cobrir o espaço $\mathbf{R^d}$, 2) Devem ser disjuntas (não se devem sobrepor) caso contrário não seria um mapeamento único (uma exceção são os conjuntos "fuzzy"). Cada borde de cada região de decisão chama-se *fronteira de decisão*. Deste ponto de vista a classificação de um padrão resulta uma tarefa simples: se determina a região de decisão ao qual o vetor pertence, e se mapeia esse vetor à classe correspondente. Ainda que a classificação seja direta, no caso de medições não ruidosas, o verdadeiro desafio é a determinação das fronteiras de decisão.

É importante destacar o caso no qual a classificação é feita num ambiente ruidoso, se as fronteiras das classes já foram determinadas. Neste caso as características extraídas

dos padrões podem ter distorções. Isto significa que a ***observação ruidosa*** de um padrão, através desta ***distorção das características***, pode ***indicar a pertença a outra classe diferente da sua verdadeira classe*** .

A seguir apresenta-se uma figura onde o espaço de características é bidimensional e se apresenta uma possível separação deste espaço em várias classes. Esta separação é do tipo geral já que as fronteiras são retas, curvas e figuras geométricas.

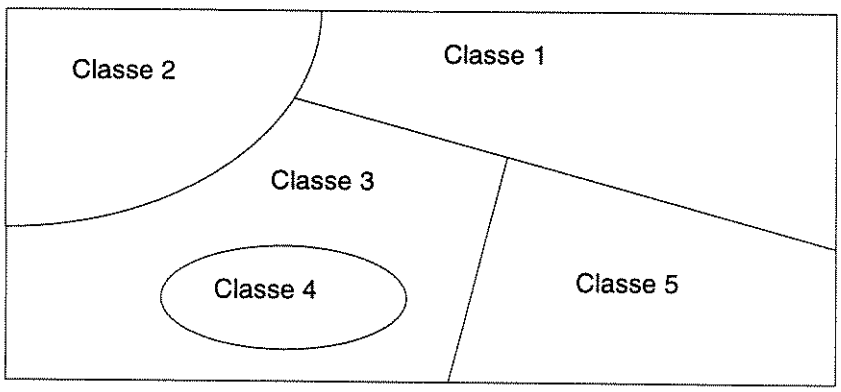


FIGURA 4.3 - Separação em classes de um espaço bidimensional

Para encontrar estas fronteiras de classificação, faz sentido utilizar a máxima informação disponível no desenho do sistema de RP. Esta informação *a priori* disponível encontra-se em forma de uma base de dados onde organizam-se amostras de padrões cuja classe é conhecida. Esta base de dados é comumente chamada ***conjunto de treinamento***. Esta informação armazenada é sumamente importante já que fornece amostras "típicas" de cada uma das classes, e informação sobre como associar as classes com as saídas desejadas. Então este conjunto de treinamento permite ao sistema de RP o "aprendizado" de informação relevante acerca dos atributos de cada classe. Associa-se este aprendizado à determinação das fronteiras de classificação do espaço de características.

Existem duas aproximações tradicionais na área de RP acerca do treinamento. Uma chama-se treinamento ou aprendizagem ***supervisionado***, já que utiliza no treinamento exemplares rotulados (com conhecimento da classe de pertença). A outra aproximação chama-se aprendizado ***não supervisionado***, onde no treinamento não utilizam-se amostras rotuladas já que se deseja que o sistema se auto organize de forma a encontrar partições naturais do espaço.

4.1.6 - Reconhecimento estatístico de padrões - critério de Bayes

A teoria de Bayes é a aproximação estatística fundamental ao problema de classificação de padrões. O Reconhecimento Estatístico de Padrões (REP) está baseado num fundamento estatístico para o reconhecimento ou classificação de padrões. Neste caso as características extraídas dos dados de entrada estão baseadas nas densidades de probabilidade a priori e a posteriori e na aplicação da regra de Bayes.

Começaremos com um caso simples de duas categorias. Seja **x** uma variável aleatória contínua, sejam w_i, onde i=1,2, os diferentes estados ou categorias às quais desejamos mapear a variável **x**, seja $P(x/w_i)$ a densidade de probabilidade condicional (conhecida) de x dado o estado w_i, esta se interpreta como a probabilidade de que um dado valor medido de x pertença ao estado w_i. Suponha que também se conhecem as probabilidades $P(w_1)$ e $P(w_2)$. Então se mede um valor de x. Como influirá essa medição sobre a nossa decisão de mapearia a um estado ou outro? Usando a regra de Bayes obtemos:

$$P(w_j/x) = P(x/w_j) * P(w_j) / P(x);\ j = 1,2 \tag{4.1}$$

com $P(x) = P(x/w_1)*P(w_1) + P(x/w_2)*P(w_2)$. Sendo as $P(w_j)$ as probabilidades a priori e $P(w_j/x)$ as probabilidades a posteriori, ou seja uma vez realizada a medida.

Calcula-se a probabilidade de erro para uma medição de x dada, como a probabilidade de mapear um padrão errôneamente à uma classe ao qual não pertence:

$$P(\text{erro}/x) = \begin{cases} P(w_1/x) & \text{se decido por } w_2; \\ P(w_2/x) & \text{se decido por } w_1; \end{cases} \tag{4.2}$$

Então a regra seria: decidir w_1 se $P(w_1/x) > P(w_2/x)$ e w_2 caso contrário. Agora é importante saber se esta regra de decisão minimiza a probabilidade de erro. Em efeito é minimizada já que:

$$P(\text{erro}/x) = \int_{-\infty}^{\infty} P(\text{erro}/x) * P(x) dx \tag{4.3}$$

Se para cada x a P(erro/x) é tão pequena quanto possível, então minimiza-se a probabilidade de erro total.

Vamos generalizar agora para o caso multi dimensional. Seja **x** um vetor de características d dimensional, $\Omega=\{w_1,...,w_S\}$ um conjunto finito de S estados e $A=\{\alpha_1,...,\alpha_A\}$, um conjunto de A possíveis ações. Seja $\lambda(\alpha_i/w_j)$ a perda por tomar a ação α_i quando o estado verdadeiro era o w_j. Seja $P(\mathbf{x}/ w_j)$ a densidade de probabilidade condicional (conhecida) de **x** dado o estado w_j, e seja a $P(w_j)$ a probabilidade a priori do estado w_j. Todas estas $\forall j=1,..,S$ e $\forall i=1,..,A$. As probabilidades a posteriori podem-se calcular utilizando a regra de Bayes:

$$P(w_j/\mathbf{x}) = P(\mathbf{x}/ w_j) * P(w_j) / P(\mathbf{x}); j = 1,..,S \tag{4.4}$$

onde $P(\mathbf{x})$ é a probabilidade total do valor de **x**.

Então se medimos um valor de **x** e decidimos tomar uma ação α_i, sendo o verdadeiro estado w_j a perda esperada, também chamada perda condicional obtemos:

$$R(\alpha_i / x) = \sum_{j=1}^{S} \lambda(\alpha_i / w_j) P(w_j / x) \tag{4.5}$$

Para calcular a perda total devemos integrar a expressão anterior sobre todos os possíveis valores de **x**:

$$R = \int R(\alpha(x) / x) p(x) dx \tag{4.6}$$

Claramente se $\alpha(\mathbf{x})$ é escolhido de forma tal, que cada ação é selecionada de forma de minimizar $R(\alpha(\mathbf{x})/\mathbf{x})$ para todo **x** minimiza-se o custo total. Então a regra de decisão de Bayes minimiza o custo total.

Nos problemas de classificação de padrões, cada estado está associado a cada uma das C classes, então A = S = C. Neste caso o custo de escolher errôneamente pode ser tomado como unitário e como nulo em caso correto. Esta é a chamada função de perda simétrica, definida como:

$$\lambda(\alpha_i / w_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \tag{4.7}$$

Isto implica que todos os erros são igualmente custosos, o que pode aplicar-se a alguns casos e a outros não (ver em [16] o exemplo da maçã e a granada). Então o custo total pode-se calcular como:

$$R(\alpha_i / x) = \sum_{j=1}^{C} \lambda(\alpha_i / w_j) P(w_j / x) \tag{4.8}$$

$$= \sum_{j=1,\neq i}^{C} P(w_j / x) \tag{4.9}$$

$$= 1 - P(w_i / x) \tag{4.10}$$

Então como $P(w_i/\mathbf{x})$ é a probabilidade condicional da ação α_i correta. A **regra de decisão de Bayes** (decidir por w_i se $P(w_i/\mathbf{x}) > P(w_j/\mathbf{x})$ $\forall$ $j \neq i$) conclui que se deve escolher a ação i que minimiza o risco condicional o que é o mesmo dizer, **maximiza a probabilidade a posteriori** $P(w_i/\mathbf{x})$ o que implica uma **mínima probabilidade de erro de classificação**.

Há muitas formas de representar classificadores de padrões. Uma forma é em termos de **funções discriminantes**: $g_i(\mathbf{x})$, i=1,..,C. O classificador mapea um padrão de características **x** a uma classe w_i se $g_i(\mathbf{x}) > g_j(\mathbf{x})$ $\forall j \neq i$. Então o classificador computa C funções discriminantes e seleciona o máximo. O classificador de Bayes representa-se fácilmente nesta ótica: para o caso geral: $g_i(\mathbf{x}) = -R(\alpha_i/\mathbf{x})$ para minimizar o custo condicional e para minimizar o erro de classificação atribui $g_i(\mathbf{x}) = P(w_i/\mathbf{x})$. Neste último caso chama-se **classificador MAP**, por maximo a posteriori.

Então o efeito da regra de decisão o de separar o espaço de características ***F*** em **C regiões de decisão**, onde as fronteiras acham-se quando igualam-se duas funções máximas discriminantes. No caso de aplicar a regra de decisão de Bayes, as regiões são chamadas regiões de decisão de Bayes.

4.2- DISTORÇÃO DE PADRÕES NO APRENDIZADO E/OU RECONHECIMENTO

4.2.1 - Definições

Existem muitos tipos de distorções possíveis como já foi mencionado. Dependendo da aplicação algum tipo de distorção estará presente. Em nosso caso nos interessa estudar especificamente a distorção de padrões de origem aleatório.

Nos referiremos ao ruído como qualquer distorção que modifique a representação de um padrão que se deseja classificar. Poderíamos distinguir estas distorções em três grupos:

• Primeiro, os ruídos chamados "tradicionais" de tipo aditivo; ruídos acústicos ambientais como ser as perturbações provenientes de tubos de raios catódicos, de ventiladores, da rede de distribuição de energia elétrica domiciliária (de freqüência 50 Hz.), etc. Tem-se encontrado que a relação sinal-ruído da voz gravada num carro com um microfone colocado no painel de controle, a uma velocidade de 90 Km/h com a janela fechada e sem ligar o ar condicionado pode cair até -5 dB. Enquanto que os ruídos de tipo aleatório como os presentes em ambientes de escritórios, carros pela via urbana e aviões de combate, é importante conhecer os valores típicos de potência, em dB de Intensidade. Estes se apresentam na tabela 4.1 (extraída de [25]).

Condição	Nível de Intensidade
Limiar de Audição	0 dB
Conversação em voz baixa	20 dB
Rádio moderada	40 dB
Conversação normal	65 dB
Rua com muito transito	70 dB
Trem elevado	90 dB
Umbral de sensação desagradavel	120 dB

TABELA 4.1 Intesidade em diversos ambientes

• Segundo, as distorções por não linearidades que afetam o sinal de voz antes de seu processamento para sua classificação. O ambiente de gravação pode ter diversos tipos de reverberação que afetem o espectro de freqüência. Os microfones em função de seu tipo e sua posição também podem afetar significativamente o espectro da voz. No caso que o sistema de reconhecimento seja utilizado na rede telefônica, o canal telefônico é uma fonte adicional de distorção. Dependendo desde onde chegue a chamada poderia passar por uma série de centrais intermediárias que afetam em forma diferente o sinal de voz.

• Terceiro, as distorções próprias da natureza humana, como a coarticulação, onde um fonema é afetado por seu contexto, e o "efeito Lombard", no qual modificam-se a emissão de sinais de voz quando se escuta um som muito forte.

Os ambientes ruidosos provocam uma seria degradação no desempenho dos sistemas de RP. Isto é, um sistema treinado com padrões registrados num determinado ambiente controlado é, em geral, incapaz de reconhecer ainda o mesmo padrão de

treinamento, quando esteja afetado por um sinal aleatório de grande potência. *Este fenômeno é uma das maiores fontes de degradação da desempenho dos sistemas de RP.*

4.2.2 - Distorção de padrões no treinamento, e/ou no reconhecimento

Pode-se separar o problema da aparição de distorções nos padrões, em forma de ruído, em **três casos** segundo o lugar no qual esteja presente:

• O **primeiro caso** seria se somente estão presentes as distorções na etapa de treinamento e não na etapa de reconhecimento. Isto significaria que somente estão disponíveis padrões de treinamento afetados de ruído e que o reconhecimento é realizado em condições de distorção nula. É possível que esta situação exista em alguma aplicação de RP, porém esta é uma situação um tanto forçada para o reconhecimento de voz já que se é possível dispor na etapa de reconhecimento de amostras de voz puras então estas poderiam ser usadas para o treinamento.

• O **segundo caso**, seria quando a presença de distorções somente estará presente na etapa de reconhecimento e se disponhem de amostras de padrões não afetados por ruídos. Para a aplicação sob estudo, o reconhecimento de voz ruidosa, consideraremos o segundo caso já que em geral é possível dispor de amostras de voz registradas em condições de baixa distorção.

• O **terceiro caso** seria quando o ruído está presente em ambas etapas. Este é o caso mais geral, onde as distorções estariam presentes tanto na etapa de reconhecimento como de treinamento e onde seria impossível registrar amostras de padrões que não estejam afetadas por ruído.

4.3 - RECONHECIMENTO DE VOZ - FONEMAS, PALAVRAS

4.3.1 - Principais aplicações

O reconhecimento de voz é uma das aplicações mais importantes do RP. A relevância desta aplicação deriva-se de suas possíveis aplicações no benefício da comunidade, tanto em ajuda a deficientes como na automação de tarefas rotineiras realizadas por operários humanos. Algumas de tais aplicações permitem uma comunicação mais simples e direta com as máquinas através da emissão de comandos de voz. A seguir se detalham algumas das principais aplicações:

- **Ajuda a deficientes** para que através da voz possam dirigir máquinas que lhes ajudem executar tarefas, ou lhes providenciem alimentos. Ainda no caso que este não possa falar corretamente mas sim emitir sons diferenciados, poderia-se resolver o problema através da criação de uma base de dados pessoal.

- **Sistemas bancários** que respondam em forma automática a **solicitações de informação sobre contas bancárias.** Desta forma poderia-se, utilizando um sistema de reconhecimento de dígitos e mais alguns poucos fonemas, atender e responder via um sistema de síntese de voz, as solicitudes de estado de conta, saldos, transferências e outras.

- **Transcrição automática de entrevistas ou conferências gravadas.** Esta ferramenta seria muito útil para jornalistas, já que ao reproduzir o conteúdo da fita poderiam realizar em forma automática a uma versão no computador para sua posterior edição. Outra aplicação sobre o mesmo princípio e o ditado ao micro de cartas ou faxes e posteriormente comandar sua impressão ou seu envio, respectivamente.

- **Comando do computador por via oral,** através da criação de uma base de dados com os fonemas correspondentes aos seus comandos principais. Desta forma poderiam-se utilizar o sistema operativo e diversos programas tipo tabela de cálculos, processadores de texto e base de dados.

4.3.2 - Diferentes blocos de sistemas de reconhecimento de voz

Apresentaremos na figura seguinte, os blocos gerais dos sistemas de reconhecimento de voz

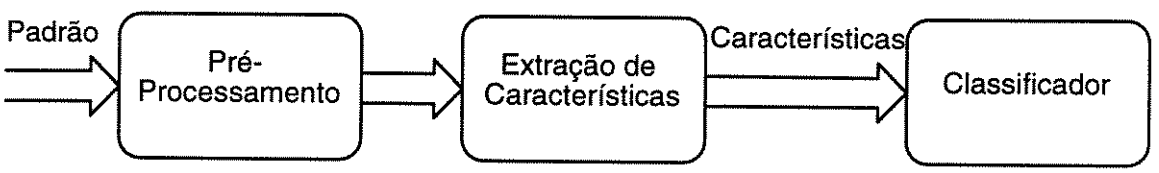


FIGURA 4.4 - Diagrama específico de um sistema de reconhecimento de voz

Vamos descrever cada um destes blocos:

Naturalmente a primeira etapa seria o **registro** dos fonemas de voz, isto é a transformação de um sinal acústico em um sinal elétrico e seu posterior armazenamento num computador. Para realizar este registro deve-se contar com um **Sensor** específico, que pode ser um microfone ou um conjunto de microfones. Nesta etapa é importante especificar alguns aspectos. Por um lado o lugar onde será feito este registro. Como possíveis alternativas, entre outras, podem ser uma sala de gravação a prova de som como lugar onde se fará o reconhecimento. Esta eleição influenciará o nível de ruído ambiente que contaminaria as amostras. Também deve-se especificar a precisão com que será feito este registro. Isto é, com que quantidade de bits se fará a conversão analógico - digital.

Uma vez armazenados os fonemas em meios magnéticos, geralmente realiza-se um **Pré-processamento.** Se pode realizar um filtragem com o objetivo de eliminar componentes de ruído não desejados. Outra possível tarefa seria uma normalização do sinal de voz, já que as placas de aquisição entregam números inteiros desde zero até um máximo que depende da quantidade de bits usada (256 para 8 bits, 4096 para 12 bits e 65535 para 16 bits). Também poderia-se mencionar a filtragem de pré-ênfase, muito usado como passo prévio à extração de coeficientes de predição linear. Esta filtragem é feito com o objetivo de compensar a queda de 6 dB/oitava no espectro do sinal de voz. Outro pré-processamento utilizado é a normalização do sinal pela extração do valor médio.

Dependendo da aplicação a próxima etapa seria a de **segmentação**. Esta é uma das tarefas mais difíceis de realizar em forma automática. Um primeiro nível de segmentação implicará a remoção de silêncios ao começo e ao final de cada fonema. Outro nível seria a extração de palavras específicas de uma frase. Um último nível de segmentação, e talvez o mais difícil, sería a extração de fonemas determinados de uma palavra ou similarmente a separação de uma palavra em seus fonemas. A dificuldade provém do fenômeno de coarticulação, que poderia ser descrito elementarmente como a distorção na pronúncia de um fonema devido ao contexto no qual se encontra.

Como próxima etapa realiza-se, em geral, uma **Extração de características** (FFT, LPC, Cepstrum). Para tal fim, segmenta-se o sinal de voz em trechos de uma duração tal que se considera que o sinal de voz é estacionário. Tipicamente se escolhe uma duração de entre 10 e 30 ms com uma superposição entre janelas adjacentes de 5 a 15 ms. Estas janelas de voz são multiplicadas por janelas do tipo Hamming ou Triangulares para reduzir o efeito de borda na estimação. Dentro das características mais usadas no análise de voz podemos citar:

a) Parâmetros de natureza **temporal**, como medidas de energia do sinal e/ou de cruzamento por zero.

B) Parâmetros de natureza **espectral**, dentro desta categoria existem diversos mecanismos para sua obtenção. Os mais populares são.

DFT: Implica a utilização da transformada discreta de Fourier no cálculo da distribuição em freqüências da energia do sinal. Utiliza-se na prática o algoritmo rápido FFT para seu cálculo. Sua vantagem é a velocidade de cálculo, apesar de ter uma baixa compressão da informação. Tipicamente é utilizado como técnica acessória de outras como os cepstrais.

LPC: Baseia-se no cálculo dos coeficientes de predição linear da voz. Estes coeficientes são a solução de um sistema de equações lineares na matriz de autocorrelação do frame. Para seu cálculo rápido utiliza-se o algoritmo de Levinson-Durbin. Tipicamente para o reconhecimento de voz utilizam-se entre 10 e 14 coeficientes. É extensamente utilizado na análise de voz e na codificação de voz. É de rápido cálculo, mas foi perdendo popularidade no reconhecimento de voz, por duas razões: não permite o desajuste da excitação do trato fonador e é difícil para incorporar aspectos da percepção humana, como a escala MEL.

Cepstrais: calculados com técnicas de processamento Homomórfico, são extraídos de uma estimação do espectro do sinal de voz. É a caraterística mais popular e a que_maior índice de reconhecimento produz. Permite em primeiro lugar o desajuste entre a excitação e o trato fonador (com o qual é extraído uma estimação da forma do trato sem influência da excitação) e a incorporação de aspectos perceptivos. Para isto definem-se os Mel-Cesptrum, onde a escala de freqüência é pesada pela escala percepção MEL.

A fórmula para o cálculo dos cepstrais a partir do espectro DFT por um banco de janelas triangulares é:

$$C(i) = \sum_{k=1}^{Nw} \log X_k \cos(i(k - \frac{1}{2})\frac{\pi}{Nw}), \quad i = 1,2,...,Nc \tag{4.11}$$

onde Nw é o número de janelas triangulares, Nc é o número de coeficientes cepstrais e X_k é a energia do k-ésimo filtro.

O processamento espectral da voz esta baseado numa modelização do ouvido humano como filtros do tipo passa faixa. Apresentamos na figura seguinte a anatomia do ouvido humano.

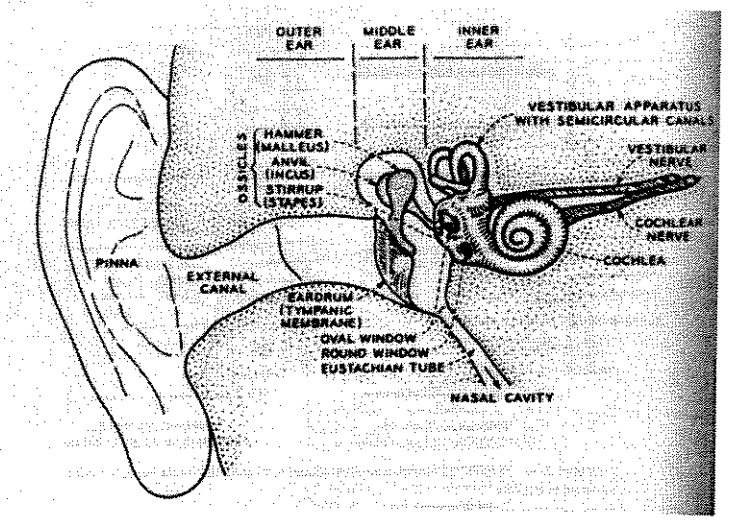


FIGURA 4.5 - Anatomia do ouvido humano

O ouvido humano pode ser divido em três seções. A primeira seção esta formada pelo canal auditivo e o tímpano. O canal auditivo introduz uma ressonância em torno dos 3000 Hz. A tímpano vibra com o som que sai do canal auditivo. A segunda seção esta formada por um conjunto de ossos, cuja função é a transformação das vibrações do tímpano na vibração do líquido interno da cóclea. A cóclea e os nervos auditivos formam a terceira seção. A vibração do líquido interno da cóclea é transformado em sinais elétricos pelos nervos auditivos. O processamento feito pelos nervos auditivos pode ser aproximado a um conjunto de filtros do tipo passa faixa.

Como última etapa está o **Classificador** que finalmente permite o mapeamento da voz à classes préviamente estabelecidas. A classificação implica o uso de uma medida de comparação entre as características para a atribuição do padrão a uma determinada classe. Como os modelos de redes neurais para o reconhecimento de voz serão detalhadas numa próxima seção, detalharemos a seguir outras técnicas com propriedades muito importantes:

DTW: (por **D**ynamic **T**ime **W**arping) é um dos métodos mais antigos de reconhecimento de voz. Baseia-se no reconhecimento mediante o cálculo das distâncias acumuladas entre a palavra incógnita e as palavras de referência armazenadas como base de dados. Requer a definição de uma medida de distância de acordo com o tipo de extração de caraterística que se esteja utilizando. A clave que popularizou este método é a solução ao

problema do alinhamento linear que não é capaz de compensar as oscilações e a duração de cada sílaba ou fonema, confrontando assim trechos diferentes de palavras. A maneira proposta é um alinhamento dinâmico no tempo (não linear), onde as trajetórias são calculadas de forma a minimizar um critério de distância acumulada, alinhando automáticamente os padrões. Na próxima figura apresenta-se um exemplo de uma trajetória calculada com DTW.

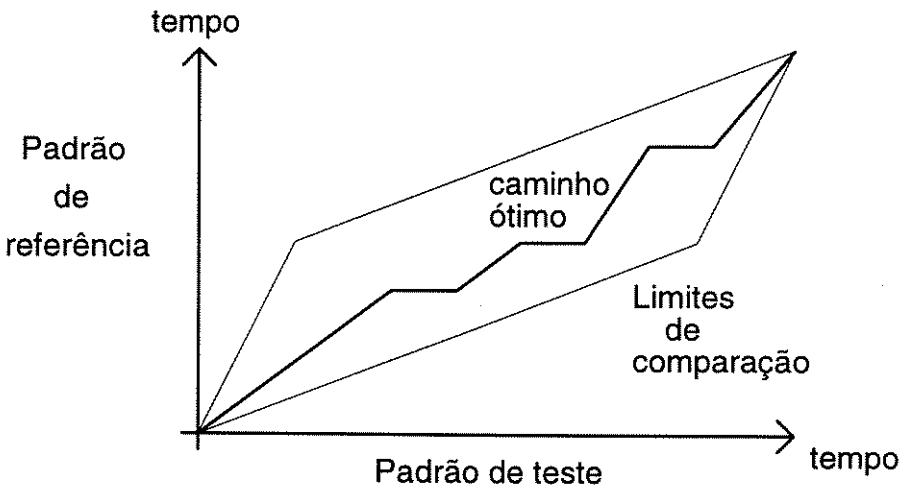


FIGURA 4.6 - Alinhamento dinâmico de tempo

Na figura 4.6 apresenta-se um exemplo de alinhamento entre um padrão de referência e um de teste, para pontos iniciais e finais semelhantes. Os limites de comparação correspondem a uma restrição global da máxima e mínima compressão que podem ter as palavras. O caminho apresentado no interior é o caminho ótimo, que corresponde ao caminho de menor distância acumulada. É importante destacar que como há uma restrição global, também há uma local, no sentido que os caminhos de distância acumulada (um deles é o ótimo) devem ser sempre crescentes.

Sua vantagem reside em que é muito simples de calcular e de implementar, devido a que o treinamento é um simples armazenamento de padrões, e tem um aceitável índice de reconhecimento. De fato, muitos sistemas comerciais utilizam esta técnica para sistemas de reconhecimento de palavras com vocabulários limitados e poucos usuários. Sua maior limitação aparece num vocabulário de dimensões importantes e múltiplos usuários, onde a quantidade de padrões de referência cresce consideravelmente e o tempo de cálculo de cada novo padrão com a base de dados se torna proibitivo.

HMM: (por **H**idden **M**arkov **M**odel) É o modelo mais utilizado atualmente para reconhecimento de voz. Baseia-se numa modelização de emissão de símbolos em estados, através de um duplo processo estocástico. O primeiro processo estocástico é que regra a transição entre estados, sendo modelado por uma probabilidade de permanecer no estado ou de passar ao seguinte, e o segundo processo estocástico é que regra a produção de símbolos em cada estado. Se bem este é um modelo de geração de símbolos, utiliza-se no reconhecimento calculando a probabilidade de que uma seqüência dada tenha sido gerada por um modelo.

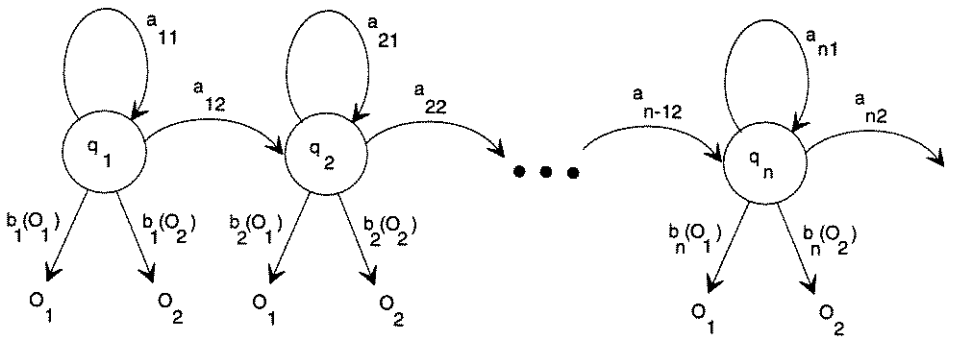


FIGURA 4.7 - Hidden Markov Model de n estados e duas observações

Na figura 4.7 se apresenta um HMM, onde a_{ik} é a probabilidade de passar do estado i ao k, e $b_j(O_k)$ é a probabilidade de observação do símbolo j no estado k.

O Hidden Markov Model é uma generalização do DTW, onde em vez de padrões de referência se compara com modelos estatísticos desses padrões de referência. Para calcular a probabilidade que un HMM tenha gerado uma seqüência a serem reconhecida, usa-se o algoritmo de Viterbi, que se apresenta a seguir:

$$\begin{aligned} &Dmin(i,t) = \max_{j=i-k,i} \{Dmin(j,t) + \log(a(j,i))\} + \log(b(i,t)) \\ &\text{com } i = 1{:}I \text{ e } t = 1{:}T \end{aligned} \tag{4.12}$$

Onde Dmin é a verosimilitude acumulada, k é o número de predecessores do estado (tipicamente k=1), I é o número de estados e T é a longitude do padrão. Si achamos a Dmin(I,T) para um modelo, podemos encontrar o caminho ótimo se invertemos o caminho desde esse ponto. Esta Dmin(I,T) é uma estimação de P(y/M), a probabilidade que o modelo y tenha gerado a palavra M. Portanto, como para modelos de diferente número de estados e padrões de longitude T, o caminho ótimo tem uma longitude T para todos os

modelos, as P(y/M) estimadas podem ser comparadas. A classificação final do modelo que gerou a palavra que se deseja reconhecer é feita como

$$y^* = \underset{y}{\operatorname{argmax}}\{P(y/M)\} \tag{4.13}$$

Os HMM tem grandes vantagens no sentido que, de forma semelhante as redes neurais, modelam a informação presente na base de dados. Neste sentido superam o método de comparação com padrões de referência o que permite sua aplicação a grandes bases de dados e a problemas de reconhecimento muito mais complexos. Sua principal desvantagem é sua falta de discriminação já que cada modelo de palavra está treinado com exemplos de sua classe unicamente, não existindo garantia que outro modelo da classe incorreta gere uma saída maior. Apesar de realizar-se importantes esforços para superar esta falência, através de um "treinamento discriminante", a origem do problema é insuperável já que em se mesmo o modelo não é discriminante. Mesmo que desde o ponto de vista do reconhecimento, esta possa ser uma falência, desde o ponto de vista do treinamento não é, já que cada modelo é treinado com uma pequena porção da base de dados, o que permite um rápido treinamento para bases de dados muito grandes.

Os modelos de Markov são os de melhor desempenho em geral para os problemas de reconhecimento de voz. Atualmente uma das grandes linhas de investigação é a de suprir as falências dos modelos de Markov com redes neurais do tipo perceptron multicamada.

4.3.3 - Diferentes problemas no reconhecimento de voz

É importante distinguir no reconhecimento de voz diferentes categorias:

Podemos desejar o reconhecimento de **palavras isoladas ou fala contínua**. Isto é se as palavras são pronunciadas com pausas controláveis entre elas, isto é uma por vez, ou se são pronunciadas em forma natural. Um exemplo do primeiro caso seria o reconhecimento de dígitos ou de ordens a um sistema de guia de um veículo. Do segundo caso seria o ditado de cartas automáticas uo sistemas de tradução de idioma automáticos.

Dentro do reconhecimento de palavras devemos diferenciar o **vocabulário restringido** ou **livre**. Isto é, se o conjunto de palavras pertence a um conjunto especificado como pode ser um conjunto de palavras, ou se o falante pode pronunciar qualquer palavra. Para este caso se utiliza o reconhecimento fonético e posteriormente se constróem as palavras pronunciadas.

Se o sistema é **dependente ou independente do locutor,** no caso em que o usuário é o mesmo com o que foi treinado o sistema, ou se é qualquer outro, respectivamente. Claramente o último problema é muito mais difícil de resolver por seu requerimento de um número muito maior de amostras de voz para obter uma boa generalização.

4.3.4 - Bases de dados de fonemas - diferentes estratégias

Como em quase todas as aplicações de redes neurais a problemas de reconhecimento de padrões, precisa-se de uma **base de dados** com padrões específicos, neste caso os padrões a reconhecer são emissões de voz. Como detalhou-se em 4.3.1 existem muitas tarefas possíveis no reconhecimento de voz, dependendo de qual se está desenvolvendo (reconhecimento de vogais, palavras, dígitos, etc), os padrões serão determinados. Mas independentemente do tipo de emissão de voz com o qual estejamos trabalhando, existem algumas particularidades das bases de dados específicas desta aplicação.

Entre algumas podemos destacar as seguintes: Na especificação da aplicação do sistema de reconhecimento de voz está o caso de qual vai a serem o grupo entre os **distintos falantes** (masculinos, femininos, crianças) com os quais vai se trabalhar. Portanto é importante em quanto à representatividade da base de dados, que esta reflita as características particulares do grupo. É conhecido nos ambientes de estudos fonéticos que as falas de cada um destes grupos é qualitativamente diferente, sendo a voz das crianças mais parecida a das mulheres que a dos homems. Outro aspecto a levar em conta é o **idioma** (português, espanhol, inglês, etc). Isto determina por exemplo a quantidade de fonemas com os quais se vai trabalhar. Também deve-se levar em conta aspectos específicos acerca da **representatividade** da base de dados. Por exemplo, quanto à forma de falar de uma zona determinada. Além disso, deve-se levar em conta particularidades da **aplicação.** Como por exemplo na aplicação de reconhecimento de dígitos emitidos através do canal telefônico, é importante que a base de dados seja registrada através deste meio. Além disso deve-se levar em conta que a maior quantidade de amostras, melhor o reconhecimento das características do falante, por isso pede-se a **repetição de emissão** (1, 2, até 5 vezes).

4.3.5 - Organização hierárquica da fala

Apresenta-se a seguir uma organização hierárquica da fala, particularmente útil para o reconhecimento de fala contínua. Isto permite distinguir diferentes níveis do

reconhecimento de voz e compreensão da mensagem que se deseja transmitir, sendo muito útil, já que, partindo dos níveis inferiores, cada nível reflete um grau maior de abstração ao respeito da mensagem involucrada no sinal de fala. Um diagrama desta hierarquia apresenta-se na figura 4.8.

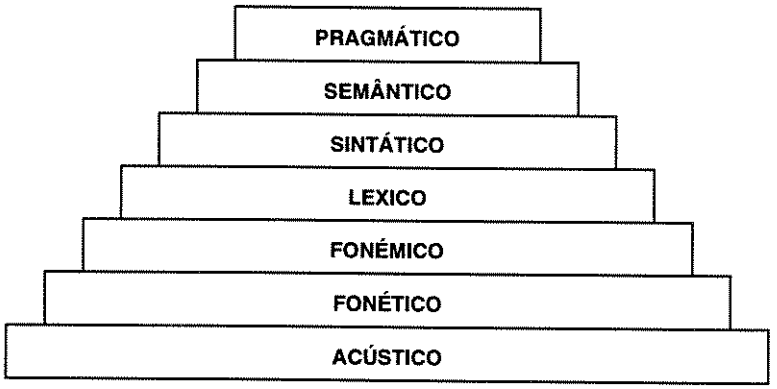


FIGURA 4.8 - Organização hierárquica da fala

Esta hierarquia começa num nível **acústico**, onde se representam parâmetros e suas variações factíveis de ser medidas do sinal sonoro; estes podem ser espectrais, de energía, de variação de freqüência tanto glótica como de formantes, duração, intervalos de silêncio, etc

O nível imediatamente superior na hierarquia é o **fonético**, que requer a segmentação da cadeia de fala contínua, sendo o som de uma frase nos elementos mais simples com valor semântico denominados fonemas. A quantidade de fonemas varia muito de acordo com a língua, região e locutor. Em geral, a quantidade total de fonemas de uma língua encontra-se entre 20 e 60.

