

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

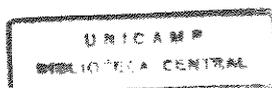
**REDES NEURAIS NEBULOSAS APLICADAS EM PROBLEMAS DE
MODELAGEM E CONTROLE AUTÔNOMO**

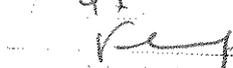
Autor: Maurício Fernandes Figueiredo *o/f*

Orientador: Fernando Antonio Campos Gomide *o/f*

o/f
Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas como requisito parcial para obtenção de Título de
Doutor em Engenharia Elétrica

Agosto, 1997



Este exemplar corresponde a redação final da tese defendida por <u>Maurício Fernandes</u> <u>Figueiredo</u> e aprovada pela Comissão de Exames. Julgada em <u>29/08/97</u>  Orientador

UNIDADE	BC
N.º CHAMADA:	F/UNICAMP
	F469r
V.	Ex.
TOMBO BC/	32102
PROC.	281/97
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	18/11/97
N.º CPD	

CM-00102282-0

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

F469r Figueiredo, Maurício Fernandes
 Redes neurais nebulosas aplicadas em problemas de modelagem e controle autônomo / Maurício Fernandes Figueiredo.--Campinas, SP: [s.n.], 1997.

Orientador: Fernando Antonio Campos Gomide.
 Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Inteligência artificial. 2. Redes neurais (Computação). 3. Automação. 4. Conjuntos nebulosos. 5. Sistemas de veículos auto-guiados. I . Gomide, Fernando Antonio Campos. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Agradecimentos

Agradeço a FAPESP (93/3034-8) e ao Projeto ESPRIT-ECLA 005 Application of Dynamic Neural Networks and Hidden Markov Models to Pattern Recognition Tasks pelos patrocínios que permitiram a realização deste trabalho.

Agradeço ao Professor Fernando Gomide pela orientação dedicada, fator essencial para que pudesse chegar ao fim desta jornada, e pelo exemplo de profissionalismo no qual sempre poderei me orientar ao trilhar meus próprios caminhos.

Não sabemos tudo. Jamais saberemos...
Que importa isto,
se a curiosidade será eterna e o prazer de saciá-la existirá para sempre?

Resumo

Neste trabalho propõem-se novas classes de redes neurais nebulosas para aplicação em modelagem e em controle autônomo. As redes neurais nebulosas aplicadas em modelagem são capazes de extrair conhecimento de dados entrada/saída e representá-lo na forma de regras nebulosas do tipo se-então, gerando modelos lingüísticos convenientes para a compreensão humana. Os parâmetros que definem as regras são determinados automaticamente durante a aprendizagem. Para efeitos de avaliação de desempenho são considerados problemas de aproximação de funções e de modelagem de sistemas dinâmicos. As redes neurais nebulosas são comparadas com diferentes abordagens presentes na literatura. A relevância das redes propostas neste trabalho é confirmada tanto no que se refere ao conjunto de suas capacidades quanto aos desempenhos observados em simulações computacionais. Entre suas principais características estão a capacidade de determinar funções de pertinência de conjuntos nebulosos e de adaptação fora de períodos restritos de aprendizagem. Uma outra classe de redes neurais nebulosas é proposta para aplicação no problema de navegação, especificamente, na composição de um sistema neural nebuloso para controle autônomo de veículos. A concepção do controlador neural é inspirada nas características dos animais e fundamentada em resultados da neurofisiologia. Ele aprende, via algoritmos genéticos, a gerar comportamentos instintivos equivalentes a busca de alvos e a desvio de obstáculos. A habilidade de coordenação destes comportamentos é adquirida enquanto o controlador interage com o ambiente ao estabelecer trajetórias para o veículo. A aprendizagem, neste caso, está fundamentada nos conceitos da teoria do condicionamento clássico. Resultados de simulação mostram que o controlador é capaz de aprender a guiar o veículo até o alvo através de trajetórias sem colisões bem como de generalizar o conhecimento adquirido em diferentes configurações de ambiente.

Abstract

In this work new classes of neural fuzzy networks are proposed for modeling and autonomous control. The neural fuzzy networks for modeling are able to extract knowledge from input/output data and to encode it explicitly in the form of if-then fuzzy rules. Therefore, linguistic models can be obtained in a form suitable for human understanding. Rule parameters are determined automatically through learning. For evaluation and analysis purposes, the proposed networks are compared with alternative approaches presented in the literature. Function approximation and dynamic system modeling problems are considered. Among the most relevant characteristics are the capacity to determine membership function profiles, and on-line adaptation. Another class of neural fuzzy networks is used to form a neural fuzzy system controller applied to autonomous navigation problem. The controller task, in this case, is to generate successful trajectories, that is, those in which the vehicle reaches targets without collision with obstacles. The general conception of the neural controller is inspired on animal capacities and relies on neurophysiology results. It learns two inborn behaviors, namely, target seeking and obstacle avoidance, through genetic algorithms techniques. To determine successful trajectories the controller also learns to coordinate the inborn behaviors. Learning of the coordination skills, is based on the classical conditioning theory. Learning occurs while the vehicle navigates and interact with the environment. Simulation results show that the neural fuzzy controller is able to guide the vehicle successfully and to generalise the control knowledge for different environments.

Lista de Símbolos

M : número de antecedentes ou dimensão do espaço de entradas da rede neural nebulosa.

N : número de regras nebulosas.

\mathbb{N} : conjunto dos números naturais.

\mathfrak{R} : conjunto dos números reais.

I_{jk} : k -ésimo intervalo de discretização relativo ao j -ésimo espaço de entradas.

Q_j : número de intervalos de discretização do espaço de entradas relativo à j -ésima dimensão.

A_j : conjunto nebuloso referente à entrada j .

a_{jk} : valor da função de pertinência do conjunto A_j correspondente ao k -ésimo intervalo de discretização.

A_j^i : conjunto nebuloso correspondente ao j -ésimo antecedente e i -ésima regra nebulosa.

a_{jk}^i : valor da função de pertinência do conjunto A_j^i correspondente ao k -ésimo intervalo de discretização.

$g^i(\cdot)$: função correspondente ao conseqüente da i -ésima regra nebulosa.

\mathbf{B}^i : vetor centro de área correspondente à i -ésima regra.

b_j^i : centro de área do conjunto nebuloso A_j^i .

$T(\cdot, \cdot)$: t-norma.

$S(\cdot, \cdot)$: s-norma.

Índice

1. Introdução	1
1.1 Motivação e Relevância.....	2
1.2 Contexto	3
1.3 Objetivos e Resultados	4
1.4 Organização do Trabalho.....	6
2. Fundamentos Teóricos	7
2.1 Introdução.....	8
2.2 Histórico	8
2.3 Redes Neurais: Conceitos Fundamentais, Modelos Básicos e Pesquisa Recente.....	9
2.4 Redes Neurais dos Tipos Kohonen e Contrapropagação	22
2.5 Conjuntos Nebulosos e Sistemas Nebulosos: Introdução.....	30
2.6 Redes Neurais Nebulosas.....	38
2.7 Resumo	43
3. Redes Neurais Nebulosas Aplicadas em Modelagem de Sistemas	45
3.1 Introdução.....	46
3.2 Rede Neural Nebulosa N1	48
3.2.1 Arquitetura	48
3.2.2 Aprendizagem.....	52
3.2.3 Aproximação	57
3.2.4 Convergência.....	62
3.2.5 Resultados de Simulação.....	66
3.3 Rede Neural Nebulosa N2	77
3.3.1 Arquitetura	77
3.3.2 Aprendizagem.....	78
3.3.3 Aproximação	87
3.3.4 Convergência.....	87
3.3.5 Resultados de Simulação.....	88
3.4 Resumo	92
4. Sistema Nebuloso Aplicado em Controle Autônomo.....	94
4.1 Introdução.....	95
4.2 Navegação Autônoma: Características do Veículo e Controlador.....	97
4.3 Repertórios CA e TS	101
4.3.1 Arquiteturas das Redes Neurais Nebulosas.....	101
4.3.2 Algoritmos Genéticos e Emergência de Comportamentos Instintivos.....	102
4.4 Repertório de Coordenação e Saída do Controlador	105
4.5 Resultados de Simulação	112
4.6 Resumo	116
5. Conclusões.....	117
5.1 Introdução.....	118
5.2 Contribuições	118
5.3 Trabalhos Futuros.....	121
Referências Bibliográficas.....	123

CAPÍTULO 1

INTRODUÇÃO

1. Introdução

1.1 Motivação e Relevância

A todo momento o progresso exigido pela sociedade põe à prova o conhecimento tecnológico e científico disponível confrontando-o com problemas de complexidade crescente. Novas soluções devem ser propostas de modo a satisfazer critérios mais rígidos de desempenho. Em muitos casos nota-se que as soluções encontradas pela teoria clássica vigente não apresentam a eficiência desejada ou são por demais custosas. Dificuldades surgem devido a diversos motivos, entre eles: necessidade de considerar sistemas mal definidos, restrições de tempo, presença de ruídos, perturbações ou alterações na dinâmica dos sistemas. Observando as soluções encontradas pela natureza para superar tais dificuldades, pesquisadores têm se inspirado e desenvolvido teorias alternativas pertencentes à denominada área da inteligência computacional. Dentre elas estão a teoria dos sistemas nebulosos [Kosko92], das redes neurais [Anderson95], dos algoritmos genéticos [Goldberg89]. Cada qual com suas características específicas, elas têm se mostrado eficientes quando os problemas exigem soluções robustas, tolerantes a falhas, de dinâmica adaptativa, capazes de considerar dados de natureza vaga ou imprecisa. Os resultados alcançados por estas abordagens têm atraído a atenção de um número cada vez maior de novos pesquisadores interessados no seu estudo. Além disso, a relativa facilidade com que suas soluções podem ser implementadas permitem sua difusão pelos mais diversos segmentos de atividades humanas, sejam elas científicas ou tecnológicas.

Redes neurais artificiais são sistemas de computação paralela e distribuída inspiradas nos seus equivalentes biológicos. Elas gozam de um conjunto amplo de características essenciais para solução de problemas complexos. A capacidade de aprendizagem talvez seja a característica que mais as diferenciem de outras abordagens. Pode-se dizer, de forma muito geral, que a aprendizagem garante a aquisição de conhecimento. Esta capacidade permite que as redes neurais adaptem-se às possíveis alterações na dinâmica ou dos parâmetros do problema, extraíam relações presentes em um conjunto de dados, incorporem dinâmicas externas, desenvolvam soluções próprias frente a situações desconhecidas. Sua versatilidade muito contribui para que sejam pesquisadas diferentes possibilidades de aplicação. Entre outros exemplos, são empregadas em: controladores de processos em aciarias, indústrias de cimento e químicas; controladores de robôs e veículos auto guiados em indústrias de manufatura; sistemas de identificação; processamento de imagens; classificação e reconhecimento de padrões.

Técnicas de inteligência computacional têm sido integradas às redes neurais visando ampliar ainda mais o conjunto de suas características individuais [Zurada94]. Quando integradas à teoria de conjuntos nebulosos dão origem às redes neurais nebulosas. Estes sistemas tornam-se dotados de características provenientes de ambas as técnicas somando vantagens e preenchendo falhas individuais. As redes neurais nebulosas herdam de um lado as capacidades de tratamento de expressões lingüísticas muito desenvolvidas nos sistemas nebulosos. Alguns modelos permitem dois procedimentos em geral infactíveis em redes neurais convencionais: inserção e extração de conhecimento, ou seja, é possível inserir conhecimento na estrutura da rede ou dela extraí-lo. Por outro lado, estes sistemas herdam todas as habilidades exibidas pelas redes neurais referentes a aprendizagem.

Conseqüentemente, além de aprenderem, elas provêem condições para que o conhecimento adquirido possa ser compreensivelmente extraído na forma lingüística.

As expectativas em torno das redes neurais nebulosas surgem não só por causa de suas referências, de um lado as redes neurais e de outro os sistemas nebulosos, ambas técnicas de reconhecido sucesso científico e tecnológico, mas também devido a seus potenciais. Diante do panorama de expectativas que se estabelece, pesquisadores têm se dedicado ao estudo de redes neurais nebulosas para aplicação em propósitos diversos. Os resultados promissores alcançados e as inúmeras questões ainda não investigadas justificam o investimento na pesquisa e no desenvolvimento destes modelos computacionais a fim de torná-los capazes de solucionar problemas e suplantiar os desafios impostos pela sociedade.

1.2 Contexto

A solução do problema de modelagem quando abordado através de sistemas nebulosos requer, na maioria das vezes, a construção de uma base de regras nebulosas. Diversos elementos devem ser especificados na realização desta tarefa, tais como: número de regras nebulosas, funções de pertinência, interpretação operacional de regras e operadores e agregação. Quanto mais complexa a aplicação envolvida maiores serão os esforços necessários para se definir os elementos adequadamente, ou seja, aqueles que correspondam a um modelo nebuloso que satisfaça aos índices de desempenho preestabelecidos. São poucos os procedimentos sistemáticos conhecidos para construção de uma base de regras nebulosa, principalmente quando a única informação disponível provém de um conjunto de dados numéricos. Procedimentos tipo tentativa-e-erro são freqüentemente as únicas opções disponíveis.

Para se evitar abordagens baseadas exclusivamente em experimentação, vários métodos alternativos de modelagem nebulosa (que resulte em um modelo nebuloso) têm sido propostos na literatura, e.g., os métodos de modelagem baseadas em agrupamento [Pedrycz93]. Estes, entretanto, apresentam alguns inconvenientes: as regras nebulosas geradas são de difícil associação semântica, não permitem descrevê-las na forma de tabelas lingüísticas e não garantem a cobertura do espaço de entradas. Outros métodos necessitam, em geral, que o projetista faça intervenções em diversas fases da modelagem.

Redes neurais nebulosas têm se apresentado como uma poderosa abordagem para construção automática de modelos nebulosos. Estes sistemas computacionais são capazes de especificar parâmetros do modelo nebuloso e codificar explicitamente a base de regras nebulosas em sua estrutura. Ou seja, o conhecimento adquirido pela rede pode ser facilmente extraído da rede na forma de regras nebulosas do tipo se-então. Além disso, diferentemente das abordagens clássicas, as redes neurais nebulosas prescindem do conhecimento físico do sistema a ser modelado. Elas são capazes de gerar o modelo a partir de dados de entrada/saída e, conseqüentemente, de eliminar deficiências causadas por variações de parâmetros. Buscando desenvolver redes neurais nebulosas cada vez mais eficientes para aplicação em modelagem nebulosa, arquiteturas e métodos de aprendizagem vêm sendo propostas. Neste contexto está inserida a primeira parte deste trabalho na qual descreve-se as propostas de duas redes neurais nebulosas (arquiteturas e

métodos de aprendizagem), discute-se suas características e são apresentados resultados de simulações e avaliações de desempenho.

A segunda parte do presente trabalho está relacionada ao problema de controle autônomo, especificamente, à navegação de veículos autoguiados. Este problema tem despertado muito interesse na comunidade científica tanto pela riqueza de novas possibilidades de investigação, principalmente na área de controle inteligente, quanto pela evidente importância associada às inúmeras aplicações industriais. Apesar dos esforços dedicados à sua solução muitas questões básicas ainda continuam em aberto.

A solução, em sua forma mais usual, requer que o veículo seja guiado até um alvo previamente especificado através de uma trajetória livre de colisões qualquer que seja o ambiente de navegação. A maior dificuldade no projeto de controladores autônomos para este fim reside nas infinitas variações topológicas possíveis do ambiente a ser navegado, ou seja, torna-se praticamente impossível considerar no projeto todas as situações que um ambiente totalmente geral pode produzir durante o deslocamento do veículo. Esta dificuldade é particularmente crítica para abordagens baseadas em inteligência artificial, nas quais os controladores devem possuir uma representação simbólica do ambiente em sua forma mais geral. Se esta representação não for completa a situação real do veículo pode eventualmente não corresponder ao modelo do ambiente contido no controlador provocando uma tomada de decisão por indefinição (decisão por “default”) inconveniente.

Controladores capazes de aprender e adaptar-se têm sido adotados por serem em princípio capazes de superar tais dificuldades. A cada vez que estes controladores deparam-se com uma situação desconhecida eles podem adquirir e acumular novos conhecimentos visando construir e completar seu modelo interno de ambiente. O principal foco da segunda parte deste trabalho está inserido na pesquisa de controladores autônomos adaptativos, especificamente no desenvolvimento de redes neurais nebulosas para controle autônomo de veículos autoguiados.

1.3 Objetivos e Resultados

Este trabalho tem como objetivo geral introduzir novas arquiteturas e métodos de aprendizagem para redes neurais nebulosas aplicadas em problemas de modelagem e de controle autônomo.

Duas classes de redes neurais nebulosas (denominadas N1 e N2 respectivamente) são apresentadas para aplicação em problemas de modelagem nebulosa. Tanto N1 quanto N2 são baseadas no modelo proposto em [Figueiredo95^b]. Em ambas propostas o conhecimento da rede é codificado compreensivelmente em sua estrutura na forma de regras nebulosas do tipo se-então. Esta característica traz dois benefícios imediatos: de um lado é possível inserir conhecimento inicial na rede neural nebulosa visando acelerar o processo de aprendizagem; por outro, o conhecimento adquirido durante a fase de aprendizagem, disponível na forma de regras nebulosas, pode ser utilizado tanto no auxílio à projetos de sistemas nebulosos quanto no estudo das propriedades dos sistemas modelados.

A rede neural nebulosa N1 aprende em duas fases [Figueiredo97^b]. Na primeira fase N1 se auto-organiza determinando regras e funções de pertinência de conjuntos nebulosos. Na fase seguinte ela aprende os conseqüentes das regras utilizando uma abordagem supervisionada. As características mais importantes de N1 são: 1) fácil inserção e extração de conhecimento na forma de regras nebulosas; 2) o processamento neural emula o mecanismo de raciocínio nebuloso; 3) as funções de pertinência de cada conjunto nebuloso são determinadas a partir dos dados de treinamento sem que nenhuma informação predefinida seja necessária; 4) regras são geradas automaticamente sem qualquer procedimento adicional; 5) a partição do espaço de entradas (observada em simulação) garante que as regras cubram todo o domínio de operação do modelo e 6) N1 é um aproximador universal. Tais características possibilitam N1 especificar os principais parâmetros de projeto de sistemas nebulosos. Resultados de simulação e comparações com outras abordagens mostram que N1 atinge bons índices de desempenho, mesmo gerando um número reduzido de regras (compromisso desejável entre complexidade e precisão), superando aqueles obtidos por diferentes abordagens.

N2 compartilha várias características de N1 enumeradas anteriormente, tais como: 1, 2, 4, 5 e 6 [Figueiredo97^a]. Sua aprendizagem não limita-se a um período de treinamento. N2 aprende enquanto e sempre que seu desempenho não atingir o índice especificado. A aquisição de conhecimento implica na alteração de sua estrutura correspondendo à inserção de regras nebulosas e ajuste das funções de pertinência. Portanto, o método de aprendizagem de N2, do tipo construtivo [Ash89], capacita a rede compensar degradações de desempenho provocadas por variações nos parâmetros do modelo, ruídos ou quaisquer outras perturbações, além de especificar parâmetros importantes no auxílio de projeto de sistemas nebulosos. O desempenho de N2, avaliado por meio de simulações e comparações com outras abordagens, qualificam-na como uma ferramenta importante na abordagem de problemas de modelagem.

No contexto de controle autônomo este trabalho propõe um sistema neural nebuloso e uma rede neural nebulosa para navegação autônoma de veículos em ambientes bidimensionais [Figueiredo96]. Três módulos constituem o sistema. Dois deles são responsáveis por gerar comportamentos básicos. O primeiro direciona o veículo para seu alvo e o segundo afasta-o de obstáculos. As respostas de cada módulo são resultantes dos sinais externos captados por sensores de distância a obstáculos e de direção a alvo instalados no veículo. Devido à natureza dos comportamentos que produzem, estas respostas freqüentemente entram em conflito de propósitos. O terceiro módulo é constituído por uma rede neural nebulosa e por um neurônio nebuloso. Sua tarefa é ponderar as respostas dos outros dois módulos fazendo com que atuem na direção do veículo de forma cooperativa. Ou seja, após um período de aprendizagem, este módulo gera comportamentos distintos dos comportamentos básicos por meio dos quais o veículo alcança seu alvo perfazendo uma trajetória livre de colisões contra obstáculos. Ela aprende sua tarefa à medida que exerce o controle sobre o veículo e sem considerar qualquer informação sobre a topologia do ambiente. Inicialmente suas decisões são inadequadas, eventualmente acontecendo algumas colisões, quando então a rede neural nebulosa altera sua estrutura de forma a acomodar novos conhecimentos além de ajustar aqueles já adquiridos. A aprendizagem se encerra quando não se verifica qualquer colisão durante o curso da trajetória. Simulações mostram que a rede neural nebulosa aprende os conhecimentos fundamentais de navegação a partir do controle do veículo em um ambiente particular e generaliza-o com sucesso para ambientes distintos (diferentes

disposições de obstáculos e alvo). Portanto, ela é capaz de guiar o veículo em ambientes diferentes daquele navegado durante a aprendizagem, dirigindo-o até o alvo numa trajetória sem colisões.

1.4 Organização do Trabalho

Este primeiro capítulo apresentou o contexto dentro do qual o trabalho foi desenvolvido, seus aspectos relevantes, o resumo dos objetivos e principais resultados alcançados. O capítulo 2 provê alguns fundamentos teóricos visando facilitar a explanação do trabalho realizado e a compreensão do texto. Além disso, apresenta uma compilação de resultados recentes na área de redes neurais nebulosas proporcionando uma visão geral das principais tendências da pesquisa. Duas classes de redes neurais nebulosas aplicadas em modelagem são propostas no capítulo 3. Descreve-se suas estruturas e métodos de aprendizagem. Propriedades referentes à convergência são discutidas e a capacidade de aproximação funcional é demonstrada através de teorema. Resultados de simulação e análises comparativas com outras propostas presentes na literatura também são relatados. No capítulo seguinte o problema de navegação autônoma é focalizado. Descreve-se o sistema neural nebuloso e a rede neural nebulosa propostos para controle do veículo autônomo. As potencialidades do sistema neural nebuloso são demonstradas simulando-o em diferentes ambientes de navegação. Os resultados obtidos são analisados. O capítulo 5 conclui o trabalho fazendo uma retrospectiva de suas contribuições e apresentando sugestões para pesquisas futuras.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

2. Fundamentos Teóricos

2.1 Introdução

O atual estágio de desenvolvimento da sociedade exhibe uma necessidade premente de reproduzir em dispositivos artificiais as capacidades inteligentes de que são dotados os seres humanos. Dispositivos artificiais com tais potencialidades seriam capazes de resolver problemas de grande complexidade associados a, por exemplo, identificação de padrões, controle adaptativo, processamento de sinais, e cujas soluções são de imediata e alta importância tanto no meio científico quanto no tecnológico.

A teoria de redes neurais apresenta-se como uma das abordagens através das quais desenvolvem-se tais dispositivos. Grande esforço tem sido dedicado neste sentido. Pesquisadores são continuamente incentivados tanto pelo bom desempenho alcançado pelas redes neurais em diversas aplicações práticas quanto pela grande receptividade destes modelos computacionais no meio industrial, decorrente da facilidade e flexibilidade com que elas podem ser implementadas em situações reais. A consequência imediata deste empenho na pesquisa reflete-se em rápido progresso teórico. Uma das tendências mais recentes é fundir a teoria das redes neurais com outras diferentes teorias da inteligência computacional. Por exemplo, algoritmos genéticos têm sido empregados como métodos de aprendizagem. Já as redes neurais nebulosas resultam da integração da teoria de redes neurais com a teoria de conjuntos nebulosos. Estes modelos computacionais reúnem benefícios de ambas as teorias além de eliminarem algumas das deficiências das redes neurais tradicionais.

Este capítulo apresenta os fundamentos necessários para o entendimento das redes neurais nebulosas propostas na seqüência deste trabalho. Alguns de seus objetivos principais são: recapitular brevemente a história da pesquisa em redes neurais desde seus primórdios, discutir sua relevância e principais motivações, apresentar conceitos e características fundamentais das redes neurais e descrever os modelos básicos mais frequentemente empregados. As redes do tipo Kohonen [Kohonen90] e do tipo contrapropagação [Hetch-Nielsen89] são focalizadas em maior profundidade por serem de particular interesse para o entendimento dos capítulos seguintes. Visando apresentar o estado recente da pesquisa no contexto das redes neurais nebulosas, algumas das propostas, aplicações e implementações mais recentes são sucintamente discutidas.

2.2 Histórico

W. McCulloch e W. Pitts estabeleceram o marco da criação da teoria das redes neurais artificiais quando propuseram em 1943 o primeiro modelo matemático para um neurônio biológico [McCulloch43]. As potencialidades computacionais deste modelo foram reconhecidas rapidamente. Entretanto, faltavam-lhe características adaptativas e de aprendizagem, imprescindíveis em modelos de sistemas inteligentes. O primeiro método de aprendizagem para redes neurais foi concebido em 1949 pelo neurofisiologista D. Hebb [Hebb49]. Este método permaneceu por muito tempo limitado ao campo da psicologia e pouco favoreceu a pesquisa em redes neurais contemporânea a esta época. Ao final dos anos 50, F. Rosenblatt propôs o modelo perceptron e demonstrou o "Teorema da

Convergência para Perceptron" formalizando a teoria relacionada a aprendizagem de redes neurais [Rosenblatt58], [Rosenblatt62]. Este modelo obteve expressivo sucesso até que M. Minsky e S. Papert demonstraram matematicamente as sérias limitações computacionais das redes perceptron de uma única camada [Minsky69].

O trabalho de M. Minsky e S. Papert estabeleceu o fim de um período de grande empolgação no desenvolvimento e investigação de redes neurais e cuja principal motivação dos pesquisadores tinha origem nas enormes expectativas do que um sistema dotado de capacidades inteligentes poderia proporcionar. A falta de um método de aprendizagem para aplicação em redes mais complexas e de maior potencial computacional desestimularam os pesquisadores, fazendo com que as redes neurais permanecessem esquecidas durante a década de 70.

Somente a partir da divulgação do trabalho de Hopfield no início da década seguinte, as redes neurais reassumiram seu papel como importante área de pesquisa [Hopfield85] [Hopfield82]. O modelo dinâmico e estável proposto por Hopfield demonstrou suas potencialidades quando aplicado na solução de problemas nos quais a teoria tradicional se mostrava ineficiente, e.g. problemas relacionados a otimização e reconhecimento de padrões. O interesse pelas redes neurais foi reavivado por completo com a divulgação do método de aprendizagem do tipo gradiente para as redes multicamadas [Rumelhart86], [Widrow90]. Desde então as redes neurais são intensamente pesquisadas e empregadas nos mais variados tipos de aplicações [Lippmann87], [Hush93],[Widrow90].

Atualmente a motivação dos pesquisadores não é mantida apenas pelas expectativas de resultados futuristas tal como acontecia em meados do século com a criação dos primeiros modelos de neurônios artificiais. Os promissores resultados obtidos pelas redes neurais nos mais diversos segmentos de atividades humanas têm contribuído muito para que o trabalho de investigação e desenvolvimento de novos modelos se intensifique. Uma das tendências recentes é consorciar a teoria de redes neurais com diferentes teorias da inteligência computacional, por exemplo: algoritmos genéticos [Ichikawa92], e teoria do caos [Dingle93]. Particular sucesso tem sido obtido com a integração das redes neurais com a teoria de conjuntos nebulosos resultando nas redes neurais nebulosas discutidas nas próximas seções [Gomide92].

2.3 Redes Neurais: Conceitos Fundamentais, Modelos Básicos e Pesquisa Recente

As redes neurais surgiram do esforço de reproduzir em dispositivos artificiais as capacidades inteligentes de que o homem é dotado, proporcionadas pelo seu cérebro. Elas são modelos bastante simplificados do sistema nervoso biológico.

O cérebro humano é constituído por diversos tipos células. Os neurônios são aquelas mais diretamente responsáveis pela reação aos estímulos externos, recebendo-os e processando-os, gerando respostas adequadas a estes estímulos e transmitindo estas respostas a todo o organismo. Estima-se que o número de neurônios no cérebro seja da ordem de 10^{11} . Os neurônios recebem sinais de entrada através dos dendritos, processa-os no corpo celular ou soma e gera sinais de saída que são transmitidos para outros neurônios por meio do axônio e suas ramificações (Figura 2.1). A região onde o axônio de um neurônio encontra o dendrito de outro neurônio, denominada sinapse, é de extrema importância não só para a

transmissão de sinais entre células, mas também no que diz respeito às características apresentadas pelo cérebro como um todo (Figura 2.2). De um lado as sinapses podem alterar os sinais provenientes do axônio ou mesmo impedi-los que cheguem aos dendritos. De outro as sinapses podem sofrer modificações devido à passagem de sinais ou a atividades dos neurônios, fatos que possivelmente estão relacionados à capacidade de memória e aprendizagem [Haykin94].

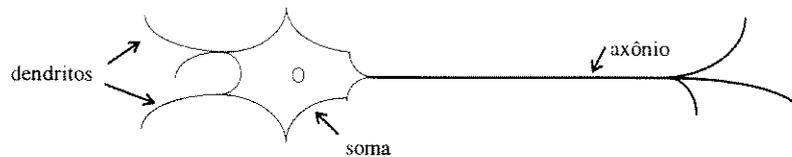


Figura 2.1: Modelo de neurônio biológico.

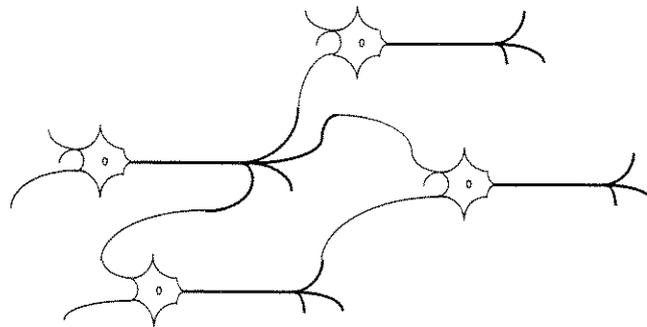


Figura 2.2: Modelo de rede neural biológica.

Pesquisadores desenvolveram as redes neurais inspirando-se no seu paralelo biológico e convencidos de que modelos simplificados herdariam, mesmo que parcialmente, algumas de suas principais características. Por este motivo as diversas propostas de definição para redes neurais se atêm às características básicas dos sistemas nervosos biológicos, particularmente ao cérebro humano. Baseando-se nos aspectos comuns presentes na maioria das definições, pode-se dizer que as redes neurais são modelos computacionais constituídos por dois componentes básicos: uma arquitetura computacional paralela e distribuída (também denominada estrutura da rede neural) e um método de aprendizagem. A arquitetura da rede neural é formada por um grande número de unidades processadoras, denominadas neurônios, que operam em paralelo e que apresentam características adaptativas. Os neurônios conectam-se entre si por meio de ligações sinápticas (ou simplesmente sinapses). O papel das sinapses relacionado à aprendizagem e à memória é representado por parâmetros numéricos denominados pesos sinápticos. O método de aprendizagem determina os pesos sinápticos adequados às exigências do problema.

As redes neurais exibem um conjunto de características e capacidades qualitativamente semelhantes aos sistemas nervosos biológicos. Se analisadas sob o ponto de vista computacional são muitas vezes bastante distintas dos computadores do tipo von Neumann (Tabela 2.1) [Jain96]. Elas são processadores do tipo paralelo e distribuído. Um grande número de neurônios operam simultaneamente e dividem suas atividades. Cada neurônio é computacionalmente simples, caso considere-se as potencialidades da rede como um todo. A memória é distribuída por todas as sinapses, está integrada ao processador e é acessada pelo seu conteúdo.

Tabela 2.1: Comparação entre processadores von Neumann e redes neurais.

Critério	Processadores von Neumann	Redes Neurais
Processador	Poucos Complexos	Grande número Simples
Memória	Separada do processador Localizada Acessada por endereço	Integrada ao processador Distribuída Acessada por conteúdo
Confiabilidade	Vulnerável	Robusta
Processamento	Centralizado Seqüencial Dependente de programação	Distribuído Paralelo Capaz de aprendizagem
Solução	Deduzida	Induzida

As redes neurais apresentam grande redundância, ou seja, neurônios semelhantes ou idênticos se repetem em diversas regiões de sua estrutura. Juntas, as características de redundância e representação distribuída do conhecimento tornam as redes neurais inerentemente robustas a perturbações e tolerantes a falhas. Deficiências em partes específicas do processador são compensadas pelo restante de sua estrutura, impedindo que o desempenho final seja criticamente degradado.

Talvez a aprendizagem seja a mais importante das propriedades exibidas pelas redes neurais. Elas são capazes de adquirir conhecimento necessário para solucionar os problemas inerentes às suas tarefas. Enquanto interagem com o ambiente, as redes neurais extraem o conhecimento contido nos estímulos que recebe. A aprendizagem das redes neurais é predominantemente indutiva e, por conseqüência, elas são capazes de generalização e reprodução do raciocínio aproximado. Dotadas da capacidade de aprendizagem, as redes neurais deixaram de ser estudadas apenas pela expectativa de virem a ser processadores numéricos utilizados para simulação de modelos e programação de algoritmos (como são os computadores modernos), mas também para tornarem-se uma teoria sedimentada e de reconhecida eficiência na solução de problemas de alta complexidade.

A literatura apresenta uma numerosa diversidade de modelos de redes neurais. Pela sua simplicidade e por serem as mais difundidas, as redes multicamadas serão consideradas em seguida a título de ilustração dos aspectos fundamentais da teoria. O modelo de neurônio que constitui a estrutura desta rede é tal qual apresentado na Figura 2.3. Os sinais de entrada x_i , $i = 1, \dots, N$, ao chegarem ao neurônio são ponderados pelos pesos sinápticos w_i , $i = 1, \dots, N$. Estes são, então, enviados ao soma através dos dendritos onde são agregados, ou mais especificamente, adicionados. Em seguida uma transformação não linear $F^*(.)$ (função de ativação do neurônio) age sobre o sinal resultante da agregação gerando o sinal de saída y (2.1):

$$y = F^* \left(\sum_{i=1}^N x_i w_i \right) \quad (2.1)$$

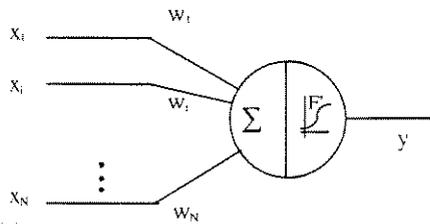


Figura 2.3: Modelo básico de neurônio artificial.

Uma das características necessárias à transformação não linear $F^*(.)$ consiste em estabelecer, para o sinal resultante da agregação, um valor de limiar abaixo do qual a saída do neurônio é fortemente reduzida, atingindo valores próximos de zero ou até mesmo nulos. Somente no caso do sinal resultante ser maior que o valor de limiar o neurônio é capaz de gerar uma saída próxima do seu valor máximo (neste caso diz-se que o neurônio dispara ou está ativado). Esta característica não linear é de fundamental importância para a aprendizagem deste tipo de rede neural. Por esta razão o efeito de limiar é equivalentemente interpretado como aquele causado por uma entrada constante $x_0 = -1$ ponderada por um peso sináptico w_0 . Portanto, a partir da equação (2.1), resulta (2.2):

$$y = F\left(\sum_{i=1}^N x_i w_i - x_0 w_0\right) \text{ ou, equivalentemente,}$$

$$y = F\left(\sum_{i=0}^N x_i w_i\right) \quad (2.2)$$

em que $F(.)$ é tipicamente qualquer função do tipo sigmóide, por exemplo: $F(x) = 1/(1 + \exp(-\beta x))$ (Figura 2.4).

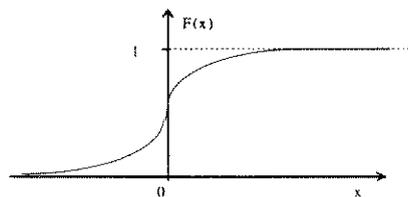


Figura 2.4: Função sigmóide.

Os neurônios da rede neural multicamadas são conectados de tal maneira que sua arquitetura apresenta, sob o ponto de vista do fluxo de sinais, uma formação em camadas (Figura 2.5). A primeira camada é a camada de entrada. Ela é constituída de neurônios (representados por círculos pequenos) cuja única função é receber os sinais provenientes do ambiente e transmiti-los para a rede neural propriamente dita. Portanto, estes neurônios são totalmente diferentes dos neurônios básicos (Figura 2.3) que compõem a arquitetura da rede a partir da segunda camada. Os neurônios pertencentes à segunda camada recebem somente os sinais provenientes da primeira camada. Os neurônios da terceira camada recebem somente sinais de saída provenientes dos neurônios da segunda camada e assim por diante até a última camada, cujos neurônios geram as saídas ou respostas da rede neural. Na Figura 2.5 p_j^k é o j -ésimo neurônio da camada k , $k = 2, \dots, K$, e w_{ij}^k é o peso sináptico correspondente à sua conexão com o neurônio i da camada anterior. O número

de neurônios da camada k é N^k . Os sinais de entrada (entradas) e de saída (saídas) da rede neural são $x_i, i = 1, \dots, N$, e $y_j, j = 1, \dots, N^k$, respectivamente.

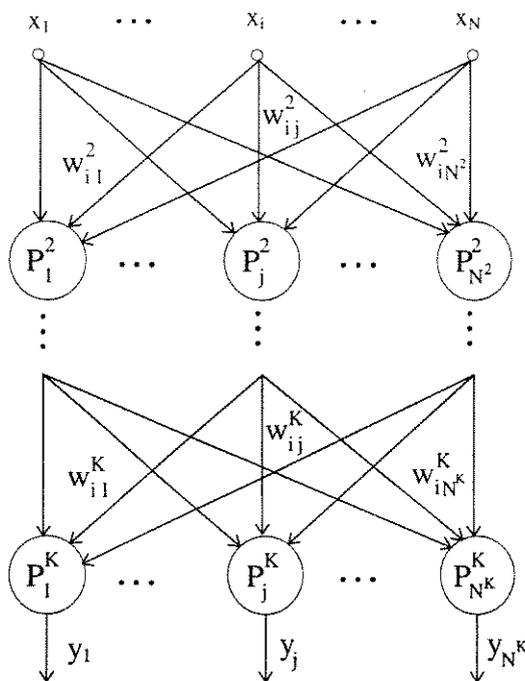


Figura 2.5: Estrutura da rede multicamadas.

Somente com sua estrutura a rede neural já processa os sinais de entrada. No entanto para que ela também seja capaz de aprender é necessário se definir um método de aprendizagem através do qual os pesos sinápticos sejam convenientemente determinados.

O método fundamental de aprendizagem para as redes multicamadas é o método do tipo gradiente [Rumelhart86]. Segundo este método a rede neural extrai seu conhecimento de um conjunto de exemplos (pares de vetores) Γ disponível durante a aprendizagem, em que $\Gamma = \{(\mathbf{u}^1, \mathbf{v}^1), \dots, (\mathbf{u}^r, \mathbf{v}^r), \dots, (\mathbf{u}^R, \mathbf{v}^R)\}$, $\mathbf{u}^r = (u_1^r \dots u_i^r \dots u_N^r)'$ e $\mathbf{v}^r = (v_1^r \dots v_i^r \dots v_{N^K}^r)'$. O vetor \mathbf{v}^r (saída desejada) representa o sinal que a rede neural deve aprender a reproduzir em suas saídas quando o vetor \mathbf{u}^r for considerado como entrada. O método consiste em apresentar os vetores \mathbf{u}^r como entradas para a rede neural, comparar as respectivas saídas $\mathbf{y}^r = (y_1^r \dots y_i^r \dots y_{N^K}^r)'$ com os vetores \mathbf{v}^r e ajustar os pesos sinápticos de maneira que as diferenças encontradas sejam reduzidas ou, equivalentemente, de forma que o desempenho da rede neural torne-se mais próximo do desejado.

Seja \mathbf{w} o vetor de pesos sinápticos da rede neural. A função $E(\mathbf{w})$ definida em (2.3) é usualmente utilizada para avaliar o desempenho da rede multicamadas. $E(\mathbf{w})$ é proporcional ao erro quadrático total apresentado pelas respostas da rede neural \mathbf{y}^r considerando os pares $(\mathbf{u}^r, \mathbf{v}^r)$:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{r=1}^R \|\mathbf{v}^r - \mathbf{y}^r\|^2 \quad (2.3)$$

A aprendizagem pode ser vista como um problema de otimização. Ou seja, dado um conjunto de sinais de entrada da rede neural \mathbf{u}^r , os respectivos sinais desejados como saída \mathbf{v}^r e uma função para avaliação de desempenho $E(\mathbf{w})$, o problema se resume em determinar um conjunto de pesos sinápticos w_{ij}^k tal que a rede neural satisfaça o índice de desempenho estabelecido. No caso da função $E(\mathbf{w})$ definida acima, este problema é definido tal como em (2.4):

$$\underset{w_{ij}^k}{\text{Min}} E(\mathbf{w}) \quad (2.4)$$

O método de aprendizagem do tipo gradiente surge como uma consequência natural para a solução do problema (2.4). O método do gradiente determina um processo iterativo para ajuste de \mathbf{w} visando solucionar o problema de otimização definido em (2.4). A cada iteração t , $\mathbf{w}(t)$ é ajustado de forma a encontrar a solução ótima global do problema. O vetor de pesos sinápticos inicial $\mathbf{w}(0)$ é escolhido aleatoriamente. O método do gradiente determina cada componente $w_{ij}^k(t + 1)$ de acordo com a equação (2.5):

$$w_{ij}^k(t + 1) = w_{ij}^k(t) - \mu \frac{\partial E}{\partial w_{ij}^k} \quad (2.5)$$

em que μ é o fator de aprendizagem.

