



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Dep. de Eng. de Computação e Automação Industrial

Abordagem Neurofuzzy para Modelagem de Sistemas Dinâmicos Não Lineares

Michel Bortolini Hell

Prof. Dr. Fernando Gomide (Orientador)

Prof. Dr. Pyramo Pires Costa Jr. (Co-orientador)

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para obtenção do título de **Doutor em Engenharia Elétrica**. Área de Concentração: Automação.

Banca Examinadora:

Prof. Dr. Fernando Gomide – DCA - FEEC - UNICAMP (Orientador)

Prof. Dr. Walmir Matos Caminhas – DEE - UFMG

Profa. Dra. Roseli Aparecida Francelin Romero – ICMC - USP-São Carlos

Profa. Dra. Rosângela Ballini – IE - UNICAMP

Prof. Dr. Romis Ribeiro de Faissol Attux – DCA - FEEC - UNICAMP

Campinas, Outubro de 2008

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

H367a Hell, Michel Bortolini
Abordagem neurofuzzy para modelagem de sistemas dinâmicos não lineares / Michel Bortolini Hell. – Campinas, SP: [s.n.], 2008.

Orientadores: Fernando Antônio Campos Gomide, Pyramo Pires Costa Jr.

Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Modelos matemáticos. 2. Inteligência artificial. 3. Redes neurais (computação). 4. Sistemas Nebulosos. I. Gomide, Fernando Antônio Campos. II. Costa Jr., Pyramo Pires. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês: Neurofuzzy approach for nonlinear dynamical systems modeling
Palavras-chave em Inglês: Mathematical models, Artificial intelligence, Neural networks (Computer science), Neurofuzzy systems
Área de concentração: Automação
Titulação: Doutor em Engenharia Elétrica
Banca Examinadora: Walmir Matos Caminhas, Roseli Aparecida Francelin Romero, Rosângela Ballini, Romis Ribeiro de Faissol Attux
Data da defesa: 19/11/2008
Programa de Pós Graduação: Engenharia Eletrica

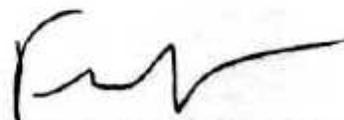
COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Michel Bortolini Hell

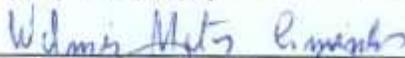
Data da Defesa: 19 de novembro de 2008

Título da Tese: "Abordagem Neurofuzzy para Modelagem de Sistemas Dinâmicos Não Lineares"

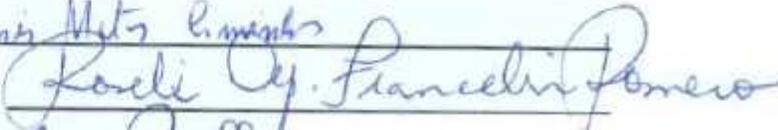
Prof. Dr. Fernando Antônio Campos Gomide (Presidente): _____



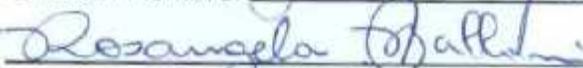
Prof. Dr. Walmir Matos Caminhas: _____



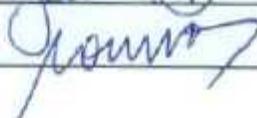
Profa. Dra. Roseli Aparecida Francelin Romero: _____



Profa. Dra. Rosângela Ballini: _____



Prof. Dr. Romis Ribeiro de Faissol Attux: _____



“Toda nossa ciência, comparada com a realidade, é primitiva e infantil e, no entanto, é a coisa mais preciosa que temos.”

Albert Einstein

À Rafa, eterna companheira, por sua força e coragem perante as dificuldades. Por ter lutado comigo em busca dos meus ideais. À ela dedico não só este trabalho, mas também toda a minha vida. À ela dedico o meu incondicional amor.

“Nunca se esqueça, nem um segundo, que eu tenho o amor maior do mundo, ...”

Agradecimentos

Ao Prof. Dr. Fernando Gomide, pela orientação, por sua grande contribuição para minha formação acadêmica e científica e pelo exemplo de profissionalismo e humildade que só os grandes possuem, nos quais sempre me orientarei ao trilhar meus caminhos.

Ao Prof. Dr. Pyramo Pires da Costa Jr., pela co-orientação, por me abrir as portas do mundo científico e pelo empenho na realização deste trabalho. Minha profunda admiração a este pesquisador que de forma única levou-me a desbravar novas fronteiras. Obrigado pela amizade e apoio incondicional durante todos estes anos.

Ao Prof. Dr. Fernando José Von Zuben e aos colegas do LBIC Renan, Pablo, Hamilton, Wilfredo, Rodrigo, Alan e Guilherme, que me receberam de braços abertos na fase final deste trabalho.

À memória do Prof. Dr. Adimilson Bosco Chitarra, pessoa extraordinária e fonte de inspiração que me levou a buscar os caminhos da pesquisa e que tão prematuramente completou sua missão na terra.

À FAPESP, pelo apoio financeiro N° 2003/05042-1.

Aos colegas do LASI da PUC-MG, em especial ao Daniel e ao Lucas, pelo incentivo.

Ao Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas (DCA-FEEC-UNICAMP) e a todos os professores, colegas e funcionários que de alguma forma contribuíram para este trabalho.

Aos meus pais, Amilton e Lizete, que sempre me apoiaram e acreditaram nos meus ideais e que sempre foram o exemplo de caráter e honestidade que nortearam minhas ações. A eles dedico meu eterno amor.

Aos meus irmãos Charles e Milena, à Eliana e aos meus sobrinhos, Charlinhos e Mariana, que mesmo a mais de 1.000 quilômetros de distância, todos sempre me deram muita força para que eu atingisse meus objetivos.

Ao Arnaldo e à Neuza, verdadeiros pais que, além de me presentear com a coisa que mais amo na vida, sempre me apoiaram e incentivaram para que eu chegasse até aqui.

À Isabela, ao Marcelo e ao Luiz Felipe, pelos inúmeros momentos de distração e pelas cavalgadas que sem dúvida contribuíram para que fosse possível concluir este trabalho.

Aos meus amigos Alessandro e Juliana pelos muitos momentos felizes que ajudaram a amenizar as dificuldades e muito ajudaram nesta caminhada.

E mais uma vez à Rafa, companheira de todas as horas, pelo amor e compreensão que me dedicou durante todos estes anos. Por suportar as ausências e por caminhar ao meu lado, superando juntos todas as dificuldades. Agradeço à ela também pelas valiosas discussões na área de neurociência.

Resumo

Este trabalho propõe um procedimento sistemático para obtenção de modelos de sistemas dinâmicos não-lineares complexos utilizando redes neurais nebulosas. As redes neurais nebulosas aplicadas em modelagem são capazes de extrair conhecimento de dados entrada/saída e representar este conhecimento na forma de regras nebulosas do tipo se-então, gerando modelos lingüísticos convenientes para compreensão humana. Duas novas classes de redes neurais nebulosas são propostas a partir de generalizações dos neurônios lógicos *AND* e *OR*. Estas generalizações, denominadas unineurons e nullneurons, implementam, além da plasticidade sináptica, outra importante característica dos neurônios biológicos, a plasticidade neuronal. Desta forma, os neurônios propostos neste trabalho são capazes de modificar parâmetros internos em resposta à alterações, permitindo que unineurons e nullneurons variem individualmente de um neurônio *AND* para um neurônio *OR* (e vice-versa), dependendo da necessidade do problema. Conseqüentemente, uma rede neural nebulosa composta por unineurons e nullneurons é mais geral do que as redes neurais nebulosas similares sugeridas na literatura. Além da introdução de redes neurais com unineurons e nullneurons, um novo algoritmo de treinamento para obtenção de modelos nebulosos de sistemas dinâmicos é proposto utilizando aprendizado participativo. Neste algoritmo, uma nova informação fornecida à rede por meio de um dado entrada/saída é comparada com o conhecimento que já se tem a respeito do sistema. A nova informação só tem influência na atualização do conhecimento se não entrar em conflito com o conhecimento adquirido anteriormente. Como conseqüência, redes neurais nebulosas que utilizam este novo algoritmo de treinamento são mais robustas a dados de treinamento com valores que correspondem a comportamentos anômalos ou mesmo a erros durante a obtenção destes dados. As abordagens propostas foram utilizadas para desenvolver modelos para previsão de séries temporais e modelagem térmica de transformadores de potência. Os resultados experimentais mostram que os modelos aqui propostos são mais robustos e apresentam os melhores desempenhos, tanto em termos de precisão quanto em termos de custos computacionais, quando comparados com abordagens alternativas sugeridas na literatura.

Palavras-chave: Modelagem, sistemas dinâmicos, redes neurais, sistemas nebulosos, redes neurais nebulosas, unineurons, nullneurons, aprendizado participativo.

Abstract

This work suggests a systematic procedure to develop models of complex nonlinear dynamical systems using neural fuzzy networks. The neural fuzzy networks are able to extract knowledge from input/output data and to encode it explicitly in the form of if-then rules. Therefore, linguistic models are obtained in a form suitable for human understanding. Two new classes of fuzzy neurons are introduced to generalize *AND* and *OR* logic neurons. These generalized logic neurons, called unineurons and nullneurons, provide a mechanism to implement synaptic plasticity and an important characteristic of biological neurons, the neuronal plasticity. Unineurons and nullneurons achieve synaptic and neuronal plasticity modifying their internal parameters in response to external changes. Thus, unineurons and nullneurons may individually vary from a *AND* neuron to a *OR* neuron (and vice-versa), depending upon the necessity of the modeling task. Neural fuzzy networks constructed with unineurons and nullneurons are more general than similar fuzzy neural approaches suggested in literature. Training algorithms for neural fuzzy networks with unineurons and nullneurons are also studied. In particular, a new training algorithm based on the participatory learning is introduced to develop fuzzy models of dynamical systems. In the participatory learning algorithm, a new information brought to the network through an input/output data is first compared with the knowledge that the network already has about the model. The new information influences the update of the knowledge only if it does not conflict with the current knowledge. As a result, neural fuzzy networks trained with participatory learning show greater robustness to training data with anomalous values than their counterparts. The neural fuzzy network and training algorithms suggested herein are used to develop time series forecast models and thermal models of power transformers. Experimental results show that the models proposed here are more robust and perform best in terms of accuracy and computational costs when compared against alternative approaches suggested in the literature.

Keywords: Modeling, dynamical systems, neural networks, fuzzy systems, neural fuzzy networks, unineurons, nullneurons, participatory learning.

Sumário

Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Símbolos	xxi
Lista de Siglas e Abreviações	xxiii
Trabalhos Publicados Pelo Autor	xxiv
1 Introdução	1
1.1 Motivação e Relevância	1
1.2 Objetivos	4
1.3 Organização do Trabalho	5
2 Fundamentos Teóricos	7
2.1 Introdução	7
2.2 Modelagem Matemática	7
2.2.1 Classificações de um Modelo Matemático	8
2.2.2 Abordagens Baseadas em Inteligência Computacional	9
2.3 Conceitos de Teoria de Conjuntos Nebulosos	11
2.3.1 Conjuntos Nebulosos	12
2.3.2 Normas Triangulares	12
2.3.3 Uninormas e Nullnormas	14
2.3.4 Operador de Compromisso	16
2.3.5 Relações Nebulosas	16
2.3.6 Composição de Relações Nebulosas	17
2.3.7 Equações Relacionais Nebulosas	19
2.3.8 Solução do Problema de Estimação (SPE)	20
2.4 Neurônios e Redes Neurais Nebulosas	23
2.4.1 Neurônios Nebulosos	24
2.4.2 Redes Neurais Nebulosas	29
2.5 O Espaço Neural Multidimensional	33
2.6 Aprendizado Participativo	34
2.7 Resumo	38

3	Redes Neurais Nebulosas com Aprendizado Participativo	39
3.1	Introdução	39
3.2	Análise de Convergência do Aprendizado Participativo	42
3.3	Estrutura da Rede Neural Nebulosa	48
3.4	Treinamento de Redes Neurais Nebulosas com AP	51
3.4.1	Geração das Funções de Pertinência	52
3.4.2	Atualização dos Pesos Associados à Rede Neural de Agregação	53
3.4.3	Atualização dos Pesos Associados ao Sistema de Inferência Nebuloso	53
3.4.4	Procedimentos de Treinamento com AP	55
3.5	Resumo	59
4	Generalização de Neurônios Nebulosos	61
4.1	Introdução	61
4.2	Neurônio Nebuloso Baseado em Uninormas: O Unineuron	63
4.3	Neurônio Nebuloso Baseado em Nullnormas: O Nullneuron	67
4.4	Redes Neurais Nebulosas baseadas em Nullneurons e Unineurons	76
4.5	Aprendizado para Redes Baseadas em Unineurons e Nullneurons	78
4.6	Resumo	83
5	Resultados Experimentais e Aplicações	85
5.1	Introdução	85
5.2	Predição de Séries Temporais	87
5.2.1	Previsão de Carga de Curto Prazo	87
5.3	Aplicações a Problemas de Identificação de Sistemas	96
5.3.1	Problema do Forno a Gás de Box&Jenkins	97
5.3.2	Modelagem Térmica de Transformadores de Potência	98
5.4	Resumo	109
6	Conclusão	111
	Referências bibliográficas	114

Lista de Figuras

2.1	Conjunto nebuloso F em um universo de discurso X	13
2.2	Realizações alternativas de uma uninorma: (a) <i>and</i> -dominada, (b) <i>or</i> -dominada . . .	15
2.3	Sistema relacional nebuloso de uma entrada - uma saída	19
2.4	Neurônio nebuloso do tipo I	25
2.5	Neurônio nebuloso do tipo II	25
2.6	Neurônio nebuloso do tipo III	26
2.7	Neurônio nebuloso lógico do tipo <i>AND</i>	28
2.8	Neurônio nebuloso lógico do tipo <i>OR</i>	29
2.9	Processador lógico do tipo Soma de Mintermos	30
2.10	Processador lógico do tipo Produto de Maxitermos	31
2.11	Modelo de rede neural nebulosa para classificação de padrões	32
2.12	Modelo de rede neural nebulosa recorrente	33
2.13	Comportamento da aprendizagem participativa	36
2.14	Aprendizado participativo com alerta	36
2.15	Comportamento do índice de alerta e da variação das convicções: (a) Índice de compatibilidade (b) Índice de alerta (c) Variação das convicções.	37
3.1	Exemplo de Rede Neural Clássica de 3 entradas e 2 saídas.	40
3.2	Exemplo de Rede Neural Nebulosa de 3 entradas e 2 saídas.	41
3.3	Convergência do Aprendizado Participativo para Variável Aleatória com Distribuição Normal: (a) Valor inicial $v^0 = 0$, (b) Valor inicial $v^0 = 1$	47
3.4	Convergência do Aprendizado Participativo para Variável Aleatória com Distribuição Diamétrica Não-Central: (a) Valor inicial $v^0 = 0$, (b) Valor inicial $v^0 = 1$	48
3.5	Modelo de rede neural nebulosa	49
3.6	Exemplo de Geração das Funções de Pertinência para $\mathbf{x} \in \mathbb{R}^2$ e $y \in [0, 1]$	52
4.1	Unineuron Aplicado a um Problema Bi-dimensional.	67
4.2	Superfície de Decisão do Unineuron para $\mathbf{w}=[0.3 \ 0.8]$, t-norma = min, s-norma = max e: (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$	68
4.3	Superfície de Decisão do Unineuron para $\mathbf{w}=[0.3 \ 0.8]$, t-norma = produto algébrico, s-norma = soma probabilística e: (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$	69

4.4	Superfície de Decisão do Unineuron para $w=[0.3 \ 0.8]$, t-norma = produto limitado, s-norma = soma limitada e: (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$	70
4.5	Nullneuron Aplicado à Problema Bi-dimensional.	72
4.6	Superfície de Decisão do nullneuron para $w=[0.3 \ 0.8]$, t-norma = min, s-norma = max e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$	73
4.7	Superfície de Decisão do nullneuron para $w=[0.3 \ 0.8]$, t-norma = produto algébrico, s-norma = soma probabilística e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$	74
4.8	Superfície de Decisão do nullneuron para $w=[0.3 \ 0.8]$, t-norma = produto limitado, s-norma = soma limitada e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$	75
4.9	Modelo de rede neural nebulosa recorrente com unineurons e nullneurons	76
5.1	Curva de carga para o período de 13 a 19 de abril de 2001.	88
5.2	Curva de carga para o período de 13 a 19 de maio de 2001.	89
5.3	Curva de carga com perfis anômalos.	91
5.4	Resultados para previsão de carga - Caso 1: (a) MLP , (b) ANFIS , (c) NN-AND , (d) NN-OR , (e) AND-AP	92
5.5	Resultados para previsão de carga - Caso 1: (f) OR-AP , (g) UNI-R , (h) NULL-R , (i) NULL-G	93
5.6	Resultados para previsão de carga - Caso 2: (a) MLP , (b) ANFIS , (c) NN-AND , (d) NN-OR , (e) AND-AP	94
5.7	Resultados para previsão de carga - Caso 2: (f) OR-AP , (g) UNI-R , (h) NULL-R , (i) NULL-G	95
5.8	Resultados obtidos para o forno a gás de Box e Jenkins: (a) MLP , (b) ANFIS , (c) NN-AND , (d) NN-OR , (e) AND-AP	99
5.9	Resultados obtidos para o forno a gás de Box e Jenkins: (f) OR-AP , (g) UNI-R , (h) NULL-R , (i) NULL-G	100
5.10	Dados de treinamento - Caso 1.	103
5.11	Resultados da modelagem térmica de transformadores de potência (Dados de Teste 1 - DT1): (f) OR-AP , (g) UNI-R , (h) NULL-R , (i) NULL-G	105
5.12	Resultados da modelagem térmica de transformadores de potência (Dados de Teste 2 - DT2): (a) MLP , (b) ANFIS , (c) NN-AND , (d) NN-OR , (e) AND-AP	106
5.13	Resultados da modelagem térmica de transformadores de potência (Dados de Teste 2 - DT2): (f) OR-AP , (g) UNI-R , (h) NULL-R , (i) NULL-G	107
5.14	Dados de treinamento - Caso 2.	108

Lista de Tabelas

5.1	Parâmetros Estruturais dos Modelos Implementados - Previsão de Carga	90
5.2	Resultados para Previsão de Carga - Caso 1	96
5.3	Resultados para Previsão de Carga - Caso 2	96
5.4	Resultados para o Forno a Gás de Box&Jenkins	98
5.5	Resultados da Literatura para o Forno a Gás de Box&Jenkins	98
5.6	Principais Características do Transformador a ser Modelado	102
5.7	Resultados da Modelagem da Transformadores - Caso 1	103
5.8	Resultados da Modelagem de Transformadores - Caso 2	108

Lista de Símbolos

\mathbb{R}	- Conjunto dos números reais
\mathbb{C}	- Conjunto dos números complexos
$A_i^{\lambda_i}$	- λ_i -ésimo conjunto nebuloso da entrada i
$(x \ t \ y)$	- Norma triangular (t-norma) entre x e y
$T(x, y)$	- Norma triangular (t-norma) entre x e y
$(x \ s \ y)$	- Conorma triangular (s-norma) entre x e y
$S(x, y)$	- Conorma triangular (s-norma) entre x e y
$U(x, y; g)$	- Uninorma entre x e y com elemento identidade g
$\tilde{U}(x, y; g)$	- Uninorma de dominância controlada entre x e y com elemento identidade g
$N(x, y; u)$	- Nullnorma entre x e y com elemento de absorção u
$\tilde{N}(x, y; u)$	- Nullnorma alternativa entre x e y com elemento de absorção u
$\tilde{C}_u(\cdot)$	- Operador de compromisso parametrizado por u
$\sup_{x \in X} \{\cdot\}$	- Operador supremo sobre o conjunto X
$\inf_{x \in X} \{\cdot\}$	- Operador ínfimo sobre o conjunto X
$X \times Y$	- Produto cartesiando entre os espaços X e Y
$[0, 1]^n$	- Hipercubo unitário de dimensão n , i.e., $\overbrace{[0, 1] \times [0, 1] \times \dots \times [0, 1]}^n$
$[\mathbf{x}^k, \mathbf{y}^k]$	- k -ésimo dado de treinamento
$\hat{\mathbf{y}}^k$	- Saída obtida no instante k
\mathbf{w}_ℓ	- ℓ -ésima linha da matriz W
q^{-s}	- Operador de atraso de s passos
$E\{\cdot\}$	- Esperança matemática
$Pr\{\cdot\}$	- Função probabilidade
$P[\cdot]$	- Função distribuição de probabilidades

Lista de Siglas e Abreviações

ANFIS	-	<i>Adaptative Fuzzy Inference System</i>
AP	-	Aprendizado Participativo
ARMAX	-	<i>Autoregressive Moving Average with Exogenous Inputs</i>
ARX	-	<i>Autoregressive with Exogenous Inputs</i>
FIR	-	<i>Finite Impulse Response</i>
FIS	-	<i>Fuzzy Inference System</i>
FT	-	Função de Transferência
GD	-	Método do Gradiente
IC	-	Inteligência Computacional
IIR	-	<i>Infinite Impulse Response</i>
MAPE	-	<i>Mean Average Percentage Error</i>
MLP	-	<i>Multilayer Perceptron</i>
MSE	-	<i>Mean Square Error</i>
NARMAX	-	Nonlinear Autoregressive Moving Average with Exogenous Inputs
NARX	-	<i>Nonlinear Autoregressive with Exogenous Inputs</i>
NDEI	-	<i>Non-dimentional Error Index</i>
RA	-	Reforço Associativo
RMSE	-	<i>Root Mean Square Error</i>
RNA	-	Rede Neural Artificial
SPE	-	Solução do Problema de Estimação

Trabalhos Publicados Pelo Autor

Capítulos de Livro

1. E. Lima, M. Hell, R. Ballini and F. Gomide, “Evolving Fuzzy Modeling Using Participatory Learning”, In: P. Angelov D. Filev and N. Kasabov (Eds.), *Evolving Intelligent Systems: Methodology and Applications*, 440pp., John Willey and Sons, 2008.
2. M. Hell, L. Secco, P. Costa Jr., F. Gomide, “Recurrent Neural Approaches for Power Thermal Transformer Modeling”. In: S. K. Pal et. al. (Org.), *Pattern Recognition and Machine Intelligence*, 804pp., Heidelberg: Springer Verlag, 2005.

Artigos em Periódicos

1. M. Hell, P. Costa Jr., F. Gomide, “Recurrent Neurofuzzy Network in Thermal Modeling of Power Transformers”, *IEEE Transactions on Power Delivery*, Vol. 22, No. 2, pp. 904-910, April 2007.
2. M. Hell, P. Costa Jr., F. Gomide, “Participatory Learning in Power Transformers Thermal Modeling”, *IEEE Transactions on Power Delivery*, Vol. 23, No. 4, pp. 2058-2067, Oct. 2008.
3. M. Hell, P. Costa Jr., F. Gomide, “Unineurons-based hybrid neurofuzzy network in nonlinear system modeling”, *Nonlinear Analysis: Theory, Methods & Applications*, 2008 (aceito), .
4. D. Leite, M. Hell, P. Costa Jr., F. Gomide, “Real-time fault diagnosis of nonlinear systems”, *Nonlinear Analysis: Theory, Methods & Applications*, 2008 (aceito).
5. M. Hell, P. Costa Jr., F. Gomide, “A Novel Method for Thermal Modeling of Power Transformers Based on Nullneurons”, *Electric Power Systems Research*, 2008 (submetido).
6. M. Hell, P. Costa Jr., F. Gomide, “Neurofuzzy Short-Term Load Forecasting with Participatory Learning”, *Engineering Applications of Artificial Intelligence*, 2008 (submetido).
7. M. Hell, P. Costa Jr., W. Pedrycz, F. Gomide, “Generalized Fuzzy Set-Based Neurons: Unineurons, Nullneurons and Neurofuzzy Networks”, *IEEE Transactions on Fuzzy Systems*, 2008 (submetido).

Artigos em Congressos

1. M. Hell, P. Costa Jr., F. Gomide, “Neurons and neural fuzzy networks based on nullnorms”, *Proceedings of 10th Brazilian Symposium on Artificial Neural Networks, SBRN 08*, pp. 1-6, Salvador-BA, Brasil, October 2008.
2. M. Hell, P. Costa Jr., F. Gomide, “Redes Neurofuzzy Baseadas em Nullneurons”, *Anais do XVII Congresso Brasileiro de Automática, CBA 08*, pp. 1-6, Juiz de Fora-MG, Brasil, Setembro 2008.
3. M. Hell, P. Costa Jr., F. Gomide, “Hybrid Neurofuzzy Computing With Nullneurons”, *Proceedings of IEEE International Joint Conference on Neural Networks, IJCNN '08*, pp. 3652-3658, Hong Kong-HK, China, June 2008.

4. M. Hell, R. Ballini, P. Costa Jr., F. Gomide, “Treinamento de Redes Neurofuzzy via Aprendizado Participativo”, *Anais do VIII SBAI - Simpósio Brasileiro de Automação Inteligente*, pp. 1-6, Florianópolis-SC, Brasil, pp. 1-6, Outubro 2007.
5. M. Hell, R. Ballini, P. Costa Jr., F. Gomide, “Training Neurofuzzy Networks with Participatory Learning”, *New Dimensions in Fuzzy Logic and Related Technologies - Proceedings of the 5th EUSFLAT Conference*, Vol. II, pp. 231-236, Ostrava, Czech Republic, September 2007.
6. M. Hell, P. Costa Jr., F. Gomide, “New Neurofuzzy Training Procedure Based on Participatory Learning Paradigm”, *Proceedings of IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2007*, pp. 1952-1957, London, UK, July 2007.
7. M. Hell, P. Costa Jr., F. Gomide, “Nullneurons-Based Hybrid Neurofuzzy Network”, *Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS '07*, pp. 331-336, San Diego-CA, USA, June 2007.
8. D. Leite, M. Hell, P. Diez, B. Gariglio, L. Nascimento, P. Costa Jr., “Real-Time Model-Based Fault Detection and Diagnosis for Alternators and Induction Motors”, *Proceedings of IEEE International Electric Machines & Drives Conference, IEMDC '07*, pp. 202-207, Antalya, Turkey, May 2007.
9. M. Hell, L. Secco, P. Costa Jr., F. Gomide, “Redes Neurais e Neuro-Fuzzy Aplicadas a Modelagem Térmica de Transformadores de Potência”, *Anais do VII Congresso Brasileiro de Redes Neurais, CBRN 05*, pp. 1-6, Natal-RN, Brasil, Outubro 2005.
10. M. Hell, L. Secco, P. Costa Jr., F. Gomide, “Elman Recurrent Neural Network in Thermal Modeling of Power Transformers”, *Proceedings of International Conference on Power Engineering Systems, ICOPES 2005*, pp. 1-6, Rio de Janeiro-RJ, Brazil, April 2005.
11. M. Hell, L. Secco, P. Costa Jr., F. Gomide, “Rede Neuro-Fuzzy Recorrente Aplicada à Modelagem Térmica de Transformadores de Potência”, *Anais do XV Congresso Brasileiro de Automática, CBA 04*, pp. 1-6, Gramado-RS, Brasil, Setembro 2004.

Capítulo 1

Introdução

1.1 Motivação e Relevância

Modelagem, análise e controle de sistemas dinâmicos têm atraído grande interesse durante as últimas décadas. O estudo dessas áreas cresceu em importância nos últimos anos devido, principalmente, ao aumento do grau de complexidade dos sistemas a serem estudados e o alto nível de automação presente nas mais diversas áreas dos setores produtivos. Estes fatores têm levado à necessidade do desenvolvimento de modelos cada vez mais precisos e robustos a fim de tornar os processos mais confiáveis, reduzindo assim custos operativos (Narendra & Parthasarathy, 1990; Griño et al., 2000).

O campo de utilização desses modelos é muito amplo, destacando-se aplicações como: controle, supervisão, predição, simulação, otimização, etc. Antes dos computadores digitais, os cálculos usados nessas aplicações consumiam muito tempo, e os erros provenientes de simulações eram consideráveis. Hoje em dia, o uso de computadores permite estudar modelos mais detalhados e algoritmos complexos para o projeto e análise de sistemas. Com o aumento do poder computacional, pode-se impor especificações de desempenho mais rígidas nos projetos de engenharia, levando à necessidade de modelos mais detalhados do processo em estudo (Figueiredo, 2000).

Outra área que tem apresentado um grande avanço nos últimos anos é o da inteligência computacional. Em particular, redes neurais artificiais, sistemas nebulosos e suas combinações têm motivado inúmeros trabalhos e uma larga gama de aplicações vêm sendo propostas na literatura. Um dos principais objetivos das pesquisas em inteligência computacional é o de criar sistemas computacionais que emulem características especiais inerentes aos seres humanos, como aprendizado, intuição, raciocínio lógico aproximado, capacidade de generalização, entre outras (Haykin, 1998).

Aplicações de sistemas inteligentes na modelagem de sistemas dinâmicos têm se mostrado altamente promissoras (Jin et al., 1995; Suykens et al., 1996; Tsoukalas & Uhrig, 1997; Giles et al., 1998; Griño et al., 2000; Ouyang et al., 2005). Em especial, a integração entre redes neurais artificiais e

sistemas nebulosos tem se mostrado eficientes quando os problemas exigem soluções robustas, tolerantes a falhas, de dinâmica adaptativa e capazes de considerar dados de natureza vaga ou imprecisa (Gomide & Pedrycz, 2007).

Neste sentido, as redes neurais nebulosas se apresentam como uma poderosa ferramenta na modelagem de sistemas dinâmicos. Estes sistemas computacionais são capazes de especificar parâmetros de um modelo nebuloso e codificar explicitamente a base de regras nebulosas em sua estrutura. Isto implica que o conhecimento adquirido pela rede pode ser facilmente extraído na forma de regras nebulosas do tipo se-então. Além disso, ao contrário das abordagens clássicas, as redes neurais nebulosas prescindem do conhecimento físico do sistema a ser modelado, pois são capazes de gerar o modelo a partir de dados de entrada/saída e, conseqüentemente, de eliminar deficiências causadas por variações de parâmetros (Caminhas et al., 1999).

Com o intuito de se desenvolver redes neurais nebulosas cada vez mais eficientes para aplicação em modelagem nebulosa, novas arquiteturas e métodos de aprendizagem vêm sendo propostos na literatura. Dentre as estruturas pode-se citar as redes neurais nebulosas formadas pelos chamados neurônios lógicos AND e OR propostos por Pedrycz e Rocha em 1993 (Pedrycz & Rocha, 1993).

Estes tipos de neurônio apresentam uma forte semelhança com o modelo clássico de neurônio de McCulloch-Pitts (McCulloch & Pitts, 1943), o que os torna particularmente atrativos para a construção de redes neurais nebulosas que herdem as capacidades de treinamento das redes neurais clássicas. Além disso, esses neurônios fornecem uma forma intuitiva para a extração de regras nebulosas a partir dos modelos treinados, tornando-os lingüisticamente interpretáveis (Figueiredo et al., 2004; Pedrycz, 2006).

Os neurônios lógicos podem ser vistos como uma das formas mais próximas de se representar um neurônio biológico, pois as atividades químicas presentes nas sinapses e sinais elétricos gerados no axônio da célula nervosa real estão, respectivamente, associados ao processamento simbólico e numérico nos neurônios nebulosos (Rocha, 1992). Aliados à mecanismos de treinamento, estes modelos de neurônio possuem ainda a capacidade de simular uma importante característica do neurônio biológico, a chamada “plasticidade sináptica”. Esta característica faz com que um neurônio biológico possa alterar a influência de uma determinada entrada individual no processamento global.

Apesar de existirem outros modelos de neurônio além dos neurônios lógicos que também simulam a plasticidade sináptica, que no jargão da inteligência computacional é chamada de ajuste de pesos, existe uma outra característica inerente aos neurônios biológicos que é pouco considerada na formulação dos modelos de neurônios artificiais propostos até o momento: a chamada “plasticidade neuronal”.

A plasticidade neuronal é definida como sendo a capacidade que um neurônio possui de alterar seus mecanismos internos de processamento em resposta à alterações externas e à necessidade de

adaptação a condições mutantes e a estímulos repetidos (Kandel et al., 2000). De fato, um neurônio biológico pode mudar completamente suas principais características dependendo da necessidade do ambiente. Como exemplo, pode-se citar o caso em que neurônios sensitivos passam a atuar como neurônios motores quando estes últimos sofrem algum tipo de lesão.

Assim, a proposição de um novo modelo de neurônio artificial que herdasse as características dos neurônios nebulosos e ainda fosse capaz de simular a plasticidade neuronal poderia ser de grande valia na construção de modelos de sistemas dinâmicos mais gerais e que seriam capazes de melhor se adaptar a um conjunto de dados de treinamento, se comparados aos modelos existentes na literatura.

É neste contexto que se insere uma das propostas deste trabalho, na qual são apresentadas duas novas classes de neurônios nebulosos em que a introdução de um parâmetro adicional faz com que estes novos neurônios possam variar entre um neurônio lógico do tipo AND até um neurônio lógico do tipo OR de forma a melhor se adaptar às necessidades do problema a ser resolvido. A rede neural nebulosa composta por estes novos modelos de neurônios espelha em sua estrutura um conjunto de regras se-então que usam diferentes conectivos em seus antecedentes. Diferentemente das abordagens propostas até o momento, em que só é possível representar um sistema de inferência nebuloso que utilize apenas um tipo de conectivo nos antecedentes.

Além da nova estrutura de rede neural nebulosa composta por essas novas classes de neurônios nebulosos, este trabalho propõe um novo processo de treinamento inspirado no paradigma do “aprendizado participativo” (Yager, 1990). Esta proposta tem por objetivo reduzir a necessidade que os processos atuais tem de que as bases de dados utilizadas no treinamento sejam extremamente confiáveis no sentido do comportamento que se deseja aprender. Ou seja, nas abordagens atuais, toda a informação contida na base de treinamento é assimilada pelo sistema em treinamento, seja esta informação relevante ou errônea e/ou conflitante. Esta característica gera, na maioria das vezes, comportamentos não desejados, pois informações não relevantes são indiscriminadamente assimiladas pelo sistema.

Desde que para se obter uma base de dados confiável é, provavelmente, um dos principais problemas para os mais diversos sistemas, torna-se necessário a adoção de métodos que dotem o sistema de um certo discernimento a cerca do que deve ser ou não assimilado. Neste sentido, o processo de treinamento aqui proposto pode ser visto como uma forma mais intuitiva de aprendizado, que sob certa ótica captura muitas das características do aprendizado humano. De fato, após uma determinada etapa de seu aprendizado, o ser humano passa a ter convicções próprias e passa a não mais aceitar qualquer tipo de informação como verdade absoluta. Intuitivamente, toda vez que uma nova informação é apresentada a um ser humano, este a compara com tudo aquilo que já foi aprendido a respeito desta nova informação e, caso esta seja muito diferente do conhecimento já adquirido, a tendência é que a informação não seja assimilada.

O paradigma do aprendizado participativo aborda justamente esta característica do aprendizado humano, fornecendo, em linhas gerais, uma forma de aprendizado em que o conhecimento que já se tem a respeito do ambiente também faça parte do processo de treinamento, sendo de crucial importância na determinação do que será ou não assimilado pelo sistema (Hell et al., 2007a).

Apesar de o aprendizado participativo apresentar claros indícios de que sua aplicação seria vantajosa para se melhorar tais características dos processos de treinamento existentes, até o momento nenhum algoritmo eficaz baseado nesse paradigma havia sido proposto na literatura. Neste trabalho, entretanto, um processo de treinamento que utiliza as idéias do aprendizado participativo é proposto para uma classe de redes neurais nebulosas, gerando assim uma nova abordagem para a modelagem nebulosa de sistemas dinâmicos não-lineares.

Vale ressaltar ainda que uma das primeiras propostas deste trabalho foi o de tratar sistemas dinâmicos não-lineares com dinâmica parcial conhecida, ou seja, onde parte da dinâmica do sistema já fosse descrita de forma analítica, em particular, na forma de equações diferenciais. Os modelos gerados representariam um complemento do que o modelo analítico não é capaz de representar, obtendo-se assim um modelo que se aproxima mais do sistema real. Porém, observou-se que, se um determinado modelo é capaz de representar toda a dinâmica de um sistema real, superando os modelos analíticos parciais, e ainda possuindo interpretabilidade lingüística, este pode ser facilmente adaptado a fim de tratar dinâmicas parciais. Desta forma, os modelos propostos foram desenvolvidos para tratar a dinâmica completa dos sistemas reais a serem modelados, tornando assim o trabalho mais geral.

1.2 Objetivos

O objetivo principal deste trabalho é desenvolver um procedimento sistemático para modelagem de sistemas dinâmicos complexos, aliando dois paradigmas da inteligência computacional, redes neurais e sistemas nebulosos, a fim de construir modelos neuro-nebulosos capazes de tratar conhecimento impreciso na modelagem de sistemas dinâmicos não-lineares.

Duas novas abordagens são apresentadas para aplicação em problema de modelagem nebulosa. Na primeira delas, desenvolveu-se um novo processo de treinamento para redes neurais nebulosas baseado no paradigma do aprendizado participativo. O objetivo deste método é o de emular uma das formas de aprendizado do ser humano a fim de que o resultado final do processo não seja tão fortemente influenciado por erros ou dados conflitantes presentes nos dados de treinamento.

Na segunda abordagem, duas novas classes de neurônios nebulosos são propostas com a finalidade de se introduzir modelos de neurônios nebulosos mais gerais dos que os apresentados até o momento na literatura. Uma vez que estes novos neurônios apresentam uma importante característica dos neurônios biológicos chamada plasticidade neuronal, eles são capazes de modificar seu proces-

samento interno de acordo com a necessidade do problema a ser resolvido. Isto faz com que o comportamento de uma rede neural nebulosa formada por estes novos tipos de neurônio possa melhor se aderir à uma base de dados, fornecendo modelos de sistemas dinâmicos mais bem adaptados aos dados.

Um resultado que pode ser associado a esta segunda abordagem diz respeito à interpretação lingüística da rede neural nebulosa composta por esses novos neurônios. As estruturas de redes neurais nebulosas propostas na literatura podem ser interpretadas como sistemas de inferência nebulosos que utilizam apenas um tipo de conectivo no antecedente das regras nebulosas. Com esta nova abordagem, o sistema de inferência nebuloso embutido na rede neural nebulosa gerada ao final do processo de treinamento poderá utilizar mais de um tipo de conectivo no antecedente das regras, sendo que a escolha destes conectivos é feita pelo próprio processo de treinamento, dependendo da necessidade do problema a ser resolvido.

Por fim, este trabalho tem por objetivo mostrar que as abordagens propostas são promissoras na modelagem de sistemas dinâmicos não-lineares, demonstrando experimentalmente que estas abordagens superam tanto em termos de precisão quanto em termos de custo computacional todas as abordagens alternativas testadas para o mesmo fim.

1.3 Organização do Trabalho

Neste primeiro capítulo foi apresentado o contexto no qual o trabalho está inserido, seus aspectos relevantes, os problemas que motivaram sua proposição bem como um resumo dos principais objetivos e dos modelos que estão sendo propostos. O capítulo 2 fornece alguns fundamentos teóricos com a finalidade de facilitar a compreensão das propostas aqui apresentadas. Além disso, é apresentada uma classificação dos modelos de sistemas dinâmicos, a fim de posicionar o trabalho dentre as diversas abordagens existentes. No capítulo 3, o novo processo de treinamento baseado no aprendizado participativo é apresentado. Propriedades relativas à convergência deste novo método são discutidas e comprovadas por meio de teoremas. Duas novas classes de neurônios nebulosos são introduzidas no Capítulo 4. Descrevem-se também estruturas e métodos de treinamento para redes neurais nebulosas compostas por esses novos neurônios nebulosos. No capítulo seguinte, todos os modelos nebulosos propostos neste trabalho são aplicados na modelagem de sistemas dinâmicos não-lineares a fim de comprovar sua eficácia. Comparações com metodologias alternativas propostas na literatura são também apresentadas. O Capítulo 6 conclui o trabalho, apresentando sua principais contribuições e fornecendo sugestões para pesquisas futuras.

Capítulo 2

Fundamentos Teóricos

2.1 Introdução

Este capítulo tem por objetivo apresentar os principais fundamentos teóricos que serviram de base para as técnicas e algoritmos desenvolvidos neste trabalho. Dentre estes, fundamentos pode-se ressaltar as formas clássicas mais usadas para representar sistemas dinâmicos lineares e não-lineares, as principais maneiras existentes para se determinar os modelos com seus parâmetros e estruturas e as principais características de alguns métodos de identificação de sistemas dinâmicos.

Outro tópico abordado neste capítulo se refere aos métodos alternativos de se representar sistemas dinâmicos, em especial voltados à teoria de conjuntos nebulosos, redes neurais nebulosas bem como paradigmas de aprendizado aplicados a tais sistemas.

2.2 Modelagem Matemática

Modelos matemáticos são representações analíticas que têm o propósito de aproximar as características observadas de um dado sistema real. Alguns dos objetivos para se construir um modelo são aumentar o conhecimento sobre um sistema, projetar um controlador, simular cenários ou realizar previsões sobre o comportamento de um sistema. Aplicações diversas requerem modelos em formatos diferenciados. Modelos físicos, como uma maquete, são úteis na visualização de como os componentes físicos do sistema interagem entre si. Gráficos são outras formas de modelos: eles podem apresentar relações de causa e efeito de maneira concisa e intuitiva. Porém, o tipo de modelo mais comum em engenharia é o modelo matemático. Modelos matemáticos podem ser obtidos por métodos que vão desde a análise das leis físicas que se aplicam ao sistema até uma abordagem puramente numérica, baseada em um conjunto de dados experimentais. A escolha do tipo do modelo depende do propósito definido para o mesmo e do tipo e quantidade de informação disponível sobre

o sistema.

No contexto deste trabalho, um modelo é entendido como uma representação de um sistema dinâmico. Como será visto na seqüência, um modelo pode ser definido ou classificado de diversas outras formas além da matemática. A próxima seção apresenta de forma resumida uma classificação de modelos matemáticos.

2.2.1 Classificações de um Modelo Matemático

Muitas são as classificações para modelos matemáticos, não existindo um padrão unanimemente aceito (Dorf & Bishop, 1998). Nesta seção, serão descritas apenas as classificações mais relevantes para este trabalho. Para tanto, considere a seguinte expressão:

$$y = f(u) \tag{2.1}$$

onde u representa a entrada do modelo, y sua saída e $f(\cdot)$ a função que descreve o comportamento do sistema. Geralmente, as entradas e saídas de um sistema são denotadas por $u_1(t), u_2(t), \dots, u_n(t)$ e $y_1(t), y_2(t), \dots, y_m(t)$, respectivamente. No domínio do tempo discreto, utiliza-se normalmente o índice k (como $u(k)$ e $y(k)$).

Assim sendo, primeiramente pode-se distinguir um modelo matemático quanto à forma com que se relacionam as entradas e saída do modelo. Neste contexto, o modelo pode ser classificado como *Linear* ou *Não-Linear*. Um modelo linear é aquele que atende ao princípio da sobreposição de efeitos, isto é, se $y_1(t) = f(u_1(t))$ e $y_2(t) = f(u_2(t))$ então uma entrada do tipo $au_1(t) + bu_2(t)$ produzirá uma saída da forma $ay_1(t) + by_2(t)$, onde a e b são constantes. Qualquer relação ou modelo que não satisfaça essa condição é não-linear.

A importância da linearidade em um modelo reside nas simplificações obtidas na sua análise. Além disso, é possível em um modelo linear isolar os efeitos das variáveis de entrada e estudá-las separadamente. No entanto, a maioria dos sistemas reais é não-linear, caso não sejam feitas restrições quanto aos valores de suas entradas. A análise em determinadas faixas de operação pode vir a fornecer certo grau de linearidade ao modelo. Porém, dependendo da aplicação, a consideração da natureza não-linear do sistema pode ser essencial nas tarefas realizadas, não podendo ser descartada do modelo. Os modelos de interesse nesse trabalho são não-lineares.

Uma segunda forma de classificar modelos é baseada na característica da variável independente temporal. Um modelo *contínuo* é aquele no qual as variáveis são contínuas no tempo, ou seja, é possível acompanhar o sistema (e nele atuar) em qualquer instante de tempo. No caso de sistemas dinâmicos, esses modelos são geralmente descritos por equações diferenciais. Em contrapartida, os modelos *discretos* são aqueles nos quais as variáveis do sistema evoluem em instantes específicos do

tempo, agora discretizado. Os instantes de tempo nos quais as variáveis podem, por exemplo, alterar seus valores correspondem ao momento no qual alguma amostragem é realizada ou a memória de um computador é lida. Modelos discretos são representados por equações a diferenças. Em engenharia, o fator mais sugestivo para o uso de modelos discretos é a disponibilidade de um computador digital para processamento de dados. Os modelos abordados neste trabalho são discretos.