O nível seguinte é o **fonêmico**. Este refere-se a determinadas regras elásticas de interação de fonemas. Esta interação implica que um fonema pronunciado num determinado contexto, pode ser modificado de forma tal que o som emitido pertença a outro fonema. Um exemplo em espanhol poderia ser: a oclusiva /t/ de "ritmo" se transforma as vezes em /d/ de "ridmo", ao ficar nasalizada pela nasal /m/.

O próximo nível é o **lexico**. Neste agrupam-se os fonemas dentro do léxico admitido ou "dicionário". Quanto menor é o número de vocábulos de um dado léxico, tanto mais fácil é achar as regiões delimitadas por certos elementos discriminatórios que permitem separar as palavras. Neste nível delimitam-se as possíveis combinações de fonemas dentro de um certo conjunto.

Este léxico está fortemente definido pelo tipo de aplicação na qual vai-se desenvolver o reconhecedor de voz. Em algumas aplicações de laboratório pode-se delimitar o conjunto de vogais. Em aplicações mais práticas, por exemplo para sistemas bancários pode-se delimitar em dez dígitos e alguns poucos fonemas a mais. Outras aplicações como os sistemas de reserva automática de passagems requer de um léxico um pouco maior. Então a medida que aumenta a complexidade da tarefa o número permitido de fonemas é maior.

O nível seguinte é o **sintático**, que é onde se tomam decisões acerca das regras gramaticais da linguagem que tem a ver com a construção de frases, tais como que o artigo preceda ao nome, etc.

Este nível de análise é muito usado pelos humanos já que é limitativo e portanto redundante. Nos permite distinguir a linguagem educada da linguagem "tarzaniana", e também nos permite reconhecer a linguagem de uma pessoa estrangeira com poucos conhecimentos do idioma, já que em ambos casos as regras da sintaxe não são corretamente utilizadas.

O seguinte nível é o **semântico** onde analisa-se o sentido das frases já que se parte que ele contém uma mensagem inteligível para os interlocutores. Uma frase sintáticamente correta pode não ter significado aceitável. Uma exceção à regra anterior são as composições dos poetas, onde num primeiro nível de análise não exista um significado, porém logo de analisar mais profundamente a mensagem transmitida podem ser mais sugerente ou provocadora.

Por último o nível **pragmático** é onde se analisam frases que para ter um valor semântico requerem de frases anteriores.

4.4 - REDES NEURAIS E RECONHECIMENTO DE VOZ

4.4.1 - Introdução e antecedentes

Uma das primeiras motivações para a criação e desenvolvimento das redes neurais artificiais foi a classificação de padrões tal como apresentou-se no capítulo 2. Isto é, separar o espaço de estados por fronteiras que delimitam cada uma das classes.

Uma das principais aplicações das redes neurais artificiais, é o *reconhecimento de voz*, naturalmente tratando de emular a extraordinária capacidade humana de reconhecer vozes ainda em condições extraordinariamente desfavoráveis. Além disso, devido a sua capacidade de aprendizagem automático e generalização diretamente a partir de amostras de fala. Isto implica que são capazes de descobrir regularidades e relações a partir de exemplos em forma automática e usa - los para classificar. Tudo isto permite obter um excelente desempenho de reconhecimento.

Porém a aplicação destes modelos matemáticos na classificação de emissões de voz foi tardia. Entre os primeiros antecedentes pode-se mencionar os trabalhos de J. Burr em 1986 [5] onde aplica o perceptron multicamada ao reconhecimento de fonemas em idioma inglês, mais concretamente na classificação do chamado "E-set", as letras que em inglês ao ser pronunciadas contem uma "e": /b/, /c/, /d/, /e/, /p/, /t/, /g/, /f/. O destacável é que este conjunto é altamente confuso e os resultados de discriminação são muito bons. Outro antecedente importante é o de Kohonen em sua famoso livro "Self Organization and Associative Memories" [34], cuja primeira edição é de 1980, onde apresenta a teoria das redes neurais auto-organizadas e a teoria de aprendizagem não supervisionada. Uma das aplicações apresentadas é o reconhecimento de fonemas. Atualmente existem muitos investigadores dedicados ao estudo desta aplicação e inclusive congressos muito específicos tratam somente deste assunto.

Um trabalho que marcou um etapa é o de Robinson & Fallside com sua "Recurrent Error Propagation Network" [7], onde melhorou o índice de reconhecimento, sobre uma mesma base de dados muito conhecida (TIMIT), dos modelos ocultos de Markov (Hidden Markov Models), que é a técnica estatística até agora mais utilizada para o reconhecimento de voz.

4.4.2 - Distorções mais importantes dos padrões de voz

Assim como se mencionou as fontes de distorção mais importantes dos padrões em forma genérica, é importante destacar as distorções mais importantes dos padrões de voz. Por um lado isto dará uma aproximação adequada para visualizar a dificuldade desta tarefa, e por outro lado para compreender como cada estrutura, a serem apresentada logo, enfrenta as possíveis distorções.

Uma das distorções que primeiro são tomadas em conta são as de ***amplitude***. As pessoas utilizam diferentes amplitudes de voz, segundo seu sexo, idade, condições ambientais e outras. Mas este tipo de distorção é mais importante do que parece. A

diferença de amplitude de um mesmo fonema pronunciado em diferentes palavras, se dão em função dos fonemas no seu contexto. Inclusive existe uma diferença de amplitude de um mesmo falante pronunciando a mesma palavra. Com os tipos de caraterística dos sistemas de reconhecimento utilizadas na atualidade (LPC e Cepstrum), este tipo de distorção pode-se minimizar significativamente.

Outro tipo de distorção importante é a distorção do padrão de ***freqüência***. Em parte são causadas por distorções de amplitude, mas existem outras fontes deste tipo de distorção. O conteúdo de energia espectral depende muito da forma do trato fonador bocal e nasal. Tanto diferenças anatômicas entre as pessoas, como outras características do trato como a quantidade de saliva, resultam em conteúdos espectrais de enorme variação, ainda que para o mesmo fonema. As ferramentas matemáticas de extração de características espectrais tratam de compensar esta variabilidade com relativo sucesso.

Outro tipo de distorção importante e próprio da natureza da voz é o ***temporal***. A duração dos fonemas é de grande variação, por um lado dependendo da palavra na qual são pronunciados, de sua posição dentro da palavra, e por outro lado dependendo do conteúdo emocional que se deseja dar à mesma. Tudo isso provoca tanto compressões como dilatações nos fonemas.

Entre as distorções próprias da natureza humana podemos acrescentar, a ***coarticulação***. Este fenômeno explica-se como a modificação na pronúncia de um fonema dependendo do contexto no qual é pronunciado. Para o reconhecimento de palavras esta distorção não é tão importante, porque a base de treinamento está formada por exemplos destas palavras, mas para o reconhecimento de grandes vocabulários, onde a base de treinamento é fonética, esta distorção provoca uma séria deficiência dos sistemas de reconhecimento independentes do contexto. Também podemos acrescentar a ***omissão*** de determinados fonemas.

As ***falas locais*** também produzem distorções importantes nas características da fala. Finalmente mencionaremos as diferenças entre a ***fala masculina, feminina e infantil***. Um fenômeno conhecido é que a menor freqüência glótica melhora a estimação espectral da voz. O aumento da freqüência glótica nas mulheres e as crianças provoca uma dificuldade maior na estimação do seu espectro e portanto de seu reconhecimento. Tanto assim que os sistemas de reconhecimento modernos separam os modelos de reconhecimento dos homems e das mulheres.

4.4.3 - Diferentes estruturas propostas

É importante apresentar algumas (entre as conhecidas pelo autor) das mais importantes arquiteturas de aplicação das redes neurais artificiais ao reconhecimento de voz. É impossível referir-se a todas já que em cada congresso se apresentam novas formas de aplicação, além disso existem alguns compêndios mais específicos onde dirigir-se em caso de necessidade de maior profundização como o de Deller e Proakis [31]. O propósito deste capítulo é simplesmente o de comparar arquiteturas ressaltando alguns aspectos de interesse como sua robustez à distorções genéricas e às próprias da voz, sua adequação ao reconhecimento de fonemas complexos e outros. Mencionaremos exclusivamente as referidas a perceptrons multicamada. Detalharemos, dentro do possível, suas vantagens e desvantagens. Mencionaremos primeiro as arquiteturas onde as redes neurais estejam em forma exclusiva e logo, brevemente, as outras arquiteturas onde estejam em conjunto com outras técnicas estabelecidas de reconhecimento de voz como os HMM.

Grande parte das arquiteturas apresentadas, baseiam-se na extração de características espectrais de curto prazo para a classificação de padrões de voz. Entre estas podem-se destacar as Energias de banco de filtros, FFT, LPC ou Cepsturm, podendo ou não estar combinados com a escala de percepção MEL. No caso que a arquitetura utilize características de outro tipo, será devidamente explicitada.

4.4.3.1 - Arquiteturas exclusivas com redes neurais:

- **RECONHECIMENTO DE PADRÃO ESPECTRAL ESTÁTICO:**

Nesta arquitetura se apresenta o fonema ou palavra que se deseja reconhecer como entrada na forma de um padrão de várias características espectrais sucessivas. Isto é, se reconhece uma janela de tempo fixa de duração de vários frames. Esta arquitetura usa-se conjuntamente com um algoritmo de alinhamento no tempo, isto é, de detecção de princípio e fim de palavra, afim de alinhar a palavra nas entradas. A entrada pode ser unidimensional ou bidimensional, ver Burr D. J. [13], e vai ter o tamanho dado pela longitude do padrão a classificar (para LPC tipicamente se utilizam entre 12 e 14 coeficientes) multiplicado pela quantidade de frames da janela de tempo. Para cada padrão que se apresenta na entrada, a saída da rede é única correspondendo tipicamente uma por cada palavra diferente que se deseja classificar.

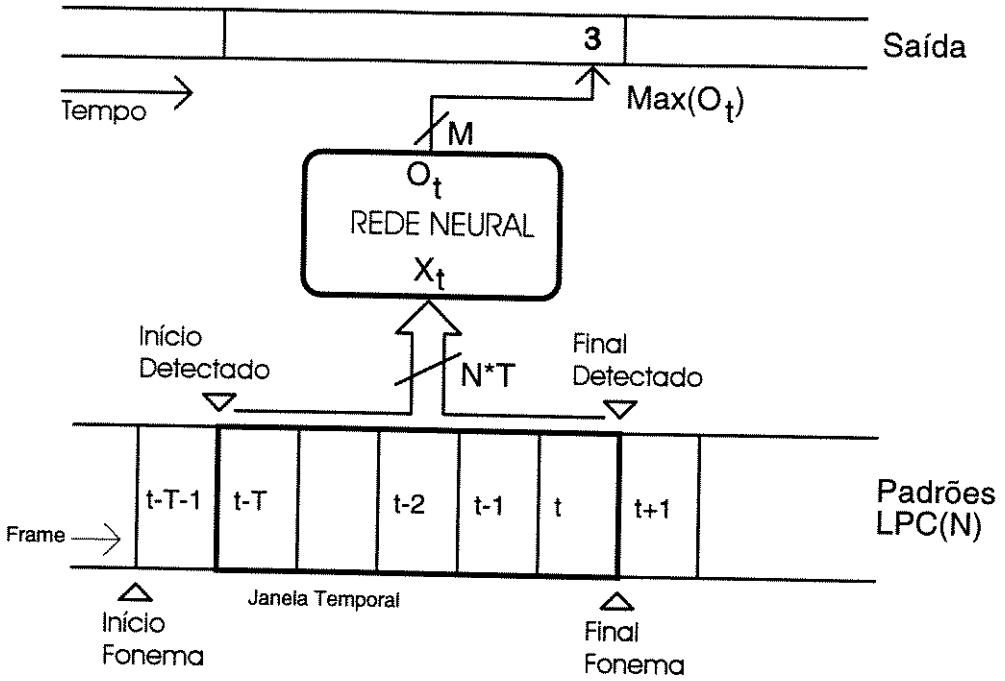


FIGURA 4.9 - Reconhecimento Estático

No gráfico da figura 4.9 apresenta-se um exemplo para a rede de reconhecimento estático de reconhecimento de uma palavra, neste caso o dígito 3. Se utilizam como características temporais os padrões LPC, que estão representados sucessivamente no tempo, correspondendo-se com cada frame, em forma de retângulos verticais com uma indicação do índice temporal. Se apresenta além disso uma indicação do começo e fim reais do fonema e também os inícios e fim ***detectados*** por um algoritmo de detecção. Observa-se na figura um caso possível de ocorrer na prática, onde o fim do fonema foi corretamente detectado mas não o princípio do fonema. A entrada da rede neural é um vetor $\mathbf{X_t}$ formado por uma ***janela temporal*** (apresentada com linha grossa) de T vetores com N valores (a dimensão do vetor de coeficientes LPC) cada uma, o qual indica uma dimensão total do vetor de N*T. A saída da rede, que ao ocorrer no tempo t está alinhada com o frame do tempo t, é um vetor de M dimensional (onde M é o número de classes). A determinação da classe a qual pertençe o padrão realiza-se pelo cálculo do mínimo erro relacionado aos vetores desejados da classe. Como os vetores desejados são os vetores diretores dos eixos cartesianos, a determinação da classe pode-se realizar pelo cálculo do máximo do vetor de saída $\mathbf{O_t}$, o que é semelhante a calcular a maior projeção sobre cada eixo. Na parte superior apresenta-se um vetor de indicação de classes ao longo do tempo e sua correspondente validez temporal, neste caso a duração da janela temporal.

VANTAGENS: permite o reconhecimento de palavras, quer dizer, cadeias de fonemas diferentes; estabelece relações temporais; extrai relações de contexto.

DESVANTAGEMS: o tamanho da rede é excessivo o que aumenta o custo computacional; janela fixa de tempo; não suporta distorções temporais como translação e dilatações; dependência do alinhamento estrito do padrão; baixa eficiência da base de dados (uma palavra gera um único padrão de treinamento); um aumento no número de características multiplica o tamanho do padrão.

- **RECONHECIMENTO DE PADRÃO ESPECTRAL PURO:**

A entrada da rede é do tamanho das características espectrais e a saída pode ser tanto uma indicação de pertencer a uma classe como uma estimação da probabilidade de emissão, ver Robinson T. e Fallside F [15]. Isto é, independentemente do tempo se reconhece o padrão espectral do fonema. Existem duas variações desta arquitetura onde a rede é estática, ou a rede é dinâmica. Este padrão espectral pode ser tanto uma estimação do espectro de potência como os coeficientes de predição linear, que é extraído frame a frame, como um vetor de saídas temporais de filtros passa-faixa, usualmente espaçados em freqüência segundo escalas perceptuais.

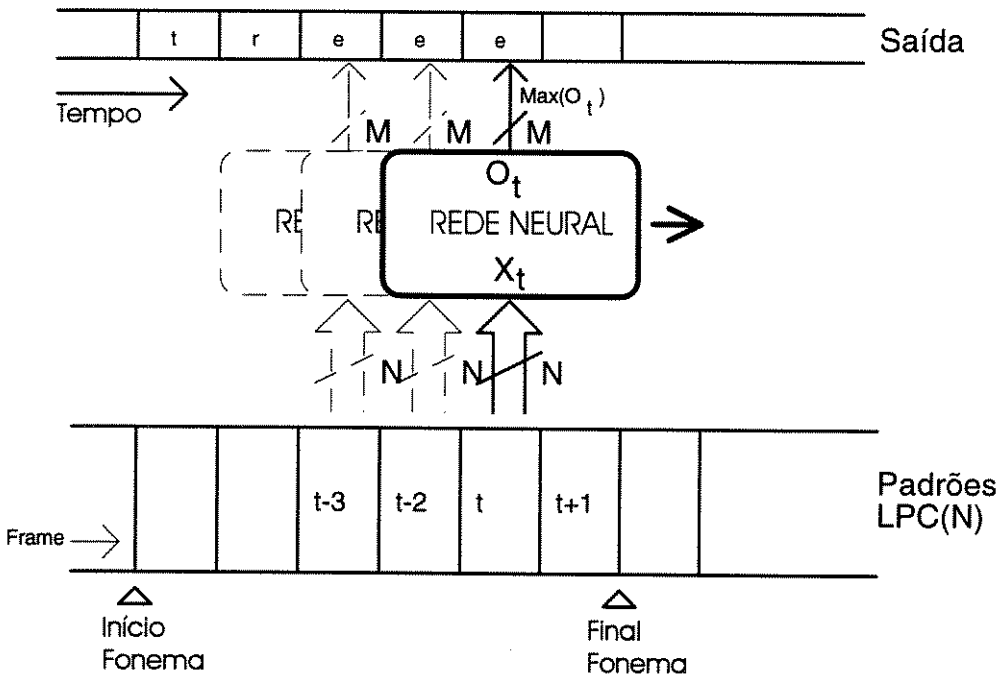


FIGURA 4.10 - Reconhecimento Espectral

No gráfico da figura 4.10 apresenta-se um exemplo para a rede de reconhecimento espectral de reconhecimento de uma palavra, de novo é o caso do dígito 3. Como para este caso a rede é de tipo estática (sem realimentação) somente reconhece fonemas, pelo qual as saídas são as letras da palavra /três/. Utilizam-se novamente como características temporais

os padrões LPC. A entrada da rede neural é o vetor $\mathbf{X_t}$ que neste caso está formado únicamente pelos N valores correspondentes ao vetor de coeficientes LPC de um único frame, o qual da uma dimensão total do vetor de N. A saída da rede correspondente a cada frame, é novamente um vetor M dimensional. A determinação da classe que pertence o vetor realiza-se do mesmo jeito que para o caso anterior. Na parte superior apresenta-se um vetor de indicação de classes e sua correspondente validez temporal, que neste caso é cada frame. Apresentam-se correspondendo a cada frame as saídas produzidas. Observa-se as diferentes durações de cada fonema, podendo ser de poucos frames no caso da /t/ e a /r/ ou de vários frames como a /e/.

VANTAGENS: permite a flexibilização do tamanho das características o qual implica uma melhor discriminação; independência do tempo; rede pequena em comparação com 1); suporta distorções de dilatação e translação; alta eficiência de base de dados já que de um mesmo fonema se podem extrair muitos padrões de treinamento. Estabelece relações temporais (rede dinâmica).

DESVANTAGEMS: não estabelece relações temporais (rede estática); não suporta distorções contextuais (rede estática); dificuldade para o reconhecimento de palavras.

- **RECONHECIMENTO DE PADRÃO ESPECTRAL DINÂMICO:**

Se apresenta à rede ao longo do fonema sucessivamente vários padrões espectrais e somente deseja-se que ao final do fonema apresente na saída a indicação da classe. A ideia é forçar a aprendizagem de correlações temporais entre os padrões espectrais. São exponentes típicos desta arquitetura as redes de Waibel A. chamadas "Time Delay Neural Networks", [14] e as de Principe *et al.* chamadas "Gamma Nets", [33].

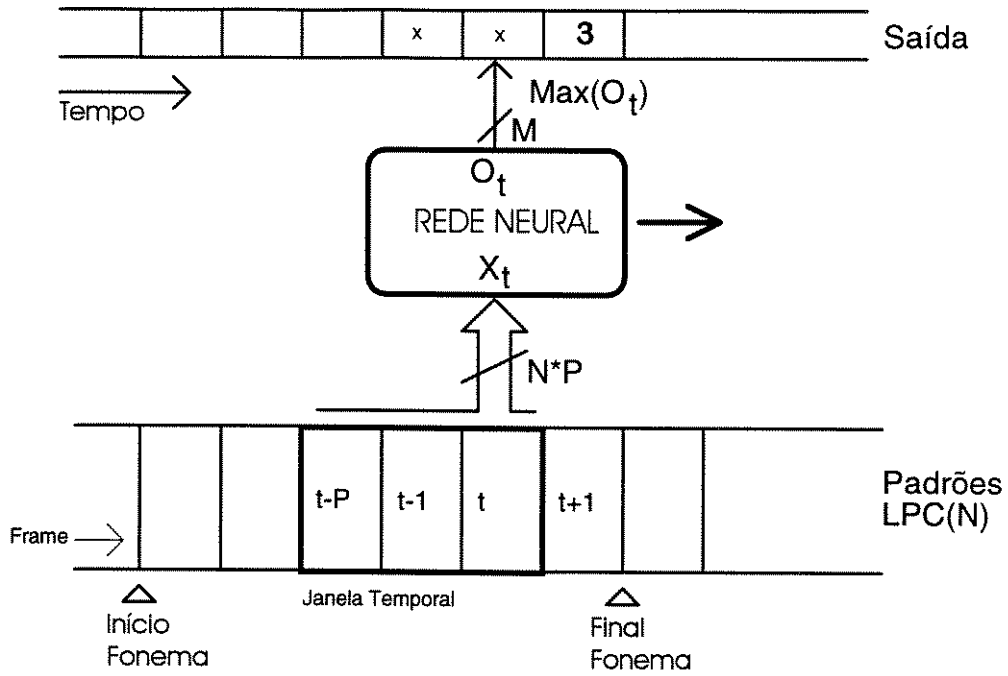


FIGURA 4.11 - Reconhecimento TDNN

No gráfico da figura 4.11 se apresenta um exemplo para a rede de reconhecimento dinâmico de uma palavra, de novo é o caso do dígito 3. Utilizam-se novamente como características temporais os padrões LPC. A entrada da rede neural é um vetor $\mathbf{X}_t$ formado por uma ***janela temporal*** (linha grossa) de P vetores com N valores (a dimensão do vetor de coeficientes LPC) cada um, o qual da uma dimensão total do vetor de P*T. A saída da rede é significativa somente ao final do fonema (pelo qual nos frames anteriores mapeia-le uma "x" que significa "não importa"), é novamente um vetor de M dimensional. A determinação da classe de pertença do vetor realiza-se da mesma forma que nos casos anteriores. Na parte superior apresenta-se um vetor de indicação de classes e sua correspondente validez temporal, que neste caso é toda a duração do fonema.

VANTAGENS: seu tamanho é intermediário entre os dois apresentados anteriormente; independência do tempo já que não precisa alinhamento; suporta distorções; permite o reconhecimento de palavras; aprende correlações temporais; janela regulável de tempo (Gamma Nets)

DESVANTAGEMS: precisa altos tempos de treinamento; baixa eficiência da base de dados; relativa independência do tamanho das características; janela fixa de tempo (TDNN).

• RECONHECIMENTO POR ESTIMAÇÃO NÃO LINEAR :

Esta arquitetura baseia-se na capacidade da rede neural de predizer as amostras futuras de séries temporais caóticas [20], [32]. Existem duas variações desta arquitetura onde a estimação se faz sobre as amostras temporais do fonema, Tishby N, [40] ou onde a predição se faz sobre os padrões espectrais, Levin E. [41]. Em ambos casos usa-se a rede como uma generalização não linear de um preditor. Se usa como índice de reconhecimento o erro de predição de cada um dos modelos das classes.

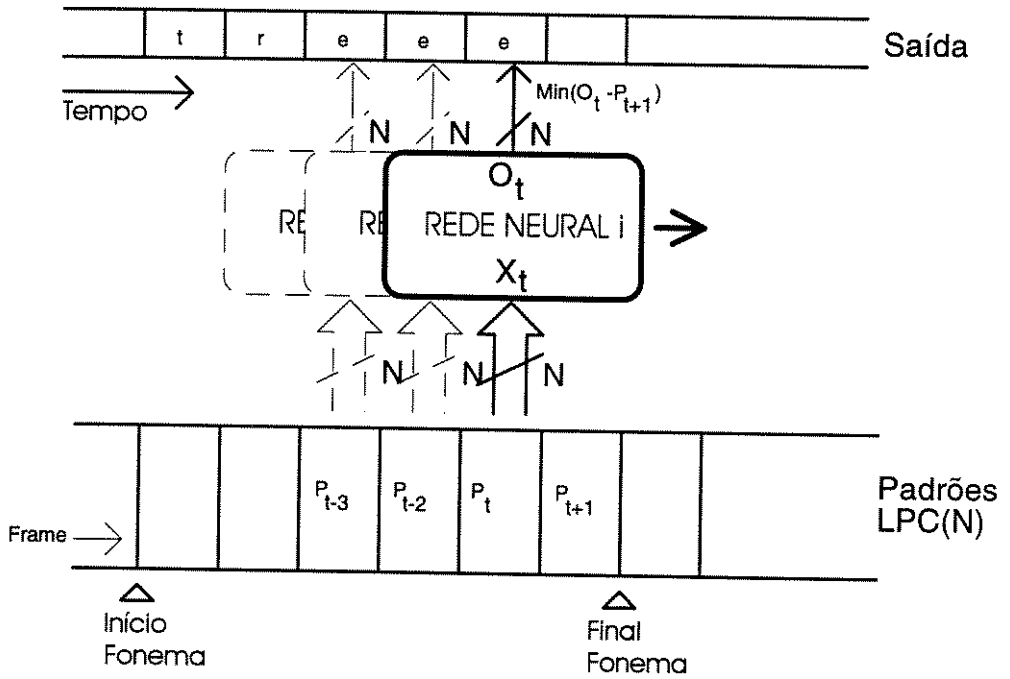


FIGURA 4.12 - Reconhecimento por predição não linear

No gráfico da figura 4.12 apresenta-se um exemplo para a rede de predição não-linear de reconhecimento de uma palavra, novamente é o caso do dígito 3, e utilizam-se novamente como características temporais os padrões LPC. A entrada da rede neural é um vetor $\mathbf{X_t}$ formado por vetores com N. A saída da rede, no tempo t, é como se explicou anteriormente, um vetor N dimensional, que é uma estimação não linear do vetor de coeficientes LPC no tempo t+1, quer dizer P_{t+1}. A determinação da classe pertencente ao vetor realiza-se, à diferença dos casos anteriores, através do cálculo do mínimo erro de estimação entre várias redes, neste caso apresenta-se a rede neural **i.** Na parte superior se apresenta um vetor de indicação de classes e sua correspondente validez temporal, que neste caso é toda a duração do fonema.

VANTAGENS: modelos detalhados; suporta distorções; permite reconhecimento de palavras (Levin).

DESVANTAGEMS: precisa alinhamento (Levin); grande custo computacional ao processar amostras (Tishby); baixa discriminatividade por modelos individuais das classes; altos tempos de treinamento.

4.4.3.2 - Arquiteturas híbridas com redes neurais:

Os sistemas de reconhecimento automático de fala mais avançados estão baseados em técnicas de modelação utilizando HMM. Apesar de proporcionar importantes melhoras no desempenho do reconhecimento automático nos últimos anos, os melhores níveis atuais de desempenho são ainda inadequados para a maioria das aplicações não triviais. Alguns estudos sugerem que a integração dos HMM com redes neurais poderia melhorar os índices de reconhecimento de fala.

O marco de HMM tem certas debilidades para seu uso em reconhecimento de fala, entre elas: pobre discriminação entre classes fonéticas na classificação baseada em segmentos de fala, e a necessidade de tratar como estatisticamente independentes certos processos que efetivamente não são independentes. As redes neurais por si mesmas também tem certas debilidades para seu uso em reconhecimento de fala, como ser sua inabilidade para tratar com a natureza seqüencial de fala e a necessidade de amostras de treinamento rotuladas para seu treinamento.

Os sistemas híbridos ANN-HMM de melhor desempenho baseiam-se na utilização da rede neural como estimador das probabilidades a posteriori da classe fonética q_j dado o vetor de entrada instantâneo Y, isto é $P(q_j/Y)$. A aproximação pelas redes neurais de probabilidades já foi desenvolvida no capítulo 2, seção 2.4.4. Dado que os HMM precisam, para o cálculo da probabilidade do modelo, as probabilidades a priori, se transformam as probabilidades a posteriori com a regra de Bayes. No reconhecimento se utiliza cada saída da rede como a probabilidade da classe fonética, prévia transformação em probabilidades a priori, se alimenta o modelo HMM para, mediante o algoritmo de Viterbi, calcular a probabilidade que o modelo tenha gerado a emissão presente.

Para o reconhecimento de grandes vocabulários treina-se a rede neural de forma tal que cada saída represente uma classe fonética.

Apresentaremos um algoritmo para o cálculo dos modelos de Markov com redes neurais. Divide-se o treinamento no cálculo das probabilidades de transição por um lado e o cálculo das probabilidades de observação por outro.

Para o cálculo das probabilidades de transição, utiliza-se o algortimo de Viterbi. No processo de cálculo da P(y/M), devem guardar-se os seguintes valores:

$$\begin{aligned} n(i,j) &\triangleq \text{numero de transições de i a j, } \forall i \\ n(i,\text{-}) &\triangleq \text{numero de transições desde i, } \forall i \end{aligned} \tag{4.14}$$

Logo de obter estes dados, para todo tempo, calcula-se as probabilidades da seguinte forma:

$$\begin{aligned} \hat{a}(i,j) &= n(i,j) / n(i,\cdot) \\ \hat{a}(i,i) &= 1 - \hat{a}(i,j) \end{aligned} \tag{4.15}$$

Para o cálculo das probabilidades de observação, se segmentam as palavras em unidades fonéticas. Cada segmento é usado como uma classe no treinamento da rede neural, que calcula as probabilidades a posteriori b(t,j). Com a regra de Bayes calculam-se as probabilidades a priori b(j,t), que são as que precisa o HMM.

Se a rede é independente de contexto, os fonemas diferentes são atribuidos a sua classe correspondente. Se a rede é dependente do contexto, procura-se diferenciar os fonemas segundo seu contexto. Por exemplo, no primeiro caso se diferencia as vogais puras /a/, /e/, /i/, /o/, /u/, no segundo caso se procura diferenciar as vogais em diferentes contextos [nasal]/a/[nasal], [plosiva]/a/[plosiva], [labial]/a/[labial], etc, e idem com as demais vogais.

Assim o modelo de Markov de cada palavra está formado pelas saídas da rede correspondentes aos fonemas da palavra. Os experimentos com este tipo de modelação tem superado aos HMM independentes do contexto para grandes bases de dados [35]. Atualmente se está experimentando com arquiteturas ANN-HMM dependentes do contexto.

4.5- RECONHECIMENTO DE VOZ RUIDOSA - DESEMPARELHAMENTO DE CONDIÇÃO

4.5.1 - Introdução:

É um tema específico da área de reconhecimento de voz e tem sido tratado com múltiplas estratégias, mesmo antes do desenvolvimento das redes neurais. Apresentaremos conceitos acerca do mesmo como o Condition Mismatch e posteriormente mencionaremos algumas propostas para lidar com este problema.

4.5.2 - "Condition Mismatch"

O fenômeno denominado por alguns autores [1] como "Condition Mismatch", algo assim como Desemparelhamento de Condição, significa que as condições nas quais foi realizada a aprendizagem não são as mesmas que as condições presentes durante o reconhecimento.

Este é um fenômeno muito comum na aplicação do reconhecimento de voz. As amostras de voz utilizadas no treinamento são recolhidas, quase sem exceção, em ambientes onde as perturbações estão ausentes (por exemplo salas a prova de som, com paredes isoladas acústicamente) ou estão controladas (gravações com um mínimo ruído de características conhecidas). Porém as aplicações típicas do reconhecimento de voz como as mencionadas em 4.5 implicam a presença de ruídos de distribuição estatística desconhecida e variável com o tempo.

Conseqüentemente, é um objetivo importante o desenvolvimento de mecanismos para que os sistemas de reconhecimento, não somente mantenham sua desempenho baixo condições emparelhadas mas que, além disso, não sofram uma dramática degradação quando o fenômeno de desemparelhamento de condição ocorre.

4.5.3- Estratégias propostas de solução

Podemos *dividir as aproximações ao problema do reconhecimento de voz ruidosa*, de acordo ao lugar *onde se aplica uma ferramenta de robustez* do diagrama clássico de classificadores (ver figura 4.4). As primeiras aproximações, como o realce ou pré-filtragem, procuram a eliminação do ruído no padrão prévia a extração de

características. Outras técnicas, como a EIH apresentada ao final desta seção, procura a estimação robusta de características baseada em modelos cocleares biológicos. As técnicas apresentadas por Mansour and Juang (1989) [1], dentro do marco da classificação por alinhamento dinâmico do tempo apresentam um cálculo robusto de distâncias para um tipo especial de característica. Finalmente Carlson B. *et al.* [37], baseando-se no trabalho anterior, apresenta um mecanismo para o cálculo robusto de probabilidades no marco dos HMM. Começaremos esta seção com um trabalho de Paliwal K. K. [39], onde se comparam as redes neurais com outras técnicas de classificação para o reconhecimento de voz ruidosa.

• No paper de Paliwal K. K. [39] se comparam as redes neurais "multilayered feed-forward Perceptron" (MLP) frente a vários classificadores clássicos (como Maximum Likelihood (ML) e K-Nearest Neighbor (kNN)) no reconhecimento de vogais inglesas afetadas por ruído branco, no caso independente do falante. Demonstra-se que o MLP proporciona, com os coeficientes cepstrais, um melhor índice de reconhecimento que os classificadores ML e kNN. Na seguinte tabela 4.2 se extraem os resultados da tabela 6 de [39]:

SNR (dB)	Classificador MLP	Classificador ML	Classificador kNN
∞	91.3	92.7	88.2
35dB	90.2	91.1	84.2
30dB	88.2	85.6	78.0
25dB	84.0	74.2	69.8
20dB	75.6	64.7	60.7
15dB	58.9	46.7	49.1

TABELA 4.2 Comparação de Classificadores, em % correto, extraídos de [39].

• Uma das técnicas mais populares e mais simples para adaptar modelos de bom nível de reconhecimento com sinais de baixo nível de ruído ao reconhecimento ruidoso, é o mecanismo de **imunização ao ruído**. Por imunização entende-se o treinamento de sistemas de reconhecimento com amostras de voz ruidosas. O ruído escolhido em cada aplicação depende do problema específico, mesmo que para casos gerais se usa ruído branco. Além da eleição do tipo de ruído deve-se escolher o nível do ruído com o qual modificar as amostras de voz que serão usadas no treinamento. Em geral utiliza-se uma mistura de vários níveis de ruído para cobrir muitas alternativas de contaminação. Este é um dos métodos mais diretos de reconhecimento ruidoso já que não implica modificar o sistema de reconhecimento disponível. Mas ao mesmo tempo, é o método mais vulnerável ao desemparelhamento de

condição, já que se tanto o tipo de ruído como seu nível diferem dos valores utilizados no treinamento o porcentagem de reconhecimento se verá muito afetada.

• Outro método importante é o **realce da voz** prévio ao seu reconhecimento. É importante destacar que dependendo da aplicação do sistema de reconhecimento de voz ruidosa, utilizam-se mecanismos específicos de aplicação. Concretamente em casos onde o ruído provenha de uma única e particular fonte será possível extrair informação acerca da estatística do ruído gerado colocando sensores ambientais e logo utilizar esta informação para realçar ("enhance") a voz. Quando o ruído modifica em forma aditiva o sinal de voz se podem usar métodos de cancelação adaptativa de ruído, utilizando duas fontes de sinal. Neste caso deve-se prover uma segunda entrada de sinal através de um ou mais microfones ambientais. Na figura 4.13 se apresenta um exemplo de um sistema de cancelação adaptativa de ruído, onde o sinal de voz é x(n), ó ruído somado é y1(n), o sinal observado é z(n), que é a soma do sinal de voz e o ruído, e finalmente uma segunda fonte de ruído é y2(n). A estimação final do sinal de voz sem ruído é $\hat{x}(n)$, que resulta de resta do sinal observado a estimação do ruído $\hat{y}1(n)$.