A derivada parcial $\partial E / \partial w_{ij}^k$ é calculada explicitando-se a função $E(\mathbf{w}(t))$ e aplicando-se sucessivamente a regra da cadeia. Para que o ajuste seja tal que $E(\mathbf{w}(t+1)) \leq E(\mathbf{w}(t))$, o fator μ deve ser escolhido convenientemente, ou seja, $\mu = \mu(t)$.

Devido às características da função sigmóide e à estrutura da rede, as derivadas parciais com relação aos pesos sinápticos de uma camada podem ser formuladas a partir de derivadas parciais com relação à camada posterior. Por esta razão a expressão associada à derivada em (2.5) torna-se simples. O cálculo de ajustes é feito a partir das camadas posteriores, caracterizando um processo inverso ao processamento dos sinais de entrada pela rede neural (o termo em língua inglesa utilizado para denominar este processo de ajuste é *backpropagation* nome pelo qual tanto a rede quanto o método tornaram-se popularmente conhecidos). O algoritmo de ajuste dos pesos sinápticos correspondente ao método de aprendizagem do tipo gradiente é tal como segue:

1. Inicialização: Inicialize o vetor de pesos sinápticos \mathbf{w} de maneira aleatória.
2. Processamento: Determine o valor de $E(\mathbf{w}(t))$ computando os erros apresentados pela rede para cada exemplo $(\mathbf{u}^r, \mathbf{v}^r)$ do conjunto Γ .
3. Ajuste: Ajuste os pesos sinápticos de acordo com a expressão (2.5).
4. Avaliação da Aprendizagem: Caso a rede neural tenha alcançado o índice de desempenho estabelecido ou o número de ajustes tenha ultrapassado um valor máximo especificado, vá para o passo 5. Caso contrário retorne ao passo 2.
5. Fim da aprendizagem.

As redes neurais multicamadas tais que $K \geq 2$ são capazes de aproximar, com precisão arbitrária, qualquer função real desde que seus pesos sinápticos sejam convenientemente especificados (o número de descontinuidades apresentadas pela função deve ser finito e seu conjunto imagem deve estar contido no conjunto imagem da rede neural) [Hornik89], [Cybenko89], [Funahashi89]. Devido a esta propriedade, esta rede neural é considerada um aproximador universal. No entanto o método de aprendizagem nem sempre encontra a solução ótima global para o problema definido em (2.4). Em geral o processo de aprendizagem estabelecido pelo correspondente algoritmo encerra-se apontando soluções \mathbf{w}^* próximas a um mínimo local da função $E(\mathbf{w})$. Neste caso $\nabla E(\mathbf{w}^*) \equiv 0$ e $\mathbf{w}(t) \equiv \mathbf{w}(t + 1) = \mathbf{w}^*$. Alguns propostas têm sido apresentadas visando reduzir os problemas de mínimo local [Chang92].

As redes multicamadas são extensivamente estudadas. Descrições mais detalhadas e exemplos de aplicações podem ser encontrados em um grande número de referências: [Widrow90], [Hush93], [Werbos90], [Haykin94], [Hetch-Nielsen89] entre outras.

Os diversos modelos de redes neurais distinguem-se e são classificadas tanto com relação às suas arquiteturas quanto com relação aos seus métodos de aprendizagem. As arquiteturas podem ser agrupadas segundo as classes recorrente e não-recorrente. Nas arquiteturas não-recorrentes as conexões impossibilitam que as saídas dos neurônios façam parte do conjunto de suas próprias entradas ou nelas influenciem. Redes neurais com tais arquiteturas são estáticas, ou seja, para cada sinal de entrada a rede neural exhibe uma única saída. As redes multicamadas são um exemplo desta classe de redes neurais. Já o modelo proposto por J. Hopfield, cuja arquitetura é mostrada na Figura 2.6, pertence à classe de redes recorrentes [Hopfield82], [Hopfield85]. Estas são sistemas dinâmicos em que cada neurônio exhibe estados intermediários (representados, por exemplo, por suas saídas) antes que atinjam um estado de equilíbrio. Outras arquiteturas caracterizam-se pelo fato dos neurônios se interconectarem totalmente. As redes do tipo Hopfield possuem arquitetura totalmente interconectada. As redes multicamadas pertencem a classe de redes neurais cujos neurônios são parcialmente interconectados. Algumas arquiteturas estabelecem um posicionamento espacial dos neurônios. Elas são arquiteturas do tipo reticulado. Este tipo de arquitetura permite que o conceito de distância entre neurônios seja definido, conceito este imprescindível às redes neurais do tipo Kohonen [Kohonen90], [Kohonen82].

Um dos critérios utilizados para se classificar os métodos de aprendizagem refere-se ao conteúdo das informações trocadas durante a aprendizagem da rede neural. Segundo este critério, os métodos de aprendizagem podem ser organizados em três grandes classes: supervisionados, não-supervisionados e por reforço. Nos métodos supervisionados a aprendizagem das redes neurais é realizada a partir de um conjunto de exemplos, cada qual formado por um par entrada/saída. Para cada exemplo, a rede neural recebe a entrada e deve aprender a reproduzir a saída correspondente. Portanto, não há necessidade de que a rede neural descubra quais saídas devem ser apresentadas como resposta às entradas. As saídas corretas são estabelecidas pelos exemplos de treinamento. Nos métodos não-supervisionados a aprendizagem se processa sem que saídas corretas sejam fornecidas. Estes métodos capacitam as redes neurais extrair correlações presentes nos dados de entradas. Após a aprendizagem, a rede neural torna-se capaz de classificar suas entradas de acordo com as correlações aprendidas. Os métodos de aprendizagem por reforço podem ser vistas como um meio termo entre os métodos já apresentados. Eles prescindem de supervisão. As redes neurais aprendem sem que disponham da informação a respeito

das respostas desejadas para as respectivas entradas apresentadas. No entanto, elas recebem informações adicionais que acompanham as entradas, ou seja, suas respostas são avaliadas de acordo com algum critério de desempenho estabelecido. O resultado desta avaliação é fornecido para a rede neural para que esta utilize-o visando o ajuste de seus pesos sinápticos. A rede neural deve encontrar quais saídas maximizam seu desempenho [Hetch-Nielsen89], [Haykin94], [Hentz91].

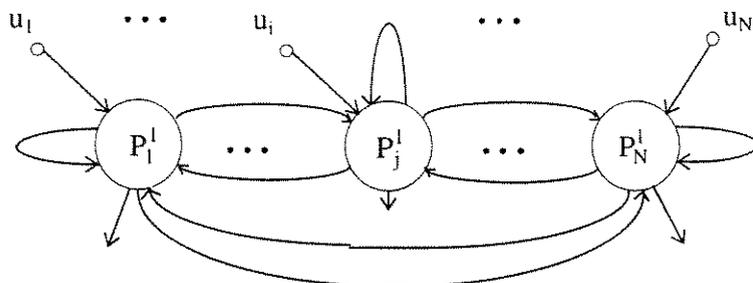


Figura 2.6: Modelo de rede neural recorrente.

Há diversos métodos de aprendizagem propostos em literatura. Os métodos do tipo correção de erro [Widrow90], do tipo Hebb [Hebb49], [Kosko92] e competitivo [Malsburg73], [Kohonen90] são aqueles que mais se destacam devido aos resultados que proporcionam e por servirem de base para outras diferentes propostas.

A rede neural cujo método de aprendizagem é do tipo correção de erro extrai seu conhecimento de um conjunto de pares de entrada/saída. Estes métodos são supervisionados e possuem como paradigma o método do tipo gradiente descrito anteriormente para a rede multicamadas. O erro $e = |y - v|$ entre a saída da rede y e a saída requerida v fornecida no conjunto de exemplos, ambas com relação à mesma entrada, é utilizado para cálculo do ajuste dos pesos sinápticos. A Figura 2.7 apresenta um esquema simplificado do método. Tanto a rede neural quanto o supervisor recebem o mesmo sinal de entrada u proveniente do ambiente. A rede neural gera a saída y que é comparada com a saída v determinada pelo supervisor. O sinal de erro $e(.)$ é então enviado à rede neural para que proceda os ajustes dos seus pesos sinápticos.

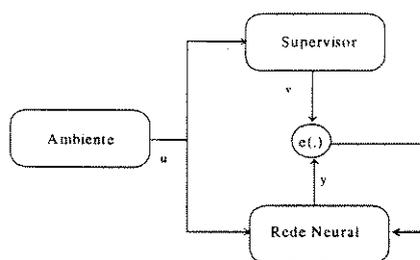


Figura 2.7: Método de aprendizagem supervisionado.

Os métodos do tipo Hebb são baseados na regra de ajuste sináptico proposto pelo neurofisiologista D. Hebb em 1949. Resultados de experimentos realizados com sistemas nervosos biológicos levaram D. Hebb concluir a lei de ajuste sináptico, nomeada por regra de Hebb em sua homenagem [Anderson95], [Hebb49]. Na sua essência esta lei afirma:

Quando um neurônio i está suficientemente próximo para estimular um neurônio j contribuindo freqüentemente para seu disparo, surgem alterações celulares em um ou em ambos os neurônios no sentido de aumentar a influência de i no disparo de j .

Em geral, considera-se que as alterações celulares correspondam a uma variação nos pesos sinápticos. Lembrando-se de que o disparo de um neurônio acontece quando este gera uma saída diferente de zero, a regra de Hebb pode ser formalizada tal qual em (2.6):

$$w_{ij}(t + 1) = w_{ij}(t) + \mu y_i(\mathbf{u}_i(t))y_j(\mathbf{u}_j(t)) \quad (2.6)$$

em que μ é o fator de aprendizagem.

O método de aprendizagem do tipo Hebb possui as seguintes características mais importantes:

1. Não há qualquer dependência do método com relação ao desempenho da rede;
2. Ajustes no peso sináptico somente acontecem se houver sincronismo entre disparos dos neurônios;
3. O ajuste se baseia em informações locais, ou seja, é importante considerar somente o sincronismo de disparo entre os neurônios envolvidos.

A regra de Hebb tem sido modificada visando incorporar outras propriedades, e.g., representação de comportamento de esquecimento e outras características [Kosko92].

Os métodos competitivos, tal como os métodos do tipo Hebb, possuem uma fundamentação neurofisiológica [Fausett94], [Kohonen90]. Eles são aplicados em redes neurais cujos neurônios desenvolvem uma dinâmica competitiva entre si, ou seja, para cada entrada apresentada à rede neural somente o neurônio vencedor e os neurônios em sua vizinhança permanecem ativados após a rede neural ter alcançado o estado de equilíbrio. Os ajustes dos pesos sinápticos dependem do resultado final desta competição. Este método será descrito em maiores detalhes na seção referente às redes do tipo Kohonen apresentada no item 2.4.

Alguns métodos de aprendizagem, além de ajustarem os pesos sinápticos da rede neural, também alteram a arquitetura, adicionando novos neurônios à topologia ou mesmo suprimindo alguns dos já existentes. Tais métodos são classificados como construtivos. Eles apresentam uma vantagem preponderante sobre os métodos que não alteram a arquitetura da rede neural: a arquitetura é dimensionada segundo a complexidade do problema. Ou seja, métodos construtivos não necessitam que a arquitetura da rede neural seja predefinida pelo projetista, evitando, de um lado, que a rede neural não corra o risco de se tornar incapaz de aprender devido ao subdimensionamento de sua arquitetura, e de outro, que a rede neural deixe de ser utilizada no total de suas potencialidades [Zuben96], [Ash89], [Fahlman90].

Cada classe de rede neural apresenta características particulares que se mostram mais apropriadas a um determinado tipo de aplicação ou tarefa. É possível classificar os diversos tipos de aplicação de acordo com os problemas básicos a serem solucionados em cada uma delas. Esta classificação permite um balizamento importante para escolha de um modelo de rede neural mais adequado a cada tarefa. Em seguida, algumas das classes mais

importantes são enumeradas juntamente com exemplos de trabalhos nos quais são relatados sua investigação:

1. Aproximação: Estão associadas a esta classe as tarefas em que a rede neural deve aprender uma relação funcional desconhecida $f(\cdot)$ a partir de um conjunto de pares ordenados do tipo (x, y) , $y = f(x)$. Em termos práticos, este tipo de problema surge em, e.g., modelagem ou identificação de sistemas. Os pares ordenados do tipo (x, y) correspondem ao mapeamento entrada/saída produzido pela planta. Ao aprender este mapeamento a rede neural extrai o modelo requerido. Os trabalhos de [Chen95^b] e [Poggio90] investigam as redes neurais atuando em problemas de aproximação de funções e [Walt96] em problemas de modelagem [Hornik89], [Cybendo89].
2. Associação: Algumas tarefas exigem que padrões prototípicos possam ser recuperados a partir dos próprios padrões degradados (mal definidos ou deformados devido a perturbações). As redes neurais são capazes de realizar este tipo de tarefa de maneira natural pois suas memórias são endereçadas por conteúdo. Inicialmente as redes neurais aprendem quais são os padrões prototípicos. Após a aprendizagem, ao serem estimuladas por algum padrão degradado, as redes neurais são capazes de identificar, recuperar e exibir o padrão prototípico que mais se assemelhe ao padrão apresentado na entrada. Vários trabalhos apresentam resultados com relação ao desenvolvimento de memórias associativas baseadas na teoria de redes neurais [Wang94], [Leurg94], [Kosko92], [Anderson95].
3. Classificação: No problema de classificação deve-se identificar a qual das classes previamente definidas cada padrão melhor se ajusta. A aprendizagem da rede neural depende de um supervisor que lhe apresente exemplos, cada qual formado por algum padrão e a classe a que pertence. Este tipo de problema é claramente associado aos métodos de aprendizagem supervisionados. O problema de reconhecimento de voz pode ser visto como um problema de classificação. Redes neurais compõem uma das principais frentes de pesquisa para desenvolvimento de sistemas capazes de reconhecer a voz humana [Sokol95], [Liu92]. Em [Chakrabarti95] e [Sackingen92] as redes neurais são aplicadas em classificação de padrões em sinais de radar e reconhecimento de caracteres, respectivamente.
4. Agrupamento: Neste tipo de problema, padrões devem ser agrupados de acordo com as similaridades que apresentem. Nenhuma informação sobre critérios que permitam o agrupamento dos padrões é fornecida. Estes critérios devem ser identificados a partir do próprio conjunto de padrões. Problemas de agrupamento são freqüentemente encontrados em análise de imagens. Resultados importantes têm sido obtidos através de redes neurais [Vinod94], [Phola96]. [Kohonen90].
5. Controle: A tarefa de controle é cumprida satisfatoriamente se os sinais de controle enviados ao sistema mantém seu desempenho dentro dos intervalos de variação requeridos [Narendra90]. As redes neurais aprendem a executar esta tarefa tanto através de métodos supervisionados quanto por reforço. No primeiro caso a aprendizagem é auxiliada por um supervisor que fornece os sinais de controle corretos, respectivos a cada situação apresentada pelo sistema. No segundo, as redes neurais ajustam seus pesos sinápticos utilizando-se apenas da avaliação de seu desempenho. Redes neurais têm contribuído consideravelmente nesta área de aplicação. Em

[Krishnakumar94] elas são empregadas em controle de helicópteros. Em [Hesselroth94], controlam braços de robôs pneumáticos.

6. Predição: No problema de predição deseja-se determinar o comportamento futuro de uma série temporal de dados $\{y(t_1), y(t_2), \dots, y(t_n)\}$, ou seja, o objetivo é determinar o dado correspondente a um instante t^* a partir de dados conhecidos em instantes t tal que $t < t^*$. A abordagem via redes neurais é investigada em várias aplicações, por exemplo, na predição de temperatura em conversores de aciarias [Keuthen95] e de volume de negócios [Thiesing95].
7. Otimização: Esta classe de tarefa é das mais importantes. Ela requer que se encontre soluções que maximizem ou minimizem determinado índice de desempenho ou função objetivo. Por exemplo, uma destas tarefas de otimização está relacionada ao problema do caixeiro viajante. Algumas classes de redes neurais são muito convenientes para o tratamento deste tipo de problema [Xia96], [Jagota 95], [Hopfield85].

Métodos de aprendizagem e arquiteturas não são específicos para um único tipo de rede neural, ou seja, um único método ou arquitetura pode ser associado a diversos modelos distintos. Além disso, uma única rede neural pode ser constituída por métodos de aprendizagem pertencentes a classes diferentes. Combinações de arquiteturas e métodos de aprendizagem dão origem a inúmeras propostas de novas classes de redes neurais. Algumas alcançam destacada importância devido aos resultados teóricos proporcionados, ao pioneirismo de conceitos e/ou idéias, à facilidade de implementação ou ao bom desempenho na solução de determinadas tarefas.

Diversas outras classes de redes neurais, diferentes das redes multicamadas, têm sido estudadas com intensidade. Por exemplo, a classe de redes neurais do tipo RBF (“Radial Basis Function”) [Haykin94], [Poggio90], [Broomhead88]. Elas possuem arquiteturas não-recorrentes constituídas por três camadas (uma camada de entrada). Simplificadamente, cada neurônio da segunda camada estabelece regiões de domínio no espaço de entradas tornando-se seu representante. A saída de cada um destes neurônios é ponderada por um peso sináptico que identifica a região que o neurônio representa. As respostas de todos os neurônios são agregadas na camada de saída da rede neural. Em problemas de classificação, os limites de cada classe são estabelecidos pelos neurônios da segunda camada e identificados pelos pesos sinápticos que ponderam suas saídas. Diferentes métodos podem ser adotados para aprendizagem das redes do tipo RBF. O método do tipo gradiente é freqüentemente empregado, embora haja a possibilidade de combinar métodos não-supervisionados e supervisionados. Por serem aproximadores universais, tal como as redes multicamadas, elas são importantes em problemas de aproximação de funções, modelagem e correlacionados [Gorinevsky95], [Chen95^b].

Outra classe de grande difusão na comunidade científica reúne as redes neurais do tipo Hopfield [Hopfield85]. Conforme discutido previamente, elas são sistemas dinâmicos devido suas arquiteturas recorrentes. Sob certas circunstâncias é possível demonstrar que tais sistemas são estáveis analisando-os por meio de funções de energia do tipo Liapunov. Para cada entrada apresentada à rede neural, seus estados intermediários compõem uma trajetória no seu espaço de estados. Esta trajetória é convergente para um estado estável correspondente a um mínimo local de sua função de energia. Em uma de suas principais aplicações, estas redes neurais são utilizadas como memórias associativas. Nestes casos, a

cada mínimo na função de energia deve corresponder a um padrão a ser armazenado. A configuração da rede neural é estabelecida a partir dos padrões prototípicos disponíveis utilizando-se as regras do tipo Hebb. Cada entrada apresentada à rede neural, em geral um padrão corrompido ou incompleto, força seus estados a deslocarem-se rumo a um mínimo local correspondente ao padrão armazenado mais semelhante à entrada. Ainda devido às suas propriedades dinâmicas, elas também são utilizadas em problemas de otimização combinatorial. As redes do tipo Hopfield têm sido pesquisadas em diversos tipos de aplicações, tais como identificação [Zhang96], otimização [Jagota95] e classificação [Linc96]. Em [Abe96] tais redes são estudadas visando melhorar sua velocidade de convergência.

Além delas ainda podem ser citadas as redes neurais do tipo ART [Carpenter87], [Grossberg87], [Linj96], [Carpenter95]. Elas lidam com um dilema intrínseco da aprendizagem: como adquirir novos conhecimentos sem que os existentes sejam perdidos. As redes neurais do tipo LVQ (“Learning Vector Quantization”) aprendem de forma supervisionada e utilizam o método competitivo [Haykin 94]. Há ainda outras de igual importância, tais como: as redes do tipo Kohonen [Kohonen90], [Kohonen82] e do tipo contrapropagação [Hetch-Nielsen89] que serão descritas na seção seguinte.

O sucesso das redes neurais como abordagem eficiente na solução de problemas complexos é proporcional e avança paralelamente ao desenvolvimento teórico. Uma parte significativa do esforço científico tem sido destinada à solução de questões teóricas de caráter fundamental, levantadas desde o desenvolvimento dos primeiros modelos básicos conhecidos. Dentre as principais questões estão: convergência da rede neural [Fang96], convergência do método de aprendizagem [Chang92] e capacidade de memória associada à arquitetura da rede neural [Leurg94]. Outras questões referem-se ao grau de robustez a perturbações [Chakrabarti95], à tolerância a falhas [Jim96], à velocidade de aprendizagem e processamento [Abe96] e à complexidade do método de aprendizagem [Kolen91]. Resultados para tais questionamentos têm promovido, como consequência, a criação de novos modelos de redes neurais, arquiteturas e métodos de aprendizagem.

Algumas linhas de pesquisa, cujos questionamentos e objetivos são bastante particulares, criam suas próprias teorias e estabelecem segmentos independentes dentro da pesquisa em redes neurais. Um destes segmentos parte do princípio de que a compreensão do comportamento inteligente não pode ser obtida independentemente do estudo do sistema nervoso biológico. Um de seus objetivos principais é enriquecer os modelos de redes neurais utilizando-se de resultados obtidos pela neurofisiologia. Sob esta perspectiva a Teoria de Seleção de Grupos Neurais (TSGN) vem desenvolvendo seus modelos [Edelman87]. A aprendizagem e características das redes neurais são explicados a partir da teoria da seleção natural que governa a evolução das espécies na natureza. Por meio desta abordagem a TSGN justifica a redundância nos sistemas nervosos biológicos e introduz conceitos tais como população de grupos neurais (grupos de neurônios) e seleção de grupos neurais. Além disso, de acordo com a TSGN, os ajustes sinápticos são funções da média da atividade dos neurônios consideradas no tempo e no espaço, diferindo-se significativamente da lei de Hebb que estabelece ajustes para sinapses relacionadas a atividades sincronizadas e locais. A TSGN tem sido aplicada em sistemas de controle autônomo, e.g., veículos autoguiados [Edelman87], [Vershure92] e [Reeke90].

Alguns pesquisadores argumentam que os modelos de redes neurais atuais apresentam comportamentos coletivos, mas não paralelos, pois qualquer mudança de atividade em um neurônio influencia a atividade de outro [Zak91]. M. Zak propõe modelos que não obedecem as condições necessárias para solução de equações diferenciais, tal como no caso das redes neurais dinâmicas usuais. Em consequência, seus modelos caracterizam-se pelo paralelismo computacional e por uma dinâmica dependente de condições iniciais, propriedade esta que favorece a capacidade de aprendizagem, em particular a criatividade.

A pesquisa em redes neurais tem se beneficiado com a integração de teorias da inteligência computacional aos seus modelos. Pelas suas características os algoritmos genéticos [Goldberg89] têm sido integrados às redes neurais, especificamente na composição de métodos de aprendizagem não-supervisionados. Os algoritmos genéticos podem ser vistos como métodos de otimização inspirados nos mecanismos que regem a evolução das espécies na natureza. Dois procedimentos básicos reproduzem os mecanismos naturais: seleção e recombinação. O método consiste basicamente em definir um conjunto de redes neurais (população de indivíduos) cada qual representada por um vetor (cromossomo) formado pelos seus pesos sinápticos. As redes neurais são inicialmente avaliadas de acordo com seus desempenhos na solução da tarefa ou problema de interesse. Em seguida um novo conjunto de soluções (nova geração) é formado a partir dos indivíduos iniciais. O mecanismo de seleção estabelece quais indivíduos (redes neurais) são escolhidos para gerar a nova população. Aqueles indivíduos de melhor desempenho possuem maior probabilidade de influenciar na formação da nova geração. O mecanismo de recombinação estabelece como os novos indivíduos serão gerados a partir das redes neurais selecionadas. Várias iniciativas têm investigado esta integração de abordagens da inteligência computacional [Ichikawa92], [Whitehead96], [Oliveira94] e [Park95].

Ainda sob esta mesma perspectiva, alguns modelos têm sido desenvolvidos exibindo dinâmicas caóticas presentes em modelos reais [Holden83]. Além de contribuir no sentido da modelagem [Aihara90], a teoria do caos também traz importantes contribuições com respeito à memória [Adachi92], à aprendizagem e ao raciocínio [Dingle93]. A. Dingle mostra que seu modelo, baseado em diagrama de bifurcação de sistemas caóticos, pode resolver algumas deficiências de aprendizagem apresentadas por redes neurais do tipo Kohonen.

Redes neurais nebulosas resultam da integração entre a teoria de redes neurais e a teoria de conjuntos nebulosos e lógica nebulosa e reúnem benefícios provenientes de ambas áreas da inteligência computacional [Takagi90]. De um lado as redes neurais nebulosas herdaram todas as características computacionais das redes neurais inclusive sua capacidade de aprendizagem. Por outro lado a teoria de conjuntos nebulosos e lógica nebulosa provê técnicas e conceitos fundamentais que possibilitam a representação formal de diversos aspectos lingüísticos de maneira simples e direta [Lee90]. Em consequência, as redes neurais nebulosas são potencialmente capazes de processar conhecimento de maneira que possa ser compreensivelmente extraído na forma lingüística. A introdução de conhecimento em sua arquitetura também é possível com equivalente comodidade. Além disso, desde que os sistemas nebulosos são, em sua essência, incapazes de aprender, as redes neurais nebulosas podem ou auxiliar no seu projeto determinando parâmetros tais como: funções de pertinência, regras nebulosas, etc.; ou substituí-los com a vantagem proporcionada pelas características de aprendizagem e adaptação [Figueiredo97^a], [Figueiredo97^b].

Atualmente as redes neurais formam uma área de pesquisa que não limita seu progresso a modelos baseados nos sistemas nervosos biológicos ou que neles perfeitamente se assemelhe. No entanto todas as novas propostas de uma ou outra forma apresentam propriedades comuns que qualitativamente as aproximam das redes neurais na forma que inicialmente foram concebidas, tais como: processamento paralelo, robustez, adaptação, etc.. Mais do que simples semelhanças que justifiquem sua denominação, são estas propriedades, principalmente a capacidade de aprendizagem, que mantêm os crescentes interesse e receptividade com relação às redes neurais. Como já foi enfatizado, as redes neurais além de serem sistemas computacionais, tornaram-se uma abordagem para solução de problemas devido à sua capacidade de aprendizagem. Diante de problemas de alta complexidade as redes neurais superam muitas das dificuldades que tornam abordagens clássicas ineficientes ou mesmo impotentes. Enquanto os métodos tradicionais são dependentes de um modelo físico do problema e no qual baseiam sua solução, as redes neurais não são limitadas por esta necessidade imprescindível. Também não se ressentem de conseqüências indesejáveis causadas por simplificações inadequadas do modelo (em que perturbações ou outros fatores influentes são desconsiderados) e que levam a eventuais, penosas e desgastantes reformulações da solução inicial. As redes neurais aprendem diretamente de dados que recebe de seu ambiente, portanto eliminam o esforço de modelagem física do problema e consideram todos os fatores mais importantes para a busca da solução. Além disso, quaisquer alterações de natureza física ou estocástica com relação ao ambiente são absorvidas pelas redes neurais em conseqüência de seu poder de adaptação. Comprova-se o sucesso das redes neurais observando-se o desempenho obtido em inúmeras aplicações a que estão submetidas: controle de braço de robô [Tsuji96], estabilidade de helicópteros [Krishnakumar94], análise de imagens de radar [Chakrabarti95], modelagem de processos industriais [Walt96], predição de instabilidades aerodinâmicas e controle de aeronaves [Faller95], reconhecimento de padrões em sinais de eletrocardiogramas [Suzuki95], etc.

Apesar do grande empenho da pesquisa na área, as redes neurais ainda estão longe de preencher as expectativas formadas nos seus primórdios. Algumas deficiências dos atuais modelos são: forte dependência entre o desempenho das atuais propostas e os parâmetros que as definem, sendo necessários parâmetros distintos para cada problema particular; e diferentes aplicações requerem diferentes aparatos computacionais que possibilitem a aprendizagem das redes neurais. No entanto os resultados obtidos mostram que as redes neurais ainda possuem muito potencial a ser explorado e o panorama que se apresenta indica um futuro bastante promissor nesta área da inteligência computacional.

2.4 Redes Neurais dos Tipos Kohonen e Contrapropagação

É possível se constatar, tanto no homem como em outros animais, que cada tipo de sinal sensorial é processado em uma área bastante específica do cérebro. Os sinais sensoriais são mapeados em suas respectivas áreas de forma que suas relações topológicas sejam preservadas, ou seja, sinais sensoriais semelhantes são representados em regiões cerebrais próximas. Por esta razão estas áreas ficaram conhecidas como mapas topográficos.

As redes neurais do tipo Kohonen emulam os sistemas nervosos biológicos sob este aspecto, ou seja, elas são capazes de formar mapas topográficos em sua arquitetura de

acordo com as características topológicas presentes nos sinais de entrada. Elas são reconhecidas não só por esta particularidade mas também pela simplicidade da arquitetura e do método de aprendizagem e pelo desempenho na solução de problemas de agrupamento [Kohonen90], [Kohonen82].

A arquitetura desta classe de redes neurais é do tipo reticulado. Esta arquitetura estabelece que cada neurônio possui uma posição espacial na rede neural, fator essencial para a formação dos mapas topográficos. Usualmente os neurônios são dispostos espacialmente em arranjos unidimensionais ou bidimensionais pelo fato de possibilitarem fácil observação dos mapas topográficos formados. A Figura 2.8 mostra a arquitetura reticulada unidimensional formada por P neurônios. Cada neurônio está conectado a todas as entradas da rede neural. O peso sináptico do neurônio P_j correspondente à entrada x_i é w_{ij} . A organização espacial dos neurônios torna possível se estabelecer o conceito de distância e de vizinhança entre neurônios. A distância V_{ji} entre os neurônios P_j e P_i pode ser convenientemente definida através de seus índices, por exemplo, $V_{ji} = |i - j|$. A vizinhança Π_j do neurônio P_j é formada pelo conjunto de neurônios que estão a uma distância menor que um valor máximo. Três vizinhanças distintas do neurônio P_j são ilustradas na Figura 2.9 (as conexões da rede neural foram omitidas).

A aprendizagem da rede neural do tipo Kohonen depende exclusivamente dos sinais de entrada. Além de ser não-supervisionado, o método de aprendizagem é competitivo. Após receberem um sinal de entrada, os neurônios competem entre si disputando o controle sobre os procedimentos relacionados à aprendizagem. O vencedor da competição é determinado de acordo com as distâncias dos vetores de pesos sinápticos ao vetor de entrada.

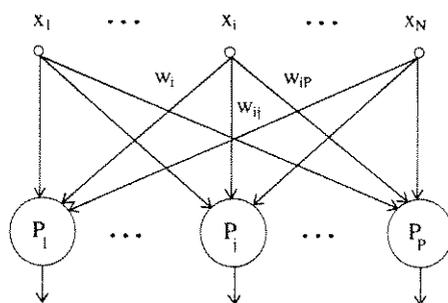


Figura 2.8: Arquitetura unidimensional da rede do tipo Kohonen.

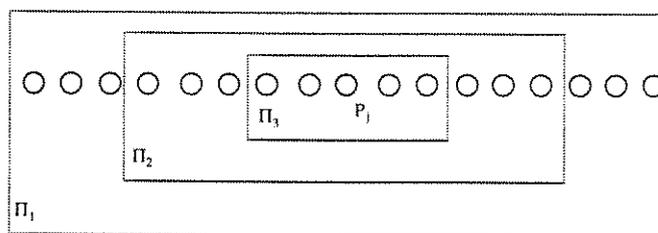


Figura 2.9: Vizinhanças do neurônio P_j .

Os pesos sinápticos são ajustados a cada sinal de entrada apresentado à rede neural. Considere que $\mathbf{u}(t)$ seja a t -ésima entrada apresentada à rede neural, $t \in Z^+ = \{0, 1, \dots\}$, e

que para esta entrada o vetor de pesos sinápticos do neurônio j seja $\mathbf{w}_j(t)$. Assim que os neurônios recebem a entrada $\mathbf{u}(t)$ cada um deles computa a distância $D_j(t)$ entre esta entrada e seu vetor de pesos sinápticos $\mathbf{w}_j(t)$ tal como em (2.7):

$$D_j(t) = \|\mathbf{u}(t) - \mathbf{w}_j(t)\|; \quad (2.7)$$

em que: $\|\cdot\|$ é a norma Euclidiana (outras normas podem ser utilizadas), $\mathbf{w}_j(t) = (w_{1j}(t) \dots w_{ij}(t) \dots w_{Nj}(t))$ e $\mathbf{u}_j(t) = (u_1(t) \dots u_i(t) \dots u_N(t))$.

O neurônio vencedor $\xi(t) \in Z^+$, correspondente à entrada $\mathbf{u}(t)$, possui o vetor de pesos sinápticos mais próximo da entrada e pode ser determinado de acordo com (2.8):

$$\xi(t) = \arg \min_j (D_j(t)). \quad (2.8)$$

A equação de ajuste dos pesos sinápticos é tal qual em (2.9). Observa-se a partir desta equação que o ajuste sináptico é fortemente dependente do neurônio vencedor. Somente neurônios pertencentes à sua vizinhança são ajustados. Outra característica importante que pode ser observada a partir de (2.9) é que o vetor de pesos sinápticos após o ajuste torna-se mais próximo do vetor da entrada.

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \eta(u_i(t) - w_{ij}(t)), & \text{se } j \in \Pi_{\xi(t)}; \\ w_{ij}(t), & \text{caso contrário,} \end{cases} \quad (2.9)$$

em que η é o fator de aprendizagem ($0 < \eta < 1$).

Tanto η quanto a vizinhança $\Pi_{\xi(t)}$ variam dinamicamente ao longo da aprendizagem, conforme justificativa discutida adiante.

Considere que a influência de $\Pi_{\xi(t)}$ sobre o ajuste dos pesos sinápticos seja modelada pela função $\pi(\xi(t), j, t)$ definida em (2.10) e ilustrada na Figura 2.10:

$$\pi(\xi(t), j, t) = \begin{cases} 0, & \text{se } \|\xi(t) - j\| > \delta(t); \\ 1, & \text{caso contrário,} \end{cases} \quad (2.10)$$

em que $\delta(t) \geq 0$ (largura de vizinhança) define os elementos de $\Pi_{\xi(t)}$ correspondente à t -ésima entrada.

Desta forma a equação (2.9) resulta:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\pi(\xi(t), j, t)(u_i(t) - w_{ij}(t)). \quad (2.11)$$

A função $\pi(\xi(t), j, t)$ também pode ser definida tal como em (2.12) para que o ajuste dos pesos sinápticos seja ponderado diferentemente para os neurônios pertencentes à vizinhança $\Pi_{\xi(t)}$.

$$\pi(\xi(t), j, t) = \exp^{-(|j-\xi(t)|/\Delta(t))^2} \quad (2.12)$$

em que $\Delta(t)$ possui função análoga a $\delta(t)$ em (2.10) e corresponde à vizinhança efetiva com relação ao ajuste de pesos sinápticos.

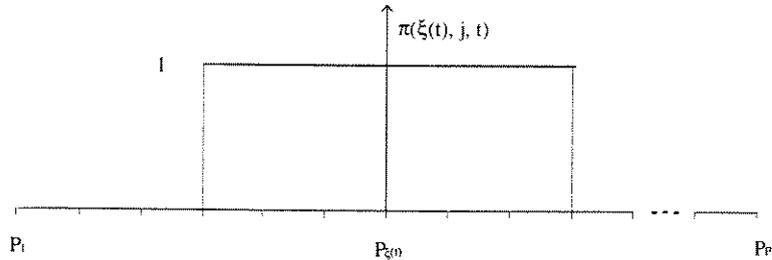


Figura 2.10: Função $\pi(\xi(t), j, t)$ para $\delta(t) = 3$.

Para que o processo de aprendizagem seja capaz de formar mapas topográficos sobre a arquitetura da rede neural, as dinâmicas do fator de aprendizagem $\eta(t)$ e da vizinhança do neurônio vencedor devem apresentar algumas características particulares. A vizinhança do neurônio vencedor deve variar ao longo do processo de aprendizagem. Logo que as primeiras entradas são apresentadas à rede neural, $\Pi_{\xi(t)}$ pode conter todos os neurônios. À medida que a aprendizagem progride, a vizinhança deve ser reduzida até que $\Pi_{\xi(t)}$ contenha somente os neurônios adjacentes ao neurônio vencedor ($\delta(t) = 1$) ou somente o neurônio vencedor ($\delta(t) = 0$). A dinâmica do fator de aprendizagem $\eta(t)$ é importante tanto no que se refere à formação dos mapas topográficos quanto à convergência do algoritmo. O fator $\eta(t)$ deve ser inicializado em um valor máximo e reduzido a cada entrada apresentada à rede neural. Ele deve ser modelado por meio de função de t monotônica decrescente positiva. Equações recorrentes (2.13) cujos efeitos são equivalentes a decaimentos exponenciais são possíveis escolhas para $\delta(t)$ (ou $\Delta(t)$ conforme o caso) e $\eta(t)$:

$$\begin{aligned} \delta(t+1) &= \rho\delta(t) \\ \eta(t+1) &= \theta\eta(t), \end{aligned} \quad (2.13)$$

em que: ρ e θ , $0 < \rho, \theta < 1$, são constantes e tanto $\eta(0)$ quanto $\delta(0)$ devem ser definidos.

Após um número suficiente de ajustes, cada vetor sináptico tende a representar um subconjunto do espaço de entradas. Este subconjunto é tal que a distância de quaisquer de seus vetores ao vetor sináptico que o representa é a menor possível quando comparada aos outros vetores sinápticos da rede neural. Ou seja, se w_j representa o subconjunto C_j , então para qualquer vetor $u(t)$ pertencente a C_j tem-se:

$$D_j(u(t)) \leq D_i(u(t)), \quad i = 1, \dots, P.$$

Portanto, o vetor de pesos sinápticos do neurônio vencedor representa o subconjunto a que pertence o vetor de entrada. A Figura 2.11 ilustra os vetores sinápticos w_j (após a aprendizagem ter sido completada) e os respectivos subconjuntos que representam C_j , $j = 1, \dots, 5$, contidos no espaço de entradas U .

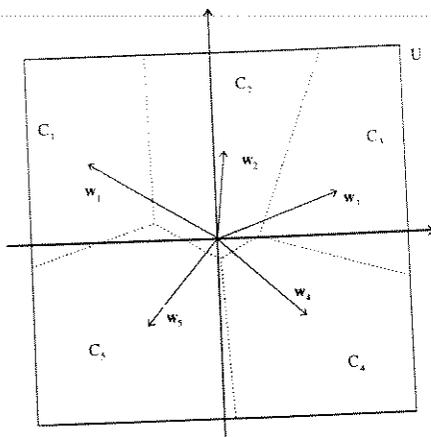


Figura 2.11: Vetores sinápticos w_j após a aprendizagem e seus respectivos subconjuntos C_j .

A aprendizagem da rede neural completa-se quando os mapas topográficos estão bem formados sobre a arquitetura da rede neural. Neste caso observa-se que não há mudanças em quaisquer dos subconjuntos determinados pelos respectivos vetores sinápticos. O algoritmo de aprendizagem é tal como segue:

1. Inicialização: Escolha os valores para os pesos sinápticos $w_{ij}(0)$ de forma aleatória, preferivelmente considerando uma distribuição uniforme. É desejável que os vetores $w_j(0)$ possuam módulos semelhantes e que concentrem-se em uma região restrita do espaço de entradas. Defina os modelos para o decaimento da largura de vizinhança e do fator de aprendizagem. Defina a função $\pi(\xi(t), j, t)$.
2. Estímulo: Apresente à rede neural um vetor $u(t)$ escolhido aleatoriamente pertencente ao conjunto de vetores de entrada.

3. Competição: Compute as distâncias $D_j(t)$, $j = 1, \dots, P$, com relação à entrada $u(t)$ e determine o neurônio vencedor $\xi(t)$ para o qual a distância $D_j(t)$ é mínima:

$$D_j(t) = \|u(t) - w_j(t)\|,$$

$$\xi(t) = \arg \min_j (D_j(t)).$$

4. Ajuste e Atualização: Ajuste os pesos sinápticos de acordo com a equação:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\pi(\xi(t), j, t)(u_i(t) - w_{ij}(t)).$$

Atualize o fator de aprendizagem $\eta(t)$ e a função $\pi(\xi(t), j, t)$.

5. Avaliação de Convergência: Retorne ao passo 2 caso os mapas topográficos não estejam bem formados. Caso contrário siga para 6.

6. Fim da aprendizagem.

Quando os mapas topográficos são bem formados duas características podem ser observadas:

- 1) Todo o espaço de entradas é representado na arquitetura da rede neural. Cada neurônio (através de seus pesos sinápticos) representa um subconjunto específico deste espaço. Além disso, a relação topológica existente entre os vetores sinápticos se mantém sobre os respectivos neurônios a que estão associados, ou seja, neurônios próximos estão associados à vetores sinápticos próximos.
- 2) O mapeamento topográfico reflete não só as relações topológicas mas também as características estatísticas presentes na distribuição dos vetores de entrada. Cada região do espaço de entradas é representada de maneira proporcional à probabilidade com que seus vetores são apresentados à rede neural durante o processo de aprendizagem. Um maior número de neurônios é associado às regiões cujos vetores de entrada são escolhidos com maior probabilidade.