Finalmente, algumas vezes há incerteza de natureza probabilística no valor de algum parâmetro do modelo. Se essa incerteza for significativa, um modelo *estocástico* deve ser usado. Nesse modelo, algumas variáveis são descritas em termos de uma função de distribuição de probabilidade, envolvendo, por exemplo, suas médias e variâncias. Os modelos *determinísticos*, por outro lado, não representam possíveis aleatoriedades presentes no sistema real. Assim, a saída em um dado instante para um modelo determinístico é completamente definida pelos parâmetros do modelo e por suas entradas. A característica de determinismo está presente nos modelos desenvolvidos no presente trabalho, os quais contemplam apenas incertezas de natureza nebulosa.

Além destas características principais, pode-se ainda classificar um modelo como sendo a parâmetros distribuídos ou concentrados, variante ou invariante no tempo, autônomo ou não autônomo e não-paramétrico ou paramétrico. Os modelos propostos neste trabalho são caracterizados, em todos os pares anteriores, pelas segundas propriedades.

2.2.2 Abordagens Baseadas em Inteligência Computacional

O uso de técnicas de inteligência computacional (IC) na modelagem de sistemas dinâmicos, em especial para sistemas não-lineares, vem sendo largamente difundido na literatura nos últimos anos. Este interesse é motivado principalmente pelas características destas técnicas que, permitem a construção de modelos que são aproximadores universais de funções. De fato, dependendo da técnica utilizada, é possível aproximar com precisão arbitrária qualquer função não-linear contínua definida em uma região compacta (limitada e fechada) (Cybenko, 1989; Wang, 1998) tomando-se como base tanto informações quantitativas quanto qualitativas.

Dentre as técnicas de IC utilizadas na modelagem de sistemas dinâmicos, pode-se ressaltar as redes neurais artificiais (RNA's), a teoria de conjuntos nebulosos (*fuzzy*) e suas combinações.

O conceito de RNA's foi introduzido por McCulloch e Pitts em 1943 (McCulloch & Pitts, 1943), e propunha uma simplificação do que se sabia a respeito do neurônio biológico naquela época. Porém, só em 1958 foi introduzido o conceito de redes neurais (Rosenblatt, 1958). O modelo proposto por Rosenblatt, conhecido como *multilayer perceptron* (MLP), era composto por uma estrutura de rede em camadas tendo como unidades básicas neurônios artificiais do tipo McCulloch e Pitts e uma regra de aprendizado. Hoje em dia algoritmos especiais de treinamento são usados no treinamento de redes neurais MLP. O mais conhecido destes algoritmos é o algoritmo de retropropagação, conhecido como

algoritmo *backpropagation*, que se baseia no conceito do gradiente descendente (Haykin, 1998).

A capacidade de representação das redes neurais está relacionada à sua arquitetura. As redes MLP de uma camada resolvem apenas problemas linearmente separáveis. Para problemas não-linearmente separáveis, deve-se usar redes com uma ou mais camadas intermediárias. Com uma camada intermediária uma rede neural pode implementar qualquer função contínua em um domínio compacto (Cybenko, 1989).

No contexto de modelagem de sistemas dinâmicos, já foram estudadas diversas arquiteturas para redes neurais. Em particular, é possível estender os modelos FIR, ARX e ARMAX para os modelos NNFIR, NNARX e NNARMAX, sendo NN o acrônimo em inglês para rede neural (*neural network*). Esses modelos traduzem o fato de as entradas da rede neural serem compostas pelos termos regressores dos respectivos modelos lineares (Norgaard et al., 2000).

A teoria de conjuntos nebulosos é outra técnica de inteligência computacional que pode ser aplicada na modelagem de sistemas dinâmicos. Uma série de arquiteturas baseadas nesta teoria já foram demonstradas como aproximadores universais (Gomide & Pedrycz, 2007; Wang, 1998). Isto significa que são capazes de aproximar qualquer função contínua em um domínio compacto com qualquer nível de precisão desejado.

Além da propriedade de aproximação universal dos modelos nebulosos, estes ainda adicionam uma nova dimensão à informação contida no modelo. Trata-se da dimensão lingüística, que fornece descrições intuitivas sobre o comportamento do sistema modelado (Espinosa et al., 2004). Isto faz com que modelos nebulosos possam tratar tanto informações de natureza quantitativa como qualitativa, podendo a modelagem ser feita por meio de informações obtidas a partir do conhecimento que se tem a respeito do sistema e/ou de informações extraídas diretamente dos dados do processo.

Os modelos nebulosos podem ser divididos em três classes principais (Babuska & Verbruggen, 1996):

- Modelos Nebulosos Lingüísticos;
- Modelos Nebulosos Relacionais;
- Modelos Nebulosos Funcionais.

Os modelos nebulosos lingüísticos são aqueles que utilizam uma base de regras se-então e um método de inferência para relacionar entradas e saídas. Este modelo é denominado lingüístico por fazer uso direto da representação lingüística das regras (Babuska & Verbruggen, 1996; Ljung, 1999). Por sua vez, os modelos nebulosos relacionais, como o próprio nome já diz, representam o mapeamento entre os conjuntos nebulosos de entrada e saída pelas relações nebulosas (Gomide & Pedrycz, 2007). E por fim, os modelos nebulosos funcionais (Takagi-Sugeno) são modelos com termos lingüísticos no antecedente e uma função das entradas no conseqüente de uma regra nebulosa.

Ambas as técnicas de IC mencionadas até o momento possuem vantagens e desvantagens na modelagem de sistemas dinâmicos. De um lado, as RNA's falham em representar o conhecimento de forma explícita e no tratamento de informação imprecisa, porém apresentam características de aprendizado. Por outro lado, os sistemas nebulosos baseados em regras se-então são limitados no que diz respeito à aquisição automática de conhecimento, porém valem-se de suas características para o tratamento e manipulação de termos lingüísticos e de informação imprecisa.

A partir desta constatação, observou-se que redes neurais e sistemas nebulosos poderiam ser unificados pois, além destas abordagens exibirem características comuns, cada uma apresenta propriedades exclusivas que contribuem para compensar as deficiências individuais da outra (Kosko, 1991). O resultado desta simbiose pode ser verificado em duas grandes classes de modelos: os **sistemas neuro-nebulosos** e as **redes neurais nebulosas** (Lin & Lee, 1996).

Os sistemas neuro-nebulosos são aqueles em que a habilidade de aprendizado das RNA's é empregada no ajuste dos parâmetros de um sistema de inferência nebuloso. Um exemplo clássico desta classe de modelos é o sistema de inferência neuro-adaptativa, ou ANFIS¹, proposto por Jang em 1993 (Jang, 1993). Nele, um algoritmo baseado no gradiente descendente, similar ao *backpropagation* é usado para ajustar a forma das funções de pertinência dos conjuntos nebulosos que granularizam o espaço de entrada, bem como o ajuste das funções de saída de um modelo nebuloso funcional do tipo Takagi-Sugeno.

Já nas redes neurais nebulosas, utiliza-se a estrutura de uma RNA, onde os neurônios artificiais clássicos são substituídos por neurônios nebulosos (Pedrycz, 1993). Estes neurônios são implementados por meio de normas e co-normas triangulares, que generalizam as operações de união e interseção de conjuntos clássicos para a teoria de conjuntos nebulosos. Assim, a rede neural passa a ser vista como um sistema lingüístico, preservando a capacidade de aprendizado da RNA. Exemplos destas redes podem ser encontradas em (Figueiredo et al., 2004) e (Lin & Lee, 1996).

Em particular, os modelos desenvolvidos neste trabalho se encaixam neste último grupo, o das redes neurais nebulosas. Assim sendo, para apresentarmos as estruturas e algoritmos desenvolvidos neste trabalho, é necessário apresentar alguns conceitos sobre teoria de conjuntos nebulosos e sistemas nebulosos. Esta tarefa fica a cargo das próximas seções.

2.3 Conceitos de Teoria de Conjuntos Nebulosos

Conforme mencionado anteriormente, com a finalidade de melhor compreender os algoritmos e estruturas propostas neste trabalho, é necessário apresentar alguns conceitos sobre conjuntos nebulosos e teoria de sistemas nebulosos, conforme será feito a seguir.

¹do inglês *Adaptive NeuroFuzzy Inference System*

2.3.1 Conjuntos Nebulosos

A noção de conjunto nebuloso foi introduzida por Lotfi Zadeh em meados da década de 60 (Zadeh, 1965). Intuitivamente, um conjunto nebuloso é um conjunto cujos limites não são bem definidos, ou seja, um conjunto onde a transição entre pertencer e não pertencer ao conjunto é gradual ao invés de abrupta. Mais precisamente, um conjunto nebuloso F em um universo de discurso X é caracterizado por uma *função de pertinência* F , que associa a cada elemento $x \in X$ um grau $F(x) \in [0, 1]$.

Vale ressaltar que um conjunto clássico A em X é um caso particular de um conjunto nebuloso, onde os graus de associação (ou graus de pertinência) são:

$$A(x) = \begin{cases} 0 & \text{se } x \notin A \\ 1 & \text{se } x \in A \end{cases}.$$

No caso de conjuntos clássicos, $A(x)$ é a *função característica* de A . Assim sendo, um conjunto nebuloso pode ser formalmente definido como se segue.

Definição 2.1 (Conjunto nebuloso) *Seja X um universo de discurso, ou seja, uma coleção discreta ou contínua de objetos x . Um conjunto nebuloso F em X é um conjunto de pares ordenados da forma:*

$$F = \{(x, F(x)) | x \in X\} \quad (2.2)$$

onde $F(x)$ é o grau de pertinência do elemento x no conjunto nebuloso F e F mapeia um elemento x de X no intervalo $[0, 1]$, isto é, $F : X \rightarrow [0, 1]$.

$F(x)$ é a função de pertinência de F . Note que funções de pertinência para conjuntos nebulosos generalizam a idéia de funções características para conjuntos clássicos. Se o universo $X = \{x_1, x_2, \dots, x_n\}$ é finito, então F pode ser representada por um vetor n -dimensional $F = (f_1, f_2, \dots, \dots, f_n)$, com $f_i = F(x_i)$. A Figura 2.1 fornece um exemplo de um conjunto nebuloso no qual a função de pertinência captura o conceito de um número inteiro *próximo a 55*. Neste exemplo o conjunto nebuloso F modela a noção linguisticamente expressa de uma quantidade inteira próxima de 55 em um universo finito de números inteiros $X = \{15, 25, \dots, 85, 95\}$. Assim $n = 9$, $F = \{(15, 0), (25, 0.0111), (35, 0.1353), (45, 0.6065), (55, 1.0), (65, 0.6065), (75, 0.1353), (85, 0.0111), (95, 0)\}$ e, usando a notação vetorial equivalente, F é expresso por $F = (0, 0.0111, 0.1353, 0.6065, 1.0, 0.6065, 0.1353, 0.0111, 0)$.

2.3.2 Normas Triangulares

Norma triangulares (t-normas) e co-normas triangulares (t-conormas ou s-normas) são operações que generalizam a interseção e união para a teoria de conjuntos nebulosos. Na teoria de conjuntos

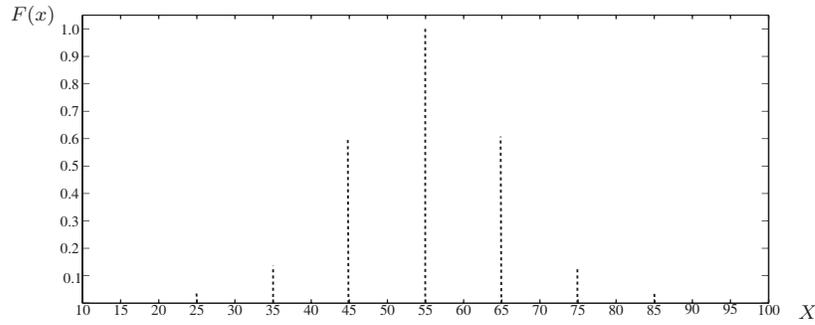


Figura 2.1: Conjunto nebuloso F em um universo de discurso X

clássicos, a interseção e a união são executadas, em termos de suas funções características, utilizando-se os operadores \min e \max , respectivamente. Estas normas também fornecem um modelo operacional para os conectivos lógicos E (*and*) e OU (*or*) na lógica proposicional clássica.

Normas triangulares estendem as operações padrão de conjuntos clássicos para conjuntos nebulosos, e são definidas da seguinte forma:

Definição 2.2 (t-norma) *Uma norma triangular é uma operação binária t no intervalo $[0, 1]$ que é comutativa, associativa, monotônica e que apresenta 1 como elemento neutro, ou seja, é uma função $t : [0, 1] \times [0, 1] \rightarrow [0, 1]$ tal que, para todo $x, y, z \in [0, 1]$:*

1. $x t y = y t x$ (*comutatividade*),
2. $x t (y t z) = (x t y) t z$ (*associatividade*),
3. $x t y \leq x t z$ quando $y \leq z$ (*monotonicidade*),
4. $x t 1 = x$ (*elemento neutro 1*).

Uma notação alternativa para t-normas é dada por $T(x, y) = xty$.

Definição 2.3 (s-norma) *Uma co-norma triangular (ou s-norma) é uma operação binária s no intervalo $[0, 1]$ que é comutativa, associativa, monotônica e que apresenta 0 como elemento neutro, ou seja, é uma função $s : [0, 1] \times [0, 1] \rightarrow [0, 1]$ tal que para todo $x, y, z \in [0, 1]$:*

1. $x s y = y s x$ (*comutatividade*),
2. $x s (y s z) = (x s y) s z$ (*associatividade*),
3. $x s y \leq x s z$ quando $y \leq z$ (*monotonicidade*),

4. $x \text{ s } 0 = x$ (*elemento neutro 0*).

Uma notação alternativa para s-normas é dada por $S(x, y) = xsy$. Diversos exemplos de t-normas e s-normas podem ser encontrados em (Gomide & Pedrycz, 2007).

2.3.3 Uninormas e Nullnormas

Uninormas e nullnormas representam uma forma de generalizar t-normas e s-normas, relaxando as restrições relacionadas à seus elementos neutros, ou seja, ao invés do elemento neutro ser restrito a 1 (t-norma) ou 0 (s-norma), permite-se que este elemento assumira qualquer valor no intervalo unitário. Para enfatizar esta característica, o elemento neutro passa a ser chamado de *elemento identidade* para as uninormas e de *elemento de absorção* para as nullnormas. Formalmente, uninormas e nullnormas podem ser definidas da seguinte forma (Calvo et al., 2001):

Definição 2.4 (Uninorma) *Uma uninorma é um operador binário $U : [0, 1] \times [0, 1] \rightarrow [0, 1]$ que satisfaz as seguintes propriedades:*

$$\text{Comutatividade } U(x, y; g) = U(y, x; g)$$

$$\text{Monotonicidade } U(x, y; g) \geq U(z, v; g) \text{ para } x > z \text{ e } y > v$$

$$\text{Associatividade } U(x, U(y, z; g); g) = U(U(x, y; g), z; g)$$

$$\text{Elemento Neutro } \exists \text{ algum } g \in [0, 1] \text{ tal que } U(x, g; g) = x \quad \forall x \in [0, 1]$$

Note que uma uninorma é, de fato, uma generalização dos conceitos de t-normas e s-normas. Conforme mencionado anteriormente, uma uninorma difere das t-normas e s-normas apenas no que diz respeito ao elemento neutro. Para t-normas e s-normas, tem-se valores fixos para os elementos neutros, enquanto, para as uninormas, este valor pode ser escolhido no intervalo $[0, 1]$. Desta forma, a característica de uma uninorma pode variar suavemente de um *and* puro para um *or* puro quando o elemento identidade varia de 1 para 0. Obviamente, para o caso em que $g = 1$ a uninorma U é uma t-norma enquanto que para o caso em que $g = 0$ a uninorma U é uma s-norma.

Na literatura, é possível encontrar uma grande variedade de realizações de uninormas (Yager & Rybalov, 1996; Yager, 2001). Uma das realizações mais usadas está vinculada à seguinte família de construtores:

$$U(x, y; g) = \begin{cases} g \times T\left(\frac{x}{g}, \frac{y}{g}\right), & \text{se } x, y \in [0, g] \\ g + (1 - g) \times S\left(\frac{x-g}{1-g}, \frac{y-g}{1-g}\right), & \text{se } x, y \in [g, 1] \\ \min(x, y) \text{ ou } \max(x, y), & \text{caso contrário.} \end{cases} \quad (2.3)$$

onde t e s são normas triangulares. Esta construção não requer nenhum tipo de dualidade entre as t-normas e s-normas específicas. A Figura 2.2 mostra uma representação gráfica para as duas variações fundamentais da uninorma apresentada em (2.3), baseadas nos operadores \max e \min .

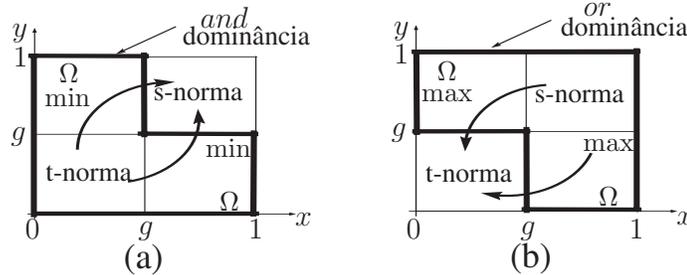


Figura 2.2: Realizações alternativas de uma uninorma: (a) *and*-dominada, (b) *or*-dominada

É possível verificar em (2.3) que realizações específicas por meio da escolha dos operadores máximo e mínimo fornecem um certo tipo de flexibilidade para as regiões intermediárias (denotadas por Ω na Figura 2.2). Conforme discutido em (Pedrycz, 2006), a uninorma U pode ser *and*-dominada se for escolhido o operador mínimo (maior das t-normas) para as regiões intermediárias Ω , ou então *or*-dominada, caso seja escolhido o operador máximo (menor das s-normas) para estas regiões.

Definição 2.5 (Nullnorma) *Uma nullnorma é um operador binário $N : [0, 1] \times [0, 1] \rightarrow [0, 1]$ que é comutativo, monotônico, associativo, e tal que existe um elemento $u \in [0, 1]$ chamado elemento de absorção, que satisfaz $N(u, x; u) = x$, para todo $x \in [0, 1]$ e também:*

$$\begin{cases} N(0, x; u) = x, & \forall x \leq u \\ N(1, x; u) = x, & \forall x \geq u \end{cases}$$

Da mesma forma que para as uninormas, o elemento de absorção u correspondente a uma nullnorma N é único. Note que uma nullnorma também é uma generalização de t-normas e s-normas, pois, por definição, quando $u = 0$ a função N é uma t-norma, enquanto que $u = 1$ faz de N uma s-norma.

Uma realização específica de nullnorma é apresentada em (Calvo et al., 2001), e pode ser traduzida na seguinte família de construtores:

$$N(x, y; u) = \begin{cases} u \times S\left(\frac{x}{u}, \frac{y}{u}\right), & \text{se } x, y \in [0, u] \\ u + (1 - u) \times T\left(\frac{x-u}{1-u}, \frac{y-u}{1-u}\right), & \text{se } x, y \in [u, 1] \\ u, & \text{caso contrário.} \end{cases} \quad (2.4)$$

Note que (2.4) apresenta grande similaridade com (2.3) e também não requer nenhum tipo de relação de dualidade entre as t-normas e s-normas usadas nesta construção.

2.3.4 Operador de Compromisso

Alguns dos temas deste trabalho empregam em seu desenvolvimento o conceito de *Operador de Compromisso*. Este conceito é definido da seguinte forma (Klement et al., 2000):

Definição 2.6 (Operador de Compromisso) Uma função $\tilde{C}_u : [0, 1] \rightarrow [0, 1]$ dada por

$$\begin{aligned}\tilde{C}_u(a) &= (1 - u)C(a) + uC(C(a)) \\ &= (1 - u)C(a) + ua\end{aligned}\tag{2.5}$$

é chamada de operador de compromisso, onde $u \in [0, 1]$ e $\tilde{C}_0(a) = C(a) = 1 - a$.

Note que $\tilde{C}_{1-u}(a) = \tilde{C}_u(1 - a) = 1 - \tilde{C}_u(a)$ e

$$\tilde{C}_u(a) = \begin{cases} C(a) & \text{para } u = 0 \\ \frac{1}{2} & \text{para } u = \frac{1}{2} \\ a & \text{para } u = 1 \end{cases}\tag{2.6}$$

A função \tilde{C} é uma negação forte para $u = 0$ (Gomide & Pedrycz, 2007).

2.3.5 Relações Nebulosas

Uma relação nebulosa R expande a noção de conjuntos nebulosos para um universo multi-dimensional e representa a noção de associação parcial entre os elementos dos universos. Por exemplo, um grau de pertinência $R(x, y)$ representa o grau de associação entre o elemento $x \in X$ e $y \in Y$ (Gomide & Pedrycz, 2007). Formalmente, para um universo bidimensional, uma relação nebulosa pode ser definida da seguinte forma:

Definição 2.7 (Relação Nebulosa) Sejam X e Y dois universos. Uma relação nebulosa R é um subconjunto nebuloso do produto cartesiano $X \times Y$

$$R : X \times Y \rightarrow [0, 1]$$

onde $x \in X$, $y \in Y$ e $R(x, y)$ é o grau com que (x, y) são compatíveis com a relação R .

Para exemplificar o conceito de relação nebulosa, considere d_{fs} , d_{nf} , d_{ns} , d_{gf} como sendo artigos científicos cujo assunto principal são sistemas nebulosos, sistemas neuro-nebulosos, sistemas neurais e algoritmos genéticos, com palavras-chave w_f , w_n e w_g , respectivamente. Assim sendo, a relação R

em $D \times W$, $D = \{d_{fs}, d_{nf}, d_{ns}, d_{gf}\}$ e $W = \{w_f, w_n, w_g\}$ pode, por exemplo, assumir a seguinte forma matricial:

$$R = \begin{bmatrix} 1 & 0 & 0.6 \\ 0.8 & 1 & 0 \\ 0 & 1 & 0 \\ 0.8 & 0 & 1 \end{bmatrix}$$

Uma vez que os universos D and W são finitos e discretos, R pode ser representada por uma matriz 4×3 (4 artigos e 3 palavras-chave). Neste exemplo $R(d_{fs}, w_f) = 1$ significa que o conteúdo do artigo d_{fs} é totalmente compatível com a palavra-chave w_f enquanto que $R(d_{fs}, w_n) = 0$ e $R(d_{fs}, w_g) = 0.6$ indicam que o artigo d_{fs} não trata de sistemas neurais, mas parte de seu conteúdo está relacionado com algoritmos genéticos.

Relações entre variáveis podem ser definidas usando *regras nebulosas* na forma:

$$SE \ x_1 \text{ é } F_1 \ E \ x_2 \text{ é } F_2 \ E \ \dots \ E \ x_n \text{ é } F_n \ ENTÃO, \ z \text{ é } Z_\ell. \quad (2.7)$$

Regras nebulosas definem uma relação entre antecedentes (x_1, x_2, \dots, x_n) e conseqüentes (z) em termos dos conjuntos nebulosos F_1, F_2, \dots, F_n e Z_ℓ . Uma regra nebulosa (2.7) é um exemplo de relação nebulosa da forma $R : F_1 \times F_2 \times \dots \times F_n \times Z_\ell \rightarrow [0, 1]$, onde $R(x_1, x_2, \dots, z)$ representa o grau de associação entre as variáveis antecedentes e conseqüentes.

2.3.6 Composição de Relações Nebulosas

Relações nebulosas podem ser compostas usando-se de diferentes operadores da teoria de conjuntos. De fato, diferentes famílias de operadores podem ser utilizados para compor relações nebulosas, dependendo da escolha de algumas t-normas e s-normas específicas.

Os dois mais importantes tipos de composição de relações nebulosas são as composições do tipo sup-t e inf-s. Para definir estes dois tipos de composição, considere X, Y e Z como universos distintos. Assim sendo, pode-se apresentar as seguintes definições:

Definição 2.8 (Composição sup-t) A composição sup-t das relações nebulosas $G : X \times Z \rightarrow [0, 1]$ e $W : Z \times Y \rightarrow [0, 1]$ é uma relação nebulosa $R : X \times Y \rightarrow [0, 1]$ cuja função de pertinência $R(x, y)$ é dada por:

$$R(x, y) = \sup_{z \in Z} \{G(x, z) \ t \ W(z, y)\} \quad \forall x \in X \text{ e } \forall y \in Y \quad (2.8)$$

onde t é uma t-norma. A composição sup-t de G e W é denotada simbolicamente por $R = G \circ W$.

Como exemplo, considere as relações nebulosas G e W dadas pelas matrizes relacionais (3×4)

e (4×2) apresentadas a seguir:

$$G = \begin{bmatrix} 1.0 & 0.6 & 0.5 & 0.5 \\ 0.6 & 0.8 & 1.0 & 0.2 \\ 0.8 & 0.3 & 0.4 & 0.3 \end{bmatrix} \quad W = \begin{bmatrix} 0.6 & 0.1 \\ 0.5 & 0.7 \\ 0.7 & 0.8 \\ 0.3 & 0.6 \end{bmatrix} \quad (2.9)$$

Neste caso, como se trata de universos finitos, o procedimento de se obter a composição das relações nebulosas G e W é similar ao processo de multiplicação de matrizes, apenas substituindo-se o somatório pela função máximo (uma vez que os universos são finitos, encontrar o máximo significa também encontrar o supremo) e o produto algébrico pela t-norma correspondente. Também de forma similar à multiplicação de matrizes, a composição é definida apenas quando o número de colunas da primeira relação nebulosa é o mesmo que o número de linhas da segunda relação. Assim, se G é uma relação $m \times n$ e W é uma relação $n \times p$, a composição destas duas relações R será uma relação $m \times p$.

Por exemplo, tomando-se (2.8) e escolhendo o mínimo como t-norma, a composição sup-min $R = G \circ W$, $R = [r_{ij}]$ é calculada da seguinte forma:

$$\begin{aligned} r_{11} &= \sup(1.0 \ t \ 0.6, 0.6 \ t \ 0.5, 0.5 \ t \ 0.7, 0.5 \ t \ 0.3) = 0.6 \\ r_{21} &= \sup(0.6 \ t \ 0.6, 0.8 \ t \ 0.5, 1.0 \ t \ 0.7, 0.2 \ t \ 0.3) = 0.7 \\ &\dots \\ r_{32} &= \sup(0.1 \ t \ 0.8, 0.3 \ t \ 0.7, 0.4 \ t \ 0.8, 0.3 \ t \ 0.6) = 0.4 \end{aligned}$$

$$R = \begin{bmatrix} 0.6 & 0.6 \\ 0.7 & 0.8 \\ 0.6 & 0.4 \end{bmatrix}$$

Definição 2.9 (Composição inf-s) A composição inf-s das relações nebulosas $G : X \times Z \rightarrow [0, 1]$ e $W : Z \times Y \rightarrow [0, 1]$ é uma relação nebulosa $R : X \times Y \rightarrow [0, 1]$ cuja função de pertinência $R(x, y)$ é dada por:

$$R(x, y) = \inf_{z \in Z} \{G(x, z) \ s \ W(z, y)\} \quad \forall x \in X \text{ and } \forall y \in Y \quad (2.10)$$

onde s é uma s-norma. A composição inf-s de G e W é denotada simbolicamente por $R = G \bullet W$.

O procedimento para se obter a composição inf-s é o mesmo daquele utilizado para a composição sup-t, substituindo-se o supremo pelo ínfimo e a t-norma por uma s-norma. Assim sendo, usando-se (2.10) e escolhendo-se o máximo como s-norma, a composição inf-s $R = G \bullet W$, $R = [r_{ij}]$ para as

relções G e W definidas em (2.9) é obtida da seguinte forma:

$$\begin{aligned} r_{11} &= \inf(1.0 \text{ s } 0.6, 0.6 \text{ s } 0.5, 0.5 \text{ s } 0.7, 0.5 \text{ s } 0.3) = 0.5 \\ r_{21} &= \inf(0.6 \text{ s } 0.6, 0.8 \text{ s } 0.5, 1.0 \text{ s } 0.7, 0.2 \text{ s } 0.3) = 0.3 \\ &\dots \\ r_{32} &= \inf(0.1 \text{ s } 0.8, 0.3 \text{ s } 0.7, 0.4 \text{ s } 0.8, 0.3 \text{ s } 0.6) = 0.6 \end{aligned}$$

$$R = \begin{bmatrix} 0.5 & 0.6 \\ 0.3 & 0.6 \\ 0.3 & 0.6 \end{bmatrix}$$

2.3.7 Equações Relacionais Nebulosas

Equações relacionais nebulosas estão relacionadas com a noção de composição nebulosa. De fato, uma relação nebulosa R em $X \times Y$ pode ser vista como um modelo de um sistema nebuloso cuja entrada é um conjunto nebuloso F em X e cuja saída é um outro conjunto nebuloso H em Y (Figura 2.3). A relação nebulosa R descreve a relação de dependência entre as entradas e saídas do sistema (Gomide & Pedrycz, 2007).

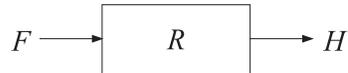


Figura 2.3: Sistema relacional nebuloso de uma entrada - uma saída

Considere o seguinte modelo relacional nebuloso:

$$H = F \diamond R \quad (2.11)$$

onde o operador \diamond representa ou uma composição sup-t ou uma composição inf-s. Para uma dada entrada F e uma relação R , a saída H é encontrada utilizando-se a definição da composição escolhida. Neste ponto, dois problemas fundamentais podem ser levantados com relação às equações relacionais nebulosas:

P1. Dados F e H , determine R (*Problema de Estimação*);

P2. Dados H e R , determine F (*Problema Inverso*);

Aqui será abordada apenas a solução para o problema de estimação, pois somente ele será utilizado na construção dos algoritmos que fazem parte deste trabalho. Para a solução do problema inverso ver (Gomide & Pedrycz, 2007).

2.3.8 Solução do Problema de Estimação (SPE)

Antes de introduzir a solução para o problema de estimação, é necessário primeiro verificar a existência de solução (ou soluções), uma vez que, para alguns casos, este problema pode não ter solução.

A condição de existência de solução para o problema de estimação depende do tipo de composição escolhido. A seguir, são apresentadas as condições e as soluções para os dois tipos de composição discutidos anteriormente: sup-t e inf-s.

SPE para equações relacionais com composição sup-t

Especificamente falando, o conjunto nebuloso de saída H de uma equação relacional nebulosa com composição sup-t $H = F \circ R$ é calculado da seguinte forma:

$$H(y) = \sup_{x \in X} \{F(x) \text{ t } R(x, y)\} \quad (2.12)$$

Uma condição para a existência de uma solução para a equação relacional apresentada em (2.12) é dada pelo seguinte Teorema (Lin & Lee, 1996):

Teorema 2.1 *O Problema de Estimação (P1) para equações relacionais com composição do tipo sup-t possui solução (ou soluções) se e somente se a altura² do conjunto nebuloso F for maior ou igual à altura do conjunto nebuloso H , isto é:*

$$\sup_{x \in X} \{F(x)\} \geq H(y) \quad \text{para todo } y \in Y \quad (2.13)$$

Demonstração: *Se a expressão (2.13) não for válida, tem-se que:*

$$\sup_{x \in X} \{F(x)\} < H(y) \quad \text{para algum } y \in Y \quad (2.14)$$

É possível verificar que, se a expressão (2.14) for verdadeira, torna-se claramente impossível obter uma solução para a equação relacional em (2.12). Uma vez que não existe solução para a equação relacional (2.12), o problema P1 também é infactível.

Por outro lado, assumamos que a expressão (2.13) é válida. Considere $x^ \in X$ como sendo o elemento para o qual $F(x^*)$ é máximo, isto é, $F(x^*) = \sup_{x \in X} \{F(x)\}$. Se os valores de pertinência da matriz relacional R forem definidos de tal forma que $R(x^*, y) = H(y) \forall y \in Y$ e $R(x, y) = 0 \forall x \in X$*

² A Altura h de um conjunto nebuloso F corresponde ao seu maior grau de pertinência, entre todos os elementos do conjunto, isto é, $h(F) = \sup_{x \in X} F(x)$.

e $x \neq x^*$ e $\forall y \in Y$, então a matriz R claramente satisfaz a equação (2.13). É fácil verificar que R também é uma solução para o problema de estimação e pode-se concluir que existe solução (soluções) para P1 se e somente se a condição (2.13) é válida. \square

Para encontrar a solução do problema P1 para equações relacionais com composição sup-t é necessário primeiramente introduzir o operador φ . Este operador é uma função $\varphi : [0, 1] \rightarrow [0, 1]$ definida da seguinte forma:

$$a \varphi b = \sup\{c \in [0, 1] \mid a t c \leq b\} \quad (2.15)$$

Assim sendo, a solução maximal \hat{R} para o problema de estimação P1 relacionado à equações relacionais nebulosas do tipo $H = F \circ R$, se existe, é dada por:

$$\begin{aligned} \hat{R} &= F' \varphi H \\ \hat{R}(x, y) &= F(x) \varphi H(y) \end{aligned} \quad (2.16)$$

onde F' denota a transposta de F .

Como exemplo, considere $F = (0.3, 0.8, 0.5)$ e $H = (0.6, 0.3)$ como sendo dois conjuntos nebulosos discretos. É possível verificar que $\sup_{x \in X}\{F(x)\} = 0.8 \geq H(y) \forall y \in Y$ e, portanto, existe SPE. Assumindo uma composição sup-min (mínimo como t-norma) e usando (2.16), a relação $\hat{R} = [\hat{r}_{ij}]$, tal que $H = F \circ \hat{R}$, é calculada como se segue.

$$\begin{aligned} \hat{r}_{11} &= f_1 \varphi h_1 = \sup(c \in [0, 1] \mid \min(0.3, c) \leq 0.6) = 1.0 \\ \hat{r}_{21} &= f_2 \varphi h_1 = \sup(c \in [0, 1] \mid \min(0.8, c) \leq 0.6) = 0.6 \\ &\dots \\ \hat{r}_{32} &= f_3 \varphi h_2 = \sup(c \in [0, 1] \mid \min(0.5, c) \leq 0.3) = 0.3 \end{aligned}$$

$$\hat{R} = \begin{bmatrix} 1.0 & 1.0 \\ 0.6 & 0.3 \\ 1.0 & 0.3 \end{bmatrix}$$

É fácil verificar que a composição sup-t $F \circ \hat{R}$ tem como resultado o conjunto nebuloso H . Este resultado será utilizado mais adiante, na construção do algoritmo de treinamento baseado no aprendizado participativo, proposto neste trabalho.

SPE para equações relacionais com composição inf-s

Em uma equação relacional nebulosa com composição inf-s $H = F \bullet R$, o conjunto nebuloso de saída H é calculado da seguinte forma:

$$H(y) = \inf_{x \in X} \{F(x) \text{ s } R(x, y)\} \quad (2.17)$$

Analogamente ao apresentado para as equações relacionais nebulosas com composição sup-t, é preciso verificar a existência de SPE para o caso de equações relacionais com composição inf-s. Para isso, considere o seguinte teorema:

Teorema 2.2 *O Problema de Estimação (P1) para equações relacionais com composição do tipo inf-s possui solução (ou soluções) se e somente se o menor valor de pertinência do conjunto nebuloso F for menor ou igual ao menor valor de pertinência do conjunto nebuloso H , isto é:*

$$\inf_{x \in X} \{F(x)\} \leq H(y) \quad \text{para todo } y \in Y \quad (2.18)$$

Demonstração: *A demonstração deste teorema é similar àquela apresentada no Teorema 2.1.*

Da mesma forma que o apresentado para equações relacionais com composição sup-t, é necessário introduzir um operador para se encontrar a solução do problema P1 para equações relacionais com composição inf-s. Desta forma, considere agora função $\beta : [0, 1] \rightarrow [0, 1]$ definida da seguinte forma:

$$a \beta b = \inf \{c \in [0, 1] \mid a \text{ s } c \geq b\} \quad (2.19)$$

Então, a solução minimal \tilde{R} para o problema de estimação P1 associado à equações relacionais nebulosas do tipo $H = F \bullet R$, se existe, é dada por:

$$\begin{aligned} \tilde{R} &= F' \beta H \\ \tilde{R}(x, y) &= F(x) \beta H(y) \end{aligned} \quad (2.20)$$

Mais uma vez, considere o exemplo em que $F = (0.3, 0.8, 0.5)$ e $H = (0.6, 0.3)$. Aqui, a existência de solução também pode ser verificada, pois tem-se que $\inf_{x \in X} \{F(x)\} = 0.3 \leq H(y) \forall y \in Y$. Assim, escolhendo-se o máximo como s-norma e usando a equação (2.20), tem-se que a relação $\tilde{R} = [\tilde{r}_{ij}]$ tal que $H = F \bullet \tilde{R}$ pode ser calculada da seguinte forma:

$$\begin{aligned} \tilde{r}_{11} &= f_1 \beta h_1 = \inf (c \in [0, 1] \mid \max(0.3, c) \geq 0.6) = 0.6 \\ \tilde{r}_{21} &= f_2 \beta h_1 = \inf (c \in [0, 1] \mid \max(0.8, c) \geq 0.6) = 0 \\ &\dots \\ \tilde{r}_{32} &= f_3 \beta h_2 = \inf (c \in [0, 1] \mid \max(0.5, c) \geq 0.3) = 0 \end{aligned}$$

$$\tilde{R} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0 \\ 0.6 & 0 \end{bmatrix}$$

É possível verificar que a composição $F \bullet \tilde{R}$ tem como resultado exatamente o conjunto nebuloso H . Vale ressaltar que, tanto para composições inf-s quanto para composições sup-t, as equações aqui apresentadas fornecem apenas as soluções minimais e maximais, respectivamente, e, portanto, não garantem unicidade da solução. Para exemplificar este fato, considere a seguinte matriz relacional:

$$\bar{R} = \begin{bmatrix} 0.6 & 0.3 \\ 1.0 & 1.0 \\ 0.6 & 1.0 \end{bmatrix}$$

Calculando-se a composição $F \bullet \bar{R}$, também obtêm-se o conjunto nebuloso H . Portanto, uma vez encontrada a solução através das equações (2.16) e (2.20) ainda é possível fazer uso de outras técnicas de forma a se obter uma solução mais adequada ao que se deseja, caso esta exista. Este resultado também será empregado mais adiante na formulação do algoritmo de treinamento proposto neste trabalho.

2.4 Neurônios e Redes Neurais Nebulosas

Conforme mencionado anteriormente, a integração entre redes neurais e teoria de conjuntos nebulosos é atrativa na modelagem de sistemas dinâmicos. Em particular, redes neurais nebulosas herdam as características computacionais das redes neurais, suas arquiteturas e seus métodos de aprendizagem e adquirem dos sistemas nebulosos a capacidade de representação e processamento de conhecimento lingüístico, assim como os demais recursos oferecidos pela teoria de conjuntos nebulosos.

Sob esta perspectiva, três categorias de combinações foram propostas na literatura (Lin & Lee, 1996), traduzindo-se, cada uma delas, em:

- Redes neurais baseadas em neurônios nebulosos;
- Modelos neurais *fuzzificados*;
- Redes neurais clássicas com treinamento nebuloso.

Os modelos propostos neste trabalho são englobados pela primeira categoria, o das redes neurais baseadas em neurônios nebulosos e, portanto, esta será discutida com mais detalhes a seguir.

2.4.1 Neurônios Nebulosos

Um neurônio nebuloso é um neurônio artificial muito semelhante ao modelo clássico proposto por McCulloch-Pitts que reflete a natureza nebulosa de um neurônio biológico (Rocha, 1987). Este neurônio possui a habilidade de tratar informações imprecisas e/ou incertas pois sua formulação é baseada na teoria de conjuntos nebulosos.

Apesar de um grande número de neurônios nebulosos já terem sido propostos, estes podem ser classificados em três tipos distintos (Lin & Lee, 1996): (a) neurônios nebulosos com entradas não-nebulosas combinadas com pesos nebulosos (Tipo I), (b) neurônios nebulosos com entradas nebulosas que são combinadas com os pesos nebulosos (Tipo II) e (c) neurônios nebulosos descritos por equações lógicas nebulosas (Tipo III). A seguir, estes três tipos de neurônios nebulosos são brevemente apresentados.

Neurônios Nebulosos do Tipo I

O neurônio nebuloso do tipo I, apresentado na Figura 2.4, é aquele no qual as entradas são não-nebulosas, ou seja, as entradas são números reais e os pesos de conexão conjuntos nebulosos chamados de pesos nebulosos.

Assim, para n entradas não nebulosas x_1, x_2, \dots, x_n existem n , conjuntos nebulosos A_i , $i = 1, \dots, n$, sendo que as operações de ponderação através dos pesos são substituídas por operações com funções de pertinência, também conhecidas como operações de *fuzzificação*. O resultado de cada operação de fuzzificação, denotado por $A_i(x_i)$, é o grau de pertinência (ou valor de associação) da entrada x_i no conjunto nebuloso A_i .

Formalmente, um neurônio nebuloso do tipo I é um mapeamento $N_1 : \mathbb{R}^n \rightarrow [0, 1]$ definido da seguinte forma:

$$N_1(x_1, x_2, \dots, x_n) = A_1(x_1) \otimes A_2(x_2) \otimes \dots \otimes A_n(x_n) \quad (2.21)$$

onde o operador de agregação \otimes pode ser qualquer mecanismo de agregação tais como os operadores min, max ou qualquer outra t-norma ou s-norma discutidas anteriormente.

Neurônios Nebulosos do Tipo II

Este tipo de neurônio nebuloso, denotado por N_2 e apresentado na Figura 2.5, é similar aos do tipo I, exceto pelo fato de que todas as entradas e saídas são conjuntos nebulosos ao invés de valores não-nebulosos. Cada uma das entradas não nebulosas X_i é composta com seu respectivo peso nebuloso A_i através de um operador de ponderação ‘*’, que pode ser, por exemplo, a multiplicação entre dois conjuntos nebulosos, composições nebulosas, entre outros (veja (Gomide & Pedrycz, 2007)

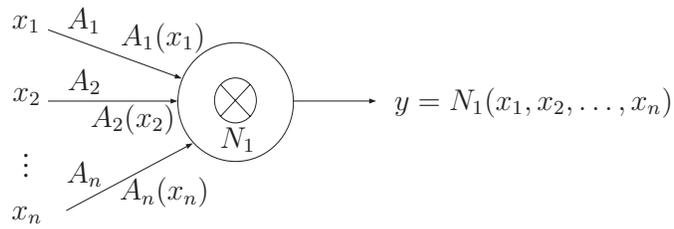


Figura 2.4: Neurônio nebuloso do tipo I

para mais detalhes). O resultado desta ponderação é um outro conjunto nebuloso $X'_i = A_i * X_i$, $i = 1, \dots, n$. Todos os conjuntos nebulosos X'_i são então agregados através de um operador de agregação \otimes para produzir o conjunto nebuloso de saída Y .

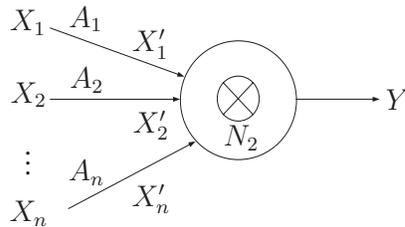


Figura 2.5: Neurônio nebuloso do tipo II

Este neurônio produz um mapeamento $N_2 : [0, 1]^n \rightarrow \times[0, 1]$, e pode ser matematicamente descrito da seguinte forma:

$$\begin{aligned} X'_i &= A_i * X_i, \quad i = 1, 2, \dots, n \\ Y &= N_2(X'_1, X'_2, \dots, X'_n) = X'_1 \otimes X'_2 \otimes \dots \otimes X'_n \end{aligned} \quad (2.22)$$

Neurônios Nebulosos do Tipo III

A relação de entrada/saída de neurônio nebuloso do tipo III, apresentado na Figura 2.6, é representada por uma regra nebulosa se-então da forma:

$$SE \ X_1 \ E \ X_2 \ E \ \dots \ E \ X_n, \ ENTÃO \ Y \quad (2.23)$$

onde X_1, X_2, \dots, X_n são as entradas e Y é a saída. Um neurônio nebuloso do tipo III pode também ser descrito por uma relação nebulosa R , por exemplo:

$$R = X_1 \times X_2 \times \dots \times X_n \times Y, \quad (2.24)$$

ou, no caso geral,

$$R = f(X_1, X_2, \dots, X_n, Y), \quad (2.25)$$

onde $f(\cdot)$ representa uma função de implicação. Assim, a partir das entradas x_1, x_2, \dots, x_n (nebulosas ou não), e de acordo com a regra de inferência composicional adotada, o neurônio nebuloso do tipo III fornece uma saída da forma:

$$Y = x_1 \diamond (x_2 \diamond (\dots \diamond (x_n \diamond R) \dots) \dots), \quad (2.26)$$

onde \diamond representa um tipo de composição nebulosa como, por exemplo, uma composição sup-t ou inf-s. Para este neurônio, as entradas podem ser tanto nebulosas quanto não-nebulosas, sendo que um neurônio nebuloso do tipo III com entradas não-nebulosas pode ser visto como um caso especial deste neurônio com entradas nebulosas. Uma rede neural nebulosa composta por neurônios do tipo III aparece como uma excelente arquitetura para extração de regras a partir de dados de treinamento.