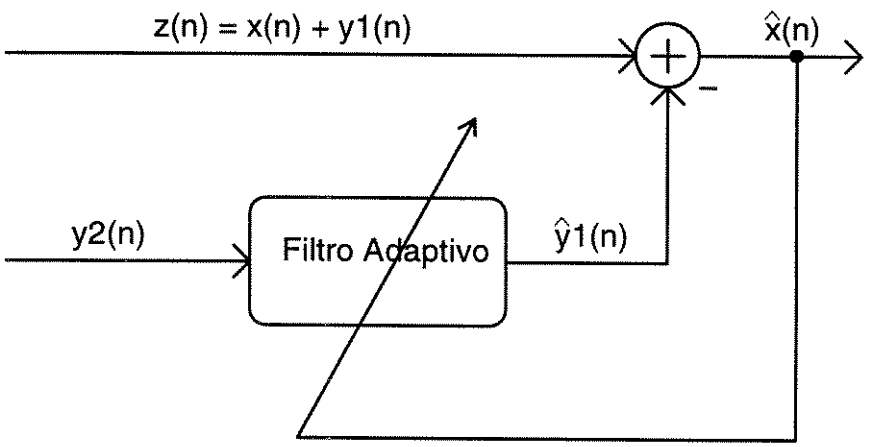


FIGURA 4.13 - Cancelação Adaptativa de ruído

Sistemas de reconhecimento para pilotos de combate de aeronaves, helicópteros e outros são particularmente dados para esta aplicação. Existe muito esforço para levar adiante esta linha de investigação com bons resultados.

• A idéia das **medidas de distorção robustas** é a de calcular distâncias enfatizando seletivamente as áreas do espectro de potência que são menos contaminadas pelo ruído. Um esquema de compensação de ruído pode ser interpretado implicitamente como a definição de uma medida de distorção robusta.

Os primeiros mecanismos de compensação do ruído procuram reforçar os picos espectrais, já que como possuem maior concentração de energia, tem uma melhor relação sinal - ruído que os vales espectrais. Estes mecanismos são particularmente úteis em conjunto com as modelações somente pólos, tipo AR.

Uma das medidas de comparação de espectros que melhor resultados produzem em reconhecimento de voz são os cepstrais, pelo qual a investigação para melhorar sua robustez frente ao ruído tem sido intensa. As medidas de distância cepstral pesada (chamada "liftered" em inglês) tem a seguinte forma:

$$d(Cr,Ct)=\sum_{n} w^2(n)(Cr(n)-Ct(n))^2 \tag{4.16}$$

onde Cr(n) e T(n) são os coeficientes cesptrais dos dois espectros a serem comparados e w(n) é a função de ponderação. Quando se utilizam funções de ponderação corretas tem-se encontrado melhorias no reconhecimento de voz tanto em condições limpas como ruidosas. Observando que o ruído branco afeta mais o espectro de somente pólos seria aconselhável aplicar um lifter cepstral para desenfatizar os termos de baixa "quefrencia" (escala logarítmica de freqüências).

A busca de uma medida de distorção robusta pode ser mais efetiva usando-se ferramentas matemáticas para um marco analítico dos efeitos do ruído nos parâmetros cepstrais. Mansor and Juang (1989) [1] reportaram tanto analiticamente como experimentalmente, que o ruído branco aditivo causa que a norma do vetor cepstral (longitude do vetor cepstral, excluído o termo de ordem zero) se reduz mas a orientação permanece aproximadamente constante.

Definimos que duas funções contínuas F e G de x são da *mesma orientação* se F e G são ambas monotonamente decrescentes, ou são ambas monotonamente crescentes, com respeito de x; e são de direção oposta se uma é monotonamente crescente e a outra é monotonamente decrescente.

Lema: Seja x uma função real de $w \in D$, e sejam F e G, duas funções reais de x. Seja também $\int_D F(x(w))dw = 0$, obtemos:

$$\int_D F(x(w))G(x(w))dw \begin{cases} \geq 0, & \text{se F y G são da mesma orientacão} \\ \leq 0, & \text{se F y G são de orientacão oposta} \end{cases} \tag{4.17}$$

Se F(x) é monótona e $\int_D F(x(w))dw = 0$, existe pelo menos um x_0 na região definida por D tal que $F(x_0) = 0$. Por outro lado:

$$\int_D F(x(w))G(x(w))dw = \int_D F(x(w))(G(x(w)) - G(x_0))dw \tag{4.18}$$

Como G(x) é também monótona, $G(x) - G(x_0)$ tem o mesmo signo de F(x) se ambas tem a mesma orientação, e tem signo oposto de F(x) se são de direção oposto.

O modelo do espectro de potência aditivo esta definido como $1/|A(e^{jw})|^2 + \gamma$, onde γ é não negativo e é uma função da relação sinal - ruído. A(z) tem todos os zeros dentro do círculo unitário e $1/A(z)A(z^{-1}) + \gamma$ é analítico no círculo unitário o que corresponde a uma seqüência de energia finita. Usando o teorema de Parseval, relacionamos a energia cepstral ao espectro logaritmico do sinal por:

$$G(\gamma) = 2\sum_{i=1}^{\infty} c_i^2 = \int_{-\pi}^{+\pi} \left[\ln F(w) - \int_{-\pi}^{+\pi} \ln F(w)\, dw/2\pi \right]^2 dw/2\pi \tag{4.19}$$

onde F(w) é definido como:

$$F(w) = 1/|A(e^{jw})|^2 + \gamma$$

Note-se que em ambos lados da equação (4.19) não foi incluído o termo zero cepstral já que este corresponde a ganância e normalmente é tratado em forma diferente do resto. Na equação (4.19) expressamos a energia no domínio cepstral como uma função de γ . Levando a derivada de $G(\gamma)$ respeito de γ, obtemos:

$$dG(\gamma)/d\gamma = 2\int_{-\pi}^{+\pi} \left[\ln F(w) - \int_{-\pi}^{+\pi} \ln F(w)\, dw/2\pi \right] \frac{1}{F(w)} dw/2\pi \tag{4.20}$$

Do lema, pode-se mostrar facilmente que a derivada de $G(\gamma)$ é negativa para γ positivos. Sem o termo 1/F(w) o resultado da integração é zero, então qualquer constante pode ser adicionada a 1/F(w) sem mudar o resultado da integração. Se escolhemos a constante para que seja:

$$\exp(-\int_{-\pi}^{+\pi} \ln F(w)\, dw/2\pi) \tag{4.21}$$

obtemos:

$$dG(\gamma)/d\gamma = 2\int_{-\pi}^{+\pi}\left[\ln F(w) - \int_{-\pi}^{+\pi}\ln F(w)\,dw/2\pi\right] \cdot\left[\frac{1}{F(w)} - \exp\left(\int_{-\pi}^{+\pi}\ln F(w)\,dw/2\pi\right)\right]dw/2\pi \le 0 \tag{4.22}$$

Já que os dois termos entre parêntese tem signos opostos para cada w. Logo a energia no domínio cesptral, excluído o termo zero de "quefrencia", é uma função decrescente com γ e o máximo facilmente demonstra-se que ocorre com γ=0, correspondendo com o modelo limpo. Então deduzimos facilmente que:

$$|Cr| \ge |Ct| \tag{4.23}$$

onde Cr denota os coeficientes cepstrais do modelo limpo (modelo de referência) e Ct denota os do modelo ruidoso (de teste). A igualdade anterior é válida também para vetores truncados de cepstrais que são mais utilizados no reconhecimento de voz, já que estes em geral convergem rapidamente a zero.

Também é interessante notar que do produto escalar entre Cr e Ct, usando este modelo, é uma quantidade positiva. Isto pode-se demonstrar-se expressando o produto escalar no domínio logaritmico do espectro:

$$2\sum_{i=1}^{\infty} c_{t,i}c_{r,i} = \int_{-\pi}^{+\pi}\ln F(w)\ln[1/|A(e^{-jw})|^2]dw/2\pi \ge 0 \tag{4.24}$$

logo de invocar o lema e a propriedade da meia nula do espectro logaritmico de um sistema de fase mínima somente pólos. Como o produto escalar entre dois vetores essencialmente define o co-seno do ângulo entre eles, a desigualdade anterior implica que este ângulo é sempre menor ou igual a $\pi/2$. A desigualdade é também válida para vetores truncados como estabelecimos anteriormente. ■

O cômputo de distâncias euclidianas é afetado por uma distorção no módulo dos vetores comparados. Este resultado sugere o uso da operação de projeção para formular medidas de distorção robustas, no caso onde os padrões de referência são limpos e os padrões a serem reconhecidos são obtidos em condições desconhecidas. Em Mansor and Juang (1989) a seguinte fórmula foi a que teve melhores resultados num conjunto de medidas de distorção propostas:

$$d(Cr,Ct) = |Ct|(1 - Cr^T Ct / (|Cr||Ct|)) \tag{4.25}$$

onde Cr e Ct são os cepstrais dos padrões de referência e teste, respectivamente e T denota transposição. Em uma experiência utilizando a técnica de DTW (apresentada em 4.3.2), do tipo dependente do falante e palavras isoladas, obtiveram-se resultados superiores a outras técnicas conhecidas.

No trabalho de Carlson B. *et al.* [37] realizou-se uma aplicação das medidas de distorção por projeção para o cálculo robusto de funções densidade de probabilidade no contexto do reconhecimento robusto de voz por HMM. A projeção é utilizada na correção do módulo do vetor a serem reconhecido no contexto do cálculo de funções densidade de probabilidade gaussianas. A seguir apresenta-se a fórmula utilizada no trabalho mencionado:

$$\log d(Cr,Ct) = (\mu t - \lambda \mu r)^T C^{-1} (\mu t - \lambda \mu r) + \log|C| + N \log 2\pi \qquad (4.26)$$
$$\text{onde: } \lambda = (\mu t^T C^{-1} \mu r) / (\mu r^T C^{-1} \mu r)$$

onde μr e μt são os vetores de medias dos vetores de referência e teste Cr e Ct, respectivamente, C é a matriz de covariança do vetor μr e N é a ordem do vetor de parâmetros. Neste trabalho somente utiliza-se uma equalização de primeira ordem, já que somente se corrige o vetor de medias dos cepstrais de teste. Uma vez calculada robustamente a função densidade de probabilidade, o resto do HMM é semelhante ao padronizado. O experimento realizado é o de reconhecimento ruidoso dependente do falante com um conjunto de 34 palavras. Partindo de um reconhecimento do 98.8% com baixo ruído, se obteve um 80.6% a 5dB SNR, com a utilização de coeficientes Mel-Cepstrum.

• O sistema auditivo humano percebe a voz melhor que qualquer máquina quando uma distorção ruidosa está presente. Baseado nesta premissa, Ghitza (1986) [36], propôs um **modelo biológico computacional**, chamado *Ensemble Interval Histogram* (EIH), para representar o padrão de disparo dos nervos auditivos, que pode ser robusto à corrupção do ruído. O modelo EIH tem (1) um conjunto de 85 filtros cocleares simulados que separa a sinal de voz na banda 200 - 3200 Hz, (2) medidas de cruzamento de níveis para cada uma das saídas dos filtros cocleares e (3) a acumulação dos histogramas dos intervalos de cruzamento de níveis. O histograma de "ensemble" resultante é reminiscente de um espectro, mas com as não linearidades e resolução em freqüência não uniforme próprias do sistema auditivo humano. Os resultados de aplicação do EIH, conjuntamente com um esquema de alinhamento dinâmico indicou melhoras sensíveis no reconhecimento de voz ruidosa masculina.

Em Sandhu S. e Ghitza O. (1995) [38], realiza-se uma comparação entre EIH e os coeficientes Mel-Cepstral para diversas condições adversas, para um problema de classificação de fonemas da base de dados TIMIT. Estas condições adversas baixo estudo se dividem em voz limpa, canal telefônico e distorção por reverberação. Com o uso de características estáticas EIH melhora sensivelmente o reconhecimento de canal telefônico (de 10.1% a 20.8%) enquanto que é superado nas outras condições por poucas décimas. Com o uso de características estáticas e dinâmicas nos cepstrais, o EIH é superado em todos os casos pelos coeficientes Mel-Cepstral.

CAPÍTULO 5

PROPOSTA DE UMA REDE NEURAL DE RECONHECIMENTO ADAPTATIVO.

5.1- PROPOSIÇÃO DO PROBLEMA.

Apresentaremos neste capítulo uma ***proposta de reconhecedor ruidoso baseado em redes neurais.*** Propõe-se, focalizar o problema do reconhecimento de padrões ruidosos quando estejam disponíveis para o treinamento os correspondentes padrões limpos. Isto impõe uma restrição na generalidade do problema, existem muitas aplicações incluídas neste caso. Particularmente as aplicações onde o registro dos padrões para o treinamento seja feito em condições controladas, isto é em condições tais que estes não estejam afetados por severas distorções. Porém na prática, na etapa de reconhecimento, não podem-se garantir estas condições controladas e os padrões podem estar afetados por sinais aleatórios. A falta de robustez frente esta situação provoca uma importante perda de desempenho de reconhecimento.

5.2- CLASSIFICADORES ROBUSTOS

A proposta que se apresentará a seguir está incluída dentro dos **classificadores robustos**. Isto é, propõe mecanismos para que o classificador seja robusto à distorção ruidosa dos padrões. Isto em contraposição com a estratégia clássica de filtrar o ruído na etapa das caraterísticas (chamado caraterísticas invariantes) e evitar que passem ao classificador. Exemplo desta última foi apresentada na seção 4.5.3, como EIH. Esta estratégia clássica tem sua razão de ser, é mais fácil aproveitar as ferramentas e não modificar todo o sistema de reconhecimento pela presença de ruído. Mas o grande problema desta aproximação é que o **classificador não é robusto ao ruído**, portanto em níveis de distorção onde as caraterísticas não possam filtrar todo o ruído, existirá um erro no classificador.

5.3- PROPOSTA DE UM NEURÔNIO DE RECONHECIMENTO ADAPTATIVO

O modelo clássico do neurônio (chamado neurônio clássico), detalhado no capítulo 2, carateriza-se por uma função não linear crescente, saturada e diferenciável sobre uma somatória do resultado de uma transformação linear efetuada sobre o padrão de entrada. O nome clássico da transformação linear é "pesos sinápticos" ou simplesmente pesos, definidos como $W \in \Re^{Nx1}$. A função não linear é, em geral, do tipo "sigmóide" caracterizada por uma equação $1/(1+ \exp(-W^t \cdot X))$. Ver figura 5.1. Este neurônio tem a propriedade de separar em dois, por um hiperplano, o espaço de estados [2] e se enquadra

dentro das redes estáticas [5], devido a que carece de memória já que sua saída somente depende da entrada atual.

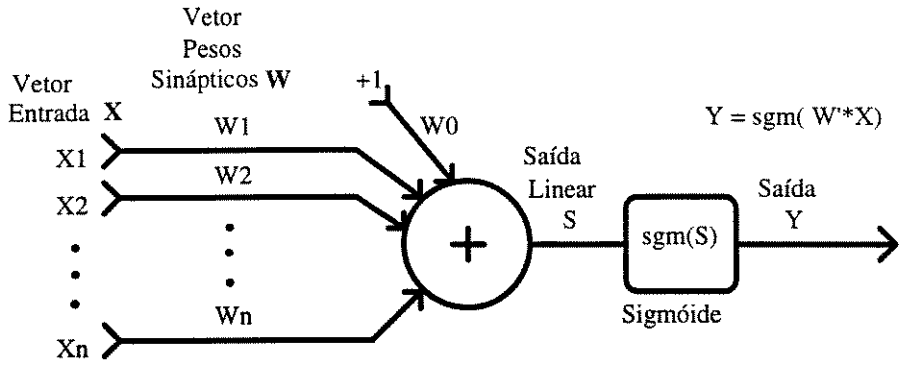


FIGURA 5.1: Neurônio Clássico

A partir deste modelo, se propõe um novo modelo de neurônio, chamado ***neurônio de reconhecimento adaptativo***, com o objetivo de superar o desempenho do neurônio clássico em ambientes ruidosos. Especificamente, seu nome deve-se a que incorpora um estimador, na etapa de reconhecimento, de caraterísticas adaptativas (filtro de Kalman).

À diferença do neurônio clássico, este novo modelo de neurônio caracteriza-se por uma função não linear crescente, saturada e diferenciável sobre uma soma do resultado de ***duas transformações lineares*** efetuadas sobre o padrão de entrada. Estas podem-se separar numa fixa e outra adaptativa.

A **transformação linear fixa** é efetuada pelos pesos sinápticos tradicionais de uma rede multi camada em avanço (feedforward multilayered). Estes são calculados por um processo de aprendizagem do tipo backpropagation.

A **transformação linear adaptativa**, destinada a estimar o padrão original a partir de sua observação ruidosa, é implementada por um filtro de Kalman definido da seguinte forma: identifica-se ao Vetor de Estado X(t) com o padrão original, sem ruído. A equação de processo atribui ao novo estado a transformação linear do estado anterior, através da Matriz de Transição de Estados $\Theta_{t,t-1}$, somada ao Ruído de Estados v(t). Neste caso, a *Matriz de Transição de Estados é a matriz identidade* e o *Ruído de Estados é nulo.* Identifica-se ao Vetor de Observações $\tilde{X}(t)$ com a observação ruidosa do padrão original. A equação de medida atribui ao vetor de observações a transformação linear do vetor de estados, pela Matriz de Medição C(t), somado ao erro de medição o ruído de medição v(t). Neste caso, a *Matriz de Medição é a matriz identidade* e o *ruído de medição é o ruído que afeta ao padrão original.*

Equação de Processo: $$X(t) = X(t-1) \tag{5.1}$$

Equação de Medição: $$\tilde{X}(t) = X(t) + v(t) \tag{5.2}$$

Portanto, a equação de atualização da estimação do estado é :

$$\hat{X}(t) = \hat{X}(t-1) + K_t * (\tilde{X}(t) - \hat{X}(t)) \tag{5.3}$$

onde $\hat{X}(t)$ é a nova estimação do padrão original, $\hat{X}(t)$ é a estimação anterior, K_t é a matriz de ganho de Kalman, e $\tilde{X}(t)$ é a observação ruidosa do padrão. Nesta equação (5.3) pode observar-se melhor acerca da transformação linear adaptativa, com memória calculada pelo filtro de Kalman.

Então, a transformação total realizada pelo neurônio proposto é:

$$Y(t) = sgm(W^T(\hat{X}(t-1) + K_t * (\tilde{X}(t) - \hat{X}(t-1)))) \tag{5.4}$$

onde $Y(t)$ é a saída do neurônio e $sgm(\bullet)$ é a função sigmóide descrita anteriormente. O que é o mesmo:

$$\begin{aligned} \hat{X}(t) &= \hat{X}(t-1) + K_t * (\tilde{X}(t) - \hat{X}(t-1))) \\ Y(t) &= sgm(W^T\hat{X}(t)) \end{aligned} \tag{5.5}$$

Um diagrama deste modelo proposto apresenta-se na figura 5.2.

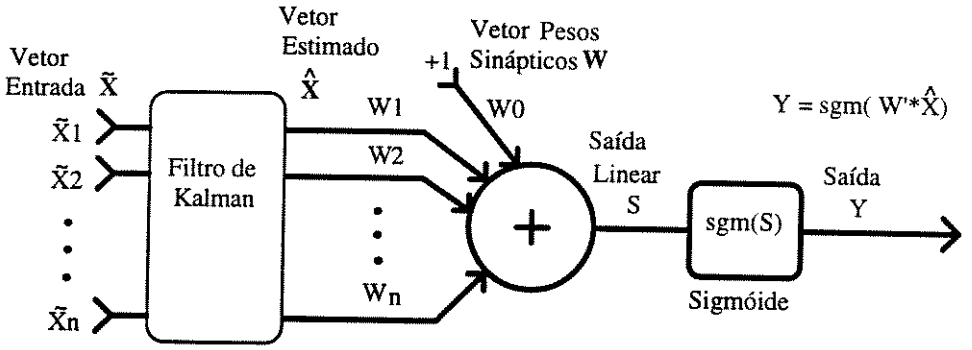


FIGURA 5.2: Neurônio de Reconhecimento Adaptativo

5.3.1– PROPRIEDADES DO NEURÔNIO PROPOSTO:

- este classifica a estimação, e não a observação, do padrão original. Portanto, *poderia classificar um padrão como pertencente a uma classe mesmo que todas suas observações caiam fora desta.* Obteríamos isto se a estimação deste padrão original viesse a cair na classe;

- o neurônio herda as propriedades do filtro de Kalman, que é um estimador de estado ótimo para sistemas lineares em presença de ruído branco. Isto será desenvolvido em profundidade na próxima seção;

- modifica a caraterística do neurônio clássico, de estática para dinâmica, no sentido que a saída, não somente depende da entrada atual mas também, de entradas passadas. Neste sentido incorpora uma "memória" ao neurônio;

- este novo neurônio permite a aprendizagem e o reconhecimento em ambientes ruidosos;

- precisa que estejam disponíveis várias amostras ruidosas de um mesmo padrão para poder realizar o processo de estimação;

- tem um maior custo computacional de cálculo que o neurônio clássico.

A idéia desta proposta é a de aproveitar a informação redundante que dispõe-se em quase todas a aplicações de reconhecimento de padrões. Esta informação redundante, explica-se a partir do teorema de amostragem: apesar de que a frequência mínima para recuperar um sinal, a partir do mesmo amostrado, seja o duplo da máxima frequência deste, em geral utiliza-se uma frequência razoávelmente maior. Isto implica que, na escala dada pelo período de amostragem, a duração temporal do padrão é amplificada. Logo, dispõe-se de um conjunto de amostras que representam ao mesmo padrão. Esta informação redundante é utilizada pelo neurônio adaptativo, num processo de estimação do padrão original, a partir de suas observações ruidosas, implementado por um filtro de Kalman.

A **posição** proposta para o **filtro de Kalman** não é a única possível. Seria interessante estudar outras posições possíveis. Devido a que a transformação realizada pelos pesos sinápticos, antes do produto escalar, é uma transformação linear, é indistinto que o filtro de Kalman esteja antes ou depois dos mesmos. Outra possível posição seria **a posteriori do produto escalar** e antes da função sigmóide. A vantagem desta seria uma sensível diminuição do custo computacional no cálculo do filtro de Kalman, ao ser aplicado a um sistema escalar como seria a saída do produto escalar somado ao ruído. No caso que o vetor ruído branco original fosse uma emissão multidimensional que muda a cada instante de tempo, o ruído resultante logo do produto escalar continuaria sendo branco. Na próxima figura 5.3 se apresenta ao modelo proposto com o filtro de Kalman a posteriori do produto escalar.

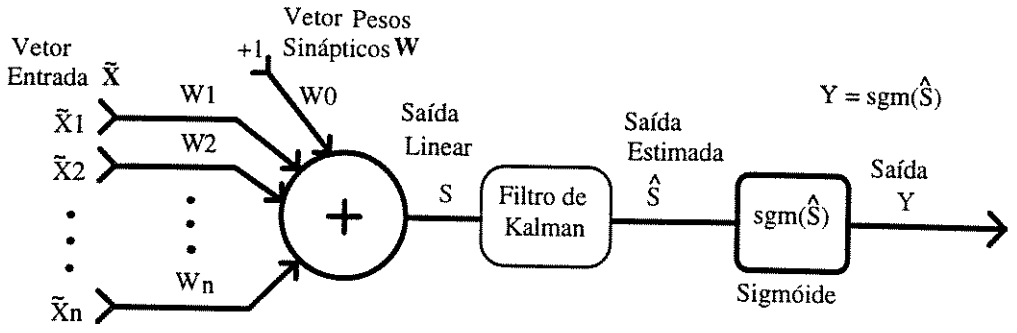


FIGURA 5.3: Neurônio de Reconhecimento Adaptativo, com o filtro de Kalman a posteriori do produto escalar

Estudaremos a estatística, *para padrões estáticos*, da variável aleatória resultante do produto escalar do vetor de pesos multiplicado pelo padrão estático adicionado de ruído branco de média nulo. Provaremos que a média desta variável aleatória resulta o produto escalar do vetor original e sua variância resulta a variância do ruído branco afetado de uma constante positiva. Se definimos $(X+N)\in \Re^{Nx1}$ como o vetor estático observado $X\in \Re^{Nx1}$ tal que $X = [X1\ X2\ ...\ XN]^T$, $E[X] = X$ e $E[XX^T] = \underline{0}$ (padrão estático) somado ao vetor ruído branco $N\in \Re^{Nx1}$,tal que $N = [N1(t)\ N2(t)\ ...\ NN(t)]^T$ de média nula (com $E[N] = 0$ e $E(NN^T) = \sigma^2 I$) com $N\in N$ e estando todos os componentes de ambos não correlacionados para todo instante de tempo (isto é $E[Xi\ Nj(t)] = 0\ \forall(i, j, t)$, o qual implica $E[XN^T] = \underline{0}$ e que $E[NX^T] = \underline{0}$). Se a saída linear do neurônio através do produto escalar é $s = W^T(X+N)\in \Re$, obtemos:

$$E[s] = E[W^T(X+N)] = W^TE[X+N] = W^T(E[X]+E[N]) = W^TE[X] = W^TX. \quad (5.6)$$

No desenvolvimento anterior utilizamos a propriedade de linearidade do operador valor esperado e o fato que o valor esperado do ruído branco ser nulo.

Utilizando a propriedade de linearidade do operador valor esperado e do fato que o padrão e o ruído branco estão fora de correlação, deduzimos.

$$E[s^2] = E[(W^T(X+N))\ (W^T(X+N))^T] =$$

$$= E[W^TXX^TW + W^TXN^TW + W^TNX^TW + W^TNN^TW] =$$

$$= W^TE[XX^T]W + W^TE[XN^T]W + W^TE[NX^T]W + W^TE[N^TN]W =$$

$$= 0 + 0 + 0 + W^T\sigma^2IW =$$

$$= \sigma^2W^TIW\ ;\ (W^TIW > 0) \quad (5.7)$$

Comparando os resultados das equações (5.6) e (5.7) com os valores esperados e variância do padrão estático original adicionado de ruído branco com as mesmas especificações, $E[X+N] = X$; $E[(X+N)(X+N)^T] = \sigma^2 I$, conclui-se que estamos trabalhando com duas variáveis aleatórias com as mesmas estatísticas de primeira e segunda ordem, salvando as diferenças das dimensões.

Apesar de que a demonstração tenha sido feita para padrões estáticos, é válida também como aproximação de padrões com uma variação muito lenta no tempo a respeito da escala de amostragem.

Das demonstrações anteriores conclui-se que seria vantajoso trabalhar com este último modelo por seu reduzido custo computacional e sua equivalência estatística de primeira e segunda ordem. Mesmo assim o custo que se deve pagar por utilizar um escalar em vez do vetor original contaminado por ruído branco, é que quando o ruído não seja branco a variância da estimação escalar será uma função complicada de toda a matriz de correlação do ruído, como pode ver-se na penúltima fila da equação (5.7): $E[s^2] = f(...;W^T E[N^T N]W)$. É importante destacar que o ruído branco é uma idealização e que nos casos reais o ruído não será branco. A estimação vetorial tem mais graus de liberdade e ainda o caso de ruído diferente do branco é um caso tratável, conhecendo a estatística do ruído.

5.3.2- FUNDAMENTAÇÃO TEÓRICA DE SUA OTIMALIDADE.

Sabe-se que o filtro de Kalman é um estimador de estado ótimo em presença de ruído branco. O filtro estará atuando em condições ótimas, se for válida a relação linear entre o padrão e a perturbação, quer dizer, se a perturbação é do tipo aditivo. Então suposto válido o modelo linear de observação ruidosa do padrão e a perturbação do tipo ruído branco, ao estar atuando o filtro em forma ótima, o neurônio está dispondo da máxima informação possível sobre o padrão. Então suposto o classificador, como classificador ótimo, (difícil de verificar), estaria-se minimizando o erro de classificação da observação ruidosa do padrão.

No capítulo 3 provou-se que se partimos de um valor inicial da estimação de estado como a média do valor inicial da verdadeira variável, o filtro de Kalman é não polarizado. Utilizando isto como premissa provaremos que a saída do neurônio adaptativo aproxima-se à saída de um neurônio cuja entrada fosse o valor médio da variável aleatória.

TEOREMA 5.1: (*Da igualdade entre os valores esperados das saídas do modelo de neurônio proposto e o estandarizado para ruído branco, em primeira ordem*)

Hipótese: Seja $X(t)\in\Re^{Nx1}$; $N\in\mathbf{N}$, um vetor de entrada no tempo $t\geq 0$, a serem classificado, seja $W\in\Re^{Nx1}$ um vetor de pesos fixo, seja $N(t)\in\Re^{Nx1}$ um vetor gerado por uma realização de um processo aleatório de tipo ruído branco, de valor médio nulo e variância $\sigma^2 I$. Dado um modelo de neurônio cuja saída $y\in\Re$, dados o vetor de pesos W e o vetor de entrada X(t)+N(t) é $y(t) = sgm(W^T*(X(t)+N(t)))$, onde T denota transposição e sgm(•) denota a função sigmóide definida pela seguinte equação: $sgm(p) = 1/(1+\exp(-p))$. Seja $y^{RA}(t)$ a saída do modelo do neurônio de reconhecimento adaptativo definido por $y^{RA}(t)=sgm(W^T\hat{X}(t))$, considerando que para este modelo o vetor de entrada ao neurônio é o vetor $\hat{X}(t)$ estimado pelo filtro de Kalman, sobre a entrada atual X(t)+N(t). O valor em tempo 0 da estimação do filtro de Kalman do modelo proposto é igual ao valor esperado em tempo 0 da variável no ruidosa $\hat{X}(0)=E[X(0)]$. Seja $p\in\Re$ uma constante real, pequena.

Tese: Os valores esperados das saídas do modelo de neurônio de reconhecimento adaptativo $E[y^{RA}(t)]$ com entrada ruidosa e do modelo de neurônio clássico com entrada sem ruído E[y(t)], coincidem. Esta igualdade é válida para todo tempo t, numa *aproximação de primeira ordem*, isto é: $E[y^{RA}(t)] = E[y(t)]; \forall\ t\geq 0$.

Demonstração: Desenvolveremos a continuação a função não linear sgm(•), em série de Mac-Laurin em p: $sgm(p) = sgm(0) + sgm^1(0)\ p^1 / 1! + ... + sgm^n(0)\ p^n / n!$ onde o supra índice sobre a função sigmóide denota "derivada de ordem n", e o supra índice sobre p denota, "elevado a n-ésima potência". A equação anterior, sendo uma função par, pode-se escrever da seguinte forma mais compacta:

$$sgm(p)=\sum_{i=1}^{n} sgm^{(2i-1)}(0)\ (p)^{(2i-1)} / (2i-1)! \qquad (5.8)$$

Se substituimos p pela entrada atual do neurônio sem ruído $W^TX(t)$ obtemos:

$$y(t)=sgm(W^TX(t))=\sum_{i=1}^{n} sgm^{(2i-1)}(0)\ (W^TX(t))^{(2i-1)} / (2i-1)! \qquad (5.9)$$

Tomando da expressão anterior a aproximação de primeira ordem,

$$y(t)=sgm(W^TX(t))\cong sgm(0)\ W^TX(t) \qquad (5.10)$$

Tomando o valor esperado da saída do neurônio estandarizado:

$$E[y(t)]\cong sgm(0)W^TE[X(t)] \qquad (5.11)$$

Vamos realizar o mesmo desenvolvimento para a saída do neurônio proposto com entrada ruidosa $y^{RA}(t)$, considerando que o vetor de entrada ao neurônio é o estimado pelo filtro de Kalman $\hat{X}(t)$, tomando a aproximação de primeira ordem:

$$y^{RA}(t) \cong sgm(0)W^T\hat{X}(t) \tag{5.12}$$

Levando em conta a hipótese que o valor de $\hat{X}(0)=E[X(0)]$, se nos remitimos à equação 3.79, do capítulo terceiro, onde se demonstra que com esta hipótese cumpre-se que $E[\hat{X}(t)]=E[X(t)]$, $\forall\ t\geq 0$, obtemos

$$\begin{aligned} E[y^{RA}(t)] &\cong sgm(0)W^T E[\hat{X}(t)] \\ &\cong sgm(0)W^T E[X(t)] \end{aligned} \tag{5.13}$$

Finalmente comparando as equações (5.8) e (5.10) obtemos:

$$E[y^{RA}(t)] = E[y(t)];\ \ \forall t \text{ (Em primera ordem)} \tag{5.14}$$■

O teorema anterior mostra um resultado importante, no sentido que a estimação da saída do modelo de reconhecimento adaptativo proposto é não polarizada, para o caso de um vetor de entrada constante no tempo. É importante destacar que o resultado anterior poderia ter-se alcançado com um modelo do neurônio estandarizado com um vetor de entrada como o valor médio do vetor ruidoso. Mas para justificar o uso do filtro de Kalman devemos mostrar um resultado mais forte que o anterior. Neste sentido trataremos de derivar uma relação entre a propriedade de mínima variância da estimação do filtro de Kalman e a variância das saídas.

TEOREMA 5.2: (*Da propriedade de mínima variância da saída estimada do modelo de reconhecimento adaptativo, em primeira ordem*)

Hipótese: A mesma do teorema 5.1

Tese: A variância entre as saídas dos neurônios clássico e de reconhecimento adaptativo y(t) e $y^{RA}(t)$, é mínima em primeira ordem.

Demonstração: De acordo as definições de $y^{RA}(t)$ e y(t), e usando aproximações de primeira ordem, obtemos:

$$\begin{aligned} y(t) - y^{RA}(t) &\cong sgm(0)(W^T X(t) - W^T\hat{X}(t)) \\ &\cong sgm(0)(W^T(X(t) - \hat{X}(t))) \\ &\cong sgm(0)W^T\tilde{X}(t) \\ &\cong sgm(0)\tilde{X}^T(t)W \end{aligned} \tag{5.15}$$

Se elevamos a equação ao quadrado em ambos termos

$$\begin{aligned} (y(t) - y^{RA}(t))^2 &\cong sgm(0)(\tilde{X}^T(t)W)sgm(0)(W^T\tilde{X}(t)) \\ &= sgm(0)^2\tilde{X}^T(t)WW^T\tilde{X}(t) \end{aligned} \tag{5.16}$$

aplicamos o operador valor esperado E[•], obtemos:

$$\begin{aligned} E[(y(t)-y^{RA}(t))^2] &\cong E[sgm(0)^2(\tilde{X}^T(t)WW^T\tilde{X}(t))] \\ &\cong sgm(0)^2 E[(\tilde{X}^T(t)WW^T\tilde{X}(t))] \end{aligned} \tag{5.17}$$

Onde para passar da segunda linha utilizamos a propriedade de linearidade do operador valor esperado, sendo $sgm(0)^2$ uma constante, o valor esperado de uma variável aleatória multiplicada por uma constante é igual a constante multiplicada pelo valor esperado da variável aleatória.

Analisemos a expressão dentro do valor esperado à direita da igualdade da equação anterior:

$$(\tilde{X}^T(t)W)(W^T\tilde{X}(t)) \geq 0 \Rightarrow (\tilde{X}^T(t)WW^T\tilde{X}(t)) \geq 0 \tag{5.18}$$

Por ser o quadrado de um número qualquer a expressão entre parêntese é maior que zero. Se reagrupamos os termos temos uma forma quadrática, onde a matriz central é WW^T; por ser um produto de um mesmo vetor, esta matriz é simétrica. Além disso, como a forma quadrática é maior ou igual a zero, a matriz WW^T é chamada semi definida positiva.

Como o filtro de Kalman está trabalhando numa condição ótima, sua estimação de estado é de mínima variância

$$\text{Filtro de Kalman (ótimo)} \Rightarrow \min\{E[\tilde{X}^T(t)\tilde{X}(t)]\} \tag{5.19}$$

Demonstra-se que o mínimo de $E[(\tilde{X}^T(t)WW^T\tilde{X}(t))]$ é independente da matriz WW^T. Por tanto, pode-se afirmar que:

$$\min\{E[\tilde{X}^T(t)WW^T\tilde{X}(t)]\} \Leftrightarrow \min\{E[\tilde{X}^T(t)\tilde{X}(t)]\} \tag{5.20}$$

Então de acordo a equação anterior e a (5.14) podemos finalmente afirmar que:

$$\min\{E[\tilde{X}^T(t)WW^T\tilde{X}(t)]\} \Rightarrow \min\{E[(y(t)-y^{RA}(t))^2]\} \tag{5.21}$$

Da equação (5.21), concluimos que a propriedade de mínima variância do filtro de Kalman, implica a mínima variância entre as saídas dos neurônios estandarizados e de reconhecimento adaptativo. ■

De acordo com os teoremas 5.1 e 5.2, temos que, em primeira ordem, o modelo do neurônio proposto, comparando sua saída com o neurônio clássico para um padrão de entrada somado de ruído branco, é não polarizado e de mínima variância.