Em geral as redes neurais do tipo Kohonen comportam-se adequadamente desde que os parâmetros sejam escolhidos de forma conveniente. Entretanto, não são conhecidos resultados teóricos que indiquem qual o melhor modelo (mais adequados para a aprendizagem da rede neural) para as dinâmicas da vizinhança do neurônio vencedor e do fator de aprendizagem, respectivamente $\delta(t)$ (ou $\Delta(t)$ conforme o caso) e $\eta(t)$. Sabe-se, porém, que são de fundamental importância para que os mapas topográficos sejam formados satisfatoriamente. As heurísticas existentes sempre são acompanhadas de uma apuração através da tentativa e erro. Em simulações é conveniente dividir a aprendizagem em duas fases. Na primeira, é importante a organização dos vetores sinápticos sobre o espaço de entradas, ou seja, os subconjuntos representados pelos pesos sinápticos tornam-se bem definidos. O número de entradas necessárias para a organização dos pesos sinápticos deve ser determinado por tentativa e erro. Somente na segunda fase o fator de aprendizagem é reduzido lentamente até atingir um valor mínimo, permanecendo constante até o final da simulação. Nesta fase os vetores sinápticos convergirão para os valores que melhor representem seus respectivos subconjuntos.

Se o fator de aprendizagem é constantemente reduzido durante o processo de aprendizagem a convergência dos vetores sinápticos é garantida. Caso contrário os subconjuntos representados pelos vetores sinápticos podem variar indefinidamente. Note-se que, entretanto, a boa formação dos mapas topográficos não é garantida pela convergência forçada dos vetores sinápticos alcançada pela redução do fator de aprendizagem. É necessário que as dinâmicas relacionadas à vizinhança do neurônio vencedor e ao fator de aprendizagem sejam convenientemente escolhidas.

Considerando-se que os vetores sinápticos representam subconjuntos do espaço de entradas delimitados de acordo com as relações topológicas e características estatísticas deste espaço, as redes neurais do tipo Kohonen têm sido empregadas nos casos em que a solução de problemas de agrupamento torna-se importante, por exemplo em classificação de padrões [Mitra94], [Figueiredo95^a]. Elas têm sido associadas a redes neurais de outras

classes, por exemplo, redes neurais multicamadas ou do tipo LVQ [Haykin94]. Nestes casos as redes do tipo Kohonen têm duas funções importantes: reduzir a dimensão do espaço de entradas e compactar os dados utilizados durante a aprendizagem. As duas funções são realizadas por estas redes neurais em consequência de suas propriedades. O mapeamento do espaço de entradas sobre sua arquitetura pode ser entendido como uma redução de dimensões, ou seja, vetores no espaço de entradas serão representados por neurônios organizados sobre uma arquitetura definida, eventualmente, em dimensão inferior. A compactação de dados se sucede durante a aprendizagem da rede neural desde que cada vetor sináptico representa todos os vetores pertencentes a um particular subconjunto do espaço de entradas. A associação das duas redes neurais se faz pospondo-se a segunda rede neural à rede do tipo Kohonen, ou seja, as entradas da segunda rede neural são formadas pelas saídas da primeira. Entre outras aplicações estão: controle de braços de robôs [Hesselroth94], [Campos96], construção de modelos de imagens 3D [Chen96] e planejamento de sistemas de potência [Al-Sakaf95].

As redes do tipo contrapropagação foram desenvolvidas a partir das redes do tipo Kohonen tendo como principal objetivo sua aplicação em problemas relacionados à aproximação de funções [Hecht-Nielsen89], [Nie95]. A arquitetura desta rede neural (em sua versão mais simples) é do tipo não-recorrente formada por duas camadas (Figura 2.12). A primeira camada é definida de forma equivalente à rede neural do tipo Kohonen. Cada um dos P neurônios desta camada recebe todas as entradas x_i , $i = 1, \dots, N$, provenientes do ambiente. O peso sináptico do neurônio j associado à entrada x_i é w_{ij} . Os vetores sinápticos w_{ij} são ajustados utilizando-se o método competitivo descrito anteriormente, em geral restringindo-se ao ajuste somente ao neurônio vencedor. As saídas dos neurônios da primeira camada são definidas de acordo com o resultado da competição. Para cada entrada apresentada à rede neural a saída y_j^1 do j -ésimo neurônio desta camada é 1 se este é vencedor, caso contrário é zero. Cada neurônio j da primeira camada está conectado a cada neurônio k da segunda camada através de conexões z_{jk} . Esta segunda camada computa as saídas da rede neural.

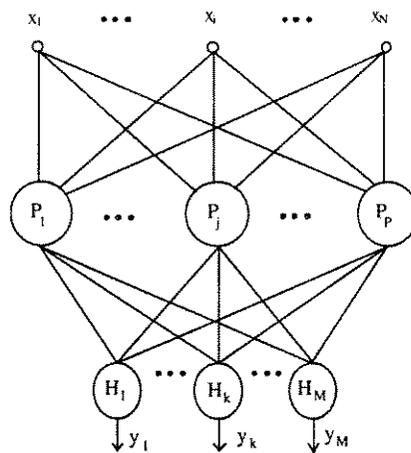


Figura 2.12: Arquitetura das redes neurais do tipo contrapropagação.

A saída $y_k(t)$ do k -ésimo neurônio da segunda camada é definida por:

$$y_k(t) = \sum_{j=1}^P (y_j^1(t) z_{jk}(t))$$

Uma vez que:

$$y_j^l(t) = \begin{cases} 1, & \text{se } j = \xi(t); \\ 0, & \text{se } j \neq \xi(t), \end{cases} \quad (2.14)$$

em que $\xi(t)$ é o índice do neurônio vencedor da camada de Kohonen correspondente à entrada $\mathbf{u}(t)$, obtém-se:

$$y_k(t) = z_{\xi(t)k}(t). \quad (2.15)$$

Os ajustes dos pesos sinápticos da segunda camada dependem de supervisão, ao contrário da aprendizagem dos pesos sinápticos w_{ij} . Tal como nas redes multicamadas com funções de ativação sigmoidais (“multilayer perceptron”) a rede neural do tipo contrapropagação recebe durante o processo de aprendizagem pares de vetores $(\mathbf{u}(t), \mathbf{v}(t))$, em que $\mathbf{u}(t) = (u_1(t), \dots, u_N(t))'$ é a entrada da rede neural e $\mathbf{v}(t) = (v_1(t), \dots, v_M(t))'$ corresponde ao vetor que a rede neural deve exibir em sua saída ao ser estimulada por esta entrada (saída desejada). Em geral, os pares $(\mathbf{u}(t), \mathbf{v}(t))$ estabelecem uma relação funcional, ou seja, $\mathbf{v}(t) = g(\mathbf{u}(t))$ tal que $g(\cdot): U \subset \mathbb{R}^n \rightarrow V \subset \mathbb{R}^M$. A equação de ajuste para os pesos sinápticos $z_{jk}(t)$ é tal como descrita em (2.16) [Hetch-Nielsen89]:

$$z_{jk}(t+1) = z_{jk}(t) + \mu(v_k(t) - z_{jk}(t))y_j^l(t) \quad (2.16)$$

em que $0 < \mu < 1$ é o fator de aprendizagem.

Considerando-se (2.14) e (2.15) a equação de ajuste (2.16) pode ser reescrita tal como em (2.17):

$$z_{jk}(t+1) = \begin{cases} z_{jk}(t) + \mu(v_k(t) - z_{jk}(t)), & \text{se } j = \xi(t); \\ z_{jk}(t), & \text{se } j \neq \xi(t). \end{cases} \quad (2.17)$$

A mesma equação (2.17) pode ser obtida ao aplicar-se o método de aprendizagem do tipo gradiente no ajuste dos pesos $z_{jk}(t)$. Seja $\mathbf{z}(t)$ a matriz de pesos sinápticos $z_{jk}(t)$. Defina a função índice de desempenho $E(\mathbf{z}(t))$ tal como em (2.18):

$$E(\mathbf{z}(t)) = 1/2 \sum_{k=1}^M (v_k(t) - y_k(t))^2. \quad (2.18)$$

Neste caso, para cada par $(\mathbf{u}(t), \mathbf{v}(t))$ apresentado à rede neural o ajuste do peso sináptico $z_{jk}(t)$ é computado segundo a expressão (2.19):

$$z_{jk}(t+1) = z_{jk}(t) - \mu \partial E(z_{jk}(t)) / \partial z_{jk}(t), \quad (2.19)$$

em que μ é o mesmo definido em (2.16).

O valor da derivada parcial de $E(\mathbf{z}(t))$ é dada por:

$$\frac{\partial E(z_{jk}(t))}{\partial z_{jk}(t)} = -(v_k(t) - y_k(t))y_j^1(t). \quad (2.20)$$

Substituindo (2.20) na expressão (2.19) e considerando-se (2.14) e (2.15) chega-se a (2.17).

Note-se que os ajustes de $z_{jk}(t)$ são controlados pelas saídas dos neurônios da primeira camada. À medida que a aprendizagem progride, os vetores w_j (formados pelos pesos sinápticos w_{ij}) definem uma repartição do espaço de entradas em subconjuntos C_j , cada vetor w_j representando um subconjunto C_j , tal qual na rede do tipo Kohonen. Após os conjuntos C_j terem sido bem definidos, os pesos sinápticos $z_{jk}(t)$ serão ajustados apenas quando $u(t) \in C_j$, pois somente neste caso y_j^1 é diferente de zero. Em consequência, a média do vetor $z_j(t) = (z_{j1}(t) \dots, z_{jM}(t))'$ ao longo de todos os ajustes tenderá para a média dos vetores $v(t)$ tais que $u(t) \in C_j$. Acrescente-se ainda que quanto maior o número de neurônios na primeira camada menor será a variação possível entre todos os vetores de $u(t)$ pertencentes a cada C_j . Conseqüentemente, menor será a variação dos $v(t)$ associados a estes vetores $u(t)$. Portanto, considerando-se um número suficientemente grande de neurônios na primeira camada é possível se alcançar a aproximação desejada para os pares ordenados $(u(t), v(t))$. No caso extremo e hipotético é possível se associar a cada entrada $u(t)$ um neurônio na primeira camada de forma que a aproximação seja exata. Este raciocínio conduz à prova de que as redes do tipo contrapropagação são aproximadores universais [Hetch-Nielsen89].

2.5 Conjuntos Nebulosos e Sistemas Nebulosos: Introdução

Segundo a teoria clássica de conjuntos, cada conjunto $A \subset U$, sendo U o conjunto universo, está associado a uma função característica (ou função de pertinência) $A(x): U \rightarrow \{0, 1\}$ (2.18). Esta função define se o elemento x do universo U pertence ou não pertence ao conjunto A .

$$A(x) = \begin{cases} 1, & \text{se } x \in A; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.18)$$

Considere o seguinte exemplo: $U = \{1; 3; 6; 7; 9; 11; 34\}$ e $A = \{1; 11; 34\}$. Então, a função $A(x)$ é definida pelo seguinte conjunto de pares ordenados: $\{(1; 1), (3; 0), (6; 0), (7; 0), (9; 0), (11; 1), (34; 1)\}$.

Muitas vezes os conjuntos representam classes definidas sobre atributos tais como velocidade, peso, altura, custo, desempenho, erro, etc.. Seja U o conjunto universo de todas as velocidades de automóveis entre 0km/h e 200km/h. Seja A igual ao conjunto das velocidades maiores que 50km/h. A função característica do conjunto A é dada por:

$$A(x) = \begin{cases} 1, & \text{se } x > 50\text{km/h;} \\ 0, & \text{caso contrário.} \end{cases}$$

Nem sempre a definição (2.18) possibilita uma representação conveniente de certos tipos de classes, particularmente aquelas definidas por meio de termos pouco precisos. Suponha a classe das velocidades altas. Em geral, para se representar esta classe através de um conjunto clássico A , define-se um parâmetro v^a tal que velocidades acima de v^a sejam consideradas altas (Figura 2.13). Desta forma tem-se:

$$A(x) = \begin{cases} 1, & \text{se } x \geq v^a; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.19)$$

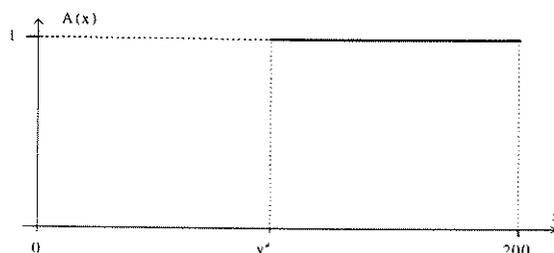


Figura 2.13: Função de pertinência definida em (2.19).

Este artifício traz alguns inconvenientes. Velocidades muito próximas e inferiores a v^a não pertencem ao conjunto A . Por exemplo, se v^a é igual a 100km/h então, segundo a função característica (2.19), 99km/h e 98km/h são velocidades que não pertencem à classe das velocidades altas. Esta conclusão não está de acordo com a noção intuitiva para tais velocidades quando comparadas com o parâmetro v^a . Espera-se que velocidades inferiores e relativamente próximas a v^a também sejam consideradas velocidades altas embora não tanto quanto velocidades acima de v^a . Ou seja, quanto mais elevada a velocidade maior é o grau com que se deseja que ela pertença a classe das velocidades altas. A teoria clássica de conjuntos falha neste tipo de representação pois estabelece que um elemento pertence ou não pertence ao conjunto, sendo impossível uma gradação de pertinência.

A teoria de conjuntos nebulosos abandona a rigidez da teoria clássica de conjuntos introduzindo os conceitos de conjunto nebuloso e grau de pertinência [Zadeh65]. A teoria de conjuntos nebulosos pode ser entendida como uma extensão da teoria clássica de conjuntos. Cada elemento x do universo de discurso U está associado ao conjunto nebuloso A com um certo grau definido pela função de pertinência $A(x): U \rightarrow [0, 1]$. A função $A(x)$ assume valores em todo o intervalo $[0, 1]$ e não somente seus extremos, tal como estabelece a teoria clássica.

A teoria de conjuntos nebulosos torna mais intuitiva a representação da classe de velocidades altas citada anteriormente. É possível representar esta classe de velocidades pelo conjunto nebuloso A definido pela função de pertinência (2.20) (Figura 2.14):

$$A(x) = \left(1 + \exp(-(x - 100))\right)^{-1} \quad (2.20)$$

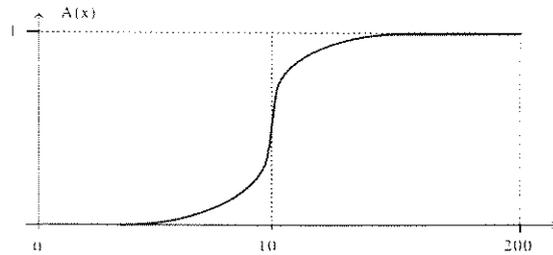


Figura 2.14: Função de pertinência A(x).

A função de pertinência em (2.20) é um exemplo particular. Outras funções podem ser escolhidas para se definir o conjunto nebuloso que modela a classe das velocidades altas (Figura 2.15).

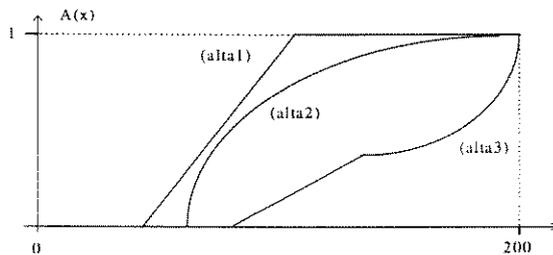


Figura 2.15: Diferentes funções de pertinência do conceito alta.

De forma análoga, conceitos para atributos distintos podem ser modelados através de conjuntos nebulosos. A Figura 2.16 mostra funções de pertinência do tipo triangular correspondentes a três conceitos do atributo temperatura: baixa, média e alta.

Na teoria clássica as operações entre conjuntos são realizadas por meio de operadores do tipo união, interseção e negação. Os principais operadores entre conjuntos nebulosos são definidos por duas classes de funções de duas variáveis denominadas t-norma e s-norma, respectivamente. A t-norma mínimo (\min) e s-norma máximo (\max) são as mais comumente utilizadas (2.21). Considere dois conjuntos nebulosos A e B e suas respectivas funções de pertinência A(x) e B(x), então:

$$\min(A, B) = \min(A(x), B(x)) = \begin{cases} A(x), & \text{se } A(x) \leq B(x); \\ B(x), & \text{caso contrário,} \end{cases} \quad (2.21)$$

$$\max(A, B) = \max(A(x), B(x)) = \begin{cases} A(x), & \text{se } A(x) \geq B(x); \\ B(x), & \text{caso contrário.} \end{cases}$$

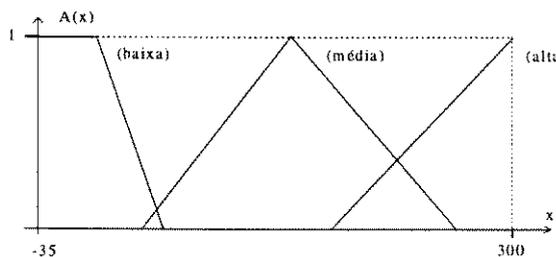


Figura 2.16: Conjuntos nebulosos correspondentes às temperaturas baixa, média e alta.

Existe um grande número de t-normas e s-normas definidas na teoria de conjuntos nebulosos, e.g.: produto, produto drástico (t-normas) e soma limitada e soma drástica (s-normas). Todas as funções pertencentes a classe de t-normas e s-normas devem satisfazer a um conjunto de propriedades (2.22). Sejam x, y, z, u e v elementos pertencentes ao universo de discurso no qual estão definidas as s-normas $S(., .)$ e as t-normas $T(., .)$. Seja $R(., .)$ uma função pertencente a classe das s-normas ou das t-normas indistintamente. Então, $\forall x, y, z$ e $v \in [0,1]$, as seguintes propriedades são válidas [Yager94]:

- 1 Comutatividade: $R(x, y) = R(y, x)$.
- 2 Associatividade: $R(x, R(y, z)) = R(R(x, y), z)$.
- 3 Monotonicidade: Se $x \leq u$ e $y \leq v \rightarrow R(x, y) \leq R(u, v)$. (2.22)
- 4 Condições de contorno: $S(0, z) = z, S(1, z) = 1$; e
 $T(0, z) = 0, T(1, z) = z$.

Conceitos vagos ou imprecisos tais como aqueles previamente citados (pequeno, grande, muito grande, etc.) são com freqüência empregados na forma lingüística para descrever conhecimento. Em muitos casos o conhecimento é expresso por meio de regras do tipo se-então, e.g., “Se a pressão é baixa então o volume é grande” ou “Se a temperatura é alta então a certeza de infecção é alta”. A teoria da lógica nebulosa foi desenvolvida a partir da teoria de conjuntos nebulosos para tratar formalmente tais expressões. Considere o exemplo “Se a pressão é alta então a temperatura é baixa”. A seguinte regra nebulosa pode ser utilizada em sua representação:

Se X é A então Y é B .

Nesta regra X e Y são variáveis nebulosas correspondentes aos atributos pressão e temperatura, respectivamente. Os conjuntos nebulosos A e B são conceitos associados a estes atributos. O conjunto nebuloso A corresponde ao conceito alta enquanto B refere-se ao conceito baixa.

Talvez a mais importante diferença entre a lógica clássica e a lógica nebulosa encontra-se nos mecanismos de inferência. A lógica nebulosa possibilita modelar o raciocínio intuitivo e aproximado usualmente utilizado (de forma inconsciente ou não). Este, de uma forma geral, baseia-se na seguinte heurística: se um fato implica uma conclusão então fatos semelhantes implicam em conclusões também semelhantes. Considere o mecanismo de inferência nebuloso:

- premissa 1: X é A
premissa 2: Se X é A então Y é B (2.23)
conclusão: Y é B

A premissa 2 (regra) em (2.23) estabelece o conhecimento apenas com relação ao antecedente X é A . Muito embora a premissa 1 (fato) apresente um fato diferente, eventualmente muito semelhante, é possível concluir algo a respeito de Y . Este resultado é

compatível com os mecanismos de inferência aproximados. ou seja, é possível se obter conclusões para fatos que não correspondem exatamente ao conhecimento existente.

Considere um caso mais geral em que o mecanismo de inferência nebulosa seja aplicado sobre uma regra nebulosa definida para M variáveis X_i em seu antecedente (2.24):

$$\begin{array}{l} \text{premissa 1: } X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \text{ e } \dots X_M \text{ é } A_M \\ \text{premissa 2: } \underline{\text{Se } X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \text{ e } \dots X_M \text{ é } A_M \text{ então } Y \text{ é } B} \end{array} \quad (2.24)$$

conclusão: $Y \text{ é } B$

A implementação do mecanismo de inferência visando se obter o conjunto nebuloso B não é única. Segundo um dos métodos mais conhecidos [Yager94], o conjunto B é computado de acordo com a seguinte seqüência de procedimentos: comparação, agregação e conclusão:

1. Comparação: O fato descrito na premissa 1 é comparado, separadamente em cada variável, com o correspondente antecedente da premissa 2. Seja P_i o resultado da comparação para a i -ésima variável, então:

$$P_i = \bigwedge_{x \in U_i} \left(T(A_i(x), A_i(x)) \right), \quad (2.25)$$

em que: \bigwedge e T são, respectivamente, uma s-norma e uma t-norma e U_i é o universo de discurso da i -ésima variável.

(A notação $\bigwedge_{x \in U_i} (., .)$ deve ser entendida como a aplicação da norma $\bigwedge(., .)$ sobre todo $x \in U_i$ recorrendo-se, para tanto, à sua propriedade associativa.)

2. Agregação de antecedentes: Os resultados individuais M_i (2.25) são tomados em conjunto gerando o valor H (2.26). A agregação corresponde à modelagem dos conectivos “e” presentes na regra nebulosa.

$$H = \bigwedge_{i=1}^M (P_i) \quad (2.26)$$

em que: $\bigwedge_{i=1}^M (x_i) = T(x_M, T(x_{M-1}, T(\dots, T(x_2, x_1))))$.

3. Conclusão: Nesta etapa a função de pertinência do conjunto nebuloso B (2.27) é gerada através de um operador I e do resultado da agregação H (2.26). Um dos operadores I mais utilizados é a t-norma min definida anteriormente.

$$B(y) = \bigwedge_{y \in U_y} (P, B(y)), \quad (2.27)$$

em que U_y é o universo de discurso com relação à variável nebulosa Y .

Uma última etapa adicional torna-se necessária se existir mais de uma regra.

4. Agregação de regras: Esta etapa corresponde à agregação das conclusões inferidas para cada regra individualmente tal como em (2.28). Considere, portanto, que existam N regras nebulosas e que o resultado da inferência para a regra i tenha sido B^i , então:

$$B(y) = \sum_{i=1}^N (B^i(y)), \quad (2.28)$$

em que a s -norma sobre N argumentos é definida de maneira análoga a (2.25).

As teorias de conjuntos nebulosos e da lógica nebulosa apresentam-se atualmente bastante desenvolvidas e são tratadas em vasta literatura [Zimmermann87], [Klir95], [Yager94] e [Kaufmann75]. A rápida explanação dos fundamentos destas teorias teve como principal objetivo mostrar as diferenças mais marcantes quando comparadas às teorias clássicas. Tais diferenças, tanto no que se refere aos conjuntos nebulosos quanto aos mecanismos de inferência, refletem-se basicamente no poder de representação formal de expressões lingüísticas e do raciocínio aproximado. Especificamente, conceitos vagos (pequeno, médio, muito grande, intenso, fraco, etc.) e o raciocínio intuitivo, descrito na forma de regras do tipo se-então, recebem um tratamento conveniente sob o aspecto matemático e computacional.

Os sistemas nebulosos são sistemas computacionais com múltiplas aplicações, desenvolvidos a partir das teorias de conjuntos nebulosos e da lógica nebulosa, visando aproveitar de maneira pragmática o potencial destas teorias. Estes sistemas processam conhecimento e inferem conclusões (saídas) a partir das entradas apresentadas. Quatro componentes básicos constituem os sistemas nebulosos: interface de entrada, módulo de inferência, bases de regras e dados e interface de saída (Figura 2.17). A interface de entrada transforma, quando necessário, os sinais provenientes do ambiente (entrada) em conjuntos nebulosos de forma que sejam convenientemente tratados pelo restante do sistema. Ela define os conjuntos A_i da premissa 1 em (2.24). A base de regras contém as regras nebulosas do tipo se-então que representam o conhecimento do sistema nebuloso. A base de dados contém informações sobre parâmetros de outros módulos do sistema, e.g., operadores de agregação e inferência. O módulo de inferência gera as respostas do sistema na forma de conjuntos nebulosos, de acordo com o método de inferência escolhido, e.g., aquele definido pelas equações (2.25) a (2.28). As respostas exibidas pelo módulo de inferência são transformadas de maneira a se tornarem compatíveis com o ambiente na interface de saída [Lee90], [Pedrycz93], [Figueiredo93].

Os sistemas nebulosos processam informação na forma de conjuntos nebulosos. As classes de funções de pertinência $A(x)$, $x \in U$, mais freqüentemente empregadas para implementação da base de regras nebulosas são definidas por dois parâmetros: valor modal (z) e dispersão (δ) e satisfazem as seguintes propriedades:

Normalidade: $A(z) = 1$;

Simetria: $\forall u, v \in U$ se $|z - u| = |z - v|$ então $A(u) = A(v)$. (2.29)

Satisfazem estas propriedades funções de pertinência típica do tipo Gaussianas ou triangulares, entre outras (Figura 2.18).

Quando a entrada do sistema nebuloso é um número $z \in \mathfrak{X}$ a interface de entrada gera um conjunto nebuloso A que o representa da forma mais adequada. As funções de pertinência $A(x)$ mais utilizadas são do tipo unitária:

$$A(x) = \begin{cases} 1, & \text{se } x = z; \\ 0, & \text{se } x \neq z. \end{cases} \quad (2.30)$$

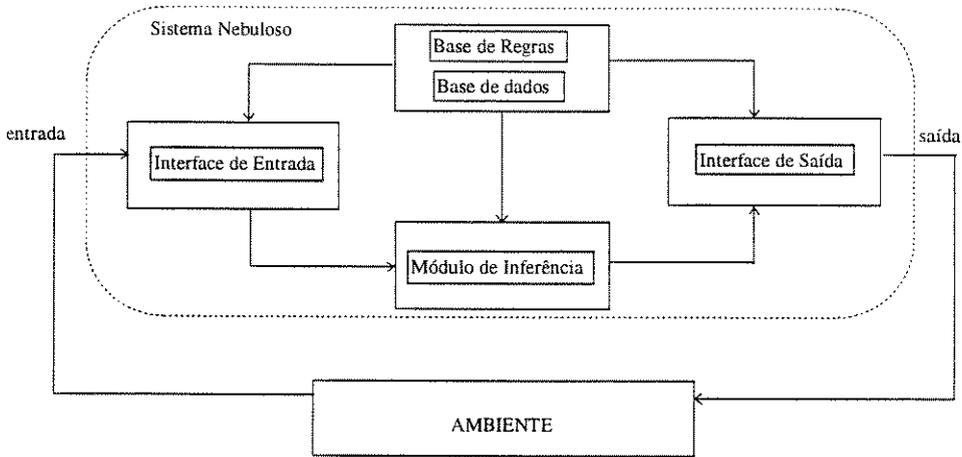


Figura 2.17: Sistema nebuloso: módulos principais e respectivas interações.

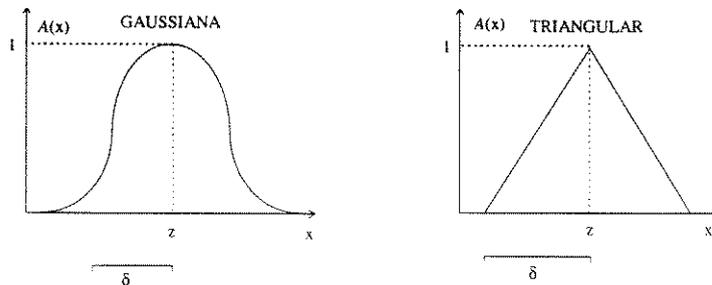


Figura 2.18: Funções de pertinência típicas.

Muitas vezes é interessante trabalhar com os universos de discurso discretizados, principalmente para efeitos de cálculos computacionais. Por exemplo, seja x a entrada do sistema nebuloso, $I_k = (x_{kI}, x_{kF}]$ o k -ésimo intervalo de discretização e $A(x)$ um conjunto unitário, então:

$$A(x) = \begin{cases} 1, & \text{se } x \in I_{k^*}; \\ 0, & \text{se } x \in I_k, k \neq k^*. \end{cases} \quad (2.31)$$

A Figura 2.19 mostra as representações gráficas das funções de pertinência unitária nos universos contínuo e discretizado.

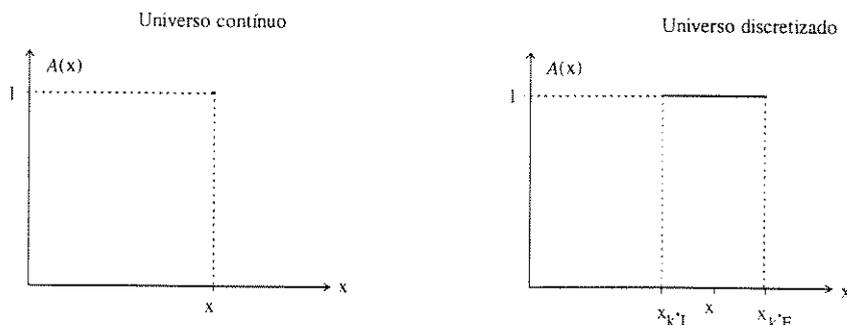


Figura 2.19: Funções de pertinência unitárias em universos contínuos ou discretizados.

A interface de saída possui a mesma finalidade que a interface de entrada, só que no sentido inverso. Existem vários métodos utilizados em interfaces de saída para se gerar números $\bar{y} \in \mathfrak{X}$, saída do sistema nebuloso, representativos dos conjuntos nebulosos $B(y)$. O método centro de área é o mais utilizado (2.28):

$$\bar{y} = \frac{\sum_{k=1}^K y_k B(y_k)}{\sum_{k=1}^K B(y_k)}; \quad (2.32)$$

em que: y_k é o valor representativo do k -ésimo intervalo de discretização do universo de discurso da variável de saída e K é o número de discretizações deste universo de discurso.

Os sistemas nebulosos aproveitam todo o potencial exibido pelas teorias em que se baseiam, principalmente no que se refere à representação do conhecimento e à reprodução do raciocínio aproximado. A princípio, se o conhecimento necessário e disponível para resolver um problema é passível de ser representado através do formalismo das teorias de conjuntos nebulosos e da lógica nebulosa, então os sistemas nebulosos podem ser utilizados na solução deste problema. A importância dos sistemas nebulosos torna-se evidente nos casos em que a modelagem do problema via métodos tradicionais é impraticável. Sem modelos ou diante de modelos de extrema complexidade a abordagem clássica torna-se impotente. Nestes casos os sistemas nebulosos podem ser empregados como uma abordagem alternativa poderosa [Yager94]. Mas não é somente nos casos extremos que os sistemas nebulosos oferecem vantagens. Obter a solução via abordagem clássica pode tornar-se um processo árduo e custoso. Além das etapas de modelagem, análise do modelo e síntese da solução intrínsecas a esta abordagem, muitas vezes a solução teórica deve ser ajustada à realidade da aplicação prática. Em contraste, os sistemas nebulosos eliminam muitas etapas presentes nas abordagens tradicionais. A solução que apresentam na forma de expressões lingüísticas já embute de forma codificada tanto as leis mais importantes que regem o modelo físico do problema quanto as incertezas mais influentes a ele associadas. Os sistemas nebulosos também evitam reformulações drásticas da solução muito comuns nas abordagens tradicionais, usualmente causadas por simplificação excessiva do modelo inicial ou por aproximações inadequadas durante o desenvolvimento da solução. Os sistemas nebulosos caracterizam-se pela sua simplicidade no que se refere à teoria que os fundamenta, e pela forma direta e intuitiva com que são empregados, ou seja, são capazes de utilizar de maneira pragmática o conhecimento codificado em expressões lingüísticas, espelhando características inteligentes correspondentes à representação do conhecimento e ao raciocínio aproximado [Pedrycz93]. Além disso, algumas classes de sistemas nebulosos são aproximadores

universais [Castro95]. Em essência, estes sistemas são capazes de processar o conhecimento de forma compreensível e de manipular incertezas e imprecisões visando solucionar problemas reais complexos [Zimmermann87].

Para o projeto de sistemas nebulosos é imprescindível que vários parâmetros sejam definidos, principalmente no que se refere à construção da base de regras, por exemplo: número de regras nebulosas, funções de pertinência correspondentes aos antecedentes e conseqüentes de cada regra, operadores que representam a semântica das regras e operadores de agregação [Bastian95]. Desde suas origens, os sistemas nebulosos são projetados usualmente por meio de métodos empíricos. Alguns fatores contribuíram para que esta abordagem se tornasse mais comum entre aquelas adotadas pelos projetistas: forma direta e simples com que as regras nebulosas do tipo se-então podem ser formuladas a partir de regras existentes na forma lingüística; a teoria de conjuntos nebulosos possibilita grande flexibilidade com respeito à definição das funções de pertinência e operadores em geral, e.g., operadores de agregação; o desempenho dos sistemas nebulosos não sofre de degradações significativas causadas por parâmetros definidos de forma não ótima. No entanto, este tipo de abordagem torna-se inconveniente nos casos em que: há necessidade de se automatizar os procedimentos do projeto ou de se oferecer auxílios na execução desta tarefa; o problema exige que sua solução seja constantemente adaptada, e; não há conhecimento disponível na forma lingüística, mas que pode ser obtido através de um conjunto de dados numéricos [Nie95], [Figueiredo95^b].

Um grande esforço tem sido dedicado ao desenvolvimento de metodologias capazes de projetar sistemas nebulosos também nos casos em que a abordagem por tentativa-e-erro torna-se inadequada ou impotente [Manoranjan95]. Técnicas de programação matemática [Fukumoto95] e algoritmos de agrupamento [Delgado95], algoritmos genéticos [Ishibuchi95] são algumas das ferramentas utilizadas neste sentido. Redes neurais nebulosas, tais que a representação do conhecimento em suas arquiteturas se faz de forma compreensível, têm se caracterizado como ferramentas de grande potencial no projeto destes sistemas, pois são capazes de adquirir conhecimento e, conseqüentemente, definir os parâmetros de sistemas nebulosos [Abe95], [Lin95], [Mitra96], [Lin91], [Figueiredo97^a]. Esta classe de redes neurais é focalizada na próxima seção.

Muito embora ainda restem questões em aberto, os sistemas nebulosos encontram-se atualmente profundamente difundidos nas diferentes áreas do conhecimento e servem às mais variadas aplicações práticas: controle de plantas de energia solar [Rubio95], reconhecimento de padrões [Grader95], monitorização de qualidade de água [Denoeux95], controlador de freio ABS [Mauer95], estabilizador para imagens de câmera de vídeo [Egusa95], etc. Toda esta receptividade e interesse em torno desta abordagem confirma seu bom desempenho e simplicidade e atesta seu grande sucesso.

2.6 Redes Neurais Nebulosas

À medida que as teorias relacionadas às redes neurais e aos sistemas nebulosos se desenvolviam, observou-se que estas abordagens poderiam ser unificadas, resultando em sistemas de características mais completas no que se refere às suas capacidades inteligentes. A unificação ocorreu de forma bastante natural, pois além destas abordagens exibirem características comuns, e.g., tratamento de incertezas e raciocínio aproximado,

cada uma apresenta propriedades exclusivas que contribuem para compensar deficiências individuais da outra. De um lado as redes neurais falham em representar o conhecimento de forma explícita, mas exibem sua inclinação para o aprendizado. Por outro lado os sistemas nebulosos são fracos sob o ponto de vista da aquisição automática de conhecimento, porém valem-se amplamente de seu dom com respeito à manipulação de termos lingüísticos. O resultado desta simbiose se concretizou nas redes neurais nebulosas. Elas herdaram as características computacionais das redes neurais, suas arquiteturas e seus métodos de aprendizagem; e adquiriram dos sistemas nebulosos a capacidade de representação e processamento compreensível do conhecimento, bem como os demais recursos oferecidos pelas teorias de conjuntos nebulosos e lógica nebulosa.

Muito embora os modelos clássicos de neurônios serem potenciais candidatos ao papel de unidades computacionais das redes neurais nebulosas (desde que mapeiem o resultado da agregação de suas entradas sobre o intervalo $[0, 1]$), novas propostas de neurônios (neurônios nebulosos) mais adequados a esta classe de redes neurais foram pesquisadas.

A possibilidade de modelagem mais fiel dos sistemas nervosos biológicos se constitui em uma das principais motivações para o desenvolvimento de neurônios nebulosos. Atividades químicas presentes nas sinapses e sinais elétricos gerados no axônio da célula nervosa real são respectivamente associados ao processamento simbólico e numérico no neurônio nebuloso [Rocha92]. Sob esta perspectiva vários modelos foram propostos. No modelo proposto em [Figueiredo95^a] sinais de entrada e pesos sinápticos do neurônio nebuloso parametrizam funções de pertinência de conjuntos nebulosos (Figura 2.20). As atividades sinápticas, a agregação de sinais no corpo celular e a geração de sinais no axônio são modelados respectivamente por operadores monotônicos, operador de agregação e uma função de codificação não linear. Este modelo define uma classe de neurônios nebulosos bastante geral, pois dependendo da escolha conveniente de operadores é possível se representar diferentes propostas, inclusive o modelo clássico de neurônio. Ainda neste trabalho o modelo proposto é utilizado para compor uma rede neural nebulosa aplicada no reconhecimento de padrões. A organização espacial dos neurônios é garantida por uma arquitetura do tipo reticulado bidimensional e o método de aprendizagem adotado é do tipo competitivo. Ao final da aprendizagem os mapas topográficos formados na arquitetura da rede neural nebulosa estabelecem agrupamentos através dos quais os padrões apresentados são identificados.

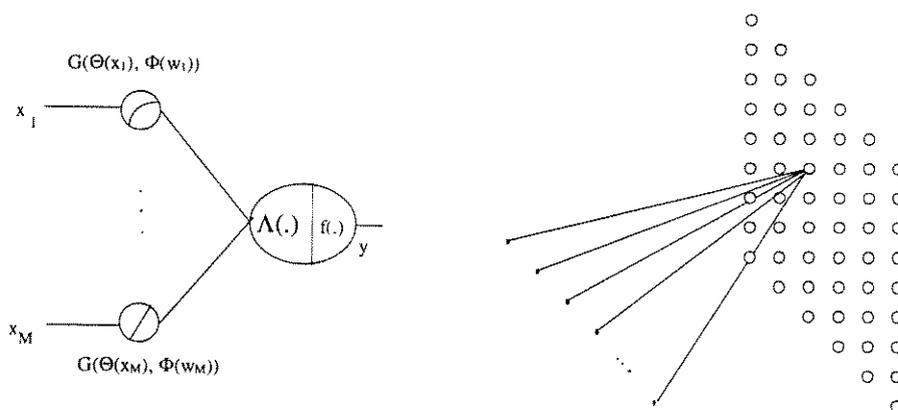


Figura 2.20: Modelo de neurônio e uma arquitetura de rede neural nebulosa.

Já os modelos propostos em [Pedrycz95] foram desenvolvidos sob uma perspectiva distinta visando usufruir das potencialidades da teoria da lógica nebulosa. Eles reproduzem operações lógicas baseadas em t-normas e s-normas. Dois tipos principais de neurônios nebulosos são propostos: conjunção e disjunção. Neurônios do tipo conjunção agregam sinais de acordo com t-normas enquanto os neurônios do tipo disjunção modelam esta operação através de s-normas. Ambos os modelos são utilizados na formação de arquiteturas de redes neurais nebulosas denominadas processadores lógicos que demonstram certas analogias com as funções Booleanas da lógica clássica.

Nem todas as redes neurais nebulosas possibilitam representar em suas estruturas conhecimento na forma lingüística, por exemplo, aquelas citadas anteriormente ([Figueiredo95^a] e [Pedrycz95]). Esta classe de redes neurais nebulosas utiliza os recursos das teorias de conjuntos nebulosos e lógica nebulosa visando principalmente obter as capacidades de modelagem de incertezas e/ou do raciocínio aproximado.

As redes neurais nebulosas cujas arquiteturas representam o conhecimento de forma compreensível (explícito) apresentam duas diferenças importantes com relação à classe anterior. Além de possibilitarem a inserção e/ou extração de conhecimento na forma de regras do tipo se-então nebulosas, são também capazes de projetar sistemas nebulosos definindo seus parâmetros à medida que adquirem conhecimento durante o período de aprendizagem. Elas estabelecem um mapeamento entre regras nebulosas e o conhecimento presente em suas arquiteturas, são capazes de realizar raciocínio aproximado, e interfacear entradas e saídas. Por estas razões estas redes neurais nebulosas podem ser vistas sob diferentes perspectivas: dos sistemas nebulosos, das redes neurais ou das redes neurais nebulosas. Em seguida, apresentam-se alguns exemplos adicionais a título de ilustração.