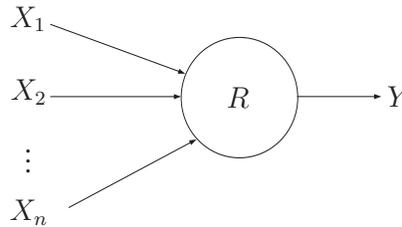


Figura 2.6: Neurônio nebuloso do tipo III

Há ainda uma variação do neurônio nebuloso do tipo III, chamado de *neurônio lógico*, proposto em (Pedrycz & Rocha, 1993), que pode ser visto como uma combinação dos neurônios do tipo II e III. A estrutura deste neurônio é muito similar à apresentada para os neurônios nebulosos do tipo III, exceto pelo fato de que pesos de conexão w_i são associados a cada uma das entradas x_i . Desta forma, um neurônio lógico executa um mapeamento do espaço formado pelo produto cartesiano entre o espaço de entrada e o espaço dos pesos no intervalo unitário, isto é, $X \times W \rightarrow [0, 1]$.

A forma com que entradas e pesos são agregados definem dois tipos de neurônios lógicos: neurônios lógicos do tipo *AND* e neurônios lógicos do tipo *OR*. De fato, o nome *neurônio lógico* está associado às operações lógicas de disjunção (*or*) e conjunção (*and*) executadas por estes neurônios, e que, aqui,

conforme mencionado anteriormente, são implementadas através de suas generalizações nebulosas, as t-normas e s-normas. As implementações para os dois tipos de neurônios lógicos são apresentadas a seguir.

Neurônio Lógico AND

O neurônio lógico do tipo *AND* é obtido escolhendo-se uma s-norma como operador de ponderação ‘*’ na expressão (2.22), enquanto o operador de agregação \otimes é implementado através de uma t-norma. Isto faz com que, para este caso particular, a equação (2.22) possa ser reescrita da seguinte forma:

$$\begin{aligned} X'_i &= A_i s X_i, \quad i = 1, 2, \dots, n \\ Y &= X'_1 t X'_2 t \dots t X'_n \end{aligned} \quad (2.27)$$

ou, de uma forma mais concisa

$$y = \text{and}(\mathbf{x}, \mathbf{w}) = \mathop{\mathbf{T}}_{i=1}^n (x_i s w_i) \quad (2.28)$$

onde $\mathbf{x} = [x_1, x_2, \dots, x_n]$ é o vetor de entradas, $\mathbf{w} = [w_1, w_2, \dots, w_n]$ é o vetor de pesos, $y \in [0, 1]$ é o valor de saída e $\mathop{\mathbf{T}}_{i=1}^n (\cdot)$ representa uma t-norma, ou seja, $\mathop{\mathbf{T}}_{i=1}^n (z_i) = z_1 t z_2 t \dots t z_n$.

Sob a ótica do neurônio nebuloso do tipo III, o neurônio lógico *AND* também pode ser visto como uma relação nebulosa entre entradas e saída. Em particular, se o operador min for escolhido para implementar a t-norma na equação (2.28), a saída deste neurônio pode ser vista como uma composição min-s (inf-s) entre entradas e pesos, ou seja:

$$y = \mathbf{x} \bullet \mathbf{w}$$

Conseqüentemente, cada neurônio lógico do tipo *AND* representa uma regra nebulosa da forma:

$$SE (x_1 \text{ OU } w_1) E (x_2 \text{ OU } w_2) E \dots E (x_n \text{ OU } w_n), \text{ ENTÃO } Y \quad (2.29)$$

A figura 2.7 apresenta a estrutura para o neurônio lógico do tipo *AND*.

Em geral, é associada à saída deste neurônio uma função de ativação $\varphi_{\text{and}}(\cdot)$, tomada como sendo um mapeamento não-linear monotonicamente crescente. Entretanto, neste trabalho, esta função foi escolhida como sendo a função identidade, isto é, $\varphi_{\text{and}}(y) = y$.

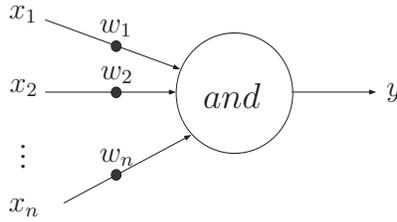


Figura 2.7: Neurônio nebuloso lógico do tipo *AND*

Neurônio Lógico *OR*

A obtenção do neurônio lógico do tipo *OR* é feita escolhendo-se uma t-norma como operador de ponderação e uma s-norma como operador de agregação. Assim, a saída deste tipo de neurônio é calculada da seguinte forma:

$$y = or(\mathbf{x}, \mathbf{w}) = \mathop{\text{S}}_{i=1}^n (x_i \text{ t } w_i), \quad (2.30)$$

onde $\mathop{\text{S}}_{i=1}^n (\cdot)$ representa uma s-norma, ou seja, $\mathop{\text{S}}_{i=1}^n (z_i) = z_1 \text{ s } z_2 \text{ s } \dots \text{ s } z_n$.

Da mesma forma que para o neurônio lógico do tipo *AND*, o neurônio lógico do tipo *OR* também pode ser visto como uma relação nebulosa entre entradas e saída. Aqui, o caso particular é aquele em que se escolhe o operador \max para implementar a t-norma em (2.30), o que faz com que a saída y possa ser vista como uma composição max-t (sup-t) entre entradas e pesos, ou seja:

$$y = \mathbf{x} \circ \mathbf{w}$$

Portanto, cada neurônio lógico do tipo *OR* representa uma regra nebulosa da forma:

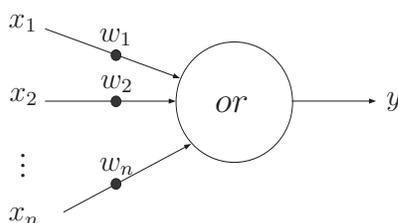
$$SE (x_1 \text{ E } w_1) \text{ OU } (x_2 \text{ E } w_2) \text{ OU } \dots \text{ OU } (x_n \text{ E } w_n), \text{ ENTÃO } Y \quad (2.31)$$

A figura 2.8 apresenta a estrutura do neurônio lógico do tipo *OR*.

Da mesma forma que para o neurônio *AND*, neste neurônio a função de ativação associada à saída do neurônio ($\varphi_{or}(\cdot)$) também foi escolhida como sendo a identidade, isto é, $\varphi_{or}(y) = y$.

É possível verificar que os neurônios lógicos dos tipos *AND* e *OR* realizam operações lógicas “puras” entre os valores de pertinência, sendo que o propósito dos pesos de conexão é diferenciar os diferentes níveis de impacto que cada entrada individual deve ter no resultado final da agregação.

Este tipo de neurônio apresenta uma forte semelhança com os modelos clássicos de neurônio, o que os torna particularmente atrativos para a construção de redes neurais nebulosas que herdem as

Figura 2.8: Neurônio nebuloso lógico do tipo *OR*

capacidades de treinamento das redes neurais clássicas. Além disso fornecem uma forma intuitiva para a extração de regras nebulosas a partir dos modelos treinados, tornando-os linguisticamente interpretáveis.

Na seção a seguir, são apresentadas as principais estruturas de redes neurais nebulosas baseadas nos neurônios lógicos dos tipos *AND* e *OR*, e que servirão de base para os modelos propostos neste trabalho.

2.4.2 Redes Neurais Nebulosas

Diversas estruturas de rede neurais nebulosas foram propostas na literatura (Buckley & Hayashi, 1994; Lin & Lee, 1996). Em particular, esta seção trata das redes formadas por neurônios lógicos *AND* e *OR* (Pedrycz, 1993), apresentando algumas das estruturas de redes já propostas na literatura (Pedrycz et al., 1995; Caminhas et al., 1999; Ballini & Gomide, 2002; Figueiredo et al., 2004) nas quais será fundamentada a estrutura utilizada neste trabalho.

As primeiras estruturas a utilizar os neurônios *AND* e *OR* foram propostas por Pedrycz em 1995 (Pedrycz et al., 1995). Estas estruturas, denominadas de processadores lógicos, demonstram certas analogias com as funções Booleanas da lógica clássica.

O primeiro tipo de processador lógico, ilustrado na Figura 2.9, é formado por h neurônios *AND* em sua camada intermediária e por um único neurônio *OR* na camada de saída. A função dos neurônios *AND* é fornecer uma seqüência de mintermos generalizados dos sinais de entrada. Esses mintermos generalizados são então combinados através da união lógica fornecida pelo neurônio *OR* situado na camada de saída. Desta forma, a rede resultante fornece um tipo de aproximação lógica das relações entre o vetor de entrada e sua respectiva saída conhecida, como soma de mintermos.

O segundo tipo de processador lógico proposto por Pedrycz, apresentado na Figura 2.10, fornece uma estrutura dual à apresentada anteriormente e fornece uma relação lógica conhecida como produto de maxitermos. Nesta topologia, a camada intermediária fornece uma família de maxitermos

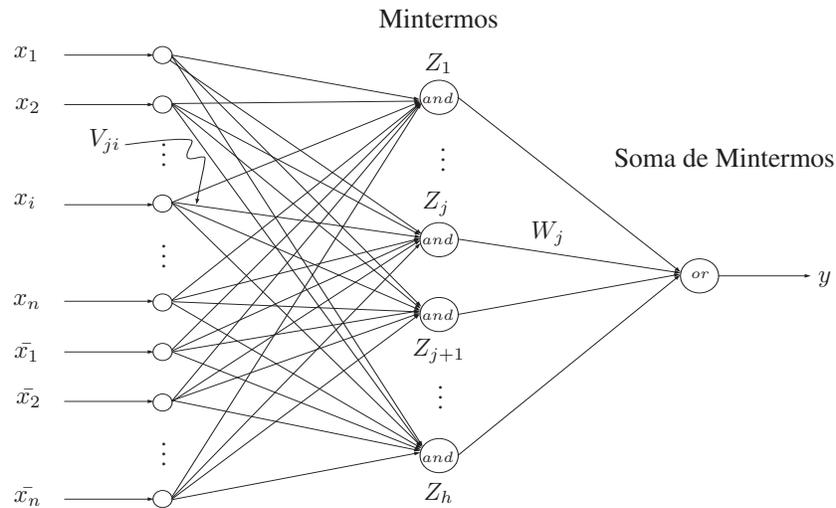


Figura 2.9: Processador lógico do tipo Soma de Mintermos

generalizados das entradas. A camada de saída, composta por um único neurônio *AND*, tem a função de agregar os maxitermos generalizados fornecidos pela camada anterior.

Para ambas as redes, as entradas x_i são entradas nebulosas, sendo que as entradas \bar{x}_i representam o complemento das entradas x_i . Os pesos de conexão dados por $V_{ji} \in [0, 1]$ representam a importância individual da entrada x_i no processamento do j -ésimo neurônio da camada intermediária, enquanto os pesos de conexão $W_j \in [0, 1]$ representam a importância individual da saída do j -ésimo neurônio da camada intermediária na composição final da saída.

Outra importante estrutura de rede neural nebulosa composta por neurônios lógicos *AND* e *OR* foi proposta por Caminhas *et. al.* em 1999 para resolver o problema de classificação de padrões (Caminhas et al., 1999). Esta estrutura, ilustrada na Figura 2.11, possui a característica de representar o conhecimento de forma explícita, apresentando duas diferenças importantes com relação à classe anterior. Além de possibilitarem a inserção e/ou extração de conhecimento na forma de regras nebulosas do tipo se-então, são também capazes de projetar sistemas nebulosos definindo seus parâmetros à medida que adquirem conhecimento durante o período de aprendizagem.

Nesta arquitetura, as entradas x_i são variáveis não-nebulosas que são fuzzificadas na primeira camada da rede, onde cada neurônio representa um conjunto nebuloso, sendo que $A_i^{\lambda_i}$ representa o λ_i -ésimo conjunto nebuloso da i -ésima entrada. Estes neurônios podem ser vistos como neurônios nebulosos do tipo I que possuem apenas uma entrada.

As saídas dos neurônios da primeira camada são combinadas com seus respectivos pesos $w_{ki} \in$

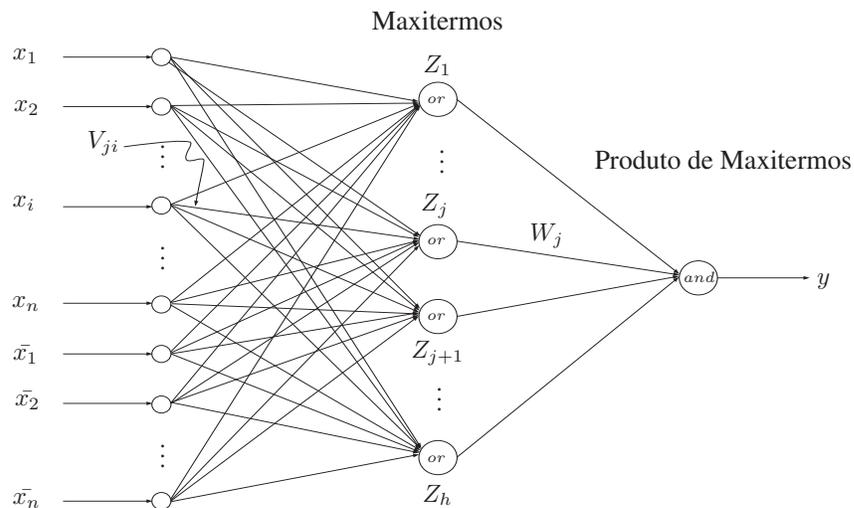


Figura 2.10: Processador lógico do tipo Produto de Maxitermos

$[0, 1]$ pelos neurônios *AND* que formam a primeira camada intermediária da rede. A saída de cada um destes neurônios *AND* y_k pode ser vista como a saída de uma regra nebulosa do tipo se-então. A segunda camada intermediária da rede executa a agregação das regras nebulosas representadas pela primeira camada intermediária. Desta forma, as saídas y_k da regras nebulosas são agregadas com seus respectivos pesos $v_{j\ell} \in [0, 1]$ através dos neurônios *OR*.

A saída de cada um destes neurônios *OR* representam o grau de certeza com que a entrada $\mathbf{x} = [x_i], i = 1, \dots, n$ pertence a uma determinada classe, isto é, Z_ℓ é o grau de certeza com que \mathbf{x} pertence à classe C_ℓ . A saída final da rede é dada por uma função competitiva MAX, sendo que a classe a qual pertence a entrada \mathbf{x} será aquela que apresentar o maior valor de Z_ℓ associado a ela.

Por fim, pode-se ainda apresentar a estrutura de rede neural nebulosa recorrente proposta em (Ballini & Gomide, 2002). Nesta estrutura, a função de agregação das saídas das regras nebulosas é feita através de uma rede neural clássica, formando assim a estrutura híbrida apresentada na Figura 2.12.

Esta estrutura apresenta duas partes distintas. A primeira delas é formada por duas camadas de neurônios nebulosos e representa um sistema de inferência nebuloso. A primeira destas duas camadas é formada pelos mesmos neurônios de fuzzificação usados na estrutura da Figura 2.11. A segunda camada desta parte da rede pode ser formada por neurônios *AND* ou neurônios *OR*. O processamento nas duas primeiras camadas desta rede é muito similar ao da estrutura anterior. Entretanto, aqui são associadas recorrências locais e globais aos neurônios *AND/OR*, fazendo com que a entrada destes neurônios seja acrescida das saídas dos próprios neurônios *AND/OR* no instante anterior, sendo estas

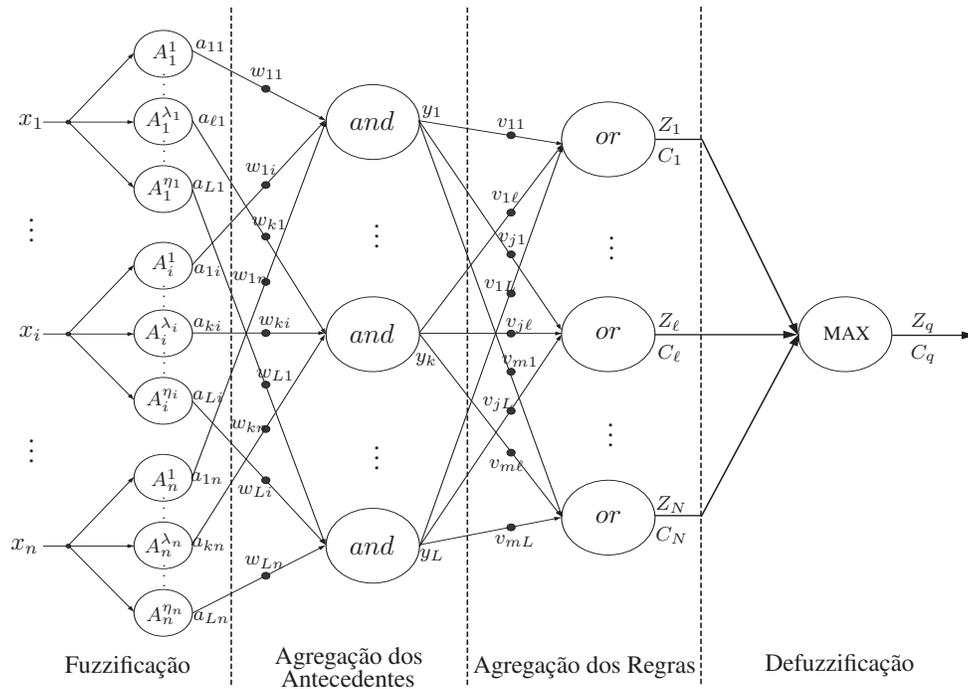


Figura 2.11: Modelo de rede neural nebulosa para classificação de padrões

saídas ponderadas pelos pesos de conexão $r_{k\ell}$ (peso de recorrência entre os neurônios lógicos $k, k = 1, \dots, L$ e $\ell = 1, \dots, L$ na segunda camada da rede).

A segunda parte é formada por uma rede neural clássica de uma camada e funciona como um agregador não-linear, agregando a saída das regras nebulosas da primeira parte da rede e fornecendo uma saída global para a rede. A saída desta parte da rede é dada por $y_j = f\left(\sum_{k=1}^L v_{jk} * z_k\right)$, onde $f(\cdot)$ é uma função de ativação monotônica contínua.

O aprendizado desta rede é feita em três etapas. Na primeira etapa, um algoritmo de agrupamento é utilizado para granularizar o espaço de entrada. Na segunda etapa o método do gradiente é utilizado para corrigir os pesos relacionados aos neurônios clássicos na camada de saída. Por fim, na terceira etapa aplica-se uma regra de reforço associativo para corrigir os pesos relacionados aos neurônios nebulosos na camada intermediária. Maiores detalhes são apresentados em (Ballini & Gomide, 2002).

Da mesma forma que a estrutura anterior, esta rede permite a representação do conhecimento na forma de regras se-então. Além disso, a incorporação das recorrências nos neurônios nebulosos na camada intermediária da rede faz com que esta seja capaz de capturar as relações temporais entre entradas/saídas distintas, tornando-se assim uma alternativa na modelagem de sistemas dinâmicos.

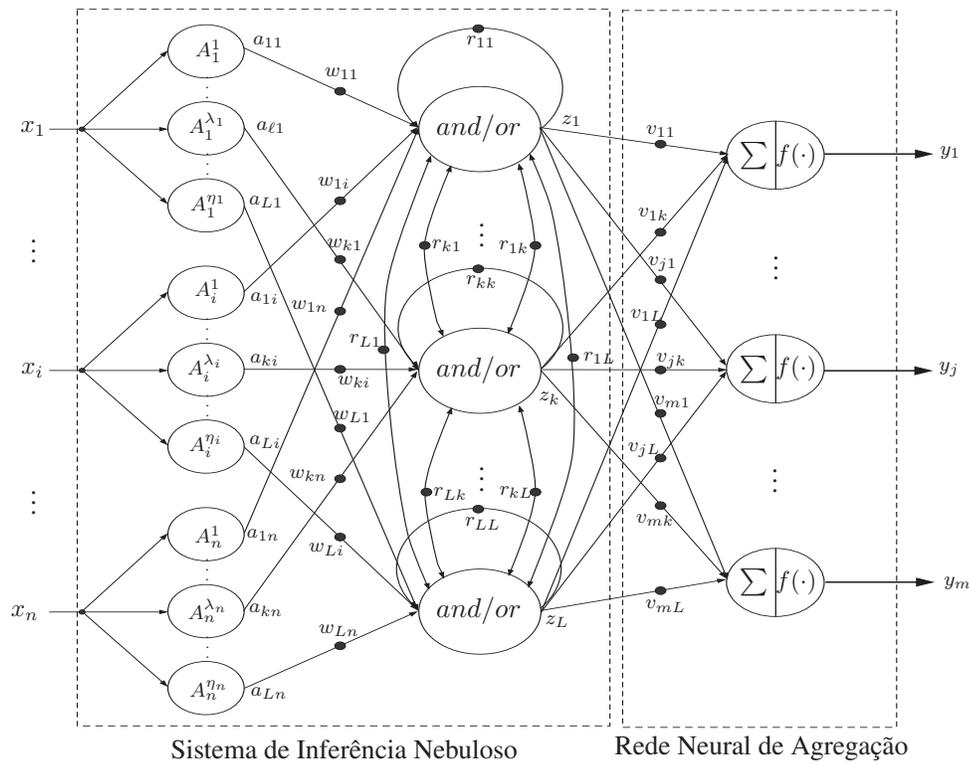


Figura 2.12: Modelo de rede neural nebulosa recorrente

2.5 O Espaço Neural Multidimensional

O espaço neural multidimensional \mathcal{N} proposto por Figueiredo surge como uma forma de se obter uma realização estrutural e paramétrica para uma classe de modelos qualitativos implementados através de redes neurais clássicas (Figueiredo, 2003).

Este espaço é definido como sendo um espaço separável de Hilbert de mapeamentos não-lineares f , que mapeiam um dado vetor \mathbf{x} a partir de um espaço de dados X (o qual também é um espaço separável de Hilbert ou um espaço Euclidiano) para um vetor m -dimensional \mathbf{y} composto por m saídas da forma $y_j = f_j(\mathbf{x})$, onde $f_j(\cdot)$ são funcionais analíticos expressos como séries funcionais de Volterra em X (Volterra, 1959). Os funcionais f_j pertencem a um espaço de Hilbert apropriadamente construído, denotado por F , também introduzido por De Figueiredo (Figueiredo & DwyerIII, 1980), como uma generalização de um espaço de Fock simétrico. Uma vez definido o espaço F é possível definir o Espaço Neural \mathcal{N} como um produto direto do espaço F , isto é:

$$\mathcal{N} = \overbrace{F \times F \times \dots \times F}^m, \quad (2.32)$$

o que faz com que os membros de \mathcal{N} tenham um domínio comum (X). De Figueiredo mostra que os membros de \mathcal{N} são otimamente implementados como uma rede neural. Portanto, o produto escalar em \mathcal{N} mede a similaridade entre duas redes neurais que representam os mapeamentos f e g , e a norma em \mathcal{N} , quando usada como $\|f - g\|_{\mathcal{N}}$ expressa a distância métrica entre essas duas redes.

Estes conceitos se apresentaram como altamente promissores na modelagem de sistemas dinâmicos e inspiraram estudos no sentido de estender estas idéias para tratar também redes neurais nebulosas, tornando-se inicialmente um dos principais objetivos deste trabalho. Entretanto, análises experimentais demonstraram que para fornecer uma realização estrutural e paramétrica para um problema específico, a abordagem de De Figueiredo exige que os vetores que formam o conjunto de dados de treinamento sejam todos linearmente independentes entre si, sendo que o número de neurônios na camada intermediária das redes geradas por este método é exatamente igual ao número destes vetores de treinamento.

Estas análises levaram a crer que, apesar de fornecer uma formulação matemática adequada à implementação de redes neurais clássicas, o espaço \mathcal{N} possui desvantagens com relação à complexidade dos algoritmos utilizados se comparado às redes neurais nebulosas apresentadas anteriormente, fazendo com que esta abordagem não fosse considerada neste trabalho.

2.6 Aprendizado Participativo

A característica principal do aprendizado participativo é que o impacto de uma observação no aprendizado ou na revisão das convicções depende de sua compatibilidade com as convicções atuais. Em particular, uma observação conflitante com as convicções atuais é desconsiderada (Yager, 1990).

Este modelo de aprendizado captura muitas das características inerentes ao aprendizado humano, uma vez que, quando já possuem algum conhecimento a respeito de determinado assunto, os seres humanos tendem a não aceitar de imediato novos conceitos que não estejam de acordo com aquilo que já aprenderam sobre tal assunto até o presente momento.

Para exemplificar este conceito, considere v como sendo uma variável que representa o que já se aprendeu a respeito de um determinado ambiente. O objetivo do aprendizado participativo é estimar o valor desta variável baseado em uma seqüência de observações $x^k \in [0, 1]$, $k = 1, 2, \dots$, que codificam o conhecimento a respeito do valor da variável $v \in [0, 1]$. Neste sentido, x^k é uma manifestação do valor de v na k -ésima observação. Assim, usa-se o vetor x^k como um meio para "aprender" o valor de v . O processo de aprendizado é participativo se a contribuição de cada observação x^k para

o processo de aprendizado depende de sua aceitação pelos valores correntes de v (Yager, 1990). Isto significa que, para ser relevante para o processo de aprendizagem, x^k deve estar próximo de v . Um mecanismo para atualização dos valores de v (chamados genericamente de crenças ou convicções, por Yager) é:

$$v^{k+1} = v^k + \alpha_k \rho_k (x^k - v^k) \quad (2.33)$$

onde $k = 1, \dots, P$ e P é o número de observações; $v^{k+1} \in [0, 1]$ é a nova convicção a respeito do sistema; $v^k \in [0, 1]$ é a convicção corrente; x^k é a observação corrente; $\alpha_k \in [0, 1]$ é a taxa de aprendizado (que pode ser constante ou variável); e $\rho_k \in [0, 1]$ é o grau de compatibilidade entre x^k e v^k . Uma das formas de se calcular ρ_k é:

$$\rho_k = 1 - \frac{1}{n} \sum_{i=1}^n d_i^k \quad (2.34)$$

sendo $d_i^k = |x_i^k - v_i^k|$. Quando $\rho_k = 0$, tem-se que a observação é muito diferente da convicção corrente pois, como $d_i^k \in [0, 1]$ para que ρ_k seja igual a zero é necessário que cada uma dos elementos x_i^k e v_i^k devem estar em extremos opostos, ou seja, se $x_i^k = 0$ então $v_i^k = 1$ ou se $x_i^k = 1$ então $v_i^k = 0$ para todo $i = 1, \dots, n$. Neste caso tem-se que, a partir de (2.33) que $v^{k+1} = v^k$ e o sistema está totalmente fechado para o aprendizado.

Por outro lado, se $\rho_k = 1$ tem-se a partir de (2.33) que $v^{k+1} = v^k + \alpha_k (x^k - v^k)$, e portanto o sistema está totalmente aberto para o aprendizado. Entretanto, para que ρ_k seja igual a 1 é preciso que $x_i^k - v_i^k = 0$ para todo i . Esta condição implica que $v^k = x^k$ e assim tem-se que $v^{k+1} = v^k$, o que implica que não ocorre nenhum aprendizado (Yager & Filev, 1993a).

Estes fatos implicam que quando uma nova observação é muito distinta do que já se sabe a respeito do ambiente ela é considerada como incompatível e não entra no processo de aprendizado. Por outro lado, quando uma nova observação é igual ao conhecimento que já se tem a respeito do ambiente, não é necessário mais aprender já que esta observação não contribui com o aprendizado. Assim, o comportamento da atualização das convicções se dá de acordo com um compromisso entre observações muito próximas da convicção atual e de observações conflitantes com tal convicção. Esta idéia pode ser visualizada na Figura 2.13, que apresenta um exemplo deste comportamento para um espaço bidimensional, onde a convicção atual se encontra no centro do espaço.

Um ponto que pode ser levantado a respeito do processo de treinamento descrito é que este ignora a situação em que uma seqüência de baixos valores de ρ_k 's são observados durante um longo período de tempo. Neste caso, o sistema deve se tornar mais aberto para aprender com as novas observações, uma vez que para este caso é a convicção que se tem a respeito do sistema que pode estar errada e não as novas observações. Esta situação pode ser traduzida em um índice de "alerta", usado para

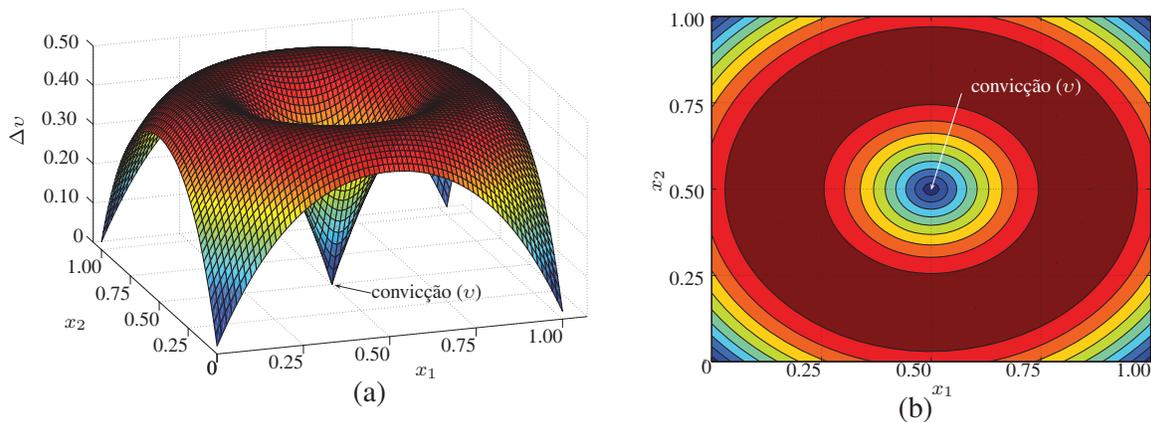


Figura 2.13: Comportamento da aprendizagem participativa

influenciar o processo de aprendizado, como mostra a Figura 2.14. Quanto maior o índice de alerta, menor é a confiança que se tem a respeito da convicção atual do sistema e observações conflitantes se tornam importantes no processo de atualização das convicções.

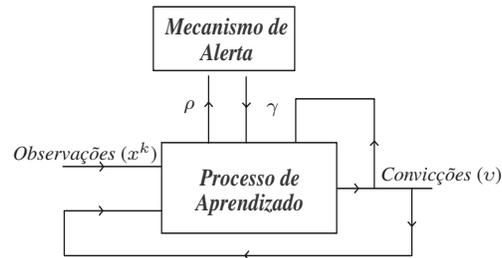


Figura 2.14: Aprendizado participativo com alerta

Assim, considere $\gamma_k \in [0, 1]$ como sendo o índice de alerta. Quanto mais alto for o valor de γ_k mais alerta está o sistema. Um mecanismo para determinar o índice de alerta é:

$$\gamma_{k+1} = \gamma_k + \lambda((1 - \rho_{k+1}) - \gamma_k) \quad (2.35)$$

onde $\lambda \in [0, 1]$ é uma constante que controla a taxa com a qual o índice de alerta é ajustado; quanto mais próximo λ está de 1, mais sensível fica o sistema à variações de compatibilidade. O índice de alerta pode ser visto como o complemento da confiança na convicção atualmente presente no sistema. A Figura 2.15 ilustra o comportamento do índice de alerta e da atualização das convicções para diferentes seqüências de índices de compatibilidade.

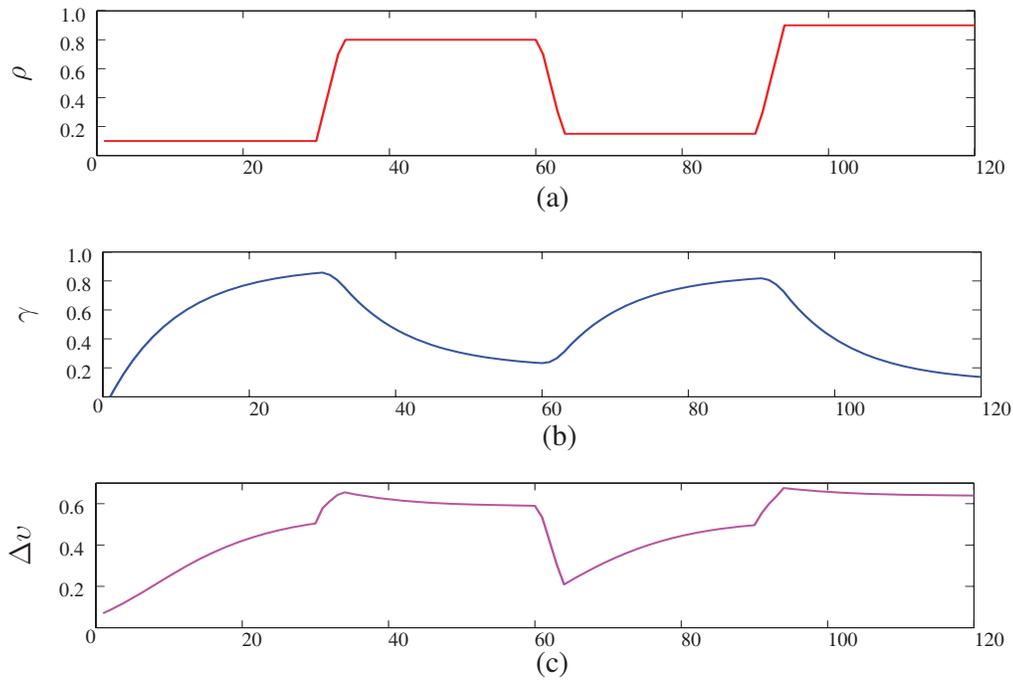


Figura 2.15: Comportamento do índice de alerta e da variação das convicções: (a) Índice de compatibilidade (b) Índice de alerta (c) Variação das convicções.

A expressão (2.33) pode então ser reescrita de forma a incorporar o mecanismo de alerta da seguinte forma:

$$v^{k+1} = v^k + \alpha_k \rho_k^{1-\gamma_k} (x^k - v^k) \quad (2.36)$$

ou na forma completa:

$$v^{k+1} = x^k + \alpha_k \left(1 - \frac{1}{n} \sum_{i=1}^n |x_i^k - v_i^k| \right)^{1-\gamma_k} (x^k - v^k) \quad (2.37)$$

Como pode ser verificado em (2.36) e (2.37), enquanto ρ_k mede o quanto o sistema muda sua credibilidade em suas próprias convicções, o índice de alerta γ_k atua como um crítico que avisa quanto uma convicção deve ser modificada frente a novas evidências.

Este paradigma de aprendizado pode ser visto como uma base para o desenvolvimento de algoritmos de treinamento para sistemas baseados em inteligência computacional, tais como redes neurais, sistemas fuzzy e suas hibridizações. De fato, o aprendizado participativo foi utilizado para desenvolver um eficiente algoritmo de agrupamento fuzzy não supervisionado (Silva et al., 2005) e, também, para encontrar estruturas de base de regras em procedimentos de modelagem fuzzy evolutiva

(Lima et al., 2006).

Uma das propostas deste trabalho é a utilização do aprendizado participativo na síntese de um algoritmo de treinamento para uma classe de redes neurofuzzy, conforme apresentado no próximo capítulo.

2.7 Resumo

O objetivo deste capítulo foi descrever em linhas gerais o contexto de modelagem de sistemas dinâmicos, foco do presente trabalho. Inicialmente apresentou-se as principais propriedades de modelos matemáticos, apontando quais destas propriedades estão presentes nos modelos propostos nesta tese. Em suma, os modelos aqui propostos podem ser classificados como modelos não-lineares, discretos, determinísticos, a parâmetros concentrados, não autônomos e paramétricos.

Além disso, foram descritos neste capítulo os conceitos gerais da teoria de conjuntos nebulosos, com ênfase nas equações relacionais nebulosas e as soluções para o problema de estimação nebuloso. Redes neurais nebulosas baseadas em neurônios lógicos do tipo *AND/OR* também foram introduzidos para servir de motivação para uma nova classe de redes neurais nebulosas que será objeto do capítulo 4.

Por fim, este capítulo apresentou os conceitos básicos a respeito do paradigma do aprendizado participativo, com a finalidade de facilitar a compreensão do algoritmo de treinamento para redes neurais nebulosas proposto no capítulo seguinte.

Capítulo 3

Redes Neurais Nebulosas com Aprendizado Participativo

3.1 Introdução

Neste capítulo propõe-se um novo procedimento para se treinar redes neurais nebulosas, tomando-se como base o paradigma do aprendizado participativo (AP) apresentado no capítulo anterior.

Conforme mencionado, os métodos de treinamento existentes, tanto para redes neurais clássicas quanto para redes neurais nebulosas, apresentam uma forte dependência no que diz respeito à qualidade da base de dados utilizada para o treinamento. De fato, é possível obter bons resultados com os métodos de treinamento apenas nos casos em que os dados de treinamento sejam representativos do comportamento que se deseja aprender (Yager, 1990).

Apesar de o AP apresentar claros indícios de que sua aplicação seria vantajosa para se melhorar tais características dos processos de treinamento existentes, até o momento nenhum algoritmo eficaz baseado nesse paradigma foi proposto na literatura. A vantagem do AP se deve, em parte, ao fato de que uma das suas principais características é a comparação entre a informação que chega ao sistema e o conhecimento já armazenado, a fim de se calcular o quanto essa nova informação é compatível com as convicções atuais e, assim, atualizar ou não o conhecimento (Seção 2.6).

Na maioria das redes neurais, nebulosas ou não, o conhecimento que já se tem a respeito do meio ou do processo sob estudo fica armazenado nos pesos de conexão, ou, mais especificamente, nas matrizes onde são armazenados tais pesos. Portanto, para aplicar o AP a estas redes, torna-se necessário traduzir a nova informação (em geral na forma de um par de treinamento entrada/saída) para um formato matricial que possa ser comparado com as matrizes de pesos.

Para se entender melhor este problema, considere a rede neural clássica de 3 entradas e 2 saídas apresentada na Figura 3.1.

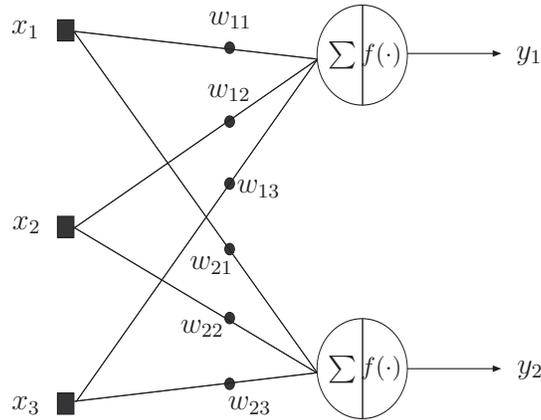


Figura 3.1: Exemplo de Rede Neural Clássica de 3 entradas e 2 saídas.

Para esta rede tem-se que o vetor de entrada $\mathbf{x} \in \mathbb{R}^3$, o vetor de saída $\mathbf{y} \in [0, 1]^2$ e a matriz de peso $W = [w_{ij}]$ (onde está armazenado o conhecimento que esta rede já possui) pertence ao $\mathbb{R}^{2 \times 3}$. De forma geral, pode-se obter uma saída $\hat{y}_i = [\hat{y}_i]$, $i = 1, 2$, para esta rede a partir de uma determinada entrada \mathbf{x} , da seguinte forma:

$$\hat{y}_i = f \left(\sum_{j=1}^3 x_j w_{ij} \right), \quad i = 1, 2. \quad (3.1)$$

Considere agora um par formado por uma entrada e uma saída na forma $[\mathbf{x}, \mathbf{y}]$. Para que o AP possa ser aplicado no treinamento desta rede, seria necessário encontrar, a partir de um par $[\mathbf{x}, \mathbf{y}]$ uma matriz $G = [g_{ij}]$, $i = 1, 2, j = 1, 2, 3$, de tal forma que, substituindo-se w_{ij} na equação (3.1) por g_{ij} , o resultado obtido fosse exatamente o vetor \mathbf{y} , ou seja:

$$y_i = f \left(\sum_{j=1}^3 x_j g_{ij} \right), \quad i = 1, 2. \quad (3.2)$$

Desta forma, a matriz $G = [g_{ij}]$ representaria a informação que está sendo trazida para a rede através do par entrada/saída $[\mathbf{x}, \mathbf{y}]$ e, portanto, poderia ser utilizada no cálculo do índice de compatibilidade para o AP, conforme mencionado no capítulo anterior.

Apesar de existirem alguns métodos (Stewart, 1973; Björck, 1996) que permitem encontrar a matriz G para o exemplo da Figura 3.1, estes métodos são, em geral, computacionalmente onerosos, o que inviabilizaria a aplicação do AP para problemas com dimensões mais elevadas. Por outro lado, considere a rede neural nebulosa apresentada na Figura 3.2, onde Θ representa ou a função *and*

definida em (2.28) ou a função *or* definida em (2.30).

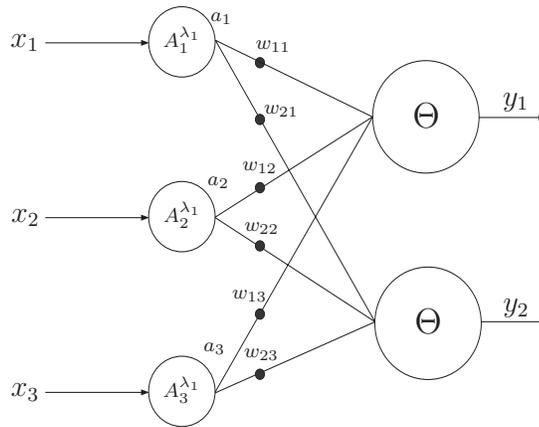


Figura 3.2: Exemplo de Rede Neural Nebulosa de 3 entradas e 2 saídas.

Conforme visto no capítulo anterior, dependendo da escolha dos operadores que representam as t-normas e s-normas nas funções *and* e *or*, o mapeamento entrada/saída dado pela rede da Figura 3.2 pode ser visto como uma relação nebulosa da forma:

$$\mathbf{y} = \mathbf{a} \diamond W, \quad (3.3)$$

onde $\mathbf{a} = [a_i]$ ($a_i = A_i^{\lambda_1}(x_i)$) é o vetor que armazena as saídas dos neurônios nebulosos da primeira camada da rede (que podem ser vistos como neurônios nebulosos do tipo I com uma única entrada) e o operador \diamond representa uma composição inf-s (para o caso em que $\Theta = \text{and}$ e a t-norma = min) ou uma composição sup-t (para o caso em que $\Theta = \text{or}$ e a s-norma = max).

Considere novamente o par entrada/saída dado por $[\mathbf{x}, \mathbf{y}]$. Como o vetor \mathbf{a} pode ser facilmente obtido através da propagação de \mathbf{x} através da primeira camada de neurônios nebulosos (N_1), a matriz G que representa a informação trazida pelo par $[\mathbf{x}, \mathbf{y}]$ deve ser tal que:

$$\mathbf{y} = \mathbf{a} \diamond G. \quad (3.4)$$

Neste caso, G pode ser obtida a partir da solução do problema de estimação nebuloso, conforme seção 2.3.8 do capítulo anterior. Este método fornece uma forma simples e rápida de se obter a informação que está sendo trazida pela nova observação que chega à rede, viabilizando, assim, o uso do aprendizado participativo para esta classe de redes neurais nebulosas.

Um dos objetivos deste trabalho é justamente utilizar esta idéia para a formulação de um algoritmo

de aprendizado para redes neurais nebulosas baseado no AP. Antes, porém, torna-se necessária uma análise da estabilidade e convergência do AP.