Os teoremas foram demostrados com séries de Mac Laurin, próximo da origem. Mas como este não condiciona o resultado, estes teoremas também são válidos para qualquer outro ponto real, sempre que a perturbação seja pequena próximo desse ponto.

5.4 - REDES NEURAIS E RUÍDO PRESENTE SOMENTE NO RECONHECIMENTO

Estudaremos no contexto das redes neurais a problemática do reconhecimento ruidoso. Claramente, dos três casos apresentados em 4.2.2, o terceiro (onde se dispõem de amostras de treinamento limpas e o reconhecimento ruidoso) é o mais fácil de tratar, já que é possível definir zonas de classificação no espaço de estados em concordância com as diferentes classes. O que restaria resolver na etapa de reconhecimento seria um problema de representação robusta dos padrões nessas zonas de classificação. Isto pode ser melhor explicado estudando o efeito que o ruído causa nos padrões a serem reconhecidos. O ruído transforma um padrão em outro distinto e esta transformação é aleatória. Na **observação ruidosa de um padrão pode se evidenciar uma perda das caraterísticas próprias da classe a qual este pertence**. O reconhecimento pela rede neural clássica radica em atribuir a observação à classe a qual esta pertence. Por tanto, classifica diretamente a observação ruidosa. Para que não exista um erro de reconhecimento, a observação ruidosa deve pertencer a mesma classe do padrão original e isto depende de dois fatores:

• Da distancia do padrão original da fronteira de sua classe no espaço de estados, definida pelo classificador. Obviamente quanto mais longe encontra-se da fronteira, menor a probabilidade de que uma observação ruidosa deste padrão caia fora da classe.

• Da potência (variância) do ruído. Quanto mais alta seja, maior a probabilidade de que a observação ruidosa caia fora da classe a qual pertence o padrão original.

5.5.- APLICAÇÃO: CLASSIFICAÇÃO RUIDOSA COM NEURÔNIO.

Se treinou um neurônio, num ambiente sem ruído, para a classificação de um padrão bidimensional de forma tal a separar o espaço de estados (o plano) com uma reta que passa pela origem com inclinação -1.

Para o reconhecimento, se escolheu um padrão qualquer pertencente a uma das classes, próximo à fronteira desta. Esta eleição arbitrária não influencia as conclusões da experiência a serem realizada, já que estas podem ser observadas em qualquer padrão, somente tomando em conta sua posição relativa à fronteira da classe. Foram feitas diversas experiências de reconhecimento para diferentes relações sinal - ruído. Em cada uma delas, se consideraram 30 observações ruidosas (ruído branco gaussiano) do padrão original.

Apresentamos a seguir dois gráficos, primeiro para o neurônio clássico e logo para o neurônio adaptativo. O padrão limpo é denotado com uma "**O**" e na posição de cada observação, denota-se com uma "**+**" se esta foi classificada pela rede com saída positiva e

com uma "*" se foi classificada com saída negativa. Pode-se observar que a saída desejada (correta) correspondente ao padrão limpo é uma "+" por pertencer à classe superior.

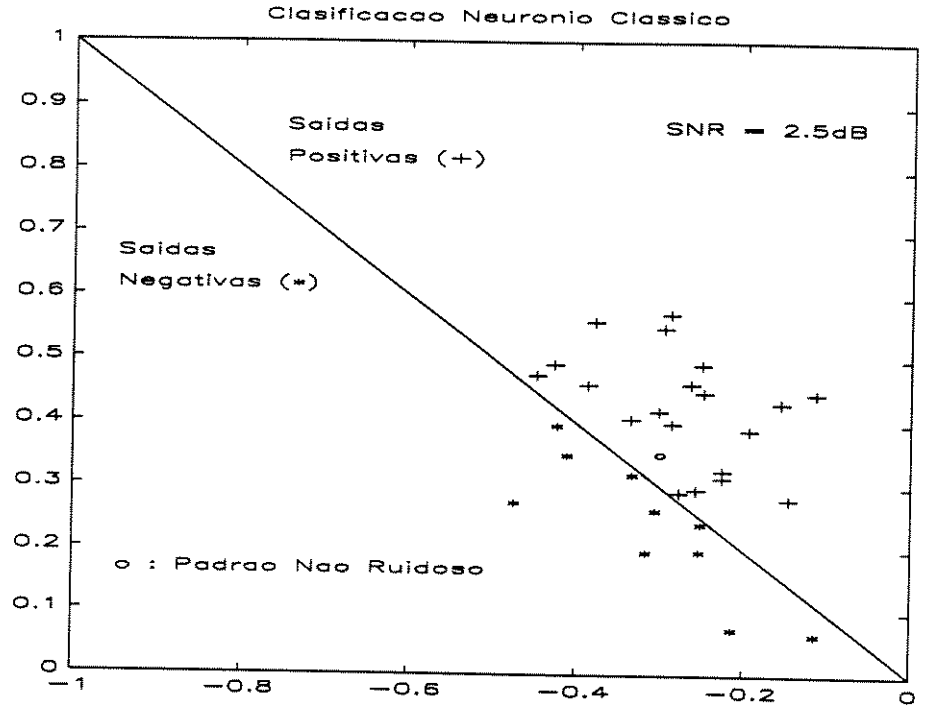


FIGURA 5.4: Classificação bidimensional com neurônio clássico.

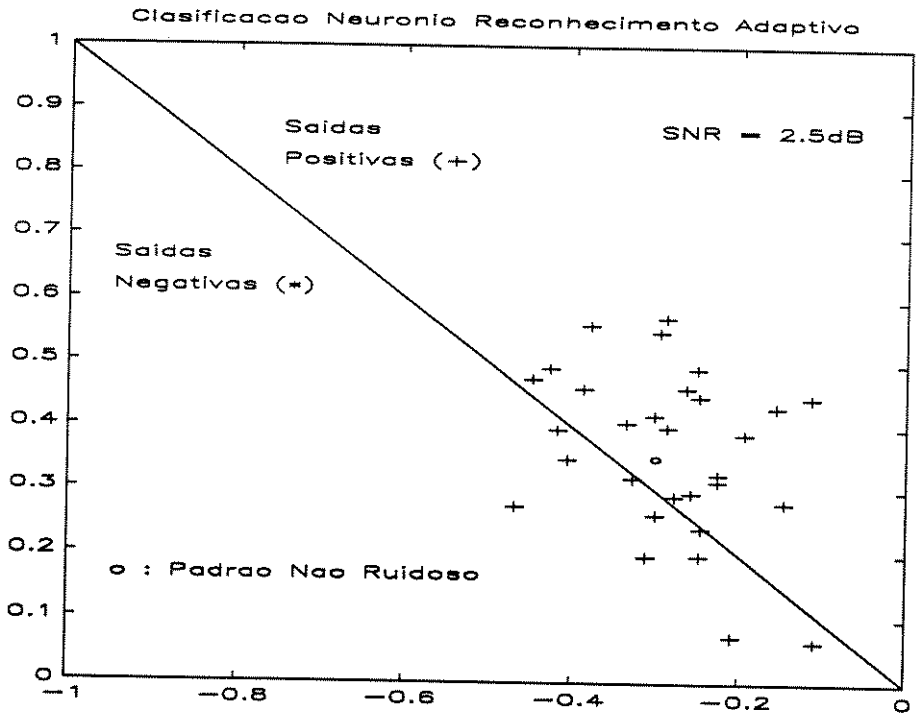


FIGURA 5.5: Classificação bidimensional com neurônio adaptativo.

Observamos um resultado surpreendente que é caraterístico do modelo do neurônio proposto: Se bem na figura 5.4, observa-se que a fronteira das classes divide as

saídas correspondentes aos padrões que caem num ou outro lado, na figura 5.5 observa-se que todas as observações, ainda as que caem na classe oposta a do padrão limpo, são classificadas com uma saída positiva. Isto é assim porque, *de acordo com a ordem e a posição das observações, a estimação que se vai construindo do padrão original cai sempre (neste caso) na classe correta.*

Apresentamos a seguir, os resultados da primeira simulação de reconhecimento de padrões ruidosos com o neurônio adaptativo.

SNR / **Neurônio**	**15 dB**	**10 dB**	**7.5 dB**	**5 dB**	**2.5 dB**	**0 dB**
Clássico	**65.8 %**	**56.9 %**	**53.4 %**	**52.2 %**	**50.7 %**	**48.2 %**
Reconhece. Adaptativo	**88.8 %**	**88.6 %**	**85.3 %**	**77 %**	**77 %**	**69.6 %**

TABELA 5.1 : Resultados Reconhecimento Neurônios Clássico / Adaptativo (média 10 repetições)

Na tabela 5.1 apresentam-se os resultados de reconhecimento em ambientes ruidosos, com distinta relação sinal - ruído, dos neurônios clássico e adaptativo. É evidente que a porcentagem de reconhecimento do neurônio clássico, somente depende da razão entre a quantidade de realizações que caíram na classe correta e as que caíram na classe oposta. Em quanto ao neurônio adaptativo, a porcentagem de reconhecimento mantém-se superior à do neurônio clássico, devido a que se classifica a estimação do padrão ruidoso e não sua observação.

5.6 - REDE NEURAL DE RECONHECIMENTO ADAPTATIVO:

A proposta do ***neurônio de reconhecimento adaptativo*** se estende naturalmente a uma ***rede neural de reconhecimento adaptativo***, do tipo multi camada em avanço (multilayered feedforward). Onde todos os neurônios da rede são neurônios de reconhecimento adaptativo já que o filtro de Kalman forma parte da estrutura computacional do neurônio. Neste caso não somente se realiza uma estimação do padrão original atrás do ruído, pelos neurônios da primeira camada, mas também realiza-se uma estimação sobre as saídas dos neurônios intermediários. Isto será desenvolvido em maior profundidade na próxima seção. Portanto as saídas da rede são produto de sucessivas estimações e classificações. Isto potencia fortemente a capacidade de classificação de padrões ruidosos da rede neural proposta, como se verá nas simulações realizadas. Na

próxima figura 5.6 apresenta-se uma rede neural de reconhecimento adaptativo de duas camadas, com M+1 neurônios na camada intermediária, L+1 neurônios na camada de saída e com um vetor de entrada de ordem n.

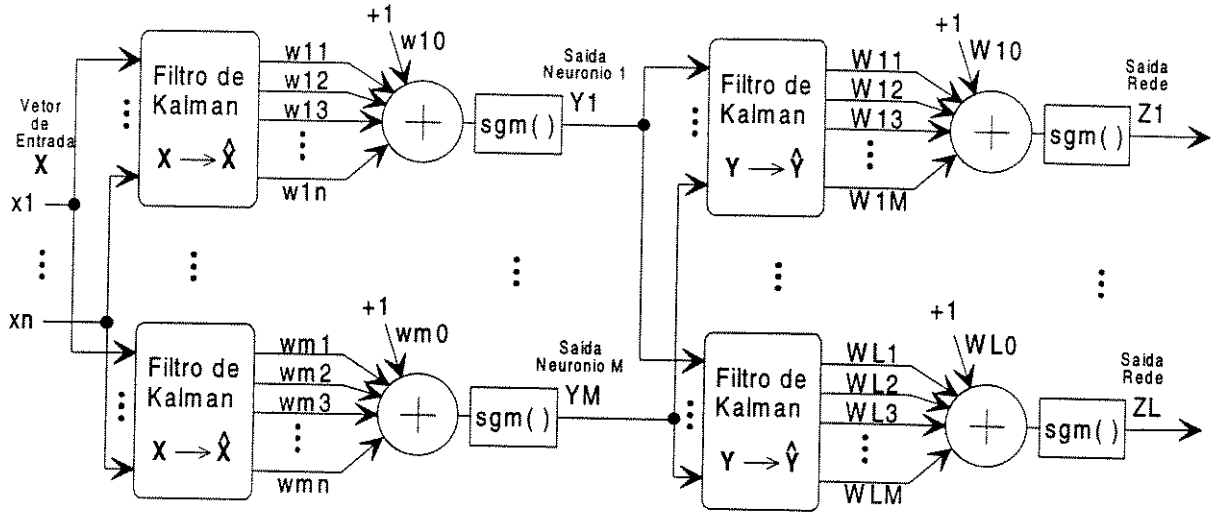


FIGURA 5.6 - Esquema geral de rede neural.

Segundo o diagrama as equações da rede neural de reconhecimento adaptativo são, para o caso de saída única:

$$\begin{aligned}
&\hat{X}(t) = \hat{X}(t-1) + K1_t * (X(t) - \hat{X}(t-1)) \\
&Yi(t) = sgm([wi0\ wi1\ \ win]\hat{X}(t)) \quad ; i = 1:M \\
&\hat{Y}(t) = \hat{Y}(t-1) + K2_t * (Y(t) - \hat{Y}(t-1)) \\
&Zj(t) = sgm([Wj0\ Wj1\ \ WjM]\hat{Y}(t)) \quad ; j = 1:L
\end{aligned} \tag{5.22}$$

onde $K1_t$ e $K2_t$ são as matrizes de Kalman dos filtros de Kalman da primeira e segunda camada no tempo t respectivamente. O vetor de saída da primeira camada Y(t) é $Y(t) = [1\ Y1(t)\ Y2(t)\ ...\ YM(t)]^T$, e o vetor de saída da rede neural Z(t) é $Z(t) = [Z1(t)\ Z2(t)\ ...\ ZL(t)]^T$, onde ^{T} denota transposição.

5.7.- PROPRIEDADES DA REDE NEURAL DE RECONHECIMENTO ADAPTATIVO.

As propriedades da rede neural de reconhecimento adaptativo são herdadas do neurônio de reconhecimento adaptativo. Uma propriedade importante, já anteriormente destacada, é que os neurônios de reconhecimento adaptativo estão em camadas o que implica que se potencia o processo de estimação. A importância desta estrutura deriva-se da comparação com uma estrutura onde os filtros de Kalman fossem somente um pré-procesamento, quer dizer somente estiveram na primeira camada. Neste último caso a rede classificaria o padrão estimado pelos filtros de Kalman. Se a perturbação é o

suficientemente forte poderia ocorrer que a estimação não fosse tão boa e o padrão seja semelhante a outro, durante um breve período. Neste caso a rede neural não poderia corrigir o erro já que classifica a observação diretamente. Por outro lado com a estrutura proposta, ao realizar-se uma estimação sobre as saídas dos neurônios intermediários, quando ocorra este erro durante um breve período de tempo, este poderia serem corrigido já que poderia serem interpretado pelos filtros de Kalman intermediários como uma perturbação e não modificar sua estimação. Veremos esto mais claramente nas simulações.

Demonstrou-se nos teoremas 5.1 e 5.2 que o modelo de neurônio de reconhecimento adaptativo é não polarizado e de mínima variância numa aproximação de primeira ordem. Estes mesmos teoremas poderiam-se aplicar a uma rede de neurônios de reconhecimento adaptativo de uma camada. Porém uma conseqüencia inevitável da rede de neurônios de reconhecimento adaptativo de mais de uma camada é a ***perda da condição de otimalidade***. Esta é uma condição intrínseca à estrutura proposta e não depende de condições externas, isto é, mesmo que a perturbação fosse de tipo ruído branco e fosse válida a relação linear entre o padrão e a perturbação. Os filtros de Kalman das camadas intermediárias realizam uma estimação sobre as saídas dos neurônios das primeiras camadas e como estas tem uma transferência não linear, mesmo que a perturbação fosse ruído branco, a perturbação da saída não o seria. Então os filtros de Kalman das camadas intermediárias atuam numa condição de subotimalidade independentemente das condições externas, o qual se transfere ao modelo de rede de reconhecimento adaptativo.

5.8.- ASPECTOS DE IMPLEMENTAÇÃO.

Um aspecto importante desta proposta é o ***custo computacional*** do conjunto, dado que para aplicações reais as redes podem ter dimensões apreciáveis. Sem dúvida este esquema proposto é mais custoso computacionalmente que a rede neural clássica. O maior custo do filtro de Kalman é a inversão de matriz. O custo da inversão de matriz é proporcional à dimensão do vetor elevada a três. Vamos a estudar nesta seção mecanismos para aliviar este grande custo computacional.

Faz-se necessário diferenciar o problema do custo computacional nas duas etapas do desenvolvimento de um sistema de reconhecimento de voz com o sistema proposto. Quanto ao treinamento, como o algoritmo "backpropagation" utilizado requer o uso de toda a base de dados em várias iterações, para diminuir o custo computacional desta etapa, cria-se uma nova base de dados a partir da disponível, filtrando a mesma com filtros de Kalman da primeira camada, uma única vez. Este é um mecanismo válido já que o treinamento modifica os pesos posteriores destes filtros e os padrões de treinamento são não ruidosos.

Para os filtros de Kalman da camada de saída a seguinte **hipótese** foi feita: como eles estão para estimar o verdadeiro sinal a partir de suas amostras ruidosas e no treinamento, as saídas das camadas intermediárias são não ruidosas, sua utilidade é nula nesta etapa. *Portanto, os neurônios das camadas de saída são treinados sem os filtros de Kalman.* Outra necesidade desta hipótese e a dificuldade matemática de tratar a propagação do gradiente através do filtro de Kalman. Nos experimentos observaremos se esta hipótese foi acertada.

Dado que as entradas dos neurônios da primeira camada são as mesmas, pode-se ***unir os filtros de Kalman de uma camada num único filtro de Kalman*** . Isto explica-se a partir de que todos os filtros vão estimar o mesmo padrão de modo que estariamos fazendo cálculos redundantes. Então a rede neural de reconhecimento adaptativo com uma primeira otimização apresenta-se na figura 5.7, em vez da figura anterior.

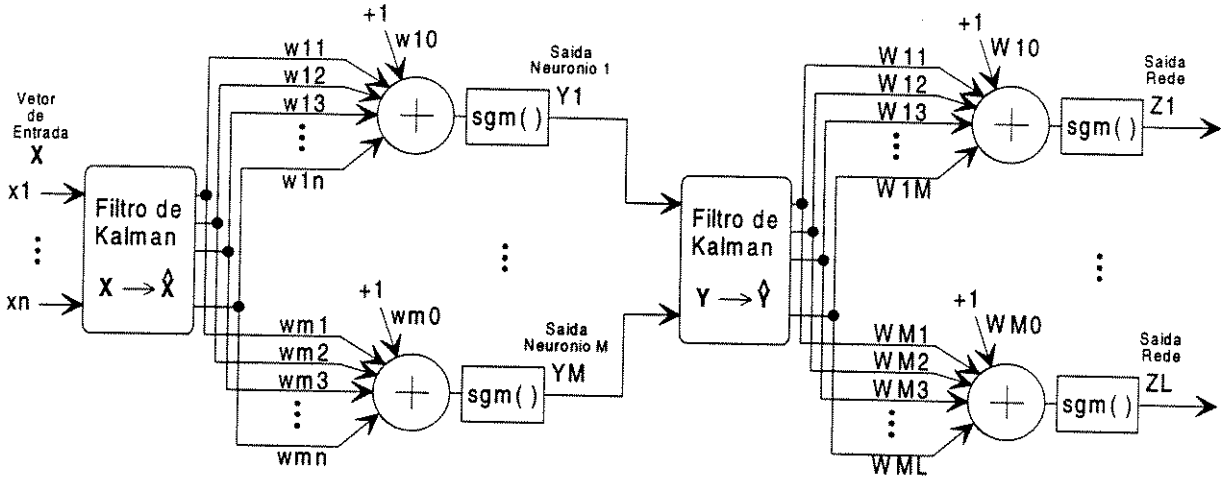


FIGURA 5.7 - Rede neural com KF únicos por camada.

O maior custo computacional dos filtros de Kalman provém da inversão da matriz de auto correlação do processo de inovações (equação (3.38) e seção 3.27 do capítulo 3). Porém dependendo de como são as equações de estado, é possível que esta ***matriz*** seja ***escalar***, aliviando muito o custo computacional do algoritmo.

Em casos onde a matriz não seja escalar, uma forma de minimizar este peso computacional, é considerar o uso de ***filtros de Kalman escalares***. Estes aplicam-se se na equação de estado quando for possível separar a matriz de transição de estado em dois sistemas diferentes. Mas mesmo que esta hipótese não seja válida, com o objetivo de tornar possível a aplicação prática desta proposta postulamos esta separabilidade e trabalhamos com filtros de Kalman escalares. Um diagrama da rede com esta última modificação apresenta-se na figura 5.8.

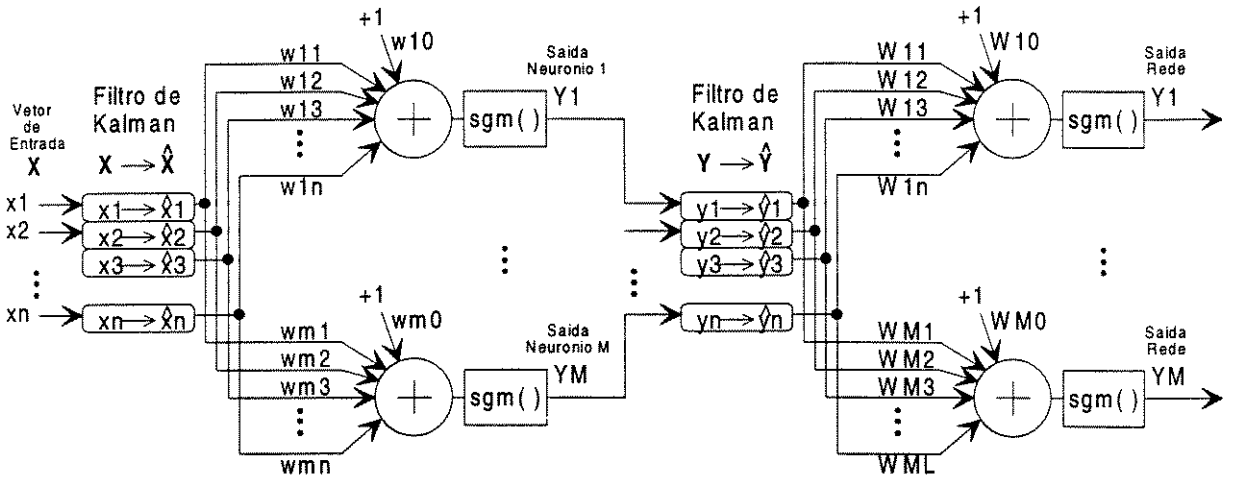


FIGURA 5.8 - Rede neural com KF escalares.

Outra condição importante para a aplicação prática desta rede proposta é a ***característica de adaptabilidade.*** Postulou-se que a classificação pelo neurônio de reconhecimento adaptativo baseia-se na disponibilidade de várias amostras de um mesmo padrão. Porém o padrão pode mudar com o tempo isto é, o sistema deve adaptar-se de forma tal de poder modificar a estimação na medida que muda o padrão. Para este fim utilizam-se os filtros de Kalman adaptativo, apresentados no capítulo 3, seção 3.3.2. Agora surge outro problema, como escolher os fatores de esquecimento. Podem ser escolhidos individualmente, por camadas ou um para toda a rede. Parece razoável a eleição por camadas, como se verá nas simulações, já que os filtros de Kalman de cada camada se enfrentam a situações diferentes de estimação.

5.9.- APLICAÇÃO: CLASSIFICAÇÃO DE SÉRIES DE PADRÕES PARA O PROBLEMA XOR.

Como segundo exemplo de reconhecimento de padrões ruidosos, treinou-se uma rede neural num ambiente não ruidoso para resolver o problema XOR. Este, se caracteriza pela atribuição de um valor unitário, se os dois sinais de entrada são diferentes, e nulo no caso contrário. A rede neural que resolve este problema, tem duas camadas com dois neurônios na camada oculta e um na camada de saída. A rede neural de reconhecimento adaptativo tem a mesma estrutura que a rede clássica, mas está formada por neurônios de reconhecimento adaptativo.

Realizaram-se provas de reconhecimento de séries de padrões, utilizando os padrões binários **[-1,-1], [1,-1], [1,1],** afetados por ruído branco gaussiano em diferentes relações sinal - ruído. A duração total do vetor de reconhecimento é de 600 observações

correspondendo 200 observações a cada um dos padrões. Resultados similares foram obtidos com outras séries de padrões.

Devido a que o filtro de Kalman tem memória infinita, isto é, usa toda a informação desde o começo da experiência para a estimação, inseriu-se um fator de esquecimento na equação recursiva da matriz de auto correlação do erro de estimação [8]. Ao contar com uma memória finita o filtro de Kalman pode estimar vetores que mudam ao longo do tempo. Experimentou-se com distintos valores e os melhores resultados obtiveram-se com 1.2 para os neurônios da primeira camada e 1.05 para o neurônio da segunda camada. Deste modo, este último neurônio tem uma memória de maior duração.

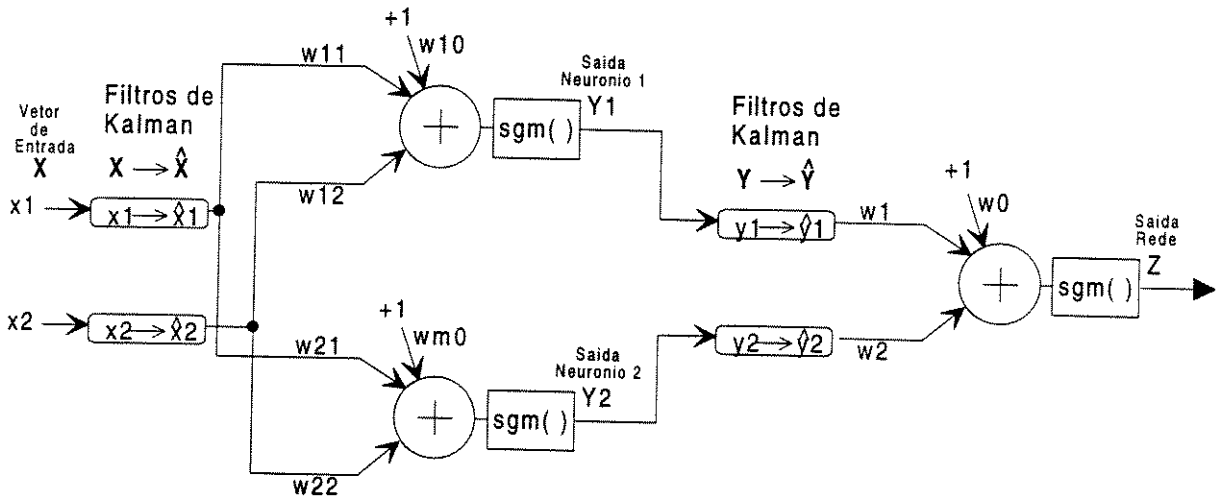


FIGURA 5.9 - Diagrama rede para estimação do problema XOR

Apresentamos a seguir do diagrama da estrutura da rede, os gráficos que comparam as saídas desejadas com as correspondentes às das redes neural clássica e de reconhecimento adaptativo, para os seguintes níveis de SNR: 20dB, 10dB, 7.5dB, 5dB, 2.5dB, 0dB, -2.5dB, -5dB