O principal foco do trabalho apresentado em [Lin95] é modelar (através de regras nebulosas do tipo se-então) sistemas, extraindo conhecimento a partir de dados de entrada e saída. Sob esta perspectiva uma classe de redes neurais nebulosas é proposta (Figura 2.21). Sua arquitetura não-recorrente é formada por 4 camadas. Os pesos sinápticos da segunda e quarta camadas correspondem aos parâmetros que definem as funções de pertinência dos antecedentes e conseqüentes das regras nebulosas, respectivamente. A segunda e terceira camadas processam as entradas de forma análoga aos procedimentos de inferência (2.25) e (2.26). Na Figura 2.21 cada grupo de neurônios envoltos em um retângulo tracejado realiza estes procedimentos com relação a uma regra específica. Os resultados obtidos pela terceira camada ponderam os conseqüentes de cada regra nebulosa para então serem combinados no único neurônio da última camada. Os autores introduzem o conceito de curvas nebulosas com as quais determinam pesos sinápticos iniciais e número de regras nebulosas com o qual definem a arquitetura da rede. O método de aprendizagem através do qual a rede neural nebulosa ajusta seus pesos sinápticos é do tipo gradiente.

A arquitetura da rede neural nebulosa sugerida em [Lin91] é parcialmente predefinida (Figura 2.22). Ela torna-se completamente formada ao término do aprendizado. As partições de cada universo de discurso são predefinidas e determinam os grupos de neurônios que compõem a segunda e a quarta camadas. Cada neurônio nestas camadas representa um conjunto nebuloso. As regras nebulosas são representadas por neurônios da terceira camada. Suas entradas provém de neurônios da segunda camada (que representam antecedentes) e suas saídas destinam-se a neurônios da quarta camada (estes representam

conseqüentes). A quinta camada realiza as operações equivalentes à interface de saída dos sistemas nebulosos. Os parâmetros que definem as funções de pertinência são ajustados durante o período de aprendizagem que compreende uma fase não supervisionada e outra supervisionada. O número de regras nebulosas codificadas na rede (número de neurônios da terceira camada) também é determinado durante a aprendizagem. O desempenho da rede neural nebulosa é verificado em aplicações relacionadas a controle de trajetória de veículo e a seqüenciamento em sistemas de manufatura.

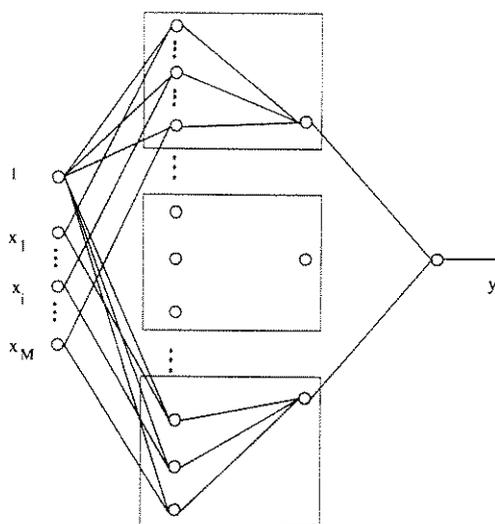


Figura 2.21: Rede neural nebulosa de arquitetura de 4 camadas não recorrente.

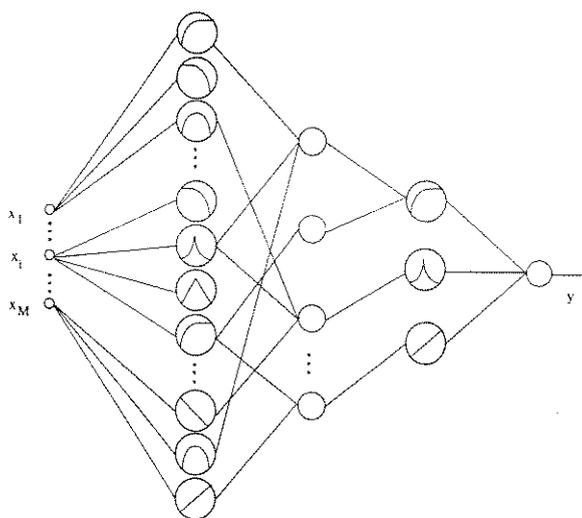


Figura 2.22: Modelo de rede neural nebulosa de arquitetura variável.

J. Nie inspirou-se na rede neural do tipo contrapropagação, tanto no que se refere a sua arquitetura quanto aprendizagem, para desenvolver seu modelo de rede neural nebulosa [Nie95]. Três camadas compõem a rede neural nebulosa. Os neurônios da segunda camada computam unificadamente os procedimentos de comparação e agregação de antecedentes, restando ao único elemento da terceira camada o procedimento de agregar os conseqüentes de cada regra. Os pesos sinápticos da segunda camada representam parâmetros das funções de pertinência com respeito aos antecedentes das regras nebulosas.

Conseqüentes (números reais) são representados pelos pesos sinápticos da terceira camada. A arquitetura não é predefinida no que se refere ao número de neurônios de cada camada. Neurônios são inseridos na segunda camada durante a fase de aprendizagem. Os pesos sinápticos são ajustados de acordo com métodos supervisionados e não-supervisionados. J. Nie demonstra as potencialidades de sua rede neural nebulosa na solução de problemas de modelagem (Figura 2.23).

Os três modelos apresentados, assim como outras propostas, diferenciam-se pela arquitetura, métodos de aprendizagem, parâmetros predefinidos, representação do conhecimento; e têm como objetivo comum integrar as características cognitivas simbólicas presentes em sistemas nebulosos com as potencialidades de aprendizagem das redes neurais. Esta classe de redes neurais nebulosas são capazes de extrair conhecimento a partir de um conjunto de dados numéricos e traduzir de maneira inteligível por meio de regras lingüísticas.

Outros modelos têm sido desenvolvidos sob diferentes motivações e visando aplicações distintas. Em [Keller92] propõe-se uma rede neural nebulosa para executar os procedimentos computacionais de sistemas nebulosos. O conhecimento presente nas regras nebulosas é explicitamente codificado nos pesos sinápticos da rede neural. Este modelo caracteriza-se pelo fato da resposta rede reduzir-se ao “modus ponens” tradicional se a entrada puder ser representada de acordo com a teoria de conjuntos clássica.

A rede neural nebulosa proposta em [Mitra94] é baseada na rede do tipo Kohonen. O modelo é capaz de se auto-organizar, mesmo na presença de dados de entrada cujas classes não são bem definidas em suas fronteiras. Nestes casos a rede neural nebulosa exhibe melhores resultados que o modelo original. Após a aprendizagem, ela é capaz de determinar, para cada entrada desconhecida, o grau de pertinência da entrada com respeito as várias classes existentes. A rede neural nebulosa é aplicada no problema de reconhecimento de voz, considerando vários extensões de sua arquitetura e conjuntos de treinamento.

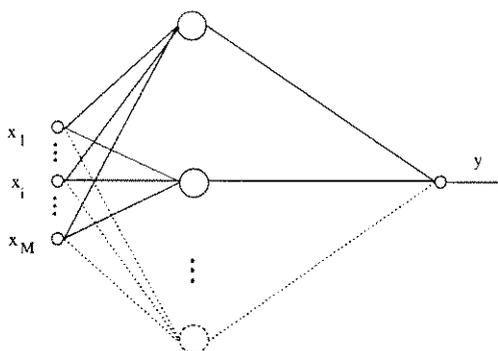


Figura 2.23: Rede neural nebulosa de arquitetura variável com 3 camadas.

Dois objetivos principais nortearam a pesquisa que culminou com os modelos propostos em [Abe95]. O primeiro foi desenvolver redes neurais nebulosas cujas habilidades na solução de problemas de classificação fossem equivalentes às apresentadas por redes neurais tradicionais, porém que apresentassem menor esforço de aprendizagem. O segundo foi desenvolver modelos cujo desempenho fosse satisfatório mesmo em problemas de grande dimensão. A rede neural resultante extrai regras nebulosas diretamente dos dados

numéricos de entrada/saída presentes em problemas de classificação de padrões. As regras nebulosas são determinadas através de um processo recorrente em que as superposições entre classes são eliminadas. A rede neural nebulosa é comparada com outros modelos de redes neurais quando aplicadas em problemas de classificação.

As redes neurais nebulosas dispõem de recursos integrados provenientes de áreas já consolidadas e reconhecidamente profícuas. Observa-se que suas potencialidades são investigadas nos mais diversos tipos de aplicações: controle de concentração de monóxido de carbono [Tanaka95], detecção de agentes químicos [Vuorimaa95], navegação de veículos autônomos [Hiraga95], controle de tráfego urbano [Baglio94], caracterização de sinais cardíacos [Chen95^a], controle de braço de robô [Doersan94] etc.. Entre suas características mais interessantes está a capacidade de aprender a partir de dados numéricos e exibir o conhecimento adquirido de forma compreensível. Elas também são capazes de exibir o comportamento inverso. Elas codificam conhecimento simbólico que as capacita desempenhar tarefas computacionais. Portanto, elas são capazes de alterar a natureza do conhecimento, de numérico a simbólico e vice-versa, adequando-se às necessidades prementes.

Os modelos descritos nesta seção são exemplos resultantes de intensa pesquisa e desenvolvimento e comprovam a grande expectativa em torno destes sistemas computacionais adaptativos e inteligíveis.

2.7 Resumo

Neste capítulo apresentou-se os conceitos fundamentais relacionados às teorias de redes neurais e de conjuntos nebulosos. Foram ressaltadas as potencialidades mais importantes destas teorias que as tornam abordagens alternativas aos métodos tradicionais na solução de problemas complexos.

Os conceitos relacionados às redes neurais foram introduzidos sob a perspectiva dos sistemas nervosos biológicos nos quais elas foram inspiradas. Saliaram-se suas características mais importantes, tais como: capacidade de adaptação e aprendizagem, processamento paralelo e raciocínio aproximado. Foram apresentados modelos, classes de arquiteturas e de métodos de aprendizagem. Algumas das linhas de pesquisa mais recentes na área de redes neurais foram citadas, entre elas aquela relacionada ao desenvolvimento de redes neurais nebulosas. As redes neurais do tipo Kohonen e do tipo contrapropagação foram enfatizadas por serem de interesse particular para o entendimento dos capítulos posteriores.

A introdução à teoria de conjuntos nebulosos foi direcionada para a descrição dos sistemas nebulosos, pois nestes se concentram as potencialidades de aplicação prática daquela teoria. Os sistemas nebulosos foram descritos considerando-se cada um de seus componentes principais: interface de entrada, base de conhecimento, máquina de inferências e interface de saída. Foram enfatizadas algumas de suas características, tais como: representação de conhecimento na forma lingüística, processamento de incertezas, raciocínio aproximado.

O principal objetivo deste capítulo foi alcançado com a descrição de modelos de redes neurais nebulosas. As redes neurais nebulosas unificam as teorias de redes neurais e de conjuntos nebulosos. Elas acumulam as potencialidades e eliminam deficiências de cada teoria. Por estas razões a pesquisa em redes neurais nebulosas permanece intensa refletindo expectativas no que se refere ao desenvolvimento de modelos computacionais, conexionistas e inteligíveis, capazes de auxiliar em projetos de sistemas nebulosos e de propiciar maior fidelidade aos modelos biológicos.

Nos capítulos seguintes, contribuições à pesquisa nesta área são apresentadas através de propostas de novas classes de redes neurais nebulosas.

CAPÍTULO 3

REDES NEURAIS NEBULOSAS APLICADAS EM MODELAGEM DE SISTEMAS

3. Redes Neurais Nebulosas Aplicadas em Modelagem de Sistemas

3.1 Introdução

Constituir modelos nebulosos de sistemas reais significa reproduzir o mapeamento entrada/saída destes sistemas através de, por exemplo, sistemas nebulosos discutidos no capítulo anterior. O sistema real e seu modelo nebuloso, representado por um sistema nebuloso, submetidos às mesmas entradas retornam a mesma saída como resposta. Modelo e sistema real são equivalentes sob a perspectiva de entrada/saída. Desta forma, compor um modelo nebuloso pode ser entendido como projetar um sistema nebuloso equivalente ao sistema real, o que por sua vez significa determinar os parâmetros que o definem.

No projeto de sistemas nebulosos é imprescindível que vários parâmetros sejam definidos, principalmente no que se refere à inserção de conhecimento na base de regras, por exemplo: número de regras nebulosas, funções de pertinência correspondentes aos antecedentes e conseqüentes de cada regra [Bastian95]. Estes são os de maior importância para o processo de modelagem por estarem associados diretamente ao conhecimento do sistema nebuloso. (Outros itens de projeto incluem: operadores de agregação e que determinam a semântica das regras, parâmetros das interfaces de entrada e saída. Estes estão mais relacionados aos mecanismos e operações necessários aos procedimentos computacionais do sistema nebuloso e, portanto, não precisam ser determinados para cada modelo em particular, podendo permanecer comuns a todos eles.)

Desde suas origens, os sistemas nebulosos são projetados por meio de métodos empíricos e procedimentos baseados em tentativa e erro. Este tipo de abordagem tornou-se usual devido a características intrínsecas aos sistemas nebulosos, tais como: forma direta e simples com que as regras nebulosas do tipo se-então podem ser formuladas a partir de regras existentes na forma lingüística a teoria de conjuntos nebulosos possibilita grande flexibilidade com respeito a definição de seus parâmetros e; o desempenho dos sistemas nebulosos não sofre degradações significativas causadas por parâmetros definidos de forma não ótima. No entanto, a abordagem do tipo tentativa e erro, quando adotada nos casos em que o sistema real exhibe complexidade significativa, pode se tornar inconveniente. Quanto maior a complexidade associada ao sistema real, maiores são as dificuldades para se determinar os parâmetros mais adequados ao modelo nebuloso.

Pesquisadores têm se empenhado no desenvolvimento de metodologias capazes de projetar sistemas nebulosos nos casos em que a abordagem por tentativa e erro torna-se inviável ou impotente [Manoranjan95]. Em [Wakabayashi95] são utilizadas equações de relações nebulosas integrada a técnicas de programação matemática no projeto de sistemas nebulosos. Entretanto, a solução de tais equações demanda grande esforço. Algumas ferramentas de modelagem nebulosa baseiam-se em algoritmos de agrupamento [Delgado95]. Muitas vezes estes métodos não compõem regras nebulosas em que o conhecimento possa ser expresso com relação a cada variável lingüística separadamente e por vezes a base de regras nebulosas formada pode não cobrir completamente o espaço de entradas. De uma forma geral, tanto uma quanto a outra abordagem envolvem intervenções de um agente externo em várias etapas do processo de modelagem [Lin91].

Redes neurais nebulosas têm se mostrado uma poderosa abordagem na determinação de parâmetros para modelagem nebulosa. Diversas classes de redes neurais nebulosas têm sido sugeridas, principalmente aquelas em que o conhecimento pode ser inserido ou extraído na forma lingüística. Estes modelos computacionais são capazes de automatizar o processo de modelagem, adquirir o conhecimento necessário para especificar os parâmetros da base de regras e representar este conhecimento na forma de regras nebulosas do tipo se-então, potencialidades não disponíveis em outros métodos de modelagem.

Conforme discutido no capítulo anterior, a classe de redes proposta em [Lin91] utiliza, numa primeira fase, procedimentos não supervisionados para aprender a topologia e adquirir conhecimento inicial. Em seguida, um algoritmo supervisionado sintoniza os parâmetros buscando uma especificação ótima. A aprendizagem das redes neurais nebulosas propostas em [Lin95] é fortemente dependente de sua configuração inicial. Pesos sinápticos e arquitetura são determinados após análise dos dados disponíveis através do conceito de curvas nebulosas. O ajuste dos pesos sinápticos iniciais é realizado utilizando-se o método de aprendizagem baseado no gradiente.

Estas redes neurais fazem parte de uma extensa classe de modelos propostos na literatura que compartilham uma limitação comum: a aprendizagem depende da especificação prévia do perfil das funções de pertinência. Em geral, define-se uma classe de funções simétricas, e.g., Gaussianas, para composição de conjuntos nebulosos. Seus parâmetros, valor modal e dispersão (ver capítulo 2), são determinados durante a aprendizagem da rede neural nebulosa. Outro tipo de limitação observada em redes neurais nebulosas é devido à prática comum da aprendizagem restringir-se a um único e limitado período de treinamento. As redes neurais nebulosas começam operar quando encerram o aprendizado. Embora esta estratégia seja muito útil ela não considera que o desempenho possa ser degradado se perturbações ou variações de parâmetros do sistema ocorram durante a operação da rede neural nebulosa. O método de aprendizagem proposto em [Nie95] tem como objetivo evitar este tipo de deficiência. Seu modelo de rede neural nebulosa, mesmo estando em operação, é capaz de alterar sua arquitetura sempre que não satisfizer o desempenho desejado. No entanto, não fornece funções de pertinência para cada variável.

As duas classes de redes neurais nebulosas (denominadas N1 e N2 respectivamente) apresentadas em seguida visam eliminar tais deficiências. Tanto N1 quanto N2 são baseadas no modelo apresentado em [Figueiredo95^b]. Em ambas propostas o conhecimento da rede é explicitamente codificado em sua estrutura na forma de regras nebulosas do tipo se-então.

A rede N1 apresenta diversas características [Figueiredo97^b]: 1) fácil inserção e extração de conhecimento na forma de regras lingüísticas nebulosas; 2) o processamento neural emula o mecanismo de raciocínio nebuloso; 3) as funções de pertinência de cada conjunto nebuloso são determinadas a partir dos dados de treinamento sem que nenhuma informação predefinida seja necessária; 4) regras são geradas automaticamente sem qualquer procedimento adicional; 5) a partição do espaço de entradas garante que as regras cubram todo o domínio de operação do modelo (resultado observado em simulação) e 6) N1 é um aproximador universal.

$$Z(x) = z_k, \text{ se } x \in I_k,$$

em que: $I_k = (x_{kI}, x_{kF}]$ é o k-ésimo intervalo de discretização, $k = 1, \dots, Q$.

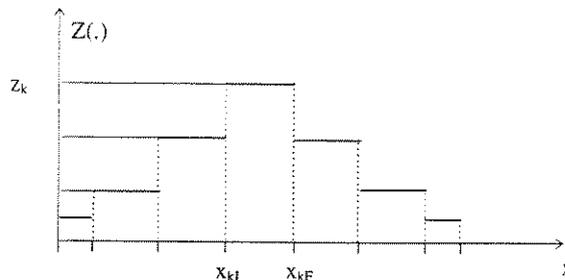


Figura 3.1: Função de pertinência discretizada $Z(\cdot)$ correspondente ao conjunto nebuloso Z .

Desta forma, define-se a_{jk}^i como sendo o valor da função de pertinência do conjunto A_j^i para todo $x_j \in I_{jk} = (x_{jkI}, x_{jkF}]$, ou seja:

$$A_j^i(x_j) = a_{jk}^i, \text{ se } x_j \in I_{jk} = (x_{jkI}, x_{jkF}]$$

Conforme capítulo anterior, o método adotado para obter a conclusão y , resultado da inferência nebulosa, consiste em uma seqüência de três estágios:

- 1) Comparação: Para cada regra i e cada antecedente j compute a medida de possibilidade P_j^i com relação aos conjuntos nebulosos A_j e A_j^i :

$$P_j^i = \sum_{k=1}^{Q_j} (T(a_{jk}, a_{jk}^i)), \quad (3.2)$$

em que: $\sum_{k=1}^{Q_j} (\)$ é computada tal qual em (2. 25) para todos os intervalos de discretização relativos à variável de entrada x_j e; Q_j é o número de intervalos I_{jk} associado à variável x_j .

- 2) Agregação de antecedentes: Para cada regra i calcule o nível de ativação H^i definido por:

$$H^i = \prod_{j=1}^M (P_j^i). \quad (3.3)$$

- 3) Agregação de regras: A saída y é computada tal qual em (3.4):

$$y = \frac{\sum_{i=1}^N H^i g^i(x)}{\sum_{i=1}^N H^i}. \quad (3.4)$$

A rede neural nebulosa N1 é constituída por neurônios nebulosos pertencentes a uma classe bastante geral, cujo modelo é baseado em conceitos inspirados na neurofisiologia (Figura 3.2). Embora os operadores sinápticos e de agregação tradicionalmente utilizados restrinjam-se a operações de produto e soma, sabe-se que as atividades do neurônio são modeladas com maior fidelidade por operadores mais complexos, em particular operadores do tipo t-norma e s-norma [Rocha92].

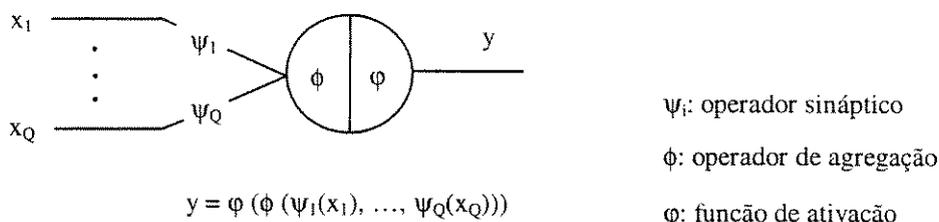


Figura 3.2: Modelo de neurônio nebuloso.

Uma arquitetura não recorrente com 5 camadas compõe a rede neural nebulosa N1 (Figura 3.3). A primeira camada é dividida em grupos de neurônios. Cada grupo corresponde a uma variável nebulosa de entrada. No caso de M entradas serem definidas, então há M grupos de neurônios na primeira camada. Cada neurônio nesta camada representa um intervalo de discretização $I_{jk} = (x_{jkl}, x_{jkF}]$ pertencente ao respectivo espaço de entrada. Há tantos neurônios em cada grupo quantos forem os intervalos de discretização existentes. Esta camada equivale à interface de entrada dos sistemas nebulosos. Cada grupo de neurônios é responsável por transformar as entradas não nebulosas para que as camadas seguintes possam tratar a informação adequadamente. Eles geram os valores a_{jk} que definem a função de pertinência do conjunto A_j . Após receberem um sinal de entrada x_j , os neurônios do j -ésimo grupo o decodifica e o transmite para a segunda camada. O sinal a_{jk} é transmitido pelo k -ésimo neurônio localizado no j -ésimo grupo. Definindo-se ψ e ϕ como funções identidade, a saída a_{jk} é dada por $a_{jk} = \phi(x_j)$. A função de decodificação $\phi(\cdot)$ definida para este neurônio é tal qual apresentada na Figura 3.4.

A segunda camada é formada por N grupos (número de regras nebulosas predefinido), cada qual com M neurônios (número de antecedentes de cada regra). Esta camada realiza o primeiro estágio da inferência nebulosa, a saber, comparação. O j -ésimo neurônio do i -ésimo grupo representa, através de seus pesos sinápticos a_{jk}^i , a função de pertinência do conjunto A_j^i (portanto, o i -ésimo grupo é composto por neurônios que representam as funções de pertinência de todos os conjuntos nebulosos correspondentes aos antecedentes da i -ésima regra). Além disso, ele calcula a comparação $P_j^i(x_j)$ entre os conjuntos A_j e o conjunto que representa A_j^i . Através de uma conexão cujo peso sináptico é a_{jk}^i , este neurônio recebe a saída a_{jk} do k -ésimo neurônio pertencente ao j -ésimo grupo da primeira camada. Considerando-se que $\psi = t$ -norma, $\phi = s$ -norma e $\phi =$ função identidade, a saída deste neurônio é $P_j^i(x_j)$ definido em (3.2).

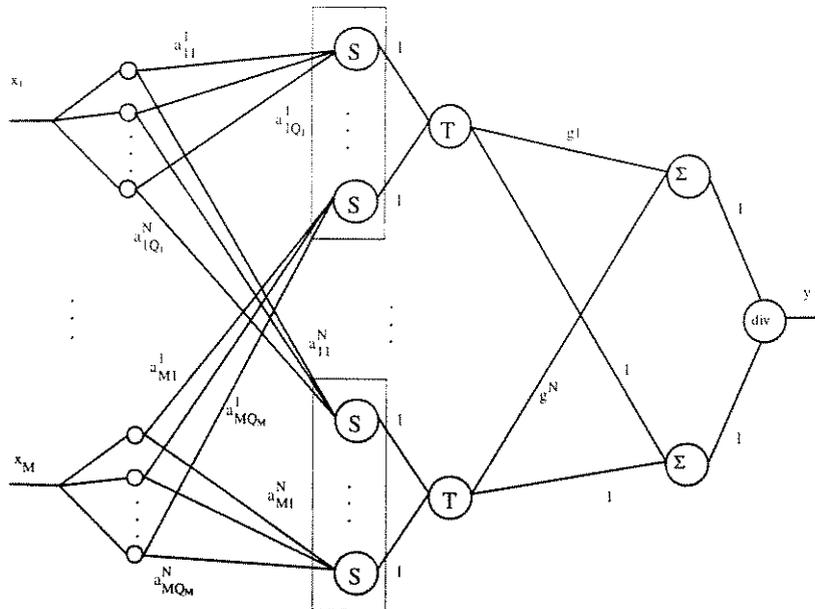
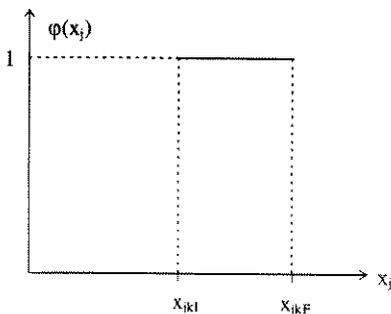


Figura 3.3: Rede neural nebulosa N1.

Há N neurônios na terceira camada. O i -ésimo neurônio está conectada a todos os neurônios do i -ésimo grupo da segunda camada. Todas as conexões possuem peso sináptico unitário. Ele realiza o estágio de agregação de antecedentes pois estão definidos: ψ = função identidade, ϕ = t-norma e φ = função identidade. Sua saída é $H^i(\mathbf{u})$ definida em (3.3).



$$\varphi(x_j) = \begin{cases} 1, & \text{se } x_j \in I_{jk} = (x_{jkI}, x_{jkF}]; \\ 0, & \text{caso contrário.} \end{cases}$$

Figura 3.4: Função de ativação dos neurônios da primeira camada.

São dois os neurônios na quarta camada, ambos conectados a todos os neurônios da terceira camada. Um dos neurônios conecta-se com o i -ésimo neurônio da terceira camada através de uma sinapse com peso $g^i(\mathbf{x})$. Seus operadores ψ e ϕ são respectivamente o produto algébrico e a soma algébrica e φ é a função identidade. Este neurônio computa o numerador de (3.4). As conexões do outro neurônio da quarta camada com os neurônios da terceira camada têm peso sináptico unitário. Considerando-se: ψ = função identidade, ϕ = soma algébrica e φ = função identidade, então sua saída equivale ao denominador da inferência nebulosa definida em (3.4). A última camada é constituída por um único neurônio que computa o valor de agregação de regras, ou seja, sua saída é $y(\mathbf{x}) = \frac{\sum_{i=1}^N H^i g^i}{\sum_{i=1}^N H^i}$. Portanto, suas conexões possuem pesos sinápticos unitários, ψ , ϕ e φ são a função identidade, o operador de divisão e a função identidade, respectivamente.

Cada uma das camadas da rede neural nebulosa em N1 possui uma correspondência com os diversos estágios de computação da inferência nebulosa descrita de (3.2) a (3.4), além de que os pesos sinápticos da rede representam tanto o conhecimento da rede neural como também representam as funções de pertinência dos antecedentes e os conseqüentes de regras nebulosas. Desta forma é possível visualizar N1 ou como sistemas nebulosos ou como uma rede neural.

3.2.2 Aprendizagem

A estratégia de aprendizagem da rede neural nebulosa N1 admite, em alguns de seus aspectos gerais, comparações com aquela utilizada pela rede do tipo contrapropagação [Hetch-Nielsen89] (seção (2.3)). O procedimento de aprendizagem se divide em duas fases. A primeira corresponde à aprendizagem não supervisionada e envolve somente uma parte da arquitetura, mais especificamente a segunda camada. A rede aprende, então, as funções de pertinência dos antecedentes das regras nebulosas enquanto ajusta seus pesos sinápticos. Tal como na rede do tipo contrapropagação o método utilizado é do tipo competitivo baseado nas redes do tipo Kohonen. A segunda fase adota uma técnica supervisionada para aprendizagem dos conseqüentes das regras e envolve apenas o ajuste de pesos da quarta camada.

Para aprendizagem da rede N1 considere disponível um conjunto de pares entrada/saída $\Gamma = \{(\mathbf{u}^1, v^1), \dots, (\mathbf{u}^r, v^r), \dots, (\mathbf{u}^R, v^R)\}$ em que $\mathbf{u}^r = (u_1^r \dots u_j^r \dots u_M^r)'$ é a entrada a ser apresentada à rede neural e v^r é a saída desejada como resposta apresentada pela rede para a entrada \mathbf{u}^r . Durante a primeira fase, não supervisionada, somente as entradas \mathbf{u}^r são apresentadas à N1.

Defina o vetor centro de área $\mathbf{B}^i = (b_1^i \dots b_j^i \dots b_M^i)'$, tal como em (3.5):

$$b_j^i = \left(\sum_{k=1}^{Q_j} a_{jk}^i \theta_{jk} \right) / \sum_{k=1}^{Q_j} a_{jk}^i ; \quad (3.5)$$

em que θ_{jk} é o centro do intervalo $(x_{jkl}, x_{jkF}]$ associado ao k-ésimo neurônio de um dos M grupos da primeira camada (Figura 3.4).

Note que, de acordo com a definição (3.5), b_j^i é o centro de área do conjunto nebuloso A_j^i (j-ésimo antecedente da i-ésima regra) e \mathbf{B}^i é o vetor centro de área com respeito a i-ésima regra, ou seja, correspondente ao i-ésimo grupo de neurônios da segunda camada.

Neurônios da segunda camada compõem N grupos (na Figura 3.3 eles estão envolvidos por um retângulo tracejado), tais quais definidos na seção anterior. Estes N grupos competem quando uma entrada é apresentada à rede N1. Para fins de aprendizagem eles são arranjados sobre um reticulado unidimensional. Cada grupo corresponde a uma posição no reticulado de modo que o conceito de vizinhança possa ser admitido entre grupos. Considere que $\mathbf{u}(t) = \mathbf{u}^r \in Z$, $r \in \{1, \dots, R\}$, seja a t-ésima entrada apresentada, t

$\in \mathbb{N} = \{1, 2, \dots\}$ (note que o argumento t também é utilizado para indicar valores de parâmetros da rede neural nebulosa NI quando a entrada $\mathbf{u}(t)$ é apresentada). O i -ésimo grupo vence a competição se a distância entre os vetores $\mathbf{B}^i(t)$ e $\mathbf{u}(t)$ é a menor entre todos os grupos. Somente os neurônios (da segunda camada) correspondentes aos grupos vizinhos do neurônio vencedor ajustam os pesos sinápticos $a_{jk}^i(t)$. Para ajustar $a_{jk}^i(t)$ determina-se para $j = 1, \dots, M$, os índices q e p tais que: o q -ésimo neurônio do grupo j da camada de entrada é ativado se a entrada é igual a $b_j^i(t)$ e o p -ésimo neurônio do grupo j da camada de entrada é ativado se a entrada é igual a $u_j(t)$. Portanto, considere que $a_{jk} = \varphi(x_j)$ seja a saída do k -ésimo neurônio do j -ésimo grupo da primeira camada e x_j seja sua entrada; então p e q são dados por (3.6):

Para se definir o índice q , considere $x_j = b_j^i(t)$ e para se definir o índice p , considere $x_j = u_j(t)$, desta forma, tem-se:

$$\text{Se } x_j = b_j^i(t) \rightarrow a_{jk} = \begin{cases} 1, & \text{se } k = q, \text{ ou seja, } b_j^i(t) \in I_{jq} = [x_{jqL}, x_{jqF}]; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.6)$$

$$\text{Se } x_j = u_j(t) \rightarrow a_{jk} = \begin{cases} 1, & \text{se } k = p, \text{ ou seja, } u_j(t) \in I_{jp} = [x_{jpL}, x_{jpF}]; \\ 0, & \text{caso contrário.} \end{cases}$$

Ajustes de $a_{jk}^i(t)$ dependem de p , q e k tal como segue (3.7):

1) $p \geq q$

$$a_{jk}^i(t+1) = \begin{cases} (1-\alpha)a_{jk}^i(t), & \text{se } k < q, & (1) \\ (1-\alpha)a_{jk}^i(t) + \alpha, & \text{se } q \leq k \leq p, p \neq q & (2) \\ a_{jk}^i(t), & \text{caso contrário.} & (3) \end{cases}$$

2) $p \leq q$ (3.7)

$$a_{jk}^i(t+1) = \begin{cases} (1-\alpha)a_{jk}^i(t), & \text{se } k < p, & (1) \\ (1-\alpha)a_{jk}^i(t) + \alpha, & \text{se } p \leq k \leq q, & (2) \\ a_{jk}^i(t), & \text{caso contrário.} & (3) \end{cases}$$

em que $\alpha = \alpha(t) = \eta(t)\pi(\xi(t), i, t)$, $\eta(t)$ é o fator de aprendizagem e $\pi(\xi(t), i, t)$ é a função de vizinhança.

A Figura 3.5 ilustra como são distribuídas as leis de ajuste (1), (2) e (3), conforme (3.7), a serem aplicadas sobre $a_{jk}^i(t)$ para cada intervalo I_{jk} , dados os intervalos I_{jp} e I_{jq} .

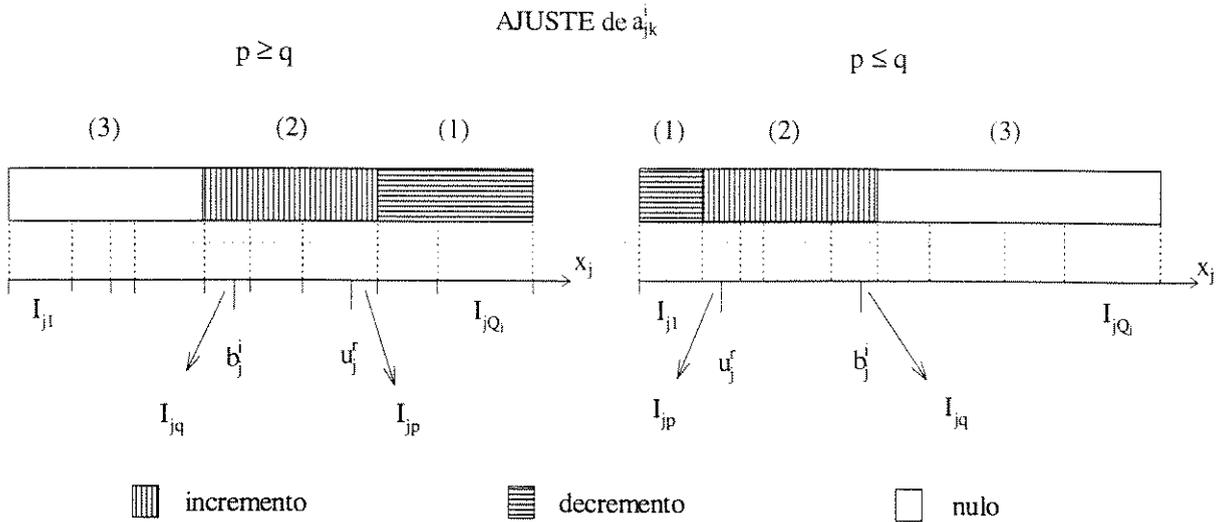


Figura 3.5: Ajuste de $a_{jk}^i(t)$, $k = 1, \dots, Q_j$, de acordo com os intervalos I_{jp} , I_{jq} e I_{jk} .

O algoritmo correspondente à primeira fase de aprendizagem é tal como segue:

1. Inicialização: Defina o número de regras nebulosas N . Escolha os valores para os pesos sinápticos $a_{jk}^i(1) \in [0, 1]$ de forma aleatória, preferivelmente considerando uma distribuição uniforme. Defina os modelos para a função vizinhança $\pi(\xi(t), i, t)$ e do fator de aprendizagem $\eta(t)$, ambos satisfazendo as condições de monotonicidade decrescente.
2. Estímulo: Apresente à rede neural um vetor $\mathbf{u}(t) \in \Gamma$, conjunto de vetores de entrada.
3. Competição: Compute as distâncias $D_i(t)$, $i = 1, \dots, N$, com relação à entrada $\mathbf{u}(t)$ e determine o grupo vencedor $\xi(t)$ para o qual a distância $D_i(t)$ é mínima:

$$D_i(t) = \|\mathbf{u}(t) - \mathbf{B}^i(t)\|,$$

$$\xi(t) = \arg \min_i (D_i(t)).$$

4. Ajuste e Atualização: Ajuste os pesos sinápticos $a_{jk}^i(t)$ apenas na vizinhança do neurônio vencedor de acordo com (3.7). Atualize $\alpha(t)$ em (3.7) bem como a função de vizinhança $\pi(t)$.
5. Avaliação de Convergência: Se os ajustes são menores que ε ou $t > \text{sup}$, ε e sup valores predefinidos então siga para 6, caso contrário retorne para 2.
6. Fim da Aprendizagem.

Durante esta fase as entradas são apresentadas e os pesos sinápticos são ajustados. O treinamento se encerra quando um critério de parada for satisfeito, e.g., fator de aprendizagem ou ajustes sinápticos menores que um valor mínimo ou número de iterações maior que valor máximo predefinidos.

Na fase seguinte a rede neural nebulosa aprende segundo uma estratégia supervisionada. Desde que as camadas 1, 2 e 3 da rede neural N1 já estão totalmente determinadas elas são utilizadas nesta fase da aprendizagem para computação dos valores da agregação de antecedentes $H^i(\cdot)$ (ativação da i -ésima regra nebulosa). Dados de entrada/saída (\mathbf{u}^r, v^r) são apresentados à rede neural nebulosa N1. Os conseqüentes das regras nebulosas, os pesos sinápticos $g^i(\mathbf{x})$, são então determinados.

Considere o conjunto G^i dado por (3.8):

$$G^i = \{r \in \{1, \dots, R\} \mid H^i(\mathbf{u}^r) \geq H^h(\mathbf{u}^r), \forall h = 1, \dots, N\}; \quad (3.8)$$

em que $H^i(\mathbf{u}^r)$ é tal qual em (3.3).

O conjunto G^i é formado pelos índices das entradas tais que o i -ésimo neurônio da terceira camada apresenta a maior saída ($H^i(\mathbf{u}^r)$). Baseado no conjunto G^i o peso sináptico $g^i(\mathbf{x})$ é definido por (3.9) (outros autores também definem conseqüentes como função das variáveis de entrada [Takagi85]):

$$g^i(\mathbf{x}) = (\text{MAX}^i - \text{STAR}^i) \exp(-\bar{\sigma}^i \|\bar{\mathbf{u}}^i - \mathbf{x}\|^2) + (\text{MIN}^i - \text{STAR}^i) \exp(-\underline{\sigma}^i \|\underline{\mathbf{u}}^i - \mathbf{x}\|^2) + \text{STAR}^i \quad (3.9)$$

em que:

$$\begin{aligned} \bar{\mathbf{u}}^i &= \mathbf{u}^s \mid v^s \geq v^r, s \in G^i \text{ e } \forall r \in G^i; \\ \text{MAX}^i &= v^r \mid \mathbf{u}^r = \bar{\mathbf{u}}^i; \\ \underline{\mathbf{u}}^i &= \mathbf{u}^s \mid v^s \leq v^r, s \in G^i \text{ e } \forall r \in G^i; \\ \text{MIN}^i &= v^r \mid \mathbf{u}^r = \underline{\mathbf{u}}^i; \\ \text{STAR}^i &= \frac{1}{\#G^i} \sum_{r \in G^i} v^r, \text{ em que } \#G^i \text{ é a cardinalidade de } G^i; \\ \bar{\sigma}^i \text{ e } \underline{\sigma}^i &\text{ são parâmetros.} \end{aligned}$$

Os pares $(\bar{\mathbf{u}}^i, \text{MAX}^i)$ e $(\underline{\mathbf{u}}^i, \text{MIN}^i)$ representam respectivamente os pontos de máximo e mínimo da função a ser aproximada considerando-se todas as entradas $\mathbf{u}^r, r \in G^i$. O valor de STAR^i é a média de todos os valores de v^r tais que $r \in G^i$.

Os valores de $\text{MAX}^i, \text{MIN}^i$ e STAR^i são determinados a partir dos dados disponíveis para aprendizagem de acordo com a definição em (3.9). O método de aprendizagem utilizado para ajuste dos parâmetros $\bar{\sigma}^i$ e $\underline{\sigma}^i$ é baseado no método do tipo gradiente. Para tanto, define-se a função índice de desempenho $E(\mathbf{u}(t))$ como a função erro quadrático: $E(\mathbf{u}(t)) = \frac{1}{2}(v(t) - y(t))^2$, em que $\mathbf{u}(t) = \mathbf{u}^r, r \in \{1, \dots, R\}$ e $v(t) = v^r$ de forma que $(\mathbf{u}^r, v^r) \in \Gamma$. Para cada par de entrada/saída $(\mathbf{u}(t), v(t)) = (\mathbf{u}^r, v^r) \in \Gamma$ o ajuste que incide sobre $\bar{\sigma}^i$ e $\underline{\sigma}^i$ é calculado de acordo com (3.10):

$$\sigma^i(t+1) = \sigma^i(t) - \mu(t) \partial E / \partial \sigma^i; \quad (3.10)$$

em que: $\sigma^i(t)$ representa $\bar{\sigma}^i$ ou $\underline{\sigma}^i$, indistintamente, e $\mu(t)$ é o fator de aprendizagem.