3.2 Análise de Convergência do Aprendizado Participativo

Para que um algoritmo seja considerado como um potencial candidato ao treinamento de sistemas baseados em inteligência computacional, ele deve ser, antes de mais nada, um método estável (o que garante que, se o valor inicial do problema é suficientemente próximo da solução, então o algoritmo converge para algum ponto também próximo da solução) e convergente (o que garante que, se o valor inicial do problema é suficientemente próximo da solução, então o algoritmo caminha para a solução).

Tanto o conceito de estabilidade quanto o de convergência, quando demonstrados para um método numérico, mostram que este método converge para uma determinada solução após um determinado número de iterações. Desta forma, é possível verificar que o AP é um método convergente analisando-se o seguinte lema (Yager & Filev, 1993b), que os autores chamam de lema de estabilidade do AP:

Lema 3.1 (Estabilidade do Aprendizado Participativo) *Considere o algoritmo do aprendizado participativo*¹:

$$v^{k+1} = v^k + \alpha_k \rho_k (x^k - v^k) \quad (3.5)$$

onde $\alpha_k \in [0, 1]$ é a taxa de aprendizado (que pode ser constante $\alpha_k = \alpha$ ou variável). Se o índice de compatibilidade ρ_k for definido de tal forma que este pertença ao intervalo $[0, 1]$, então o algoritmo (3.5) é estável.

Demonstração:

A expressão (3.5) pode ser vista como um sistema dinâmico e re-escrita como:

$$v^{k+1} = (1 - \alpha_k \rho_k) v^k + (\alpha_k \rho_k) x^k. \quad (3.6)$$

Se o termo $\alpha_k \rho_k$ é uma constante, i.e., $\alpha_k \rho_k = \alpha \rho$, então o sistema (3.6) é estacionário e será estável se e somente se todos os seus pólos forem localizados no círculo unitário, ou seja:

$$|1 - \alpha \rho| < 1,$$

ou então $-1 < 1 - \alpha \rho < 1$. Assim, para ser estável o termo $\alpha \rho$ deve satisfazer a seguinte condição:

$$0 < \alpha \rho < 2.$$

¹ Note que a equação (3.5) é apenas uma transcrição da equação (2.33).

Como, por definição, a taxa de aprendizado α pertence ao intervalo $[0, 1]$, para garantir estabilidade o índice de compatibilidade deve pertencer ao intervalo:

$$0 < \rho < 2,$$

o que torna o sistema estável se ρ é definido como pertencente ao intervalo $[0, 1]$. Entretanto, o aprendizado participativo é um processo dinâmico, e o índice de compatibilidade, em geral, não é uma constante, mas sim uma variável. Para o caso em que o índice de compatibilidade não é considerado como uma constante, é possível se utilizar a recursão na expressão (3.6) para se obter:

$$v^{k+1} = \prod_{i=0}^k \{(1 - \alpha_i \rho_i) v^0\} + f(\alpha_0, \dots, \alpha_k, \rho_0, \dots, \rho_k, x^0, \dots, x^k) \quad (3.7)$$

onde $f(\cdot)$ é uma função não-linear. Pode-se verificar em (3.7) que a condição $0 < \rho_k < 2$ para cada k também implica na estabilidade do sistema (3.6) para qualquer condição inicial v_0 no caso em que ρ não é uma constante.

Portanto, definindo o índice de compatibilidade ρ_k no intervalo $[0, 1]$, garante que o algoritmo é estável. \square

O lema 3.1 demonstra que o algoritmo AP converge para alguma solução próxima à solução desejada (definição de estabilidade), apesar de não dizer nada a respeito da solução encontrada. Para que se possa analisar o algoritmo AP no sentido de dizer onde se encontra a solução obtida por ele, é necessário demonstrar o conceito de convergência para este algoritmo. Este conceito ainda não foi demonstrado na literatura, porém isso pode ser feito analisando-o como um método de aproximação estocástica (Robbins & Monro, 1951).

O problema de aproximação estocástica originalmente trata da estimação das raízes de uma função desconhecida através de aproximações sucessivas baseadas em observações com ruído (Kushner & Yin, 2003). A aproximação estocástica pode ser interpretada como uma equação a diferenças estocástica onde ajustes recursivos nos parâmetros, dadas algumas condições, fazem com que o sistema apresente uma convergência assintótica. Esta abordagem é utilizada na análise de convergência de algoritmos recursivos, como é o caso do AP.

Considere o seguinte lema (Dvoretzky, 1956; Saridis, 1970):

Lema 3.2 (Aproximação Estocástica) Considere o algoritmo de aproximação estocástica:

$$v^{k+1} = v^k + \zeta_k (z^k - v^k), \quad (3.8)$$

onde $\{\zeta_k\}$ é uma seqüência de ganhos e z^k é a observação da variável v (que se deseja aproximar) corrompida por um ruído finito de média zero e^k , no instante k , isto é:

$$z^k = v + e^k. \quad (3.9)$$

Re-escrevendo-se a equação (3.8) utilizando (3.9), obtém-se:

$$v^{k+1} = v^k + \zeta_k (F_k + e_k), \quad (3.10)$$

onde $F_k = v - v^k$ é um termo de correção e e_k é o termo associado ao ruído. Se a seqüência de ganho $\{\zeta_k\}$ em (3.10) satisfaz

$$\lim_{k \rightarrow \infty} \zeta_k = 0, \quad \sum_{k=1}^{\infty} \zeta_k = \infty, \quad \sum_{k=1}^{\infty} \zeta_k^2 < \infty \quad (3.11)$$

e a seqüência de correção $\{F_k\}$ satisfaz

$$E \left\{ \|v^{k+1} + \zeta_{k+1}(F_k + e_k)\|^2 / v^{k+1} \right\} < E \left\{ \|v^{k+1} + \zeta_{k+1}F_k\|^2 / v^{k+1} \right\} + \zeta_{k+1}^2 E \left\{ \|e_k\|^2 / v^{k+1} \right\}$$

$$E \left\{ \|v^0\|^2 \right\} < \infty \quad ; \quad E \left\{ \|e_k\|^2 \right\} \leq \sigma^2 < \infty, \quad (3.12)$$

onde $E \{ \cdot \}$ representa o valor. Então:

$$Pr \left\{ \lim_{k \rightarrow \infty} \|v^k - v\| = 0 \right\} = 1 \quad (3.13)$$

onde $Pr \{ \cdot \}$ é a função probabilidade, e

$$\lim_{k \rightarrow \infty} E \left\{ \|v^k - v\|^2 \right\} = 0. \quad (3.14)$$

A demonstração deste lema pode ser obtido em (Dvoretzky, 1956). A condição (3.11), sob o ponto de vista prático, é a mais relevante (Spall, 2003). Esta condição fornece um balanceamento cuidadoso no decaimento do ganho ζ_k , fazendo com que este não decaia nem muito rápido nem muito devagar. A primeira parte desta condição ($\lim_{k \rightarrow \infty} \zeta_k = 0$) pode ser interpretada com um efeito suave no termo de correção aleatório. A segunda parte ($\sum_{k=1}^{\infty} \zeta_k = \infty$) fornece um esforço de correção ilimitada e a terceira parte ($\sum_{k=1}^{\infty} \zeta_k^2 < \infty$) garante cancelamento mútuo de erros individuais para grandes números de iterações (Saridis, 1974).

Em particular, a primeira e a segunda partes da condição (3.11) garantem que o ganho se aproxime de zero suficientemente rápido para eliminar os efeitos do ruído quando as iterações se aproximam da solução v , enquanto a terceira parte garante que esta aproximação é suficientemente lenta para evitar convergências prematuras (falsas) do algoritmo (Spall, 2003). A condição (3.12) significa que não

existe acoplamento cruzado ente F_k e e_k e que a busca não é inicializada em uma incerteza infinita sobre os parâmetros (Saridis, 1974).

As expressões (3.13) e (3.14) do lema 3.2 implicam que a seqüência $\{v^k\}$ converge em média quadrática e com probabilidade 1 para v quando $k \rightarrow \infty$.

Utilizando o Lema 3.2 é possível obter o seguinte resultado:

Teorema 3.1 (Convergência do Aprendizado Participativo) *Considere o algoritmo de aprendizado participativo:*

$$v^{k+1} = v^k + \alpha_k \rho_k (x^k - v^k), \quad (3.15)$$

onde $\rho_k \in [0, 1]$ e α_k é tal que:

$$\alpha_k = \frac{\alpha}{k+1}, \quad (3.16)$$

onde $\alpha > 0$ é uma constante.

Se x^k em (3.15) é uma observação de v em um determinado instante k tal que:

$$x^k = v + e^k, \quad (3.17)$$

onde e^k é um ruído aleatório finito de média zero independente de v , então v^k converge em média quadrática e com probabilidade 1 para v quando $k \rightarrow \infty$.

Demonstração:

É possível verificar que o mecanismo de ajuste de crenças dado em (3.15) possui todas as características da forma analítica de um algoritmo de aproximação estocástica apresentada em (3.8). Além disso, a seqüência de ganhos $\{\alpha_k \rho_k\}$, com α_k dado por (3.16) atende às condições (3.11) do Lema 3.2, pois conforme citado em (Spall, 2003), seqüências harmônicas da forma $\{l/(k+c)\}$, $(l, c) > 0$ satisfazem à estas condições.

Verifica-se ainda que os termos e^k e $v - v^k$ são, por hipótese, totalmente desacoplados, o que implica que (3.15) corresponde a um caso particular do teorema de aproximação estocástica de Dvoretzky (Dvoretzky, 1956) e portanto

$$\Pr \left\{ \lim_{k \rightarrow \infty} \|v^k - v\| = 0 \right\} = 1$$

$$\lim_{k \rightarrow \infty} E \left\{ \|v^k - v\|^2 \right\} = 0 \quad , \quad (3.18)$$

o que conclui a prova. \square

Colorário 3.2 (Caso Particular de Convergência do AP) *Seja X uma variável aleatória com função de distribuição dada por $P[X \leq x] = H(x)$ tal que:*

$$E\{X\} = \int_{-\infty}^{\infty} x dH(x) = \mu, \quad (3.19)$$

onde $E\{X\}$ é o valor esperado de X e μ é um valor constante. Suponha que exista uma constante positiva C tal que

$$Pr[|X| \leq C] = \int_{-C}^C dH(x) = 1. \quad (3.20)$$

Seja $\{x^k\}$ uma seqüência de observações de X . De acordo com (3.19) (que garante que o valor esperado de X é um valor fixo finito μ) e (3.20) (que mostra que existem limitantes superior e inferior para os valores das observações de X), cada observação x^k pode ser vista como:

$$x^k = E\{X\} + \delta_k = \mu + \delta_k, \quad (3.21)$$

onde δ_k pode ser visto como um ruído aleatório finito, com a mesma distribuição de X e com valor esperado igual a zero.

Se a seqüência $\{x^k\}$ é processada pelo algoritmo de aprendizado participativo (3.15), com α_k definido por (3.16), tem-se que v^k converge em média quadrática e com probabilidade para o valor esperado de X , i.e.,

$$\begin{aligned} Pr \left\{ \lim_{k \rightarrow \infty} \|v^k - E\{X\}\| = 0 \right\} &= 1 \\ \lim_{k \rightarrow \infty} E \left\{ \|v^k - E\{X\}\|^2 \right\} &= 0. \end{aligned} \quad (3.22)$$

Demonstração:

A partir de (3.21), a expressão (3.15) pode ser re-escrita da seguinte forma:

$$v^{k+1} = v^k + \alpha_k \rho_k (E\{X\} + \delta_k - v^k) = v^k + \alpha_k \rho_k ([E\{X\} - v^k] + \delta_k) \quad (3.23)$$

A expressão (3.23) pode ser vista como um método de aproximação estocástica, onde o termo de correção é $F_k = E\{X\} - v^k$, a seqüência de ganho é $\zeta_k = \alpha_k \rho_k$ e o termo que representa o erro estocástico é $e_k = \delta_k$.

Como já foi mencionado, o termo de ganho $\zeta_k = \alpha_k \rho_k$, com α_k definido em (3.16), atende a todas as condições impostas em (3.12). Também é possível verificar que não existe termo cruzado entre $F_k = v - v^k$ com $v = E\{X\}$ e $e_k = \delta_k$ satisfazendo, portanto, a condição (3.13).

Conforme demonstrado no Teorema 3.1, as crenças v^k convergem em média quadrática e com

probabilidade 1 para um valor v quando $k \rightarrow \infty$. Como, para este caso em particular, tem-se que $v = E\{X\}$, pode-se concluir que, se x^k são observações de uma variável aleatória X conforme definida em (3.19) e (3.20), então v^k converge para o valor esperado de X , em média quadrática e com probabilidade 1, quando $k \rightarrow \infty$, concluindo assim a demonstração do teorema. \square

Para ilustrar os resultados do Colorário 3.2 são apresentados a seguir dois experimentos feitos com variáveis aleatórias com diferentes funções de distribuição e valores iniciais. No primeiro deles, uma seqüência de valores $\{x^k\}$ foi gerada a partir de observações de uma variável aleatória X , com distribuição normal e valor esperado $E\{X\} = \mu = 0.5$. A seqüência $\{x^k\}$ foi apresentada ao AP, para dois valores iniciais distintos, com a finalidade de mostrar que a dinâmica do processo independe do valor inicial. O resultado deste experimento é mostrado na Figura 3.3.

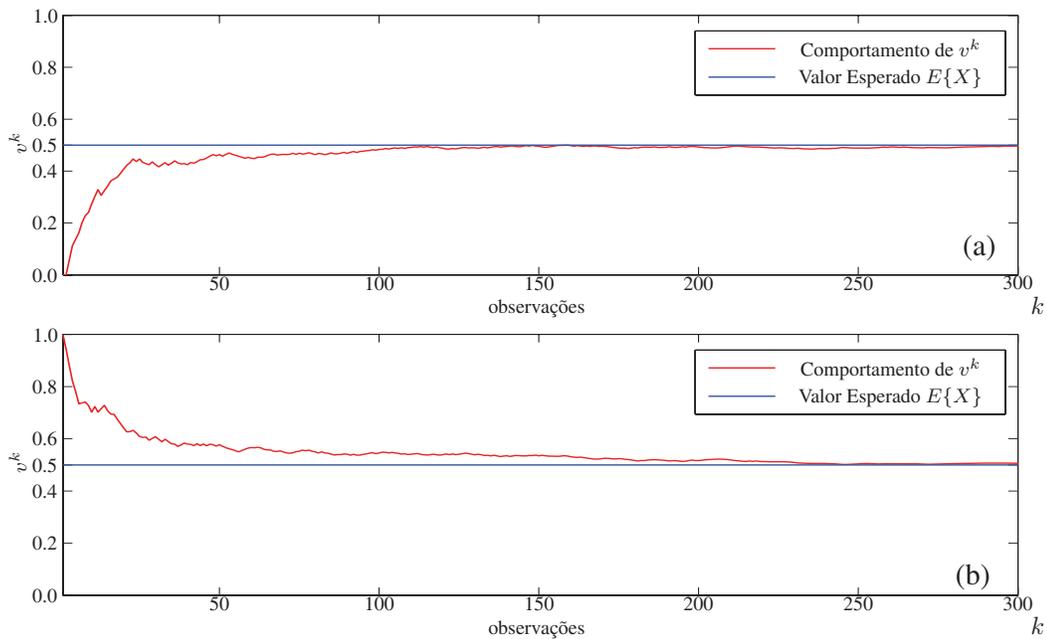


Figura 3.3: Convergência do Aprendizado Participativo para Variável Aleatória com Distribuição Normal: (a) Valor inicial $v^0 = 0$, (b) Valor inicial $v^0 = 1$.

No segundo experimento a seqüência $\{x^k\}$ foi gerada a partir de observações de uma variável aleatória X , com distribuição diamétrica não-central (*noncentral chi-square*) e valor esperado $E\{X\} = \mu = 0.3$, para verificar que o Teorema 3.2 vale para qualquer tipo de distribuição, inclusive as assimétricas. A seqüência $\{x^k\}$ foi novamente apresentada ao AP, para dois valores iniciais distintos. O resultado deste experimento é mostrado na Figura 3.4.

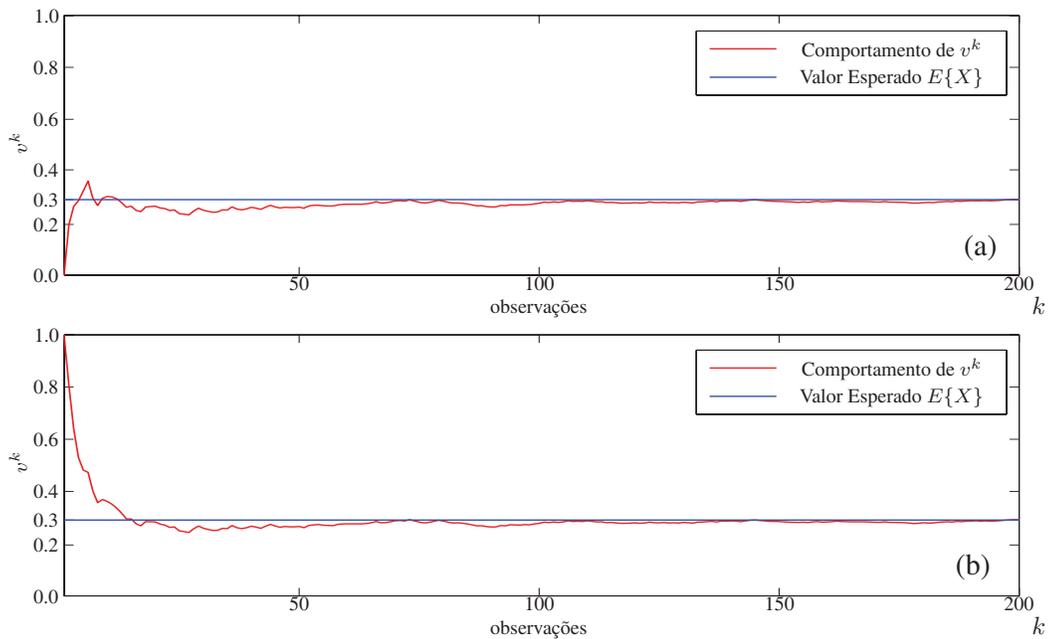


Figura 3.4: Convergência do Aprendizado Participativo para Variável Aleatória com Distribuição Diamétrica Não-Central: (a) Valor inicial $v^0 = 0$, (b) Valor inicial $v^0 = 1$.

Uma vez demonstradas as propriedades de estabilidade e convergência para o aprendizado participativo, já é possível introduzir o procedimento de treinamento para redes neurais nebulosas baseado neste paradigma. Antes, porém, é necessário apresentar a estrutura da rede neural utilizada neste trabalho. Esta tarefa é delegada à próxima seção.

3.3 Estrutura da Rede Neural Nebulosa

Neste trabalho será considerada uma versão estática da estrutura recorrente proposta em (Ballini & Gomide, 2002) e apresentada na seção 2.4 do Capítulo 2, onde as recorrências locais e globais são desconsideradas.

Esta escolha faz com que a rede utilizada neste trabalho seja também um modelo híbrido, cuja estrutura é apresentada na figura 3.5.

A estrutura da rede da figura 3.5 é composta por duas partes. A primeira parte possui duas camadas de neurônios nebulosos e representa um sistema de inferência nebuloso. A primeira destas camadas é formada por neurônios cujas funções de ativação são as funções de pertinência dos conjuntos nebulosos que formam uma partição do espaço de entrada. Estes neurônios também podem

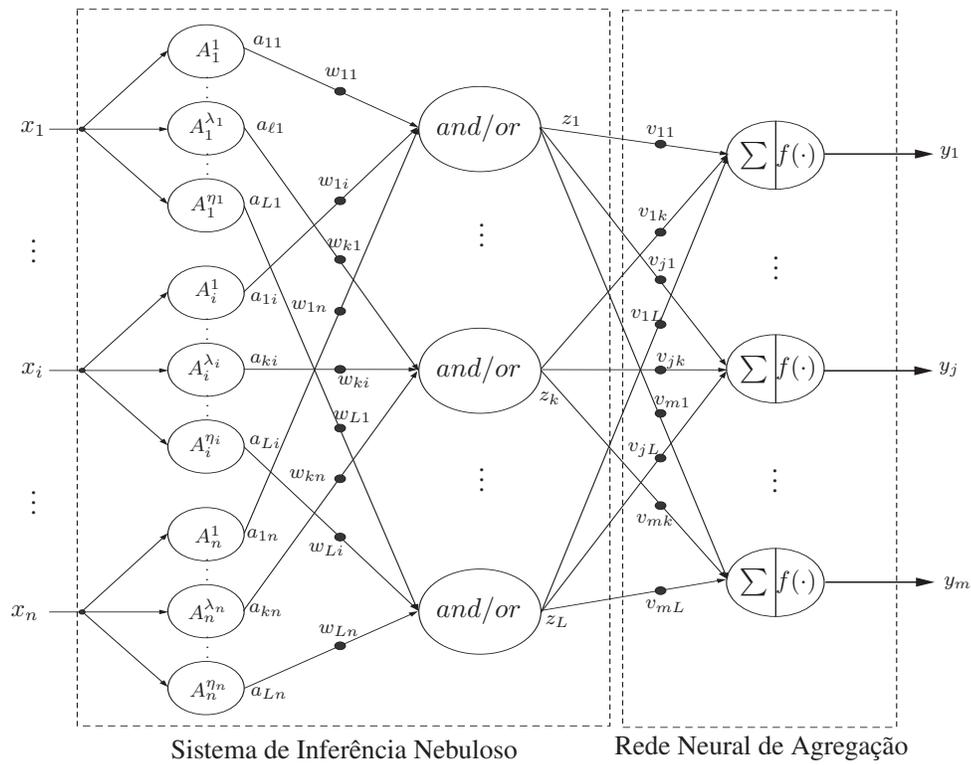


Figura 3.5: Modelo de rede neural nebulosa

ser vistos como neurônios nebulosos do tipo I, sendo que cada um deles possui apenas uma entrada e, portanto, a saída deste neurônio será o grau de pertinência da entrada no conjunto nebuloso que representa o peso.

Para cada dimensão x_i de um vetor de entrada n -dimensional \mathbf{x} existem η_i neurônios nebulosos do tipo I, sendo que o peso nebuloso associado a esta entrada é dado pelo conjunto nebuloso $A_i^{\lambda_i}$, $\lambda_i = 1, \dots, \eta_i$ cujas funções de pertinência correspondem à granularização da variável x_i . Assim, as saídas da primeira camada intermediária são os graus de pertinência associados aos valores das entradas, isto é, $a_{\ell i} = A_i^{\lambda_i}(x_i)$, $i = 1, \dots, n$ e $\ell = 1, \dots, L$; onde L é o número de neurônios na segunda camada de neurônios nebulosos.

Os neurônios nebulosos da segunda camada podem ser escolhidos como sendo um dos dois tipos de neurônios lógicos apresentados na seção anterior. Caso sejam escolhidos neurônios lógicos do tipo *AND*, a função implementada nesta camada passa a ser a função *and* e é computada segundo a expressão (2.28).

É possível verificar que, para esta escolha, a estrutura da rede codifica um conjunto de regras

nebulosas se-então $R = \{R_\ell, \ell = 1, \dots, L\}$ da forma:

$$R_\ell : SE (x_1 \text{ é } A_1^{\lambda_1} \text{ OU } w_{\ell 1}) \dots E (x_i \text{ é } A_i^{\lambda_i} \text{ OU } w_{\ell i}) \dots E (x_n \text{ é } A_n^{\lambda_n} \text{ OU } w_{\ell n}) \text{ ENTÃO, } z \text{ is } z_\ell. \quad (3.24)$$

Já para o caso em que neurônios lógicos do tipo *OR* são escolhidos para implementar a segunda camada da rede, a função implementada é a função *or*, sendo que saída de cada um destes neurônios passa a ser computada segundo a expressão (2.30). Desta forma, a rede passa a codificar o seguinte conjunto de regras nebulosas se-então $R = \{R_\ell, \ell = 1, \dots, L\}$:

$$R_\ell : SE (x_1 \text{ é } A_1^{\lambda_1} \text{ E } w_{\ell 1}) \dots \text{ OU } (x_i \text{ é } A_i^{\lambda_i} \text{ E } w_{\ell i}) \dots \text{ OU } (x_n \text{ é } A_n^{\lambda_n} \text{ E } w_{\ell n}) \text{ ENTÃO, } z \text{ is } z_\ell. \quad (3.25)$$

É possível verificar que existe uma forte correspondência entre a estrutura da primeira parte da rede neurofuzzy e um conjunto de regras fuzzy, ou alternativamente, uma base de regras fuzzy. Além disto, o processamento do esquema induzido pela estrutura da rede está de acordo com os princípios da teoria de conjuntos fuzzy e de raciocínio aproximado (Gomide & Pedrycz, 2007), sendo que a interpretação para este tipo de modelo é amplamente conhecida e pode ser encontrada em (Lin & Lee, 1996).

A segunda parte é formada por uma rede neural clássica de apenas uma camada que, conforme mencionado anteriormente, tem o propósito de agregar as saídas do sistema de inferência nebuloso, fornecendo uma saída para a rede. Assim, a saída do j -ésimo neurônio de saída é dada por:

$$y_j = f \left(\sum_{\ell=1}^L z_\ell v_{j\ell} \right), \quad (3.26)$$

onde $f(\cdot)$ é uma função de ativação monotônica contínua. Neste trabalho tomou-se como função de ativação dos neurônios da camada de saída a função sigmoideal, i.e., $f(\xi) = (1 + \exp(-\xi))^{-1}$.

Assim sendo, a dinâmica da rede neural nebulosa apresentada na Figura 3.5 pode ser resumida da seguinte forma:

Passo 1: Calcula-se a matriz $A = [a_{\ell i}]$, $\ell = 1, \dots, L$, $i = 1, \dots, n$, onde $a_{\ell i} = A_i^{\lambda_i}(x_i)$ é a saída do λ_i -ésimo neurônio nebuloso do tipo I da entrada x_i que vai na direção do ℓ -ésimo neurônio lógico na segunda camada;

Passo 2: Calcule-se o vetor $\mathbf{z} = [z_\ell]$, $\ell = 1, \dots, L$ da seguinte forma:

$$z_\ell = \mathbf{T}_{i=1}^n (a_{\ell i} \text{ s } w_{\ell i}) \text{ Para neur\u00f4nios do tipo AND, ou} \quad (3.27)$$

$$z_\ell = \mathbf{S}_{i=1}^n (a_{\ell i} \text{ t } w_{\ell i}) \text{ para neur\u00f4nios do tipo OR;} \quad (3.28)$$

Passo 3: Finalmente calcula-se o vetor de sa\u00edda $\mathbf{y} = [y_j]$, $j = 1, \dots, m$ atrav\u00e9s da express\u00e3o (3.26).

Para o caso especial, onde se escolhe a fun\u00e7\u00e3o min para representar a t-norma na express\u00e3o (3.27), esta pode ser re-escrita na forma:

$$z_\ell = \mathbf{a}_\ell \bullet \mathbf{w}_\ell^T, \quad (3.29)$$

onde $\mathbf{a}_\ell = [a_{\ell 1}, \dots, a_{\ell n}]$ \u00e9 o vetor contendo a ℓ -\u00e9sima linha da matriz A , $\mathbf{w}_\ell = [w_{\ell 1}, \dots, w_{\ell n}]$ \u00e9 o vetor contendo a ℓ -\u00e9sima linha da matriz W e \mathbf{w}_ℓ^T denota o transposto de \mathbf{w}_ℓ . Da mesma forma, se a fun\u00e7\u00e3o max for escolhida para representar a s-norma na equa\u00e7\u00e3o (3.28), esta pode ser re-escrita na forma:

$$z_\ell = \mathbf{a}_\ell \circ \mathbf{w}_\ell^T. \quad (3.30)$$

A pr\u00f3xima se\u00e7\u00e3o apresenta o algoritmo de treinamento com aprendizado participativo para a classe de redes neurais nebulosas apresentadas nesta se\u00e7\u00e3o.

3.4 Treinamento de Redes Neurais Nebulosas com AP

Esta se\u00e7\u00e3o prop\u00f5e um novo algoritmo de treinamento baseado no aprendizado participativo (AP) para a classe de redes neurais nebulosas h\u00edbridas da Figura 3.5, apresentadas na se\u00e7\u00e3o anterior. Vale ressaltar que o algoritmo aqui apresentado pode ser estendido para uma grande variedade de classes de redes neurais nebulosas, bem como para alguns sistemas neuro-nebulosos.

O algoritmo proposto \u00e9 dividido em tr\u00eas etapas:

Fase 1: Gera\u00e7\u00e3o das fun\u00e7\u00f5es de pertin\u00eancia que representam os pesos nebulosos associados \u00e0 primeira camada de neur\u00f4nios;

Fase 2: Atualiza\u00e7\u00e3o dos pesos associados \u00e0 rede neural cl\u00e1ssica que representa o agregador n\u00e3o-linear de sa\u00edda;

Fase 3: Atualiza\u00e7\u00e3o dos pesos relacionados aos neur\u00f4nios l\u00f3gicos AND/OR que representam o sistema de infer\u00eancia nebuloso da rede.

Cada uma destas etapas \u00e9 descrita nas se\u00e7\u00f5es a seguir.

3.4.1 Geração das Funções de Pertinência

A geração das funções de pertinência que implementam os pesos dos neurônios nebulosos do tipo I na primeira camada da rede apresentada na Figura 3.5 se dá por meio do uso do algoritmo de agrupamento nebuloso *fuzzy c-means* (Bezdek, 1981), sendo que tanto informações relativas ao espaço de entrada quanto ao espaço de saída são incluídas neste agrupamento. De forma geral, se o espaço de entrada tem dimensão n e o espaço de saída tem dimensão m , o agrupamento é feito no espaço $n + m$ -dimensional, sendo que cada um dos centros de grupo encontrados neste espaço é projetado no espaço de entrada. O resultado desta projeção é o valor modal das funções de pertinência nos seus respectivos universos de discurso.

Um exemplo da geração das funções de pertinência para um problema onde o espaço de entrada é bidimensional, $\mathbf{x} = [x_1, x_2] \in \mathbb{R}^2$, e o espaço de saída é unidimensional, $\mathbf{y} = y \in [0, 1]$, com funções de pertinência Gaussianas é apresentado na Figura 3.6. As dispersões das Gaussianas são ajustadas de forma a se obter um quadro cognitivo no espaço de entrada.

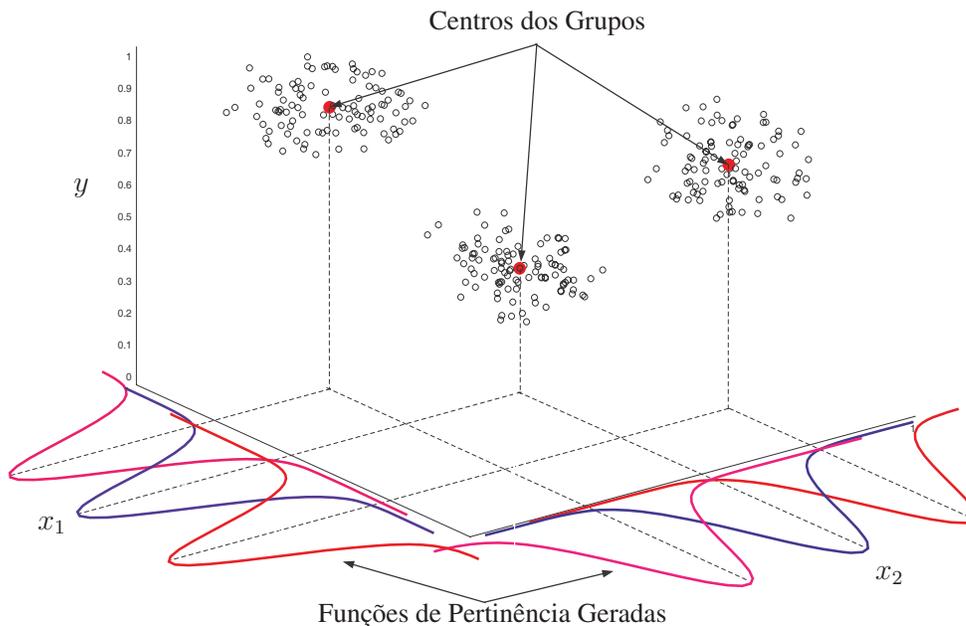


Figura 3.6: Exemplo de Geração das Funções de Pertinência para $\mathbf{x} \in \mathbb{R}^2$ e $y \in [0, 1]$.

3.4.2 Atualização dos Pesos Associados à Rede Neural de Agregação

Assim como no capítulo anterior, aqui os pesos associados à rede neural de agregação (uma rede neural clássica que faz a agregação ponderada das saídas do sistema de inferência nebuloso representado pela primeira parte da rede) são ajustados através do método do gradiente. Assim sendo, a variação dos pesos $\Delta v_{j\ell}^k$ associados à esta camada para um dado par de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$ em um determinado instante k , pode ser calculada da seguinte forma:

$$\Delta v_{j\ell}^k = \alpha_1 (y_j^k - \hat{y}_j^k) f'(\psi_j^k) z_\ell^k \quad (3.31)$$

onde \hat{y}_j^k é a saída obtida para o j -ésimo neurônio na camada de saída, y_j^k é a saída desejada para este mesmo neurônio, z_ℓ^k é a saída obtida para o ℓ -ésimo neurônio lógico na segunda camada de neurônios nebulosos, $v_{j\ell}^k$ é o peso de conexão entre a j -ésima unidade de saída e a ℓ -ésima unidade na segunda camada de neurônios nebulosos, $f'(\psi_j^k) = f(\psi_j^k)(1 - f(\psi_j^k))$ é a derivada da função de ativação dos neurônios na camada de saída avaliada em ψ_j^k , $\psi_j^k = \sum_{\ell=1}^L (v_{j\ell}^k z_\ell^k)$, e $\alpha_1 \in [0, 1]$ é a taxa de aprendizado relativa a esta rede.

Assim, a atualização dos pesos é calculada utilizando $\Delta v_{j\ell}^k$ para o instante $k + 1$ de acordo com a seguinte expressão:

$$v_{j\ell}^{k+1} = v_{j\ell}^k + \Delta v_{j\ell}^k \quad (3.32)$$

3.4.3 Atualização dos Pesos Associados ao Sistema de Inferência Nebuloso

Um aspecto importante no desenvolvimento do algoritmo de treinamento proposto neste trabalho está relacionado com a forma de se ajustar os pesos referentes aos neurônios lógicos *AND/OR* na parte da rede que representa o sistema de inferência nebuloso da rede da Figura 3.5. Nesta seção, desenvolve-se uma forma alternativa às propostas da literatura para se proceder os ajustes dos pesos $w_{\ell i}$'s, utilizando-se o aprendizado participativo.

Considerando-se as idéias apresentadas no início deste capítulo, para que o AP possa ser aplicado no ajuste dos pesos $w_{\ell i}$'s, é necessário encontrar inicialmente uma matriz $G^k = [g_{\ell i}^k]$, associada a um par de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$, de tal forma que, propagando-se a entrada \mathbf{x}^k através da rede, a saída obtida $\hat{\mathbf{y}}^k$ para esta rede seja exatamente o vetor \mathbf{y}^k .

Conforme mencionado anteriormente, quando o operador *min* é escolhido para implementar a *t*-norma nos neurônios *AND* ou o operador *max* é utilizado como *s*-norma nos neurônios *OR*, a matriz G^k pode ser vista como uma matriz relacional (expressões (3.29) e (3.30), respectivamente), e é possível encontrá-la utilizando-se para isso a solução do problema de estimação apresentada na seção 2.3.8.

Para isso, é necessário inicialmente se obter os valores de saída z_ℓ^k dos neurônios lógicos na segunda camada da rede, associados com a saída desejada \mathbf{y}^k , de tal forma que, quando propagados através desta camada (expressão (3.26)), se obtenha como saída da rede novamente o vetor \mathbf{y}^k . Assim, a estimação do vetor $\mathbf{z}^k = [z_\ell^k]$, $\ell = 1, \dots, L$, deve ser feita a partir de \mathbf{y}^k e da matriz de pesos V^{k+1} da camada de saída, já atualizada conforme apresentado na seção anterior (expressão 3.32), resolvendo-se o problema de mínimos quadráticos linear restrito:

$$\min_{z_\ell^k} \left\{ \frac{1}{2} \left\| (V^{k+1})^T \mathbf{z}^k - f^{-1}(\mathbf{y}^k) \right\|^2 \right\} \text{ sujeito a } 0 \leq z_\ell^k \leq 1 \quad (3.33)$$

onde $(V^{k+1})^T$ denota a transposta de V^{k+1} e $f^{-1}(\cdot)$ é a inversa da função f em (3.26).

A matriz G^k pode então ser calculada através da solução do problema de estimação visto anteriormente. Este cálculo depende da escolha do tipo de neurônio lógico que irá compor a segunda camada da rede. Assim, se neurônios lógicos do tipo *AND* forem escolhidos para implementar esta camada, a matriz G^k é calculada da seguinte forma:

$$\mathbf{g}_\ell^k = (\mathbf{a}_\ell^k \beta z_\ell^k) = \inf (c \in [0, 1] \mid \mathbf{a}_\ell^k \cdot c \geq z_\ell^k), \quad (3.34)$$

onde $\mathbf{a}_\ell^k = [a_{\ell 1}^k, \dots, a_{\ell n}^k]$ é a ℓ -ésima linha da matriz A^k (obtida através da propagação do vetor de entrada \mathbf{x}^k através da primeira camada de neurônios nebulosos) e $\mathbf{g}_\ell^k = [g_{\ell 1}^k, \dots, g_{\ell n}^k]$ é a ℓ -ésima linha da matriz G^k . O operador β é definido em (2.19).

Vale lembrar que a solução dada por (3.34) é uma solução minimal. Isto implica que quando a equação (3.34) associa um valor igual a zero para um dos elementos da matriz G^k , isto é, $g_{\ell i}^k = 0$, este elemento pode ser substituído por qualquer valor no intervalo $[0, 1]$ que o resultado da composição inf-s $\mathbf{a}_\ell^k \bullet \mathbf{g}_\ell^k$ continuará sendo igual a z_ℓ^k , conforme já foi discutido anteriormente. Isto acontece porque $g_{\ell i}^k = 0$ implica que $a_{\ell i}^k \geq z_\ell^k$ e, portanto, o resultado da composição inf-s para este elemento independe do valor de $g_{\ell i}^k$.

No âmbito do processo de treinamento, este fato pode ser interpretado como uma forma de se dizer que o peso $w_{\ell i}^k$ associado ao elemento $g_{\ell i}^k = 0$ já está ajustado para o par de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$ e, portanto, não deve ser modificado. Para garantir que isto aconteça, cria-se uma nova matriz para representar a informação que chega à rede, denotada por \tilde{G}^k , onde associa-se o valor do peso $w_{\ell i}^k$ ao elemento $\tilde{g}_{\ell i}^k$ quando $g_{\ell i}^k = 0$, ou seja:

$$\tilde{g}_{\ell i}^k = \begin{cases} w_{\ell i}^k, & \text{se } g_{\ell i}^k = 0 \\ g_{\ell i}^k, & \text{caso contrário} \end{cases} \quad (3.35)$$

Já para o caso em que neurônios do tipo *OR* são escolhidos para implementar a segunda camada da rede, o cálculo de G^k é feito de forma similar, segundo a equação:

$$\mathbf{g}_\ell^k = (\mathbf{a}_\ell^k \varphi z_\ell^k) = \sup (c \in [0, 1] \mid \mathbf{a}_\ell^k t c \leq z_\ell^k), \quad (3.36)$$

onde o operador φ é definido em (2.15).

Para este caso, a solução obtida em (3.36) é maximal e, portanto, quando $g_{\ell i}^k = 1$ o peso associado a este elemento não precisa ser ajustado. A matriz \tilde{G}^k pode então ser calculada da seguinte forma:

$$\tilde{g}_{\ell i}^k = \begin{cases} w_{\ell i}^k, & \text{se } g_{\ell i}^k = 1 \\ g_{\ell i}^k, & \text{caso contrário} \end{cases} \quad (3.37)$$

É possível verificar que, se a matriz de pesos $W^k = [w_{\ell i}^k], \ell = 1, \dots, L, i = 1, \dots, n$ for substituída pela matriz \tilde{G}^k calculada conforme apresentado e o vetor de entrada \mathbf{x}^k for propagado por toda a rede, a saída obtida $\hat{\mathbf{y}}^k$ será exatamente igual a saída desejada \mathbf{y}^k , o que indica que a matriz \tilde{G}^k também codifica a informação que o par de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$ fornece para a rede. Assim, o grau de compatibilidade ρ_k associado à observação $[\mathbf{x}^k, \mathbf{y}^k]$ pode ser calculado da seguinte forma:

$$\rho_k = \frac{1}{L \times n} \sum_{\ell=1}^L \sum_{i=1}^n |\tilde{g}_{\ell i}^k - w_{\ell i}^k|. \quad (3.38)$$

Após ser calculado o índice de compatibilidade ρ_k associado ao par de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$, pode-se calcular a variação dos pesos relacionados ao sistema de inferência nebuloso representado pela primeira parte da rede, de acordo com a seguinte regra:

$$\Delta w_{\ell i}^k = \alpha_2 (\rho_k)^{1-\gamma_k} (\tilde{g}_{\ell i}^k - w_{\ell i}^k), \quad (3.39)$$

onde γ_k é o índice de alerta calculado conforme a expressão (2.35) e α_2 é a taxa de aprendizado.

Por fim, a atualização dos pesos desta parte da rede é feita utilizando-se $\Delta w_{\ell i}^k$ de acordo com:

$$w_{\ell i}^{k+1} = w_{\ell i}^k + \Delta w_{\ell i}^k. \quad (3.40)$$

A próxima seção fornece uma visão geral dos procedimentos de treinamento propostos, tanto para redes com neurônios lógicos do tipo *AND* quanto do tipo *OR* na segunda camada.

3.4.4 Procedimentos de Treinamento com AP

Considere uma seqüência de pares de treinamento da por $[\mathbf{x}^k, \mathbf{y}^k], k = 1, \dots, P$, onde P é o número total de pares. A seguir, são apresentados os procedimentos gerais para o processo de treinamento proposto neste trabalho, para os dois tipos de neurônios lógicos apresentados anteriormente. Vale ressaltar que, para ser possível a aplicação do AP a este tipo de rede, é necessário utilizar-se

a função min para implementar a t-norma nos neurônios *AND* e a função max para implementar a s-norma nos neurônios *OR*.

Algoritmo de treinamento baseado no AP para redes baseadas em neurônios lógicos do tipo *AND*

O procedimento completo baseado no aprendizado participativo para redes onde neurônios lógicos do tipo *AND* utilizados para implementar a segunda camada de neurônios nebulosos é o seguinte:

1. Granularizar o espaço de entrada e obter as funções de pertinência conforme seção 3.4.1;
2. Inicializar os pesos $w_{\ell i}, \ell = 1, \dots, L, i = 1, \dots, n$ aleatoriamente, no intervalo $[0, 1]$;
3. Inicializar os pesos $v_{j\ell}, j = 1, \dots, m$ aleatoriamente, no intervalo $[-0.1, 0.1]^2$;
4. Inicializar o índice de alerta em $\gamma_k = 1$ ¹;
5. Para cada um dos pares de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$ fazer:
 - 5.1. Calcular os valores $a_{\ell i}^k, \ell = 1, \dots, L, i = 1, \dots, n$ propagando-se o vetor de entrada \mathbf{x}^k através da primeira camada de neurônios nebulosos;
 - 5.2. Obter o vetor $\hat{\mathbf{z}}^k = [\hat{z}_\ell^k], \ell = 1, \dots, L$, cujas componentes são as saídas obtidas para os neurônios lógicos tipo *AND* através da equação relacional:

$$\hat{z}_\ell^k = \mathbf{a}_\ell^k \bullet (\mathbf{w}_\ell^k)^T ; \quad (3.41)$$

onde $\mathbf{w}_\ell^k, \ell = 1, \dots, L$ é a ℓ -ésima linha da matriz de pesos relacionados à primeira parte da rede da Figura 3.5.