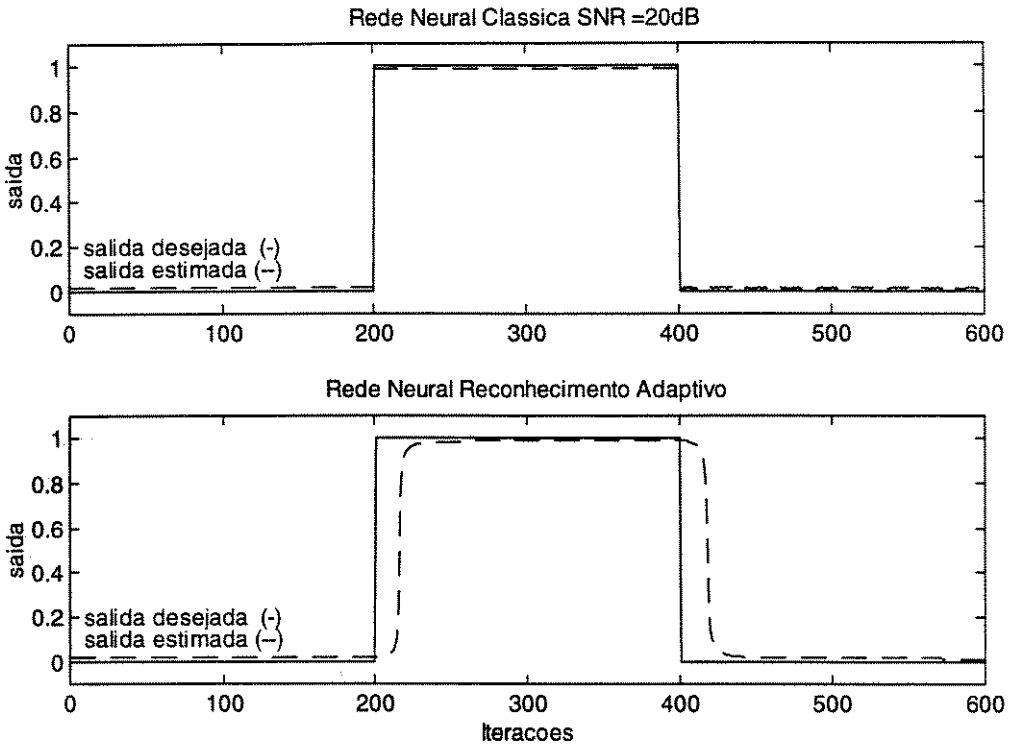


FIGURA 5.10: Evolução temporal das saídas das redes para SNR de 20dB

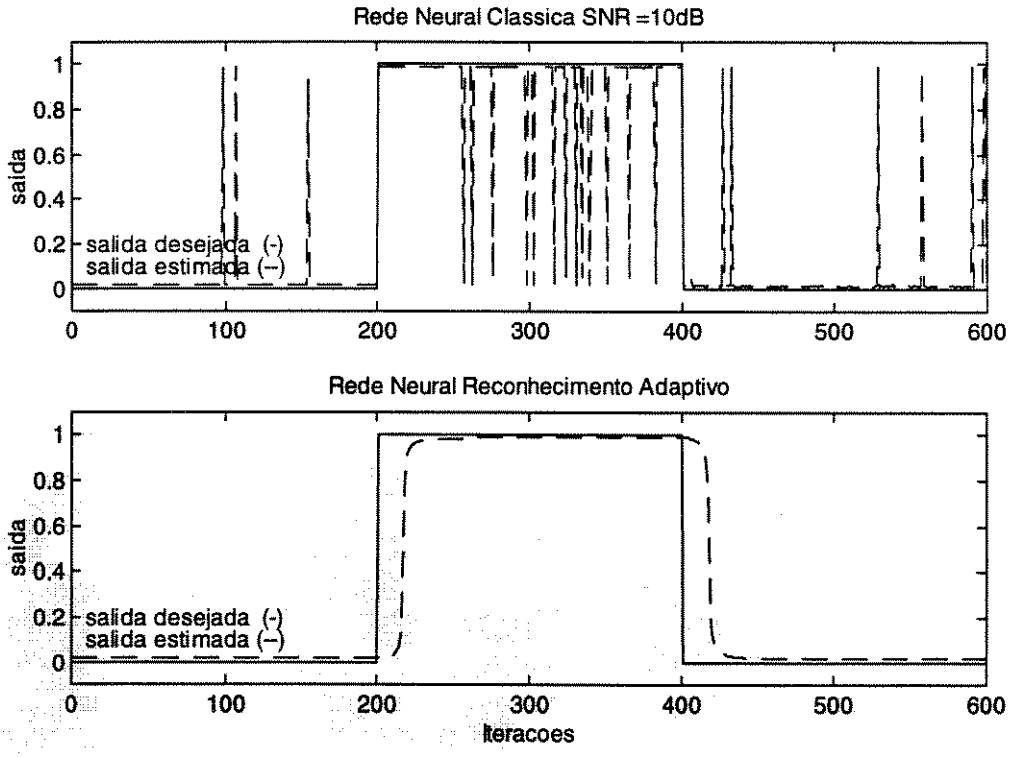


FIGURA 5.11: Evolução temporal das saídas das redes para SNR de 10dB

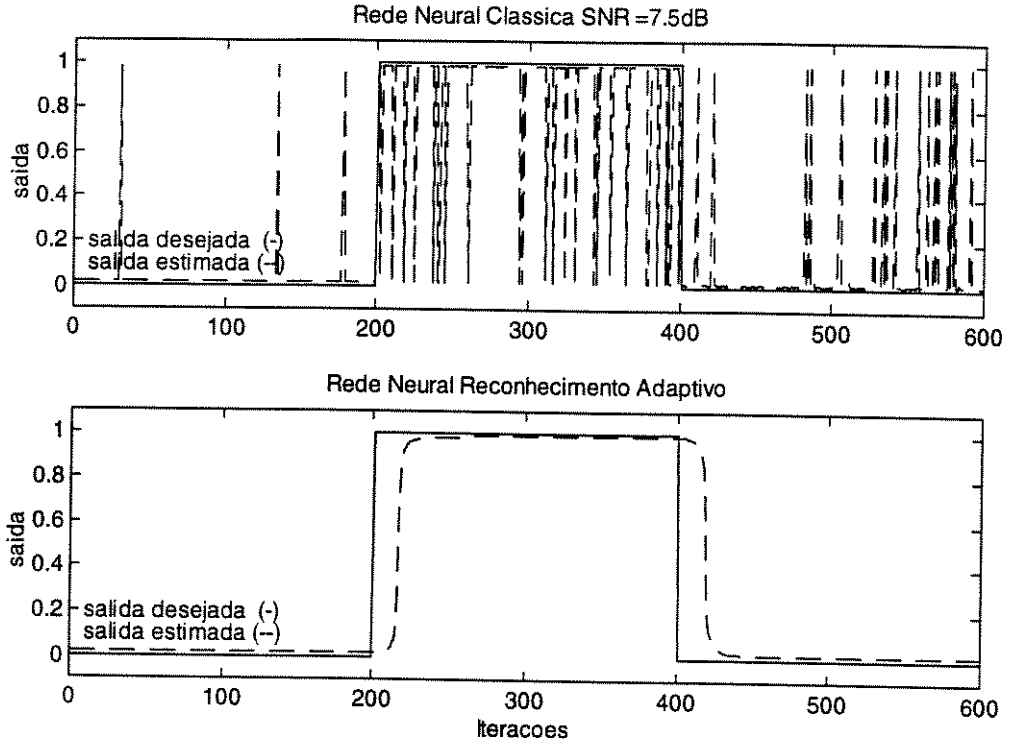


FIGURA 5.12: Evolução temporal das saídas das redes para SNR de 7.5dB

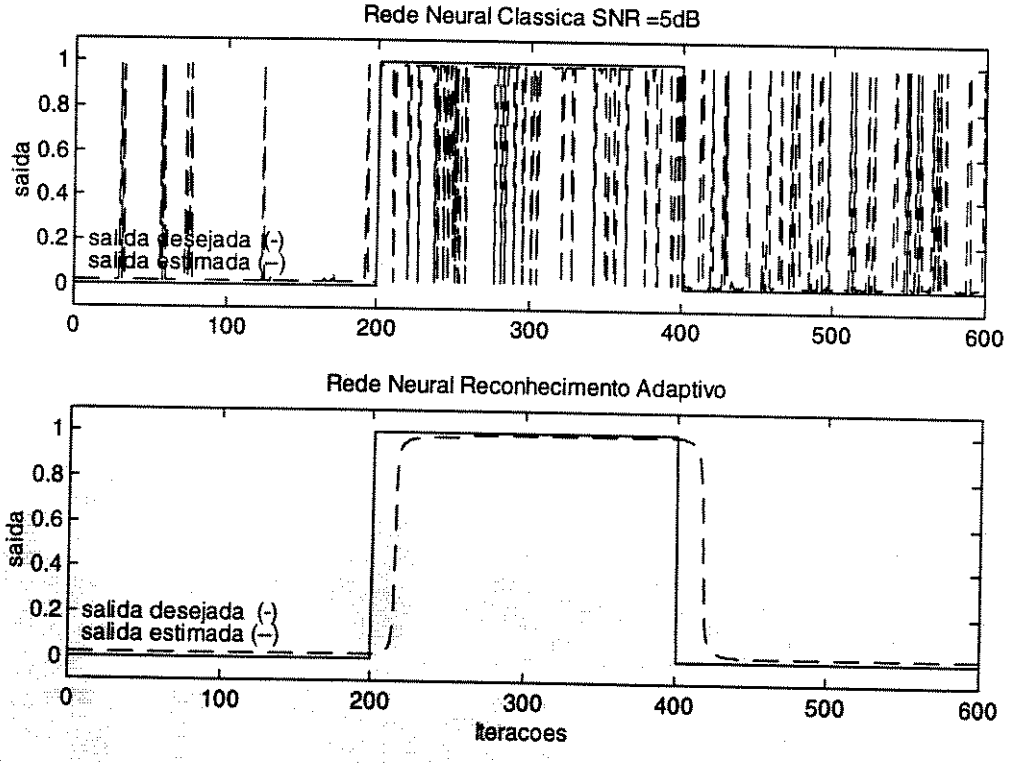


FIGURA 5.13: Evolução temporal das saídas das redes para SNR de 5dB

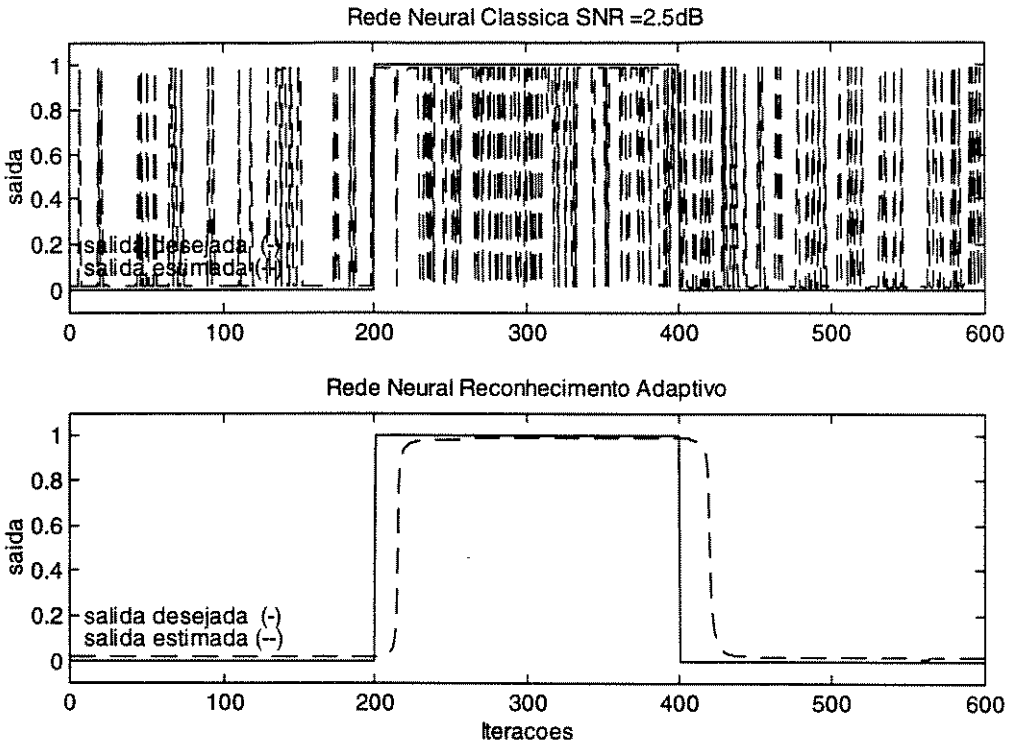


FIGURA 5.14: Evolução temporal das saídas das redes para SNR de 2.5dB

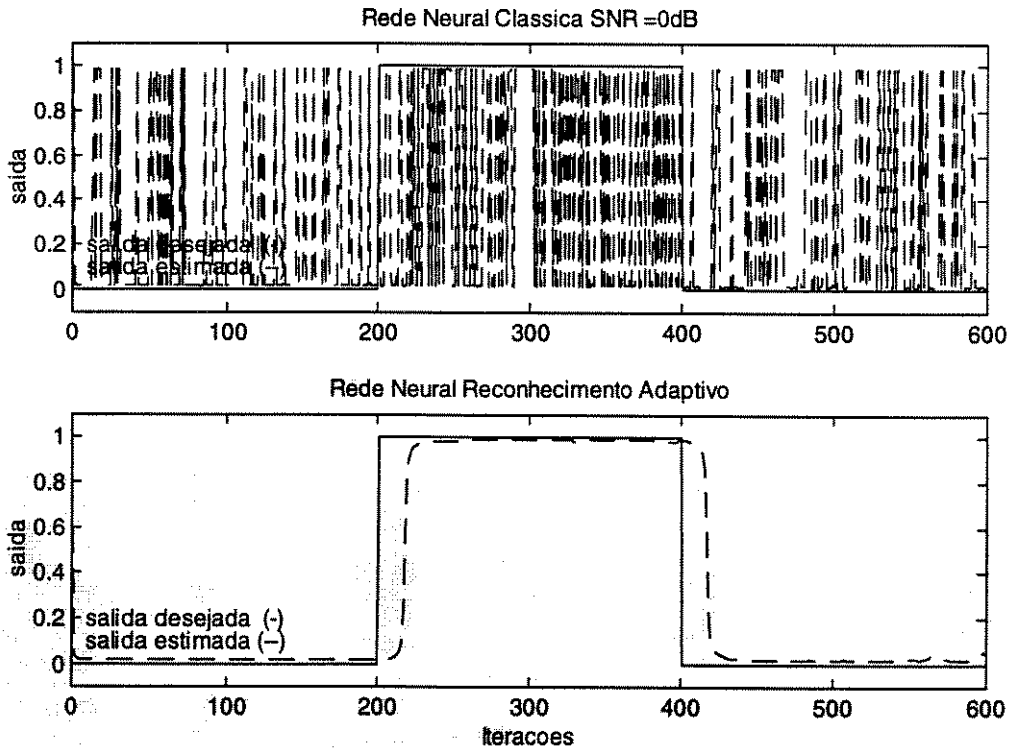


FIGURA 5.15: Evolução temporal das saídas das redes para SNR de 0dB

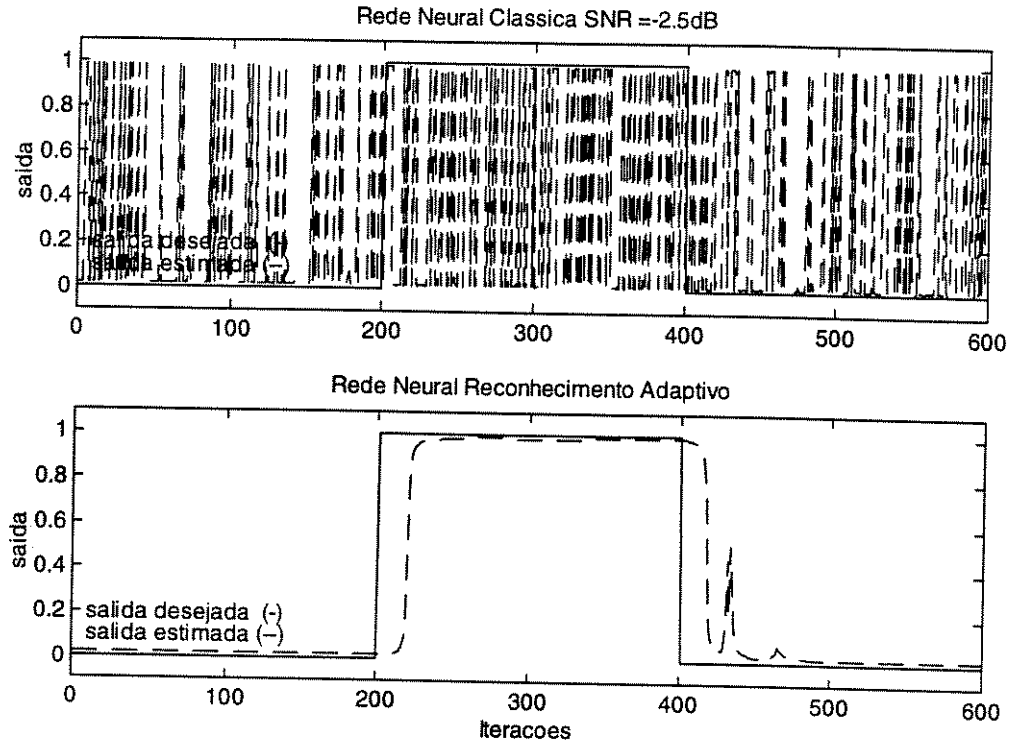


FIGURA 5.16: Evolução temporal das saídas das redes para SNR de -2.5dB

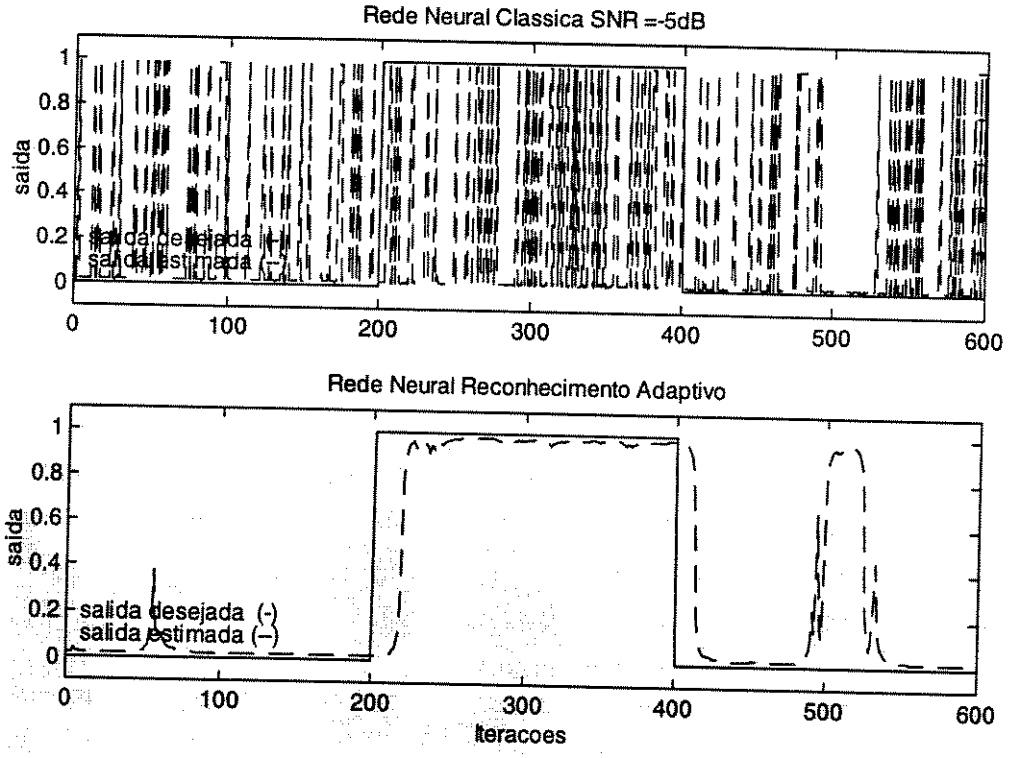


FIGURA 5.17: Evolução temporal das saídas das redes para SNR de -5dB

Apresenta-se à continuação a tabela 5.2 que resume os resultados de reconhecimento das séries já descritas em diferentes SNR, para as redes clássica e de reconhecimento adaptativo. O índice de reconhecimento foi computado comparando-se cada uma das saídas desejadas com as obtidas ao longo de todo o vetor (ponto a ponto), de forma tal que, se a diferença é menor de 0.5, se considera reconhecida, e caso contrário se for maior que 0.5.

SNR / Rede Neural	20 dB	10 dB	7.5 dB	5 dB	2.5 dB	0 dB	-2.5 dB	-5 dB
Clássica	**100 %**	**97.7 %**	**92.7 %**	**86.1 %**	**76.9 %**	**71.5 %**	**64 %**	**62.3 %**
Reconhec. Adapt.	**94 %**	**94.2 %**	**94 %**	**94 %**	**94.1 %**	**94.1 %**	**93 %**	**92.5 %**

TABELA 5.2 : Resultados Reconhecimento Série de Padrões Rede Neural Clássica / Adaptativo (média 5 repetições)

Na tabela 5.2 revela-se o verdadeiro potencial da rede de reconhecimento adaptativo. Enquanto a potência de ruído aumenta desde a décima parte (10dB SNR) a mais do triplo da potência do sinal (-5dB SNR), o índice de reconhecimento da rede proposta somente cai desde 94.2% até 92.5%. Quer dizer, enquanto que a potência de ruído aumenta 32 vezes, a porcentagem de reconhecimento sofre uma diminuição mínima.

Esta desempenho se obtém através de um ***duplo mecanismo:***.

- Por um lado, a ***estrutura em camadas*** dos neurônios de reconhecimento adaptativo ***reforça o processo de estimação***, no sentido de sucessivos filtragems do sinal.

- Por outro lado, como pode ser visto nas figuras anteriores, a memória finita dos filtros de Kalman provoca uma ***inércia na mudança de classificação.*** Esta é uma propriedade muito importante já que, mesmo que os neurônios intermediários indiquem através de suas saídas, que a estimação do vetor limpo mudou de classe, o último neurônio não mudará a saída da rede ao menos que, sua estimação sobre as saídas da camada intermediária não mude de classe. Isto permite que um vetor possa serem atribuido a uma classe, mesmo que suas observações ruidosas caiam fora de sua classe, ou sua estimação pelos neurônios, intermediários caia temporariamente fora da mesma.

Este mecanismo constitui justamente a ***essência do reconhecedor robusto***, isto é, prover ***mecanismos internos robustos de classificação*** frente a perturbações.

Este comportamento descrito pode ser observado nos gráficos seguintes onde apresentam-se a evolução dos sinais intermediários das redes neurais clássica e de reconhecimento adaptativo. Reproduzimos a seguir a figura 5.9 onde se indicam as saídas correspondentes as próximas figuras.

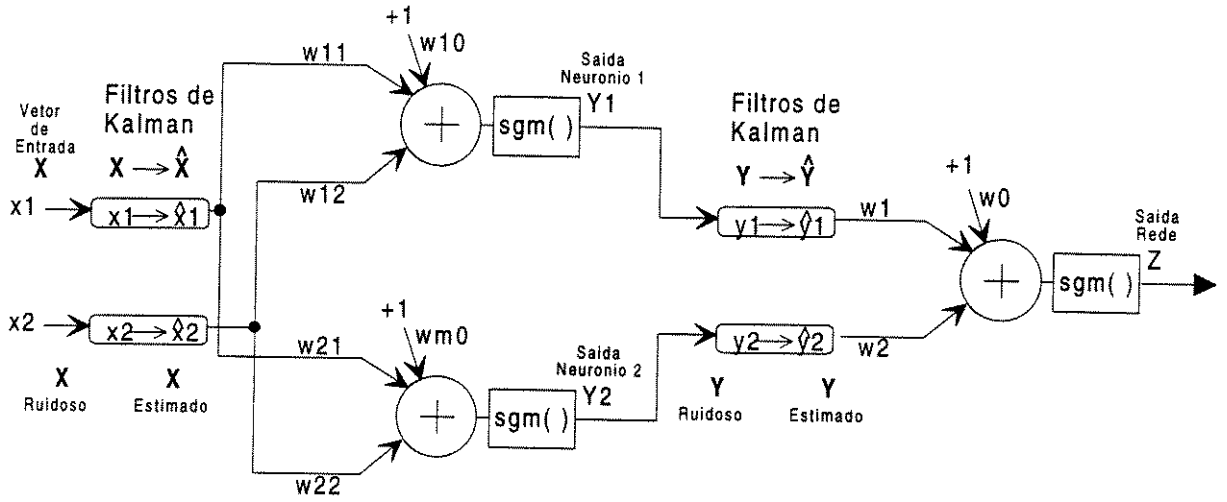


FIGURA 5.9 (Repetida)- Diagrama rede para estimação do problema XOR

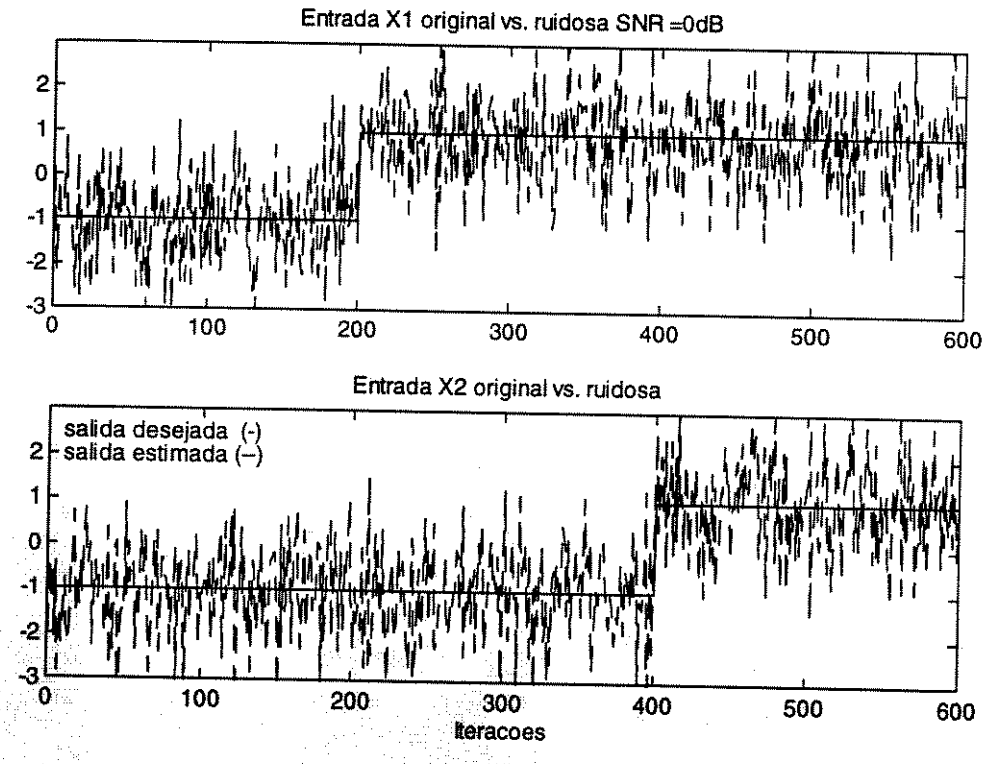


FIGURA 5.18: Evolução temporal das entradas original e ruidosa para SNR de 0dB

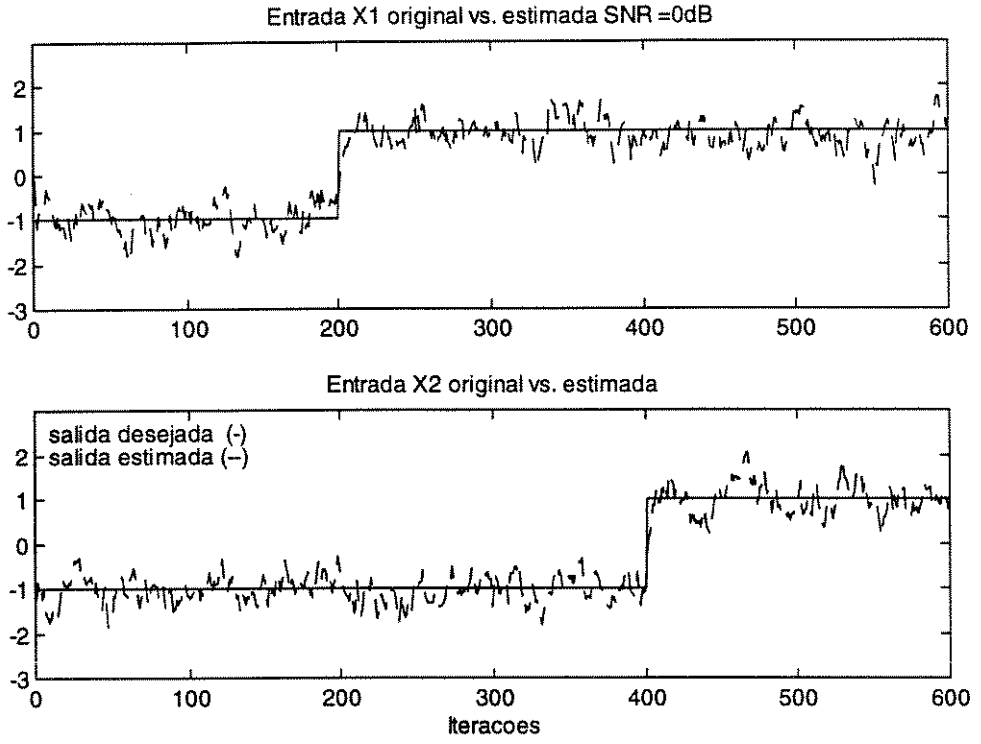


FIGURA 5.19: Evolução temporal das entradas original e estimada para SNR de 0dB

Observamos nas figuras 5.18 e 5.19 uma grande melhoria na estimação dos valores originais dos padrões que efetuam os filtros de Kalman da primeira camada para um SNR de 0dB. Esta é uma primeira melhoria do sistema, no sentido que simplifica o problema das camadas seguintes.

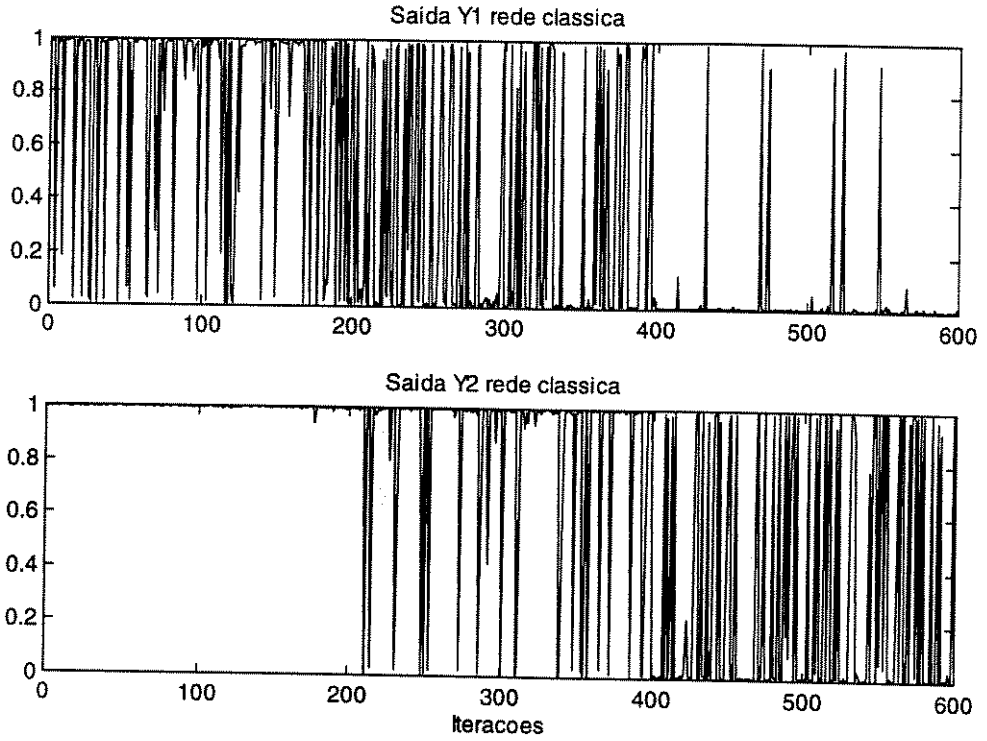


FIGURA 5.20: Evolução temporal das saídas Y da rede clássica para SNR de 0dB

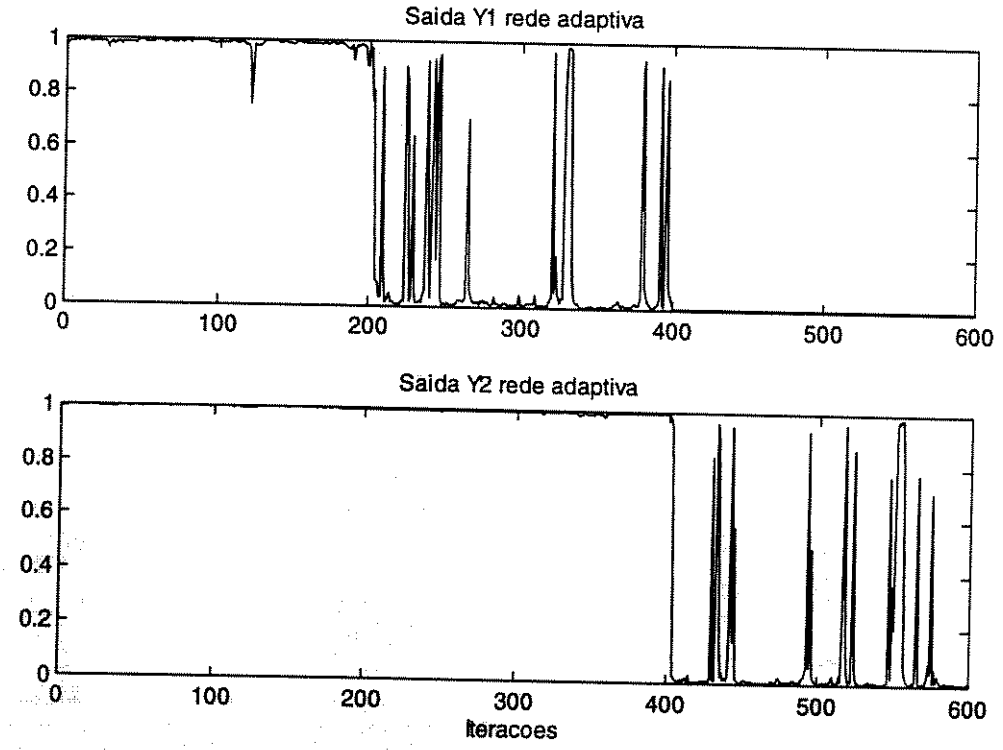


FIGURA 5.21: Evolução temporal das saídas Y para a rede adaptativa para SNR de 0dB

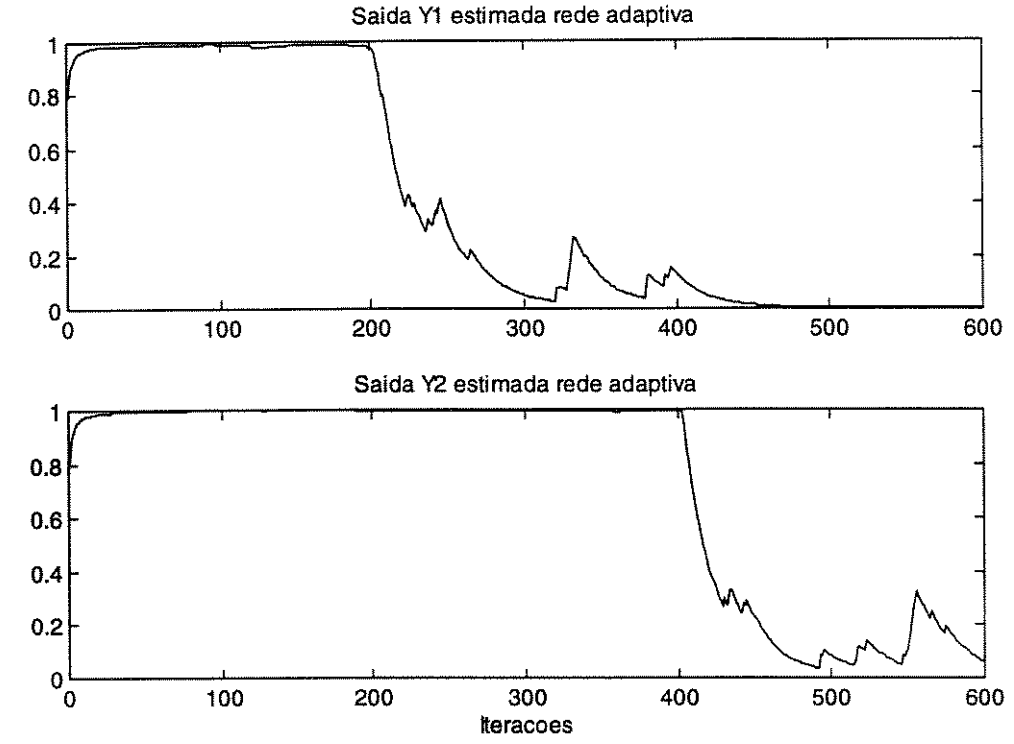


FIGURA 5.22: Evolução temporal das saídas Y Estimadas para a rede adaptativa para SNR de 0dB

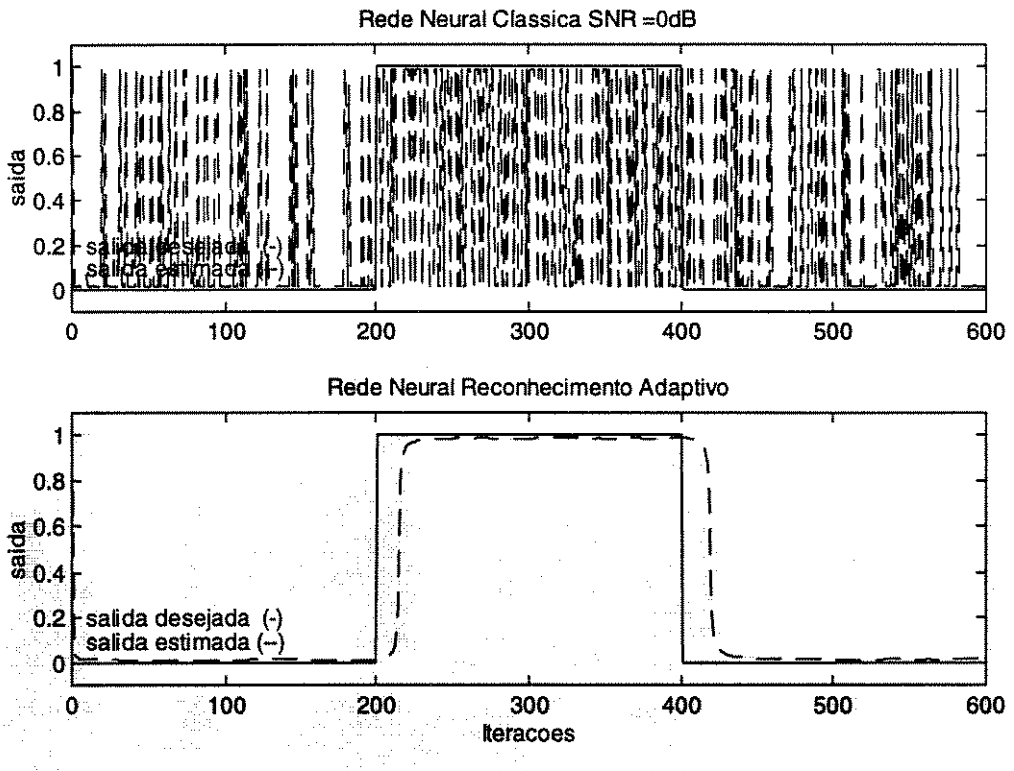


FIGURA 5.23: Evolução temporal das saídas das redes para SNR de 0dB

Nas figuras 5.20, 5.21, 5.22 e 5.23 observamos as saídas dos neurônios da primeira camada para a rede clássica e as saídas e suas estimações dos neurônios da primeira camada de reconhecimento adaptativo, respectivamente. Comparando as figuras 5.20 e 5.21, vemos as melhorias das saídas da rede de reconhecimento adaptativo, respeito das saídas da rede clássica, produto das estimações dos filtros de Kalman da primeira camada. É importante destacar que mesmo sendo significativa a melhoria, alguns erros no são totalmente corrigidos. É aqui onde atuam os filtros de Kalman da segunda camada. Observamos como os erros de classificação da figura 5.21, são filtragens já que devido à inércia de classificação, somente produzem uma pequena mudança no valor da estimação que não é acusada na saída (ver figura 5.23). Conclui-se que os resultados da rede de reconhecimento adaptativo estão muito mais próximos dos valores desejados que os da rede clássica.

Observamos na figura 5.21, outro possível erro da rede. Os filtros de Kalman precisam de uma estimação inicial e esta, na falta de informação a priori, é escolhida aleatoriamente. Se esta estimação inicial não cai na classe correta (definida pelo vetor original atrás das observações ruidosas), será indicada na saída. Depois de umas poucas iterações, começam a atuar os mecanismos descritos e obtém-se uma estimação mais confiável do vetor original, até cair na classe correta.

É importante observar a ***dependência*** que existe entre o ***índice de reconhecimento e as condições do problema, para o modelo de rede proposto***. Uma das condições do experimento realizado foi tomar uma longitude do padrão de 600 pontos. Suposta a existência de três estados, obtivemos como resultado 200 pontos por classe. Esta especificação define um dado índice de reconhecimento, já que o modelo proposto é adaptativo, o qual é função da relação, por um lado, entre os retardos e a adaptação, e por outro lado entre a parte estável do padrão. Na próxima tabela 5.3 observaremos os índices de reconhecimento das redes clássica e de reconhecimento adaptativo, para uma ***SNR fixa de* 0dB**, ***modificando a longitude total do padrão*** utilizado na experiência anterior isto é, com três classes.

Longitude Total	75 pontos	150 pontos	300 pontos	600 pontos	1200 pontos
Rede Clássica	70.30 %	71.16 %	68.16 %	70.16 %	70.58 %
Rede Reconhec. Adaptativo	56.66 %	77.05 %	86.83 %	94.08 %	97.25 %

TABELA 5.3 : Resultados Reconhecimento Série de Padrões Rede Neural Clássica / Adaptativo (média 5 repetições), para SNR = 0dB

Observa-se na tabela 5.3 como a porcentagem da rede clássica se mantém constante para diferentes tamanhos de padrão total, já que somente depende da relação sinal-ruído constante durante a experiência. A rede de reconhecimento adaptativo, pelo contrário depende da longitude do padrão, passando de porcentagens baixas de reconhecimento em padrões de curta longitude, para muito bons índices em padrões de alta longitude.

Por comparação entre as tabelas 5.2 e 5.3 observa-se que o modelo proposto, para as condições deste problema, revela robustez ao ruído pelo uso das redundâncias para correlacionar a estimação do padrão atual com os anteriores. Porém a dependência das redundâncias o torna mais sensível a padrões menos redundantes. A rede clássica é dependente da relação sinal-ruído, já que esta é a que determina, estatisticamente, a porcentagem de padrões que cairão dentro da classe original, isto influi diretamente na sua saída por sua classificação instantânea do padrão de entrada. Já que esta porcentagem é uma medida relativa, o modelo clássico não é dependente da longitude total do padrão.

CAPÍTULO 6

REDE NEURAL DE RECONHECIMENTO ADAPTATIVO APLICADA AO RECONHECIMENTO DE VOZ

6.1- INTRODUÇÃO

Nesta seção exploramos a aplicação do modelo de rede proposta ao problema do reconhecimento ruidoso de fonemas. Não existindo uma forma direta de aplicação do modelo, serão propostas duas arquiteturas diferentes, tomando em consideração as características particulares do problema. É fundamental para o sucesso de sua aplicação, assim como em qualquer outra aplicação do presente modelo a problemas reais, a satisfação das propriedades por um lado do modelo e pelo outro do problema de reconhecimento, detalhadas anteriormente.

Neste sentido a aplicação do modelo proposto não se restringe ao reconhecimento de voz ou de sinais temporais, podendo aplicar-se ao reconhecimento de padrões distribuídos espacialmente, como o reconhecimento de dígitos impressos ou de caracteres manuscritos.

Os melhores sistemas de reconhecimento de voz, tais como HMM e DTW, baseiam seu reconhecimento de baixo nível numa representação espectral do sinal da voz. O ouvido humano também realiza uma decomposição do sinal temporal da voz em uma representação espectral. Neste sentido será feito um esforço para a extração de uma representação espectral da voz para sua classificação ruidosa de acordo com as características próprias do modelo proposto.