As derivadas parciais com respeito aos parâmetros $\bar{\sigma}^i$ ou $\underline{\sigma}^i$ são dadas por (3.11):

$$\begin{aligned} \partial E / \partial \bar{\sigma}^i &= (v(t) - y(t)) \bar{H}^i(\mathbf{u}(t)) \|\bar{\mathbf{u}}^i - \mathbf{u}(t)\|^2 (\text{MAX}^i - \text{STAR}^i) \exp(-\bar{\sigma}^i(t) \|\bar{\mathbf{u}}^i - \mathbf{u}(t)\|^2) \\ \partial E / \partial \underline{\sigma}^i &= (v(t) - y(t)) \bar{H}^i(\mathbf{u}(t)) \|\underline{\mathbf{u}}^i - \mathbf{u}(t)\|^2 (\text{MIN}^i - \text{STAR}^i) \exp(-\underline{\sigma}^i(t) \|\underline{\mathbf{u}}^i - \mathbf{u}(t)\|^2) \end{aligned} \quad (3.11)$$

em que: $y(t)$ é a saída da rede neural nebulosa N1 correspondente à entrada $\mathbf{u}(t)$ e

$$\bar{H}^i(\mathbf{u}(t)) = H^i(\mathbf{u}(t)) / \sum_{h=1}^N H^h(\mathbf{u}(t)).$$

Desta forma, o algoritmo relativo à segunda fase de aprendizagem é dado por:

1. Inicialização: Determine os valores de MAX^i , MIN^i e STAR^i analisando os dados disponíveis em Γ de acordo com a definição de $g^i(\mathbf{x})$ em (3.8) (as camadas 1, 2 e 3 podem auxiliar nesta tarefa). Escolha valores positivos para os parâmetros $\bar{\sigma}^i$ e $\underline{\sigma}^i$ aleatoriamente dentro de um intervalo predefinido conveniente. Defina o fator de aprendizagem $\mu(t)$.
2. Estímulo: Apresente à rede neural um par de entrada/saída $(\mathbf{u}(t), v(t))$ escolhido aleatoriamente pertencente ao conjunto Γ de dados.
3. Ajuste: Ajuste os parâmetros $\bar{\sigma}^i$ ou $\underline{\sigma}^i$ de acordo com (3.11).
4. Avaliação de Convergência: Repita os procedimentos a partir de 2 até que os ajustes sejam menores que ε ou $t > \text{sup}$, ε e sup definidos previamente.

Resumindo, a primeira fase da aprendizagem determina funções de pertinência correspondentes aos antecedentes de cada regra nebulosa. Elas estabelecem as regiões do espaço de entradas para o qual cada regra é disparada, ou seja, $H^i(\mathbf{x}) \neq 0$. A segunda fase determina o conseqüente $g^i(\mathbf{x})$ de cada regra i . A cada entrada apresentada à rede neural, o conseqüente $g^i(\mathbf{x})$ da regra i contribui para a resposta da rede com peso igual a $H^i(\mathbf{x})$. Quando uma única regra i é disparada a resposta da rede é igual ao conseqüente $g^i(\mathbf{x})$ calculado para a entrada apresentada. Portanto, pode-se estabelecer uma certa analogia da rede N1 com a rede do tipo contrapropagação. Ambas subdividem durante a primeira fase de aprendizagem o espaço de entradas em subconjuntos. Para cada subconjunto elas estabelecem uma aproximação da função representada pelo conjunto Γ . No caso da rede N1 esta aproximação é feita pela função $g^i(\mathbf{x})$. Note-se, ainda, que não é objetivo da estratégia competitiva adotada na primeira fase construir mapas topográficos sobre a estrutura da rede, mas sim realizar um particionamento do espaço de entradas. O conseqüente $g^i(\mathbf{x})$ é definido visando aproximar funções dentro de um subconjunto de seu domínio através de seus pontos de máximo ($\bar{\mathbf{u}}^i$, MAX^i) e de mínimo ($\underline{\mathbf{u}}^i$, MIN^i) neste subconjunto. Outras características de $g^i(\mathbf{x})$ são demonstradas na próxima seção. A discussão sobre a convergência do algoritmo está reservada à seção 3.2.4.

3.2.3 Aproximação

As características da rede neural nebulosa N1 permitem que ela seja analisada sob o ponto de vista de sistemas nebulosos. Esta sua característica dual serve de auxílio na demonstração do teorema 3.1 (apresentado nesta seção) no qual demonstra-se que a rede N1 é um aproximador universal. Inicialmente é importante constatar através do lema 3.1 algumas características de $g^i(\mathbf{x})$ definida na seção anterior.

Lema 3.1: O conseqüente $g^i(\mathbf{x})$ é limitado pelos seus parâmetros MAX^i e MIN^i :

$$MIN^i \leq g^i(\mathbf{x}) \leq MAX^i, \forall \mathbf{x} \in \mathfrak{R}^M. \quad (3.12)$$

Prova: A função $g^i(\mathbf{x})$ definida em (3.9) é reescrita (3.13):

$$g^i(\mathbf{x}) = (MAX^i - STAR^i) \exp(-\bar{\sigma}^i \|\bar{\mathbf{u}}^i - \mathbf{x}\|^2) + (MIN^i - STAR^i) \exp(-\underline{\sigma}^i \|\underline{\mathbf{u}}^i - \mathbf{x}\|^2) + STAR^i \quad (3.13)$$

Definindo-se $\alpha = \exp(-\bar{\sigma}^i (\bar{\mathbf{u}}^i - \mathbf{x})^2)$ e $\beta = \exp(-\underline{\sigma}^i (\underline{\mathbf{u}}^i - \mathbf{x})^2)$ vem (3.14):

$$g^i(\mathbf{x}) = \alpha(MAX^i - STAR^i) + \beta(MIN^i - STAR^i) + STAR^i. \quad (3.14)$$

Observando-se que $\forall \mathbf{x} \in \mathfrak{R}^M, 0 \leq \alpha, \beta \leq 1$ e $MIN^i \leq STAR^i \leq MAX^i$, obtém-se:

$$g^i(\mathbf{x}) \leq \alpha(MAX^i - STAR^i) + STAR^i$$

A demonstração vem por contradição. Suponha que $\alpha(MAX^i - STAR^i) + STAR^i > MAX^i$, então:

$(1 - \alpha)STAR^i > (1 - \alpha)MAX^i$, ou seja, $STAR^i > MAX^i$, o que é um absurdo, portanto:

$$g^i(\mathbf{x}) \leq MAX^i.$$

Pode-se aplicar a mesma técnica para a outra desigualdade em (3.12), concluindo-se que:

$$g^i(\mathbf{x}) \geq MIN^i. \quad \blacksquare$$

Seja $U \subset \mathfrak{R}^M$ um conjunto compacto em que $\mathbf{x} \in U$ é tal que $\mathbf{x} = (x_1 \dots x_j \dots x_M)'$. Por definição diz-se que $f: U \subset \mathfrak{R}^M \rightarrow \mathfrak{R}$ é uma função contínua se e somente se $\forall \mathbf{z} \in U$ e $\forall \varepsilon > 0$ então $\exists \delta'_j(\mathbf{z}) > 0, (\varepsilon, \delta'_j(\mathbf{z}) \in \mathfrak{R})$, tal que:

$$|x_j - z_j| < \delta'_j(\mathbf{z}) \Leftrightarrow |f(\mathbf{x}) - f(\mathbf{z})| < \varepsilon, j = 1, \dots, M.$$

Portanto, também $\exists \delta_j(\mathbf{z}) > 0, \delta_j(\mathbf{z}) \in \mathfrak{R}$, tal que:

$$(z_j - \delta_j(\mathbf{z})) < x_j \leq (z_j + \delta_j(\mathbf{z})) \Leftrightarrow |f(\mathbf{x}) - f(\mathbf{z})| < \varepsilon, j = 1, \dots, M. \quad (3.15)$$

Lema 3.2: Seja $f: U \subset \mathbb{R}^M \rightarrow \mathbb{R}$ uma função contínua definida no compacto U em que $\mathbf{x} \in U$ é tal que $\mathbf{x} = (x_1 \dots x_j \dots x_M)'$. Dado $\mathbf{z}^i \in U$ e $\varepsilon > 0$, de acordo com as características da função contínua estabelecidas em (3.15), seja $\delta_j(\mathbf{z}^i)$ tal que:

$$\left(z_j^i - \delta_j(\mathbf{z}^i)\right) < x_j \leq \left(z_j^i + \delta_j(\mathbf{z}^i)\right), \Leftrightarrow |f(\mathbf{x}) - f(\mathbf{z}^i)| < \varepsilon/2, j = 1, \dots, M.$$

Considere o conjunto $O(\mathbf{z}^i)$ tal como segue:

$$O(\mathbf{z}^i) = \left\{ \mathbf{x} \in U \mid \left(z_j^i - \delta_j(\mathbf{z}^i)\right) < x_j \leq \left(z_j^i + \delta_j(\mathbf{z}^i)\right), j = 1, \dots, M \right\}.$$

Se $g^i(\mathbf{x})$ é definida tal como em (3.16):

$$g^i(\mathbf{x}) = (\text{MAX}^i - \text{STAR}^i) \exp(-\bar{\sigma}^i \|\bar{\mathbf{u}}^i - \mathbf{x}\|^2) + (\text{MIN}^i - \text{STAR}^i) \exp(-\underline{\sigma}^i \|\underline{\mathbf{u}}^i - \mathbf{x}\|^2) + \text{STAR}^i \quad (3.16)$$

em que: $\text{STAR}^i = f(\mathbf{z}^i)$;
 $\text{MAX}^i = f(\bar{\mathbf{u}}^i), \bar{\mathbf{u}}^i = \mathbf{u}, \mathbf{u} \in O(\mathbf{z}^i) \mid f(\mathbf{u}) \geq \text{STAR}^i$;
 $\text{MIN}^i = f(\underline{\mathbf{u}}^i), \underline{\mathbf{u}}^i = \mathbf{u}, \mathbf{u} \in O(\mathbf{z}^i) \mid f(\mathbf{u}) \leq \text{STAR}^i$;
 $\bar{\sigma}^i, \underline{\sigma}^i > 0$

Então $|f(\mathbf{x}) - g^i(\mathbf{x})| < \varepsilon, \forall \mathbf{x} \in O(\mathbf{z}^i)$. (3.17)

Prova: Considerando-se o Lema 3.1, pelo menos uma das desigualdades é verdadeira:

- 1) $|f(\mathbf{x}) - g^i(\mathbf{x})| \leq |f(\mathbf{x}) - \text{MAX}^i|$, ou
- 2) $|f(\mathbf{x}) - g^i(\mathbf{x})| \leq |f(\mathbf{x}) - \text{MIN}^i|$

Analisando-se o primeiro caso, tem-se:

$$|f(\mathbf{x}) - g^i(\mathbf{x})| \leq |f(\mathbf{x}) - \text{MAX}^i| \leq |f(\mathbf{x}) - f(\mathbf{z}^i) + f(\mathbf{z}^i) - \text{MAX}^i| \leq |f(\mathbf{x}) - f(\mathbf{z}^i)| + |f(\mathbf{z}^i) - \text{MAX}^i|.$$

Considerando-se que $\text{MAX}^i = f(\bar{\mathbf{u}}^i)$ e $\{\bar{\mathbf{u}}^i, \mathbf{z}^i\} \subset O(\mathbf{z}^i)$ então:

$$\begin{aligned} |f(\mathbf{x}) - g^i(\mathbf{x})| &\leq |f(\mathbf{x}) - f(\mathbf{z}^i)| + |f(\mathbf{z}^i) - f(\bar{\mathbf{u}}^i)| < \varepsilon/2 + \varepsilon/2 \text{ e} \\ |f(\mathbf{x}) - g^i(\mathbf{x})| &< \varepsilon. \end{aligned}$$

Utilizando-se de raciocínio análogo este mesmo resultado é alcançado para o segundo caso, finalizando a demonstração do lema com a confirmação de (3.17). ■

Observe que o Lema 3.2 também vale para a definição de $g^i(\mathbf{x})$ em (3.9) pois as relações de ordem entre MAX^i , MIN^i e $g^i(\mathbf{x})$ se mantém. Além disso, considerando-se que $f(\mathbf{x})$ é contínua então sempre existe $\mathbf{z} \in U$ tal que $f(\mathbf{z}) = \gamma$, para qualquer γ situado entre o máximo e o mínimo da função [Courant79].

Teorema 3.1: Seja $f: U \subset \mathfrak{R}^M \rightarrow \mathfrak{R}$ uma função contínua definida no compacto U em que $\mathbf{x} \in U$ é tal que $\mathbf{x} = (x_1 \dots x_j \dots x_M)'$. Considere as regras nebulosas e o mecanismo de inferência dos tipos definidos na seção 3.2.1 nos quais está baseada a arquitetura e o processamento do raciocínio aproximado da rede neural nebulosa N1. Então existe um conjunto de regras nebulosas tal que para qualquer $\mathbf{x} \in U$ e $\varepsilon > 0$ a expressão (3.18) é satisfeita.

$$\sup\{|f(\mathbf{x}) - y| \mid \mathbf{x} \in U\} \leq \varepsilon, \quad (3.18)$$

em que y é a resposta da rede N1 (conclusão da inferência nebulosa) para a entrada \mathbf{x} .

Prova: A demonstração do teorema será do tipo construtivo, ou seja, a demonstração consiste em exibir regras nebulosas (funções de pertinência de antecedentes e conseqüentes) tais que a tese (3.18) seja satisfeita para a inferência nebulosa definida para a rede neural N1.

Desde que $f(\cdot)$ é contínua em $\mathbf{z} \in U$, então, para cada $j, j = 1, \dots, M$, existe $\delta_j(\mathbf{z}) > 0$ tal que:

$$(z_j - \delta_j(\mathbf{z})) < x_j \leq (z_j + \delta_j(\mathbf{z})), \Leftrightarrow |f(\mathbf{x}) - f(\mathbf{z})| < \varepsilon/2, j = 1, \dots, M.$$

Para cada $\mathbf{z} \in U$ defina o conjunto $O(\mathbf{z})$ tal como segue:

$$O(\mathbf{z}) = \left\{ \mathbf{x} \in U \mid (z_j - \delta_j(\mathbf{z})) < x_j \leq (z_j + \delta_j(\mathbf{z})), j = 1, \dots, M \right\}.$$

Considerando-se $O(\mathbf{z})$ assim definido resulta:

$$U \subseteq \bigcup_{\mathbf{z} \in U} O(\mathbf{z}).$$

Desde que U é compacto existe uma subfamília finita $O(\mathbf{z}^i)$, $i = 1, \dots, N$, tal que [Castro95]:

$$U \subseteq \bigcup_{i=1}^N O(\mathbf{z}^i). \quad (3.19)$$

A partir do conjunto $O(\mathbf{z}^i)$ defina conjuntos $O_j^i, j = 1, \dots, M$, como segue:

$$O_j^i = \left\{ x_j \in \mathfrak{R} \mid (z_j^i - \delta_j(\mathbf{z}^i)) < x_j \leq (z_j^i + \delta_j(\mathbf{z}^i)) \right\}. \quad (3.20)$$

Portanto, $\mathbf{x} \in O(\mathbf{z}^i)$ se e somente se $\forall j, j = 1, \dots, M, x_j \in O_j^i$.

Baseando-se nos conjuntos O_j^i defina os conjuntos nebulosos A_j^i de forma que as seguintes restrições sejam satisfeitas (3.21):

$$A_j^i(x_j) = \begin{cases} a > 0, & \text{se } x_j \in O_j^i; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.21)$$

Também utilize os conjuntos $O(z^j)$ para definir os parâmetros de cada conseqüente $g^i(x)$ tal como em (3.22):

$$g^i(x) = (\text{MAX}^i - \text{STAR}^i) \exp(-\bar{\sigma}^i \| \bar{\mathbf{u}}^i - \mathbf{x} \|^2) + (\text{MIN}^i - \text{STAR}^i) \exp(-\underline{\sigma}^i \| \underline{\mathbf{u}}^i - \mathbf{x} \|^2) + \text{STAR}^i \quad (3.22)$$

$$\begin{aligned} \text{STAR}^i &= f(\mathbf{z}^i); \\ \text{MAX}^i &= f(\bar{\mathbf{u}}^i), \bar{\mathbf{u}}^i = \mathbf{u} \in O(\mathbf{z}^i) \mid f(\mathbf{u}) \geq \text{STAR}^i; \\ \text{MIN}^i &= f(\underline{\mathbf{u}}^i), \underline{\mathbf{u}}^i = \mathbf{u} \in O(\mathbf{z}^i) \mid f(\mathbf{u}) \leq \text{STAR}^i; \\ \bar{\sigma}^i, \underline{\sigma}^i &> 0 \end{aligned}$$

Defina intervalos de discretização $I_{jk} = (x_{jkL}, x_{jkF}]$, $k = 1, \dots, Q_j$, de maneira que todos os conjuntos O_j^i , $i = 1, \dots, N$; $j = 1, \dots, M$, possam ser representados por tais intervalos I_{jk} , ou seja, duas condições devem ser satisfeitas:

$$1) \text{ Se } x_j \in O_j^i \Rightarrow \exists k \mid x_j \in I_{jk} \text{ e } I_{jk} \subset O_j^i. \quad (3.23)$$

$$2) \text{ Se } \exists x_j \mid x_j \in I_{jk} \text{ e } x_j \in O_j^i \Rightarrow I_{jk} \subset O_j^i.$$

3) Os intervalos I_{jk} são disjuntos.

Uma possível definição dos intervalos de discretização considera todos os pontos que definem os extremos dos intervalos O_j^i , $i = 1, \dots, N$, ou seja, os intervalos I_{jk} seriam definidos a partir dos pontos $(z_j^i - \delta_j(z^i))$ e $(z_j^i + \delta_j(z^i))$. Conseqüentemente, considerando-se (3.21) e a discretização proposta, tem-se:

$$A_j^i(x) = \begin{cases} a_{jk}^i > 0, & \text{se } x_j \in I_{jk} \subset O_j^i; \\ a_{jk}^i = 0, & \text{se } x_j \in I_{jk} \not\subset O_j^i. \end{cases} \quad (3.24)$$

Considere que $\mathbf{x} \in U$ seja apresentado à rede neural nebulosa N1. Então as seguintes conclusões podem ser inferidas:

1) Considerando-se (3.19), então $\exists \Lambda \neq \emptyset$, $\Lambda = \{i \mid \mathbf{x} \in O(\mathbf{z}^i), i = 1, \dots, N\}$ e para tais valores de $i \in \Lambda$, O_j^i é tal que $x_j \in O_j^i$;

2) A partir de (3.20), (3.23) e (3.24) infere-se que se $i \in \Lambda$ então $\exists k = k_j^*$, $j = 1, \dots, M$, tal que $x_j \in I_{jk_j^*} \subset O_j^i$, e, portanto, $a_{jk_j^*}^i \neq 0$; e se $i \notin \Lambda \exists j \mid I_{jk_j^*} \not\subset O_j^i$ e $a_{jk_j^*}^i = 0$;

3) As saídas a_{jk} dos neurônios do j -ésimo grupo da primeira camada são tais que (Figura 3.4):

$$4) a_{jk} = \begin{cases} 1, & k = k_j^*; \\ 0, & \text{caso contrário.} \end{cases}$$

Computando-se os estágios de comparação e agregação de antecedentes tal qual em (3.2) e (3.3), considerando-se as propriedades das t-normas e s-normas ($T(1, a) = a$, $T(0, a) = 0$, $S(a, 1) = 1$ e $S(a, 0) = a$) e levando-se em conta as conclusões anteriores verifica-se que:

- 1) $P_j^i(x_j) = \prod_{k=1}^{Q_j} (T(a_{jk}, a_{jk}^i)) = a_{jk_j^*}^i$, pois $a_{jk_j^*}^i = 1$ e $a_{jk} = 0$, $\forall k \neq k_j^*$ (k_j^* definido acima);
- 2) $\forall i \in \Lambda, H^i(\mathbf{x}) = \prod_{j=1}^M (P_j^i(x_j)) \neq 0$, pois se $i \in \Lambda$ então $P_j^i(x_j) \neq 0$, $j = 1, \dots, M$;
- 3) $\forall i \notin \Lambda, H^i(\mathbf{x}) = \prod_{j=1}^M (P_j^i(x_j)) = 0$, pois se $i \notin \Lambda$ então $\exists j \mid I_{jk_j^*} \not\subset O_j^i$ e $a_{jk_j^*}^i = 0$.

Por esta razão, participam efetivamente do cômputo da agregação de regras (último estágio da inferência nebulosa) somente os conseqüentes $g^i(\mathbf{x})$ tais que $i \in \Lambda$.

O erro de aproximação $E(\mathbf{x})$ entre o resultado da inferência nebulosa $y = \frac{\sum_{i \in \Lambda} H^i(\mathbf{x})g^i(\mathbf{x})}{\sum_{i \in \Lambda} H^i(\mathbf{x})}$ e a função $f(\mathbf{x})$ pode ser então calculado tal como segue:

$$E(\mathbf{x}) = |y - f(\mathbf{x})| = \left| \frac{\sum_{i \in \Lambda} H^i(\mathbf{x})(g^i(\mathbf{x}) - f(\mathbf{x}))}{\sum_{i \in \Lambda} H^i(\mathbf{x})} \right| = \left| \frac{\sum_{i \in \Lambda} H^i(\mathbf{x})(g^i(\mathbf{x}) - f(\mathbf{x}))}{\sum_{i \in \Lambda} H^i(\mathbf{x})} \right|.$$

Aplicando-se a propriedade da desigualdade triangular sobre o numerador do lado direito da última expressão e reescrevendo, resulta:

$$E(\mathbf{x}) \leq \frac{\sum_{i \in \Lambda} |H^i(\mathbf{x})(g^i(\mathbf{x}) - f(\mathbf{x}))|}{\sum_{i \in \Lambda} H^i(\mathbf{x})} = \frac{\sum_{i \in \Lambda} |H^i(\mathbf{x})| |g^i(\mathbf{x}) - f(\mathbf{x})|}{\sum_{i \in \Lambda} H^i(\mathbf{x})}.$$

Considerando-se que a imagem das funções t-norma são limitadas ao intervalo unitário, a desigualdade pode ser simplificada tal como segue:

$$E(\mathbf{x}) \leq \frac{\sum_{i \in \Lambda} H^i(\mathbf{x}) |g^i(\mathbf{x}) - f(\mathbf{x})|}{\sum_{i \in \Lambda} H^i(\mathbf{x})}.$$

Finalmente, utilizando-se o Lema 3.2, conclui-se a tese (3.18):

$$E(\mathbf{x}) < \frac{\sum_{i \in \Lambda} H^i(\mathbf{x})\epsilon}{\sum_{i \in \Lambda} H^i(\mathbf{x})} = \epsilon \quad \blacksquare$$

O Teorema 3.1 também aplica-se à definição de $g^i(\mathbf{x})$ em (3.9). As argumentações são as mesmas apresentadas com relação ao Lema 3.2, ou seja, MAX^i , MIN^i e $g^i(\mathbf{x})$ mantêm suas posições relativas e a continuidade da função $f(\mathbf{x})$ garante que para qualquer γ situado entre o máximo e o mínimo da função então sempre existe $\mathbf{z} \in U$ tal que $f(\mathbf{z}) = \gamma$ [Courant79].

3.2.4 Convergência

O fator de aprendizagem $\alpha(t)$ empregado no ajuste dos pesos sinápticos $a_{jk}^i(t)$ em (3.7) garante a convergência do método não supervisionado para a rede neural nebulosa N1. Portanto, $\eta(t)$ deve ser uma função de t monotônica decrescente positiva, por exemplo, uma função com decaimento exponencial (2.13) [Haykin94].

A convergência da fase supervisionada é garantida caso defina-se $E(\mathbf{w}) = \frac{1}{2} \sum_{r=1}^R (v^r - y^r)^2$ e ajuste-se os pesos sinápticos após a apresentação de todos os pares de dados $(\mathbf{u}^r, v^r) \in \Gamma$. Além disso, o fator de aprendizagem $\mu(t)$ deve ser escolhido convenientemente. Para cada iteração t o valor de $\mu(t)$ deve ser tal que minimize o valor do índice de desempenho $E(t)$, ou seja, há necessidade de uma busca unidimensional na direção do gradiente. Neste caso, garante-se que a solução encontrada converge para um ponto de mínimo [Zuben96] (seção 3.2.2). Optou-se pelo método descrito na seção 3.2.2 pois seu custo computacional é menor e a técnica de ajuste a cada entrada apresentada também apresenta bons resultados sendo frequentemente empregada e referenciada [Haykin94].

Para ajuste dos pesos sinápticos $a_{jk}^i(t)$ nenhuma restrição foi imposta no que se refere à formação de funções de pertinência com perfis previamente especificados. É possível se obter resultados teóricos sobre o perfil encontrado pela rede neural N1 após a primeira fase da aprendizagem ter se encerrado. Para tanto considera-se as funções $A_j^i(\cdot)$ na sua forma não discretizada e que as entradas sejam apresentadas a rede N1 segundo uma distribuição de probabilidades conhecida. Admitindo-se estas hipóteses, demonstra-se no Teorema 3.2 que tais funções convergem de modo a adquirir perfis triangulares.

Os vetores centro de área $\mathbf{B}^i = (b_1^i \dots b_j^i \dots b_M^i)'$ definidos para o caso contínuo são tais que (3.25):

$$b_j^i = \int x_j A_j^i(x_j) dx_j / \int A_j^i(x_j) dx_j ; \quad (3.25)$$

em que os operadores de integração são aplicados para x_j pertencente ao respectivo universo de discurso.

As leis de ajuste sináptico relativas ao caso contínuo são análogas ao caso discreto descritas em (3.6). Para cada entrada $\mathbf{z}(t) \in U \subset \mathfrak{R}^M$, $\mathbf{z}(t) = (z_1^i \dots z_j^i \dots z_M^i)'$, apresentada a rede neural N1, todos os grupos i pertencentes à vizinhança do grupo vencedor ajustam seus pesos sinápticos $A_j^i(x_j)(t)$ de acordo com as seguintes equações de ajuste (3.26):

$$1) z_j \geq b_j^i$$

$$A_j^i(x_j)(t+1) = \begin{cases} (1-\alpha)A_j^i(x_j)(t), & \text{se } z_j < x_j, \\ (1-\alpha)A_j^i(x_j)(t) + \alpha, & \text{se } b_j^i \leq x_j \leq z_j, b_j^i \neq z_j \\ A_j^i(x_j)(t), & \text{caso contrário.} \end{cases} \quad (1)$$

$$2) z_j \leq b_j^i \quad (3.26)$$

$$A_j^i(x_j)(t+1) = \begin{cases} (1-\alpha)A_j^i(x_j)(t), & \text{se } x_j < z_j, \\ (1-\alpha)A_j^i(x_j)(t) + \alpha, & \text{se } z_j \leq x_j \leq b_j^i, \\ A_j^i(x_j)(t), & \text{caso contrário.} \end{cases} \quad (1)$$

Defina $C^i \subset U \subset \mathfrak{R}^M$ como o subconjunto do espaço de entradas tal que se $\mathbf{z}(t) \in C^i$, então o i -ésimo grupo da segunda camada torna-se o grupo vencedor. Seja $[u_{jI}, u_{jF}]$ o universo de discurso em que as funções de pertinência com relação à j -ésima dimensão são definidas e $C_j^i = [c_{ijI}, c_{ijF}] \subset [u_{jI}, u_{jF}]$ constituído apenas pelas j -ésimas componentes de todos $\mathbf{z}(t) \in C^i$. Na Figura 3.6 ilustra-se os três intervalos para $z_j \in C_j^i$ de acordo com os quais $A_j^i(x_j)(t)$, $x_j \leq b_j^i$ é ajustado (veja (3.26)). De maneira análoga é possível representar estes intervalos para os casos em que $A_j^i(x_j)(t)$, $x_j \geq b_j^i$.

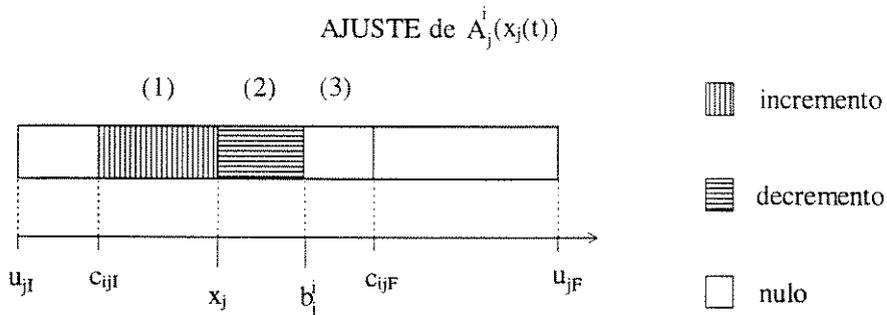


Figura 3.6: Ajuste de $A_j^i(x_j)(t)$, $x_j \leq b_j^i$, de acordo com três intervalos para $z_j \in C_j^i$.

Se $f_{ij}(z_j)$ é uma função densidade de probabilidade uniforme para $z_j \in C_j^i$, então:

$$f_{ij}(z_j) = \begin{cases} 1/(c_{ijF} - c_{ijI}), & \text{se } z_j \in [c_{ijI}, c_{ijF}] \\ 0, & \text{caso contrário.} \end{cases} \quad (3.27)$$

Portanto, a probabilidade $P_j^i(x_{js}, x_{jr})$ de $z_j \in [x_{js}, x_{jr}] \subset C_j^i$ pode ser calculada tal como segue:

$$P_j^i(x_{js}, x_{jr}) = \int_{x_{js}}^{x_{jr}} f_{ij}(z_j) dz_j = \int_{x_{js}}^{x_{jr}} \left(1/(c_{ijI} - c_{ijF})\right) dz_j = (x_{jr} - x_{js}) / (c_{ijI} - c_{ijF}). \quad (3.28)$$

Considere b_j^i fixo $\forall t \in \mathbb{N} = \{1, 2, \dots\}$, $b_j^i \in (c_{ijI}, c_{ijF})$ e $x_j \leq b_j^i$. Neste caso, defina $\tau^+ = \{t \in \mathbb{N} / z_j \in C_j^i, z_j \in [c_{ijI}, x_j]\}$ e $\tau^- = \{t \in \mathbb{N} / z_j \in C_j^i, z_j \in (x_j, b_j^i]\}$ (para simplificar a notação z_j não foi indexado pela iteração t), conseqüentemente:

$$A_j^i(x_j)(t+1) = \begin{cases} (1-\alpha)A_j^i(x_j)(t), & \text{se } t \in \tau^-, \\ (1-\alpha)A_j^i(x_j)(t) + \alpha, & \text{se } t \in \tau^+, \\ A_j^i(x_j)(t), & \text{se } t \notin \tau^+ \text{ e } t \notin \tau^-. \end{cases} \quad (3.29)$$

Suponha que $\overline{A_j^i(x_j)}$ seja o valor médio de $A_j^i(x_j)(t)$ para $t \in \mathbb{N}$, $t \in \tau^+$ e $t \in \tau^-$, quando $t \rightarrow \infty$, ou seja:

$$\lim_{T \rightarrow \infty} (1/T) \sum_{t=1}^T A_j^i(x_j)(t) = \lim_{t \rightarrow \infty} (1/T^+) \sum_{t \in \tau^+} A_j^i(x_j)(t) = \lim_{t \rightarrow \infty} (1/T^-) \sum_{t \in \tau^-} A_j^i(x_j)(t) = \overline{A_j^i(x_j)} \quad (3.30)$$

em que T^+ e T^- são as cardinalidades dos conjuntos τ^+ e τ^- , respectivamente.

Neste caso, se $\Delta_j^i(x_j)(t)$ é definido como o ajuste de $A_j^i(x_j)(t)$, então $\Delta_j^i(x_j)(t) = A_j^i(x_j)(t+1) - A_j^i(x_j)(t)$, e:

$$L = \lim_{T \rightarrow \infty} (1/T) \sum_{t=1}^T \Delta_j^i(x_j)(t) = 0. \quad (3.31)$$

Teorema 3.2: Considere que entradas $\mathbf{z}(t) \in U \subset \mathfrak{R}^M$ sejam apresentadas à rede NI para sua aprendizagem de forma que a função densidade de probabilidade condicional com relação aos $z_j \in C_j^i$ seja do tipo uniforme. Considere que os centros de área $\mathbf{B}^i \in C^i$, $b_j^i \in (c_{ijI}, c_{ijF})$, sejam constantes durante toda a aprendizagem e que o valor médio de $A_j^i(x_j)(t)$ seja $\overline{A_j^i(x_j)}$ tal como em (3.30). Então, para qualquer α , $0 < \alpha < 1$, os valores médios de $A_j^i(x_j)(t)$, $i = 1, \dots, N$, $j = 1, \dots, M$, adquirem perfis triangulares.

Prova: Inicialmente considere $x_j \leq b_j^i$. A média dos ajustes $\Delta_j^i(x_j)(t)$ pode ser calculada tal como segue:

$$L = \lim_{T \rightarrow \infty} (1/T) \left(\sum_{t \in \tau^+} \Delta_j^i(x_j)(t) + \sum_{t \in \tau^-} \Delta_j^i(x_j)(t) \right). \quad (3.32)$$

De acordo com a definição para $\Delta_j^i(x_j)(t)$ e (3.29) tem-se:

$$\Delta_j^i(x_j) = \begin{cases} (1 - A_j^i(x_j)(t))\alpha, & \text{se } t \in \tau^+; \\ -A_j^i(x_j)(t)\alpha, & \text{se } t \in \tau^-. \end{cases}$$

Substituindo-se valores em (3.32) de acordo com as definições dos conjuntos τ^+ e τ^- , e considerando-se suas cardinalidades (T^+ e T^- , respectivamente), resulta:

$$L = \lim_{T \rightarrow \infty} (1/T) \left((T^+/T^+) \sum_{t \in \tau^+} \left((1 - A_j^i(x_j)(t))\alpha \right) + (T^-/T^-) \sum_{t \in \tau^-} \left(-A_j^i(x_j)(t)\alpha \right) \right). \quad (3.33)$$

Tomando-se a primeira parcela na expressão do limite em (3.33), vem:

$$L_a = \lim_{T \rightarrow \infty} (1/T) \left((T^+/T^+) \sum_{t \in \tau^+} \left((1 - A_j^i(x_j)(t))\alpha \right) \right) = \lim_{T \rightarrow \infty} (T^+/T) \left((1/T^+) \sum_{t \in \tau^+} \left((1 - A_j^i(x_j)(t))\alpha \right) \right) \quad (3.34)$$

Através de (3.28) sabe-se que $\lim_{T \rightarrow \infty} (T^+/T) = P_j^i(c_{ijl}, x_j)$, pois a probabilidade da entrada x_j pertencer a $[c_{ijl}, x_j]$ é igual a probabilidade de t pertencer a τ^+ . Considerando-se (3.30), pode-se reescrever (3.34) distribuindo-se o operador $\lim_{T \rightarrow \infty}$ para fatores e parcelas, tal como segue:

$$L_a = \lim_{T \rightarrow \infty} (T^+/T) \left(\left(\lim_{T \rightarrow \infty} (1/T^+) \sum_{t \in \tau^+} \alpha \right) - \left(\lim_{T \rightarrow \infty} (1/T^+) \sum_{t \in \tau^+} (A_j^i(x_j)(t)\alpha) \right) \right).$$

Substituindo-se valores na expressão anterior resulta:

$$L_a = P_j^i(c_{ijl}, x_j) \left(\alpha - \overline{A_j^i(x_j)} \alpha \right).$$

Operando-se de forma análoga para a segunda parcela do limite em (3.33), tem-se que:

$$\lim_{T \rightarrow \infty} (1/T) \left((T^-/T^-) \sum_{t \in \tau^-} \left(-A_j^i(x_j)(t)\alpha \right) \right) = -\alpha P_j^i(x_j, b_j^i) \overline{A_j^i(x_j)}$$

Portanto, (3.32) pode ser simplificada para:

$$\lim_{T \rightarrow \infty} (1/T) \sum_{t=1}^T \Delta_j^i(x_j)(t) = \alpha P_j^i(c_{ijt}, x_j) - \alpha P_j^i(c_{ijt}, x_j) \overline{A_j^i(x_j)} - \alpha P_j^i(x_j, b_j^i) \overline{A_j^i(x_j)}. \quad (3.35)$$

Considerando-se (3.31) pode-se equacionar (3.35) da seguinte forma:

$$\alpha P_j^i(c_{ijt}, x_j) - \alpha P_j^i(c_{ijt}, x_j) \overline{A_j^i(x_j)} - \alpha P_j^i(x_j, b_j^i) \overline{A_j^i(x_j)} = 0,$$

o qual, resolvendo-se para $\overline{A_j^i(x_j)}$, resulta:

$$\overline{A_j^i(x_j)} = P_j^i(c_{ijt}, x_j) / (P_j^i(c_{ijt}, x_j) + P_j^i(x_j, b_j^i)). \quad (3.36)$$

A expressão (3.36) pode ser reescrita substituindo-se valores de acordo com (3.28). Conseqüentemente o valor de $\overline{A_j^i(x_j)}$ é dado por:

$$\overline{A_j^i(x_j)} = (c_{ijt} - x_j) / (c_{ijt} - b_j^i), \quad x_j \leq b_j^i. \quad (3.37)$$

Procedendo-se de maneira semelhante para $x_j > b_j^i$ (inclusive redefinindo-se conjuntos τ^+ e τ^- em (3.29)) conclui-se:

$$\overline{A_j^i(x_j)} = (c_{ijF} - x_j) / (c_{ijF} - b_j^i), \quad x_j > b_j^i. \quad (3.38)$$

As expressões (3.37) e (3.38) representam equações de retas que interceptam-se em $x_j = b_j^i$, neste caso $\overline{A_j^i(x_j)} = 1$, e possuem zeros respectivamente em $x_j = c_{ijt}$ e $x_j = c_{ijF}$, caracterizando perfis triangulares para $\overline{A_j^i(x_j)}$, $i = 1, \dots, N$, $j = 1, \dots, M$. ■

3.2.5 Resultados de Simulação

Nesta seção, as características da rede neural nebulosa N1 são demonstradas quando aplicada em aproximação de funções. São apresentados resultados de simulação e comparações com outras abordagens. Também são apresentados experimentos para ilustrar a capacidade de aprendizagem da rede N1 no que se refere à formação de perfis para as funções de pertinência $A_j^i(\cdot)$ bem como a capacidade de aproximação da função $g^i(\mathbf{x})$.

Em todos os experimentos, os universos de discurso correspondentes a cada dimensão são discretizados em 30 intervalos. O número de regras codificadas pela rede neural é definido

caso a caso. A inicialização de outros parâmetros relativos à arquitetura da rede neural N1 e/ou de seu método de aprendizagem são especificados tal como segue.

Os centros de área $b_j^i(t)$ são inicializados ($t = 0$) escolhendo-se valores aleatoriamente em seus respectivos universos de discurso. As funções de pertinência são inicializadas da seguinte forma:

$$a_{jk}^i(0) = \begin{cases} 0 & \text{se } k \neq q; \\ 0,1 & \text{se } k = q. \end{cases}$$

Conforme já visto em (3.7), o fator $\alpha(t)$ é resultado do produto de outros dois fatores: o fator de aprendizagem $\eta(t)$ e a função de vizinhança $\pi(\xi(t), i, t)$, ou seja:

$$\alpha(t) = \eta(t)\pi(\xi(t), i, t).$$

O fator de aprendizagem $\eta(t)$ é inicializado em $\eta(0) = 0,5$ e sua dinâmica obedece a seguinte equação recorrente:

$$\eta(t + 1) = \eta(t) - 0.0001 \eta(t).$$

A função de vizinhança $\pi(\xi(t), i, t)$ é definida de acordo com os valores da vizinhança $\delta(t)$. Para t tal que $\delta(t) \geq 1$, $\pi(\xi(t), i, t)$ é definida por:

$$\pi(\xi(t), i, t) = \begin{cases} 1 / (1 + \sqrt{|\xi(t) - i|}) & \text{se } |\xi(t) - i| \leq \delta(t); \\ 0, & \text{se } |\xi(t) - i| > \delta(t) \end{cases}$$

em que: $\delta(t+1) = 50/(t+1)$ e $\xi(t)$ é o neurônio vencedor para a entrada $u(t)$.

Para t tal que $\delta(t) < 1$ a função de vizinhança $\pi(\xi(t), i, t)$ estabelece que somente os grupos $\xi(t)$ e $\underline{\xi}(t)$, grupo vencedor e o grupo segundo melhor avaliado na competição, respectivamente, atualizam seus pesos sinápticos. O grupo $\underline{\xi}(t)$ é determinado de acordo com a expressão que se segue:

$$\underline{\xi}(t) = \arg \min_i (\|B^i - u^r(t)\|), i = 1, \dots, N \text{ e } i \neq \xi(t);$$

em que: $\arg \min_i (a_i) = j$ se $a_j \leq a_i, \forall i \in \{1, \dots, N\}$.

A expressão que define a função de vizinhança para $\delta(t) < 1$ é a seguinte:

$$\pi(\xi(t), i, t) = \begin{cases} 1, & \text{se } i = \xi(t); \\ \|B^{\underline{\xi}(t)} - u^r(t)\| / \|B^i - u^r(t)\|, & \text{se } i = \underline{\xi}(t) \text{ e } \|B^{\underline{\xi}(t)} - B^{\xi(t)}\| > \|B^{\underline{\xi}(t)} - u^r(t)\|; \\ 0, & \text{caso contrário.} \end{cases}$$

Em caso de empate, tanto na determinação de $\xi(t)$ quanto de $\underline{\xi}(t)$, o grupo de menor índice é escolhido.