- 5.3. Calcular a saída da rede $\hat{\mathbf{y}}^k = [y_j^k]$, utilizando a equação:

$$\hat{y}_j^k = f \left(\sum_{\ell=1}^L \hat{z}_\ell^k v_{j\ell}^k \right), j = 1, \dots, m.; \quad (3.42)$$

² A inicialização dos pesos no intervalo $[-0.1, 0.1]$ tem o propósito de evitar o problema de paralisia da rede neural, pois caso o resultado da ponderação entre pesos e entradas seja um valor muito alto, este coincide com a região linear da função de transferência do neurônio. Como nesta região a derivada em qualquer ponto é igual a zero, o gradiente também torna-se nulo e conseqüentemente não há ajuste de pesos

¹ Como a rede ainda não possui crenças a respeito do ambiente, a inicialização de γ_k em 1 faz com que esta aprenda com tudo que lhe é ensinado, até que um conhecimento seja obtido.

5.4. Calcular o erro de aproximação $\mathbf{e}^k = [e_j^k]$ $j = 1, \dots, m$, de acordo com:

$$e_j^k = y_j^k - \hat{y}_j^k, \quad (3.43)$$

onde y_j^k é a j -ésima componente do vetor de saídas desejadas \mathbf{y}^k ;

5.5. Calcular o vetor gradiente de erro da seguinte forma:

$$\delta_j^k = e_j^k f'(\psi_j^k), \quad j = 1, \dots, m, \quad \ell = 1, \dots, L, \quad (3.44)$$

onde $f'(\psi_j^k)$ é a derivada de $f(\cdot)$ em $\psi_j^k = \sum_{\ell=1}^m (v_{j\ell}^k z_\ell^k)$;

5.6. Calcular a variação dos pesos relacionados à camada de saída $v_{j\ell}^k$ de acordo com:

$$\Delta v_{j\ell}^k = \alpha_1 \delta_j^k z_\ell^k, \quad j = 1, \dots, m, \quad \ell = 1, \dots, L, \quad (3.45)$$

onde α_1 é a taxa de treinamento para a camada de saída;

5.7. Atualizar os pesos da camada de saída de acordo com:

$$v_{j\ell}^{k+1} = v_{j\ell}^k + \Delta v_{j\ell}^k; \quad (3.46)$$

5.8. Calcular a saída desejada para os neurônios lógicos *AND* $\mathbf{z}^k = [z_\ell^k]$, $\ell = 1, \dots, L$ relacionada com a saída desejada da rede $\mathbf{y}^k = [y_j^k]$, $j = 1, \dots, m$ resolvendo o seguinte problema de otimização:

$$\min_{z_\ell^k} \left\{ \frac{1}{2} \left\| (V^{k+1})^T \mathbf{z}^k - f^{-1}(\mathbf{y}^k) \right\|^2 \right\} \text{ sujeito a } 0 \leq z_\ell^k \leq 1, \quad (3.47)$$

onde $V^{k+1} = [v_{j\ell}^{k+1}]$, $j = 1, \dots, m$, $\ell = 1, \dots, L$;

5.9. Calcular a matriz G^k encontrando a solução do problema de estimação dado por:

$$\mathbf{g}_\ell^k = \mathbf{a}_\ell^k \beta z_\ell^k; \quad (3.48)$$

5.10. Determinar a matriz \tilde{G}^k que representa a informação que chega na rede trazida pela observação $[\mathbf{x}^k, \mathbf{y}^k]$ de acordo com:

$$\tilde{g}_{\ell i}^k = \begin{cases} w_{\ell i}^k, & \text{se } g_{\ell i}^k = 0 \\ g_{\ell i}^k, & \text{caso contrário} \end{cases}, \quad \ell = 1, \dots, L, \quad i = 1, \dots, n; \quad (3.49)$$

5.11. Calcular o índice de compatibilidade ρ_k entre as matrizes $W^k = [w_{\ell i}^k]$ e $\tilde{G}^k = [\tilde{g}_{\ell i}^k]$, $\ell = 1, \dots, L$, $i = 1, \dots, n$, de acordo com:

$$\rho_k = \frac{1}{L \times n} \sum_{\ell=1}^L \sum_{i=1}^n |\tilde{g}_{\ell i}^k - w_{\ell i}^k|; \quad (3.50)$$

5.12. Calcular a variação dos pesos relacionados à segunda camada da rede $w_{\ell i}^k$ de acordo com:

$$\Delta w_{\ell i}^k = \alpha_2 (\rho_k)^{1-\gamma_k} (\tilde{g}_{\ell i}^k - w_{\ell i}^k), \quad (3.51)$$

onde α_2 é a taxa de treinamento para esta camada, $\ell = 1, \dots, L$, $i = 1, \dots, n$;

5.13. Atualizar os pesos relacionados ao sistema de inferência nebuloso de acordo com a regra:

$$w_{\ell i}^{k+1} = w_{\ell i}^k + \Delta w_{\ell i}^k, \quad \ell = 1, \dots, L, i = 1, \dots, n; \quad (3.52)$$

5.14. Atualizar o índice de alerta γ_k de acordo com a expressão (3.53), onde $\lambda \in [0, 1]$ é uma constante que controla a taxa com a qual a alerta é ajustada,

$$\gamma_{k+1} = \gamma_k + \lambda((1 - \rho_{k+1}) - \gamma_k); \quad (3.53)$$

6. Repetir os passos 5.1 - 5.14 até que uma tolerância de erro ou um número máximo de iterações seja atingido.

Algoritmo de treinamento baseado no AP para redes baseadas em neurônios lógicos do tipo OR

O procedimento baseado no aprendizado participativo para redes compostas por neurônios lógicos do tipo OR é muito similar ao apresentado anteriormente para redes com neurônios AND, sendo que para obter-se este novo algoritmo é necessário apenas proceder as seguintes alterações:

1. Substituir a equação utilizada no cálculo de \hat{z}_{ℓ}^k (3.41) por:

$$\hat{z}_{\ell}^k = \mathbf{a}_{\ell}^k \circ (\mathbf{w}_{\ell}^k)^T; \quad (3.54)$$

2. Substituir a equação utilizada para calcular os elementos da matriz G (3.48) por:

$$\mathbf{g}_{\ell}^k = \mathbf{a}_{\ell}^k \varphi z_{\ell}^k; \quad (3.55)$$

3. Substituir a equação condicional utilizada na determinação da matriz G (3.49) por:

$$\tilde{g}_{\ell i}^k = \begin{cases} w_{\ell i}^k, & \text{se } g_{\ell i}^k = 1 \\ g_{\ell i}^k, & \text{caso contrário} \end{cases}, \quad \ell = 1, \dots, L, i = 1, \dots, n; \quad (3.56)$$

É possível verificar que nos algoritmos apresentados nesta seção o conhecimento que já se tem a respeito do problema que está sendo tratado (armazenado nos pesos $w_{\ell i}$) é levado em consideração no ajuste dos pesos da rede da Figura 3.5. Assim, quando um par de treinamento $[\mathbf{x}, \mathbf{y}]$ fornece à rede uma informação conflitante com o conhecimento já armazenado, estes dados terão pouca influência no ajuste dos pesos $w_{\ell i}$. Por outro lado, se o conhecimento fornecido por $[\mathbf{x}, \mathbf{y}]$ (codificado pela matriz G) for próxima o suficiente do conhecimento já adquirido, então os dados $[\mathbf{x}, \mathbf{y}]$ terão forte influência no ajuste dos pesos $w_{\ell i}$. Estes algoritmos podem ser, portanto, classificados como algoritmos de aprendizado participativo.

Para verificar a eficácia dos algoritmos propostos estes foram utilizados no treinamento de redes neurais nebulosas da Figura 3.5 com o intuito de resolver problemas de modelagem de sistemas dinâmicos não-lineares, bem como de previsão de séries temporais. As vantagens da característica participativa do algoritmo foram verificadas através da utilização de dados não-ideais no treinamento e posterior análise de estudos comparativos desta abordagem com métodos alternativos propostos na literatura. Os resultados destas análises são apresentados no Capítulo 5.

3.5 Resumo

Este capítulo apresentou o procedimento de treinamento proposto neste trabalho, o qual é baseado no aprendizado participativo para uma classe de redes híbridas neurais nebulosas. Inicialmente foi feita uma análise sobre os critérios de estabilidade do aprendizado participativo para, em seguida, se discutir a respeito da convergência do método. Os algoritmos propostos para redes baseadas em neurônios lógicos do tipo *AND* quanto do tipo *OR* foram detalhados e discutidos.

Capítulo 4

Generalização de Neurônios Nebulosos

4.1 Introdução

Este capítulo apresenta o desenvolvimento estrutural para uma nova classe de redes neurais nebulosas baseada em generalizações dos neurônios lógicos apresentados no Capítulo 2. O objetivo destas generalizações é dotar os neurônios nebulosos propostos neste trabalho de uma importante característica análoga a dos neurônios biológicos, a plasticidade neuronal. Esta característica, muito útil ao sistema nervoso, permite o desenvolvimento de alterações funcionais e estruturais em resposta à experiência e de adaptação a condições mutantes e a estímulos repetidos. Com isso o neurônio biológico é capaz modificar a forma de processamento de seus circuitos internos dependendo da necessidade ou da situação em que se encontra (Kandel et al., 2000).

Até onde vai o conhecimento do autor, a literatura não sugere um tipo de neurônio artificial capaz de emular esta característica do neurônio biológico. Assim, os desenvolvimentos propostos neste capítulo visam gerar neurônios nebulosos que possam assumir diferentes formas de processar sinais de entrada e adaptar seus pesos de conexão, dependendo da necessidade do problema. Estes novos neurônios podem ser vistos como generalizações dos neurônios lógicos *AND* e *OR* onde a inserção de mais um parâmetro, ajustado pelo algoritmo de treinamento, faz com o tipo de processamento interno do neurônio possa variar de um neurônio puramente *AND* a um neurônio puramente *OR*, incluindo realizações entre estes dois extremos, a fim de melhor se adaptar aos dados de treinamento.

Conforme foi visto na seção 2.4 do capítulo 2, os neurônios lógicos *AND* e *OR* podem ser vistos como transformações não-lineares multivariáveis entre hipercubos unitários, i.e., $[0, 1]^n \rightarrow [0, 1]$, onde o processamento ocorre em dois níveis. No primeiro nível, os sinais de entrada (graus de pertinência) $\mathbf{a} = [a_1, a_2, \dots, a_n]$ são individualmente combinados com os pesos de conexão $\mathbf{w} = [w_1, w_2, \dots, w_n]$, $\mathbf{a}, \mathbf{w} \in [0, 1]^n$, ou seja, no primeiro nível são realizadas transformações $L_1 : (a_i, w_i) \rightarrow [0, 1]$, $i = 1, \dots, n$. No segundo nível, uma geração de agregação global L_2 é

feita sobre todos os resultados da combinação das transformações do primeiro nível, isto é:

$$z = L_2 [L_1(a_1, w_1), L_1(a_2, w_2), \dots, L_1(a_n, w_n)]. \quad (4.1)$$

Em um neurônio lógico do tipo *AND* L_1 é executada através da realização do operador lógico *or* através de uma s-norma, $L_1 = or$ enquanto L_2 é executada através da realização do operador lógico *and* através de uma t-norma, $L_2 = and$. Já a estrutura do neurônio lógico do tipo *OR* possui uma estrutura dual ao do neurônio *AND*, ou seja, para um neurônio *OR* tem-se que $L_1 = and$ e $L_2 = or$.

Neste capítulo, sugerem-se neurônios nebulosos utilizando os conceitos de uninormas e nullnormas na implementação das transformações L_1 e/ou L_2 , a fim de que uma rede neural nebulosa que utilize este tipo de neurônio seja mais geral do que àquelas que os utilizem neurônios clássicos e lógicos discutidos anteriormente.

Conforme será apresentado, estes novos neurônios são capazes de se comportar como qualquer um dos dois tipos de neurônios lógicos *AND* e *OR*, dependendo da necessidade do problema a ser tratado, a partir do ajuste de um parâmetro adicional que controla o tipo que o neurônio assume na rede. Isto faz com que, para uma rede formada por estes novos tipos de neurônios, o conhecimento a respeito do problema que se deseja tratar seja armazenado não somente nos pesos de conexão, mas também na própria forma dos neurônios e, conseqüentemente, na estrutura da rede.

Os neurônios nebulosos propostos neste trabalho são mais gerais que os neurônios nebulosos alternativos discutidos na literatura. A rede neural considerada espelha em sua estrutura um conjunto de regras se-então que usam diferentes conectivos em seus antecedentes, diferentemente de todas as abordagens propostas até o momento, em que só é possível representar um sistema de inferência nebuloso que utilize apenas um tipo de conectivo nos antecedentes (Rutkowski & Cpalka, 2005; Hell et al., 2007b, 2008a).

Nas duas próximas seções deste capítulo, desenvolvem-se os procedimentos através dos quais é possível construir as duas unidades de processamento propostas neste trabalho: os Unineurons e os Nullneurons. Como o próprio nome já diz, os unineurons são baseados no conceito de uninorma enquanto os nullneurons no conceito de nullnorma. Após o detalhamento dos processos construtivos, as seções seguintes se encarregam de sugerir estruturas de redes baseadas nos unineurons e nullneurons, assim como algoritmos de treinamento para estas redes, provendo assim um novo tipo de sistema baseado em inteligência computacional para modelagem de sistemas dinâmicos não-lineares, principal foco deste trabalho.

4.2 Neurônio Nebuloso Baseado em Uninormas: O Unineuron

Conforme discutido anteriormente, as uninormas apresentadas na definição 2.4 podem ser utilizadas para implementar a transformação local L_1 e/ou a agregação global L_2 em (4.1). Dependendo da escolha dos operadores na implementação de L_1 e L_2 , é possível ressaltar três tipos de possíveis realizações para um neurônio lógico baseado em uninormas:

- I. Agregação baseada em normas triangulares e transformação baseada em uninorma em nível local L_1 ;
- II. Agregação baseada em uninorma em nível global L_2 e transformação local baseada em normas triangulares;
- III. Transformação e agregação baseada em uninorma em ambos os níveis L_1 e L_2 ;

Os casos I e II são discutidos em (Pedrycz, 2006), onde duas categorias de processamento (tipo-*and* e tipo-*or*) são definidas para cada caso, de acordo com o mecanismo de agregação usado no nível global no caso I e local no caso II. No caso I, o neurônio tipo-*and* gerado é chamado de neurônio *AND-UNI* enquanto que o tipo-*or* é chamado de neurônio *OR-UNI*. Já para o caso II, os neurônios tipo-*and* e tipo-*or* são chamados de neurônio *UNI-AND* e neurônio *UNI-OR*, respectivamente.

O caso III não foi discutido na literatura, pois, apesar de seu alto nível de flexibilidade, a interpretação deste tipo de neurônio pode ser extremamente complexa, dada a semântica das uninormas. Além disso, segundo (Pedrycz, 2006), o desenvolvimento de algoritmos de treinamento para este tipo de sistema poderia trazer grandes dificuldades. Assim sendo, o objetivo desta seção é desenvolver um neurônio nebuloso baseado na implementação do caso III, de tal forma a preservar a interpretabilidade de um modelo nebuloso baseado neste neurônio além de tornar possível o desenvolvimento de um algoritmo de treinamento.

Para construir este neurônio, que pela nomenclatura dada em (Pedrycz, 2006) poderia ser chamado de neurônio *UNI-UNI*, mas que aqui será chamado simplesmente de Unineuron, é necessário inicialmente propor-se uma pequena modificação na uninorma apresentada em (2.3) da seguinte forma:

$$\tilde{U}(x, y; g) = \begin{cases} g T\left(\frac{x}{g}, \frac{y}{g}\right), & \text{se } x, y \in [0, g] \\ g + (1 - g) S\left(\frac{x-g}{1-g}, \frac{y-g}{1-g}\right), & \text{se } x, y \in [g, 1] \\ \varphi(x, y), & \text{caso contrário.} \end{cases} \quad (4.2)$$

onde a função $\varphi(\cdot)$ é definida como:

$$\varphi(x, y) = \begin{cases} \max(x, y) & \text{se } g \in [0, 0.5) \\ \min(x, y) & \text{se } g \in [0.5, 1] \end{cases}$$

É possível observar em (4.2) que a dominância da uninorma \tilde{U} é controlada pelo valor do elemento identidade g .

O processamento do unineuron é dado pela função $uni(\mathbf{x}, \mathbf{w}; g)$, onde $\mathbf{x} \in [0, 1]^n$ e $\mathbf{w} \in [0, 1]^n$, assim definido:

$$\begin{aligned} uni(\mathbf{a}, \mathbf{w}; g) &= \tilde{U} \left[\tilde{U}(a_1, w_1; (1-g)), \tilde{U}(a_2, w_2; (1-g)), \dots, \tilde{U}(a_n, w_n; (1-g)); g \right] \\ &= \tilde{U}_{i=1}^n \left[\tilde{U}(a_i, w_i; (1-g)); g \right] \end{aligned} \quad (4.3)$$

Uma característica importante a se observar em (4.3) é que a utilização do termo $1-g$ como elemento de identidade da uninorma que implementa as transformações locais L_1 associado à utilização da variável g como elemento de identidade da uninorma que representa L_2 faz com que exista uma espécie de “dualidade” entre estes operadores, exatamente como acontece com os neurônios lógicos *AND* e *OR*.

De fato é possível demonstrar que, se T e S em (4.2) são duais no sentido de De Morgan (Gomide & Pedrycz, 2007), então duas uninormas com elemento de identidade g e $1-g$ também são duais. Para isso, considere seguinte resultado:

Teorema 4.1 (Dualidade de Uninormas:) *Duas uninormas definidas conforme (4.2), onde T e S são normas triangulares duais no sentido de De Morgan, e com elementos de identidade g e $1-g$, respectivamente, também são duais no sentido de De Morgan, isto é:*

$$\tilde{U}(x, y; g) = 1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right]. \quad (4.4)$$

Demonstração:

Considere inicialmente que $x, y \in [0, g]$. Portanto tem-se que $x \leq g$ e $y \leq g$ e a uninorma $\tilde{U}(x, y; g)$ é computada por:

$$\tilde{U}(x, y; g) = g \mathbf{T} \left(\frac{x}{g}, \frac{y}{g} \right). \quad (4.5)$$

Por outro lado, se $x \leq g$ e $y \leq g$, tem-se que $(1-x) \geq (1-g)$ e $(1-y) \geq (1-g)$ e portanto $(1-x), (1-y) \in [(1-g), 1]$. Assim, a uninorma $\tilde{U}((1-x), (1-y); (1-g))$ é calculada da seguinte forma:

$$\tilde{U}((1-x), (1-y); (1-g)) = (1-g) + (1-(1-g)) \mathbf{S} \left(\frac{(1-x) - (1-g)}{1 - (1-g)}, \frac{(1-y) - (1-g)}{1 - (1-g)} \right). \quad (4.6)$$

O cálculo do segundo termo da expressão (4.4) pode ser feito da seguinte forma:

$$\begin{aligned}
1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right] &= \\
&= 1 - \left[(1-g) + (1-1+g)S \left(\frac{1-x-1+g}{1-1+g}, \frac{1-y-1+g}{1-1+g} \right) \right] \\
&= 1 - \left[(1-g) + gS \left(\frac{g-x}{g}, \frac{g-y}{g} \right) \right] \\
&= 1 - 1 + g - gS \left(1 - \frac{x}{g}, 1 - \frac{y}{g} \right) \\
&= g \left[1 - S \left(1 - \frac{x}{g}, 1 - \frac{y}{g} \right) \right]
\end{aligned} \tag{4.7}$$

Uma vez que T e S são duais pode-se aplicar a lei de De Morgan para obter-se:

$$1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right] = gT \left(\frac{x}{g}, \frac{y}{g} \right), \tag{4.8}$$

A partir de (4.5) e (4.8) pode-se concluir que:

$$\tilde{U}(x, y; g) = 1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right]$$

o que implica que se $x, y \in [0, g]$ a expressão (4.4) é válida.

Considere agora que $x, y \in [g, 1]$. Tem-se que $x \geq g$ e $y \geq g$ e a uninorma $\tilde{U}(x, y; g)$ é calculada da seguinte forma:

$$\tilde{U}(x, y; g) = g + (1-g)S \left(\frac{x-g}{1-g}, \frac{y-g}{1-g} \right). \tag{4.9}$$

Neste caso, tem-se que $(1-x) \leq (1-g)$ e $(1-y) \leq (1-g)$ e portanto $(1-x), (1-y) \in [0, (1-g)]$. A uninorma $\tilde{U}((1-x), (1-y); (1-g))$ é então computada como:

$$\tilde{U}((1-x), (1-y); (1-g)) = (1-g)T \left(\frac{(1-x)}{(1-g)}, \frac{(1-y)}{(1-g)} \right). \tag{4.10}$$

Procedendo-se um cálculo semelhante ao realizado para o caso anterior, tem-se que:

$$1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right] =$$

$$\begin{aligned}
&= 1 - \left[(1-g)T \left(\frac{1-x}{1-g}, \frac{1-y}{1-g} \right) \right] \\
&= 1 - \left[(1-g) \left\{ 1 - S \left(1 - \frac{g-x}{1-g}, 1 - \frac{g-y}{1-g} \right) \right\} \right] \\
&= 1 - \left[(1-g) \left\{ 1 - S \left(\frac{x-g}{1-g}, \frac{y-g}{1-g} \right) \right\} \right]
\end{aligned} \tag{4.11}$$

o que implica que:

$$1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right] = g + (1-g)S \left(\frac{x-g}{1-g}, \frac{y-g}{1-g} \right). \tag{4.12}$$

A partir de (4.9) e (4.12) também pode-se concluir que:

$$\tilde{U}(x, y; g) = 1 - \left[\tilde{U}((1-x), (1-y); (1-g)) \right]$$

e portando a expressão (4.4) também é válida quando $x, y \in [g, 1]$.

Para quaisquer outros valores de x e y , a uninorma (4.2) é implementada através ou do operador \min ou do operador \max . Como estes operadores são normas triangulares clássicas e as leis de De Morgan são válidas para eles, a expressão (4.4) também é válida para estes casos, o que conclui a demonstração. \square

Uma vez que a dualidade é válida para uninormas que utilizam elementos identidade g e $1-g$, um unineuron dado por (4.3) que utilize elemento de identidade g é dual a um unineuron com elemento de identidade $1-g$, assim como ocorre com os neurônios *AND* e *OR*.

Além disso, quando $0 \leq g < 0.5$ a transformação local L_1 se torna uma uninorma *and*-dominada, enquanto que a agregação global L_2 se torna uma uninorma *or*-dominada. Esta situação provê um neurônio fortemente relacionado a um neurônio do tipo *or-and* que, sob certos aspectos, se assemelha a um neurônio lógico do tipo *OR*. Em particular, quando $g = 0$ tem-se que o unineuron proposto em (4.3) equivale a um neurônio lógico do tipo *OR*, isto é, $uni(\mathbf{x}, \mathbf{w}; 0) \equiv or(\mathbf{x}; \mathbf{w})$.

Por outro lado, para o caso em que $0.5 \leq g \leq 1$ a expressão (4.3) fornece um neurônio fortemente relacionado a um neurônio do tipo *and-or* que se assemelha a um neurônio lógico do tipo *AND*. Neste caso, quando $g = 1$ tem-se que o unineuron equivale a um neurônio lógico do tipo *AND*, isto é, $uni(\mathbf{x}, \mathbf{w}; 1) \equiv and(\mathbf{x}; \mathbf{w})$.

Estes fatos evidenciam a característica de generalização com relação aos neurônios lógicos *AND* e *OR* dada pelo unineuron aqui proposto.

Um outro aspecto a ser observado é que, em geral, a estrutura de neurônio proposta em (4.3) não requer nenhum tipo de relação (dualidade, por exemplo) entre as realizações específicas das t-normas e s-normas, o que pode trazer uma flexibilidade ainda maior na construção de uma rede composta por

estas unidades.

As Figuras 4.2, 4.3 e 4.4 mostram exemplos da superfície de decisão dada pelo unineuron, Figura 4.1, para um problema bi-dimensional, ou seja, $\mathbf{x}, \mathbf{w} \in [0, 1]^2$.

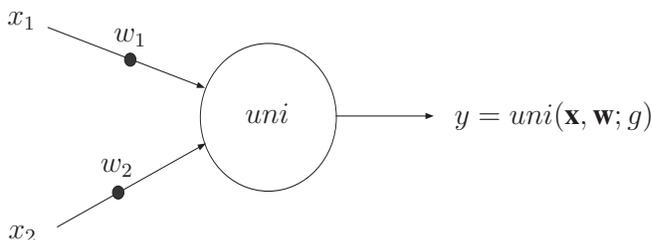


Figura 4.1: Unineuron Aplicado a um Problema Bi-dimensional.

Nestes exemplos, os valores dos pesos foram fixados em $w_1 = 0.3$ e $w_2 = 0.8$, e o valor do elemento identidade g foi sendo alterado de forma a mostrar como o comportamento do neurônio é afetado. Cada uma das figuras apresenta uma realização distinta do unineuron, que depende da escolha das t-normas e s-normas específicas da implementação. Especificamente, na Figura 4.2 foram escolhidos os operadores \min e \max para implementar a t-norma e a s-norma, respectivamente. Já na Figura 4.3, as funções escolhidas foram o produto algébrico como t-norma e a soma probabilística como s-norma, enquanto que na Figura 4.4, a escolha recaiu sobre o produto limitado como t-norma e a soma limitada como s-norma.

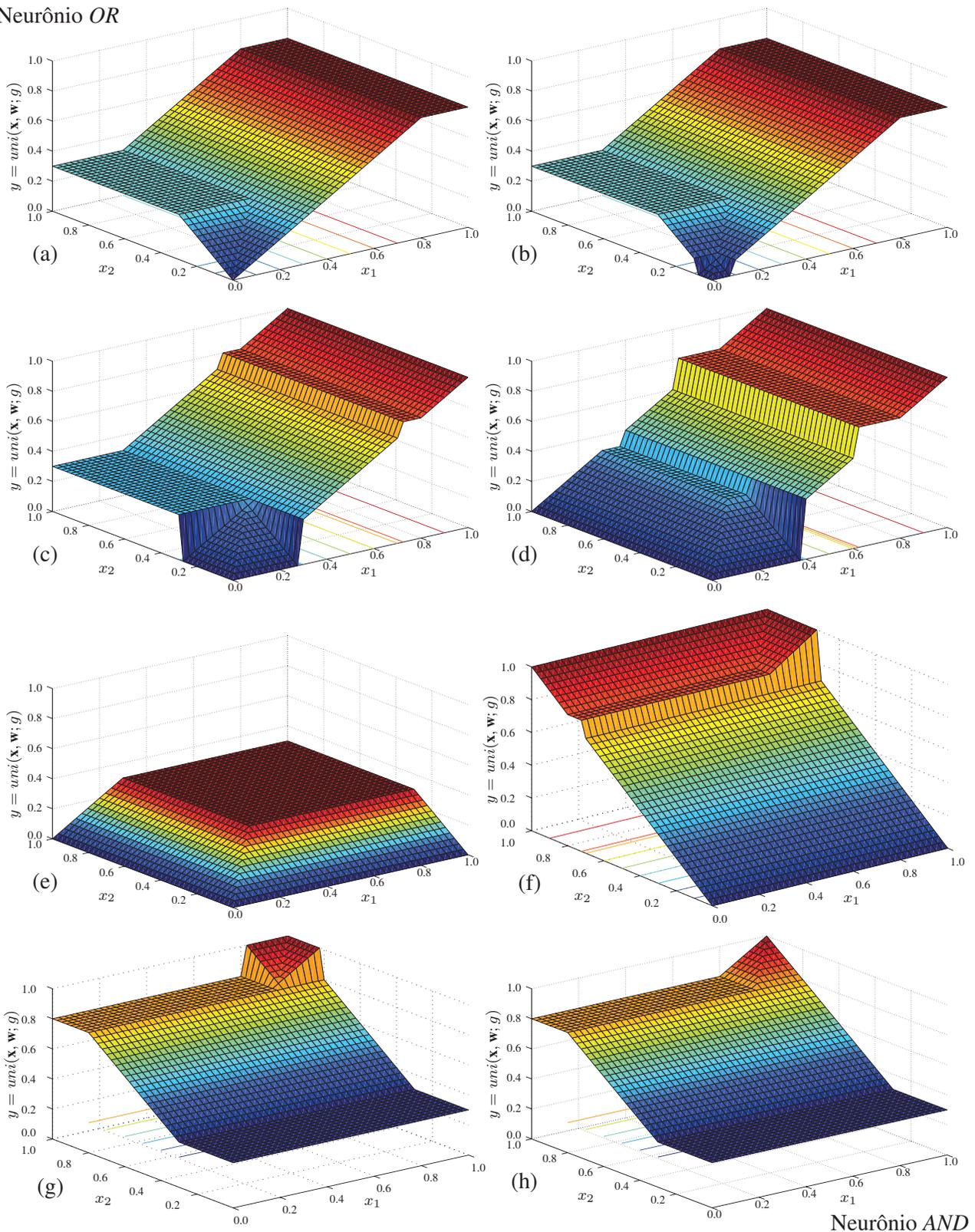
Pode-se observar nas figuras 4.2-4.4 que o comportamento do unineuron é abruptamente alterado quando o valor do elemento identidade g é modificado. Observa-se que, quando $g = 0$, a superfície de decisão do unineuron é exatamente igual a de um neurônio *OR*. A medida que o valor de g vai sendo aumentado, a superfície de decisão do unineuron vai sendo recortada até assumir a mesma forma da superfície de decisão de um neurônio *AND*.

4.3 Neurônio Nebuloso Baseado em Nullnormas: O Nullneuron

Da mesma forma que para as uninormas, as nullnormas também podem ser utilizadas para implementar as transformações locais e os agregadores globais (L_1 e L_2) em (4.1), fornecendo assim uma outra classe de neurônios nebulosos que neste trabalho são chamados de nullneurons.

Mais uma vez, será tratado o caso em que nullnormas são utilizadas na implementação da transformação no nível local e agregação no nível global, mantendo entre eles a mesma relação de dualidade como nos unineurons. Para isso, é possível utilizar a nullnorma apresentada em (2.4) de forma similar

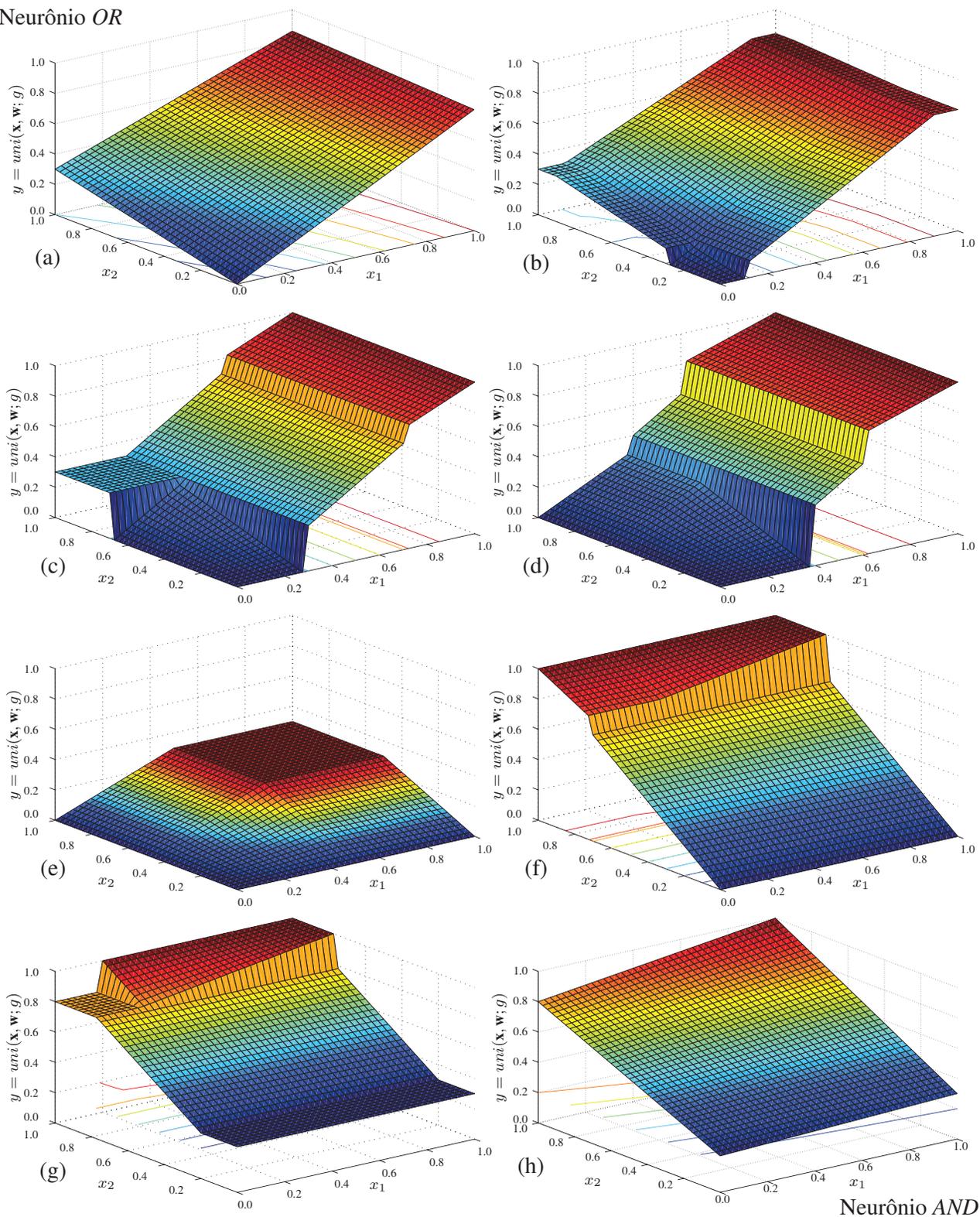
Neurônio OR



Neurônio AND

Figura 4.2: Superfície de Decisão do Unineuron para $\mathbf{w}=[0.3 \ 0.8]$, t-norma = min, s-norma = max e:
 (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$

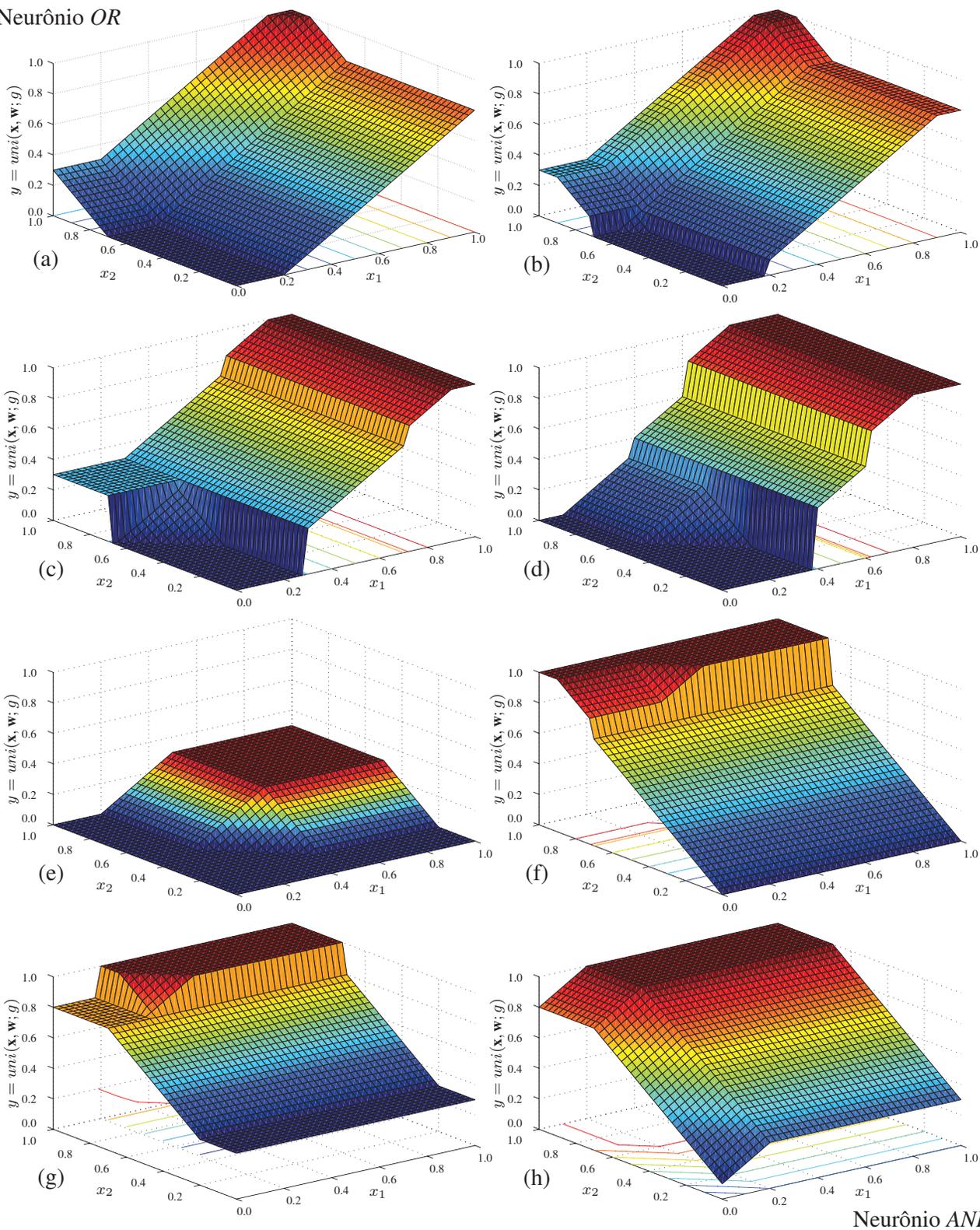
Neurônio OR



Neurônio AND

Figura 4.3: Superfície de Decisão do Unineuron para $w=[0.3 \ 0.8]$, t-norma = produto algébrico, s-norma = soma probabilística e: (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$

Neurônio OR



Neurônio AND

Figura 4.4: Superfície de Decisão do Unineuron para $w=[0.3 \ 0.8]$, t-norma = produto limitado, s-norma = soma limitada e: (a) $g = 0$, (b) $g = 0.1$, (c) $g = 0.3$, (d) $g = 0.4$, (e) $g = 0.5$, (f) $g = 0.7$, (g) $g = 0.8$, (h) $g = 1$

à da seção anterior. Entretanto, com a finalidade de se obter um neurônio em que a variação entre os dois tipos de neurônios lógicos ocorresse de forma mais suave, resolveu-se propor uma forma alternativa de construção. De fato, esta seção mostra como é possível construir um neurônio nebuloso que varia suavemente (ao invés de abruptamente como ocorre com os unineurons) de um neurônio lógico do tipo *AND* para um neurônio lógico do tipo *OR* a medida que o parâmetro u (elemento de absorção) varia de 0 para 1, utilizando-se para isso uma nova nullnorma, baseada na definição geral de nullnorma (Definição 2.5) e no operador de compromisso (Definição 2.6). Esta nova nullnorma é dada por:

$$\begin{aligned}\tilde{N}(x, y; u) &= \tilde{C}_u \left(S \left\{ \tilde{C}_u(x), \tilde{C}_u(y) \right\} \right) \\ &= \tilde{C}_{1-u} \left(T \left\{ \tilde{C}_{1-u}(x), \tilde{C}_{1-u}(y) \right\} \right)\end{aligned}\tag{4.13}$$

A expressão (4.13) fornece uma realização contínua para uma família de nullnormas. Entretanto, nesta construção, há uma necessidade de que T e S sejam duas normas triangulares duais com relação ao complemento nebuloso clássico $C(x) = 1 - x$.

É fácil verificar que \tilde{N} é uma nullnorma, pois tem-se que:

$$\tilde{N}(x, y; u) = \begin{cases} T\{x, y\} & \text{para } u = 0 \\ \frac{1}{2} & \text{para } u = \frac{1}{2} \\ S\{x, y\} & \text{para } u = 1 \end{cases}\tag{4.14}$$

Além disso, tem-se também que para $0 < u < 0.5$, a função \tilde{N} se assemelha a uma t-norma, enquanto, para $0.5 < u < 1$, esta função se assemelha a uma s-norma. Estas semelhanças são mais visíveis quando o parâmetro u está mais próximo de 0 ou de 1, respectivamente.

Assim, utilizando-se a expressão (4.13) para implementar os níveis de transformação e agregação do neurônio, de forma similar ao apresentado para o unineuron, definem-se os nullneurons pela função $null(\mathbf{x}, \mathbf{w}; u)$ da seguinte forma:

$$\begin{aligned}null(\mathbf{a}, \mathbf{w}; u) &= \tilde{N} \left[\tilde{N}(a_1, w_1; (1 - u)), \tilde{N}(a_2, w_2; (1 - u)), \dots, \tilde{N}(a_n, w_n; (1 - u)); u \right] \\ &= \tilde{N}_{i=1}^n \left[\tilde{N}(a_i, w_i; (1 - u)); u \right]\end{aligned}\tag{4.15}$$

A principal característica do nullneuron em (4.15) é a habilidade de mudar suavemente de um neurônio lógico do tipo *AND* para um neurônio lógico do tipo *OR*. De fato, a característica contínua e a ausência de dominância na função \tilde{N} fazem com que as mudanças impostas pela variação do

elemento de absorção u ocorra de forma suave. Além disso, é fácil verificar que, quando $u = 0$, o nullneuron equivale a um neurônio lógico *AND*, ou seja, $null(\mathbf{x}, \mathbf{w}; 0) \equiv and(\mathbf{x}; \mathbf{w})$, enquanto, para $u = 1$, o nullneuron equivale a um neurônio lógico *OR*, isto é, $null(\mathbf{x}, \mathbf{w}; 1) \equiv or(\mathbf{x}; \mathbf{w})$.

Com relação à dualidade, é possível verificar diretamente em (4.13) que duas nullnormas que utilizem elementos de absorção u e $1 - u$ são duais no sentido de De Morgan, e portanto, de forma similar ao que ocorre para os unineurons, dois nullneurons com elementos de absorção u e $1 - u$ são duais, da mesma forma que para os neurônios *AND* e *OR*.

Para ilustrar estas propriedades, considere o nullneuron com $\mathbf{x}, \mathbf{w} \in [0, 1]^2$ conforme Figura 4.5, onde fixou-se os valores dos pesos em $w_1 = 0.3$ e $w_2 = 0.8$ e variou-se o valor do elemento de absorção u para 3 diferentes realizações do nullneuron.

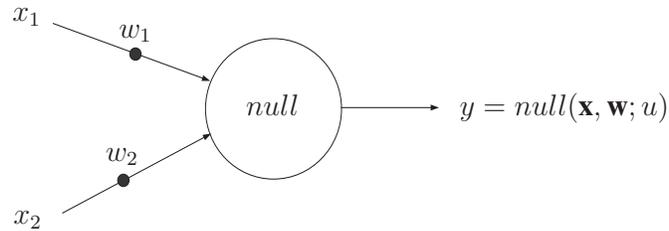


Figura 4.5: Nullneuron Aplicado à Problema Bi-dimensional.

As Figuras 4.6, 4.7 e 4.8 mostram as superfícies de decisão do nullneuron. Em particular, a Figura 4.6 mostra as superfícies de decisão do nullneuron para valores distintos de u e quando os operadores *min* e *max* são escolhidos para implementar a t-norma e a s-norma em (4.15), respectivamente. Na Figura 4.7 a função escolhida como t-norma foi o produto algébrico e como s-norma a soma probabilística. Por fim, na Figura 4.8, a escolha recaiu sobre o produto limitado como t-norma e a soma limitada como s-norma.

Note que, apesar de, nos pontos extremos $u = 0$ ($g = 1$) e $u = 1$ ($g = 0$), as superfícies de decisão serem idênticas para ambos os neurônios, entre estes dois pontos a variação da superfície dada pelo nullneuron se dá de uma forma mais suave quando comparada com àquela dada pelo unineuron.

Conforme pôde ser verificado nas duas últimas seções, unineurons e nullneurons são unidades de processamento altamente não-lineares que, da mesma forma que os neurônios lógicos *AND* e *OR*, dependem das realizações específicas dos conectivos utilizados em sua construção, ou seja, das t-normas e s-normas escolhidas. Estes novos neurônios apresentam também um alto potencial de plasticidade que pode se tornar extremamente útil no treinamento de redes neurais nebulosas que envolvem este tipo de neurônio em sua estrutura.