No presente capítulo se apresentam três arquiteturas baseadas numa caracterização espectral do sinal de voz, para o reconhecimento ruidoso de fonemas. Realiza-se um experimento de treinamento e de reconhecimento ruidoso numa base de dados de dois falantes para diferentes relações sinal - ruído.

6.2 VANTAGENS E DESVANTAGENS

Uma das motivações do modelo proposto é a utilização das redundâncias presentes em muitos problemas de reconhecimento de padrões. A utilização efetiva destas, quando um sinal é contaminado por outro aleatório descorrelacionado, tem por objeto melhorar a estimação do sinal original. Em realidade a existência de redundâncias, se não está

disponível nenhum outro conhecimento a priori, é uma condição necessária para a existência de um estimador. Portanto a exigência de redundâncias não é um grande requerimento para problemas reais. Porém dependendo de cada problema em particular mudará a quantidade de redundâncias disponíveis.

A voz é um dos sinais que se encaixa no contexto dos sinais redundantes. A causa biológica desta redundância é a relação entre o lento movimento do sistema fonador humano, devido a sua inércia, e uma alta freqüência de sinal. É importante distingüir, de acordo a que parte do sistema fonador se utiliza, que nem todos os fonemas tem o mesmo grau de redundância. As vogais pertencem ao conjunto de alta redundância, por ser geradas com todo o sistema fonador e as plosivas ao de baixa redundância, por ser labiais e/ou dentais. A amostragem possibilita capturar estas redundâncias. Isto pode visualizar-se observando a relação entre a freqüência de amostragem (entre 8 kHz até 20 kHz) e a escala de tempo da duração típica dos fonemas (entre 15 ms. a 25 ms.). No pior dos casos (freqüência de amostragem de 8 kHz e duração de 15 ms.) teremos umas 120 amostras para representar uma duração típica.

A forma de aproveitar estas redundâncias para extrair informação estatística do sinal é outro problema. O tipo de processamento temporal do sinal de voz proposto, representa uma proposta diferente à tradicionalmente utilizada. Em geral utilizam-se janelas de duração fixa com "overlap". A duração destas janelas temporais, está determinada em função da duração mínima típica dos fonemas. Este enfoque tradicional limita a utilização de redundâncias à longitude da janela, neste sentido permite ao sistema uma memória fixa de curto prazo. O tipo de processamento seqüencial, sem janelas, do modelo proposto utiliza uma memória de maior duração, determinada pelo coeficiente de esquecimento dos filtros de Kalman. Neste sentido permitiria um melhor aproveitamento da informação redundante, a qual é fundamental para melhorar o reconhecimento de fonemas ruidosos.

A voz, é para o ser humano, um meio para a transmissão de mensagens. Estas mensagens estão contidas nas palavras, formadas por uma seqüência ordenada de eventos fonéticos. Nas aplicações práticas de vocabulário restringido, deseja-se dispor de um reconhecedor de palavras. O modelo proposto é um classificador que se adapta naturalmente ao reconhecimento de eventos fonéticos. Sua aplicação ao reconhecimento de palavras precisaria de um módulo superior para integrar as saídas da rede de reconhecimento adaptativo e tomar a decisão acerca da classificação de palavras. Este módulo poderia ser tanto um modelo de Markov, um classificador por DTW ou uma rede neural recorrente.

Uma das dificuldades da aplicação do modelo proposto é que a extração de características do sinal de voz ***não deve incluir nenhuma compressão temporal da informação existente***. Isto é, por cada amostra do sinal de voz deve existir um vetor de características. A razão deste requerimento é a necessidade de preservar a escala de tempo do sinal de voz para que os filtros de Kalman possam realizar o processo de estimação sobre a mesma escala de tempo.

6.3 ARQUITETURA DE BANCO DE FILTROS

6.3.1 Idéia da proposta

De acordo com os requerimentos descritos do modelo em quanto à preservação da escala de tempo da voz, uma idéia natural seria transformar o sinal temporal da voz num conjunto de sinais temporais, com a mesma escala de tempo, como saídas de filtros digitais do tipo passa-faixa. Se implementará então uma estrutura de banco de filtros passa-faixa para o preprocesamento da sinal de voz com o objetivo de transformá-la num vetor temporal de componentes espectrais.

6.3.2 Desenho do Banco de Filtros

O desenho dos filtros requer uma série de especificações como a quantidade de filtros, sua posição em freqüência, seu tipo e sua ordem. Existem também diversos critérios de desenho dos filtros.

A eleição da **quantidade de filtros** revela uma relação de compromisso entre a discriminação em freqüência desejada e a complexidade de desenho dos filtros. Por um lado deve haver uma quantidade adequada de filtros para poder separar faixas de freqüência adjacentes, com a precisão desejada. Pelo outro lado, quanto maior seja o número de filtros, menor é o largura de faixa dos mesmos e conseqüentemente mais rigorosa é a especificação para seu desenho. Para os sinais de voz a discriminação em freqüência não precisa ser muita e usualmente com quinze a vinte filtros logram-se resultados adequados.

A eleição da **distribuição em freqüência** do banco de filtros condiciona as especificações respeito de freqüência central e largura de faixa dos filtros. Uma eleição direta seria implementar o banco de filtros numa escala linear. Por outro lado, por razões biológicas, é sabido que esta não é a melhor eleição. Estudos sobre audição, desenvolvidos em 1940 por Stevens e Volkman, revelaram que a escala em freqüência do ouvido humano não é linear. Estes investigadores realizaram um mapeamento entre a freqüência física e a

percibida, para o ouvido humano. Esta é a conhecida escala MEL, a qual é aproximadamente linear até 1000 Hz e logaritmica depois. A distribuição em freqüência do banco de filtros adotada se corresponde aproximadamente com a escala MEL e detalha-se a seguir na tabela 6.1:

Número de filtro	Freqüência Central (Hz)	Largura de Faixa (Hz)
1	150	100
2	250	100
3	350	100
4	450	110
5	570	120
6	700	140
7	840	150
8	1000	160
9	1170	190
10	1370	210
11	1600	240
12	1850	280
13	2150	320
14	2500	380
15	2900	450
16	3400	550

TABELA 6.1: Escala de freqüências MEL.

A seguinte eleição é a do **tipo de filtros**. Podem ser do tipo AR, MA ou ARMA. De acordo a qual modelo escolhido se determinará a sua ordem. É sabido que dada uma especificação em freqüência de um filtro passa faixa a ordem do filtro MA, por ser de zeros, precisa ser maior que o correspondente filtro AR, por ser de pólos.

A operação de filtragem é uma operação linear já que a equação a diferenças dos filtros propostos é linear. ***Estudaremos a seguir como esta operação linear conserva a relação aditiva entre o sinal e o ruído para ambos filtros***. Realizaremos as demonstrações para filtros MA de ordem 2 e para filtros AR de ordem 1, as conclusões são facilmente extrapoláveis a ordens maiores.

Sejam o sinal de entrada no tempo t, x(t), o sinal ruidoso no tempo t, n(t), e o sinal de saída do filtro (AR ou MA) no tempo t, y(t). A equação a diferenças do filtro MA(2) no caso sem ruído é: $y(t) = a\,x(t) + b\,x(t-1)$, onde os coeficientes a, b $\in \Re$. A equação a diferenças do filtro AR(1) no caso sem ruído é: $y(t) = a\,y(t-1) + x(t)$, onde o coeficiente a $\in \Re$. O sinal n(t) é um ruído branco gaussiano.

para filtros MA(2): (no caso ruidoso)

$$\begin{aligned} y(t) &= a\,(x(t)+n(t)) + b\,(x(t-1)+n(t-1)) = \\ &= \{a\,x(t)+ b\,x(t-1)\} + \{a\,n(t)+ b\,n(t-1)\} = \\ &= y_x(t) + y_n(t) \end{aligned} \tag{6.1}$$

para filtros AR(1): (no caso ruidoso)

$$\begin{aligned} y(t) &= a\,y(t-1) + (x(t)+n(t)) = \\ &= a\,(a\,y(t-2) + x(t-1)+n(t-1)) + (x(t)+n(t)) = \\ &= a^2\,y(t-2) + a\,x(t-1)+x(t) + a\,n(t-1)+n(t) = \\ &= a^2\,(a\,y(t-3) + x(t-2)+n(t-2)) + a\,x(t-1)+x(t) + a\,n(t-1)+n(t) = \\ &= a^3\,y(t-3) + (a^2\,x(t-2)+a\,x(t-1)+x(t)) + (a^2\,n(t-2)+a\,n(t-1)+n(t)) = \\ &= \ldots \\ &= \{\textstyle\sum_{i=0}^{t} a^i x(t-i)\} + \{\sum_{i=0}^{t} a^i n(t-i)\} \\ &= y_x(t) + y_n(t) \end{aligned} \tag{6.2}$$

Onde na equação (6.1) simplesmente aplicamos a propriedade linear do produto respeito da soma. Na equação (6.2), na segunda linha, substituimos y(t-1) por sua expressão e na terceira linha aplicamos a propriedade linear do produto respeito da soma e temos reorganizado os termos (supondo y(-1) = 0).

Concluimos de ambas equações (6.1) e (6.2) que conserva-se a relação aditiva entre o sinal e o ruído já que a saída de ambas pode-se dividir como uma soma entre a saída devida ao sinal e a devida ao ruído. Pela forma da saída devida ao ruído este se comporta como um correlacionador do ruído branco.

O filtro MA por ser de médias movéis, isto é por ser uma combinação linear ponderada das amostras do sinal durante uma janela de tempo, é um correlacionador de tantas amostras como a ordem do filtro. Então quanto ***maior a ordem do filtro MA, mais longe estará o ruído de ser branco***.

Recordando que a função de autocorrelação R(m) do sinal de saída ruidoso $y_n(t)$ do filtro MA(2) (equação (6.1)), define-se como $R(m) = E[y_n(t)y_n(t-m)]$. Recordando que o ruído é branco gaussiano, portanto se E[•] é a operação valor esperado, então $E[n(t)n(t)]= \sigma^2\ \forall t$ e $E[n(t)n(t-i)]= 0,\ \forall t,\ \forall i\neq 0$. Desenvolveremos a seguir os diversos termos desta função, R(0), R(1) e R(2):

$$E[y_n(t)^2] = E[(a\,n(t)+ b\,n(t-1))(a\,n(t)+ b\,n(t-1))] =$$

$$= a^2E[n(t)^2] + 2abE[n(t)n(t-1)] + a^2E[n(t-1)^2] =$$

$$= a^2\sigma^2 + 0 + b^2\sigma^2 = \mathbf{a^2\sigma^2 + b^2\sigma^2} \qquad (6.3)$$

$$E[y_n(t)y_n(t-1)] = E[(a\,n(t)+ b\,n(t-1))(a\,n(t-1)+ b\,n(t-2))] =$$

$$= a^2E[n(t)n(t-1)] + abE[n(t)n(t-2)] + abE[n(t-1)^2] + b^2E[n(t-1)n(t-2)]=$$

$$= 0 + 0 + ab\sigma^2 + 0 = \mathbf{ab\sigma^2} \qquad (6.4)$$

$$E[y_n(t)y_n(t-2)] = E[(a\,n(t)+ b\,n(t-1))(a\,n(t-2)+ b\,n(t-3))] =$$

$$= a^2E[n(t)n(t-2)]+abE[n(t)n(t-3)]+abE[n(t-1)n(t-2)]+b^2E[n(t-1)n(t-3)]=$$

$$= 0 + 0 + 0 + 0 = \mathbf{0} \qquad (6.5)$$

Das equações (6.3), (6.4) e (6.5), podemos concluir que o ruído na saída do filtro, é um ruído de correlação igual à ordem do filtro, já que a função de auto correlação R(m) = $\{a^2\sigma^2+b^2\sigma^2;\ ab\sigma^2;\ 0;\ 0;...\}$ anula-se a partir do segundo termo e a ordem do filtro MA(2) é igual a 2.

O filtro AR por ser autorregresivo, isto é uma combinação linear ponderada das saídas passadas somadas à entrada atual, é um ***correlacionador*** de ***todas as amostras do sinal desde o começo da filtragem***.

Desenvolveremos a seguir um termo genérico da função de autocorrelação R(m) do sinal ruidoso $y_n(t)$ do filtro AR(1) (equação (6.2)):

$$E[y_n(t)y_n(t-m)] = E[\sum_{i=0}^{t} a^i n(t-i) \sum_{j=0}^{t-m} a^j n(t-j)] =$$

$$= (\text{se } E[n(i)n(j)] = 0;\ \forall\ i \neq j)$$

$$= \sum_{i=0}^{t-m} a^{2i} E[n(t-i)n(t-i)] =$$

$$= \sum_{i=0}^{t-m} a^{2i}\sigma^2 \qquad (6.6)$$

Onde na equação (6.6) usamos a propriedade do ruído branco de ser nula a correlação cruzada e a propriedade de linearidade do operador valor esperado.

Da equação (6.6) podemos concluir que pela natureza recursiva do filtro AR, a função de auto correlação não se anula para nenhum termo. Então o filtro AR se comporta como um correlacionador em toda ordem. Porém a função de autocorrelação é decrescente pelo fato do coeficiente a serem menor do que 1.

Para a eleição do tipo de filtros devemos considerar a "coloração" do ruído branco e devemos considerar também o custo computacional. Para uma especificação de filtro passa faixa a ordem do filtro MA é maior que o filtro AR, com o qual o ***custo computacional do filtro MA é maior***. Levando em conta as conclusões obtidas e dando maior importância ao custo computacional, ***se escolheram os filtros AR***.

Com o objetivo de utilizar a menor ordem possível dos filtros realizou-se um estudo experimental de discriminação do espectro de uma vogal /a/. Selecionou-se uma parte de uma vogal /a/ pronunciada por um falante masculino e se comparou o espectro de freqüências extraído com a FFT e com sua interpolação por um modelo AR como o fornecido pelas saídas dos filtros.

Da comparação entre os gráficos obtidos dos filtros de ordem 2 e os de ordem 4 concluiu-se que, com os filtros de ordem 2 não obtem-se uma boa resolução dos dois formantes da vogal. Com os filtros de ordem 4 a resolução é melhor, podendo-se distinguir os dois formantes. De acordo com estas conclusões, adotaram-se os filtros AR de ordem 4.

Em quanto ao **critério de desenho** dos filtros existem várias alternativas. A ferramenta utilizada para o desenho dos filtros foi a função *"fdesign"* do programa matemático Matlab 3.5f. No processo de desenho dispoem-se de variáveis que se podem especificar a priori deixando liberdade ao desenhista. Das possibilidades extraíram-se duas alternativas: uma é a igualdade da ganância por filtro e outra é a igualdade de potência passante por filtro. A igualdade de ganância por filtro implica que a ganância como relação entrada saída de todos os filtros fosse a mesma. A igualdade de potência passante por filtro implica que a área sob cada filtro seja semelhante. Entre os dois escolheu-se o critério de igual potência porque se adapta melhor com as seguintes etapas do sistema de extração de características do banco de filtros.

No próximo gráfico apresenta-se um diagrama da resposta em freqüência, em escala logarítmica de magnitude, do banco de filtros final.

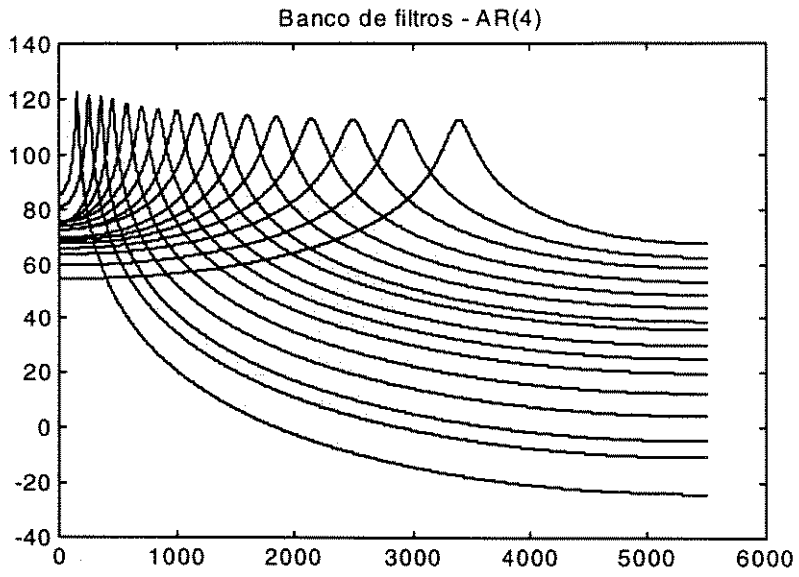


FIGURA 6.1: Resposta em freqüência do Banco de Filtros

6.3.3 Estimação de energias

O seguinte problema é que a saída dos filtros é um sinal temporal contínuo, isto é um sinal da freqüência central de cada filtro. Pelo tipo de modelo usado nos filtros de Kalman para estimar uma constante, deve-se transformar este sinal contínuo numa característica que não mude com o tempo. Naturalmente surge como eleição a energia do sinal de saída de cada filtro, já que está diretamente relacionado com a energia em cada faixa de freqüência. Para extrair esta energia utilizou-se um **operador quadrático** como uma **estimação instantânea da energia**, isto é elevou-se ao quadrado cada amostra do sinal de saída do filtro.

Demonstraremos a seguir que esta é uma **estimação polarizada** da verdadeira energia do sinal. Se $(x(t)+n(t))^2$ é o sinal observado tempo t onde x(t) é o sinal original somado de ruído branco gaussiano n(t) de valor médio nulo, e variânça $E[n(t)^2] = \sigma^2$, estando ambos descorrelacionados (isto é $E[x(t)n(t)] = 0$). A estimação da energia do sinal observada é uma estimação polarizada da energia do sinal original:

$$\begin{aligned} E[(x(t)+n(t))^2] &= E[x(t)^2+2x(t)n(t)+n(t)^2] = \\ &= E[x(t)^2]+2E[x(t)n(t)] +E[n(t)^2] = \\ &= E[x(t)^2] + E[n(t)^2] = E[x(t)^2] + \sigma^2 \end{aligned} \tag{6.7}$$

Faremos agora a mesma demonstração para o mesmo sinal anterior na saída do filtro MA(2). Escolhemos este filtro pela simplicidade da demonstração, porém a demonstração é válida para qualquer ordem de filtros MA e para os filtros AR também. De acordo com a equação (6.1):

$$
\begin{aligned}
E[y(t)^2] &= E[(a\,x(t)+ b\,x(t-1)+a\,n(t)+ b\,n(t-1))\,(a\,x(t)+ b\,x(t-1)+a\,n(t)+ b\,n(t-1))] \\
&= E[\{a^2x(t)x(t)+ b^2x(t-1)x(t-1)+2abx(t)x(t-1)\} + \\
&\quad + \{a^2x(t)n(t)+ b^2x(t-1)n(t-1)+abx(t)n(t-1) +abn(t)x(t-1)\}+ \\
&\quad + \{a^2n(t)n(t)+ b^2n(t-1)n(t-1)+2abn(t)n(t-1)\}] = \\
&= \{E[a^2x(t)x(t)+ b^2x(t-1)x(t-1)+2abx(t)x(t-1)]\} + \\
&\quad + \{a^2E[x(t)n(t)]+ b^2E[x(t-1)n(t-1)]+abE[x(t)n(t-1)] +abE[n(t)x(t-1)]\}+ \\
&\quad + \{a^2E[n(t)n(t)]+ b^2E[n(t-1)n(t-1)]+2abE[n(t)n(t-1)]\} = \\
&= \{E[a^2x(t)x(t)+ b^2x(t-1)x(t-1)+2abx(t)x(t-1)]\} + \\
&\quad + \{0 + 0 + 0\}+ \{a^2\sigma^2+ b^2\sigma^2+0\} = \\
&= \{E[a^2x(t)x(t)+ b^2x(t-1)x(t-1)+2abx(t)x(t-1)]\} + a^2\sigma^2+ b^2\sigma^2 \qquad (6.8)
\end{aligned}
$$

Portanto, de acordo com a equação (6.8) vemos que é uma estimação polarizada, já que na saída do filtro o valor esperado do quadrado da soma do sinal original somado do ruído, não é igual ao valor esperado do quadrado do sinal original. Então para o caso de ruído branco gaussiano de variânça constante, o sesgo da estimação da potência será uma constante para cada sinal de saída dos filtros e esta constante depende do valor da variânça do ruído. Quanto maior seja a variânça maior será a disparidade entre os valores a estimar e os verdadeiros valores. Quanto maior seja a ordem do filtro MA maior será a disparidade entre os valores a estimar e os verdadeiros valores.

A ***disparidade*** entre os ***valores de energia a estimar*** e os ***verdadeiros valores*** de energia terão uma ***conseqüência direta na classificação*** efetuada pela ***rede neural***. A uma dada distribuição de energias as saídas dos filtros terão a uma constante somada dependendo do nível do ruído e dos coeficientes do filtro. Isto provocará um escorregamento sua representação no espaço de estados, podendo cruzar a fronteira da classe correta e cair em outra classe. Isto induziria uma incorreta classificação. Este é o efeito de *desemparelhamento de condição* (condition mismatch), mencionado na seção 4.5.2.

6.3.4 Equações do Filtro de Kalman e modelo de estado

Este sinal da energia é a variável 'ruidosa' sobre a qual o filtro de Kalman da primeira camada extrai uma estimação do valor médio da energia da saída de cada filtro.

O modelo de estado onde se aplicam os filtros de Kalman da primeira camada, é o modelo de uma constante somada de ruído branco, onde o ***estado representa o verdadeiro sinal de energia e o estado medido é o sinal de energia atual de cada filtro.*** Portanto:

$$\text{Equação de Processo: } X(t) = X(t-1) \tag{6.9}$$

$$\text{Equação de Medição: } \tilde{X}(t) = X(t) + w(t) \tag{6.10}$$

Os filtros de Kalman da segunda camada são para estimar os verdadeiros valores nas saídas das neurônios da primeira camada.

6.3.5 Aspectos de implementação

Como alternativa ao método proposto para o cálculo da energia, detectou-se a referência [43], onde propõe-se um estimador da energia baseado em sua definição física como a energia necessária para gerar uma determinada freqüência. A derivação do algoritmo baseia-se numa aproximação válida para freqüências menores de 1/8 da freqüência de Nyquist. Este se denomina o ***algoritmo de Teager,*** e o apresentamos na seguinte equação, para um sinal do tipo $x(n) = A\cos(\Omega m + \phi)$:

$$A^2\Omega^2 \cong x(n)^2 - x(n-1)x(n+1) \tag{6.11}$$

A velocidade de convergência do algoritmo é muito boa e foi verificada em simulações replicando os resultados do trabalho. Porém como demostrou o autor do trabalho, o algoritmo é polarizado em presença de ruído branco. Alem disso, de acordo com as provas realizadas, por sua excelente capacidade de tracking se torna mais sensível ao ruído, já que o considera uma modificação da energia do sinal. Por isto desestimóu-se seu uso.

O verdadeiro modelo do filtro de Kalman requer a inversão de uma matriz de 16x16 para cada amostra do sinal de voz. Esta implementação é de um grande custo computacional, por tanto se impõe a utilização do modelo escalar do filtro de Kalman, que foi descrito na seção 3.4. A aproximação utilizada descarta a correlação entre as saídas dos diferentes filtros e os trata como processos independentes.

A energia dos filtros tem uma grande faixa dinâmica o qual é inconveniente para ser utilizado como entrada para a rede neural. Aplicou-se então uma função logaritmo com objetivo de transformar a faixa dinâmica em outra mais compacta, a qual é uma representação mais adequada de entrada à rede neural.

6.3.6 Esquema completo da arquitetura de banco de filtros

No seguinte gráfico apresenta-se em definitiva a arquitetura completa da arquitetura de banco de filtros.

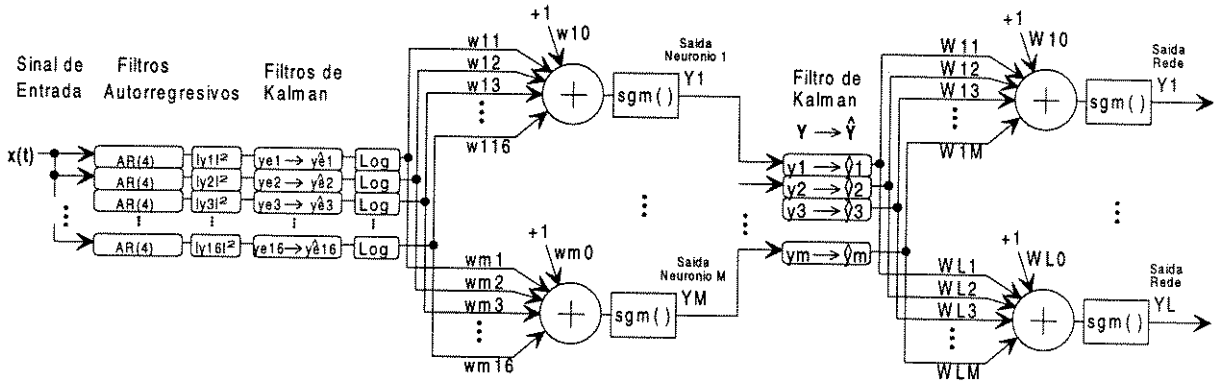


FIGURA 6.2: Arquitetura de Banco de Filtros

Filtragem AR(4): $yi(t) = [x(t)\ x(t-1)\ x(t-2)\ x(t-3)]\ [ai(t)\ ai(t-1)\ ai(t-2)\ ai(t-3)]'$; $i=1,...,16$

Operador Quadrado: $yci(t) = yi(t)^2$; $i=1,...,16$

Filtros Kalman (escalares): $yei(t) = yei(t-1) + K1i\ (yci(t) - yei(t-1))$; $i=1,...,16$

Operador Log: $yli(t) = \log(yei(t))$; $i=1,...,16$

Rede Neural Camada 1: $Yj(t) = [1\ yl1(t)\ yl2(t)\ ...\ yl16(t)]\ [wj0\ wj1\ ...\ wj16]'$; $j=1,...,M$

Filtros Kalman (escalares): $Yej(t+1) = Yej(t-1) + K2i\ (Yj(t) - Yej(t-1))$; $j=1,...,M$

Rede Neural Camada 2:

$$Zj(t) = [1\ Yl1(t)\ Yl2(t)\ ...\ YlM(t)]\ [Wj0\ Wj1\ ...\ WjM]'; \quad j=1,...,L$$

Vetor Saída: $\mathbf{Z} = [Z1(t)\ Z2(t)\ ...\ ZL(t)]$;

Neste caso os filtros de Kalman da primeira camada estão incorporados ao esquema de pré-processamento implementado pelo banco de filtros. Esta é uma das vantagens de colocar os filtros de Kalman antes da ponderação do vetor de entrada pelos pesos sinápticos. Os filtros de Kalman da segunda camada realizam uma estimação sobre as

saídas ruidosas dos neurônios da primeira camada, tal como foi descrito em sua apresentação original.

6.3.7 Análise crítico

É importante observar que este modelo de banco de filtros é uma aproximação do processo real por várias razões:

O sinal de saída dos filtros, onde é aplicado o filtro de Kalman é do tipo faixa estreita, por tanto o modelo de ruído branco, de faixa larga por definição, é inapropriado. Se utilizaramos filtros MA poderia-se modelar a "coloração" do ruído, já que se dispõe dos coeficientes do filtro. Porém isto incrementaria sensivelmente o custo computacional já que os filtros de Kalman deveriam ser vetoriais no sentido temporal (ao contrário de espacial como se vem propondo), isto implicaría uma inversão de matrizes da ordem dos filtros para cada amostra temporal.

O operador quadrático fornece uma estimação polarizada da energia instantânea do sinal na saída do filtro. A polarização é tanto maior quanto maior é a potência do ruído. Além disso como a potência do ruído é desconhecida não é possivél corrigir este problema.

É um fato conhecido que a extração de características por banco de filtros não é uma das melhores para o reconhecimento de voz. Existem outras técnicas de extração de características com melhores resultados, especificamente a predição linear, a qual forma parte da segunda arquitetura proposta.

6.4 ARQUITETURA DE PREDIÇÃO LINEAR

6.4.1 Idéia da proposta

Uma segunda arquitetura é proposta com o objetivo de melhorar os problemas encontrados na primeira arquitetura. Esta segunda arquitetura baseia-se na flexibilidade da modelagem por espaço de estados o que permite utilizar o método de predição linear, que é superior à representação espectral do banco de filtros para o reconhecimento de voz. Daremos uma justificação formal desta superioridade no item 6.5.

O modelo de estado do primeiro filtro de Kalman está baseado numa modelagem por predição linear do sinal de voz. Os coeficientes autorregresivos da predição linear são estimados pelo filtro de Kalman da primeira camada, a fim de de minimizar o erro de

predição da amostra atual. Estes coeficientes estimados são alimentados na primeira camada de neurônios da rede neural para sua classificação.

Esta arquitetura satisfaz os requerimentos acerca da conservação da escala temporal já que para cada nova amostra recalculam-se os coeficientes de predição. Esta também é uma característica que deveria ter pouca variação para fonemas muito redundantes, como por exemplo as vogais.

6.4.2 Equações do KF e modelo de estado

O modelo de estado para este sistema é:

$$\text{Equação de Processo: } X(t) = X(t-1) \tag{6.12}$$

$$\text{Equação de Medição: } u(t) = U(t)^T X(t) + w(t) \tag{6.13}$$

$$\text{onde: } \begin{matrix} X(t)^T = [a_0\ a_1\ a_2\ \ldots\ a_{N-1}] \\ U(t)^T = [u(t-1)\ u(t-2)\ \ldots\ u(t-N-1)] \end{matrix} \tag{6.14}$$

onde o **vetor de estado X** representa os **coeficientes de predição linear e** o **vetor U** representa uma **janela temporal com amostras reais do sinal de voz (sinal + ruído)**, desde t-1 até N-1, onde M é a ordem da predição. ^{T} denota transposição. O ruído w(t) é o ruído branco somado à amostra a estimar no tempo t. O valor atual u(t) é resultado do produto escalar dos coeficientes de predição linear pela janela temporal do sinal somado de ruído.

A equação de processo (6.12) tem por propósito que o estado estimado, os coeficientes de predição linear, não mude com o tempo. Este modelo aproxima uma lenta variação dos coeficientes de predição linear comparado com a escala de tempo do sinal. A equação de medição modela a combinação linear, com os coeficientes do estado, de amostras passadas da série temporal, somado de ruído branco.

Este modelo de estado foi adaptado do que na literatura é conhecido como o modelo do *equalizador adaptativo* [18], [19], no contexto das comunicações. Uma diferença é que naquele, ambos o vetor de entrada e a observação são extraídos da entrada e saída de um canal de comunicações que se deseja equalizar, enquanto que neste último formam parte da mesma série temporal. Outra diferença é que no primeiro o ruído modela o

erro de estimação dos verdadeiros coeficientes de predição, e no segundo o ruído modela o verdadeiro sinal ruidoso que se adiciona à amostra atual.

6.4.3- Estimação de espectros por predição linear.

As vantagens do método de predição linear como estimador de espectros surge a partir de assumir um modelo do sistema gerador. Neste caso, o modelo assumido é um modelo de relações lineares entre amostras sucessivas, o que conduz a um modelo espectral de pólos. A vantagem desta utilização é que a estimação das zonas de alta energia do espectro será muito boa. A maioria dos fonemas humanos caracterizam-se por espectros deste tipo, onde as zonas de alta energia chamam-se "formantes".

O modelo linear adotado está intimamente ligado com a anatomia do sistema fonador humano. Um equivalente simplificado do trato bucal humano é um tubo sem perdas, de M seções cada uma com uma superfície dada segundo o fonema emitido. Desenvolvendo as equações do tubo equivalente, através da teoria das impedâncias acústicas [24] e [31], chega-se a um sistema de equações lineares recorrentes do tipo:

$$u(n) = a1\ u(n-1) + a2\ u(n-2) + a3\ u(n-3) + ... + z(n) \qquad (6.15)$$

onde u(.) é a série temporal gerada pelo sistema com entrada de pulsos glóticos z(n). Como para fonemas vogalizados os pulsos glóticos anulam-se rapidamente, a maior parte do tempo, o sinal u(.) estará definido pelos coeficientes autorregresivos. A equação anterior, sem os pulsos glóticos, se reduz à equação de medição do modelo de estado (6.12) quando o ruído é nulo.

Porém alguns fonemas castelhanos, como por exemplo os nasais /m/, /n/ e /nh/, caracterizam-se por ter antiressonâncias (zeros espectrais) muito pronunciados, produto do acoplamento entre as cavidades nasal e bucal. A estimação destes espectros pelo método da predição linear será defeituosa. Os fonemas de tipo ruidoso, como a /s/ o a /f/, não são gerados com pulsos glóticos pelo que também neste caso haverá problemas na predição.

Outro aspecto da utilização do modelo linear, é que a mesma relação linear entre amostras sucessivas, pode assimilar-se a uma interpolação linear da função de autocorrelação do sinal. Neste sentido a determinação dos coeficientes é robusta frente ao ruído branco já que se extrai de estimações da estatística do sinal.

Como a derivação das ***equações do filtro de Kalman*** realizou-se levando em conta as características estatísticas do ruído branco, a ***determinação dos coeficientes de predição linear é ótima, no caso da perturbaçã ser ruído branco*** . Porém na prática o ruído branco afeta as estimações dos coeficientes de predição linear. Um sinal ruidoso de grande potência, fará que a série temporal seja mais descorrelacionada, o qual faz que para baixas SNR, os coeficientes de predição linear tendam ao vetor unitário.

Este comportamento determinístico de tendência ao vetor unitário limita seriamente o reconhecimento do modelo proposto para baixas SNR. A limitação surge do fenômeno de desemparelhamento de condição, já que os vetores a serem reconhecidos perdem progressivamente as características próprias da classe à qual pertencem. O importante é que o comportamento das características deixa de ser aleatório, com o qual os filtros de Kalman das seguintes camadas da rede neural não podem extrair uma estimação não polarizada do verdadeiro valor das mesmas.

6.4.4 Esquema completo arquitetura de Predição Linear

Apresenta-se a seguir o diagrama com a arquitetura de Predição Linear proposta.

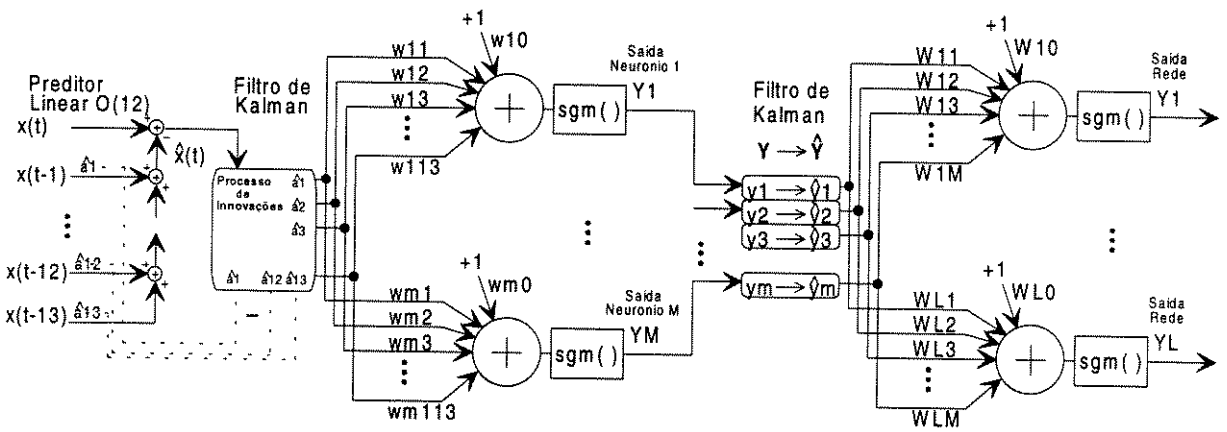


FIGURA 6.3: Arquitetura De Predição Linear

Preditor Linear O(12): $\hat{x}(t) = \sum_{i=1}^{13} \hat{a}i\ x(t-i)) = \hat{A}\ [x(t-1)\ x(t-2)\ ...\ x(t-13)]^T$

Filtro de Kalman: $\hat{A} = \hat{A} + K1\ (x(t) - \sum_{i=1}^{13} \hat{a}i\ x(t-i))$

Rede Neural Camada 1: Yj(t) = [1 â1 â2 ... â12] [wj0 wj1 ... wj12]'; j=1,...,M

Filtros Kalman (escalares): Yej(t+1) = Yej(t-1) + K2i (Yj(t)- Yej(t-1)); j=1,...,M

Rede Neural Camada 2:

Zj(t) = [1 Yl1(t) Yl2(t) ... YlM(t)] [Wj0 Wj1 ... WjM]'; j=1,...,L

Vetor Saída: $\mathbf{Z} = [Z1(t)\ Z2(t)\ ...\ ZL(t)]$;

6.4.5 Arquitetura De Predição Linear Angular.

É um fato conhecido que na medida que a potência do ruído se incrementa, o *sinal resultante* tende a ser descorrelacionado. Logo a *função de autocorrelação* tende ao vetor com um *primeiro termo significativo* e o *resto com valores muito pequenos* portanto os *coeficientes de predição linear tendem à origem.* Isto causa um aumento na confusão de classes e um aumento no erro de reconhecimento a medida que a potência de ruído aumenta.