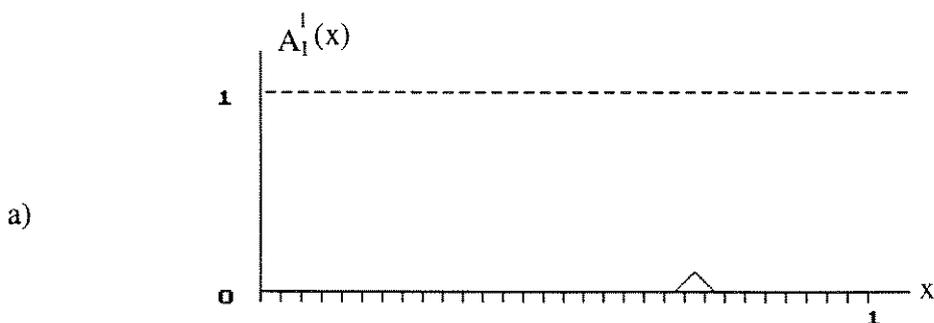
Os operadores de agregação e de processamento sináptico dos neurônios da segunda camada são definidos como a s-norma máximo e a t-norma mínimo, respectivamente. O operador de agregação dos neurônios da terceira camada é igual à t-norma mínimo.

A aprendizagem da fase não supervisionada se encerra para t tal que $\eta(t) < 0,001$.

Experimento 1: Considera-se para a rede N1 uma estrutura que representa uma única regra nebulosa formada por apenas um antecedente. Portanto, a rede N1 deve formar o perfil $A_1^i(x)$ durante a fase não supervisionada da aprendizagem. As entradas escolhidas no intervalo unitário segundo uma distribuição uniforme. A Figura 3.7 mostra os perfis de $A_1^i(x)$ em três instantes distintos: a) $t = 0$, b) $t = 500$ e c) t para o qual $\eta(t) < 0,001$. Observa-se que o perfil formado pela rede N1 aproxima-se de um triângulo isósceles.

Experimento 2: As entradas são escolhidas de acordo com uma distribuição normal de média 0,5 e variância 0,025. A Figura 3.8 ilustra o perfil final de $A_1^i(x)$ (para t tal que $\eta(t) < 0,001$). A rede N1 é definida com a mesma estrutura do caso anterior. A função de pertinência de $A_1^i(x)$ associa elevados graus de pertinência a um intervalo reduzido em torno de 0,5. Esta concentração de altos graus de pertinência é ainda mais nítida neste caso do que para o caso correspondente ao experimento 1.

Experimento 3: Os perfis $A_j^i(x)$ adquiridos por uma rede N1 cuja estrutura codifica 11 regras, cada uma com dois antecedentes, são apresentados na Figura 3.9. Neste experimento, as entradas também são escolhidas segundo uma distribuição uniforme e escolhidas no intervalo unitário $[0, 1]$. Na mesma Figura 3.9 observa-se a disposição espacial final dos centros de área B^i . Eles são representados por pontos assinalados no espaço de entrada $[0, 1] \times [0, 1]$. Note que os perfis das funções de pertinência $A_j^i(x)$ aproximam funções triangulares e que as funções triangulares definidas para uma das dimensões superpõem-se devido ao alinhamento dos respectivos centros de área.



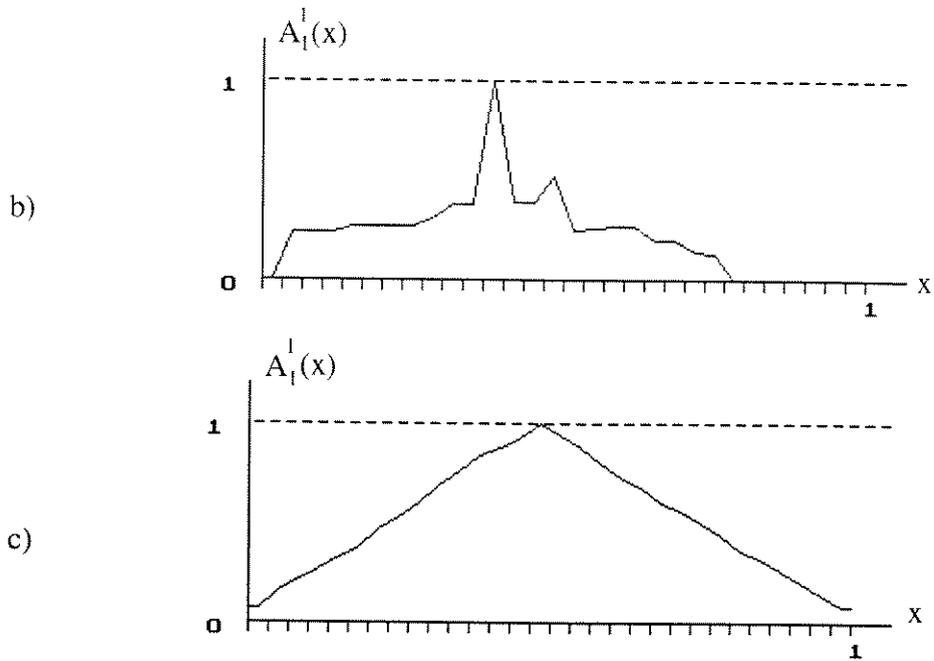


Figura 3.7: Perfis de $A_1^i(x)$ em três instantes distintos: a) $t = 0$, b) $t = 500$ e c) t tal que $\eta(t) < 0,001$.

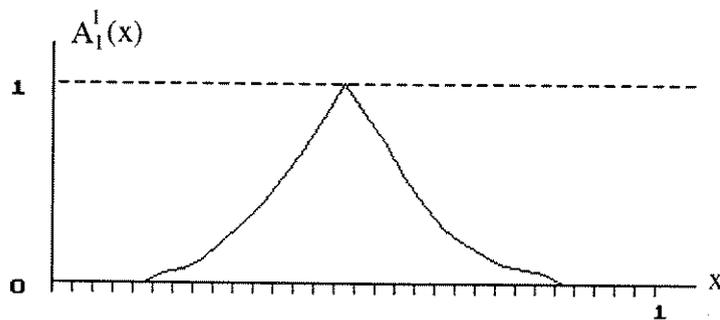


Figura 3.8: Perfil de $A_1^i(x)$ obtido no caso de entradas segundo uma distribuição normal.

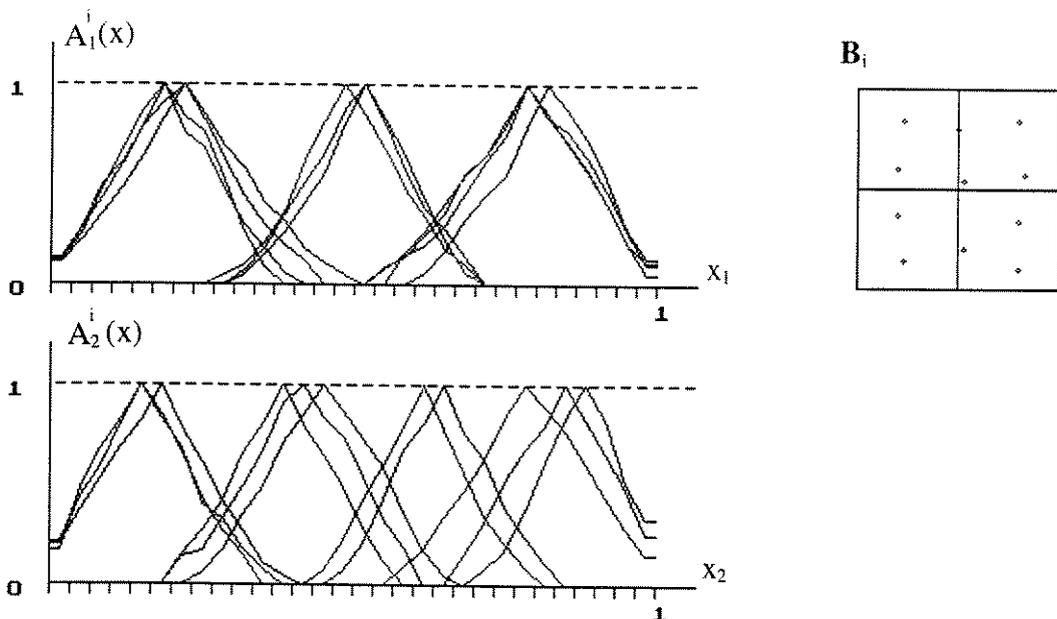


Figura 3.9: Perfis finais de $A_j^i(x)$ para 11 regras e posição dos respectivos centros de área B^i .

Nos experimentos seguintes a rede N1 é aplicada na aproximação de funções. O fator de aprendizagem $\mu = 0,01$ foi utilizado nas expressões (3.10) para ajuste dos parâmetros $\underline{\sigma}^i$ e $\bar{\sigma}^i$ nos conseqüentes $g^i(x)$. Os valores iniciais para estes parâmetros são tais que $0 < \underline{\sigma}^i(0)$, $\bar{\sigma}^i(0) < 15$. O índice de desempenho RMSE adotado para avaliação do resultado de aproximação é definido tal como segue:

$$\text{RMSE} = \sqrt{\left(\frac{1}{T}\right) \sum_{t=1}^T (v(t) - y(t))^2}$$

em que $v(t)$ e $y(t)$ são a saída desejada e saída da rede N1, respectivamente, e t está associada a t -ésima entrada apresentada a rede N1. Em cada avaliação são consideradas um total de 900 entradas ($T = 900$) diferentes das utilizadas durante a aprendizagem.

Experimento 4: Para ilustrar mais explicitamente as características e contribuições da função $g^1(x)$, a função $f(x) = x(x - 0,5)(x - 1) + 0,1$ é considerada para a tarefa de aproximação. A arquitetura da rede N1 é definida de forma que codifique uma única regra. Ela deve ser capaz de encontrar parâmetros para definição de um único conseqüente $g^1(x)$. Desde que só uma regra pode ser ativada, seu conseqüente, $g^1(x)$, deve aproximar a função $f(x)$ em todo o domínio $[0, 1]$. Dados de entrada foram apresentados à rede N1 segundo uma distribuição uniforme. A função $g^1(x)$, após aprendizagem, os seguintes parâmetros foram encontrados:

$$\bar{\sigma}^1 = 15,6;$$

$$\underline{\sigma}^1 = 14,6;$$

$$\bar{u}^1 = 0,211 \text{ e } \text{MAX}^1 = 0,148;$$

$$\underline{u}^1 = 0,788 \text{ e } \text{MIN}^1 = 0,051 \text{ e};$$

$$\text{STAR}^1 = 0,1.$$

Substituindo-se valores na definição (3.9) da função $g^1(x)$ obtém-se:

$$g^1(x) = (0,148 - 0,1)\exp(-15,6(0,211 - x)^2) + (0,051 - 0,1)\exp(-14,6(0,788 - x)^2) + 0,1;$$

ou

$$g^1(x) = 0,048\exp(-15,6(0,211 - x)^2) - 0,049\exp(-14,6(0,788 - x)^2) + 0,1.$$

O resultado da aproximação é apresentado na Figura 3.10(a). Nota-se que $g^1(x)$ aproxima a função $f(x)$ melhor entre os pontos de inflexão de máximo e mínimo do que nos extremos do intervalo $[0, 1]$ onde diverge. Na Figura 3.10(b) apresenta-se os gráficos de cada um dos termos que compõe o conseqüente $g^1(x)$ visando uma melhor compreensão de seu perfil. O resultado da aproximação de $f(x)$ obtida pela rede N1 é tal que $\text{RMSE} = 0,6e10^{-4}$.

Outros experimentos de aproximação consideram 5 funções de duas variáveis definidas no espaço $[0, 1] \times [0, 1]$, a saber (funções sugeridas e utilizadas no trabalho de [Rovatti96]):

- a) $f_1(x_1, x_2) = x_1 + x_2$;
- b) $f_2(x_1, x_2) = 0,5 + 64(x_1 - 0,5)(x_2 - 0,5)(x_1 + 0,125) / (1 + (4x_1 - 2)^2 + (4x_2 - 2)^2)$;
- c) $f_3(x_1, x_2) = 0,5 \exp(-20((x_1 - 0,5)^2 + (x_2 - 0,5)^2)) + \exp(-10(x_1^2 + x_2^2))$;
- d) $f_4(x_1, x_2) = 1 / (1 + \exp(10(x_1 - x_2)))$;
- e) $f_5(x_1, x_2) = 0,5(1 + \sin(2\pi x_1)\cos(2\pi x_2))$.

Experimento 5: Cada função $f_i(x_1, x_2)$ é aproximada a partir de um conjunto de dados formado por 2000 pares entrada/saída $((x_1, x_2), f_i((x_1, x_2)))$, $i = 1, \dots, 5$, em que $x_j \in [0, 1]$. Os dados de entrada são escolhidos em $[0, 1] \times [0, 1]$ de forma aleatória segundo uma distribuição uniforme. A imagem de cada função é normalizada para que $f_i((x_1, x_2)) \in [0, 1]$. O índice de desempenho adotado para avaliação é o mesmo RMSE definido anteriormente.

A Figura 3.11 mostra em (a) a superfície da função $f^1(\cdot)$ e em (b) a aproximação obtida pela rede N1. De forma análoga são apresentados os resultados de simulação relativos às aproximações de $f^2(\cdot)$, $f^3(\cdot)$ e $f^4(\cdot)$ nas Figura 3.12, Figura 3.13 e Figura 3.14, respectivamente (sempre a superfície da função exata é exibida em (a) e a superfície resultado da aproximação em (b)).

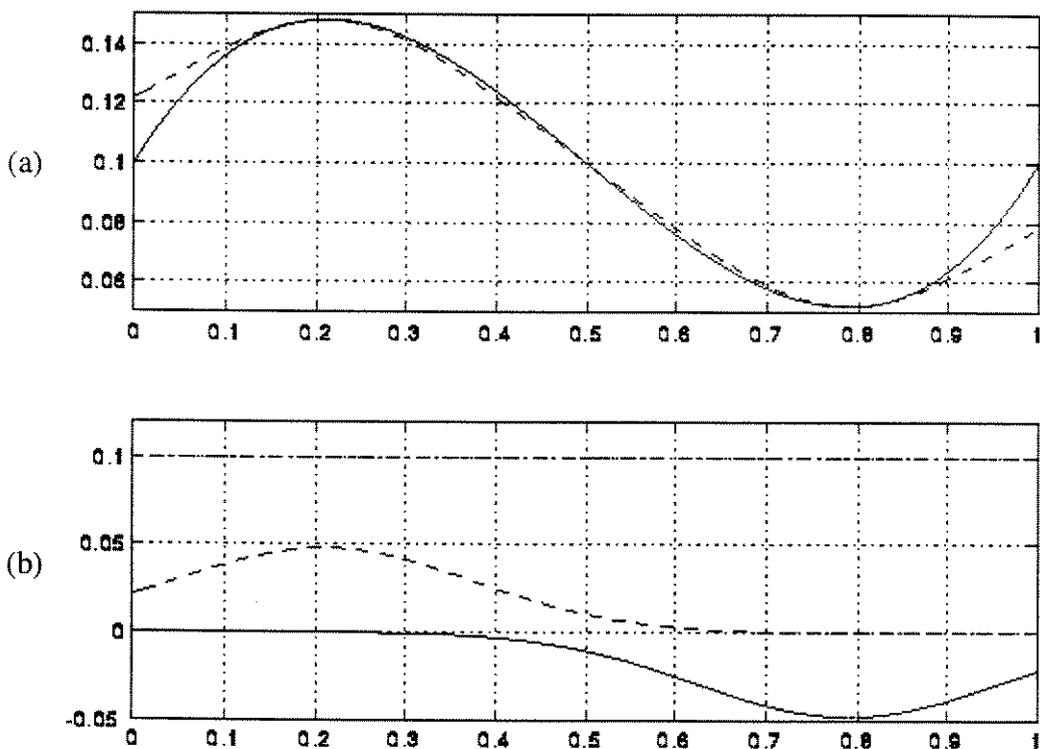
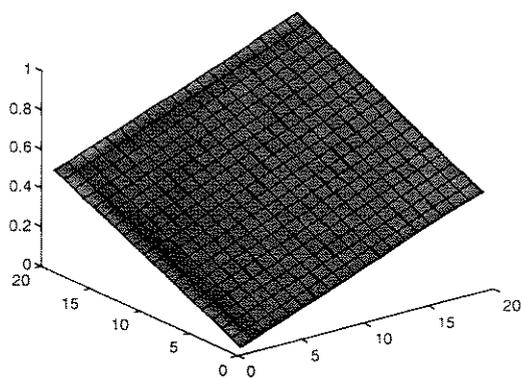
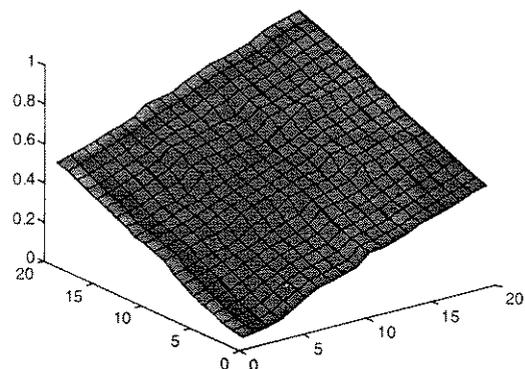


Figura 3.10: (a) Função $f(x)$ (—) e função aproximada por $g^1(x)$ (-----). (b) Parcelas de $g^1(x)$: termo constante (.....), exponencial de média \bar{u}^1 (-----) e exponencial de média u^1 (—).

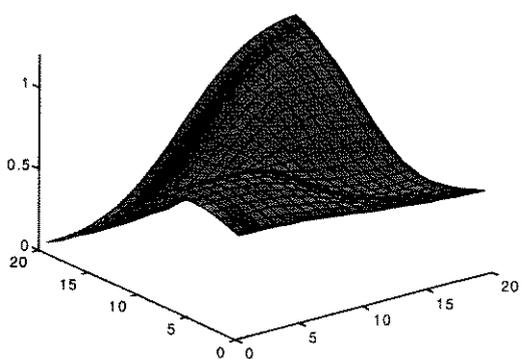


(a)

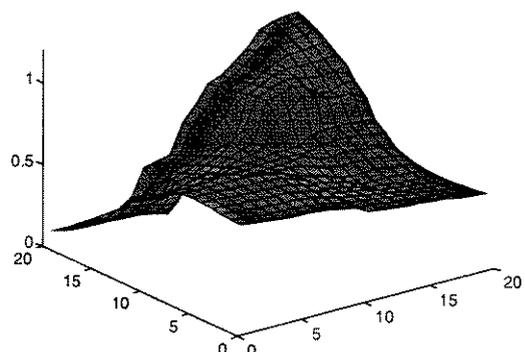


(b)

Figura 3.11: (a) Função $f^1(x_1, x_2)$ e (b) aproximação pela rede N1 (RMSE = 0,018).

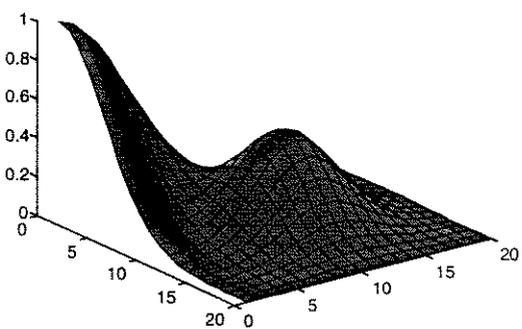


(a)

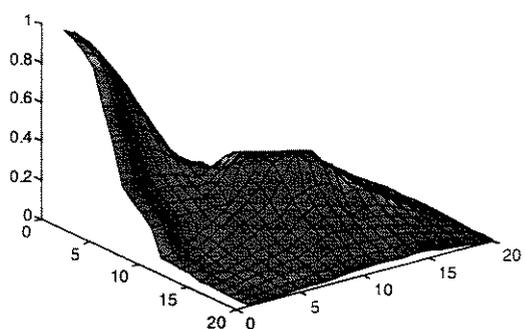


(b)

Figura 3.12: (a) função $f^2(x_1, x_2)$ e (b) aproximação pela rede N1 (RMSE = 0,036).

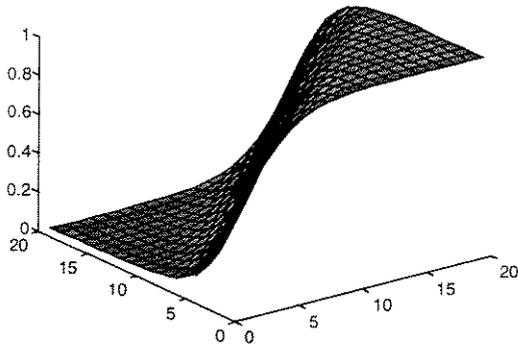


(a)

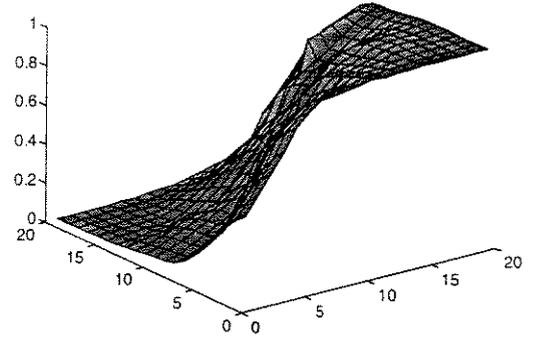


(b)

Figura 3.13: (a) Função $f^3(x_1, x_2)$ e (b) aproximação pela rede N1 (RMSE = 0,034).

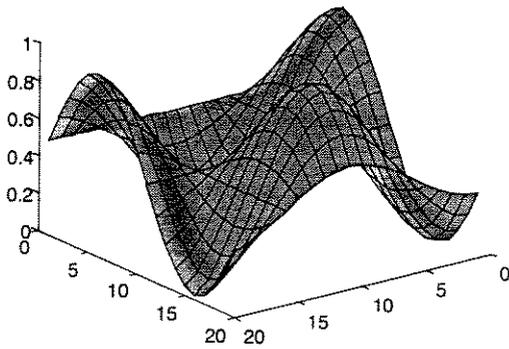


(a)

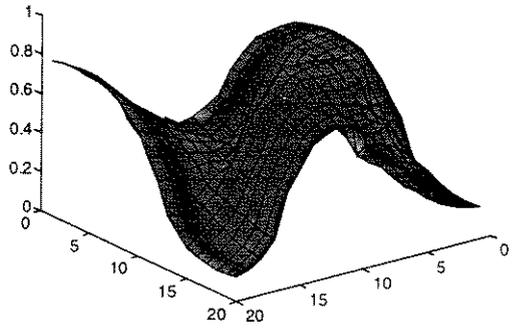


(b)

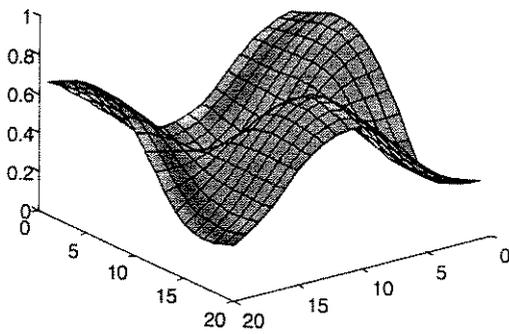
Figura 3.14: (a) Função $f^4(x_1, x_2)$ e (b) aproximação pela rede N1 (RMSE = 0,052).



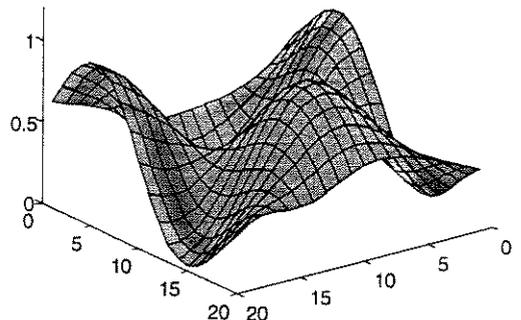
(a)



(b)



(c)



(d)

Figura 3.15: (a) Função $f^5(x_1, x_2)$ e; (b), (c) e (d) aproximações segundo N1 (RMSE = 0,09), NIE (RMSE = 0,12) e BACK (RMSE = 0,03), respectivamente.

Para fins de comparação com a rede N1 duas abordagens são escolhidas: a rede neural multicamadas (BACK) e a rede neural nebulosa (NIE) descrita em [Nie95] (ver seção 2.4).

A rede neural BACK possui uma única camada intermediária formada por 11 neurônios. Utiliza-se, em sua aprendizagem, o método baseado no gradiente (TRAINBPX) disponível no ambiente de programação MATLAB [MATLAB95].

Os parâmetros da rede neural NIE são: fator de redução igual a 0,9 (refere-se ao ajuste da dispersão de cada conjunto nebuloso), fatores de aprendizagem para as estratégias não supervisionada igual a $\eta(t) = 1/t$ e supervisionada igual a 0,85 e; valor de dispersão igual a 0,3. Um neurônio é adicionado à estrutura sempre que o erro encontrado para a resposta da rede N1 for maior que 0,25. O perfil das funções de pertinência que compõem as regras formadas pela rede NIE é definida por funções do tipo Gaussianas.

Na Figura 3.15 a superfície apresentada em (a) corresponde à função $f^5(\cdot)$ e em (b), (c) e (d), estão representadas as superfícies correspondentes às aproximações segundo as abordagens N1, BACK e NIE.

Os resultados de desempenho RMSE obtidos pela rede N1 para aproximação das diferentes funções $f^i(\cdot)$ são resumidos na Tabela 3.1. A comparação entre as abordagens N1, BACK e NIE pode ser observada a partir da Tabela 3.2.

Tabela 3.1: RMSE obtidos pela rede N1 para as funções $f^i(\cdot)$.

	$f^1(\cdot)$	$f^2(\cdot)$	$f^3(\cdot)$	$f^4(\cdot)$	$f^5(\cdot)$
RMSE	0,014	0,033	0,033	0,035	0,081

Tabela 3.2: RMSE das abordagens N1, BACK e NIE nas aproximações da função $f^5(\cdot)$.

	N1	BACK	NIE
RMSE	0,081	0,032	0,121

A aproximação mais precisa da função $f^5(\cdot)$ é obtida pela abordagem BACK. No entanto, o conhecimento obtido pela rede não está explícito em seus pesos sinápticos. Nesta mesma tarefa, a rede N1 obtém melhor desempenho que a rede NIE. Além disso, o número de regras determinado pela rede NIE para cumprir sua tarefa, um total de 142 regras, é expressivamente maior que as 11 regras codificadas na arquitetura da rede N1. Observa-se também que as aproximações obtidas pelas abordagens BACK e NIE são mais suaves que aquela associada à rede N1.

Experimento 6: Este experimento assemelha-se ao experimento 5 no que se refere ao conjunto de funções $f^i(\cdot)$, $i = 1, \dots, 5$; utilizadas na tarefa de aproximação; aos dados de entrada, escolhidos em $[0, 1] \times [0, 1]$ aleatoriamente segundo uma distribuição uniforme, e ao número de dados igual a 2000. Também normalizam-se todas as funções ($f_i((x_1, x_2)) \in [0, 1]$, $i = 1, \dots, 5$) e utiliza-se o mesmo índice de desempenho RMSE. A diferença reside na perturbação inserida nos dados utilizados para aprendizagem. Estes dados, mais precisamente os valores das funções, são perturbados por um ruído aditivo ζ , ou seja, os pares ordenados são formados por $((x_1, x_2), f^i((x_1, x_2)) + \zeta)$, $i = 1, \dots, 5$, em que $f^i((x_1, x_2)) =$

$f_i((x_1, x_2)) + \zeta$. O ruído ζ é tal que sua função densidade de probabilidade é do tipo normal de média zero e variância 0,05.

Inicialmente compara-se o desempenho da rede N1 com a abordagem descrita em [Rovatti96] (denominada FSR) considerando-se todas as funções.

Os valores RMSE das aproximações obtidas pela rede N1 com relação a todas as funções $f^i(.)$ são exibidas na Tabela 3.3. Esta mesma tabela também permite comparações com os resultados encontrados no experimento 5 e com aqueles obtidos pela abordagem FSR. Observa-se que o ruído prejudicou resultados em até 100% com relação ao caso sem ruído. Quando comparada com a abordagem FSR, a rede N1 apresenta resultados favoráveis para as aproximações das funções $f^1(.)$, $f^2(.)$ e $f^5(.)$. Os resultados obtidos por FSR foram superiores nas aproximações das funções $f^3(.)$ e $f^4(.)$. No entanto, FSR utiliza 25 regras nebulosas para aproximar $f^3(.)$ e 20 para $f^4(.)$, compondo uma base de regras maior que no caso da rede N1 (que codifica, conforme enfatizado anteriormente, apenas 11 regras nebulosas).

Tabela 3.3: RMSE obtidos por N1 e por FSR na aproximação das funções $f^i(.)$.

	$f^1(.)$	$f^2(.)$	$f^3(.)$	$f^4(.)$	$f^5(.)$
N1 (sem ruído)	0,018	0,036	0,034	0,052	0,092
N1	0,051	0,070	0,068	0,069	0,110
FSR	0,070	0,077	0,055	0,038	0,130

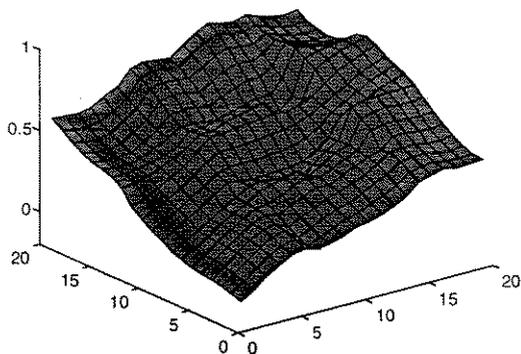
O desempenho de N1 é comparado com as abordagens BACK e NIE somente com respeito a função $f^5(.)$. As configurações destas abordagens (BACK e NIE) utilizadas no experimento 5 são mantidas neste experimento.

Os resultados das aproximações da função $f^5(.)$ obtidas por N1, BACK, NIE e FSR, considerando-se os mesmos dados perturbados com ruído, são condensados na Tabela 3.4. A rede BACK obteve o melhor resultado de aproximação, mas, como já enfatizado, seu conhecimento encontra-se explícito nos pesos sinápticos. A rede N1 alcançou desempenho superior aos obtidos por FSR e NIE, além de codificar apenas 11 regras contra 24 no caso de FSR e 144 no caso de NIE.

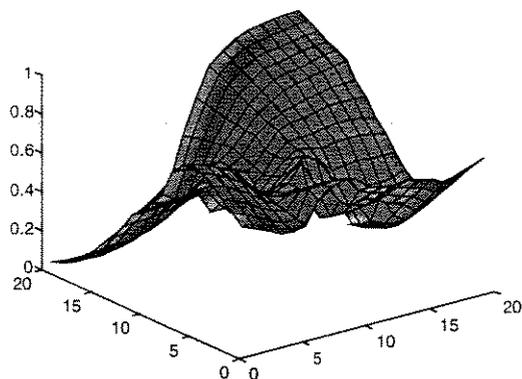
Tabela 3.4: RMSE obtidos por N1, BACK, NIE e FSR na aproximação da função $f^5(.)$.

	N1	FSR	BACK	NIE
RMSE	0,110	0,130	0,094	0,211

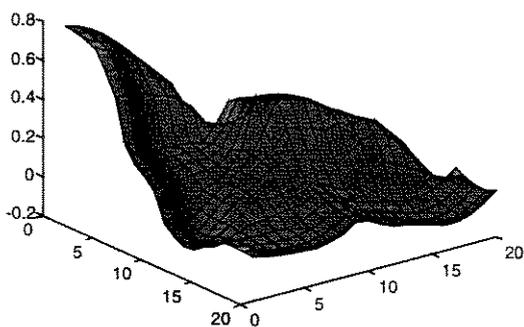
As superfícies correspondentes às aproximações obtidas pela rede N1 para as funções $f^1(.)$, $f^2(.)$, $f^3(.)$ e $f^4(.)$ são apresentadas nas Figura 3.16 (a), (b), (c) e (d), respectivamente. As aproximações da função $f^5(.)$ obtidas pelas abordagens N1, FSR, BACK e NIE são apresentadas nas Figura 3.17 (a), (b), (c) e (d) nesta ordem.



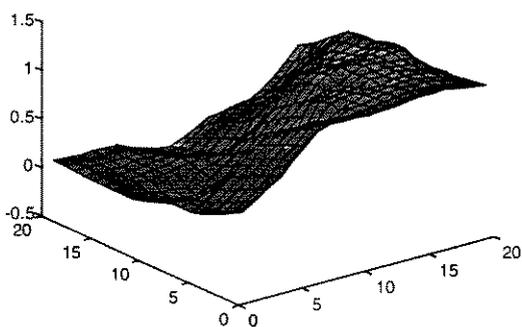
(a)



(b)

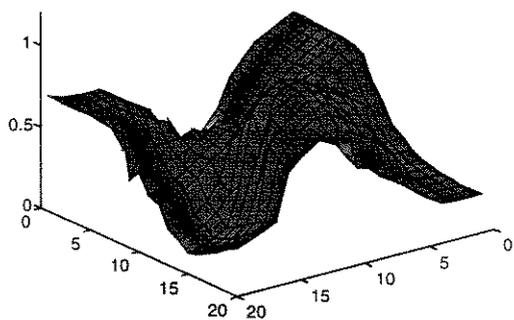


(c)

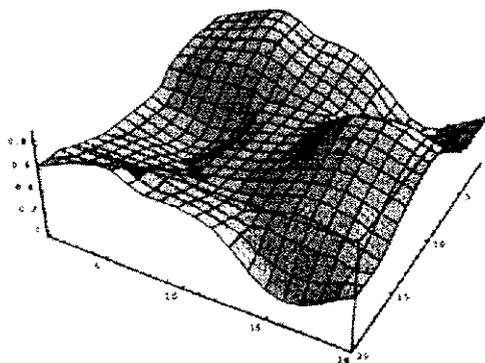


(d)

Figura 3.16: Aproximações obtidas pela rede N1 para as funções: (a) $f^1(\cdot)$ (RMSE = 0,051), (b) $f^2(\cdot)$ (RMSE = 0,070), (c) $f^3(\cdot)$ (RMSE = 0,068) e (d) $f^4(\cdot)$ (RMSE = 0,069).



(a)



(b)

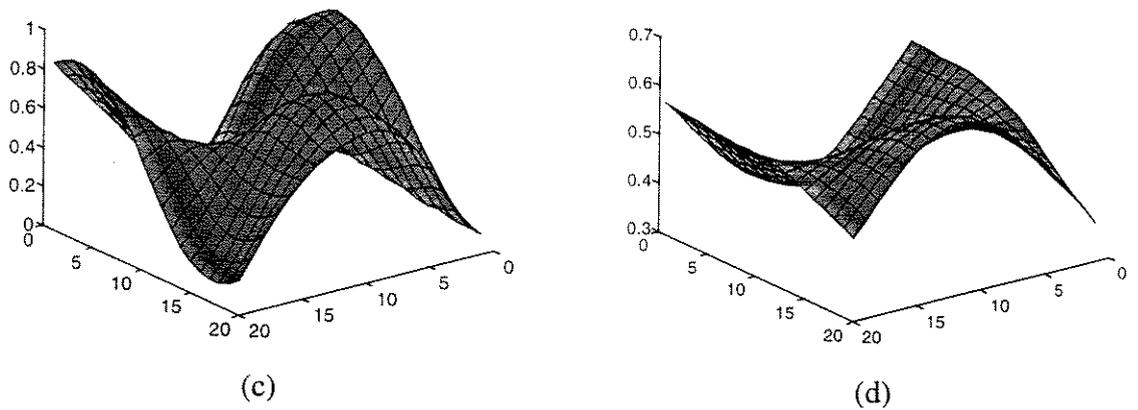


Figura 3.17: Aproximações obtidas pelas abordagens: (a) N1 (RSME = 0,11), (b) ROV (RSME = 0,13), (c) BACK (RSME = 0,09) e (d) NIE (RSME = 0,211).

3.3 Rede Neural Nebulosa N2

3.3.1 Arquitetura

Essencialmente, a rede neural nebulosa N2 [Figueiredo97^a] é uma variação de N1. Elas se assemelham tanto no que se refere à arquitetura quanto ao processamento, pois ambas são baseadas no mecanismo de inferência nebulosa definido em (3.1). A única diferença em suas arquiteturas reside nos pesos sinápticos correspondentes aos conseqüentes $g^i(\mathbf{x})$ das regras nebulosas. Na rede N2 estes pesos sinápticos são independentes das entradas da rede, ou seja, $g^i(\mathbf{x}) = g^i$, $g^i \in \mathfrak{R}$, $i = 1, \dots, N$. Todos os outros parâmetros são definidos de forma idêntica ao caso da rede N1. Assim acontece com os operadores sinápticos e de agregação e com a função de ativação dos neurônios das diversas camadas bem como com os pesos sinápticos (a menos dos pesos $g^i(\mathbf{x})$). Logo, todo o processamento da rede neural N2 é idêntico ao da rede N1 e, conseqüentemente, tal qual a rede N1, a rede neural nebulosa N2 também pode ser vista tanto como uma rede neural ou como um sistema nebuloso ambos apresentando as mesmas propriedades. A Figura 3.18 mostra a arquitetura da rede N2.

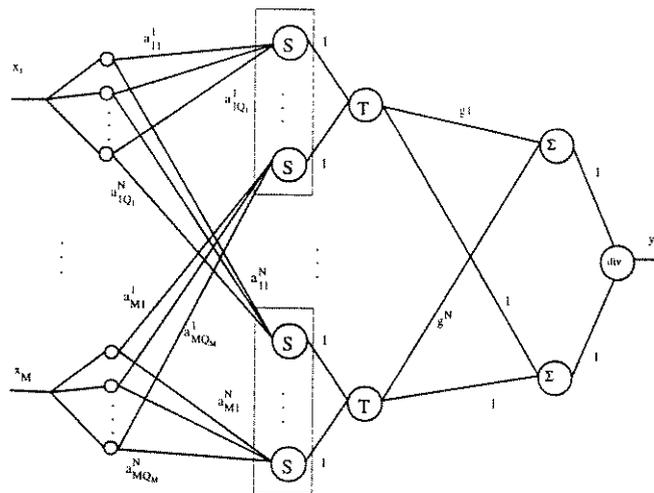


Figura 3.18: Arquitetura da rede neural nebulosa N2.

3.3.2 Aprendizagem

As redes N1 e N2 assemelham-se, em parte, também no que se refere à aprendizagem. A rede N2 deve aprender os pesos sinápticos a_{jk}^i relacionados aos neurônios da segunda camada, bem como os pesos sinápticos g^i da quarta camada. A aprendizagem dos pesos sinápticos a_{jk}^i baseia-se no método competitivo, enquanto os pesos g^i são determinados pelo método baseado em gradiente, assemelhando-se, tal qual no caso da rede N1, com a aprendizagem das redes do tipo contrapropagação. As semelhanças no processo de aprendizagem das redes N1 e N2 limitam-se até este ponto. Diferenças significativas surgem sob outros pontos de vista. De um lado a rede N1 determina, ela própria, os perfis das funções de pertinência relativas aos antecedentes das regras nebulosas. A rede N2 necessita que os perfis sejam preestabelecidos. Por outro lado nota-se que a incorporação de novos conhecimentos pela rede N2 causa menos dificuldades que no caso da rede N1. Enquanto esta requer que todo o processo de treinamento seja repetido para que novos conhecimentos sejam adquiridos, a rede N2 é capaz de aprender de forma menos custosa. Esta diferença é bastante acentuada principalmente quando entradas pertencentes a regiões do espaço não exploradas durante o período de treinamento são apresentadas a rede neural. No caso das redes do tipo N2, estas entradas são utilizadas para atualização de conhecimento. Ao contrário, as rede do tipo N1 têm como única opção apresentar saídas para as quais o desempenho provavelmente não alcança o nível aceitável.

A aprendizagem da rede N2 depende da apresentação de pares de entrada/saída. O t -ésimo par apresentado a rede N2 é dado por $(\mathbf{u}(t), v(t))$, $\mathbf{u}(t) = (u_1(t) \dots u_M(t))'$ e $v(t)$ é a saída desejada correspondente à entrada $\mathbf{u}(t)$.

Considera-se que as funções de pertinência $A_j^i(\cdot)$ possuam perfis idênticos e simétricos, sendo que cada qual é definida pelos respectivos valor modal c_j^i e dispersão ρ^i (por exemplo, funções do tipo Gaussianas, funções com perfis triangulares, etc.). Determinando-se o valor modal e dispersão de cada função $A_j^i(\cdot)$ e estando definidos os intervalos de discretização $I_{jk} = (x_{jkI}, x_{jkF}]$, $k = 1, \dots, Q_j$, $j = 1, \dots, M$, é possível determinar os pesos sinápticos a_{jk}^i tal como segue:

$$a_{jk}^i = A_j^i(x^*);$$

$$\text{em que: } x^* = (x_{jkI} + x_{jkF})/2 \text{ e } I_{jk} = (x_{jkI}, x_{jkF}].$$

O valor modal e a dispersão de cada função $A_j^i(\cdot)$ serão aprendidos segundo o método de aprendizagem competitivo. A partir deles todos os pesos a_{jk}^i , $k = 1, \dots, Q_j$, são determinados.