Neurônio AND

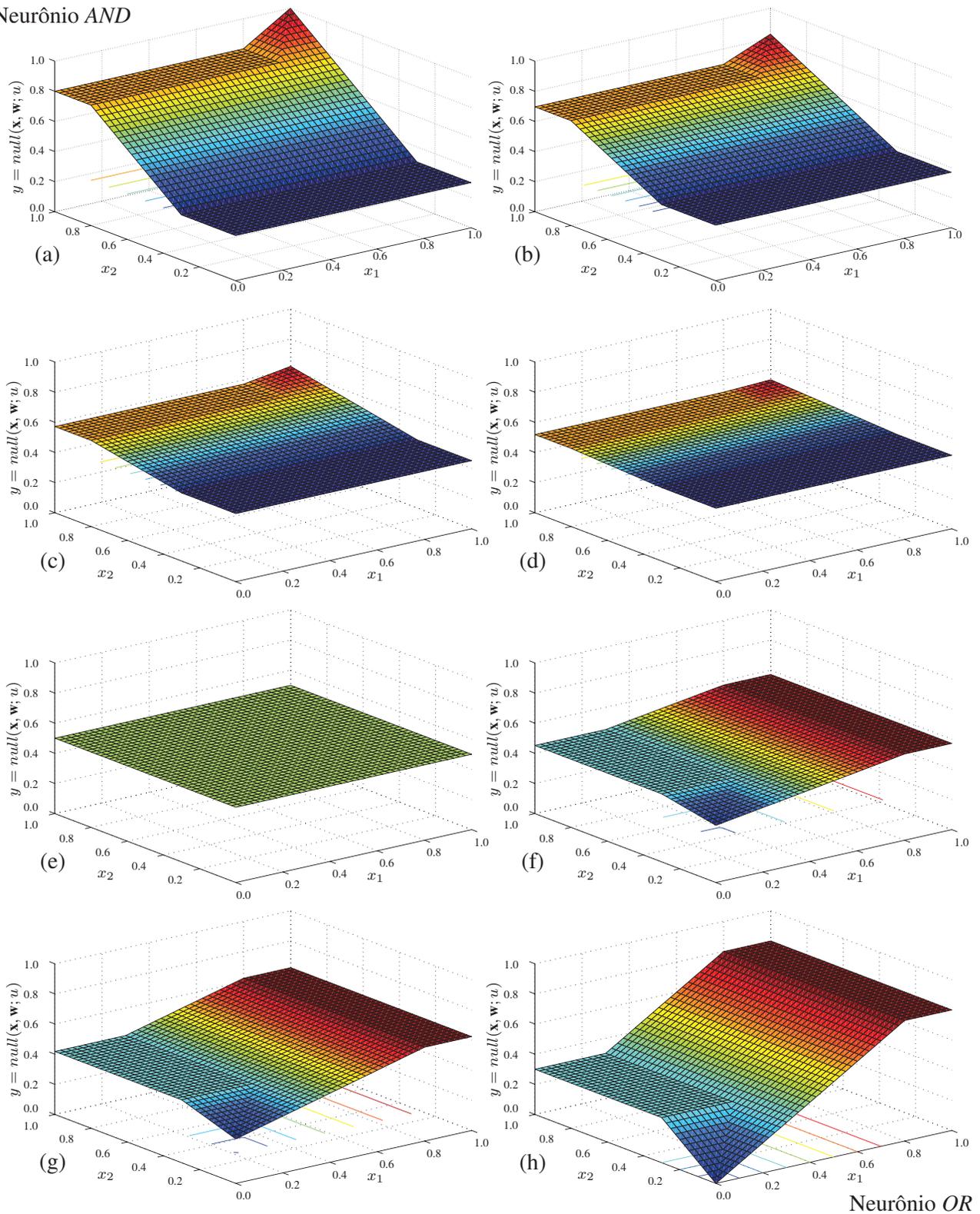


Figura 4.6: Superfície de Decisão do nullneuron para $w=[0.3 \ 0.8]$, t-norma = min, s-norma = max e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$

Neurônio AND

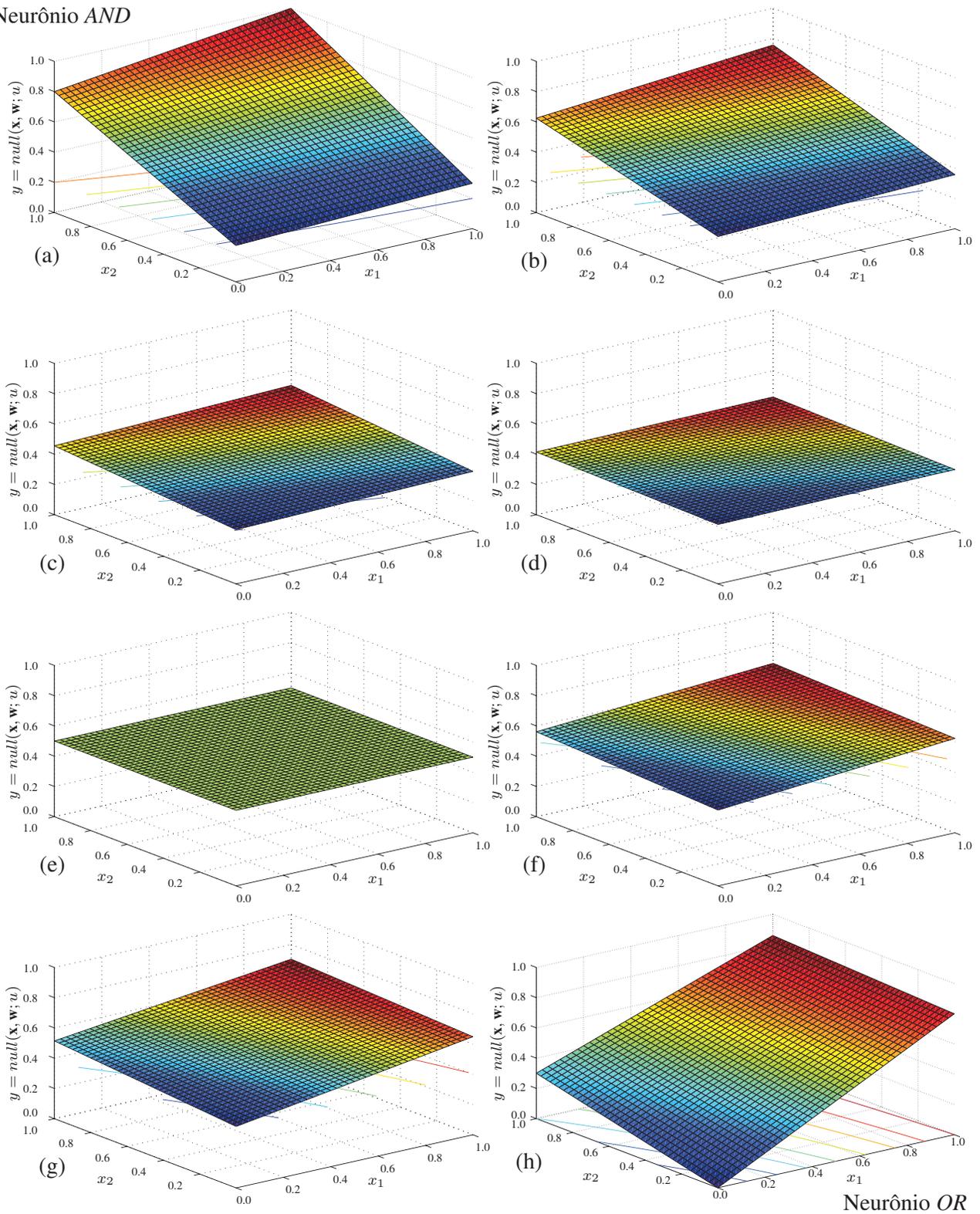


Figura 4.7: Superfície de Decisão do nullneurônio para $\mathbf{w}=[0.3 \ 0.8]$, t-norma = produto algébrico, s-norma = soma probabilística e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$

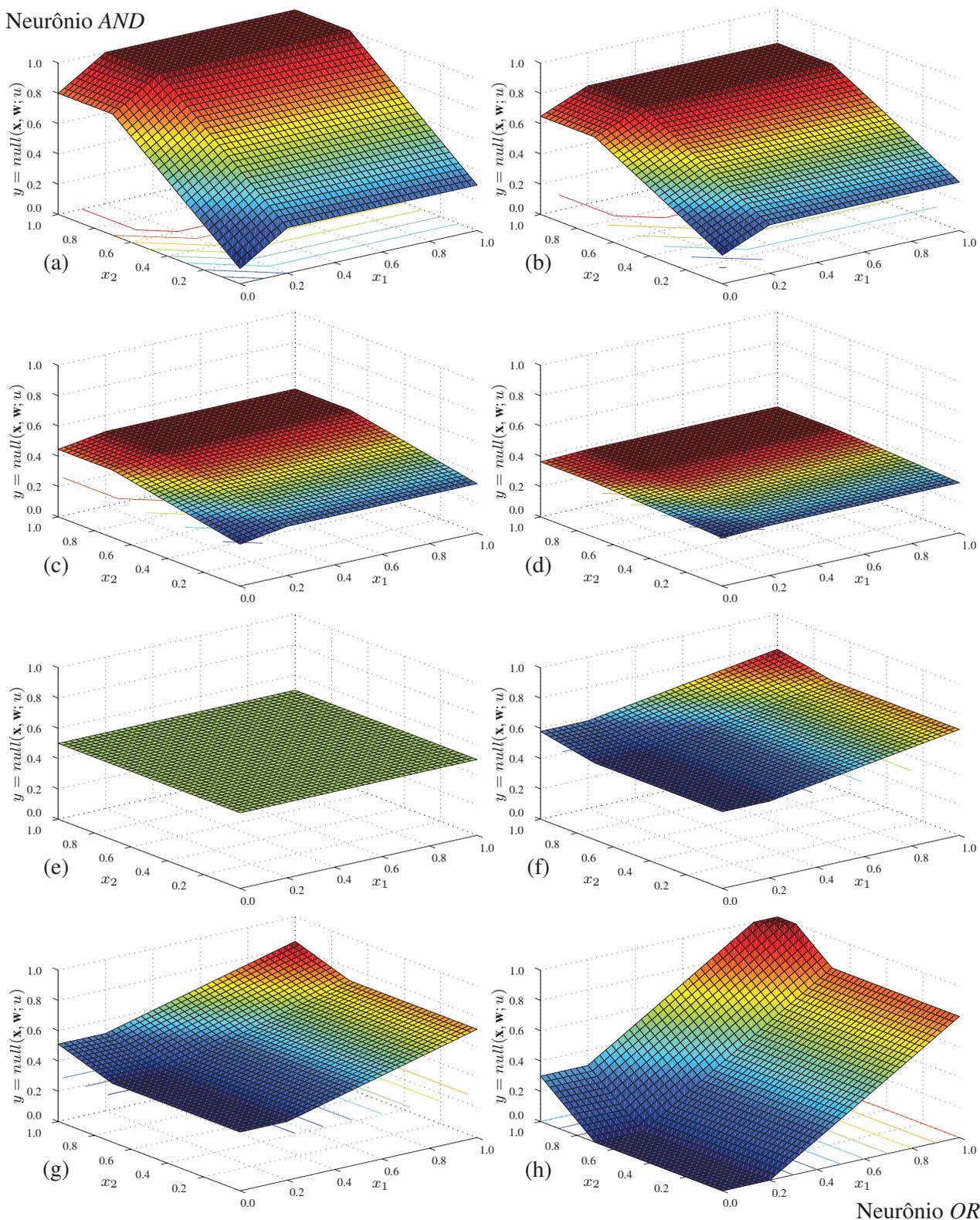
Neurônio *AND*

Figura 4.8: Superfície de Decisão do nullneuron para $\mathbf{w}=[0.3 \ 0.8]$, t-norma = produto limitado, s-norma = soma limitada e: (a) $u = 0$, (b) $u = 0.1$, (c) $u = 0.3$, (d) $u = 0.4$, (e) $u = 0.5$, (f) $u = 0.7$, (g) $u = 0.8$, (h) $u = 1$

4.4 Redes Neurais Nebulosas baseadas em Nullneurons e Unineurons

Os dois neurônios propostos nas seções anteriores podem ser utilizados nos mais diversos tipos de estruturas neurais nebulosas existentes. Em particular, todas as estruturas que utilizam neurônios lógicos em sua composição também podem ser construídas utilizando-se unineurons e nullneurons, em substituição aos neurônios *AND* e *OR*. Exemplos destas estruturas podem ser encontrados em (Pedrycz, 1993; Lin & Lee, 1996; Figueiredo et al., 2004) entre outros.

Neste trabalho, a estrutura escolhida para verificar a aplicação destes novos neurônios lógicos foi a classe de redes híbridas apresentadas na seção 2.4.2 do Capítulo 2. Desta forma, tomou-se a estrutura mostrada na Figura 3.5, reproduzida na Figura 4.9, e substituiu-se os neurônios lógicos na segunda camada de neurônios nebulosos ora por unineurons ora por nullneurons, obtendo-se assim dois novos tipos de redes neurais nebulosas a serem aplicadas à modelagem de sistemas dinâmicos não lineares.

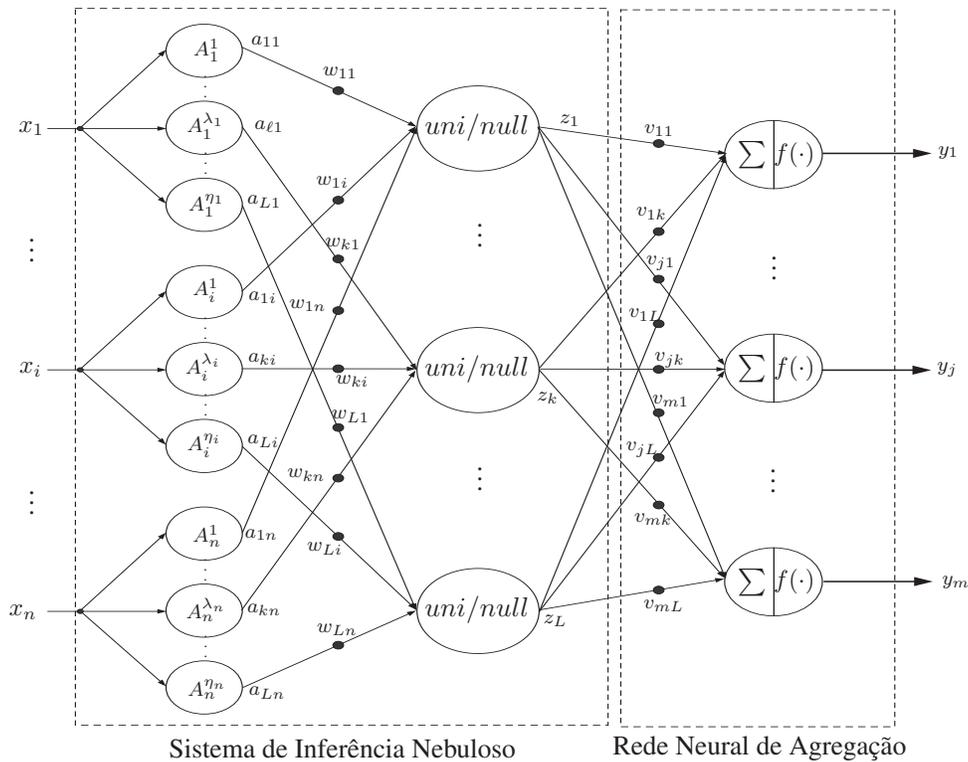


Figura 4.9: Modelo de rede neural nebulosa recorrente com unineurons e nullneurons

O processamento para cada uma destas novas redes pode ser resumido conforme apresentado a seguir.

Processamento da Rede Neural Nebulosa baseada em Unineurons

Passo 1: Calcula-se a matriz $A = [a_{\ell i}]$, $\ell = 1, \dots, L$, $i = 1, \dots, n$, onde $a_{\ell i} = A_i^{\lambda_i}(x_i)$ é a saída do λ_i -ésimo neurônio nebuloso do tipo I da entrada x_i que vai na direção do ℓ -ésimo neurônio lógico na segunda camada;

Passo 2: Calcule-se o vetor $\mathbf{z} = [z_\ell]$, $\ell = 1, \dots, L$ da seguinte forma:

$$z_\ell = \text{uni}(\mathbf{a}_\ell, \mathbf{w}_\ell; g_\ell) = \tilde{\mathbf{U}}_{i=1}^n \left[\tilde{\mathbf{U}}(a_{\ell i}, w_{\ell i}; (1 - g_\ell)); g_\ell \right] \quad (4.16)$$

onde \mathbf{a}_ℓ é um vetor igual à linha ℓ da matriz A e \mathbf{w}_ℓ é um vetor igual à linha ℓ da matriz W .

Passo 3: Calcula-se o vetor de saída $\mathbf{y} = [y_j]$, $j = 1, \dots, m$ através da expressão:

$$y_j = f \left(\sum_{\ell=1}^L z_\ell v_{j\ell} \right), \quad (4.17)$$

onde $f(\cdot)$ é a função de ativação dos neurônios da camada de saída. Aqui também utilizou-se a função sigmoïdal, i.e., $f(\xi) = (1 + \exp(-\xi))^{-1}$.

Processamento da Rede Neural Nebulosa baseada em nullneurons

Passo 1: Calcula-se a matriz $A = [a_{\ell i}]$, $\ell = 1, \dots, L$, $i = 1, \dots, n$, onde $a_{\ell i} = A_i^{\lambda_i}(x_i)$ é a saída do λ_i -ésimo neurônio nebuloso do tipo I da entrada x_i que vai na direção do ℓ -ésimo neurônio lógico na segunda camada;

Passo 2: Calcule-se o vetor $\mathbf{z} = [z_\ell]$, $\ell = 1, \dots, L$ da seguinte forma:

$$z_\ell = \text{null}(\mathbf{a}_\ell, \mathbf{w}_\ell; g_\ell) = \tilde{\mathbf{N}}_{i=1}^n \left[\tilde{\mathbf{N}}(a_{\ell i}, w_{\ell i}; (1 - g_\ell)); g_\ell \right] \quad (4.18)$$

onde \mathbf{a}_ℓ é um vetor igual à linha ℓ da matriz A e \mathbf{w}_ℓ é um vetor igual à linha ℓ da matriz W .

Passo 3: Calcula-se o vetor de saída $\mathbf{y} = [y_j]$, $j = 1, \dots, m$ através da expressão:

$$y_j = f \left(\sum_{\ell=1}^L z_\ell v_{j\ell} \right). \quad (4.19)$$

Vale notar que cada um dos unineurons e nullneurons da rede proposta possui seu próprio parâmetro de plasticidade (elemento de identidade g para unineurons e elemento de absorção u para nullneurons). Este fato, aliado à um processo de treinamento que ajuste tais elementos de forma individual, faz com que cada um dos neurônios que compõe a rede possa processar entradas e pesos de conexão de sua própria maneira, independente do estado dos demais neurônios.

Em particular, se restrições forem impostas de tal forma que, após o término do processo de treinamento, o valor final do parâmetro de plasticidade pertença ao conjunto $\{0, 1\}$, tem-se que cada um dos neurônios na segunda camada de neurônios nebulosos da rede terá se transformado ou em um neurônio lógico do tipo *AND* ou em um neurônio lógico do tipo *OR*.

Tanto o caso em que os parâmetros de plasticidade pertencem ao conjunto $\{0, 1\}$ quanto para os casos em que estes pertençam ao intervalo $[0, 1]$, a primeira parte da rede híbrida sob estudo possa representar um sistema de inferência nebuloso que utiliza mais de um conectivo no antecedente das regras nebulosas, diferentemente de todos os modelos existentes na literatura, onde apenas um conectivo pode ser utilizado em cada sistema.

Desta forma, cada um dos neurônios na segunda camada de neurônios nebulosos da rede codificaria uma regra nebulosa da forma:

$$R_\ell : SE (x_1 \text{ é } A_1^{\lambda_1} \text{ OU/E } w_{\ell 1}) \text{ E/OU } (x_2 \text{ é } A_2^{\lambda_2} \text{ OU/E } w_{\ell 2}) \dots \dots \text{ E/OU } (x_n \text{ é } A_n^{\lambda_n} \text{ OU/E } w_{\ell n}) \text{ ENTÃO, } z \text{ is } z_\ell, \quad (4.20)$$

sendo que a forma exata de cada regra é definida pelo processo de treinamento, conforme será apresentado na seção a seguir.

4.5 Aprendizado para Redes Baseadas em Unineurons e Null-neurons

Após a definição da estrutura, o próximo passo é se definir a forma com que será feito o ajuste dos parâmetros dos neurônios que formam a rede. Especial atenção deve ser dada aos novos neurônios nebulosos propostos neste capítulo, pois cada um deles é equipado com um significativo grau de flexibilidade paramétrica, que reside tanto em seus pesos de conexão $w_{\ell i}$ quanto nos parâmetro de plasticidade g ou u .

Diversos paradigmas de aprendizado podem ser usados no processo de ajuste dos parâmetros da rede. Entre eles, podemos citar, por exemplo, métodos baseados no gradiente, otimização por exames de partículas, algoritmos genéticos e regras reforço associativo. O procedimento proposto neste trabalho é um método híbrido que se baseia no gradiente descendente para fazer o ajuste dos pesos na camada de saída da rede e em uma regra de reforço associativo para ajustar os pesos associados ao

sistema de inferência nebuloso na primeira parte da rede, bem como os parâmetros de plasticidade dos unineurons e nullneurons. Este procedimento pode ser visto como uma extensão daquele apresentado em (Ballini & Gomide, 2002), onde uma nova equação baseada na regra de reforço é adicionada ao procedimento para a correção individual dos parâmetros g e u .

Neste trabalho, a fim de se manter a interpretabilidade do modelo final, que é uma das principais vantagens da estrutura neural nebulosa proposta, implementou-se ainda uma função que faz com que os parâmetros de plasticidade g e u converjam ou para 0 ou para 1, dependendo da necessidade do problema. Esta convergência se faz de forma individual nos parâmetros, ou seja, no final da fase de treinamento, cada um dos neurônios nebulosos na segunda camada da rede terá o valor de seu parâmetro de plasticidade ajustado para 0 ou 1, independentemente do valor dos demais neurônios. Esta função é dada por:

$$f_z(g_\ell) = \frac{1}{1 + \exp(-(p_1 g_\ell - p_2))} \text{ para unineurons,} \quad (4.21)$$

e

$$f_z(u_\ell) = \frac{1}{1 + \exp(-(p_1 u_\ell - p_2))} \text{ para nullneurons,} \quad (4.22)$$

onde p_1 e p_2 são parâmetros da função. Aqui assumiu-se que $p_1 = 10$ e $p_2 = 5$.

O algoritmo de treinamento é detalhado a seguir. Para tanto, considere uma seqüência de pares de treinamento da por $[\mathbf{x}^k, \mathbf{y}^k]$, $k = 1, \dots, P$, onde P é o número total de pares.

Aprendizado da Rede Neural Nebulosa baseada em Unineurons

O algoritmo para o aprendizado de redes onde unineurons são utilizados para implementar a segunda camada de neurônios nebulosos é descrito a seguir:

1. Granularizar o espaço de entrada, gerando as funções de pertinência conforme apresentado na seção 3.4.1 do Capítulo 3;
2. Inicializar os pesos $w_{\ell i}$, $\ell = 1, \dots, L$, $i = 1, \dots, n$ aleatoriamente, no intervalo $[0, 1]$;
3. Inicializar os elementos identidade g_ℓ , $\ell = 1, \dots, L$ aleatoriamente, no intervalo $[0, 1]$;
4. Inicializar os pesos $v_{j\ell}$, $j = 1, \dots, m$, $\ell = 1, \dots, L$ aleatoriamente, no intervalo $[-0.1, 0.1]$;
5. Para cada um dos pares de treinamento $[\mathbf{x}^k, \mathbf{y}^k]$ faça:
 - 5.1. Calcular a matriz A^k propagando-se o vetor de entrada \mathbf{x}^k através da primeira camada de neurônios nebulosos;

5.2. Calcular o vetor $\hat{\mathbf{z}}^k = [\hat{z}_\ell^k]$, $\ell = 1, \dots, L$, contendo as saídas obtidas para os unineurons através da expressão:

$$z_\ell = \text{uni}(\mathbf{a}_\ell, \mathbf{w}_\ell; g_\ell) = \tilde{\text{U}}_{i=1}^n \left[\tilde{\text{U}}(a_{\ell i}, w_{\ell i}; (1 - g_\ell)); g_\ell \right]; \quad (4.23)$$

5.3. Calcular o vetor de saída obtida da rede $\hat{\mathbf{y}}^k = [y_j^k]$, $j = 1, \dots, m$, utilizando a expressão:

$$\hat{y}_j^k = f \left(\sum_{\ell=1}^L \hat{z}_\ell^k v_{j\ell}^k \right); \quad (4.24)$$

5.4. Determinar o vetor com o erro de aproximação $\mathbf{e}^k = [e_j^k]$, $j = 1, \dots, m$, de acordo com:

$$e_j^k = y_j^k - \hat{y}_j^k, \quad (4.25)$$

onde y_j^k é a j -ésima componente do vetor de saídas desejadas \mathbf{y}^k ;

5.5. Determinar o vetor gradiente de erro da seguinte forma:

$$\delta_j^k = e_j^k f'(\psi_j^k), \quad (4.26)$$

onde $f'(\psi_j^k)$ é a derivada de $f(\cdot)$ em $\psi_j^k = \sum_{j=1}^m (v_{j\ell}^k z_\ell^k)$;

5.6. Calcular a variação dos pesos relacionados à camada de saída $v_{j\ell}^k$ de acordo com:

$$\Delta v_{j\ell}^k = \alpha_s \delta_j^k z_\ell^k, \quad (4.27)$$

onde α_s é a taxa de treinamento para a camada de saída;

5.7. Atualizar os pesos da camada de saída de acordo com a regra:

$$v_{j\ell}^{k+1} = v_{j\ell}^k + \Delta v_{j\ell}^k; \quad (4.28)$$

5.8. Calcular a variação dos pesos relacionados à segunda camada da rede $w_{\ell i}^k$ de acordo com:

$$\Delta w_{\ell i} = \delta \alpha_1 [1 - w_{\ell i}] - (1 - \delta) \alpha_2 w_{\ell i}, \quad (4.29)$$

onde $0 < \alpha_1 \ll \alpha_2 < 1$ são taxas de aprendizado para os pesos desta camada e δ é o

sinal de reforço dado por $\delta = 1 - \epsilon$, ϵ é o erro que se deseja minimizar, calculado como:

$$\epsilon = \frac{1}{2} \sum_{j=0}^m (y_j^k - \hat{y}_j^k)^2 \quad (4.30)$$

5.9. Atualizar os pesos relacionados ao sistema de inferência nebuloso de acordo com a regra:

$$w_{\ell i}^{k+1} = w_{\ell i}^k + \Delta w_{\ell i}^k; \quad (4.31)$$

5.10. Calcular a variação dos elementos identidade g_{ℓ}^k de acordo com:

$$\Delta g_{\ell} = \delta \alpha_3 [1 - g_{\ell}] - (1 - \delta) \alpha_4 g_{\ell}, \quad (4.32)$$

onde $0 < \alpha_3 \ll \alpha_4 < 1$ são taxas de aprendizado para os elementos identidade;

5.11. Atualizar os valores dos elementos identidade de acordo com a regra:

$$g_{\ell}^{k+1} = g_{\ell}^k + \Delta g_{\ell}^k; \quad (4.33)$$

5.12. Aplica-se a função dada em (4.21) para obter a interpretabilidade;

6. Repetir os passos 5.1 - 5.12 até que uma tolerância de erro ou um número máximo de iterações seja atingido.

Treinamento da Rede Neural Nebulosa baseada em nullneurons

O algoritmo para o treinamento de redes onde nullneurons são utilizados para implementar a segunda camada de neurônios nebulosos é muito similar àquele apresentado para redes com unineurons, sendo que para obter-se este novo algoritmo é necessário apenas substituir a equação (4.23) pela seguinte expressão:

$$z_{\ell} = null(\mathbf{a}_{\ell}, \mathbf{w}_{\ell}; u_{\ell}) = \tilde{\mathbf{N}}_{i=1}^n \left[\tilde{\mathbf{N}}(a_{\ell i}, w_{\ell i}; (1 - u_{\ell})); u_{\ell} \right]; \quad (4.34)$$

Além dos dois algoritmos apresentados, é possível ainda propor um outro processo de treinamento para redes neurais nebulosas compostas por nullneurons, onde aproveita-se da característica de continuidade do nullneuron, conforme discutido anteriormente, para se propor um novo procedimento baseado inteiramente no método do gradiente descendente.

Neste novo procedimento, a variação dos pesos relacionados à segunda camada da rede passa então a ser calculada com base no gradiente, da seguinte forma:

$$\Delta w_{\ell i} = \frac{\partial}{\partial w_{\ell i}} [null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)] \sum_{j=1}^m \{v_{j\ell}(y_j - \hat{y}_j) f'(\rho_j)\} a_{\ell i} \quad (4.35)$$

onde $\partial[null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)]/\partial w_{\ell i}$ é a derivada parcial da função $null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)$ com relação aos pesos $w_{\ell i}$. O cálculo desta derivada depende da escolha da t-norma e da s-norma que serão usadas na implementação dos nullneurons. Para exemplificar o processo de cálculo, adotou-se aqui o produto algébrico como t-norma e a soma probabilística como s-norma. Para estas normas o cálculo da derivada mencionada acima é dado por:

$$\frac{\partial}{\partial w_{\ell i}} [null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)] = (1 - 2u_\ell)^4 \left\{ \sum_{c=1}^n \left[\tilde{N}_{u_\ell}(a_{\ell c}) \times \prod_{\substack{\varsigma=1 \\ \varsigma \neq c}}^n (1 - \tilde{N}_{u_\ell}(\tau_{\ell c})) \right] \right\} \quad (4.36)$$

onde $\tau_{\ell c} = \tilde{C}_{u_\ell}(\tilde{C}_{u_\ell}(a_{\ell c}) \times \tilde{C}_{u_\ell}(w_{\ell c}))$.

Além da variação dos pesos $w_{\ell i}$, a variação dos elementos de absorção u_ℓ também é calculada com base no método do gradiente. Assim, de forma similar ao apresentado para os pesos, a variação dos elementos de absorção é calculada da seguinte forma:

$$\Delta u_\ell = \frac{\partial}{\partial u_\ell} [null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)] \sum_{j=1}^m \{v_{j\ell}(y_j - \hat{y}_j) f'(\rho_j)\} \quad (4.37)$$

Uma vez que se utilizou a equação 4.22 para se manter a interpretabilidade do modelo, o operador de compromisso dado em (2.5) toma a seguinte forma:

$$\tilde{C}_{f_z(u_\ell)}(a) = (1 - f_z(u_\ell))C(a) + f_z(u_\ell)a \quad (4.38)$$

e a equação (4.36) pode ser re-escrita como

$$\frac{\partial}{\partial w_{\ell i}} [null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)] = (1 - 2f_z(u_\ell))^4 \left\{ \sum_{c=1}^n \left[\tilde{C}_{f_z(u_\ell)}(a_{\ell c}) \times \prod_{\substack{\varsigma=1 \\ \varsigma \neq c}}^n (1 - \tilde{C}_{f_z(u_\ell)}(\tau_{\ell \varsigma})) \right] \right\}. \quad (4.39)$$

Desta forma, a derivada $\partial[null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)]/\partial u_k$ pode ser calculada da seguinte forma:

$$\begin{aligned} \frac{\partial}{\partial u_\ell} [null(\mathbf{a}_\ell, \mathbf{w}_\ell; u_\ell)] = (1 - 2f_z(u_\ell)) \frac{\partial}{\partial u_\ell} f_z(u_\ell) \left\{ \sum_{c=1}^n \left[\frac{\partial}{\partial u_\ell} h(a_{\ell c}, w_{\ell c}, f_z(u_\ell)) \times \right. \right. \\ \left. \left. \times \prod_{\substack{\varsigma=1 \\ \varsigma \neq c}}^n (h(a_{\ell \varsigma}, w_{\ell \varsigma}, f_z(u_\ell))) \right] \right\} + \frac{\partial}{\partial u_\ell} f_z(u_\ell) \left[1 - 2 \prod_{c=1}^n h(a_{\ell c}, w_{\ell c}, f_z(u_\ell)) \right] \end{aligned} \quad (4.40)$$

onde

$$h(a_{\ell c}, w_{\ell c}, f_z(u_\ell)) = 1 - \tilde{C}_{f_z(u_\ell)}(q(a_{\ell c}, w_{\ell c}, f_z(u_\ell))) \quad (4.41)$$

$$q(a_{\ell c}, w_{\ell c}, f_z(u_\ell)) = \tilde{C}_{f_z(u_\ell)}(\theta(a_{\ell c}, w_{\ell c}, f_z(u_\ell))) \quad (4.42)$$

$$\theta(a_{\ell c}, w_{\ell c}, f_z(u_\ell)) = \tilde{C}_{f_z(u_\ell)}(a_{\ell c}) \times \tilde{C}_{f_z(u_\ell)}(w_{\ell c}) \quad (4.43)$$

e as derivadas das equações (4.41)-(4.43) podem ser facilmente obtidas.

Note mais uma vez que, em todos os procedimentos apresentados, o ajuste dos parâmetros de plasticidade é feito individualmente, permitindo assim que o modelo obtido no final do processo de treinamento seja mais geral do que se a rede neural nebulosa que o implementa fosse composta apenas por neurônios do tipo *AND* ou *OR*.

Para verificar esta afirmação, as redes e procedimentos propostos neste capítulo foram empregadas na modelagem de sistemas dinâmicos não-lineares, sendo que os resultados obtidos foram comparados com os diversos tipos de modelos baseados em inteligência computacional propostos na literatura. Estes resultados são apresentados no próximo capítulo.

4.6 Resumo

Neste capítulo, introduziram-se duas novas classes de neurônios nebulosos baseadas nos conceitos de uninormas e nullnormas. Estes neurônios, chamados respectivamente de unineurons e nullneurons, apresentam uma importante característica dos neurônios biológicos, chamada de plasticidade neuronal, através da qual um neurônio é capaz de alterar seus mecanismos internos de processamento dependendo da situação ou da necessidade de aprendizagem em que este se encontre.

Esta característica, associada a um algoritmo de treinamento que ajusta individualmente a forma com que cada neurônio processa os sinais de entrada e pesos de conexão, faz com que uma rede neural nebulosa composta por estes novos tipos de neurônios se adapte melhor a um conjunto de dados de

treinamento, se tornando, assim, mais geral do que as alternativas propostas na literatura.

Além disso, se algumas restrições forem impostas durante o treinamento o modelo final obtido, além de apresentar um alto grau de interpretabilidade, pode representar um sistema de inferência nebuloso que emprega mais de um conectivo lógico no antecedente das regras nebulosas, sendo que o tipo de conectivo é decidido pelo processo de treinamento.

Capítulo 5

Resultados Experimentais e Aplicações

5.1 Introdução

Este capítulo apresenta resultados de simulações e aplicações a problemas de modelagem de sistemas dinâmicos não-lineares. Com a finalidade de mostrar a abrangência de aplicação dos algoritmos propostos neste trabalho, foram abordadas duas classes de problemas, predição de séries temporais e identificação de sistemas.

Para a primeira destas classes, considerou-se previsão de carga de curto prazo, onde o objetivo é prever, com base em dados históricos, o consumo de energia elétrica para a próxima hora de operação em uma determinada região durante um período de 24 horas.

A segunda classe considera dois problemas. O primeiro deles é o problema clássico do forno a gás de Box & Jenkins (Box & Jenkins, 1994), onde, a partir de dados relativos ao fluxo de metano utilizado no forno, deseja-se estimar a concentração de CO_2 na saída do forno. O segundo problema consiste na modelagem do comportamento térmico de transformadores de potência, sendo que, a partir de dados relativos ao carregamento elétrico (em termos da corrente de carga) e da temperatura externa, pretende-se inferir as temperaturas internas do equipamento.

Os algoritmos desenvolvidos neste trabalho foram empregados para construir os modelos de previsão e do comportamento térmico de transformadores. Em particular, cinco modelos foram desenvolvidos a partir dos algoritmos propostos neste trabalho, conforme descrito a seguir:

1. Rede neural nebulosa composta por neurônios *AND* na segunda camada da rede e treinada com o aprendizado participativo (**AND-AP**);
2. Rede Neural Nebulosa composta por neurônios *OR* na segunda camada da rede e treinada com o aprendizado participativo (**OR-AP**);

3. Rede Neural Nebulosa composta por unineurons na segunda camada da rede e treinada com o algoritmo baseado no reforço associativo (**UNI-R**);
4. Rede Neural Nebulosa composta por nullneurons na segunda camada da rede e treinada com o algoritmo baseado no reforço associativo (**NULL-R**);
5. Rede Neural Nebulosa composta por nullneurons na segunda camada da rede e treinada com o método do gradiente (**NULL-G**);

Para avaliar os cinco modelos desenvolvidos, estabeleceram-se as quatro métricas de desempenho que, geralmente, são empregadas na literatura para este tipo de avaliação. Estas métricas são o Erro Quadrático Médio (MSE ¹), a Raiz Quadrada do EQM (RMSE ²), o Índice de Erro Não-Dimensional (NDEI ³) e o erro percentual médio (MAPE ⁴). Estas métricas são calculadas através das seguintes expressões:

$$\text{MSE} = \frac{1}{P} \sum_{k=1}^P (y^k - \hat{y}^k)^2 \quad (5.1)$$

$$\text{RMSE} = \sqrt{\frac{1}{P} \sum_{k=1}^P (y^k - \hat{y}^k)^2} \quad (5.2)$$

$$\text{NDEI} = \frac{\text{RMSE}}{\text{std}(y^k)}, \quad k = 1, \dots, P \quad (5.3)$$

$$\text{MAPE} = \frac{1}{P} \sum_{k=1}^P |y^k - \hat{y}^k| \times \frac{100}{y^k} \quad (5.4)$$

onde y^k e \hat{y}^k são respectivamente a saída desejada e a saída obtida pela rede para a k -ésima entrada, $\text{std}(y^k)$ é o desvio padrão para os valores y^k e P é o número total de pares de treinamento considerados.

Em todos os casos foram, feitas diversas realizações de cada um dos modelos implementados, incluindo os modelos de comparação **MLP** e **ANFIS**. Assim, os procedimentos de inicialização, treinamento e teste foram executados dez vezes para cada um dos modelos em cada um dos problemas apresentados, a fim de se reduzir o efeito da aleatoriedade da inicialização. Os resultados obtidos por

¹do inglês *Mean Square Error*

²do inglês *Root Mean Square Error*

³do inglês *Non-dimensional Error Index*

⁴do inglês *Mean Average Percentage Error*

este modelos com base nas métricas estabelecidas acima para as melhores realizações de cada modelo são apresentados nas seções a seguir.

5.2 Predição de Séries Temporais

Há, atualmente, uma vasta literatura que mostra a relevância da previsão de séries temporais em diversos contextos, com exemplos que abrangem desde aplicações em medicina, medindo a irregularidade das batidas do coração; passam pelas pesquisas em hidrologia, prevendo variáveis de recursos hídricos; e chegam até as aplicações em economia, observando o comportamento da volatilidade do mercado financeiro (Ballini et al., 2000). A predição de séries temporais permite que valores futuros de uma variável sejam previstos tomando-se por base apenas seus valores presentes e passados.

Para verificar a eficácia dos modelos propostos para esta classe de problemas, estes foram empregados na previsão de carga de curto prazo, que constitui um dos mais importantes problemas no contexto do planejamento energético brasileiro. Os resultados obtidos são apresentados a seguir.

5.2.1 Previsão de Carga de Curto Prazo

Na operação de um sistema de energia elétrica, uma etapa importante é a determinação da programação da operação diária, a qual determina um plano de produção de energia elétrica para o(s) próximo(s) dia(s) para cada uma das unidades geradoras do sistema, geralmente em base horária. Esta operação é feita a partir de previsões de como será o comportamento do consumo do sistema, analisando-se um conjunto de dados que contém o histórico deste comportamento em um determinado período de tempo.

Especificamente, no contexto do planejamento da operação de curto prazo de sistemas de energia elétrica, a previsão da carga é importante na elaboração do programa de operação diária, em análises de segurança e estabilidade, pois erros na previsão da carga podem ter consequências sérias em termos da eficiência e segurança do sistema, como aumento de custos e não atendimento da demanda. Assim, uma previsão de boa qualidade é essencial, pois melhora o atendimento da demanda, aumentando a eficiência e a confiabilidade do sistema (Gross & Galiana, 1987).

Diversos trabalhos na literatura sugerem que redes neurais nebulosas são alternativas promissoras para obter modelos preditivos de curto prazo para previsão de carga em sistemas elétricos de potência (Ling et al., 2003; Chauhan et al., 2005). Neste contexto, os algoritmos propostos neste trabalho foram empregados a este problema e os resultados dos modelos são comparados entre si e com abordagens alternativas sugeridas na literatura.

Para realizar o treinamento de todos os modelos implementados, utilizou-se uma base de dados composta por medições horárias do consumo de energia em uma região residencial da cidade de São Paulo, fornecida pela empresa Eletropaulo Metropolitana Eletricidade de São Paulo S.A., no período compreendido entre abril de 2000 a abril de 2001. A Figura 5.1 ilustra um período da base de dados de treinamento compreendido entre os dias 13/04/2001 e 19/04/2001.

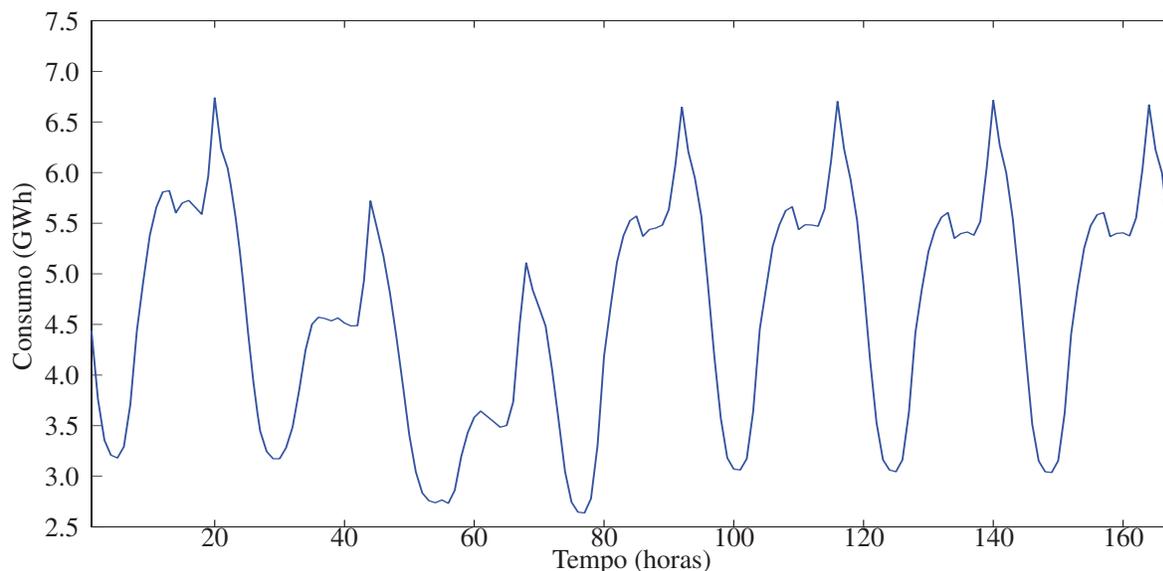


Figura 5.1: Curva de carga para o período de 13 a 19 de abril de 2001.

Já a base de dados utilizada nos testes foi composta por medições horárias feitas para o dia 16 de maio de 2001. A Figura 5.2 apresenta a curva de carga para a semana compreendida entre os dias 13 e 19 de maio de 2001, na qual estão inseridas as 24 horas que se deseja prever (16/05/2001).

Utilizando-se funções de auto-correlação parciais Ballini et al. (2000), foram adotadas duas variáveis de entrada para todos os modelos testados, sendo elas o valor da carga C com um e dois passos atrás (uma e duas horas anteriores), ou seja, $\mathbf{x} = [q^{-1}C, q^{-2}C]$, onde q^{-s} é o operador de atraso em s passos. O valor da carga no instante atual C foi tomado como sendo a saída desejada de todas os modelos implementados. Os dados de entrada e saída foram normalizados no intervalo $[0,1]$.