Num esforço para minimizar este problema uma representação alternativa foi proposta na literatura especializada. Em Mansour, e Juang (1989) [1], demonstrou-se tanto matemática como experimentalmente que na medida que a SNR diminui, hà uma diminuição na magnitude do vetor de coeficientes Cepstrais. Também mostrou experimentalmente que na medida que a SNR diminui, hà uma maior diminuição na magnitude do que no ângulo do vetor de coeficientes Cepstrais. Os teoremas fundamentais dessa referência foram apresentaremos no capítulo 4, seção 4.5.3:

Com o *objetivo* de *verificar se as conclusões anteriores podem ser aplicadas aos coeficientes de predição linear*, se estudará a *evolução deste coeficientes com a presença de ruído branco.* Para isto recordamos que estes são obtidos como solução das equações normais do preditor. Se a função de autocorrelação é denotada por: R = [R(1) R(2) R(3)], onde cada R(i) = E[x(t)x(t-i-1)], então os coeficentes de predição linear A = [A(1) A(2)] são a solução de:

$$\begin{bmatrix} R(1) & R(2) \\ R(2) & R(1) \end{bmatrix} \begin{bmatrix} A(1) \\ A(2) \end{bmatrix} = \begin{bmatrix} R(2) \\ R(3) \end{bmatrix} \tag{6.16}$$

Veremos primeiro experimentalmente como se comporta a função de autocorrelação de um sinal de voz. Apresentamos na próxima figura as sucesivas funções de autocorrelação de um trecho de 20 ms. de uma vogal /a/, multiplicado por uma janela de hamming, quando a relação sinal - ruído diminui de 40dB até 0dB em quatro passos. As funções de autocorrelação são normalizadas, tal que o primeiro termo seja unitário. É importante recordar que a normalização não afeta a solução das equações normais.

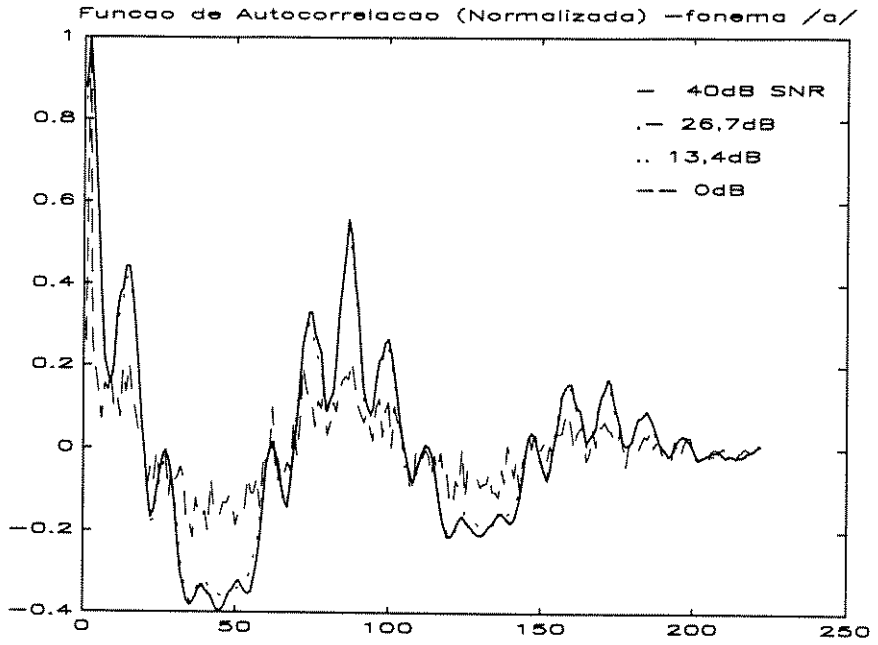


FIGURA 6.4 - Evolução função Autocorrelação com ruído para trecho fonema /a/.

Da figura anterior observamos que na médida que o ruído cresce existe uma tendência gradual da função de autocorrelação à função de autocorrelação do ruído branco, isto é, um primeiro termo significativo e o resto pequeno. Além disso vemos que os termos de maior amplitude são os que mais diminuem. Isto aconteçe por duas causas, por um lado é natural que diminuam por efeito da descorrelação do ruído branco, e por outro lado, o primeiro termo, da energía do sinal mais ruído, cresce muito mais do que os termos cruzados, portanto a normalização também faz que estes diminuam mais do que os termos de pequena amplitude.

Trataremos de modelar a variação observada da função de autocorrelação com o ruído. Como a modificação real da função de autocorrelação é estocástica não é possível encontrar una equação fechada para esta variação. Estudaremos a evolução de uma função de autocorrelação simulada com valores R= [1 -0.82 0.15].

Modelo de minimização proporcional da função de autocorrelação, onde pela multiplicação de um fator menor que um, que representa a fração de diminuição, a todos os componentes com exceção do primeiro termo fixo e igual a um. Este fator diminui com cada passo para simular um aumento do nível de ruído. Com este modelo, os termos de maior amplitude diminuirão mais do que os termos de baixa amplitude, mais todos diminuirão em forma proporcional. Na figura 6.5 se apresenta a evolução da função de autocorrelação proposta, com um fator (1-0.005*p) onde p=1:30, representa os 30 passos.

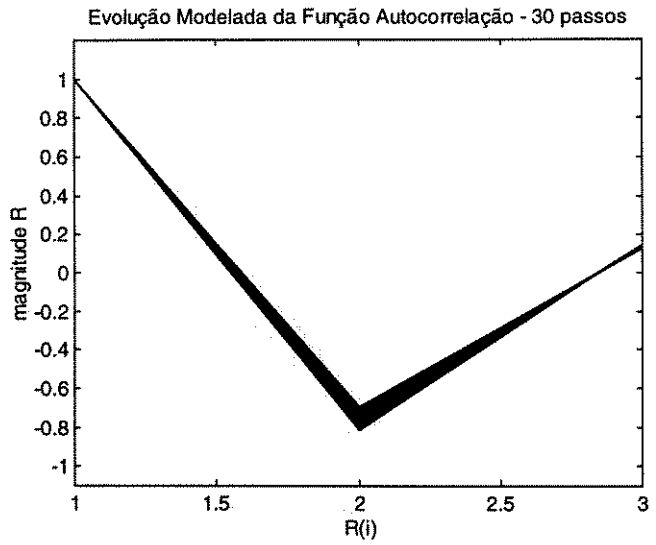


FIGURA 6.5 - Evolução função Autocorrelação de acordo ao modelo proporcional.

Na figura 6.6 se apresenta a evolução do vetor de coeficientes de predição linear A, como solução das sucesivas equações normais obtidas das funções de autocorrelação apresentadas na figura 6.5.

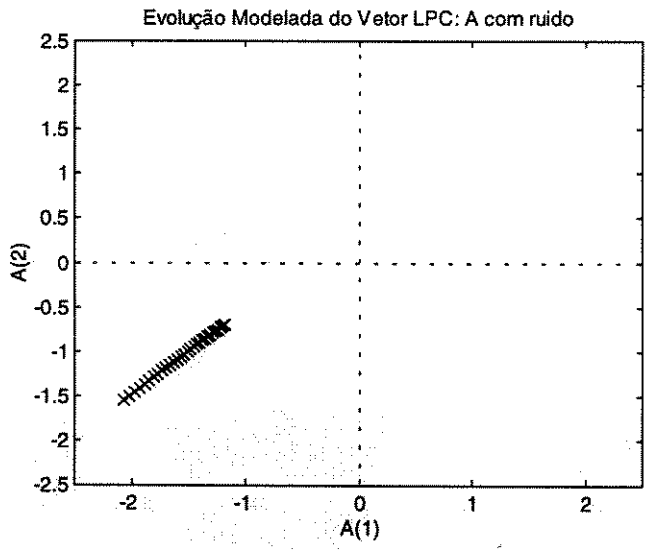


FIGURA 6.6 - Evolução do vetor de coeficientes de predição linear (30 passos).

Observamos na figura 6.6, que a evolução dos vetores de coeficientes de predição linear com o aumento do ruído, não é linear respeito da origem, pero a modificação é mas importante na magnitude do vetor que no ângulo, coincidentemente com o observado experimentalemnte por [1], para os coeficientes cepstrais.

Observamos também que a evolução é grande para uma pequena modificação da função de autocorrelação como a simulada. Isto demonstra uma grande sensibilidade dos parámetros de predição linear com a modificação da função de autocorrelação. Esta grande sensibilidade existe porque a equação de sensibilidade da solução de uma equação linear AX=B, é uma equação não linear: $\Delta X=\Delta B/\Delta A$. Pequenas modificações de ΔA afetam muito a ΔX.

Para tratar de minimizar a variabilidade encontrada *extrairemos* uma *representação angular* do vetor de predição linear normalizando pelo módulo. Apresentaremos a seguir uma figura semelhante à anterior onde os vetores são extraídos da forma indicada; notar que as margens das figuras anterior e seguinte são iguais.

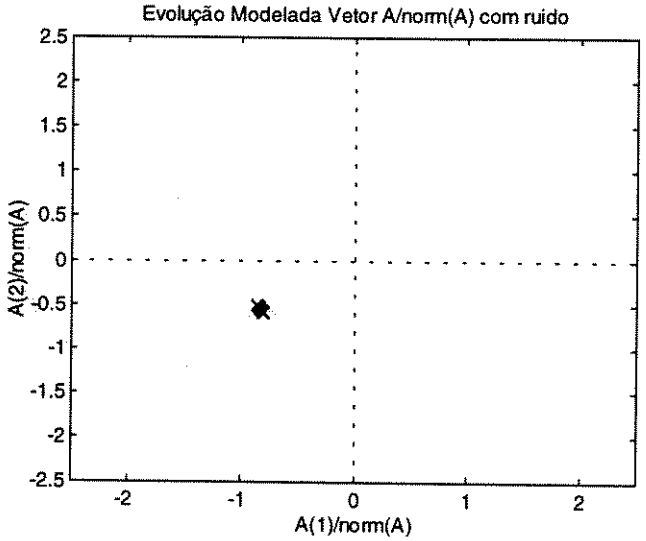


FIGURA 6.7 - Evolução vetor normalizado A/norm(A) (30 passos).

Observamos, comparando as figuras 6.6 e 6.7, que a grande variabilidade dos vetores originais foi reduzida graças a representação angular, que aproveita a maior variabilidade do módulo com respeito ao ângulo. O espaço dos novos coeficientes é o círculo unitário pela normalização.

Devemos notar que a normalização dos vetores LPC, implica vetores com termos muito pequenos e isto traz um problema para o aprendizado das redes neurais. Para aliviar este problema, os vetores foram multiplicados por um fator constante e igual a cinco. Pelo fato de serem conservados os ângulos, esta multiplicação não modifica o problema do reconhecimento.

Portanto utilizaremos este resultado para *modificar a representação do vetor de entrada na rede neural* tal que seja invariante à modificações do módulo do vetor e só dependa do seu ângulo. Esta representação foi chamada representação ***De Predição Linear Angular***. A idéia é apresentar à rede características que, como se normaliza pelo módulo, sejam invariantes a modificações da magnitude, e como se mede o ângulo representa a direção do vetor.

Apresentamos a seguir o diagrama da estrutura proposta.

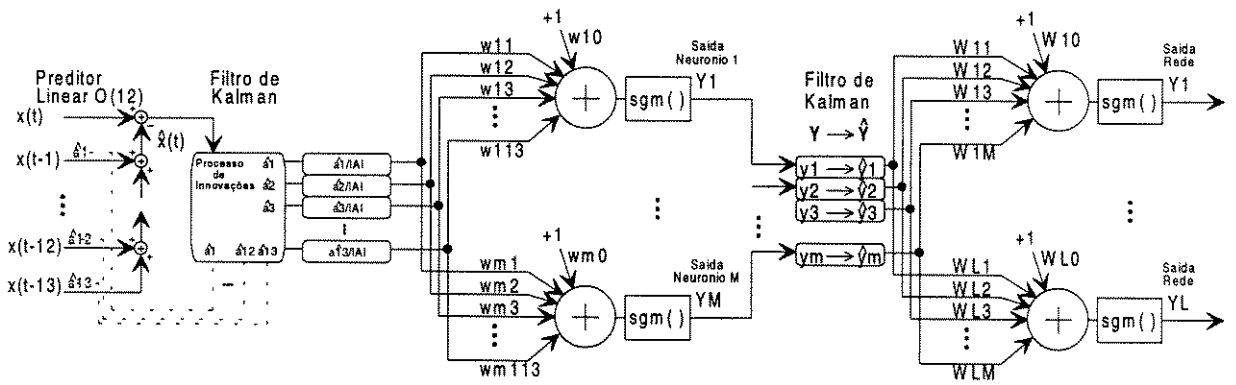


FIGURA 6.8: Arquitetura De Predição Linear Angular

Preditor Linear O(12): $\hat{x}(t) = \sum_{i=1}^{13} \hat{a}i\ x(t-i))$

Filtro de Kalman: $\hat{A} = \hat{A} + K1\ (x(t) - \sum_{i=1}^{13} \hat{a}i\ x(t-i))$

Rede Neural Camada 1:

$$Yj(t) = [1\ \hat{a}1/|A|\ \hat{a}2/|A|\ ...\ \hat{a}13/|A|] * [wj0\ wj1\ ...\ wj13]';\ j=1,...,M$$

Filtros Kalman (escalares): $Yej(t+1) = Yej(t-1) + K2i\ (Yj(t) - Yej(t-1));\ j=1,...,M$

Rede Neural Camada 2:

$$Zj(t) = [1\ Yl1(t)\ Yl2(t)\ ...\ YlM(t)]\ [Wj0\ Wj1\ ...\ WjM]';\ j=1,...,L$$

Vetor Saída: $\mathbf{Z} = [Z1(t)\ Z2(t)\ ...\ ZL(t)];$

6.5- Comparação entre arquiteturas

Existem várias vantagens desta arquitetura de predição linear respeito da arquitetura de banco de filtros. O **modelo do ruído é correto** já que como os coeficientes de predição linear são extraídos diretamente do sinal temporal de voz o ruído é efetivamente de faixa larga. Ao contrário da arquitetura anterior onde o ruído era limitado em freqüência por ter passado por um filtro passa faixa.

Não existe um operador não linear que polarize a estimação, ao contrário da arquitetura anterior onde o operador quadrático impedia que o filtro de Kalman pudesse estimar o verdadeiro valor da energia.

Compararemos ambas arquiteturas com respeito ao *custo computacional.* A arquitetura de banco de filtros requer o computo de 16 filtros de Kalman escalares por amostra. A arquitetura de Predição Linear requer o computo de um filtro de Kalman N-matricial por amostra. Mas graças à modelagem de estado utilizada, ***a matriz de correlação do processo de inovações é um escalar***, desta forma ***não há inversão de matriz*** no cômputo do filtro de Kalman, o que alivia seu custo computacional.

Em ambas arquiteturas procura-se extrair uma representação espectral do sinal de voz para sua classificação pela rede neural. *Compararemos* então *as duas representações espectrais* em quanto a sus características *como estimadores do verdadeiro espectro de potência.*

O banco de filtros realiza uma **quantificação fixa** em freqüência, já que trata de aproximar o verdadeiro espectro de freqüências por um conjunto de filtros passa faixa fixos, e alem do mais cada filtro tem um Q-fixo [42]. O problema manifesta-se quando os padrões tem informação relevante na orla entre faixas, com o que pode aparecer numa faixa no treinamento e em outra faixa no reconhecimento. Outro problema é quando pouca energia está presente numa faixa já seja para treinamento ou reconhecimento. Neste caso a energia medida é de grande variabilidade.

Pelo contrário o modelo de predição linear realiza uma **quantificação adaptativa** do espectro já que aproxima o verdadeiro espectro com M pólos de Q variável distribuídos nos picos do espectro, já que os pólos são calculados a partir dos dados. Neste caso, a diferença entre os espectros de treinamento e reconhecimento são manifestados como diferenças proporcionais nos pólos. Alem disso como a concentração dos pólos é nas zonas de alta energia, a variação das zonas de baixa energia é ignorada.

Ambas estão *baseadas* em ***modelos biológicos***. A arquitetura de Banco de Filtros está baseada no modelo simplificado do sistema auditivo humano como um banco de filtros passa faixa. Alem disso a distribuição em freqüência do banco de filtros está baseada numa escala perceptual como é a escala MEL. A arquitetura de Predição Linear está baseada no modelo do sistema fonador humano como um tubo de M seções. Mesmo que o modelo do tubo seja válido para todos os fonemas emitidos, a suposição sobre o sinal de geração é válida para um conjunto limitado de fonemas.

6.6 RECONHECIMENTO RUIDOSO DE VOGAIS

6.6.1 Especificações

A tarefa de reconhecimento sob estudo, é um problema de reconhecimento fonético dependente do falante contaminado por ruído branco gaussiano a diversas SNR. Os fonemas consistiram das cinco vogais espanholas.

O problema encarado é um problema de reconhecimento ruidoso de voz onde o ruído branco foi gerado matematicamente a posteriori da digitalização. Este é um fator importante a considerar com respeito ao alcance dos resultados. Um problema real de reconhecimento ruidoso implica uma série de fatores que não se consideraram. Por um lado é conhecido o efeito Lombard, onde modifica-se a emissão do sinal de voz quando o falante escuta ruídos de alta potência. Pelo outro lado o ruído deixa de ser branco, já que este é um ruído com propriedades matematicamente muito bem definidas e portanto ideal, também a relação sinal - ruído deixa de ser constante pela variação contínua da fonte ruidosa.

6.6.2 Base de dados, características

Separamos a base de dados em duas independentes, uma de treinamento e outra de reconhecimento. A base de dados de treinamento contém dez emissões das cinco vogais espanholas produzidas por dois falantes, um masculino e outro feminino, e a de reconhecimento contém quatro emissões das mesmas vogais emitidas pelos mesmos falantes.

Os fonemas foram registrados num ambiente de baixo ruído. A freqüência de amostragem foi de 8 kHz e o nível de quantificação foi de 8 bits. Mesmo tendo equipamento de 16 bits e maiores taxas de amostragem, a eleição realizada foi propositada. O objetivo é operar com sinais de voz não somente degradados por sinais aleatórios, mas também por distorções de quantificação e baixas taxas de amostragem.

6.6.3 Evolução do treinamento

A topología da rede neural da arquitetura de Banco de Filtros é de 17-30-5. Onde a dimensão da camada de neurônios de entrada se corresponde com a dimensão do vetor de energias somando um para o bias, a dimensão da camada intermediária é de 30 e a dimensão da camada de saída é de 5 correspondente a cada uma das cinco classes. A da arquitetura de Predição Linear é de 13-30-5.

Para o ***treinamento*** utilizaram-se ***exclusivamente amostras de voz sem ruído***. O objetivo é avaliar as arquiteturas propostas em condições de desemparelhamento de condição.

O treinamento consistiu primeiro numa transformação da base de dados de treinamento em tanto uma representação por banco de filtros Mel como numa representação de Predição Linear. Esta etapa pode ser visualizada como a criação de novas bases de dados com as características. Logo as redes neurais foram treinadas com o algoritmo "backpropagation" sobre as respectivas bases de dados de características anteriormente descritas. O treinamento finalizou quando logrou-se um erro especificado mínimo ou o erro não mudou logo de um número especificado de iterações.

Nesta etapa apareceram as primeiras ***diferenças entre as arquiteturas***. Já que a base de dados de voz era única, a evolução do respectivo treinamento revela como cada arquitetura realizou a discriminação entre os fonemas. O tempo de treinamento da arquitetura de banco de filtros MEL foi maior que o da arquitetura de Predição Linear. Alem disso finalizou pela condição de invariância do erro, num valor de erro alto, ao contrário da arquitetura de Predição Linear que finalizou pela condição de erro mínimo.

6.6.4- Definição de índices de reconhecimento

Para poder comparar diferentes arquiteturas de reconhecimento de voz, o mesmo índice de reconhecimento deve ser usado. Neste sentido devemos adaptar as particulares características do modelo proposto, isto é uma classificação por cada amostra temporal do sinal de voz, aos índices comummente utilizados na comunidade de reconhecimento de voz.

Dois índices diferentes foram propostos para revelar diferentes aspectos do reconhecimento ruidoso realizado por cada modelo. Um é o ***Índice Global***, que atribui um padrão de entrada à classe que tem o máximo da soma de cada saída individual durante a duração total do fonema. O outro é o ***Índice de Quadro*** que atribui cada quadro de 25 ms. à classe que tem o máximo da soma de cada saída individual durante a duração do quadro. Esta duração do quadro é uma medida estandarizada da duração mínima da voz. Não se utilizou solapamento entre quadros. Para deduzir finalmente este índice computam-se a quantidade de quadros corretamente atribuidos com relação ao total de quadros do fonema.

O índice global é proposto com o objetivo de avaliar o reconhecimento global do fonema ao longo de toda sua duração. Está inspirado na função verosimilitude acumulada dos logaritmos das probabilidades de reconhecimento, no contexto do algoritmo de viterbi

aplicado aos HMM. O índice de quadro é proposto com o objetivo de avaliar a evolução do reconhecimento em quadros ao longo do fonema. Tipicamente este índice indicará mais erros no começo do fonema onde os filtros de Kalman ainda não estão estabilizados, e no final do fonema, onde ocorre o decaimento do fonema, e não no centro, onde ocorre a parte estável do fonema.

6.6.5 Resultados de reconhecimento

Nas seguintes tabelas 6.2 e 6.3 apresentam-se os resultados de reconhecimento ruidoso para diversas arquiteturas e várias SNR. Na tabela 6.2 apresenta-se os resultados do índice global e na tabela 6.3 apresenta-se os resultados do índice de quadros. Cada arquitetura é avaliada usando primeiro uma rede neural com filtros de Kalman na primeira camada unicamente, e logo uma rede neural com filtros de Kalman em ambas camadas, isto é, a rede neural de reconhecimento adaptativo. Esta distinção realizou-se para estudar a contribuição do filtro de Kalman da camada intermediária.

Tipo de Arquitetura \ SNR	40dB	30dB	20dB	10dB
Banco de Filtros	85	80	55	25
Banco de Filtros Rede de Reconhecimento Adaptativo	85	80	60	35
De Predição Linear	95	95	70	30
De Predição Linear Rede de Reconhecimento Adaptativo	95	95	70	30
De Predição Linear Angular	95	95	80	60
De Predição Linear Angular Rede de Reconhecimento Adaptativo	100	95	85	55

TABELA 6.2: **Índice Global** de Reconhecimento Ruidoso (em %)

Tipo de Arquitetura \ SNR	40dB	30dB	20dB	10dB
Banco de Filtros	72.8	67.9	52.9	26.3
Banco de Filtros Rede de Reconhecimento Adaptativo	82.3	78.0	57.8	25.6
De Predição Linear	92.0	89.5	66.1	26.1
De Predição Linear Rede de Reconhecimento Adaptativo	90.0	88.0	64.6	27.6
De Predição Linear Angular	86.8	84.4	73.3	55.8
De Predição Linear Angular Rede de Reconhecimento Adaptativo	91.6	87.4	76.6	53.5

TABELA 6.3: **Índice de Quadro** de Reconhecimento Ruidoso (em %)

As conclusões das tabelas 6.2 e 6.3 são:

• A arquitetura ***de Predição Linear Angular é a de melhor comportamento global***. Para altas SNR tem um comportamento similar à arquitetura de Predição Linear para ambos índices. O que permite concluir que para baixo ruído as representações angulares e por módulo são semelhantes, já que dão níveis similares de discriminação. Para baixas SNR tem claramente o melhor índice de reconhecimento de todas as arquiteturas propostas, já que incorpora uma representação que é mais robusta às modificações dos coeficientes de predição linear com baixo ruído. É notável que o mecanismo de robustecimento da arquitetura de Predição Linear proposto comece a atuar em condições de ruído crescente, já que para 40 e 30 dB SNR ambas tem aproximadamente os mesmos resultados, e para 20 e 10 dB SNR estes divergem sensivelmente. A representação angular de Predição Linear apresenta um aumento consistente nos resultados de reconhecimento tanto nos índices global como no de quadro, o que revela que esta combinação foi a melhor entre todas as arquiteturas e variações apresentadas.

• Estes resultados revelam que a ***arquitetura de Predição Linear apresenta maiores índices de reconhecimento que a arquitetura de banco de filtros***. A maior diferença entre ambas arquiteturas se da em condições de altas SNR, onde a discriminação de padrões limpos é mais importante do que a robustez ao ruído. Porém ambas arquiteturas logram aproximadamente os mesmos resultados a baixas SNR (≈20% = 1/5 como o número de classes é 5, o índice é equiprovável). Para este caso existe uma explicação diferente para cada arquitetura. No caso da arquitetura de banco de filtros, as baixas SNR aumentam a confusão de classes com respeito aos padrões limpos, pelo sesgo na estimação da energia dos canais. No caso da arquitetura de Predição Linear a evolução tendendo ao vetor unitário é a causa dos baixos resultados de reconhecimento.

• A incorporação da representação angular na arquiteturas de Predição Linear melhorou seus índices de reconhecimento. Em condições emparelhadas os índices de ambas foram semelhantes, porém a diferença mais importante apareceu em condições desemparelhadas com SNR de 20 e 10 dB. Isto prova a robustez desta representação para condições desemparelhadas.

• A comparação em cada arquitetura com respeito ao ***uso da rede de reconhecimento adaptativo*** permite diferentes interpretações. Na arquitetura de banco de filtros seu uso provoca um pequeno aumento nos resultados de reconhecimento devido a que esta confusão é de tipo ruidoso, os filtros de Kalman da camada intermediária melhoríam as estimações das saídas da primeira camada. Porém esta melhoría não é tão importante já que existe uma grande confusão de classes.

• Os índices de quadro são consistentemente menores que os índices globais para todas as arquiteturas. Isto revela que mesmo que o fonema seja atribuido corretamente à sua classe (Índice Global = 100%), não todos os quadros são corretamente atribuidos (Índice de Quadro < 100%). Para casos de baixo ruído, onde a confusão de quadros é grande, ambos índices se aproximam.

6.6.6 Comparação com outros resultados semelhantes

Apresentou-se na seção 4.5.3, uma tabela extraída do paper de Paliwal K. K. [39] onde se comparam as redes neurais multilayered feed-forward (MLP) frente a vários classificadores clássicos (como Maximum Likelihood e K-Nearest Neighbor) no reconhecimento de vogais inglesas afetadas por ruído branco, em modo independente do falante. Esta tabela se reproduz a seguir:

SNR (dB)	Classificador MLP	Classificador ML	Classificador kNN
Clean	91.3	92.7	88.2
35dB	90.2	91.1	84.2
30dB	88.2	85.6	78.0
25dB	84.0	74.2	69.8
20dB	75.6	64.7	60.7
15dB	58.9	46.7	49.1

TABELA 6.4: Comparação de Classificadores, em % correto, extraídos de [39].

É difícil comparar os resultados da tabela anterior já que foram obtidos para um conjunto de vogais em idioma inglês, o qual implica um grau de confusão diferente, e alem disso foram obtidos em condições de independência do falante. Porém, foi o único resultado semelhante encontrado pelo autor e será comparado a seguir com os resultados obtidos com as arquiteturas propostas, salvando as diferenças mencionadas.

Como estes são índices de reconhecimento totais de fonemas, compararemos esta tabela 6.4 com a tabela 6.2 obtida com os índices globais e com a arquitetura angular de Predição Linear, por ser a melhor desempenho. Para 30 dB SNR, a arquitetura angular de Predição Linear logra uma porcentagem levemente maior que a rede neural com cepstrais (95% contra 88.2%, respectivamente), para 20 dB SNR a diferença é ainda maior chegando quase ao 10% de diferença (85% contra 75.6%, respectivamente), e o resultado final de

reconhecimento é semelhante (55% e 58.9%, respectivamente), só que foi logrado em cada caso com 5 dB de diferença (10dB e 15 dB SNR, respectivamente).

6.7 RECONHECIMENTO RUIDOSO DE PALAVRAS

6.7.1 Especificações

Na próxima seção aplicaremos o modelo proposto de rede neural ao reconhecimento ruidoso de palavras utilizando conjuntamente as redes neurais e os modelos de Markov, tal como foi apresentado na seção 4.4.3.2; isto é, usando a rede neural como estimador de probabilidades a posteriori.

6.7.2 Base de Dados

A base de palavras a reconhecer foi escolhida de comandos verbais, em espanhol, de um braço robô. As palavras da base são:

{Tomar, Soltar, Abajo, Arriba, Izquierda, Derecha, Adelante, Atras}

TABELA 6.5 - Palavras da base de dados

A base de treinamento consiste em sete emissões por palavra, emitidas por um falante masculino e a de reconhecimento consiste em três emissões independentes por palavra, emitidas pelo mesmo falante. Foram gravadas a uma freqüência de amostragem de 11025Hz com oito bits.

6.7.3 Segmentação

Como o modelo proposto só é apto para o reconhecimento fonético, as palavras devem ser separadas segúm uma classificação fonética. Então a base de dados de treinamento foi segmentada manualmente segúm a seguinte transcrição fonética:

Tomar	*/O M A R/*
Soltar	*/S O L sil A R/*
Arriba	*/A R I B A/*
Abajo	*/A B A J O/*
Izquierda	*/I S sil I E R D A/*

Derecha	*/D E R E sil CH A/*
Adelante	*/A D E L A N sil E/*
Atras	*/A sil R A S/*

TABELA 6.6 - Segmentação base de dados

Os Modelos de reconhecimento a partir da transcrição fonética são os seguintes, onde o número entre parentese é o número de estados::

Tomar (4)	*/O M A R/*
Soltar (6)	*/S O L S A R/*
Arriba (5)	*/A R I B A/*
Abajo (5)	*/A B A J O/*
Izquierda (7)	*/I S I E R D A/*
Derecha (7)	*/D E R E S CH A/*
Adelante (8)	*/A D E L A N S E/*
Atras (5)	*/A S R A S/*

TABELA 6.7 - Modelos de Reconhecimento

Duas observações devem ser feitas sobre os modelos de reconhecimento propostos. As plosivas /t/ e /p/ são de uma duração muito curta para a longa memória dos filtros de Kalman, portanto foram excluídas. Os silêncios, prévios as plosivas, são usualmente confundidos com fricativas /s/; ainda mais quando a relação sinal - ruído diminui. Portanto para uma implementação mais robusta dos modelos de palavras os silenços foram substituidos pelas /s/. Também o ditongo foi modelado como vogais separadas.

O resultado da segmentação forneçeu a seguinte quantidade de fonemas para treinamento:

A	*12*7 Emissões*
E	*5*7 Emissões*
I	*2*7 Emissões*
O	*3*7 Emissões*
B	*2*7 Emissões*

D	*3*7 Emissões*
J	*7 Emissões*
CH	*7 Emissões*
L	*2*7 Emissões*
M	*7 Emissões*
N	*7 Emissões*
R	*5*7 Emissões*
S	*2*7 Emissões*

TABELA 6.8 - Quantidade de fonemas da segmentação

Da tabela anterior observa-se que existem muito mais exemplos de alguns fonemas do que outros. Mais se utiliza-se o algoritmo clásico de "backpropagation", atualizando os pesos com cada padrão, existe o risco de que a rede aprenda melhor as classes com mais exemplos do que as outras. Isto acontece porque no começo o erro de treinamento diminui muito pelo aprendizado da classe numerosa. Nos exemplos apresentados no capítulo 2 verifica-se que no algoritmo "backpropagation", a atualização logo da apresentação de um exemplo de cada classe é muito mais rápida do que a atualização a cada padrão, e ainda aprende por igual todas as classes. Porém precisa-se de um número igual de exemplos de treinamento por cada classe. Para isto foram duplicados os exemplos das classes menos númerosas e foram descartados alguns das classes mais numerosas. No treinamento foram utilizados vinte e um exemplos de cada fonema.

6.7.4 Treinamento dos modelos

Para o cálculo das probabilidades de permanência e transicão dos modelos de Markov, foram usadas as formulas apresentadas na seção 4.4.3.2 do capítulo 4. Porém isto não é suficiente para este caso porque como o processamento é feito na escala de tempo do sinal, as probabilidades de permanência são muito próximas a um, não discriminando os estados de gran permanência dos de baixa permanência. Para evitar isto foram elevados todas as probabilidades calculadas a um exponente constante, tal que as probabilidades se afastem de um e permitam uma melhor discriminação. Ista operação não afeta os resultados de reconhecimento. Desta forma os modelos HMM resultaram:

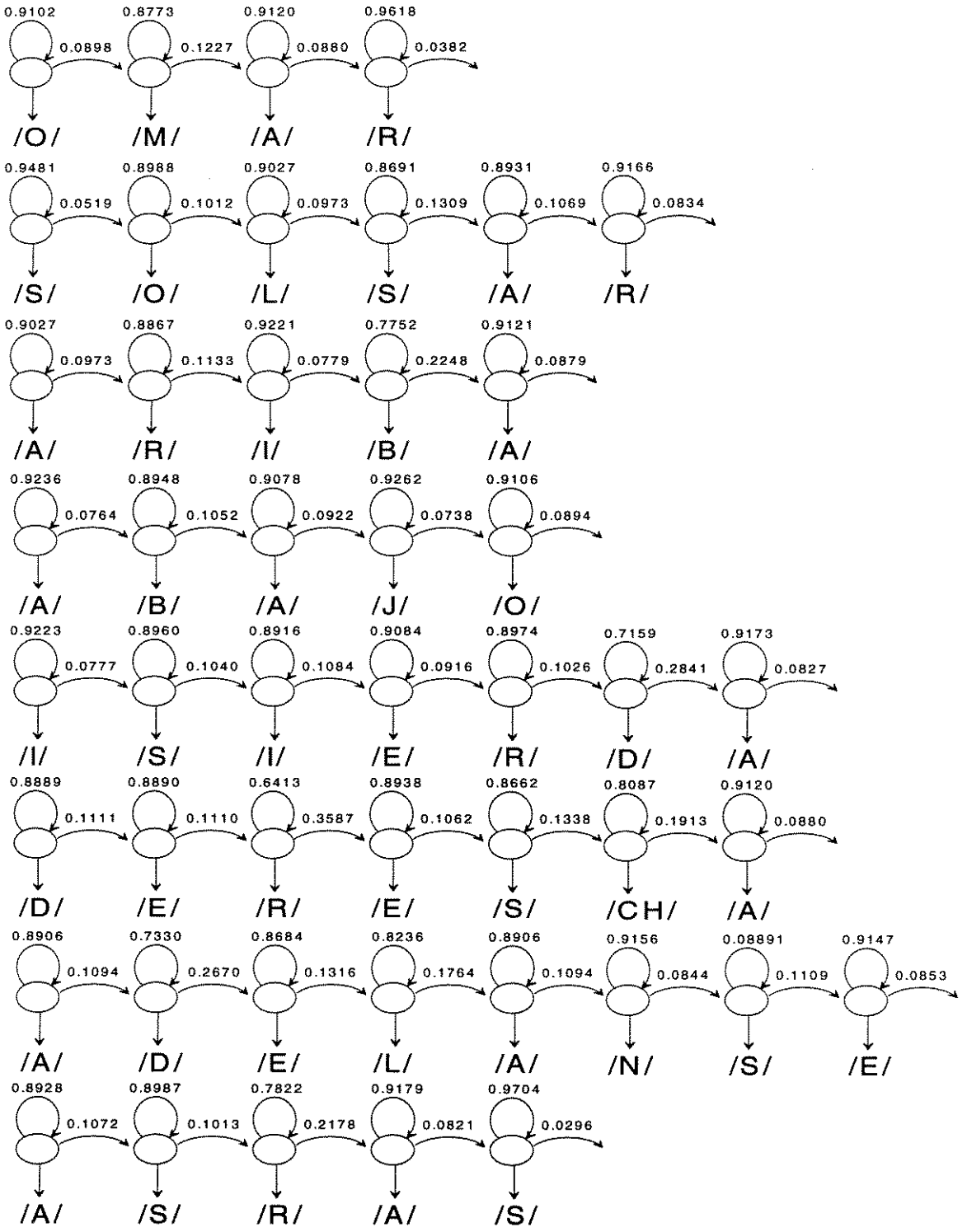


FIGURA 6.9 - Modelos HMM de reconhecimento

A primeira tentativa de rede neural para o reconhecimento fonético, foi uma rede dupla, uma para o reconhecimento de vogais, com cinqüenta neurônios na camada oculta e quatro na camada de saída, e outra para o reconhecimento de consonantes, com cem neurônios na camada oculta e nove na camada de saída. Os primeiros testes demontraram que o reconhecimento da própia base de dados sem ruído era insatisfatorio. A causa disto

era a falta de discriminação entre as duas redes. Portanto foi adotada finalmente uma única rede de *cento e cinqüenta* neurônios na camada oculta e *treze* na camada de saída, para a discriminação de todos os fonemas.