A função índice de desempenho necessária à aplicação do método baseado em gradiente é definida por: $E(\mathbf{u}(t)) = \frac{1}{2} (v(t) - y(t))^2$ ($y(t)$ é a saída da rede neural N2 correspondente à

entrada $\mathbf{u}(t)$). Desta forma, sempre que os pesos sinápticos g^i forem ajustados, a seguinte equação é utilizada:

$$g^i(t+1) = g^i(t) - \mu \partial E / \partial g^i.$$

A saída da rede neural $y(\mathbf{u}(t))$ para a entrada $\mathbf{u}(t)$ é dada pela expressão análoga a (3.4) (em que g^i é constante), ou seja:

$$y(\mathbf{u}(t)) = \frac{\sum_{i=1}^N H^i(\mathbf{u}(t))g^i}{\sum_{i=1}^N H^i(\mathbf{u}(t))}.$$

Calculando-se a derivada parcial $\partial E / \partial g^i$ e substituindo-se na expressão anterior, tem-se:

$$g^i(t+1) = g^i(t) + \mu \left((v(t) - y(t)) H^i(\mathbf{u}(t)) / \sum_{h=1}^N H^h(\mathbf{u}(t)) \right).$$

Esta última expressão será utilizada nos dois métodos de aprendizagem propostos para a rede N2, a saber: MT1 e MT2, descritos em seguida. (Nas descrições que se seguem $D(\mathbf{x}, \mathbf{z})$ significa a distância Euclidiana entre os vetores \mathbf{x} e \mathbf{z} , porém outras medidas de distância podem ser utilizadas).

Método MT1

No método MT1 o número de grupos de neurônios da segunda camada (correspondente ao número de regras nebulosas) é preestabelecido. São N grupos e há M funções $A_j^i(\cdot)$, $j = 1, \dots, M$, para cada um deles (veja (3.1) e a descrição da arquitetura de N2 em 3.2.1).

Estes grupos de neurônios, tal qual na rede N1, competem entre si sempre que a rede N2 recebe uma nova entrada. O grupo vencedor estabelece quais e como os parâmetros da rede neural nebulosa são ajustados. O resultado da competição depende da existência ou não de regras tais que $H^i(\mathbf{u}(t)) \neq 0$.

O método MT1 possui duas fases. Na primeira fase a rede neural nebulosa N2 aprende com os pares de entrada/saída disponíveis pertencentes a um conjunto Γ . Estes pares são escolhidos aleatoriamente e apresentados a rede N2. Nesta fase o conhecimento básico necessário à modelagem é adquirido pela rede N2. Assim, os valores modais c_j^i são posicionados adequadamente em seus respectivos universos de discurso e a dispersão de cada função $A_j^i(\cdot)$ é amplificada convenientemente garantindo que todo o espaço de entrada seja coberto pelas regras estabelecidas. Ou seja, para qualquer entrada apresentada à rede N2 sempre há um grupo neural i pertencente à sua segunda camada tal que $H^i(\mathbf{u}(t)) \neq 0$. A primeira fase da aprendizagem se encerra após a rede cobrir todo o espaço de entradas e o fator $\eta^i(t)$ atingir valores menores que um certo valor mínimo para todo i . Ao final da primeira fase a rede N2 já está pronta para entrar em fase de operação. A partir de então somente os pesos $g^i(t)$ são ajustados a cada par entrada/saída apresentado à rede N2. Nesta fase as entradas da rede provêm diretamente do sistema a ser modelado.

No algoritmo MT1 são utilizados diversos parâmetros e fatores de aprendizagem. Os fatores de aprendizagem $\eta^i(t)$ e $\mu \in (0, 1)$ são utilizados para ajuste dos valores modais c_j^i dos conjuntos $A_j^i(\cdot)$ e conseqüentes das regras nebulosas, respectivamente. Neste trabalho considera-se $\eta^i(t) = \varepsilon/(\alpha^i(t) + 1)$, em que $\alpha^i(t)$ é o número de vezes que o valor modal c_j^i é ajustado até a t-ésima entrada e $\varepsilon \in (0, 1)$. Para ajuste da dispersão ρ^i dos conjuntos nebulosos $A_j^i(\cdot)$ emprega-se o fator de amplificação λ .

O algoritmo de aprendizagem MT1 para a fase I é tal como segue:

1- Inicialização: Escolha o perfil para as funções $A_j^i(\cdot)$. O valor modal c_j^i e a dispersão ρ^i correspondente a cada uma destas funções deve ser inicializado aleatoriamente. A dispersão $\rho^i > 0$ devem assumir valores pequenos e o valor modal c_j^i deve pertencer ao universo de discurso da respectiva variável x_j . Faça $\alpha^i(0) = 0$, $i = 1, \dots, N$. Defina valores para ε e μ , ambos pertencentes ao intervalo $(0, 1)$, bem como para $\lambda > 1$, (mas, próximo à unidade).

2- Estímulo: Apresente à rede N2 um vetor $\mathbf{u}(t) \in \Gamma$ escolhido aleatoriamente.

3- Competição: Determine o vencedor da competição: Dois casos a) e b) são possíveis:

a) $\exists i \mid H^i(\mathbf{u}(t)) \neq 0$. Então o grupo neural $\xi(t)$ vence a competição se $H^{\xi(t)}(\mathbf{u}(t)) \geq H^i(\mathbf{u}(t))$, $\forall i \in \{1, \dots, N\}$.

b) $H^i(\mathbf{u}(t)) = 0$, $\forall i \in \{1, \dots, N\}$. Neste caso, o vencedor da competição é determinado a partir das distâncias $D(\mathbf{c}^i(t), \mathbf{u}(t))$ entre a entrada $\mathbf{u}(t)$ e o vetores $\mathbf{c}^i(t) = (c_1^i(t) \dots c_M^i(t))'$ (vetor de valores modais), $i \in \{1, \dots, N\}$. O grupo vencedor $\xi(t)$ é tal que $D(\mathbf{c}^{\xi(t)}, \mathbf{u}(t)) \leq D(\mathbf{c}^i, \mathbf{u}(t))$, $\forall i \in \{1, \dots, N\}$.

4.- Ajuste e Atualização:

4.1- Se o vencedor for escolhido segundo o caso a) então as equações para ajuste do valor modal $c_j^i(t)$, dispersão $\rho^i(t)$ e conseqüente $g^i(t)$ e para atualização $\alpha^{\xi(t)}$ de parâmetros são dadas por:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \eta^{\xi(t)}(t) \left(c_j^{\xi(t)}(t) - u_j(t) \right), j = 1, \dots, M$$

$$c_j^i(t+1) = c_j^i(t), i = 1, \dots, N, i \neq \xi(t)$$

$$\rho^{i(t)}(t+1) = \rho^{i(t)}(t), i = 1, \dots, N$$

$$g^i(t+1) = g^i(t) + \mu \left((v(t) - y(t)) H^i(\mathbf{u}(t)) / \sum_{h=1}^N H^h(\mathbf{u}(t)) \right), i = 1, \dots, N$$

$$\alpha^{\xi(t)}(t+1) = \alpha^{\xi(t)} + 1$$

$$\alpha^{i(t)}(t+1) = \alpha^{i(t)}, i = 1, \dots, N, i \neq \xi(t)$$

Prossiga o processo de aprendizagem em 5.

4.2- Caso contrário, se o vencedor for escolhido segundo o caso b) então faça:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \eta^{\xi(t)}(t) \left(c_j^{\xi(t)}(t) - u_j(t) \right), j = 1, \dots, M$$

$$c_j^i(t+1) = c_j^i(t), i = 1, \dots, N, i \neq \xi(t)$$

Se recalculando o valor de $H^{\xi(t)}(\mathbf{u}(t))$, já com os valores ajustados $\mathbf{c}^{\xi}(t+1)$, persistir o resultado $H^{\xi(t)}(\mathbf{u}(t)) = 0$, então amplifique a dispersão $\rho^{\xi(t)}(t)$ de cada função $A_j^{\xi(t)}(.)$ até que $H^{\xi(t)}(\mathbf{u}(t)) \neq 0$, ou seja:

repita

$$\rho^{\xi(t)}(t) = \lambda \rho^{\xi(t)}(t)$$

até ($H^{\xi(t)}(\mathbf{u}(t)) \neq 0$)

Em seguida conclua ajustes e atualizações de acordo com as equações abaixo:

$$\rho^{i(t)}(t+1) = \rho^{i(t)}(t), i = 1, \dots, N$$

$$g^i(t+1) = g^i(t) + \mu \left((v(t) - y(t)) H^i(\mathbf{u}(t)) / \sum_{h=1}^N H^h(\mathbf{u}(t)) \right), \text{ para } i = \xi(t)$$

$$g^j(t+1) = g^j(t), j = 1, \dots, N, j \neq \xi(t)$$

$$\alpha^{\xi(t)}(t+1) = \alpha^{\xi(t)} + 1$$

$$\alpha^{i(t)}(t+1) = \alpha^{i(t)}, i = 1, \dots, N, i \neq \xi(t)$$

Prossiga o processo de aprendizagem em 5.

5. Avaliação de Convergência: Se para qualquer $\mathbf{u}(t) \in \Gamma$, $\exists i \mid H^i(\mathbf{u}(t)) \neq 0$ e $\eta^i(t)$ é menor que um certo valor mínimo então finalize o algoritmo em 6. Caso contrário, retorne para 2.

6. Fim: Aprendizagem encerrada.

No algoritmo descrito acima utilizou-se uma função particular para o fator de aprendizagem $\eta^i(t)$. Entretanto, outras funções podem ser escolhidas desde que $\eta^i(t)$ sejam do tipo monotônica decrescente positiva, e tal que $0 < \eta^i(t) < 1$. Ela deve ser escolhida durante o passo de inicialização do algoritmo.

Após a primeira fase de aprendizagem, as regras nebulosas codificadas pela rede N2 cobrem todo o espaço de entradas. Para cada entrada apresentada a rede, pelo menos uma regra é disparada. Considera-se, então, que as funções de pertinência (relativas aos antecedentes das regras) já estejam ajustadas convenientemente. Por este motivo na

segunda fase de aprendizagem somente os consequentes são ajustados. Nesta fase os dados de entrada não estão restritos a um conjunto fixo, mas provém diretamente do sistema a ser modelado. A partir do início desta fase a rede permanece em aprendizagem contínua. O algoritmo de aprendizagem MT1 para a segunda fase é tal como segue:

1. Estímulo: Apresente à rede N2 um vetor $\mathbf{u}(t) \in \Gamma$ escolhido aleatoriamente ou proveniente do próprio sistema a ser modelado.
2. Ajuste: As equações de ajuste de pesos sinápticos são dadas por:

$$g^i(t + 1) = g^i(t) + \mu \left((v(t) - y(t)) H^i(\mathbf{u}(t)) / \sum_{h=1}^N H^h(\mathbf{u}(t)) \right), i = 1, \dots, N$$

3. Retorne para 1.

Método MT2

O método MT2 capacita a rede N2 adquirir novos conhecimentos codificando novas regras nebulosas em sua arquitetura. Para tanto o método MT2 altera a arquitetura da rede. Portanto, o número N de regras nebulosas codificadas pela rede N2 (correspondente ao número de grupos de neurônios na segunda camada) não é constante, ou seja, $N = N(t)$.

Inicialmente a rede não codifica qualquer regra nebulosa em sua arquitetura, ou seja, $N(0) = 0$. A rede N2 codifica a primeira regra nebulosa após receber o primeiro par entrada/saída, em $t = 0$. À medida que a rede N2 recebe pares entrada/saída, ela desenvolve sua arquitetura codificando novas regras nebulosas. Ela também atualiza seu conhecimento ajustando seus pesos sinápticos.

Os grupos neurais da segunda camada competem entre si sempre que a rede N2 recebe uma nova entrada $\mathbf{u}(t)$ (note que para sua aprendizagem a rede N2 recebe pares entrada/saída $(\mathbf{u}(t), v(t))$ provenientes do sistema a ser modelado). O grupo vencedor guia os procedimentos de ajuste dos pesos sinápticos.

A arquitetura da rede N2 é alterada basicamente devido a dois motivos: o desempenho desejado não é satisfeito pela rede N2 ou todos os grupos neurais da segunda camada são tais que $H^i(\mathbf{u}(t)) = 0$. Nestes casos, a rede acrescenta novos elementos a sua estrutura de maneira a codificar uma nova regra nebulosa. A literatura utiliza diversas formas para avaliar o desempenho da rede, e.g., o valor absoluto da diferença (erro) entre a resposta da rede y e a resposta desejada $v(t)$. Um dos índices de desempenho para se avaliar a rede é Para se avaliar o desempenho da rede é possível se avaliar um

No algoritmo MT2 são utilizados fatores de aprendizagem análogos aos do método MT1. Para ajuste dos valores modais c_j^i dos conjuntos $A_j^i(\cdot)$ e consequentes das regras nebulosas, utilizam-se os fatores de aprendizagem $\eta^i(t)$ e $\mu \in (0, 1)$, respectivamente. O fator $\eta^i(t)$ é definido tal qual em MT1, ou seja, $\eta^i(t) = \epsilon / (\alpha^i(t) + 1)$, em que $\alpha^i(t)$ é o número de vezes que o valor modal c_j^i é ajustado até a t-ésima entrada e $\epsilon \in (0, 1)$. Ao

contrário do método MT1 (que possui um fator de amplificação), o método MT2 utiliza um fator de redução $\gamma \in (0, 1)$ para ajuste da dispersão dos conjuntos nebulosos.

O algoritmo de aprendizagem MT2 é tal como segue:

1- Inicialização:

1.1- Defina o perfil para as funções $A_j^i(\cdot)$ com as quais as regras nebulosas são formadas. Defina a função $\eta^i(t)$. Estabeleça valores para ε , μ e γ , todos pertencentes ao intervalo $(0, 1)$. Defina $\Theta \in \mathfrak{R}$, valor inicial da dispersão das funções de pertinência.

1.2- Apresente o primeiro par $(\mathbf{u}(t), v(t))$ para a rede neural. Codifique a primeira regra nebulosa na arquitetura de N2. Para tanto, após atualizar o valor de N, defina os valores modais dos antecedentes como cada componente da entrada e o conseqüente como a saída desejada. Também inicialize o fator de aprendizagem $\alpha^i(t)$, ou seja:

$$N(t + 1) = N(t) + 1$$

Para $i = 1$ faça:

$$c_j^i(t + 1) = u_j(t), j = 1, \dots, M$$

$$\rho^i(t + 1) = \Theta$$

$$g^i(t + 1) = v(t)$$

$$\alpha^i(t + 1) = 0$$

Fim

2- Estímulo: Apresente à rede N2 um novo par $(\mathbf{u}(t), v(t))$ proveniente do próprio sistema a ser modelado.

3- Competição: Determine o vencedor da competição: Dois casos, a) e b), são possíveis:

a) $\exists i \mid H^i(\mathbf{u}(t)) \neq 0$ e $t \neq 1$. Então o grupo $\xi(t)$ vence a competição se $H^{\xi(t)}(\mathbf{u}(t)) \geq H^i(\mathbf{u}(t)), \forall i \in \{1, \dots, N\}$.

b) $H^i(\mathbf{u}(t)) = 0, \forall i \in \{1, \dots, N\}$. Determine a menor distância $D(\mathbf{c}^i(t), \mathbf{u}(t))$ entre o cada vetor $\mathbf{c}^i = (c_1^i, \dots, c_M^i)$ e o vetor de entrada $\mathbf{u}(t)$. O grupo neural vencedor $\xi(t)$ é tal que $D(\mathbf{c}^{\xi(t)}(t), \mathbf{u}(t)) \leq D(\mathbf{c}^i(t), \mathbf{u}(t)), \forall i \in \{1, \dots, N\}$.

4- Ajuste e Atualização:

4.1- Se o grupo vencedor $\xi(t)$ for escolhido de acordo com o caso a) então os seguintes procedimentos de aprendizagem devem ser obedecidos:

4.1.1- Se a rede N2 satisfizer o desempenho esperado então as equações de ajuste são dadas por:

$$c_j^{\xi(t)}(t+1) = c_j^{\xi(t)}(t) + \eta^{\xi(t)}(t)(c_j^{\xi(t)}(t) - u_j(t)), j = 1, \dots, M$$

$$c_j^i(t+1) = c_j^i(t), i = 1, \dots, N(t), i \neq \xi(t), j = 1, \dots, M$$

$$\rho^i(t+1) = \rho^i(t), i = 1, \dots, N(t)$$

$$g^i(t+1) = g^i(t) + \mu \left((v(t) - y(t)) H^i(\mathbf{u}(t)) / \sum_{h=1}^{N(t)} H^h(\mathbf{u}(t)) \right), i = 1, \dots, N(t)$$

$$\alpha^{\xi(t)}(t+1) = \alpha^{\xi(t)} + 1$$

$$\alpha^i(t+1) = \alpha^i(t), i = 1, \dots, N(t), i \neq \xi(t)$$

$$N(t+1) = N(t)$$

Prossiga o processo de aprendizagem em 5.

4.1.2- Se a rede N2 não satisfizer o desempenho estabelecido e $\exists q$ tal que $\mathbf{u}(t) = \mathbf{c}^q(t) = (c_1^q(t) \dots c_M^q(t))'$ então os seguintes ajustes são realizados:

$$c_j^i(t+1) = c_j^i(t); i = 1, \dots, N(t); j = 1, \dots, M$$

Além disso, reduza a dispersão de cada função $A_j^i(\cdot)$ para que regras nebulosas diferentes de q existentes não influenciem no resultado da entrada $\mathbf{u}(t)$:

Para $i = 1$ até $N(t)$ e $i \neq q$ faça:

Se $H^i(\mathbf{u}(t)) \neq 0$ então

repita

$$\rho^i(t) = \gamma \rho^i(t)$$

até $(H^i(\mathbf{u}(t)) = 0)$

$$\rho^i(t+1) = \rho^i(t)$$

Fim

Fim

Seja $k \neq q$ tal que $D(\mathbf{c}^k(t), \mathbf{u}(t)) \leq D(\mathbf{c}^i(t), \mathbf{u}(t)), \forall i \in \{1, \dots, N\}$, então:

$$\rho^q(t+1) = D(\mathbf{c}^k(t), \mathbf{u}(t))$$

Os demais ajustes são definidos pelas seguintes equações:

$$g^q(t+1) = v(t)$$

$$g^i(t+1) = g^i(t), i = 1, \dots, N(T), i \neq q$$

$$\alpha^q(t+1) = \alpha^q(t) + 1$$

$$\alpha^i(t+1) = \alpha^i(t), i = 1, \dots, N(T), i \neq q$$

$$N(t+1) = N(t)$$

Prossiga o processo de aprendizagem em 5.

4.1.3- Se a rede N2 não satisfizer o desempenho estabelecido e $\mathbf{u}(t) \neq \mathbf{c}^i(t) = (c_1^i(t) \dots c_M^i(t))'$, $\forall i \in \{1, \dots, N\}$, então as equações de ajuste são aplicadas de acordo com o seguinte algoritmo:

$$c_j^i(t+1) = c_j^i(t); i = 1, \dots, N(t), j = 1, \dots, M$$

Reduza a dispersão de cada função $A_j^i(\cdot)$ para que as regras nebulosas existentes deixem de influenciar no resultado da entrada $\mathbf{u}(t)$:

Para $i = 1$ até $N(t)$ faça:

Se $H^i(\mathbf{u}(t)) \neq 0$ então

repita

$$\rho^i(t) = \gamma \rho^i(t)$$

até $(H^i(\mathbf{u}(t)) = 0)$

$$\rho^i(t+1) = \rho^i(t)$$

Fim

Fim

$$g^i(t+1) = g^i(t), i = 1, \dots, N(t)$$

$$\alpha^{\xi(t)}(t+1) = \alpha^{\xi(t)} + 1$$

$$\alpha^i(t+1) = \alpha^i(t), i = 1 \dots, N(t), i \neq \xi(t)$$

Acrescente mais uma regra nebulosa à estrutura da rede N2. Os novos valores modais são iguais a cada componente da entrada, a dispersão é igual à distância ao vetor $\mathbf{c}^i(t) = (c_1^i(t) \dots c_M^i(t))'$, $i = \xi(t)$, e o conseqüente da nova regra nebulosa é igual a saída desejada:

$$N(t+1) = N(t) + 1$$

Para $i = N(t+1)$ faça:

$$c_j^i(t+1) = u_j(t), j = 1, \dots, M$$

$$\rho^i(t + 1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{u}(t))$$

$$g^i(t + 1) = v(t)$$

$$\alpha^i(t + 1) = 0$$

Fim

Prossiga o processo de aprendizagem em 5.

- 4.2- Se o grupo vencedor $\xi(t)$ for escolhido de acordo com o caso b) então é necessário alterar a arquitetura da rede N2 para que ela possa codificar mais uma regra nebulosa. Os seguintes ajustes nos pesos sinápticos e em outros parâmetros devem ser aplicados:

$$c_j^i(t + 1) = c_j^i(t); i = 1, \dots, N(t), j = 1, \dots, M$$

$$\rho^i(t + 1) = \rho^i(t)$$

$$g^i(t + 1) = g^i(t), i = 1, \dots, N(t)$$

$$\alpha^i(t + 1) = \alpha^i(t), i = 1 \dots, N(t)$$

$$N(t + 1) = N(t) + 1$$

Os valores modais para a nova regra nebulosa são iguais a cada componente da entrada, a dispersão é igual $D(\mathbf{c}^{\xi(t)}(t), \mathbf{u}(t))$ e seu conseqüente é igual a saída desejada:

Para $i = N(t + 1)$ faça:

$$c_j^i(t + 1) = u_j(t), j = 1, \dots, M$$

$$\rho^i(t + 1) = D(\mathbf{c}^{\xi(t)}(t), \mathbf{u}(t))$$

$$g^i(t + 1) = v(t)$$

$$\alpha^i(t + 1) = 0$$

Fim

Prossiga o processo de aprendizagem em 5.

5. Retorne para 2.

É importante ressaltar que cada alteração na arquitetura da rede neural nebulosa N2, tal como proposto no método MT2, significa mais que ajustar alguns parâmetros, e.g., $\rho^i(t)$, $c^i(t)$ e $g^i(t)$. Para que uma nova regra nebulosa seja codificada na arquitetura da rede N2 é necessário introduzir neurônios e ligações sinápticas nas camadas segunda, terceira e quarta de forma coerente com as definições apresentadas na seção (3.2.1).

A observação referente ao fator de aprendizagem $\eta^i(t)$ utilizado nas equações de ajuste do método MT1 também é pertinente no caso do método MT2. Ou seja, o fator de aprendizagem $\eta^i(t)$ pode admitir outras formas, porém todas devem garantir que $\eta^i(t)$ seja uma função monotônica decrescente positiva, definida de forma que $0 < \eta^i(t) < 1$.

3.3.3 Aproximação

A pequena alteração na definição do conseqüente das regras nebulosas (3.1) não compromete a capacidade de aproximação da rede N2. Nota-se que a única diferença nas arquiteturas das redes N1 e N2 reside na representação dos conseqüentes das regras nebulosas. Na rede N2 os conseqüentes são constantes e não dependentes da entrada, tal como acontece na rede N1. É possível se demonstrar, de forma análoga àquela apresentada no Teorema 3.1, que a rede N2 é um aproximador universal.

3.3.4 Convergência

Os métodos de aprendizagem MT1 e MT2 capacitam a rede N2 adquirir novos conhecimentos sempre que necessário. A rede ajusta seus pesos sinápticos para alcançar o desempenho especificado. Tanto MT1 quanto MT2 não estão limitados a um período de treinamento em que um conjunto de pares entrada/saída são utilizados para a aprendizagem. No entanto, para se analisar características de convergência destes métodos, considere que exista este período de treinamento e que pares entrada saída estejam restritos ao conjunto Γ .

O método MT1 garante convergência dos pesos sinápticos c_j^i pois o fator de aprendizagem $\eta^i(t)$ é uma função decrescente positiva em t . Considerando-se que os pesos sinápticos c_j^i tenham convergido, a análise de convergência dos pesos sinápticos g^i é possível. Suponha que o método do gradiente seja aplicado na solução do problema de otimização definido por

$$\text{Min} \sum_{(\mathbf{u}, \mathbf{v}) \in \Gamma} E(\mathbf{u})$$

em que $E(\mathbf{u}) = (\mathbf{v} - \mathbf{y})^2$ e \mathbf{y} é a saída da rede neural N2 para a entrada \mathbf{u} . Ou seja, deseja-se minimizar o erro quadrático total considerando-se todos os pares pertencentes a Γ . Se a cada ajuste o fator de aprendizagem μ for escolhido de forma a minimizar o valor de $\sum_{(\mathbf{u}, \mathbf{v}) \in \Gamma} E(\mathbf{u})$ (veja capítulo 2) então garante-se a convergência local da solução.

Além disso, o ajuste da dispersão ρ^i de cada função $A_j^i(\cdot)$ garante que o espaço de entrada seja coberto por pelo menos uma regra nebulosa, pois para cada entrada tal que $(\mathbf{u}, \mathbf{v}) \in \Gamma$, $\exists i \in \{1, \dots, N\}$ tal que $H^i(\mathbf{u}(t)) \neq 0$ (observe a descrição do algoritmo de aprendizagem MT1, especificamente, o passo 4.2 e passo 5).

Desde que o conjunto de dados Γ não é conhecido previamente, o método MT1 determina que os pesos sinápticos g^i sejam ajustados após cada entrada ser apresentada à rede. Este procedimento faz com que o comportamento de convergência seja caracterizado pela média de todos os ajustes realizados.

A grande simplicidade do método MT1 pode, em alguns casos, levar a uma limitação inconveniente. Dependendo da extensão do espaço de entradas a ser coberto pelas regras nebulosas codificadas por N2, a dispersão associada às funções de pertinência $A_j^i(.)$ de uma particular regra nebulosa pode tornar-se excessivamente ampla. Esta regra dispararia para entradas pertencentes a um subespaço relativamente vasto, sendo que um mesmo conseqüente contribuiria para a resposta da rede qualquer que fosse a entrada. Esta regra provavelmente prejudicaria o desempenho da rede. O método MT2 tem como um de seus méritos eliminar esta deficiência introduzindo novas regras nebulosas em seu conhecimento. Além disso, permite uma melhor distribuição das regras pelo espaço de entradas.

Para análise da convergência do método MT2, também considera-se um conjunto Γ com um número limitado de pares de entrada/saída. Devido à flexibilidade que o método MT2 confere à rede N2, esta é capaz de representar em sua arquitetura um número de regras nebulosas igual ao número de pares em Γ . Neste caso, cada vetor $\mathbf{c}^i = (c_1^i \dots c_M^i)'$ corresponde a uma entrada \mathbf{u} e cada conseqüente g^i a uma saída v tal que $(\mathbf{u}, v) \in \Gamma$. Esta situação ocorre se para qualquer entrada \mathbf{u} , $(\mathbf{u}, v) \in \Gamma$, apresentada pela primeira vez à rede N2, esta não satisfizer os critérios de desempenho exigidos. Portanto, a cada entrada apresentada a rede (pela primeira vez), uma nova regra nebulosa é inserida em sua arquitetura, conforme o método MT2 determina. Assim, após todos os pares pertencentes a Γ terem sido apresentados à rede N2, esta terá adquirido todo o conhecimento contido neste conjunto. Ou seja, se qualquer entrada for reapresentada à rede, seu desempenho será máximo pois sua saída é idêntica à saída desejada.

3.3.5 Resultados de Simulação

Nesta seção a rede N2 é aplicada na modelagem de um forno descrito em [Box70]. Para a modelagem deste sistema dispõe-se de 296 pares de dados entrada/saída, em que a entrada $u(t)$ é a taxa de variação do fluxo de gás e a saída $v(t)$ é a concentração de dióxido de carbono presente no gás expelido. Duas abordagens foram escolhidas para comparações subsequentes: a primeira denominada LIN é descrita em [LIN95] e a segunda, já adotada na avaliação da rede N1, denominada NIE é descrita em [NIE95]. Ambas as abordagens são redes neurais nebulosas e foram resumidamente descritas na seção (1.6). Pelo fato da rede LIN possuir uma arquitetura fixa ela é tomada como exemplo de comparação com a rede N2 associada ao método MT1. A rede NIE possui uma arquitetura variável. Por este motivo esta rede é comparada com a rede N2 associada ao método MT2.

Os dados são escalados para o intervalo $[0, 1]$. As três abordagens, N2, LIN e NIE, adotam as mesmas 5 variáveis de entrada para modelagem do forno, a saber: $u(t - 6)$, $u(t - 5)$, $u(t - 3)$, $v(t - 1)$ e $v(t - 2)$.

As funções de pertinência das regras nebulosas codificadas pela rede N2 são do tipo Gaussiana. Estas funções são definidas tal como segue:

$$A_j^i(x_j) = \begin{cases} 0, & \text{se } |x_j - c_j^i| > 2\rho^i \\ \exp(-|x_j - c_j^i|/\rho^i) & \text{caso contrário.} \end{cases}$$

O valor de Θ para inicialização da dispersão das funções de pertinência no método de aprendizagem MT2 é igual a 2. Os neurônios da segunda camada possuem como operador sináptico a t-norma mínimo e como operador de agregação de entradas a s-norma máximo. Neurônios da terceira camada possuem como operador de agregação a t-norma produto.

Os primeiros 250 dados foram utilizados para treinar a rede N2. Os dados restantes foram empregados na avaliação de desempenho através do índice E definido por:

$$E = \frac{1}{L} \sum_{t=1}^L (v(t) - y(t))^2.$$

Nas figuras apresentadas a seguir, a concentração de dióxido de carbono nos gases expelidos pelo forno é representada pela linha contínua. A respectiva saída do modelo nebuloso é apresentada pela linha tracejada.

Experimento 1: Os resultados do primeiro experimento referem-se à comparação entre os modelos obtidos pela rede N2 e pela rede LIN. A rede N2 é definida com os seguintes parâmetros: $\varepsilon = 1$, $\mu = 0,5$, $N = 3$ e $\lambda = 1,1$. Os valores iniciais para c_j^i são escolhidos aleatoriamente, enquanto $\rho^i(0) = 1$ e $g^i(0) = 0,1$, $i = 1, \dots, N$. A Figura 3.19 ilustra as respostas do modelo obtido pela rede N2. O desempenho de cada uma das redes bem como o número de regras codificadas em suas estruturas ao final da aprendizagem estão resumidos na Tabela 3.5. Os resultados relativos às redes N2 (associada a MT1) são obtidos através de simulações, enquanto aqueles relativos à rede LIN são transcritos de [Lin95].

A rede LIN apresenta um desempenho melhor que a rede N2. No entanto, sua aprendizagem demanda um grande esforço na determinação de parâmetros, e.g., número de regras nebulosas (ver seção 1.6), anterior à aprendizagem propriamente dita. A intervenção de um projetista é imperativa para o bom desempenho da rede tornando o processo pouco automático. Além disso, o número de parâmetros necessários à definição da rede LIN é maior que no caso da rede N2 para redes com iguais números de regras nebulosas codificadas em suas arquiteturas. Outro ponto desfavorável à rede LIN refere-se capacidade de aprendizagem limitada ao período de treinamento. A rede LIN passa a responder inadequadamente nos casos em que o sistema real apresente variações de parâmetros após o período de aprendizagem. Caso novos conhecimentos devam ser incorporados pela rede neural LIN, todo o processo de aprendizagem deve ser repetido. O método MT1 é capaz de ajustar parâmetros da rede N2 sempre que necessário.

Experimento 2: Em um segundo experimento comparou-se o desempenho da rede N2 (associada a MT2) com o obtido pela rede N1E. Os parâmetros especificados para a rede

N2 são tais que: $\mu = 0,5$ e $\varepsilon = 1$. A rede N2 insere uma nova regra em sua estrutura se $lv(t) - y(t) > 0.09$. A dinâmica do modelo gerado pela rede N2 é apresentada na Figura 3.20.

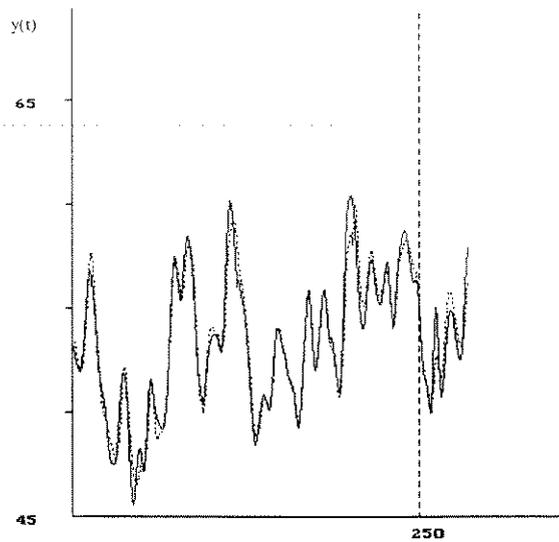


Figura 3.19: Saída da rede N2 associada ao método de aprendizagem MT1.

Tabela 3.5: Desempenhos e número de regras codificadas.

	Desempenho E	Número de regras
LIN	0,26	4
N2 + MT1	0,87	3

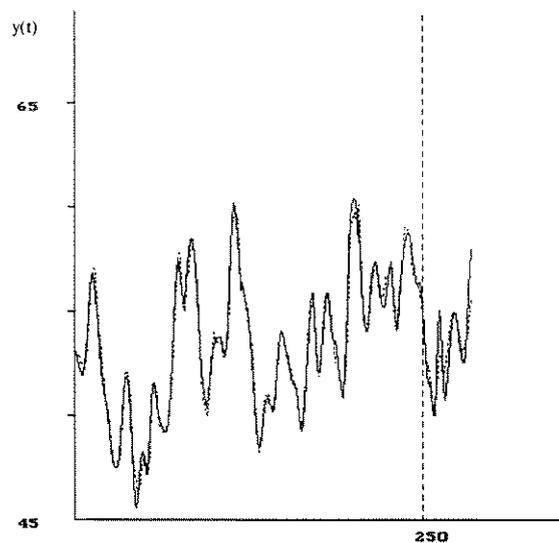


Figura 3.20: Saída da rede N2 associada ao método de aprendizagem MT2 (linha tracejada).

A rede NIE é avaliada considerando-se uma arquitetura inicial capaz de codificar uma única regra nebulosa. Seus antecedentes são definidos por valores modais escolhidos aleatoriamente em seus respectivos universos de discurso e pela dispersão igual a 1. O conseqüente também é escolhido aleatoriamente no seu correspondente domínio. O fator

de aprendizagem associado ao ajuste dos valores modais é específico para cada regra nebulosa, sendo inversamente proporcional ao número de ajustes realizados. Para o ajuste dos conseqüentes utiliza-se um fator de aprendizagem 0,5. A dispersão associada a cada nova regra nebulosa codificada pela rede NIE é igual à unidade. O fator de redução é igual a 0,9. Tal como no caso da rede N2 (associada a MT2), novas regras são incorporadas pela rede NIE se $|v(t) - y(t)| > 0,09$. A Figura 3.21 apresenta o resultado correspondente à modelagem realizada pela rede neural nebulosa NIE.

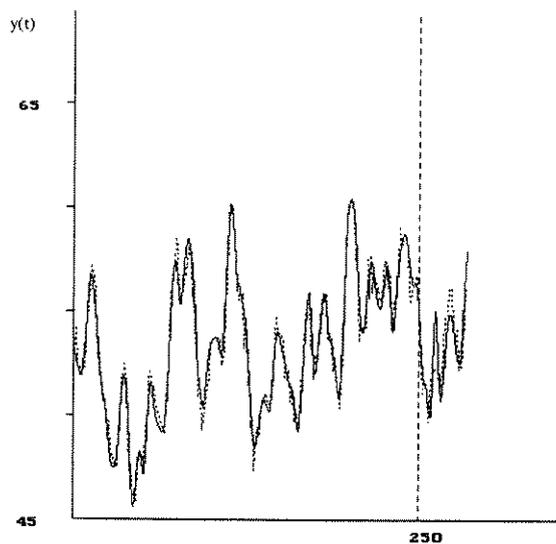


Figura 3.21: Saída da rede neural nebulosa NIE (linha tracejada).

Os resultados deste experimento, desempenho e número de regras pelas redes N2 (associada a MT2) e NIE, estão organizados na Tabela 3.6.

Tabela 3.6: Desempenho e número de regras codificadas.

	Desempenho E	Número de regras
N2+ MT2	0,87	18
NIE	1,23	27

As aprendizagens das redes NIE e N2 (associada a MT2) não estão limitadas a períodos de treinamento. Estas redes verificam seu desempenho a cada entrada apresentada e ajustam parâmetros e arquitetura se necessário. Ou seja, se o desempenho não corresponder ao índice especificado uma nova regra nebulosa é inserida em suas arquiteturas. A Tabela 3.6 mostra que a rede N2 é capaz de obter índice de desempenho melhor que a rede NIE mesmo concentrando seu conhecimento em um menor número de regras codificadas em sua arquitetura.

3.4 Resumo

Redes neurais nebulosas têm como características principais a capacidade de extrair conhecimento de um conjunto de dados numéricos e representá-lo explicitamente na forma de regras nebulosas do tipo se-então, bem como de possibilitar o processo inverso, ou seja, permitir a codificação de conhecimento descrito de forma lingüística diretamente em sua arquitetura visando aplicações em problemas reais. Estas características tornam-se evidentemente importantes no estudo de sistemas complexos e em auxílio à atividades de projeto e síntese.

Neste capítulo propôs-se duas novas classes de redes neurais nebulosas, N1 e N2, ilustrando seus potenciais através de aplicações em problemas de modelagem nebulosa. Ambas as redes neurais nebulosas possuem natureza dual, ou seja, podem ser vistas tanto como uma rede neural quanto como um sistema nebuloso - a representação e o processamento do conhecimento possuem perfeita identidade em ambos os sistemas. Conseqüentemente, elas são também importantes em projeto de sistemas nebulosos por determinarem parâmetros destes sistemas de forma automática.

As redes neurais nebulosas N1 e N2 assemelham-se em suas arquiteturas, processamento e também em alguns aspectos básicos de suas dinâmicas de aprendizagem. Tanto em N1 quanto em N2 emprega-se o método competitivo no ajuste dos antecedentes das regras nebulosas, sendo os conseqüentes ajustados de acordo com o método do gradiente. Muito embora algumas semelhanças sejam observadas nos aspectos básicos dos procedimentos de aprendizagem, diferenças significativas existem. A rede N1 possui uma estrutura fixa. Limita sua aprendizagem a um período de treinamento ou a um conjunto fixo de dados. Ela é capaz de determinar os perfis das funções de pertinência para as regras nebulosas codificadas em sua arquitetura. Ao contrário da rede N1 a rede N2 não limita sua aprendizagem a um período de treinamento. A cada entrada apresentada à rede ela verifica seu desempenho. Sempre que necessário são acrescentados elementos à sua arquitetura de forma que a rede possa comportar novos conhecimentos. A rede neural nebulosa N2 não é capaz de determinar os perfis das regras nebulosas. No entanto, apresenta grande agilidade na aquisição de conhecimento.

Os desempenhos das redes N1 e N2 foram comparados com os correspondentes a outras abordagens. As análises comparativas revelaram-nas competitivas e muitas vezes superiores às abordagens alternativas. Além disso, propriedades tais como aprendizagem de perfis para funções de pertinência em N1, ausência de períodos fixos de treinamento em N2, simplicidade e agilidade na aquisição de conhecimento e dualidade sistema nebuloso/rede neural, em ambas as redes N1 e N2, destacam-nas com relação às demais abordagens pois estas características estão ausentes nas propostas alternativas encontradas na literatura.

Muito embora as redes N1 e N2 tenham sido aplicadas em problemas de modelagem, elas são perfeitamente adequadas a diversos outros tipos de problemas (e.g., problemas de controle) em que os procedimentos de aprendizagem possam ser repetidos tais como descritos nas seções pertinentes deste capítulo.

No próximo capítulo a integração entre as teorias dos conjuntos nebulosos e das redes neurais é explorada sob uma diferente perspectiva: modelagem de sistemas nervosos

biológicos. Novos modelos de neurônios e rede neural nebulosa são propostos. Eles compõem um sistema neural nebuloso aplicado em controle autônomo, mais especificamente, em controle de veículos auto-guiados.

CAPÍTULO 4

SISTEMA NEBULOSO APLICADO EM CONTROLE AUTÔNOMO

4. Sistema Nebuloso Aplicado em Controle Autônomo

4.1 Introdução

O problema de navegação de veículos (robôs móveis) consiste na determinação de uma trajetória que permita ao veículo executar suas tarefas (atingir seus objetivos), evitando colisões. Este problema apresenta diversas versões no que se refere às habilidades que o veículo pode desempenhar, por exemplo: atingir alvos, desviar de obstáculos, tangenciar paredes, recarregar energia, explorar o ambiente, etc. O provável impacto tecnológico causado pela solução deste problema pode ser estimado pelas inúmeras aplicações práticas, inclusive industriais, em que robôs móveis podem ser utilizados, tais como: limpeza de ambientes (pisos), limpeza de embarcações, transporte de materiais e produtos finais em sistemas de manufatura, colheita e transporte de produtos agrícolas, sistemas de vigilância, etc..