Além dos modelos de redes neurais nebulosas propostos neste trabalho, foram implementados modelos de redes neurais propostos na literatura. A finalidade é comparar os resultados aqui obtidos com abordagens alternativas que já se mostraram, em termos de performance de erro e custos computacionais, superiores à maioria dos modelos clássicos empregados para o mesmo fim. Entre os modelos propostos na literatura considerou-se uma rede Perceptron de múltiplas camadas (**MLP**) (Haykin, 1998), um sistema de inferência nebuloso neuro-adaptativo (**ANFIS**) (Jang, 1993) e duas

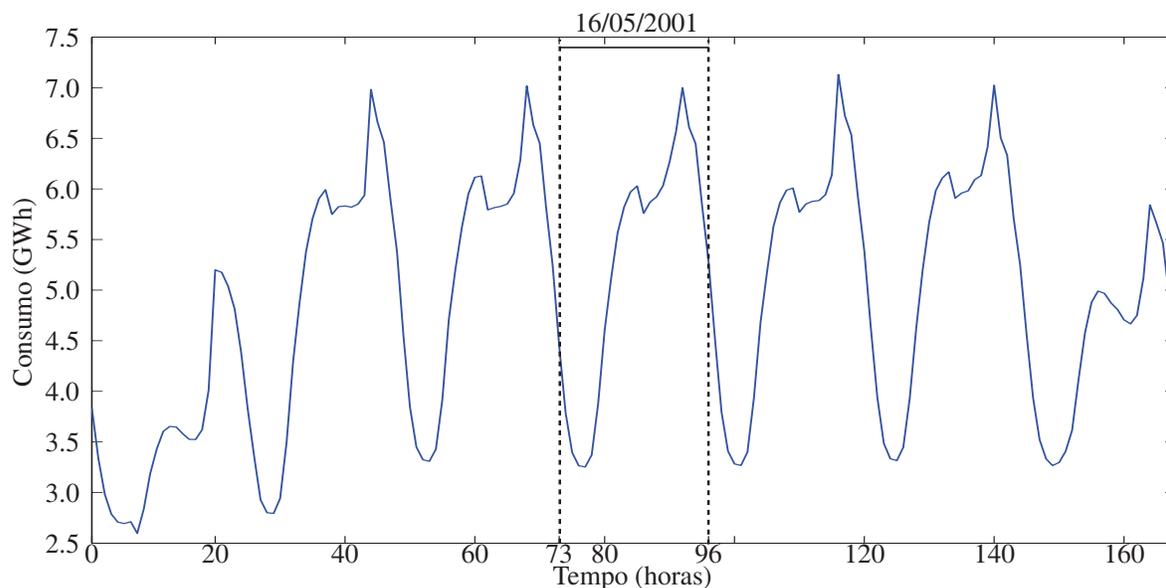


Figura 5.2: Curva de carga para o período de 13 a 19 de maio de 2001.

versões da rede neural nebulosa apresentada na seção 3.3, sendo que a primeira delas é implementada utilizando-se apenas neurônios *AND* na segunda camada da rede (**NN-AND**) e a segunda versão é implementada utilizando-se apenas neurônios *OR* nesta camada (**NN-OR**) (Hell et al., 2007a, 2008a). Ambas as versões são treinadas através de uma regra de reforço associativo.

A previsão feita aqui é do tipo um passo a frente, na qual o valor estimado pelo modelo no instante anterior não é utilizado na previsão do instante seguinte. A Tabela 5.1 apresenta a estrutura de cada um dos modelos empregados ao problema de previsão de carga, bem como o tipo de treinamento utilizado com cada um deles.

Note que o modelo **MLP** é o único que tem uma estrutura diferenciada. Isto se deve ao fato deste modelo também ser o único a possuir uma estrutura totalmente conectada, o que faz com que, mesmo com um número menor de neurônios por camada, este apresente o mesmo número de pesos de conexão que os dos demais modelos. Já o modelo **ANFIS** foi implementado com 64 regras nebulosas, o que também fornece o mesmo número de parâmetros a serem ajustados, se comparado com os outros modelos.

Com a finalidade de avaliar as características de todos os modelos, dois casos de previsão são considerados nesta seção. No primeiro caso (Caso 1), os dados utilizados durante a fase de treinamento não contêm nenhum dia especial presente nos perfis de carga. Isto quer dizer que, em todo o período no qual foram coletadas as informações presentes na base de dados de treinamento (01/04/2000 a 01/04/2001), não ocorreu nenhum tipo de evento especial que pudesse modificar a característica

Tabela 5.1: Parâmetros Estruturais dos Modelos Implementados - Previsão de Carga

Modelo	N° de Camadas	Neurônios por Camada	N° de Pesos	Tipo de Treinamento
MLP	3	[15 10 1]	190	Gradiente (GD)
ANFIS	5	[16 64 64 1 1]	192	Gradiente (GD)
NN-AND	3	[16 64 1]	192	GD + Reforço (RA)
NN-OR	3	[16 64 1]	192	GD + RA
AND-AP	3	[16 64 1]	192	AP
OR-AP	3	[16 64 1]	192	AP
UNI-R	3	[16 64 1]	192	GD + RA
NULL-R	3	[16 64 1]	192	GD + RA
NULL-G	3	[16 64 1]	192	Gradiente (GD)

sazonal da carga.

De forma contrária ao primeiro caso, no segundo caso (Caso 2), foram introduzidos na base de dados de treinamento os perfis de carga de alguns dias especiais, nos quais algum tipo de evento não sazonal altera de forma significativa a demanda de energia nestes períodos. Como exemplo gerais destes eventos não sazonais pode-se, citar um show musical realizado na praia de Copacabana para aproximadamente 1.000.000 de pessoas, blackout locais (queda de energia em toda uma região) causada por descargas atmosferas ou por causas acidentais, entre outros. A Figura 5.3 apresenta exemplos deste eventos anômalos, onde o círculo explicita a alteração causada por estes nos perfis de carga.

Assim, todos os modelos implementados foram treinados com as duas bases de dados descritas. Os testes de desempenho foram então realizados com base na previsão do consumo de energia para as 24 horas do dia 16/05/2001 (Quarta-Feira). Vale ressaltar que o perfil de carga do dia 16/05/2001 é um perfil normal, ou seja, não ocorreu nenhum evento especial neste dia.

A Tabela 5.2 apresenta os resultados obtidos por todos os modelos para o Caso 1, enquanto as Figuras 5.4 e 5.5 mostram graficamente o resultado da previsão para o mesmo caso, também para todos os modelos.

Analisando os resultados apresentados para o Caso 1, observa-se que todos os modelos propostos neste trabalho apresentam uma performance superior, tanto em termos das métricas de erro, quanto com relação aos custos computacionais envolvidos, se comparados com os métodos alternativos propostos na literatura. Em particular, os modelos baseados no unineurons e nullneurons apresentaram as melhores performances de erro com os mais baixos custos computacionais (quando associados ao treinamento por reforço associativo). Este fato demonstra experimentalmente a característica de generalização dos modelos baseados nestes novos neurônios, pois, se comparados com os demais métodos (incluindo-se aí abordagens baseadas em redes nebulosas compostas exclusivamente por

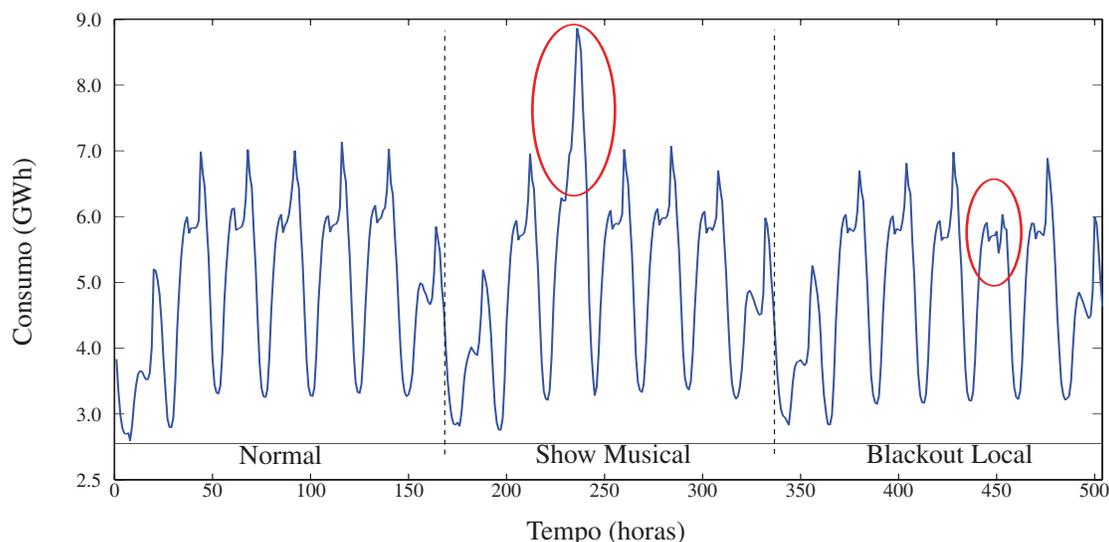


Figura 5.3: Curva de carga com perfis anômalos.

neurônios *AND* ou neurônios *OR*), estes modelos foram os que melhor se ajustaram ao conjunto de dados de treinamento, conseguindo capturar uma maior quantidade de informação a respeito da dinâmica do sistema do que os seus pares.

Vale salientar também que, um dos maiores desafios em problemas de previsão de carga está relacionado à previsão dos picos de carga diários, pois o rápido aumento no valor da carga apresentado durante estes picos torna a previsão uma tarefa complexa (Jin et al., 2005). Neste sentido, os modelos baseados no unineurons e nullneurons também apresentam as melhores performances, sendo os modelos que melhor aproximam os picos de carga.

Já para o Caso 2, o cenário é diferente. Conforme pode ser observado nos resultados da Tabela 5.3 e nos gráficos das Figuras 5.6 e 5.7, os únicos modelos capazes de manter um desempenho similar ao apresentado para o Caso 1 foram aqueles que utilizam o algoritmo de treinamento baseado no aprendizado participativo (**AND-AP** e **OR-AP**). Isto se deve ao fato de que os perfis anômalos introduzidos na base de treinamento são vistos pelo algoritmo de treinamento destes modelos como sendo muito distantes da crença que estes tem a respeito do ambiente e, portanto, não são levados em consideração durante o aprendizado. Os demais modelos não possuem este discernimento a respeito do que aprender, incorporando tudo o que lhes é apresentado durante o treinamento, fazendo com que estes não sejam robustos a dados não representativos daquilo que se deseja prever, como acontece com os modelos baseados no AP.

De fato, sempre que erros de medição ou dados espúrios estão presentes na base de dados de

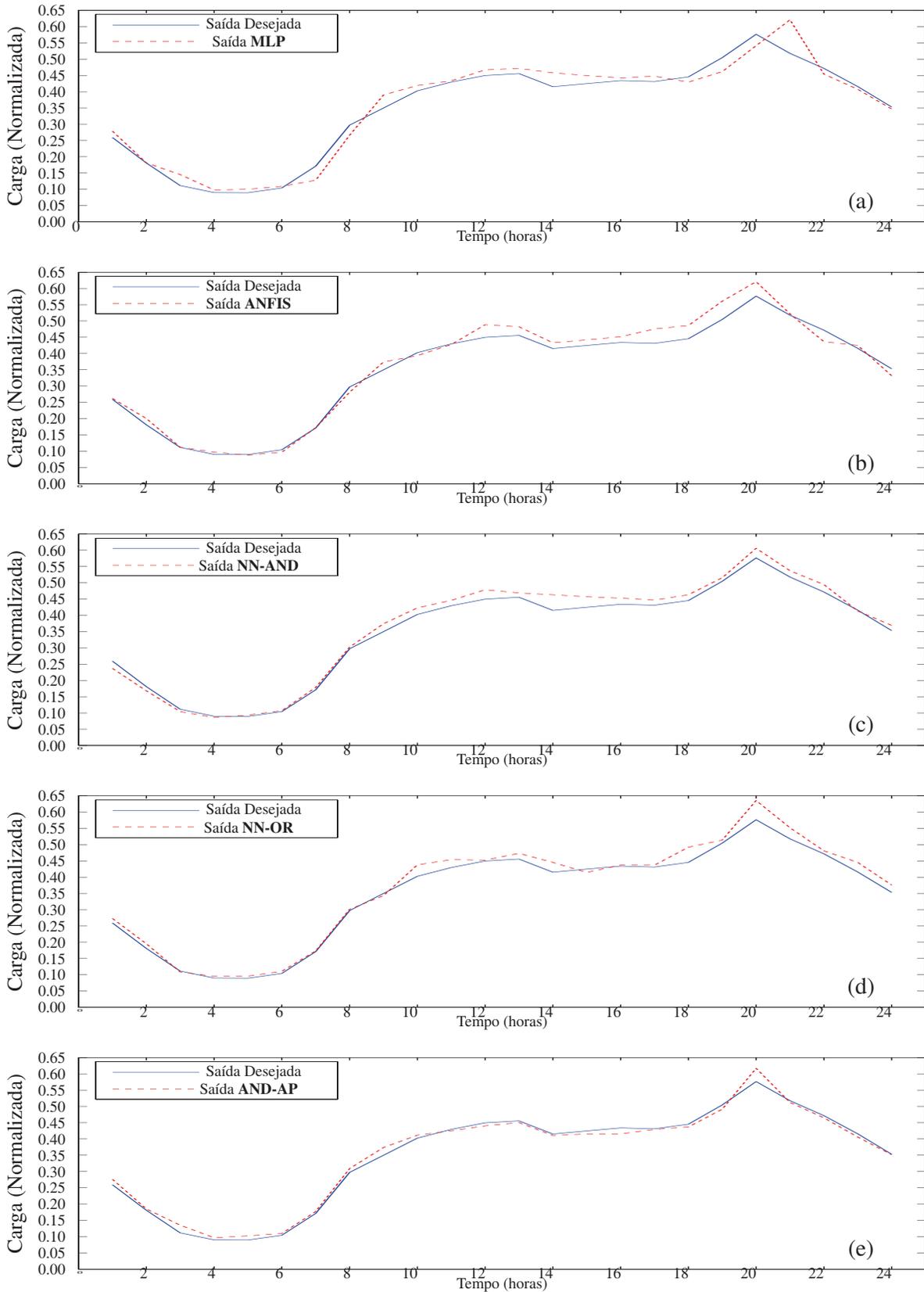


Figura 5.4: Resultados para previsão de carga - Caso 1: (a) MLP, (b) ANFIS, (c) NN-AND, (d) NN-OR, (e) AND-AP

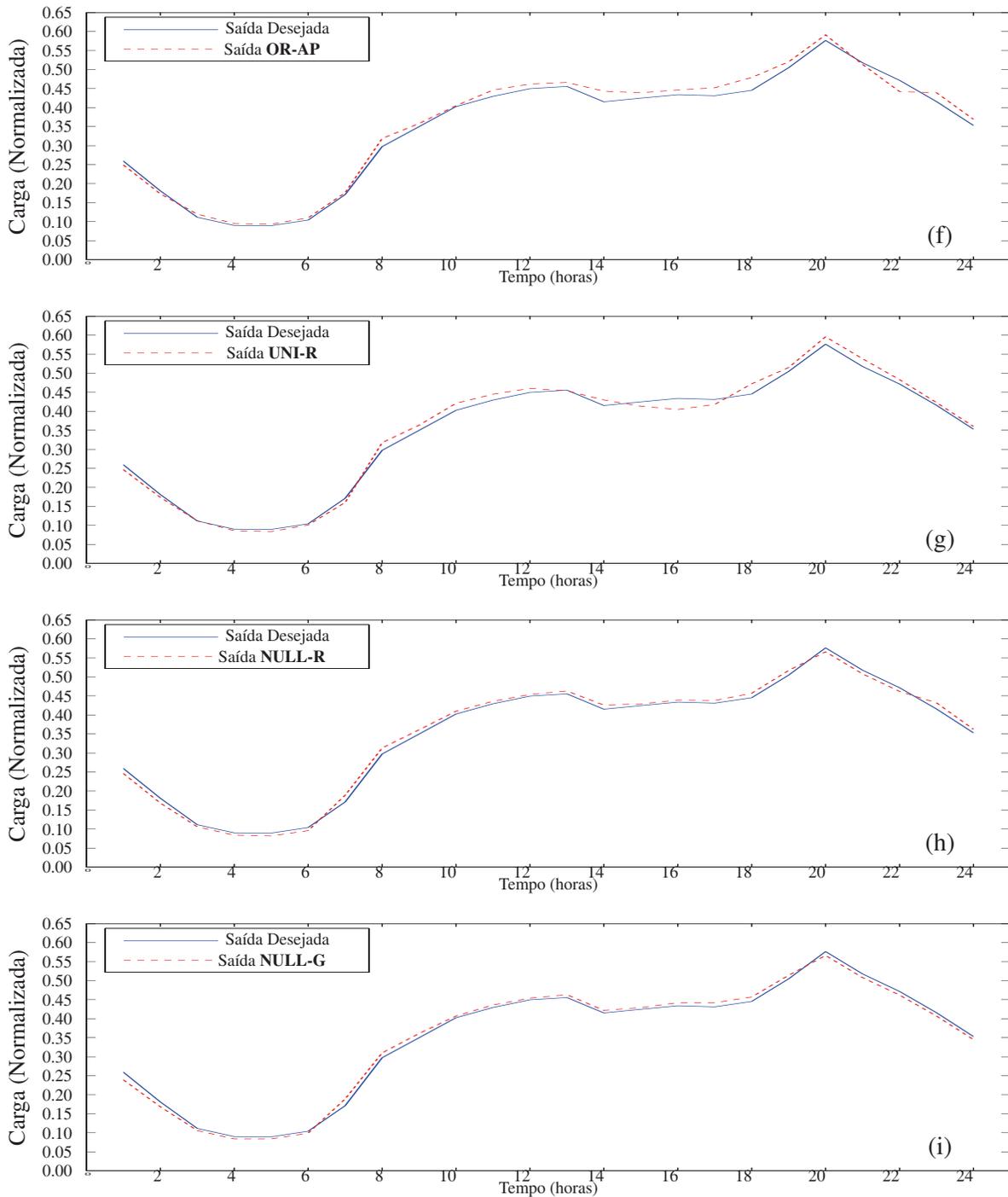


Figura 5.5: Resultados para previsão de carga - Caso 1: (f) OR-AP, (g) UNI-R, (h) NULL-R, (i) NULL-G

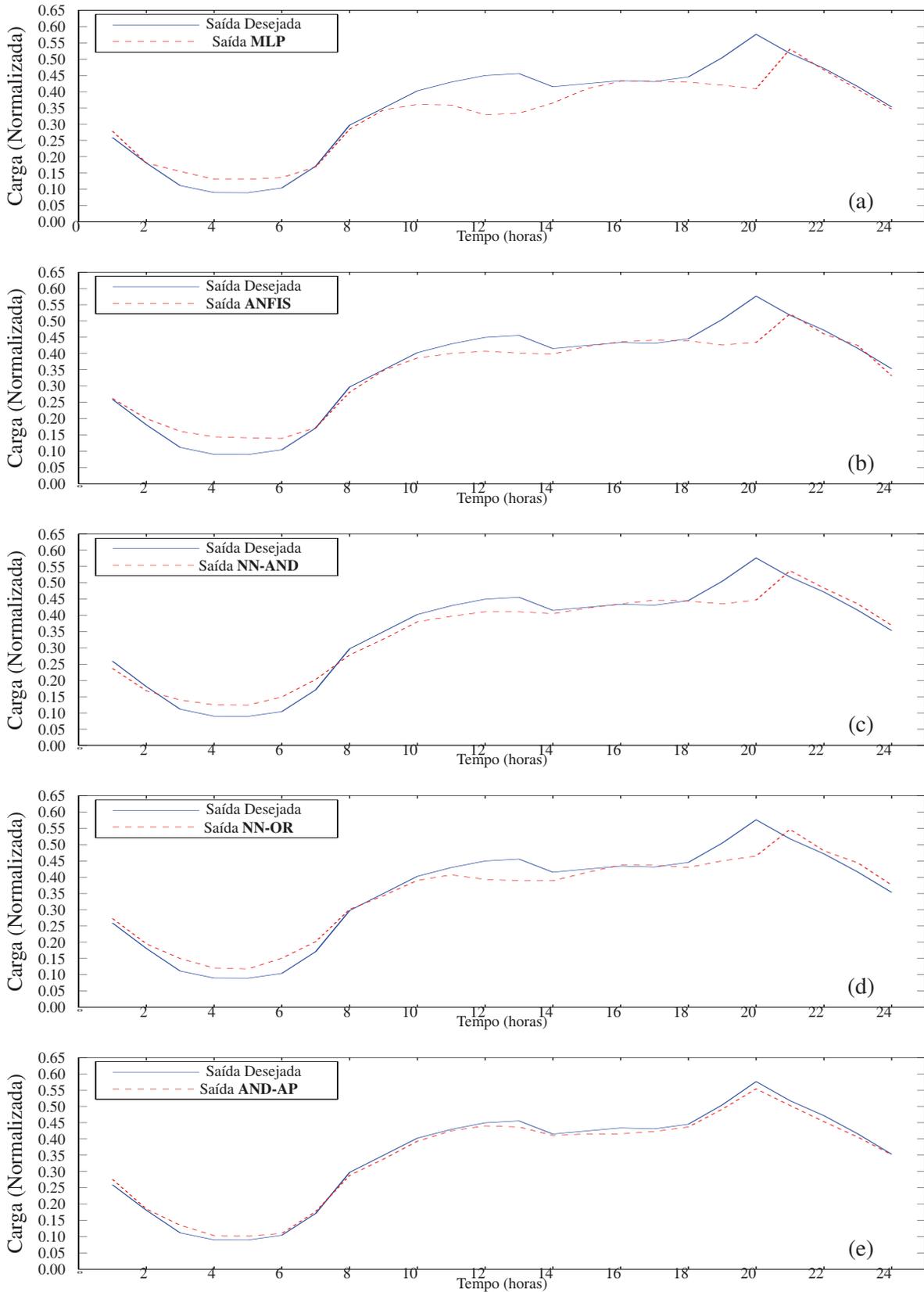


Figura 5.6: Resultados para previsão de carga - Caso 2: (a) MLP, (b) ANFIS, (c) NN-AND, (d) NN-OR, (e) AND-AP

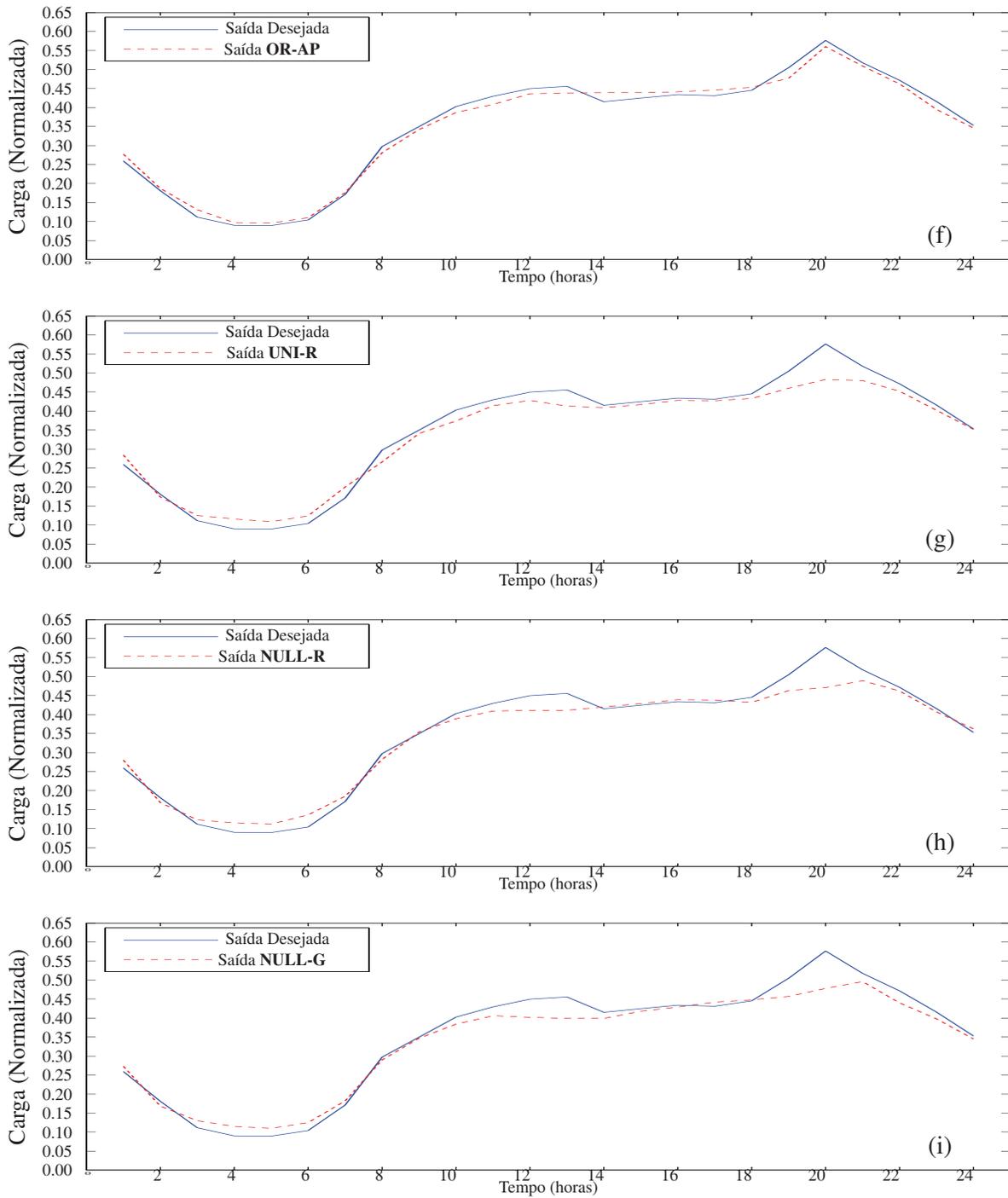


Figura 5.7: Resultados para previsão de carga - Caso 2: (f) OR-AP, (g) UNI-R, (h) NULL-R, (i) NULL-G

Tabela 5.2: Resultados para Previsão de Carga - Caso 1

Modelo	Nº de Épocas	Tempo de Trein. (min.)	MSE	RMSE	NDEI	MAPE (%)
MLP	20000	29,56	0,0010	0,0318	0,2135	7,95
ANFIS	15000	26,06	0,0006	0,0251	0,1637	5,25
NN-AND	15000	19,77	0,0004	0,0199	0,1252	4,88
NN-OR	15000	19,61	0,0005	0,0225	0,1411	4,79
AND-AP	11000	22,57	0,0002	0,0138	0,0876	4,26
OR-AP	11000	22,21	0,0003	0,0161	0,1114	4,22
UNI-R	10000	19,39	0,0002	0,0143	0,0938	3,70
NULL-R	10000	19,25	0,0001	0,0103	0,0661	3,68
NULL-G	10000	21,89	0,0001	0,0099	0,0650	3,42

Tabela 5.3: Resultados para Previsão de Carga - Caso 2

Modelo	Nº de Épocas	Tempo de Trein. (min.)	MSE	RMSE	NDEI	MAPE (%)
MLP	20000	29,15	0,0034	0,0582	0,3914	13,74
ANFIS	15000	25,87	0,0018	0,0425	0,3607	12,79
NN-AND	15000	19,09	0,0015	0,0387	0,3222	11,83
NN-OR	15000	18,91	0,0014	0,0376	0,2926	11,71
AND-AP	11000	22,24	0,0002	0,0128	0,1018	4,61
OR-AP	11000	22,01	0,0002	0,0147	0,1064	4,56
UNI-R	10000	18,13	0,0009	0,0292	0,2057	8,28
NULL-R	10000	18,65	0,0009	0,0300	0,2342	8,19
NULL-G	10000	21,19	0,0010	0,0309	0,2388	8,16

treinamento, os sistemas baseados no AP apresentam melhor desempenho na modelagem do comportamento normal do problema. Análises similares realizadas nas próximas seções confirmarão esta afirmação.

5.3 Aplicações a Problemas de Identificação de Sistemas

O intuito dos estudos em identificação de sistemas dinâmicos é o de, considerando uma seqüência de observações a respeito de um sistema real, obter-se uma realização paramétrica de um sistema matemático que consiga apresentar o mesmo comportamento dinâmico do sistema real quando ambos são submetidos às mesmas entradas.

Da mesma forma que para os problemas de previsão de séries temporais, também existe uma vasta literatura para problemas de identificação de sistemas, com aplicações nos mais diversos campos da

ciência. Neste trabalho, foram considerados dois problemas específicos a serem tratados pelos modelos aqui propostos. O primeiro destes problemas é o problema clássico do forno a gás de Box e Jenkins. No segundo problema, deseja-se modelar o comportamento térmico de um transformador de potência a partir de leituras de sua corrente de carga e das condições climáticas a que esse equipamento está exposto.

As seções a seguir apresentam com mais detalhes estes problemas, bem como os resultados obtidos pelos modelos baseados nas redes neurais nebulosas aqui propostas.

5.3.1 Problema do Forno a Gás de Box&Jenkins

O problema de identificação do forno a gás proposto por Box e Jenkins é um dos problemas mais estudados na área de modelagem e identificação de sistemas. O processo de identificação utiliza-se de um total de 296 amostras para gerar 290 pares entrada/saída obtidos a partir de um forno a gás de laboratório (Box & Jenkins, 1994). Cada amostra consiste das taxas de fluxo de metano (entradas x^k do sistema) e a concentração de CO_2 na saída do forno (saídas y^k do sistema). Este é um processo dinâmico com uma entrada x^k e uma saída y^k . O objetivo é estimar a saída corrente y^k a partir de entradas e saídas em instantes anteriores, com o menor erro possível.

Diferentes estudos ((Farag et al., 1998);(Xiao-Zhi & Ovaska, 2006);entre outros) indicam que a melhor estrutura do modelo para este sistema é:

$$y^k = f(y^{k-1}, x^{k-4}). \quad (5.5)$$

Em nossos experimentos foram definidos, empiricamente, 3 conjuntos nebulosos para granularizar cada variável de entrada das redes neurais nebulosas que compõem os modelos **NN-AND**, **NN-OR**, **AND-AP**, **OR-AP**, **UNI-R**, **NULL-R** e **NULL-G**. Conseqüentemente, estes modelos são compostos por 9 unidades de processamento na segunda camada, representando um sistema de inferência com 9 regras nebulosas. O sistema **ANFIS** também foi implementado como o mesmo número de regras nebulosas e a estrutura da rede **MLP** foi formada por apenas uma camada intermediária com 9 neurônios nesta camada.

Do total de amostras, 200 foram selecionadas para serem utilizadas no processo de treinamento. As demais 90 amostras foram reservadas para os testes de validação dos modelos obtidos. A avaliação dos modelos é feita considerando-se as mesmas métricas de erro usadas na seção anterior, conforme expressões (5.1) a (5.4).

A Tabela 5.4 apresenta os resultados obtidos para os modelos aqui considerados. Além dos resultados da Tabela 5.4, outros obtidos por modelos alternativos propostos na literatura permitem mostrar que os modelos propostos neste trabalho são competitivos no meio ao qual estes se inserem.

A Tabela 5.5 mostra os resultados obtidos por cinco abordagens alternativas, juntamente com seus autores e referências, utilizando as mesmas métricas de erro para comparação.

Tabela 5.4: Resultados para o Forno a Gás de Box&Jenkins

Modelo	Nº de Épocas	Tempo de Trein. (seg.)	MSE	RMSE	NDEI	MAPE (%)
MLP	10000	22,45	0,5817	0,7627	0,2389	1,25
ANFIS	5000	19,79	0,5112	0,7150	0,2240	1,22
NN-AND	10000	17,45	0,4766	0,6904	0,2162	1,05
NN-OR	10000	16,39	0,4899	0,6999	0,2192	1,02
AND-AP	8000	15,57	0,4493	0,6703	0,2100	0,98
OR-AP	8000	15,96	0,4528	0,6729	0,2108	0,96
UNI-R	10000	10,55	0,3974	0,6304	0,1975	0,85
NULL-R	10000	11,30	0,3861	0,6214	0,1946	0,83
NULL-G	10000	14,34	0,4034	0,6351	0,1989	0,87

Tabela 5.5: Resultados da Literatura para o Forno a Gás de Box&Jenkins

Modelo	MSE	RMSE	NDEI
Pedrycz (Pedrycz, 1984)	0,5656	0,7521	0,2356
Xu e Lu (Xu & Lu, 1987)	0,5727	0,7568	0,2370
Delgado (Delgado et al., 1997)	0,4100	0,6403	0,2006
Yoshinari (Yoshinari et al., 1993)	0,5460	0,7389	0,2315
Rutkowski (Rutkowski & Cpalka, 2005)	0,4919	0,7014	0,2197

As Figuras 5.8 e 5.9 apresentam a saída obtida para os modelos implementados neste trabalho quando se utilizam as amostras de teste. Estes resultados mostram que os modelos propostos são, no mínimo, tão bons quanto as melhores abordagens sugeridas na literatura. Em particular o modelo **NULL-R** apresentou um desempenho muito satisfatório, obtendo as menores de todas as taxas de erro, com um dos menores custos computacionais. Este fato o torna um excelente candidato para ser utilizado na modelagem de sistemas complexos, como também poderá ser observado na seção seguinte.

5.3.2 Modelagem Térmica de Transformadores de Potência

Transformadores de potência são equipamentos importantes na transmissão e distribuição de energia elétrica, e o seu correto funcionamento é vital para a operação do sistema. A perda de uma unidade

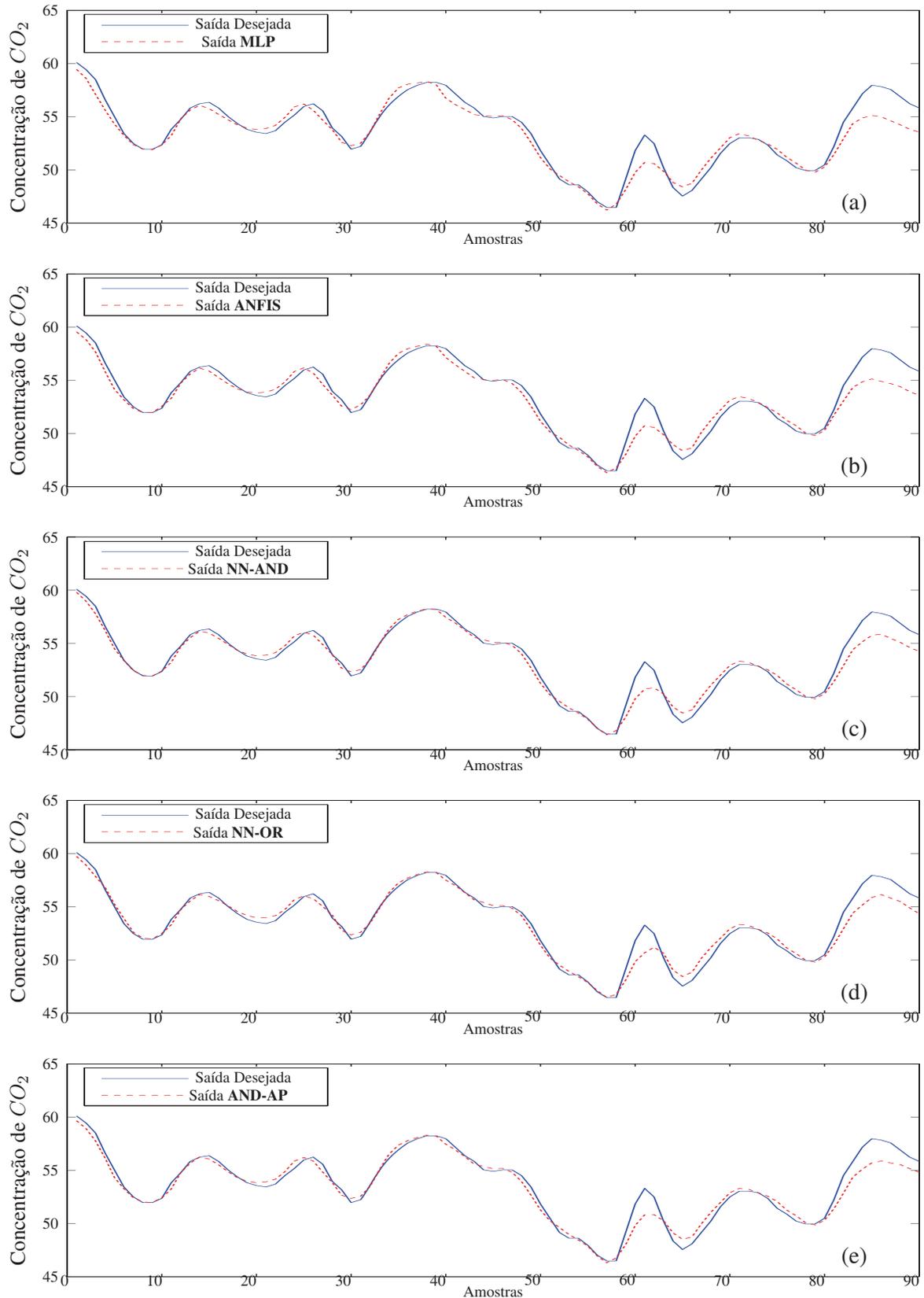


Figura 5.8: Resultados obtidos para o forno a gás de Box e Jenkins: (a) MLP, (b) ANFIS, (c) NN-AND, (d) NN-OR, (e) AND-AP

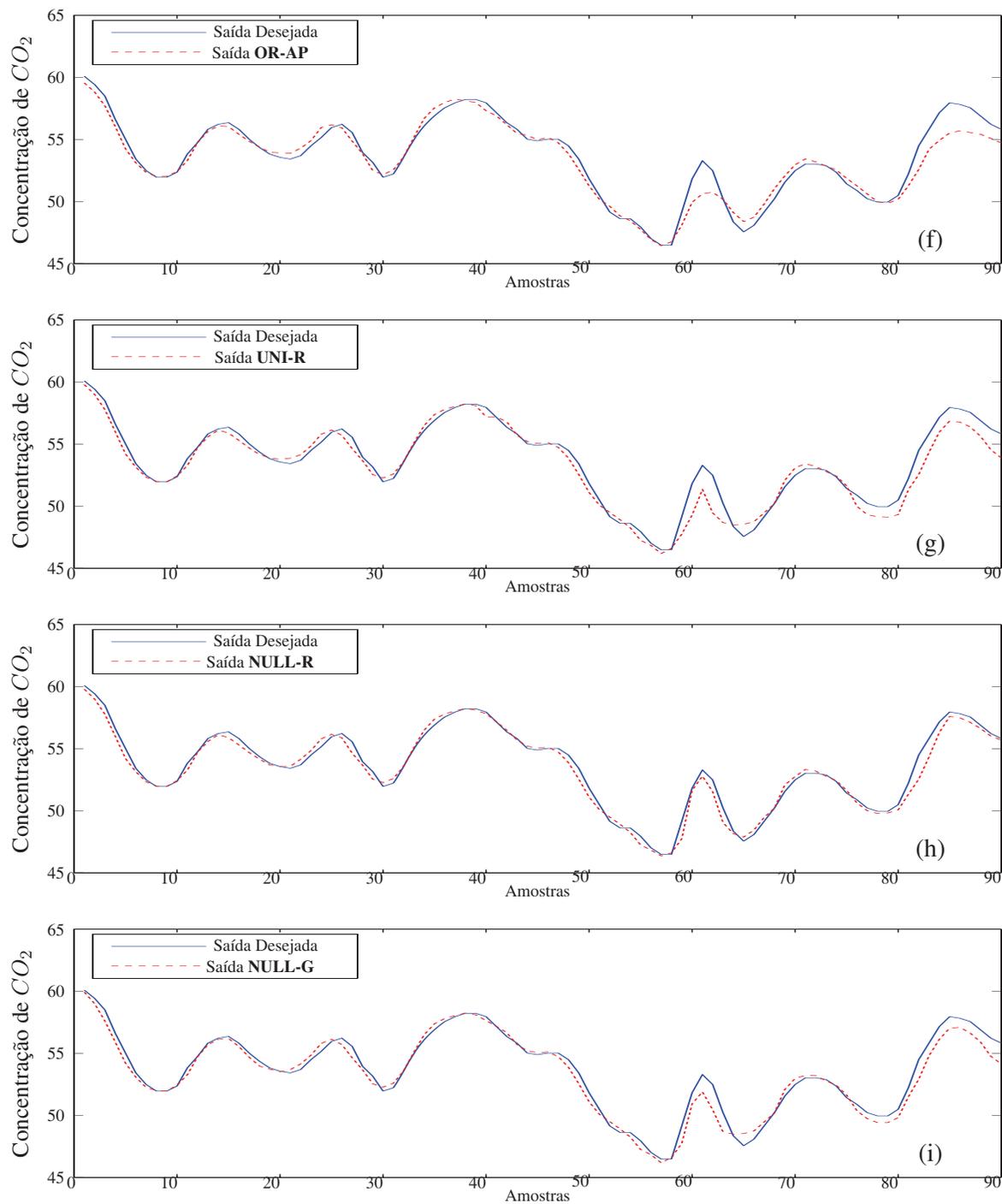


Figura 5.9: Resultados obtidos para o forno a gás de Box e Jenkins: (f) **OR-AP**, (g) **UNI-R**, (h) **NULL-R**, (i) **NULL-G**

estratégica pode ter um grande impacto na segurança, na confiabilidade e no custo do fornecimento de energia.

Um dos principais fatores na operação dos transformadores de potência é a temperatura de trabalho, mais especificamente a temperatura do ponto mais quente, ou temperatura de *hot-spot*, no topo ou no centro dos enrolamentos de alta ou baixa tensão. Esta temperatura tem uma importante influência no envelhecimento do isolamento e, conseqüentemente, na vida útil do equipamento (Hell et al., 2007c).

O valor desta temperatura pode ser calculado utilizando modelos derivados das equações de transferência de calor e características térmicas específicas do transformador. Um dos principais modelos é o seguinte (Swift, 2001):

$$\left\{ \begin{array}{l} \tau_{TO} \frac{d\Theta_{TO}}{dt} = [\Delta\Theta_{TO,U} + \Theta_A] - \Theta_{TO} \\ \tau_H \frac{d\Delta\Theta_H}{dt} = \Delta\Theta_{H,U} - \Delta\Theta_H \\ \Delta\Theta_{TO,U} = \Delta\Theta_{TO,R} \left[\frac{K^2 R + 1}{R + 1} \right]^n \\ \Delta\Theta_{H,U} = \Delta\Theta_{H,R} K^{2m} \\ \Theta_H = \Theta_{TO} + \Delta\Theta_H \end{array} \right. \quad (5.6)$$

onde τ_{TO} é a constante de tempo da temperatura de topo de óleo, Θ_{TO} é a temperatura de topo de óleo, $\Delta\Theta_{TO,U}$ é a elevação máxima da temperatura de topo de óleo, Θ_A é a temperatura ambiente, τ_H é a constante de tempo da temperatura de *hot-spot*, $\Delta\Theta_H$ é a elevação da temperatura de *hot-spot* acima da temperatura de topo de óleo, $\Delta\Theta_{H,U}$ é a elevação máxima da temperatura de *hot-spot*, $\Delta\Theta_{TO,R}$ é a elevação nominal da temperatura de topo de óleo acima da temperatura ambiente, K é a corrente de carga, R é a razão entre perda de carga em relação a carga nominal e carga não perdida para uma determinada posição de tap, $\Delta\Theta_{H,R}$ é a elevação nominal da temperatura de *hot-spot* acima da temperatura de topo de óleo, Θ_H é a temperatura de *hot-spot* no enrolamento e m e n são constantes obtidas empiricamente que dependem do método de refrigeração do equipamento.

A execução destes cálculos não é apenas uma tarefa de grande complexidade, como também leva a estimativas conservativas baseadas em algumas suposições das condições de operação. Neste cálculo são adotados fatores de segurança extremamente conservadores, visando proteger o equipamento, fazendo com que a transferência máxima de potência seja de 20 a 30% menor que a sua capacidade nominal, com a finalidade de se aumentar a sua vida útil e reduzir os riscos durante a operação (Galdi et al., 2000).

Neste sentido, a obtenção de modelos mais precisos para simular o comportamento térmico destes importantes equipamentos pode trazer grandes vantagens, tanto para a redução de custos de operação, quanto para o aumento da confiabilidade do sistema. Assim, as redes neurais nebulosas propostas

Tabela 5.6: Principais Características do Transformador a ser Modelado

Potência Nominal	25 kVA
Vprimária / Vsecundária	10 kV / 380 V
Perdas no Ferro	195 W
Perdas no Cobre	776 W
Temp. de Topo de óleo a Plena Carga	73.1 °C
Comprimento x Largura x Profundidade do tanque	64 x 16 x 80 cm
Tipo de Refrigeração	ONAN
Fabricante / ano	MACE/87

neste trabalho foram empregadas com o intuito de modelar o comportamento da temperatura de *hot-spot* Θ_H .