O treinamento descrito levou aproximadamente umas 20 horas, num PC 486DX4-100, com um algoritmo "backpropagation" feito em linguagem C, com coeficientes de aprendizagem de 0.01 e de momentum de 0.05.

6.7.5 Resultados de reconhecimento

Como entre as redes propostas as que tiveram o melhor resultado foram as de Predição Linear, somente elas foram utilizadas nesta seção. Ainda mais só se utilizaram nos primeiros resultados, redes neurais sem os filtros de Kalman na camada intermediária.

Para testar o reconhecimento de palavras com os modelos propostos, cada palavra da base de reconhecimento foi contaminada com uma soma de ruído branco gaussiano. A potência desta foi calculada a partir da relação sinal ruído desejada e a potência total da palavra. Os resultados de reconhecimento se apresentam a seguir para diversas relações sinal - ruído:

Tipo de Arquitetura \ SNR	40dB	30dB	20dB	10dB
De Predição Linear	87.5	79.2	54.2	25
De Predição Linear Angular	83.3	62.5	37.5	20.8

TABELA 6.9: Reconhecimento Ruidoso de palavras com HMM-ANN (em %)

6.7.6- Análise dos resultados

Os resultados da tabela 6.9 aparentemente contradizem os resultados obtidos no reconhecimento de vogais. Mas, para estudar as causas destes novos resultados, devemos encontrar o limite para a aproximação linear da evolução dos coeficientes LPC com o ruído.

Quando mais de 30 passos são usados na modelagem da variabilidade da função de autocorrelação, observamos que variação no ângulo aumenta e no módulo diminui. Isto pode ser observado nas figuras 6.10 e 6.11, onde são usados 100 passos.

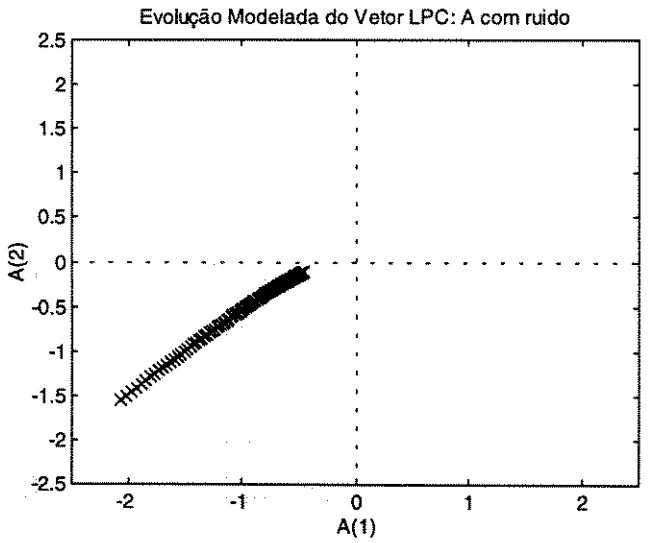


FIGURA 6.10 - Evolução modelada Coeficientes LPC (100 passos).

Na próxima figura observamos os coeficientes de predição linear normalizados obtidos dos anteriores.

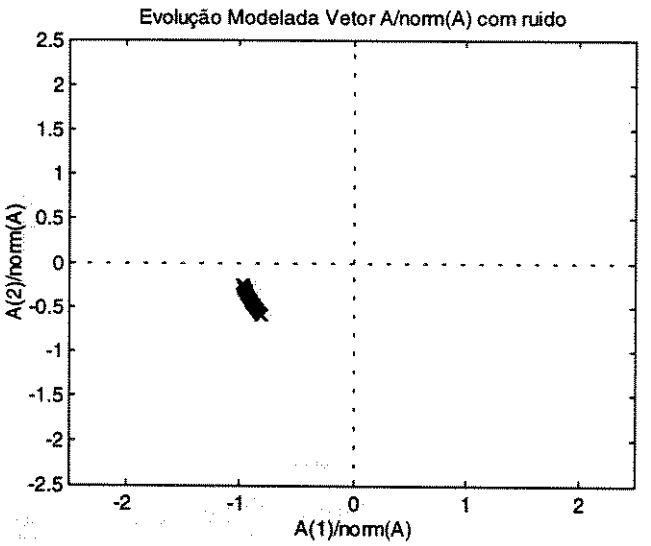


FIGURA 6.11 - Evolução modelada Coeficientes LPC Normalizados (100 passos).

Da figura 6.11 verifica-se a tendência do escorregamento do ângulo do vetor LPC normalizado, quando deixa de ser válida a relação de variabilidades entre o módulo e o ângulo. O círculo unitário é um espaço pequeno para a discriminação de classes, ainda mais se o número de classes é grande. No caso das vogais, onde o número de classes é pequeno (cinco), a variação do ângulo não é tão importante porque o espaço angular disponível é muito maior. No caso dos fonemas, onde o número de classes é grande (treze), o espaço

angular de cada classe é pequeno, portanto a variabilidade angular provoca uma confusão importante de classes. Isto explica a diferença dos resultados obtidos.

6.7.7 Rede Invariante

Para aproveitar a aproximação linear da evolução dos coeficientes LPC com o ruído sem o problema da proyeção sobre o círculo unitário, proporemos um enfoque diferente. O mecanismo de classificação implementado pela rede neural é o produto escalar do vetor de entrada com os pesos. Estes pesos representam uma partição do espaço das características por retas e estas podem ter qualquer inclinação e qualquer ordenada ao origem. Para problemas de reconhecimento com simetria radial como este, a propriedade anterior não é apropriada. Na figura seguinte daremos um exemplo disto em duas dimensões:

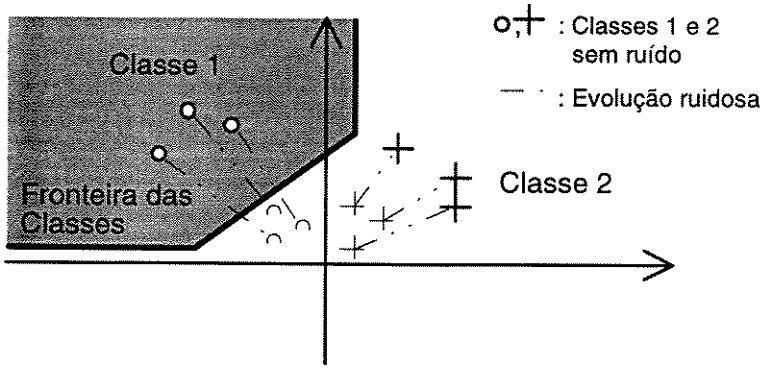


FIGURA 6.12 - Evolução das classes com ruído e partição neural clássica.

As fronteiras de classificação, com linha obscura, por ter uma ordenada ao origem diferente de zero não são robustas aos padrões, com linha ponteada, que evoluem ao origem por causa do ruído. A classificação fornecida será robusta para pequenas diminuições da SNR, quando os vetores ainda ficam dentro da classe, mas será catastrófica quando começem a cruzar as fronteiras. Isto será observado nos resultados finais.

Uma melhor aproximação a este problema sería a utilização de redes neurais com pesos sem ordenada ao origem, para ambas camadas. Neste caso a rede seria *invariante* a qualquer *dilatação e compresão linear dos vetores* a serem classificados. Chamaremos esta rede como **Rede Invariante**. Na figura seguinte observaremos a partição do espaço realizada pela rede proposta:

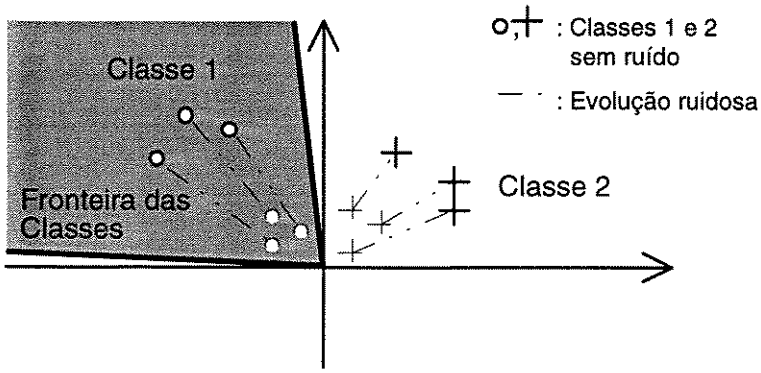


FIGURA 6.13 - Evolução das classes com ruído e partição neural invariante.

Como a separação do espaço fornecida é linear com a origem, a quantidade de neurônios na camada intermediária diminui. No exemplo anterior observamos que se precisam dois pesos para separar as classes e na outra figura se precisam três.

Também observamos que, pelo fato da partição ser linear com a origem, a modelação das fronteiras não necessáriamente será a melhor possível. Isto devería ser observado no erro final atingido no treinamento, que devería ser maior do que o erro da rede tradicional. Este resultado foi comprobado na prática.

A seguir se experimentará a combinação desta nova rede com as caracteristicas usadas anteriormente: LPC e LPC angular, para o mesmo problema de reconhecimento de palabras anterior. A rede invariante tem *cinqüenta neurônios* na camada oculta e *treze* na camada de saída.

Tipo de Arquitetura \ SNR	**40dB**	**30dB**	**20dB**	**10dB**
De Predição Linear	87.5	79.2	54.2	25
De Predição Linear - Rede Invariante	87.5	75	50	41.6
De Predição Linear Angular - Rede Invariante	83.3	75	33.3	12.5

TABELA 6.10: Reconhecimento ruidoso de palavras com rede invariante (em %)

Observamos da tabela 6.10 que até 20dB SNR os resultados das redes De Predição Linear e De Predição Linear Invariante são aproximadamente iguais. A 10dB SNR é onde as diferenças aparecem, já que a primeira logra 25% de reconhecimento e a segunda logra 41,6%. Isto implica que partindo do mesmo nível de reconhecimento sem ruído, esta nova rede é mais robusta ao ruído. Este resultado se logra pela ordenada a origem nula nos neurônios da rede.

A rede De Predição Linear Angular Invariante tem um comportamento claramente inferior à rede De Predição Linear Invariante. Esta disparidade de comportamentos é aparentemente contraditória com o fato que o treinamento destas duas redes foi feito com os mesmos vetores sem alteração angular, já que a normalização não muda o ângulo do vetor. Porém a explicação desta disparidade ilustra uma *propriedade muito importante da rede De Predição Linear Invariante.*

Quando uma rede invariante é treinada com vetores normalizados os angulos formados pelos hiperplanos estão entre os extremos observados para os vetores. Mas quando é treinada com vetores de diferente amplitude os hiperplanos separadores ficam **mais próximos dos vetores de maior amplitude**. Isto acontece porque para produzir saídas de alta amplitude nos neurônios, os pesos devem ficar o mais ortogonais possíveis dos vetores de baixa amplitude.

Na figura seguinte apresentamos um exemplo disto. Foi treinado um único neurônio invariante para a discriminação dos vetores: A=[1.2 0.2] e B=[0.1 0.3], indicados com "**o**" na figura, primeiro e logo os mesmos vetores anteriores normalizados AN=[0.9864 0.1644] e BN = [0.3162 0.9487], indicados com "**x**".

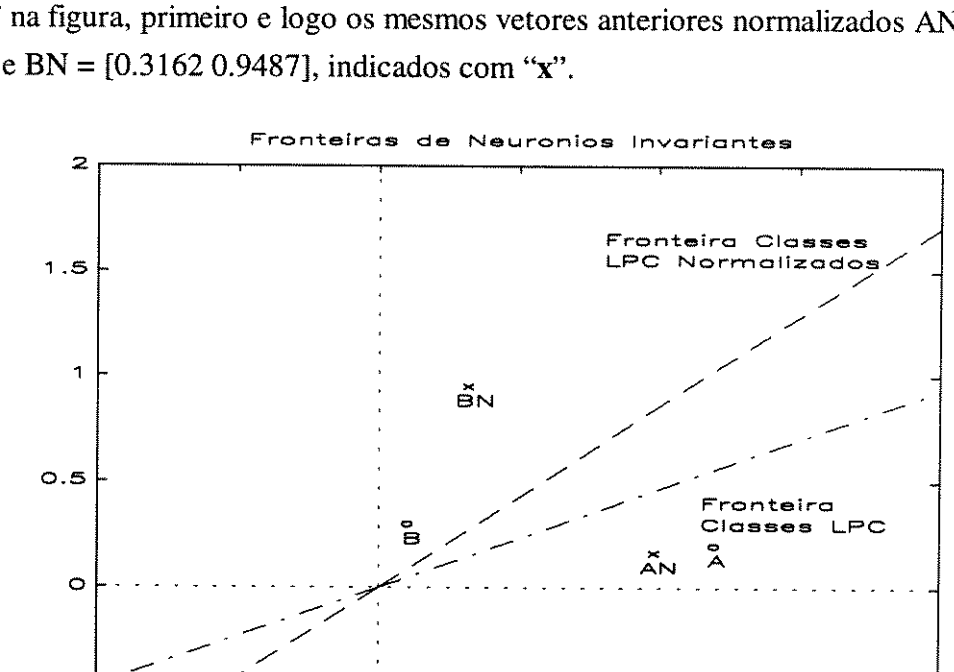


FIGURA 6.14 - Comparação fronteiras de neurônios invariantes

Observa-se que no caso dos vetores normalizados o hiperplano separador, obtido como os vetores ortogonais aos pesos da rede, passa no meio deles. No outro caso, está mais próximo do vetor de maior amplitude.

A variação angular dos vetores LPC é dependente da relação sinal - ruído (SNR). Além disso, para uma dada SNR de palavra, o nível de ruído calculado faz como que a SNR dos fonemas de menor amplitude (consoantes) seja muito mais baixa que a dos fonemas de alta amplitude (vogais). Portanto a ***variação angular dos fonemas de menor amplitude será maior que a dos fonemas de maior amplitude***.

A rede ***de Predição Linear Invariante***, pelo fato dos pesos ficarem mais próximos dos vetores de maior amplitude, ***dá maior espaço angular aos fonemas de menor amplitude que são justamente os de maior variabilidade***. Isto tem uma conseqüencia muito importante no reconhecimento ruidoso; *esta partição asimétrica é mais robusta que a partição simétrica* no sentido que otorga mais espaço angular as classes que mais precisam dele.

Na figura seguinte apresentamos um exemplo da palavra /sol/ afetada por ruído branco a uma SNR de 100dB.

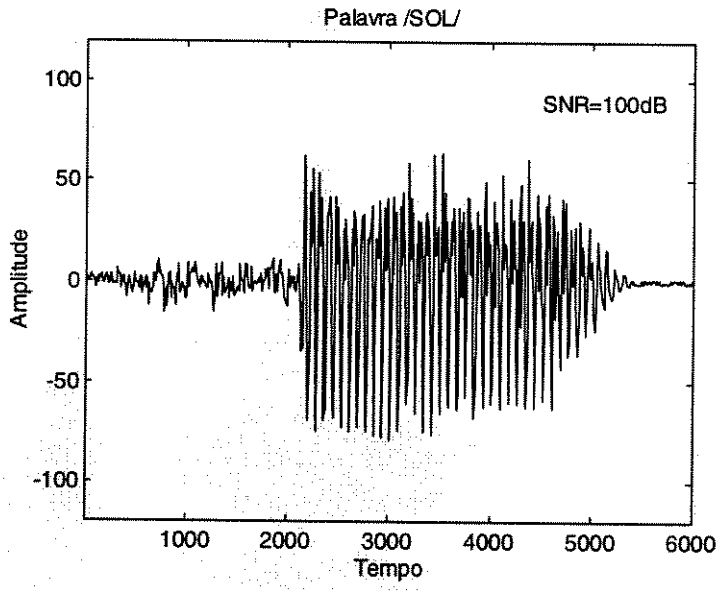


FIGURA 6.15 - Palavra /SOL/ SNR=100dB (Ruído Branco)

Na figura seguinte apresentamos os espectros em freqüência, extraído de janelas temporais de 20 ms. de duração, dos fonemas /s/ e /o/, a uma SNR de 100dB.

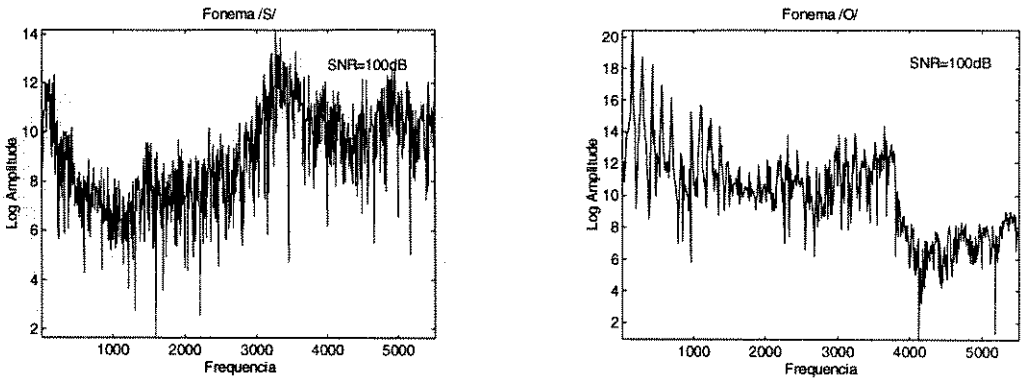


FIGURA 6.16 - Espectros em Freqüência Fonemas /S/ e /O/ SNR=100dB

Na figura seguinte observamos a mesma palavra da figura 6.15, a uma SNR de 10dB. Observamos como os fonemas de pequena amplitude são mais afetados do que os de maior amplitude.

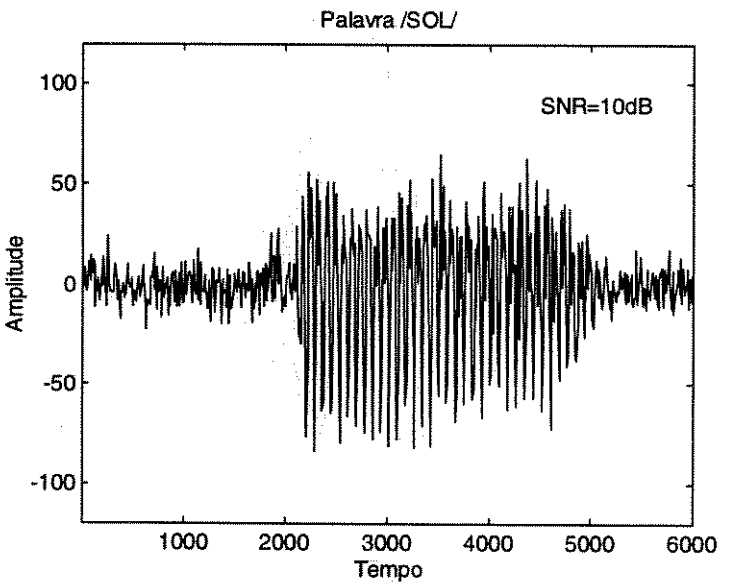


FIGURA 6.17 - Palavra /SOL/ SNR=10dB (Ruído Branco)

Observamos na figura seguinte como o espectro em freqüência do fonema /s/, de pequena amplitude, é mais afetado do que o espectro do fonema /o/, de maior amplitude.

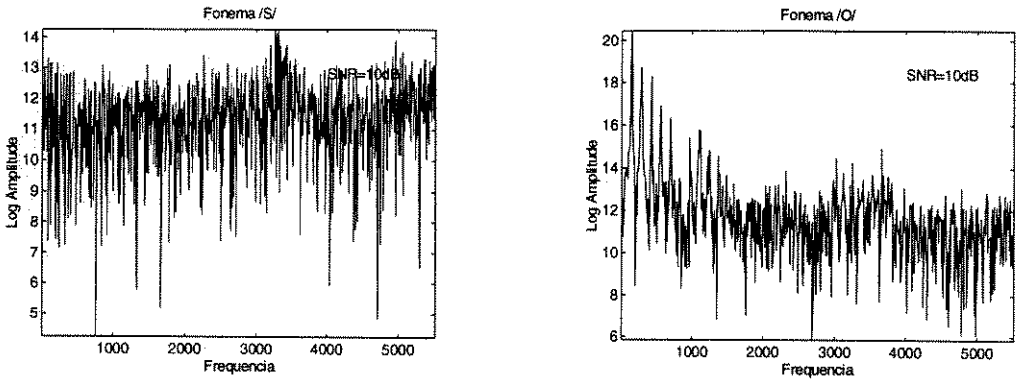


FIGURA 6.18 - Espectros em Freqüência Fonemas /S/ e /O/ SNR=10dB

A seguir estudaremos se a rede de reconhecimento adaptativo melhora os resultados das duas melhores arquiteturas da tabela anterior. O valor dos coeficientes que regulam a profundidade da memória dos filtros de Kalman da segunda camada foram assignados igual aos dos filtros da primeira camada.

Tipo de Arquitetura \ SNR	40dB	30dB	20dB	10dB
De Predição Linear	87.5	79.2	54.2	25
De Predição Linear Rede de Reconhecimento Adaptativo	87.5	79.2	54.2	29.2
De Predição Linear - Rede Invariante	87.5	75	50	41.6
De Predição Linear Rede Invariante de Reconhecimento Adaptativo	87.5	75	50	41.6

TABELA 6.11: Reconhecimento Ruidoso (em %)

Observamos da tabela 6.11 que a melhoría é mínima pelo fato das saídas dos neurônios das camadas intermédiarías serem não ruidosos.

6.8- CONCLUSÕES

O modelo de banco de filtros foi o de pior índice de reconhecimento entre as arquiteturas propostas. As razões de seu pobre desempenho são variadas. Por um lado se demostrou que a estimação de energias instantâneas, é polarizada e que a polarização depende da relação sinal - ruído. As tentativas de superar esta limitação foram infrutuosas. Por outro lado os bancos de filtros tem uma limitação natural por sua quantização não adaptativa do espectro.

Para superar os resultados anteriores foi proposto o modelo de predição linear. A suposição de um modelo de predição permitiu uma boa discriminação a altas SNR. Além disso facilitou a incorporação de mecanismos de robustecimento frente ao ruído.

Isto foi realizado na arquitetura de Predição Linear Angular. Esta foi a que teve os melhores índices no reconhecimento de vogais, tanto em altas SNR, onde a representação de Predição Linear fornece a melhor discriminação, como em baixas SNR, onde a representação angular foi a mais robusta frente ao ruído.

No reconhecimento de fonemas a rede de Predição Linear Angular teve um desempenho ruim pelo fato de que o círculo unitário é um espaço pequeno para a separação dos treze fonemas. Ainda mais, como se demonstrou, quando a SNR cai muito, a variabilidade angular é importante.

Como outra aproximação ao problema de classificação dos coeficientes LPC foi proposta a rede Invariante, que conta com pesos sem ordenada ao origem. Demonstrou-se que esta rede é mais robusta no sentido que da maior espaço angular aos fonemas de maior variabilidade pela condição desfavorável da SNR. Esta rede foi a que teve o melhor desempenho no reconhecimento de palavras.

Porém caberia perguntar-se onde estão as limitações da melhor arquitetura. Para o reconhecimento a altas SNR, é sabido que os coeficientes autorregresivos estão entre os melhores mas não são os melhores para a estimação de espectros de voz. Eles têm pior desempenho que os MEL-Cepstrum. Para o reconhecimento a baixas SNR, aparentemente o problema com o modelo autorregresivo é justamente a suposição de um modelo. Na medida que aumenta o ruído os coeficientes de predição linear têm um comportamento determinístico tendendo ao vetor unitário. O uso do ângulo do vetor e a rede invariante alivia em parte mas não soluciona o problema.

CAPÍTULO 7

CONCLUSÃO

A principal conclusão do modelo proposto e que é um bom extrator de redundâncias do padrão a serem classificado. Em casos onde o padrão temporal tem suficientes redundâncias e a perturbação é aleatória, pode-se obter bons índices de reconhecimentos em condições muito adversas. Porém nos problemas práticos a solução do problema de reconhecimento não é conhecida, isto é, o treinamento chega a mínimos locais. As redundâncias são dependentes do tipo de padrão. No caso dos fonemas alguns têm grandes redundâncias, como as vogais, e outros têm poucas redundâncias, como as plosivas e as fricativas. Por último, as pertubações aleatórias são uma aproximação das pertubações reais. Todos estes problemas afetam seriamente a performance do modelo proposto.

Os problemas achados en el reconhecimento de fonemas foram diretamente conseqüencia do desemparelhamento de condição. En todos os casos se verificaram tendenças determinísticas das características extraídas da voz, frente as quais os filtros de Kalman são pouco efetivos. Neste sentido deveria-se encontrar uma melhor combinação dos mecanismos de extração de características e os de robustez do sistema de reconhecimento.

Existem varias possíveis melhorias ao modelo proposto:

- A profundidade da memória dos filtros de Kalman é fixa o que não é bom quando se combinam padrões de curta e longa duração. Sería conveniente estudar um mecanismo adaptivo para modificar a profundidade da memória tal que creça com fonemas de longa duração como as vogais e decreça com os de curta duração, como as plosivas.

- O principal problema do modelo é que a memória está incorporada na estimação pero não na classificação. Esta última pode mudar ponto a ponto já que não tem memória, o qual não é um comportamento desejado quando a mudança de classificação dos padrões é suave. Haberia que providenciar mecanismos para correlacionar as classificações anteriores com a atual. Uma possível solução seriam as redes neurais recorrentes onde a classificação a um tempo dado depende das anteriores pela realimentação das saídas passadas.

- Seria conveniente realimentar a informação da classificação à estimação atual. Isto poderia reduzir a variabilidade da estimação, já que se um fonema está sendo corretamente classificado e sua observação sai temporariamente da classe, sua estimação poderia permanecer nela.

• Haberia que estudar mecanismos de minimização do custo computacional do modelo. Isto é fundamental para sua aplicação a problemas de grande escala. Uma possível idéia seria o trabalho em escalas de tempos diferentes nas camadas da rede. Neste sentido se poderia trabalhar com médias temporais passadas, para classificar o padrão de entrada numa escala de tempo diferente do própio sinal.

Existem várias possíveis melhorias ao modelo proposto para reconhecimento de voz:

• Estudar a aplicação da Rede Invariante usando como caractéristicas os Mel-pstral. Esta é uma linha de investigação promissora, no sentido que estas características, por um lado têm uma melhor discriminação para condições emparelhadas, e por outro lado têm o mesmo comportamento frente a condições desemparelhadas.

CAPITULO 8

BIBLIOGRAFIA:

[1] Mansour D., Juang B. H., "***A family of distorsion measures based upon projection operation for robust speech recognition***", IEEE Transactions on Acoustics Speech and Signal Processing (TASSP), Novembro 1989.

[2] Lippmann R., ***"An introduction to computing with neural nets"***, IEEE Acoustics Speech and Signal Processing (ASSP) magazine, Abril 1987.

[3] Bourlard H., Wellekens C. J. ***"Speech pattern discrimination and multilayer perceptrons"***, Computer Speech and Language, **3**, pp. 1-19, 1989.

[4] Mansour D., Juang B. H., ***"The short time modified coherence representation and noisy speech recognition"***, IEEE TASSP Julho. 1989.

[5] Hush D. R., Horne B., ***"Progress in supervised neural networks"***, IEEE ASSP Magazine, Janeiro 1993.

[6] Juang B. H., Paliwal K.K., ***"Hidden markov models with first order equalization for noisy speech recognition"***, IEEE TASSP 1992.

[7] Widrow B., Lehr M. A., ***"30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation"***, IEEE Proceedings Vol. 78, No. 9, Setembro 1990.

[8] Gelb A. Ed., **Applied Optimal Estimation**, Cambridge, MA: M.I.T. Press, 1974.

[9] Rosenblatt A., "**The *Perceptron: A probabilistic model for information storage and organization in the brain.***" Psychological Review,65:386-408, 1958.

[10] Hopfield J. J, "***Neural networks and physical systems with emergent collective computational abilities,*** " Proceedings of the National Academy of Science, USA, 79 pp. 2554-2558, 1982.

[11] Hopfield J. J., "***Neurons with a graded response have collective computational properties like those of two-state neurons.***" Proceedings of the National Academy of Science, USA, 81 pp. 3088-3092, Maio 1984.

[12] Rumelhart D. E., Hinton G. E.y Williams R. J., "***Learning internal representations by error propagation.***" Parallel Distributed Processing: Explotations in the Microstructure of Cognition, pp. 318-362. MIT Press, Cambridge MA, 1986.

[13] Burr D. J., "***A neural network digit recognizer***,"in Proc. IEEE Int. Conf. Syst. Man. Cybern. Outubro, 1986.

[14] Waibel A., et al. "***Phoneme recognition using Time-delay neural networks***", IEEE Transactions on Acoustics, Speech and Signal Proccesing, Vol. 37, No. 3, Março 1989.

[15] Robinson T. y Fallside F., "***A recurrent error propagation network speech recognition system***", Computer Speech and Language, 5, pp. 259-274, 1991.

[16] Schalkoff R. J., **Pattern Recognition, Statistical, Structural and Neural Approaches**, John Wiley & Sons, New York, 1992.

[17] Duda R. O., Hart P. E., **Pattern Clasification and Scene Analysis**, John Wiley & Sons, New York, 1973.

[18] Haykin S., **Adaptive filters**, 2nd. ed. Prentice Hall, Englewood Cliffs, N. J., 1991

[19] Haykin S., **Modern filters**, Maxwell Macmillan, New York, 1990.

[20] Haykin S., **Neural Networks, A Comprehensive Foundation**, Maxwell Macmillan, New York, 1994.

[21] Juang B. H., "***Speech recognition in adverse enviroments***", Computer Speech and Language, 1991, 5, pp. 275-294.

[22] Rocha L. F., **Procesamiento de la voz**, Curso de la EBAI II. Ed. Kapeluz, Fevereiro 1987.

[23] Oppenheim A., Schafer R., **Digital Signal Processing**, Prentice Hall, Englewood Cliffs, N. J, 1975.

[24] Rabiner L. Schafer R., **Digital Processing of Speech Signals**, Prentice Hall, Englewood Cliffs, N. J, 1978.

[25] Borzone de Manrique A. M., **Manual de Fonética Acústica**, Hachette, 1980.

[26] Proakis J. G., Manolakis D. G., **Introduction to Digital Signal Procesing**, Maxwell Macmillan, New York, 1989.

[27] Helstrom C., **Probability and Stochastic Processes for Engineers**, 2ed, Maxwell Macmillan, New York, 1991.

[28] **MATLAB 4.0, user's guide**, the MathWorks.

[29] Mahoul J., "***Linear Prediction. A tutorial review***", Proceedings IEEE, vol. 63, pp 501-580, 1975.

[30] Markel J., Gray A., **Linear Prediction of Speech**, Springer Verlag, New York, 1976.

[31] Deller J. R., Proakis J. G., Hansen J. H. L., **Discrete Time Proccesing of Speech Signals**, Mac Millan, 1993.

[32] Hertz J., Krogh A., Palmer R. G., **Introduction to the Theory of Neural Computation**, Addison-Wesley, 1991.

[33] Príncipe J. C., Kuo J, Selebi S., "***An Analisys of the Gamma Memory in Dynamic Neural Networks***", IEEE Transactions in Neural Networks, página 331, Marzo 1994.

[34] Kohonen, T, **Self Organization and Associative Memories**, 3rd Edition, Springer Verlag, New York, 1988.

[35] Franco H, ***Informe de Tareas Realizadas Correspondiente al Plan de Tesis "Reconocimiento Fonético-Acustico Automático"***, Publicación 006, Secretaría de Investigación y Doctorado, Facultad de Ingeniería, UBA, Septiembre 1992.

[36] Ghitza O, "***Auditory Nerve Representation as a Front End for Speech Recognition in a Noisy Enviroment***", Computer Speech and Language, 1, 109-130, 1986.

[37] Carlson B. A. y Clements B., ***"A Projection-Based Likelihood Measure for Speech Recognition in Noise"***, IEEE Transactions on Speech and Audio Processing, Vol 2, No 1, 97-102, Janeiro 1994.

[38] Sandhu S. y Ghitza O., ***"A Comparative Study or Mel Cepstra and EIH for Phone Classification under Adverse Conditions"***, Proceedings del International Conference on Acoustics, Speech and Signal Processing (ICASSP'95), 409-412, 1995.

[39] Paliwal K. K., ***"Neural Nets Classifiers for Robust Speech Recognition under Noisy Environments"***, Proceedings del International Conference on Acoustics, Speech and Signal Processing (ICASSP'90), 429-432, 1990.

[40] Tishby N., ***"A Dynamical Systems Approach to Speech Processing"***, Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'90), pp. 365-368, 1990.

[41] Levin E., ***"Word Recognition using Hidden Control Neural Architecture"***, Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'90), pp. 433-436, 1990.

[42] Dautrich, B. A., L. A. Rabiner and T. Martin, ***"On the effect of Varying folter bank parameters on isolated word recognition"***, IEEE Trans. Acoustics, Specch and Signal Processing, 31, 4, pp. 793-807 (1983).

[43] Kaiser, J. F., "***On a simple algorithm to calculate the 'energy' of a signal***", Proceedings del International Conference on Acoustics, Speech and Signal Processing (ICASSP'90), pp. 381-384, 1990.

[44] Ramon y Cajal, S., "**Histologie du système nerveux de l'homme et des vertébres"**. Paris: Maloine; Edition Francaise Revue: Tome I, 1952; Tome II, 1955; Madrid: Consejo Superior de Investigaciones Científicas.

[45] M. Graciarena, M. L. Andrade Netto, "***Adaptive Recognition Neural Net Architecture Comparison for Noisy Phoneme Recogntion***", Proceedings do VI Congresso Internacional RPIC '95, pp. 37-42, Bahía Blanca, Argentina, Novembro, 1995.

[46] M. Graciarena, M. L. Andrade Netto, "***Nueva arquitectura de red neural para el reconocimiento adaptivo de fonemas ruidosos***", 6to Congresso Latinoamericano de Contrôle Automático, IFAC, Río de Janeiro, Brasil, Septembro, 1994.

[47] M. Graciarena, M. L. Andrade Netto, "***Propuesta de una red neural para el reconocimiento adaptivo de patrones ruidosos***", V Congreso Internacional RPIC '93 pp. 412-417, Tucumán, Argentina, Novembro, 1993.