Diferentes estudos (Galdi et al., 2000; Villaci et al., 2005) e testes experimentais de tentativa e erro indicam que as entradas mais relevantes para os modelos são a corrente de carga (K), a temperatura de topo de óleo (Θ_{TO}) e a corrente de carga atrasada em um passo ($K q^{-1}$). Esta escolha tem mostrado uma redução da sensibilidade dos modelos com relação a flutuações nos parâmetros térmicos, que podem variar consideravelmente de um transformador para outro.

Os dados utilizados em nossos experimentos são os mesmos reportados em (Galdi et al., 2000). Estes dados foram coletados a partir de medições realizadas em um transformador experimental de potência, com enrolamentos em disco. As medições de temperatura utilizam sensores de fibra ótica inseridos nos espaçadores localizados entre os discos dos enrolamentos de alta e de baixa tensão bem como no topo do tanque onde estes enrolamentos estão imersos. O sistema de aquisição de dados coleta os valores de todos os sensores em intervalos de 5 minutos. Maiores informações a respeito da forma de aquisição dos dados podem ser encontradas em (Villaci et al., 2005). As principais características do transformador utilizado neste trabalho são apresentadas na Tabela 5.6.

Da mesma forma que feito anteriormente, os sistemas são inicialmente treinados para um conjunto de dados onde nenhum tipo de anomalia afeta os dados amostrados (Caso 1), isto é, onde não ocorrem erros de medição durante a aquisição de dados. Estes dados descrevem o comportamento da temperatura de *hot-spot* do equipamento para um carregamento de 24 horas, com amostras registradas a cada 5 minutos, como mostrado na Figura 5.10.

Após o treinamento, os modelos neurais nebulosos são utilizados para estimar a temperatura de *hot-spot* para dois conjuntos de dados (dados de teste), descrevendo duas situações diferentes de carregamento a que o transformador está sujeito durante operação normal. Na primeira situação (**DT1**), as condições de carga não ultrapassam os valores nominais do transformador, ou seja, é uma condição sem sobrecargas. Na segunda situação (**DT2**), uma sobrecarga de 30% do valor nominal

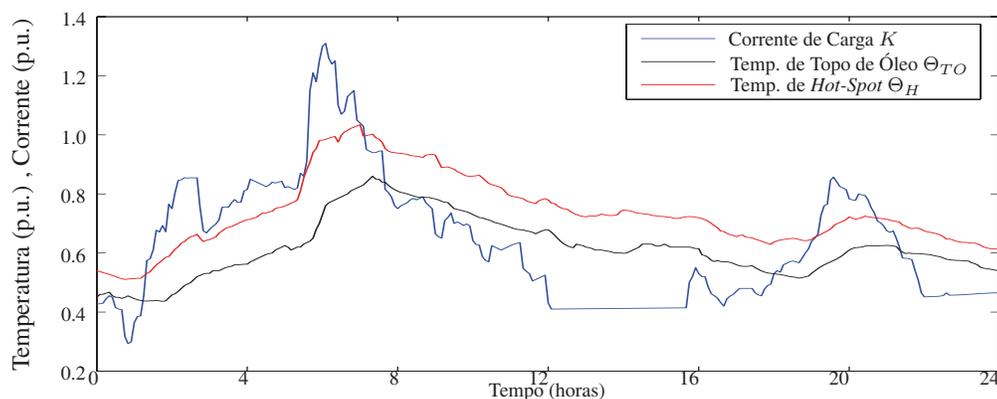


Figura 5.10: Dados de treinamento - Caso 1.

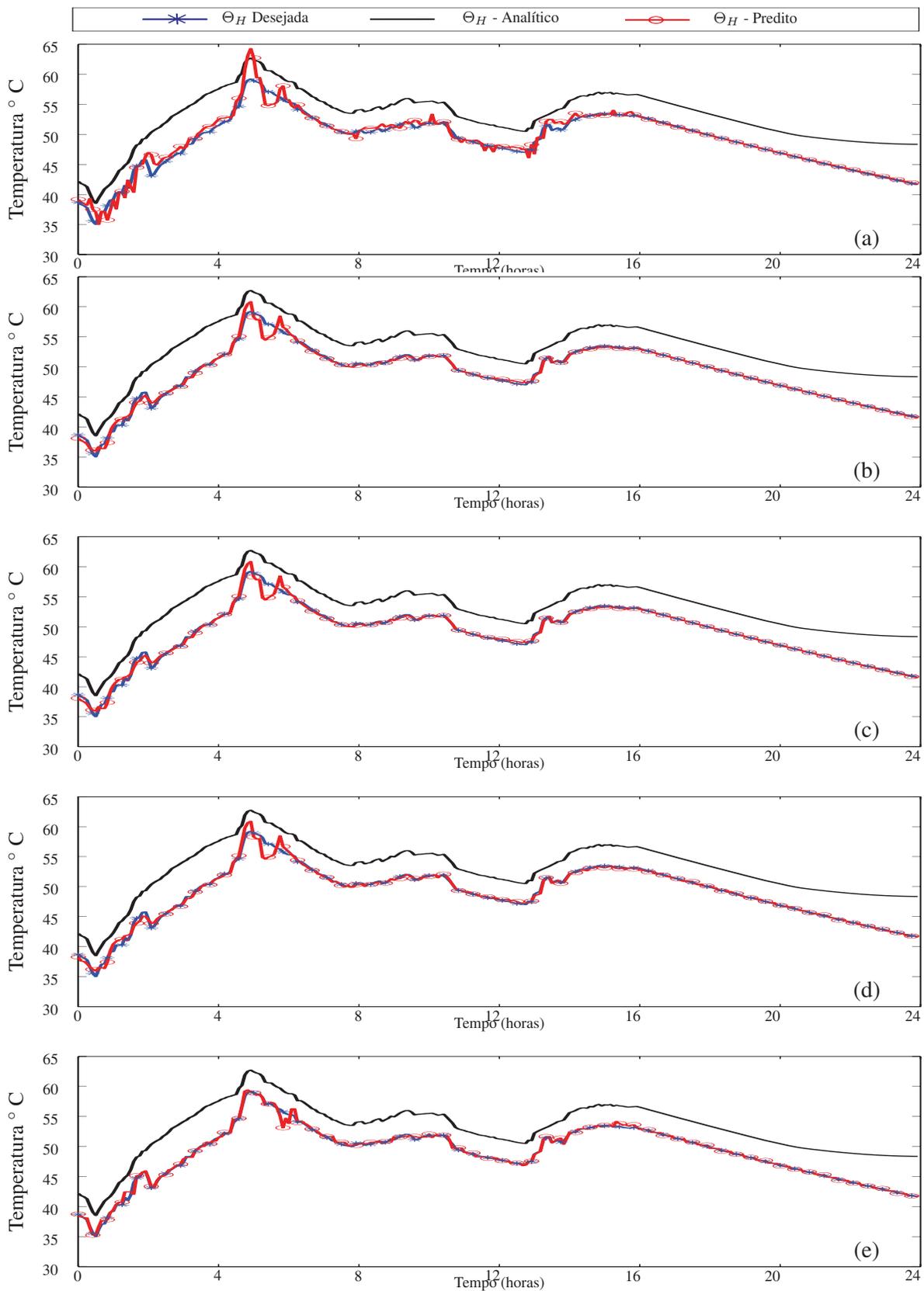
é observada durante um período de 4 horas. Os resultados para este caso (Caso 1) em ambas as situações de carregamento, são apresentados na Tabela 5.7. O resultado do modelo analítico descrito por (5.6) é também incluído para comparação.

Tabela 5.7: Resultados da Modelagem da Transformadores - Caso 1

Modelo	Nº de Épocas	Tempo de Trein. (seg.)	MSE - DT1	MSE - DT2
Analítico	-	-	17,3490	6,7822
MLP	22700	93,07	0,7901	0,2635
ANFIS	15000	84,71	0,2102	0,8168
NN-AND	11000	61,50	0,2689	0,2334
NN-OR	11000	60,94	0,2565	0,2591
AND-AP	11000	67,32	0,2104	0,2203
OR-AP	11000	68,03	0,2037	0,2535
UNI-R	10000	60,45	0,1109	0,1934
NULL-R	10000	61,74	0,0910	0,1885
NULL-G	10000	73,76	0,0907	0,1822

As Figuras 5.3.2 e 5.11 mostram os resultados graficamente para a situação de operação sem sobrecarga (**DT1**), enquanto as Figuras 5.12 e 5.13 apresentam os gráficos de resultado para a situação com sobrecarga (**DT2**).

Em um segundo momento, com a finalidade de mais uma vez verificar a característica do aprendizado participativo que faz com que os modelos baseados neste paradigma não sejam tão influenciados por erros nos dados de treinamento, foi gerado um segundo caso (Caso 2). Agora, considera-se que os dados de treinamento foram afetados por um mal funcionamento em um dos sensores de fibra



caption Resultados da modelagem térmica de transformadores de potência (Dados de Teste 1 - DT1):
(a) MLP, (b) ANFIS, (c) NN-AND, (d) NN-OR, (e) AND-AP, (f) OR-AP

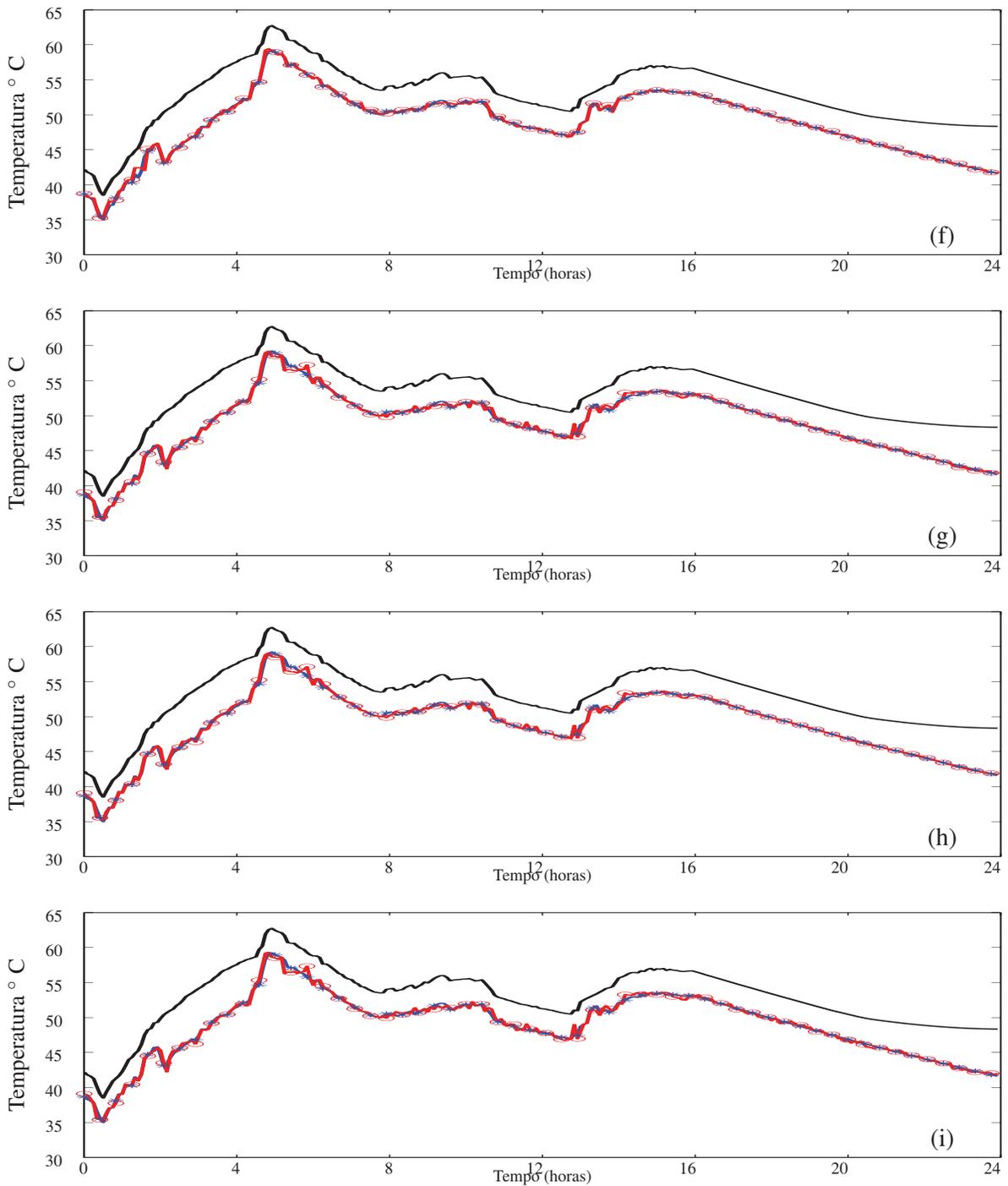


Figura 5.11: Resultados da modelagem térmica de transformadores de potência (Dados de Teste 1 - DT1): (f) OR-AP, (g) UNI-R, (h) NULL-R, (i) NULL-G

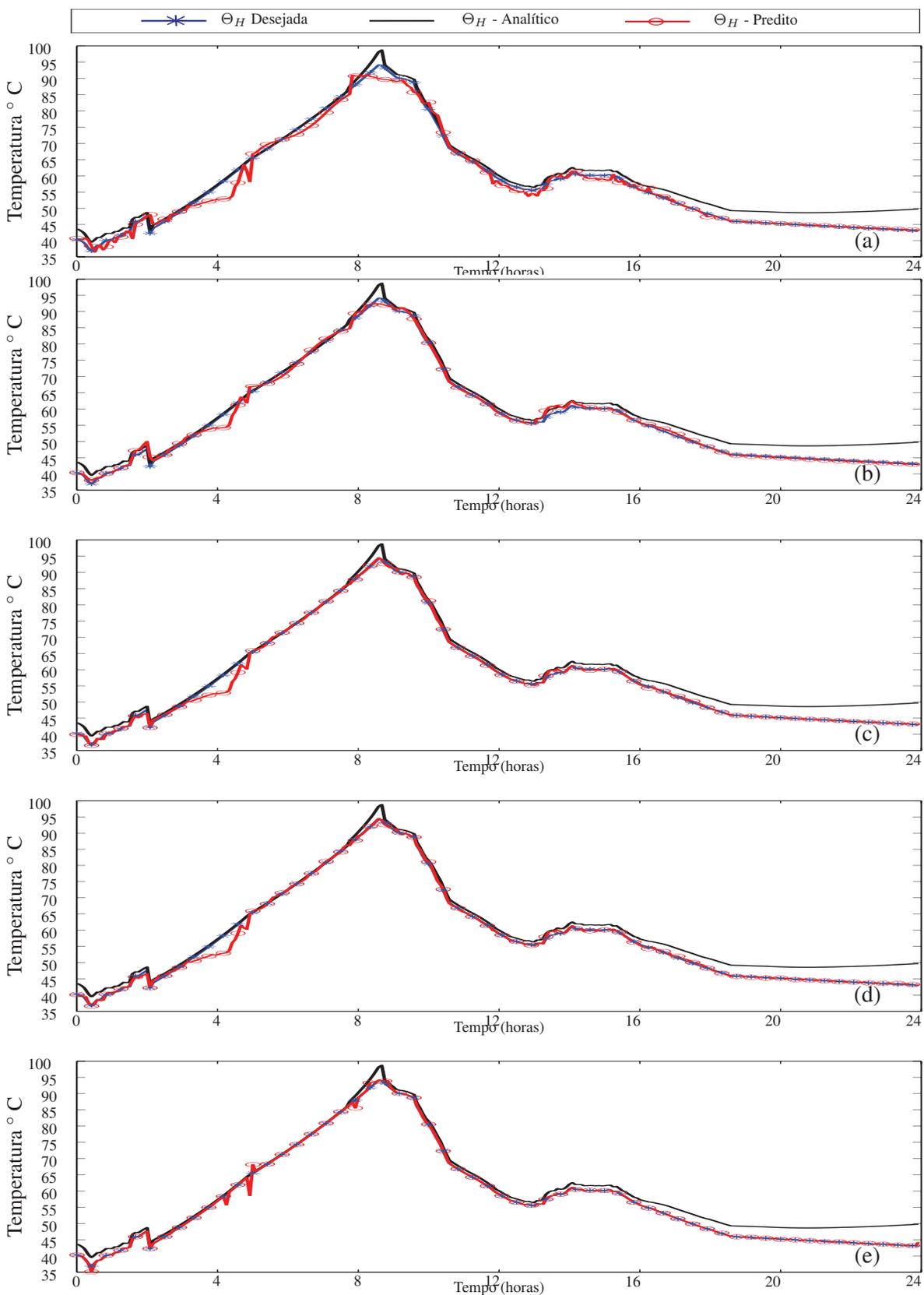


Figura 5.12: Resultados da modelagem térmica de transformadores de potência (Dados de Teste 2 - DT2): (a) MLP, (b) ANFIS, (c) NN-AND, (d) NN-OR, (e) AND-AP

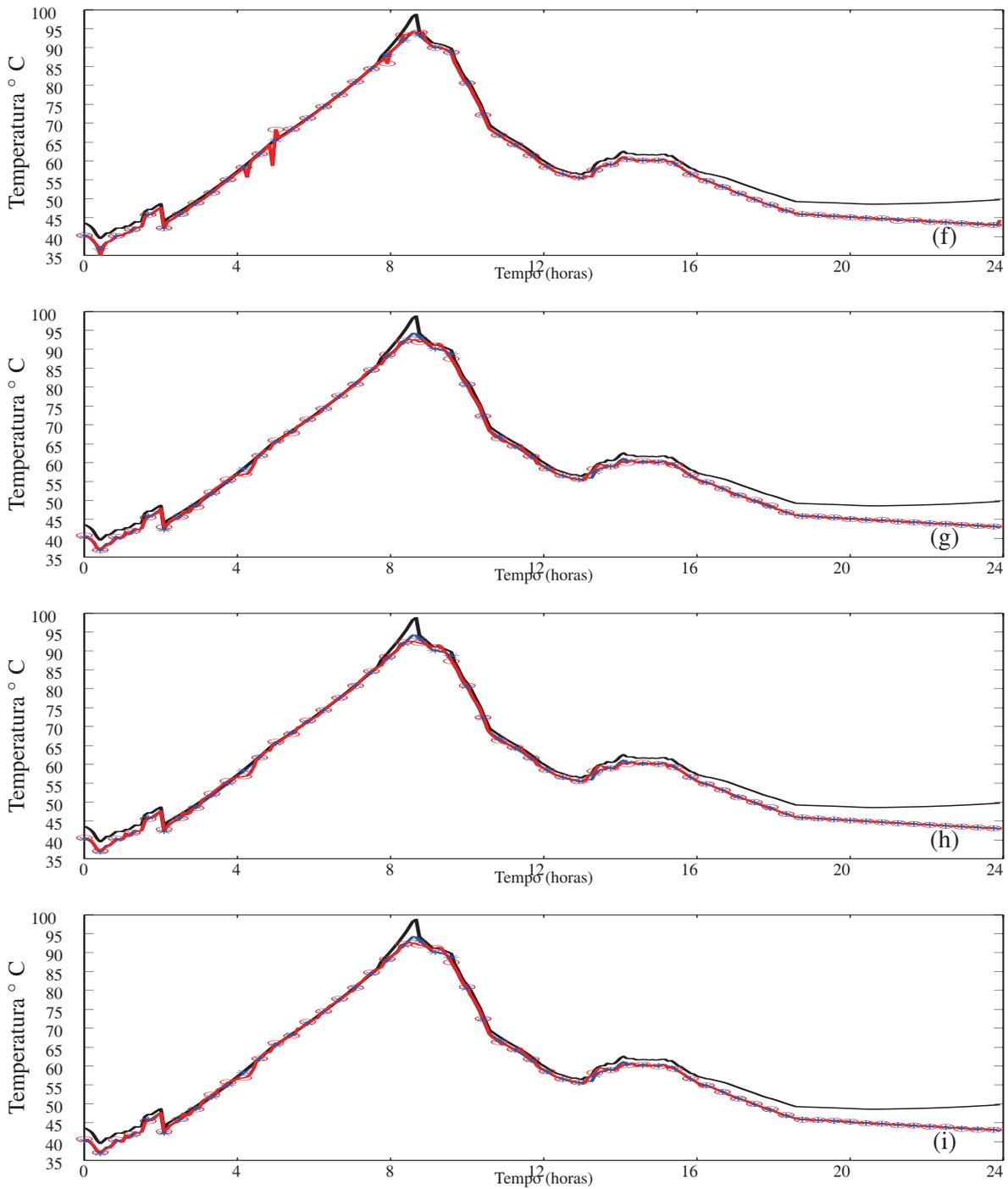


Figura 5.13: Resultados da modelagem térmica de transformadores de potência (Dados de Teste 2 - DT2): (f) OR-AP, (g) UNI-R, (h) NULL-R, (i) NULL-G

ótica que coletam as temperaturas internas do transformador. A partir desta premissa foram gerados os dados mostrados na Figura 5.14.

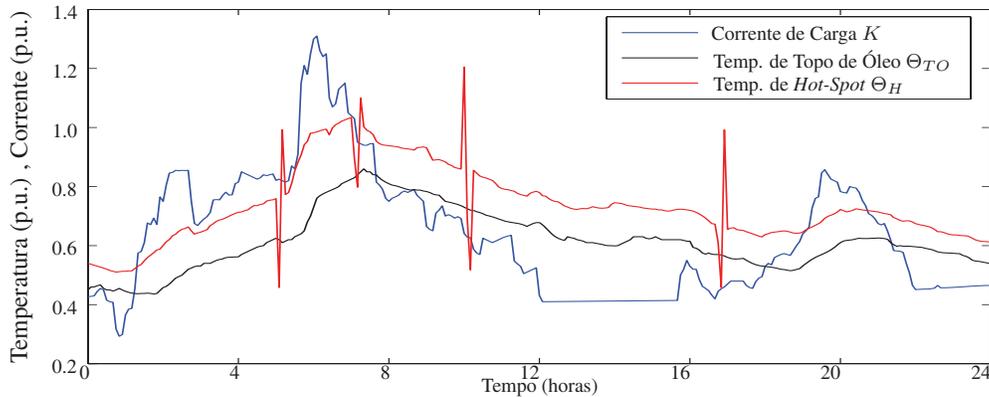


Figura 5.14: Dados de treinamento - Caso 2.

Os modelos **MLP**, **ANFIS**, **NN-AND**, **NN-OR**, **AND-AP**, **OR-AP**, **UNI-R**, **NULL-R** e **NULL-G** são então treinados com essa nova base de dados e posteriormente aplicados para estimar a temperatura de *hot-spot* para as mesmas duas situações de carregamento (nominal e de sobrecarga) discutidas anteriormente. Os resultados são apresentados na Tabela 5.8.

Tabela 5.8: Resultados da Modelagem de Transformadores - Caso 2

Modelo	Nº de Épocas	Tempo de Trein. (seg.)	MSE - DT1	MSE - DT2
Analítico	-	-	17,3490	6,7822
MLP	22700	95,23	3,2103	1,2301
ANFIS	15000	83,55	2,6608	4,0151
NN-AND	11000	63,24	1,1025	1,2430
NN-OR	11000	61,32	1,1820	1,3063
AND-AP	11000	66,87	0,2197	0,2499
OR-AP	11000	68,51	0,2113	0,2917
UNI-R	10000	65,99	1,5342	1,8431
NULL-R	10000	64,83	1,5917	1,8974
NULL-G	10000	74,48	1,6195	1,9886

Mais uma vez, é possível observar que, para os casos em que a base de dados de treinamento não é corrompida por algum tipo de anormalidade, os modelos baseados nos unineurons e nullneurons superam todos os demais, tanto em termos de performance de erro quanto em custos computacionais. Por outro lado, na presença de dados de treinamento não ideais, os únicos modelos que são capazes

de garantir um bom desempenho na solução do problema em estudo são aqueles cujos processos de treinamento são baseados no aprendizado participativo (Hell et al., 2008b). Portanto, observa-se que a associação das duas técnicas citadas poderia produzir um modelo de alto desempenho, baixo custo computacional e robusto à anomalias presentes na base de dados de treinamento.

Entretanto, para que tal associação fosse possível, seria necessário que uninormas e nullnormas pudessem ser vistas como operadores relacionais nebulosos e, conseqüentemente, pudessem ser utilizadas na síntese de novas classes de composições de relações nebulosas associadas à uma solução de problema inverso nebuloso para tais composições.

A teoria de operadores residuais aplicados a uninormas e nullnormas (Baets & Fodor, 1999; Ruiz-Aguilera & Torrens, 2007) fornece uma ferramenta que pode ser utilizada para este fim. De fato, o estudo de operadores residuais e sua aplicação a uninormas e nullnormas a fim de se propor a solução de problema inverso para relações nebulosas compostas por estas normas constituem uma proposição para futuros trabalhos relacionados às redes neurais nebulosas propostas neste trabalho.

5.4 Resumo

Este capítulo apresentou resultados de simulação obtidos utilizando as redes neurais nebulosas na modelagem de sistemas dinâmicos não-lineares, ilustrada com exemplos de previsão de séries temporais e modelagem térmica de transformadores.

Os resultados de simulação e comparações com abordagens alternativas propostas na literatura demonstram a capacidade dos modelos neurais nebulosos propostos neste trabalho de modelar os sistemas dinâmicos não-lineares, geralmente com um número reduzido de parâmetros e iterações, acarretando em menores taxas de erros e tempo de processamento.

Em particular, os modelos baseados em unineurons e nullneurons demonstraram uma performance superior para os casos em que as bases de dados utilizadas nos processos de treinamento destes modelos não continham nenhum tipo de anomalia. Já os modelos cujos processos de treinamento são baseados no aprendizado participativo se mostraram robustos à perturbações nos dados de treinamento, mantendo uma performance satisfatória quanto ao erro e sobressaindo-se assim em relação às demais alternativas.

Capítulo 6

Conclusão

Sistemas neurais nebulosos têm demonstrado grande aplicabilidade nas mais diversas áreas do conhecimento, tais como economia, matemática difusa, teoria de jogos, modelagem de sistemas, entre tantas outras. Resultado da combinação natural de dois dos maiores paradigmas da inteligência computacional, estes sistemas permitem a construção de modelos robustos e eficientes para a resolução de problemas de modelagem e mostram-se como uma opção atraente para diversos campos. Em particular, a unificação das redes neurais com os sistemas nebulosos para a construção de modelos de sistemas dinâmicos não-lineares se mostra promissora e constitui o principal motivo de estudo deste trabalho.

Duas novas abordagens baseadas em redes neurais nebulosas foram desenvolvidas com a finalidade de obter modelos de sistemas dinâmicos precisos e robustos, com baixo custo computacional.

A primeira destas abordagens constitui um novo algoritmo de aprendizado baseado no paradigma do aprendizado participativo. Neste algoritmo de treinamento, o que o sistema já aprendeu a respeito do problema que se deseja tratar é visto como um conhecimento que é levado em consideração durante a atualização do próprio conhecimento. De forma similar ao que ocorre no aprendizado humano, onde não se assimila indiscriminadamente o que é ensinado, o algoritmo aqui proposto avalia a informação que chega ao sistema e a confronta com tudo aquilo que já se aprendeu a respeito do problema. Se esta nova informação entrar em conflito com as convicções já formadas, ela é desconsiderada.

Essa é uma das formas com que os seres humanos assimilam o conhecimento. Na infância, quando crenças ainda não foram formadas, a criança aprende com tudo o que lhe é ensinado. Após um determinado momento, o homem passa a duvidar de qualquer coisa que lhe parece estranha se comparada as crenças que foram formadas no princípio de sua vida. Obviamente, o meio em que vivemos está em constante mudança, o que nos faz rever nossas crenças de tempos em tempos, quando vemos que o mundo começa a mudar.

O algoritmo de treinamento proposto funciona de forma análoga. Durante as primeiras iterações,

como ainda não se aprendeu o suficiente a respeito do sistema, fica-se totalmente aberto ao aprendizado. Após os primeiros instantes, quando o conhecimento é obtido, o aprendizado passa a ser seletivo, não levando em consideração informações que sejam muito conflitantes com o conhecimento atual. A incorporação de um índice de alerta no processo de treinamento faz com que o sistema se torne novamente aberto ao aprendizado quando informações conflitantes se repetem em um determinado intervalo de tempo, mostrando assim que as convicções que se tem podem não mais refletir a realidade e precisam ser adaptadas.

Resultados experimentais mostraram que redes neurais nebulosas que utilizam este novo algoritmo de treinamento se tornam robustas a dados de treinamento com valores que se referem a comportamentos anômalos ou mesmo a erros durante a obtenção destes dados. Estes sistemas se mostram capazes de perceber que certos dados não fazem parte do comportamento do que se desejava modelar e atenuam o efeito destes dados. De forma contrária, os sistemas que não utilizam do algoritmo de treinamento baseado no aprendizado participativo levam em consideração tudo o que lhes é apresentado, sofrendo influência de erros contidos nos dados, o que faz com que o desempenho degrade.

A segunda abordagem apresentada neste trabalho propõe a construção de duas novas classes de neurônios artificiais, os unineurons e os nullneurons, que incluem uma importante característica dos neurônios biológicos que não se apresenta em nenhum modelo de neurônio artificial proposto até o momento. Esta característica, conhecida como plasticidade neuronal, faz com que um neurônio possa alterar seus mecanismos internos em resposta à alterações externas e a necessidade de adaptação a condições mutantes e a estímulos repetidos.

Nos novos neurônios nebulosos, um parâmetro adicional, chamado de parâmetro de plasticidade, é adicionado ao modelo de neurônios lógicos. Este parâmetro, que é ajustado segundo a necessidade do problema a ser resolvido através do algoritmo de treinamento, faz com que o neurônio altere a forma com que os sinais de entrada e pesos sinápticos são processados, tornando o modelo composto por tais neurônios melhor adaptado à um problema específico.

Simulações mostraram que modelos de sistemas dinâmicos baseados em redes neurais nebulosas compostas por essas duas novas classes de neurônios apresentam os melhores desempenhos, tanto em termos de precisão quanto em termos de custos computacionais, quando comparados com modelos gerados a partir de redes neurais nebulosas compostas pelos neurônios nebulosos clássicos *AND* e *OR*, para os casos em que os dados de treinamento não apresentavam anormalidades ou erros.

As duas novas classes de neurônios nebulosos (unineurons e nullneurons), aliadas ao processo de treinamento baseado no aprendizado participativo, poderiam gerar modelos de sistemas dinâmicos que apresentariam uma alta performance no que diz respeito a erros de aproximação e custos de obtenção e operação, bem como uma robustez no sentido de minimizar o efeito de dados não ideais presentes nas bases de dados de treinamento.

Entretanto, para que se pudesse propor um modelo de sistema dinâmico que utilizasse uma rede neural nebulosa composta por unineurons ou nullneurons, e na qual o algoritmo de treinamento fosse baseado no aprendizado participativo seria necessário que uninormas e nullnormas pudessem ser vistas como operadores relacionais nebulosos e, conseqüentemente, pudessem ser utilizadas na síntese de novas classes de composições de relações nebulosas associadas a uma solução de problema inverso nebuloso para tais composições.

A teoria de operadores residuais fornece uma ferramenta que pode ser utilizada para este fim. De fato, operadores residuais para uma classe de uninormas já foram propostos na literatura. O estudo destes operadores e sua extensão para nullnormas, bem como a proposição de equações relacionais nebulosas que utilizem uninormas e nullnormas (e a solução de problema inverso para estas equações relacionais) constituem uma forte inspiração para futuros trabalhos relacionados aos desenvolvimentos aqui propostos.

Outra sugestão para continuidade deste trabalho é a utilização de outras estruturas de redes neurais nebulosas (como por exemplo redes que utilizam neurônios nebulosos na agregação das regras e redes com estruturas recorrentes), a fim de se verificar a aplicabilidade das técnicas aqui propostas a diferentes classes de redes neurais nebulosas. As estruturas aqui apresentadas podem, em alguns casos, apresentar problemas de complexidade para problemas de dimensões elevadas. A proposição de estruturas de redes que reduzam este risco pode ser um importante tópico para explorações futuras.

E, por fim, pode-se sugerir a incorporação de recorrências locais e/ou globais nos neurônios que compõem os modelos aqui propostos. Redes recorrentes já se mostraram superiores na modelagem de sistemas dinâmicos, se comparados a estruturas estáticas e, portanto, podem auxiliar na obtenção de um modelo ainda mais robusto e preciso do que os aqui sugeridos.

Referências Bibliográficas

- Babuska, R. & Verbruggen, H. (1996). An overview of fuzzy modeling for control. *Control Eng. Practice*, 4(11), 1593–1606.
- Baets, B. D. & Fodor, J. (1999). Residual operators of uninorms. *Soft Computing*, 3, 89–100.
- Ballini, R. & Gomide, F. (2002). Learning in recurrent, hybrid neurofuzzy networks. In *Proc. IEEE International Conference on Fuzzy Systems* (pp. 785–791). Honolulu, HI.
- Ballini, R., M. Figueiredo and, S. S., Andrade, M., & Gomide, F. (2000). A seasonal streamflow forecasting model using neurofuzzy network. *Information, Uncertainty and Fusion*, 1, 257–276.
- Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, NJ, USA.
- Björck, A. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, USA.
- Box, G. E. P. & Jenkins, G. M. (1994). *Time Series Analysis- Forecasting and Control*. Holden Day, CA, USA, 3^a edição.
- Buckley, J. J. & Hayashi, Y. (1994). Fuzzy neural networks: a survey. *Fuzzy Sets and Systems*, 66, 1–13.
- Calvo, T., Baets, B. D., & Fodor, J. (2001). The functional equations of frank and alsina for uninorms and nullnorms. *Fuzzy Sets and Systems*, 120, 385–394.
- Caminhas, W., Tavares, H., Gomide, F., & W.Pedrycz (1999). Fuzzy sets based neural networks: Structure, learning and applications. *Journal of Advanced Computational Intelligence*, 3(3), 151–157.
- Chauhan, B. K., Sharma, A., & Hanmandlu, M. (2005). Neuro-fuzzy approach based short term electric load forecasting. In *Proc. 2005 IEEE/PES Transmission and Distribution Conference* (pp. 1–5). Dalian, China.

- Cybenko, G. (1989). Approximation by superpositons of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2, 303–314.
- Delgado, M., Gómez-Skarmeta, A. F., & Martin, F. (1997). Fuzzy clustering based rapid prototyping for fuzzy rule-based modeling. *IEEE Trans. on Fuzzy Systems*, 5, 223–233.
- Dorf, R. C. & Bishop, R. H. (1998). *Modern Control Systems*. Addison-Wesley, MA, USA, 8^a edição.
- Dvoretzky, A. (1956). On stochastic approximation. In *Proc. Third Berkley Symp. Math. Stat. and Probability* (pp. 39–55). CA, USA.
- Espinosa, J., Vandewalle, J., & Wertz, V. (2004). *Fuzzy Logic, Identification and Predictive Control*. Springer-Verlag, London, UK.
- Farag, W. A., Quintana, V. H., & Lambert-Torres, G. (1998). A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems. *IEEE Trans. on Neural Networks*, 9(5), 756–767.
- Figueiredo, M., Ballini, R., Soares, S., Andrade, M., & Gomide, F. (2004). Learning algorithms for a class of neurofuzzy network and application. *IEEE Trans. on System, Man and Cybernetics - Part C*, 34(3), 293–301.
- Figueiredo, M. F. (2000). *Redes neurais nebulosas aplicadas em problemas de modelagem e controle autonomo*. PhD thesis, Universidade Estadual de Campinas - UNICAMP, Campinas-SP.
- Figueiredo, R. J. P. D. (2003). *Beyond Volterra and Wiener: Optimal Modelling of Nonlinear Dynamical Systems in a Neural Space for Applications in Computational Intelligence*, chapter 6, (pp. 65–98). IEEE Press, NY, USA.
- Figueiredo, R. J. P. D. & DwyerIII, T. A. W. (1980). A best approximation framework and implementation for simulation of large-scale nonlinear systems. *IEEE Trans. on Circuits and Systems*, CAS-27(11), 1005–1014.
- Galdi, V., Ippolito, L., Piccolo, A., & Vaccaro, A. (2000). Neural diagnostic system for transformer thermal overload protection. *IEE Proceedings of Electric Power Applications*, 147(5), 415–421.
- Giles, C. L., Sun, R., & Zurada, J. M. (1998). Neural networks and hybrid intelligent models: Foundations, theory, and applications. *IEEE Trans. on Neural Networks*, 9(5), 721–723.
- Gomide, F. & Pedrycz, W. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Wiley Interscience, NJ, USA.

- Griño, R., Cembrano, G., & Torras, C. (2000). Nonlinear system identification using additive dynamic neural networks - two on-line approaches. *IEEE Trans. on Circuits and Systems - Part I: Fundamental Theory and Applications*, 47(2), 150–165.
- Gross, G. & Galiana, F. D. (1987). Short-term load forecasting. *Proceedings of the IEEE*, 75(12), 1558–1573.
- Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice-Hall, NJ, USA, 2^a edição.
- Hell, M., Costa Jr., P., & Gomide, F. (2007a). New neurofuzzy training procedure based on participatory learning paradigm. In *Proc. 2007 IEEE International Conference on Fuzzy Systems* (pp. 1952–1957). London, UK.
- Hell, M., Costa Jr., P., & Gomide, F. (2007b). Nullneurons-based hybrid neurofuzzy network. In *Proc. of Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS '07* (pp. 331–336). San Diego-CA, USA.
- Hell, M., Costa Jr., P., & Gomide, F. (2007c). Recurrent neurofuzzy network in thermal modeling of power transformers. *IEEE Trans. on Power Delivery*, 22(2), 904–910.
- Hell, M., Costa Jr., P., & Gomide, F. (2008a). Hybrid neurofuzzy computing with nullneurons. In *Proc. of IEEE International Joint Conference on Neural Networks, IJCNN '08* (pp. 3652–3658). Hong Kong-HK, China.
- Hell, M., Costa Jr., P., & Gomide, F. (2008b). Participatory learning in power transformers thermal modeling. *IEEE Trans. on Power Delivery*, 23(4), 2058–2067.
- Jang, R. (1993). Anfis: Adaptive network based fuzzy inference system. *IEEE Trans. on Systems Man and Cybernetics*, 23(3).
- Jin, L., Ziyang, L., Jingbo, S., & Xinying, S. (2005). An efficient method for peak load forecasting. In *Proc. of 7th IEEE International Power Engineering Conference, 2005. IPEC' 05* (pp. 1–52). Niigata, Japan.
- Jin, Y., Jiang, J., & Zhu, J. (1995). Neural network based fuzzy identification and its application to modeling and control of complex systems. *IEEE Trans. on Systems, Man and Cybernetics*, 25(6), 990–997.
- Kandel, E. R., Schwartz, J. H., & Jessel, T. M. (2000). *Principles of Neural Science*. McGraw-Hill, NY, USA, 4^a edição.

- Klement, E. P., Mesiar, R., & Pap, E. (2000). *Triangular Norms*. Kluwer Publishers, Dordrecht, Netherlands.
- Kosko, B. (1991). *Neural Networks and fuzzy Systems: A Dynamical System Approach to Machine Intelligence*. Prentice-Hall, NJ, USA.
- Kushner, H. J. & Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, NY, USA, 2^a edição.
- Lima, E., Gomide, F., & Ballini, R. (2006). Participatory evolving fuzzy modeling. In *Proc. 2006 International Symposium on Evolving Fuzzy Systems* (pp. 36–41). Lake district, UK.
- Lin, C.-T. & Lee, C. S. G. (1996). *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems*. Prentice-Hall, NJ, USA.
- Ling, S., Leung, F., Lam, H., & Tam, P. (2003). Short-term electric load forecasting based on a neural fuzzy network. *IEEE Trans. on Industrial Electronics*, 50(6), 1305–1316.
- Ljung, L. (1999). *System identification: Theory for the user*. Prentice-Hall, NJ, USA, 2^a edição.
- McCulloch, W. S. & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133.
- Narendra, K. S. & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1), 4–27.
- Norgaard, M., Ravn, O., Poulsen, N. K., & Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, UK.
- Ouyang, C., Lee, W., & Lee, S. (2005). A tsk-type neurofuzzy network approach to system modeling problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 35(4), 751–767.
- Pedrycz, W. (1984). A identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems*, 13, 153–167.
- Pedrycz, W. (1993). Fuzzy neural networks and neurocomputations. *Fuzzy Sets and Systems*, 56(1), 1–28.
- Pedrycz, W. (2006). Logic-based neurocomputing with unineurons. *IEEE Trans. on Fuzzy Systems*, 14(6), 860–873.

- Pedrycz, W., Lam, P. C. F., & Rocha, A. F. (1995). Distributed fuzzy system modeling. *IEEE Trans. on Systems, Man and Cybernetics*, 25(5), 769–780.
- Pedrycz, W. & Rocha, A. (1993). Fuzzy set based models of neuron and knowledge-based networks. *IEEE Trans. on Fuzzy Systems*, 1(4), 254–266.
- Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3), 400–407.
- Rocha, A. F. (1987). *Brain as fuzzy System*, (pp. 507–507). Pergamon Press, NY, USA.
- Rocha, A. F. (1992). *Neural Nets: A Theory for Brains and Machines*. Springer-Verlag, NY, USA.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
- Ruiz-Aguilera, D. & Torrens, J. (2007). Distributivity of residual implications over conjunctive and disjunctive uninorms. *Fuzzy Sets and Systems*, 158, 23–37.
- Rutkowski, L. & Cpalka, K. (2005). Design and learning of adjustable quasi-triangular norms with applications to neuro-fuzzy systems. *IEEE Trans. on Neural Networks*, 13(1), 140–151.
- Saridis, G. N. (1970). Learning applied to successive approximation algorithms. *IEEE Trans. on Systems Science and Cybernetics*, SSC-6(2), 97–103.
- Saridis, G. N. (1974). Stochastic approximation methods for identification and control—a survey. *IEEE Trans. on Automatic Control*, AC-19(6), 798–809.
- Silva, L., Gomide, F., & Yager, R. R. (2005). Participatory learning in fuzzy clustering. In *Proc. 14th IEEE International Conference on Fuzzy Systems* (pp. 857–861). Reno, USA.
- Spall, J. (2003). *Introduction to stochastic search and optimization : estimation, simulation, and control*. Wiley-Interscience, NJ, USA.
- Stewart, G. W. (1973). *Introduction to Matrix Computations*. Academic Press, NY, USA.
- Suykens, J. A. K., Vandewalle, J. P. L., & Moor, B. D. (1996). *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Kluwer Academic Publishers, MA, USA.
- Swift, G. W. (2001). Adaptive transformer thermal overload protection. *IEEE Trans. on Power Delivery*, 16(4), 516–521.

- Tsoukalas, L. H. & Uhrig, R. E. (1997). *Fuzzy Neural Approaches in Engineering*. Jhon Wiley and Sons Inc., NY, USA.
- Villaci, D., Bontempi, G., Vaccaro, A., & Birattari, M. (2005). The role of learning methods in the dynamic assessment of power components loading capability. *IEEE Trans. on Industrial Eletronics*, 52(1), 280–290.
- Volterra, V. (1959). *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Press, NY, USA.
- Wang, L. X. (1998). Universal aproximation by hierarchical fuzzy systems. *Fuzzy Sets and Systems*, 93(2), 223–230.
- Xiao-Zhi, G. & Ovaska, S. J. (2006). Linguistic information feedforward-based dynamical fuzzy systems. *IEEE Trans. on Systems, Man and Cybernetics, Part C*, 36(4), 453–463.
- Xu, W. & Lu, Y. Z. (1987). Fuzzy model identification and self-learning for dynamic systems. *IEEE Trans. on System, Man and Cybernetics*, SMC-17, 683–689.
- Yager, R. R. (1990). A model of participatory learning. *IEEE Trans. on System, Man and Cybernetics*, 20(5), 1229–1234.
- Yager, R. R. (2001). Uninorms in fuzzy systems modeling. *Fuzzy Sets and Systems*, 122, 167–175.
- Yager, R. R. & Filev, D. P. (1993a). A fuzzy logic controller view of participatory learning. In *Proc. Second IEEE International Conference on Fuzzy Systems*, volume 2 (pp. 912–917).
- Yager, R. R. & Filev, D. P. (1993b). Modeling participatory learning as a control mechanism. *International Journal of Intelligent Systems*, 8, 431–450.
- Yager, R. R. & Rybalov, A. (1996). Uninorm aggregation operators. *Fuzzy Sets and Systems*, 80, 111–122.
- Yoshinari, Y., Pedrycz, W., & Hirota, K. (1993). Construction of fuzzy models through clustering techniques. *Fuzzy Sets and Systems*, 54, 157–165.
- Zadeh, L. A. (1965). Fuzzy sets. *Information Control*, 8(3), 338–353.