Muito embora existam atualmente veículos autônomos em sistemas de manufatura, em geral eles necessitam de um sistema especial para que possam desempenhar suas tarefas. Esta dependência de aparatos de apoio para controle de trajetórias mostra que a teoria atual encontra-se inadequada para projetar veículos que operem em ambientes diferentes dos ambientes específicos de manufatura. Esta constatação mostra que o problema de navegação envolve sérias dificuldades mesmo em suas versões mais simples (por exemplo, tangenciar obstáculos).

Provavelmente as maiores barreiras à solução do problema de navegação estão associadas ao desconhecimento prévio da topologia do ambiente bem como das inúmeras situações diferentes com que o veículo pode deparar-se. Qualquer proposta de solução na qual o conhecimento necessário à navegação seja inserida no controlador requer do projetista a capacidade de representar uma decisão de controle adequada para cada situação possível [Ishikawa91]. Soluções com esta característica, freqüentemente encontrada em propostas fundamentadas na inteligência artificial tradicional, são vistas com certo ceticismo [Vershure92].

O problema de navegação autônoma tem cativado pesquisadores que vêm em sua solução um desafio a ser enfrentado. Atualmente nota-se a formação de uma teoria específica relacionada ao problema de navegação de robôs autônomos [Colombetti96]. Abordagens alternativas às técnicas tradicionais têm sido intensamente investigadas. Em geral, os esforços visam desenvolver controladores inteligentes capazes de aprender o conhecimento necessário ao bom desempenho do veículo. A aprendizagem se sucede enquanto o veículo interage com o ambiente. Estas soluções tornam desnecessário o conhecimento prévio da topologia do ambiente, compensam ruídos presentes em sensores e perturbações sobre atuadores e dispensam a enumeração exaustiva de todas as possíveis situações que o veículo pode enfrentar enquanto navega.

Abordagens baseadas em técnicas de inteligência computacional têm obtido resultados importantes. Um controlador autônomo baseado em redes neurais artificiais é proposto em [Tani96]. O controlador é constituído por uma arquitetura hierarquizada. Dados provenientes de sensores são processados inicialmente em uma rede neural do tipo Kohonen. Em seguida sua saída é processada por uma rede neural multicamadas recorrente. A aprendizagem das redes neurais se faz segundo métodos não-

supervisionados e supervisionados, respectivamente. O controlador é capaz de encontrar trajetórias eficientes (sem colisões e que dirigem-se ao alvo), porém restritas ao ambiente em que ocorre a aprendizagem. Na abordagem apresentada em [Koza93] empregam-se técnicas de computação evolutiva para dotar veículos de comportamentos básicos, tais como: tangenciar paredes e posicionar cargas junto aos limites do ambiente. Em [Li94] as teorias relacionadas aos sistemas nebulosos e aos algoritmos genéticos são integradas para composição do controlador autônomo. O sistema nebuloso representa o conhecimento do controlador por meio de regras linguísticas e aprende utilizando técnicas baseadas em algoritmos genéticos. Resultados apresentados mostram que o veículo aprende a navegar em ambientes nos quais estão presentes configurações tipo corredores estreitos ou tipo rua sem-saída. Redes neurais nebulosas e algoritmos genéticos compõem o controlador autônomo apresentado em [Oliveira94] e [Oliveira95^b]. A rede neural é capaz de aprender regras nebulosas do tipo se-então correspondentes aos comportamentos de captura de alvos e desvio de obstáculos. O veículo trafega sem colisões em ambientes simples (poucos obstáculos) e apenas naqueles em que a aprendizagem se realiza. O trabalho relata que a aprendizagem baseada em algoritmos genéticos tende a gerar trajetórias específicas refletindo a necessidade de otimizar a função objetivo (“fitness”). Esta é uma das razões através da qual se justifica a fraca capacidade de generalização do controlador. Embora ele seja capaz de encontrar trajetórias no ambiente de aprendizagem, o mesmo não se verifica em ambientes diferentes deste.

Alguns pesquisadores (da área de “Artificial Life”) admitem como propostas de soluções apenas aquelas cujos resultados tenham sido obtidos em implementações reais [Brooks91]. Seus sistemas inteligentes são desenvolvidos de acordo com a teoria do comportamento [Steels94]. Eles são formados por um conjunto de subsistemas, cada qual associado a um comportamento distinto, operando sem supervisão. As interações entre subsistemas dependem de parâmetros que estabelecem níveis de prioridade para cada comportamento. Condições do ambiente e estado do veículo determinam os diferentes níveis de prioridade para cada comportamento. O comportamento global do veículo é resultado das interações entre subsistemas. Sob esta abordagem também têm sido investigados comportamentos em ambientes de navegação multi-agentes [Mataric92].

Observa-se que animais são capazes de circular por ambientes desconhecidos desviando-se de obstáculos e alcançando seus objetivos. Por causa destas habilidades eles têm sido fonte de inspiração para muitos pesquisadores que buscam uma solução para o problema de navegação.

A teoria relacionada à seleção de grupos neurais (TSGN) desenvolvida por G. Edelman [Edelman87] e seus colaboradores tenta entender o fenômeno da inteligência a partir da observação e investigação de sistemas nervosos biológicos. Em [Vershure92] e [Almassi92] são utilizados conceitos da TSGN no desenvolvimento de controladores autônomos para navegação de veículos.

A TSGN desconsidera a hipótese de que conhecimentos específicos a uma determinada tarefa estejam presentes em seus modelos antes mesmo que iniciem sua interação com o ambiente. Portanto, a menos de comportamentos instintivos que são gerados via evolução natural, a teoria da seleção de grupos neurais visa desenvolver modelos capazes de adquirir conhecimento de forma não supervisionada. O conceito de repertório neural tem sido utilizado pela TSGN com intuito de modelar sistemas neurais biológicos. Repertórios

neurais são redes neurais destinadas a desempenhar determinadas tarefas específicas, por exemplo, aquelas relacionadas aos comportamentos instintivos. Repertórios neurais atuando em grupo e capazes de interagir entre si formam sistemas neurais cujo desempenho é superior ao total dos desempenhos individuais dos repertórios.

Neste capítulo propõe-se um sistema neural nebuloso para controle autônomo de veículos [Figueiredo96]. A concepção básica do controlador é inspirada nos seres vivos. Além disso, o conceito de repertório neural, proposto pela TSGN, é adotado, visando a modelagem dos sistemas envolvidos. A arquitetura do controlador é do tipo hierárquico. Dois repertórios neurais, situados em um mesmo nível de decisão, geram comportamentos instintivos distintos: busca de alvos e desvio de obstáculos, respectivamente. No contexto do problema de navegação autônoma, sabe-se que tais comportamentos são conflitantes. Portanto, para que o veículo possa ser guiado ao seu alvo sem que colisões ocorram, é necessário que eles sejam coordenados. Um terceiro repertório neural, situado em nível hierárquico superior, realiza esta tarefa de coordenação. Ele é constituído por uma rede neural nebulosa e por um neurônio nebuloso. O sistema neural aprende sem supervisão. Comportamentos instintivos são adquiridos via técnicas de computação evolutiva as quais emulam o processo de evolução natural. O repertório de coordenação aprende segundo o método por reforço, inspirado na teoria clássica de condicionamento animal [Kandell91]. A respectiva rede neural nebulosa é do tipo construtiva. Durante a aprendizagem sua arquitetura adapta-se de forma a adquirir novos conhecimentos. Resultados de simulação confirmam as características de aprendizagem e eficiência do controlador proposto. Após a aprendizagem, o sistema neural nebuloso torna-se capaz de controlar (guiar) o veículo pelo ambiente de navegação. Além disso, o conhecimento adquirido em um ambiente particular é generalizado para diferentes tipos de ambientes. Nota-se, ainda, que uma estratégia diferente daquelas instintivas emerge no sistema. Após aprendizagem, o controlador guia o veículo de forma que este tangencie obstáculos, buscando encontrar uma trajetória que o leve até o alvo.

Nas seções seguintes são definidos o problema de navegação, o veículo, as principais características do ambiente. São descritas as redes neurais nebulosas e seus métodos de aprendizagem.

4.2 Navegação Autônoma: Características do Veículo e Controlador

O problema de navegação autônoma tratado neste capítulo pode ser definido basicamente como um problema de determinação de trajetória. O veículo deve ser capaz de navegar pelo ambiente, atingir um alvo e desviar-se de obstáculos. Outras tarefas podem estar envolvidas, por exemplo: empurrar cargas, explorar ambientes, etc.. Aqui, a tarefa exigida do veículo é tão somente navegar pelo ambiente visando atingir um alvo específico posicionado arbitrariamente sem colidir contra obstáculos. O ambiente de navegação é bidimensional.

Assume-se um veículo equipado com o controlador neural nebuloso descrito em seguida (Figura 4.1). Sensores de obstáculos, alvo, colisão e captura provêm as informações utilizadas pelo controlador. Os sensores de obstáculos (em número de 37) estão dispostos à frente do veículo, distribuindo-se no intervalo de -90° a 90° . Cada um dos sensores de obstáculo determina a distância entre o robô e o obstáculo mais próximo situado na

direção do sensor (ângulo α). O sensor de colisão detecta as colisões contra obstáculos. O sensor de alvo detecta a distância ao alvo e o ângulo β que este está com relação à direção de movimento. O sensor de captura detecta quando o veículo atinge o alvo. A direção de movimento é igual ao ângulo γ medido em relação à abscissa que orienta o ambiente (eixo x). Os sensores capturam estímulos provenientes do ambiente e os transmitem para o controlador do veículo. A cada instante o controlador recebe os sinais captados pelos sensores e determina o ângulo de ajuste para correção da direção de movimento do veículo. Após o veículo ajustar sua direção de movimento, este avança 0,3 unidades de distância.

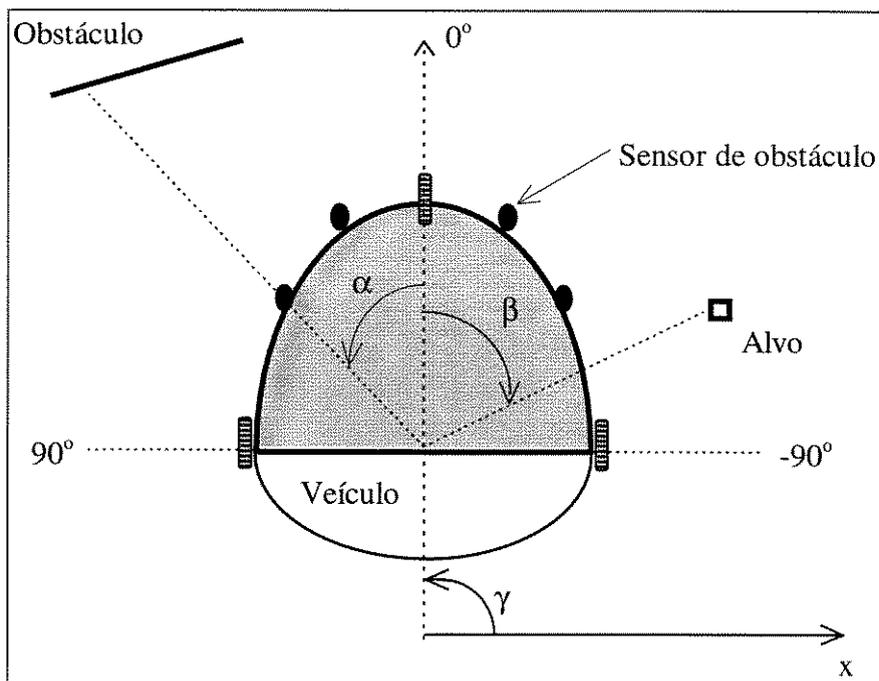


Figura 4.1: Modelo do veículo e seus sensores de obstáculo e alvo.

Duas classes de estratégias principais podem ser adotadas para resolver o problema de navegação de veículos. As estratégias pertencentes a primeira classe planejam a trajetória do veículo, a partir da posição inicial do veículo até a posição do alvo, antes que este comece a navegar. Esta estratégia considera que a topologia do ambiente seja totalmente conhecida. Abordagens tipo busca em grafos pertencem a esta classe de estratégias [Beom92]. Embora a trajetória possa ser determinada de forma ótima, existe a possibilidade de ocorrer colisões quando o ambiente real difere daquele utilizado durante a fase de planejamento. Para que colisões causadas por este motivo não ocorram, algumas abordagens refazem o plano da trajetória sempre que se constata diferenças no ambiente [Li94].

Estratégias pertencentes à segunda classe constroem a trajetória à medida que o veículo navega. Neste caso, o controlador não precisa do conhecimento da topologia do ambiente. Quaisquer alterações são sempre consideradas pelo controlador. Além disso, estas estratégias têm a vantagem de não necessitarem manter a topologia do ambiente em memória e de exigirem menor custo computacional [Tani96]. Pelo fato do controlador determinar ações de controle considerando apenas a situação corrente do veículo

(equivalente aos estímulos captados pelos sensores) esta estratégia é denominada de reativa.

Sob a perspectiva da inteligência artificial, as estratégias reativas são por vezes implementadas projetando-se o controlador com o conhecimento necessário à tarefa de navegação. Neste caso, para cada situação do veículo em seu ambiente, o controlador deve ser capaz de determinar uma ação de controle adequada. É função do projetista encontrar um conjunto restrito de situações que possam ser generalizados e garantam o bom desempenho do veículo [Ishikawa91]. Esta tarefa é reconhecidamente difícil e muitos concedem a este tipo de abordagem pouco crédito [Vershure92].

A principal dificuldade relacionada às estratégias reativas advém da multiplicidade de situações (posições) que o veículo pode deparar-se no ambiente, considerando-se um caso real. Caso o controlador estabeleça uma correspondência entre cada situação distinta e uma ação mais adequada de controle, sem ser capaz de generalizar seu conhecimento, provavelmente sua memória atingirá a exaustão. Para evitar este problema e ao mesmo tempo descartar a necessidade de um projetista, o controlador deve ser capaz de aprender e generalizar conhecimentos essenciais. Conhecimentos essenciais significam aqueles, que se generalizados, podem ser aplicados eficientemente em um grande número de situações distintas. De posse destes conhecimentos o controlador deve tornar-se capaz de guiar o veículo de forma eficiente, ou seja, perfazendo uma trajetória sem colisões até atingir o alvo [Beom95].

O controlador neural nebuloso proposto neste capítulo é inspirado nos seres vivos e fundamenta-se em alguns conceitos principais da teoria da seleção de grupos neurais (TSGN) [Edelman87]. A TSGN sustenta que a compreensão do comportamento inteligente não pode ser obtida independentemente do estudo do sistema nervoso biológico. Conseqüentemente, resultados obtidos na área da neurofisiologia têm sido de alta importância para o desenvolvimento desta teoria. Conforme discutido anteriormente, a TSGN parte do princípio que sistemas nervosos biológicos, antes de interagirem com o ambiente, não comportam conhecimentos relacionadas a tarefas específicas que o animal executará em sua vida. Tais capacidades específicas são adquiridas após inúmeras interações com o ambiente. No entanto, a TSGN admite que o sistema nervoso biológico, antes de suas interações com o ambiente, seja capaz de gerar certos comportamentos instintivos. Estes são gerados por redes neurais particulares desenvolvidas via evolução natural. Portanto, estas redes neurais não aprendem enquanto o animal interage com o ambiente. Na verdade, aprendem após um processo de seleção natural em que várias gerações estão envolvidas. Comportamentos instintivos são essenciais para a vida e desenvolvimento do animal, porém não dizem respeito a comportamentos particulares que ele possa apresentar devido a imposições do ambiente e/ou a situações que possa deparar-se em vida [Kandell91]. Reflexos musculares de recuo e captura e sensações de fome e medo são exemplos de comportamentos instintivos. As primeiras experiências do animal com seu ambiente são manifestações instintivas e, portanto, o conhecimento adquirido em vida pelo sistema nervoso é apoiado e estruturado sobre os conhecimentos correspondentes aos comportamentos instintivos. Após muitas experiências, o animal adquire conhecimento suficiente para exercer capacidades adequadas e particulares a cada situação imposta pelo ambiente.

Três repertórios neurais, atuando em níveis hierárquicos diferentes, compõem o controlador neural nebuloso (Figura 4.2) (Segundo a TSGN redes neurais associadas a comportamentos, responsáveis em produzi-los, ou a atividades específicas são denominadas repertórios). Dois repertórios neurais TS (busca de alvos, “target seaker”) e CA (desvio de obstáculos, “obstacle avoidance”) atuam em um mesmo nível hierárquico e geram comportamentos instintivos. Portanto, elas já possuem o conhecimento necessário para execução de suas tarefas mesmo antes de interagirem com o ambiente. Estas redes neurais nebulosas são construídas via algoritmos genéticos [Oliveira95^b]. Estas técnicas emulam o processo de evolução natural existente na natureza. O repertório TS é capaz de dotar o veículo do comportamento de busca de alvos, mas não considera a presença de obstáculos no ambiente. Portanto, suas decisões, se efetivadas, podem levar o veículo a colidir contra obstáculos. Os sinais de entrada do repertório TS, consistindo apenas dos ângulos ao alvo, provêm do sensor de alvos. O repertório CA gera o outro comportamento instintivo do veículo, a saber, desvio de obstáculos. Este comportamento evita que o veículo colida contra obstáculos, porém não o capacita alcançar alvos. O repertório CA recebe sinais dos sensores de obstáculos. As saídas destes repertórios correspondem a ações de controle para ajuste da direção de movimento do veículo.

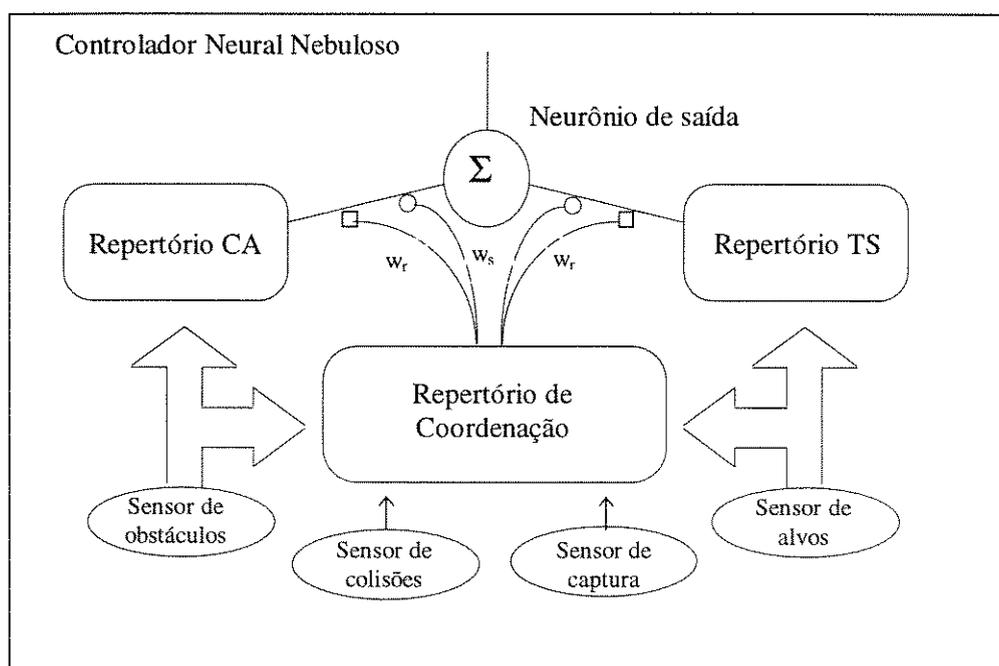


Figura 4.2: Arquitetura do controlador autônomo baseado em redes neurais nebulosas.

Os comportamentos instintivos de busca de alvos e desvio de obstáculos podem tornar-se conflitantes dependendo da configuração do ambiente. Por exemplo, situações nas quais o obstáculo se interpõe entre o alvo e o veículo provocam conflitos nas decisões determinadas pelos repertórios CA e TS. Enquanto o repertório TS determina o avanço do veículo em direção ao alvo, com conseqüente aproximação ao obstáculo, o repertório CA impede que o veículo siga este comportamento pois a colisão é inevitável neste caso. Para gerenciar as atividades dos repertórios CA e TS um repertório de coordenação se faz necessário. Este é formado por uma rede neural nebulosa e por um neurônio nebuloso. O repertório de coordenação atua em um nível hierárquico superior aos demais repertórios. A rede neural nebulosa deste repertório recebe sinais provenientes dos sensores de

obstáculos e do sensor de colisões. O neurônio nebuloso recebe sinais do sensor de captura e do sensor de alvos. Diferentemente dos repertórios CA e TS, a aprendizagem do repertório de coordenação acontece integralmente a partir de interações com o ambiente. O método de aprendizagem é do tipo por reforço e está baseado na teoria clássica do condicionamento [Vershure92], [Haykin94], [Anderson95]. A rede neural nebulosa correspondente ao repertório de coordenação é descrita em seção posterior.

Os sinais de saída gerados pelos repertórios CA e TS são enviados para o único neurônio de saída do controlador. Seus pesos sinápticos são determinados pelo repertório de coordenação. Portanto, este neurônio pondera os sinais provenientes de CA e TS de acordo com o repertório de coordenação, gerando o ângulo que ajusta a direção de movimento do veículo.

A próxima seção focaliza os repertórios CA e TS. Descrevem-se as respectivas redes neurais nebulosas que os compõem e seus métodos de aprendizagem baseados em algoritmos genéticos.

4.3 Repertórios CA e TS

Os repertórios CA e TS utilizados neste trabalho são iguais àqueles descritos no trabalho [Oliveira95^b]. Nesta seção descreve-se brevemente as arquiteturas e estratégias de aprendizagem adotadas para seu desenvolvimento.

4.3.1 Arquiteturas das Redes Neurais Nebulosas

Redes neurais nebulosas capazes de representar o conhecimento na forma de regras do tipo se-então compõem os repertórios CA e TS. Suas arquiteturas são iguais àquela da rede N2 descrita na seção 3.2. Para efeitos didáticos, as idéias básicas serão revistas em seguida. O modelo de rede neural nebulosa empregado é baseado no mecanismo de inferência nebulosa definido por [Figueiredo95^b]:

entrada:	$X_1 \text{ é } A_1 \text{ e } \dots X_M \text{ é } A_M$
	$\text{Se } X_1 \text{ é } A_1^i \text{ e } \dots X_j \text{ é } A_j^i \text{ e } \dots X_M \text{ é } A_M^i \text{ então } y \text{ é } g^i$
	\dots
regras:	$\text{Se } X_1 \text{ é } A_1^i \text{ e } \dots X_j \text{ é } A_j^i \text{ e } \dots X_M \text{ é } A_M^i \text{ então } y \text{ é } g^i$
	\dots
	$\text{Se } X_1 \text{ é } A_1^N \text{ e } \dots X_j \text{ é } A_j^N \text{ e } \dots X_M \text{ é } A_M^N \text{ então } y \text{ é } g^N$
	\dots
saída:	$y = g$

em que: X_j é uma variável nebulosa, A_j e A_j^i são conjuntos nebulosos nos respectivos espaços de entrada, $j = 1, \dots, M$; $i = 1, \dots, N$, y é a variável real definida no espaço de saída e g e $g^i \in \mathfrak{R}$.

O resultado da inferência nebulosa é obtido após três estágios básicos: comparação, agregação de antecedentes e agregação de regras.

A arquitetura da rede neural nebulosa N2 é não recorrente. O processamento neural equivale ao processo de inferência nebulosa. Para o problema tratado neste capítulo os operadores sinápticos definidos para os neurônios da segunda camada são iguais a t-norma mínimo. Todos os demais parâmetros são iguais aos da rede N2 (Figura 4.3).

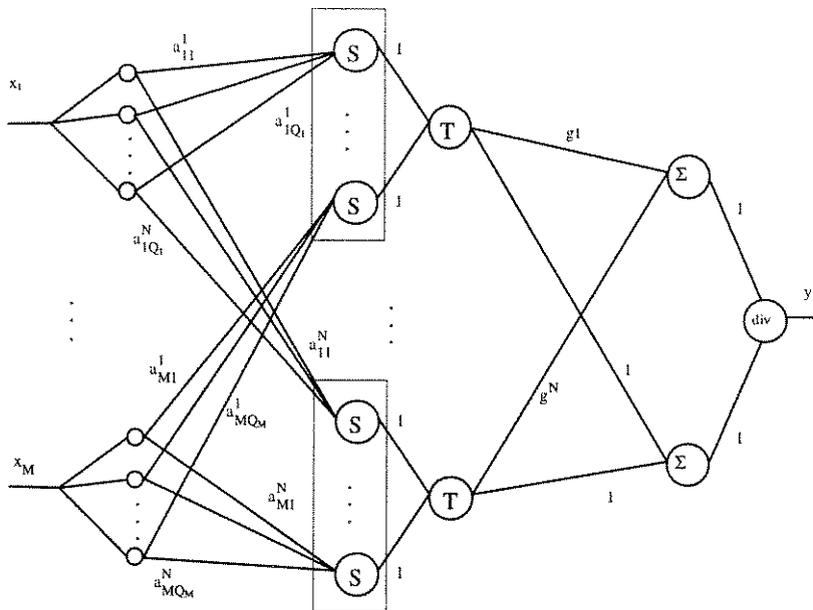


Figura 4.3: Rede neural nebulosa

O método de aprendizagem adotado está baseado nos algoritmos genéticos. Este método emula o processo de evolução natural que gera os sistemas nervosos biológicos análogos, ou seja, aqueles responsáveis pelos comportamentos instintivos.

4.3.2 Algoritmos Genéticos e Emergência de Comportamentos Instintivos

Sistemas inteligentes artificiais baseados em redes neurais frequentemente adotam métodos de aprendizagem supervisionados para solucionar tarefas de controle. Configurações típicas controlador-planta necessitam de informações entrada/saída. Caso não existam exemplos, nem supervisor, a aprendizagem torna-se infactível.

A maioria dos métodos não-supervisionados existentes está associada a problemas do tipo reconhecimento de padrões e usualmente não é capaz de determinar estratégias de controle. Em problemas de controle o método de aprendizagem deve descobrir uma estratégia de decisão que satisfaça os requisitos de desempenho e mantenha o sistema dentro de limites operacionais.

O processo de evolução natural é capaz de promover a evolução de organismos primitivos bem como de suas habilidades. Este processo em sua versão computacional pode ser empregado na solução de problemas de controle sem supervisão.

Dois mecanismos governam o processo de evolução: seleção natural e recombinação genética. Algoritmos genéticos (GA's) emulam estes mecanismos. Em essência, algoritmos genéticos equivalem a procedimentos de busca global [Goldberg89].

Tal qual em seu análogo natural, GA's operam sobre populações de indivíduos. As características operacionais de cada indivíduo estão codificadas na forma de um vetor denominado cromossomo. O mecanismo de seleção natural e uma função objetivo estabelecem um ambiente de competição. Os indivíduos são avaliados por meio da função objetivo e, de acordo com seus desempenhos, são selecionados para compor uma nova população de indivíduos (geração). Indivíduos com melhores resultados possuem maior probabilidade de seleção. Cada par de indivíduos selecionados gera um indivíduo da nova geração. O mecanismo de recombinação genética estabelece quais características dos genitores são herdadas pelos novos indivíduos. Os mecanismos de seleção natural e recombinação genética agem sucessivamente sobre cada nova população gerada. Após varias gerações, indivíduos evoluem e tendem a exibir um melhor desempenho. A Figura 4.4 mostra o ciclo fundamental que rege os algoritmos genéticos. Neste ciclo novas populações surgem sucessivamente à medida que os mecanismos de evolução agem sobre cada geração.

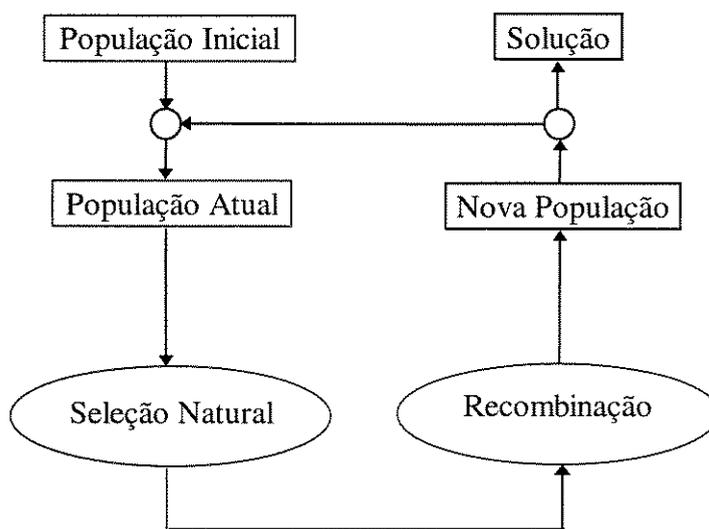


Figura 4.4: Ciclo fundamental do GA.

Há diversas técnicas utilizadas pelo mecanismo de recombinação genética bem como na formação da nova população. Basicamente, o mecanismo de recombinação genética divide os cromossomos de cada genitor (nos quais estão registradas suas características na forma de vetores). As partes formadas são recombinadas para a geração de um novo indivíduo (cruzamento) (Figura 4.5). Além disso, um operador de mutação pode alterar características herdadas modificando de forma aleatória códigos presentes nos cromossomos dos novos indivíduos.

Para aprendizagem das redes neurais nebulosas correspondentes aos repertórios CA e TS utilizou-se GA's [Oliveira95^b]. Os pesos sinápticos de cada rede neural nebulosa são

determinados durante a aprendizagem. A evolução da população mantém o melhor indivíduo de geração para geração (modelo elitista). A codificação real/inteiro é adotada na formação dos cromossomos. Os parâmetros determinados na aprendizagem são: conseqüentes das regras nebulosas e funções de pertinência.

A aprendizagem das redes neurais nebulosas correspondentes aos repertórios TS e CA é feita de forma independente (pois estes geram comportamentos independentes). Cada um destes repertórios aprende em um período próprio. Durante o respectivo período de aprendizagem, o repertório aprendiz controla o veículo sem interferência do outro (o repertório de coordenação também não é considerado durante a aprendizagem dos repertórios TS e GA). Sendo assim, a saída de qualquer dos repertórios, durante seu respectivo período de aprendizagem, ajusta a direção de movimento do veículo. Portanto, a trajetória do veículo, nestes períodos, espelha somente o comportamento associado ao repertório aprendiz. Por este motivo, os treinamentos foram executados em ambientes adequados ao tipo de aprendizagem requerido. Para aprendizagem do comportamento de busca de alvos o ambiente não possui obstáculos (somente os limites do ambiente). Por sua vez, o comportamento de desvio de obstáculos é desenvolvido em ambiente sem alvo. Esta estratégia de aprendizagem é conveniente para formação dos repertórios neurais relativos a cada um dos comportamentos instintivos.

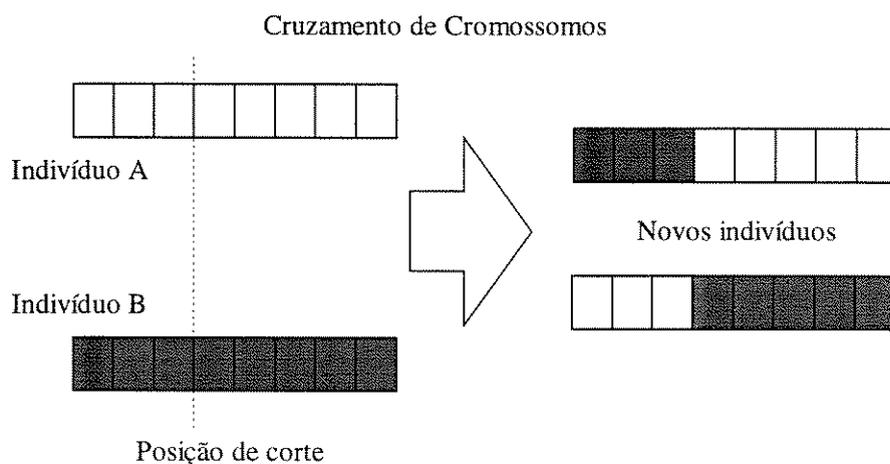


Figura 4.5: Recombinação genética: cruzamento de cromossomos.

Cada indivíduo (rede neural nebulosa) é avaliado de acordo com seu desempenho no controle do veículo. A função objetivo que avalia o comportamento de busca de alvos tem como argumentos o número de alvos alcançados e a distância média do veículo ao alvo. Para o caso do comportamento de desvio de obstáculos o desempenho do indivíduo é medido de acordo com a distância média entre o veículo e o obstáculo mais próximo.

Após os respectivos períodos de aprendizagem, cada um dos repertórios CA e TS é capaz de estabelecer os comportamentos instintivos a que estão associados: desvio de obstáculos e busca de alvos, respectivamente. Entretanto, conforme já observado, os comportamentos independentes de busca de alvos ou desvio de obstáculos não possibilitam que o veículo desempenhe o comportamento desejado em um ambiente com obstáculos e alvos. O repertório de coordenação que gerencia os comportamentos gerados por TS e CA, é o foco da próxima seção.

4.4 Repertório de Coordenação e Saída do Controlador

O repertório de coordenação é constituído por uma rede neural nebulosa e por um neurônio nebuloso. A rede neural recebe sinais provenientes dos sensores de obstáculos. Sua função é avaliar, sob a perspectiva do risco de colisão, a disposição dos obstáculos no ambiente. Sua saída pondera sinais provenientes de CA e TS visando evitar colisões. O neurônio recebe o sinal do sensor de alvos com o qual determina qual comportamento deve ser intensificado para que o veículo alcance o alvo. As saídas do repertório de coordenação estabelecem pesos sinápticos do neurônio de saída e, conseqüentemente, a ponderação dos comportamentos instintivos gerados pelos repertórios CA e TS.

A estratégia adotada para aprendizagem do repertório de coordenação baseia-se na teoria do condicionamento clássico. Para uma melhor compreensão do método de aprendizagem empregado, descrevem-se em seguida, resumidamente, os conceitos básicos relativos a esta teoria. Após, descrevem-se a rede neural nebulosa e o modelo de neurônio nebuloso utilizado na formação de sua arquitetura [Figueiredo96].

O condicionamento animal clássico é um fenômeno intimamente relacionado à aprendizagem e freqüentemente observado em sistemas nervosos biológicos [Kandell91]. Há dois tipos principais de estímulos sobre os quais apóia-se o condicionamento animal: condicionado e não-condicionado. O estímulo não-condicionado determina uma resposta particular (reflexo não-condicionado) sempre que ele é recebido pelo sistema nervoso. Em princípio, o estímulo condicionado não possui qualquer associação com um comportamento particular. Entretanto, se estímulos condicionados e não-condicionados ocorrem concomitantemente, então o sistema nervoso estabelece uma associação entre estímulos condicionados e reflexos não-condicionados. Ou seja, após a conclusão do condicionamento, o estímulo condicionado é capaz de provocar o reflexo condicionado (muito próximo do reflexo não-condicionado) mesmo na ausência do estímulo não-condicionado.

Em [Vershure92] propõe-se uma rede neural não-supervisionada cuja aprendizagem é baseada no condicionamento clássico e aprende a guiar veículos autônomos. Um dos repertórios recebe estímulos provenientes dos sensores de obstáculos e do sensor de colisão. Os sinais de colisão são estímulos não-condicionados. Se uma colisão é detectada então todos os neurônios deste repertório são ativados. Os pesos sinápticos correspondentes às conexões com os sensores de obstáculos são ajustadas de acordo com uma variação da regra de Hebb [Hebb49].

Na presente proposta, os sinais emitidos pelo sensor de colisão e pelo sensor de captura, são também definidos como sinais não-condicionados que, além de outras funções, determinam a aprendizagem da rede neural nebulosa e do neurônio nebuloso do repertório de coordenação. Sempre que ocorre uma colisão, o sensor de colisão envia um sinal para a rede neural nebulosa ativando um ciclo de aprendizagem da rede neural nebulosa. Em cada um destes ciclos, seus pesos sinápticos e sua arquitetura são ajustados. Após algumas colisões, a rede neural nebulosa associa classes de padrões capturados pelos sensores de obstáculos a situações de colisão iminente. A partir desta associação a rede neural nebulosa é capaz de gerar sinais de controle adequados à correta ponderação dos comportamentos de busca de alvos e desvio de obstáculos. De forma análoga, toda vez que o neurônio recebe um sinal do sensor de captura, sempre que o veículo atinge o alvo,

um ciclo de aprendizagem é ativado. Após alguns ciclos, o neurônio aprende como ponderar cada comportamento instintivo.

O sinal emitido devido à ocorrência de uma colisão, também induz um reflexo não-condicionado. Sempre que este sinal é emitido, o veículo volta-se para uma direção 180° com relação à sua direção de movimento.

O modelo geral de neurônio nebuloso (Figura 4.6), que compõe a arquitetura da rede neural nebulosa proposta neste trabalho, é uma generalização daquele apresentado em [Figueiredo95^a].

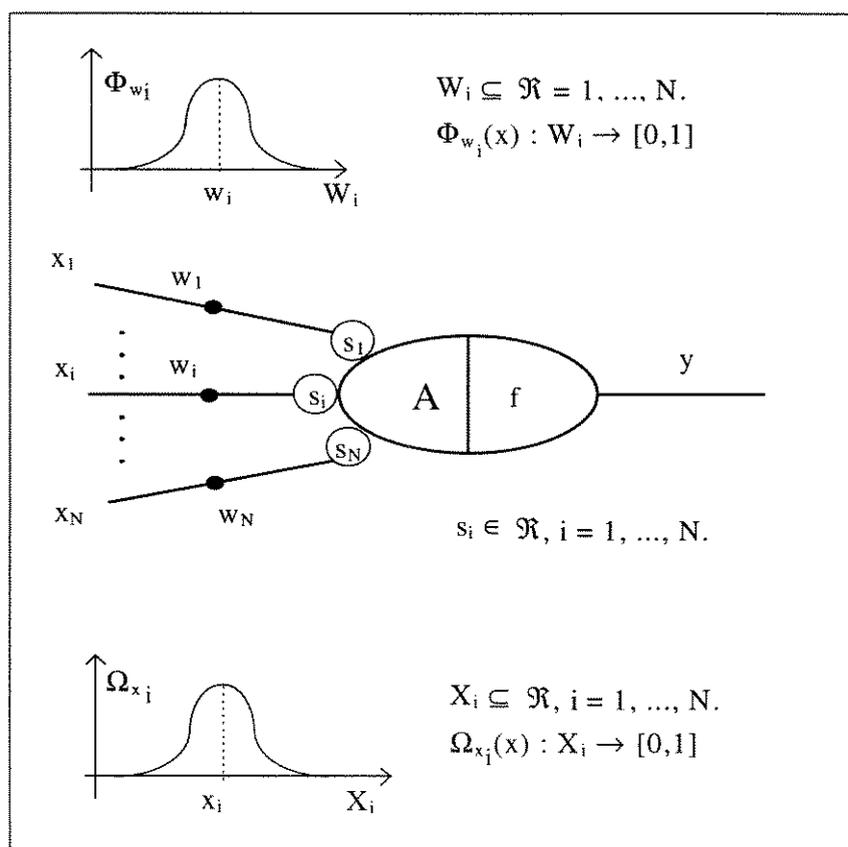


Figura 4.6: Modelo de neurônio nebuloso.

Suas características capacitam ao repertório desempenhar sua tarefa de controle autônomo. Neste modelo entradas x_i e pesos sinápticos w_i , $i = 1, \dots, N$, são parâmetros de conjuntos nebulosos $\Omega_{x_i}(x)$ e $\Phi_{w_i}(x)$, respectivamente (na Figura 4.6 os parâmetros estabelecem os valores modais dos conjuntos). O processamento sináptico equivale à duas operações sucessivas: os conjuntos $\Omega_{x_i}(x)$ e $\Phi_{w_i}(x)$ são comparados via um operador monotônico $\Delta(\cdot)$ e, em seguida, o valor encontrado na comparação é ponderado pelo peso sináptico s_i através da t-norma $T_2(\cdot)$. Tem-se então dois pesos sinápticos. O peso sináptico w_i representa a memória com relação a um padrão induzido pelas sucessivas entradas, enquanto s_i tem a função de estabelecer com qual intensidade a sinapse modifica a entrada. Após o processamento sináptico, um operador de agregação nebuloso $A(\cdot)$ [Yager94] agrega as entradas. A função de ativação $f(\cdot)$ determina a saída do neurônio a partir do sinal resultante da agregação. O modelo é descrito formalmente pela expressão (4.1):

$$y = f \left(\mathbf{A} \left(\mathbf{T}_2 \left(\Delta \left(\Phi_{w_i}(x), \Omega_{x_i}(x), s_i \right) \right) \right) \right) \quad (4.1)$$

A rede neural nebulosa do repertório de coordenação possui duas camadas e sua arquitetura é não-recorrente (Figura 4.7). Para sua descrição, bem como do algoritmo de aprendizagem, utiliza-se a seguinte notação:

- número de sensores de obstáculos : N
- número de colisões : c
- número de neurônios na primeira camada : Q
- vetor de distâncias a obstáculos : $\mathbf{v} = (v_1 \dots v_k \dots v_N)$

As componentes v_k , $k = 1, \dots, N$, do vetor de distância de obstáculos \mathbf{v} constituem as entradas de cada neurônio da primeira camada da rede neural nebulosa. A cada instante de tempo t a rede neural nebulosa recebe um vetor $\mathbf{v}(t)$ proveniente do sensor de distância.

Além destes, os pesos sinápticos com relação à primeira camada são definidos por:

- peso sináptico de comparação (entrada i e neurônio j) : w_{ij}
- peso sináptico de ponderação (entrada i e neurônio j) : S_{ij}

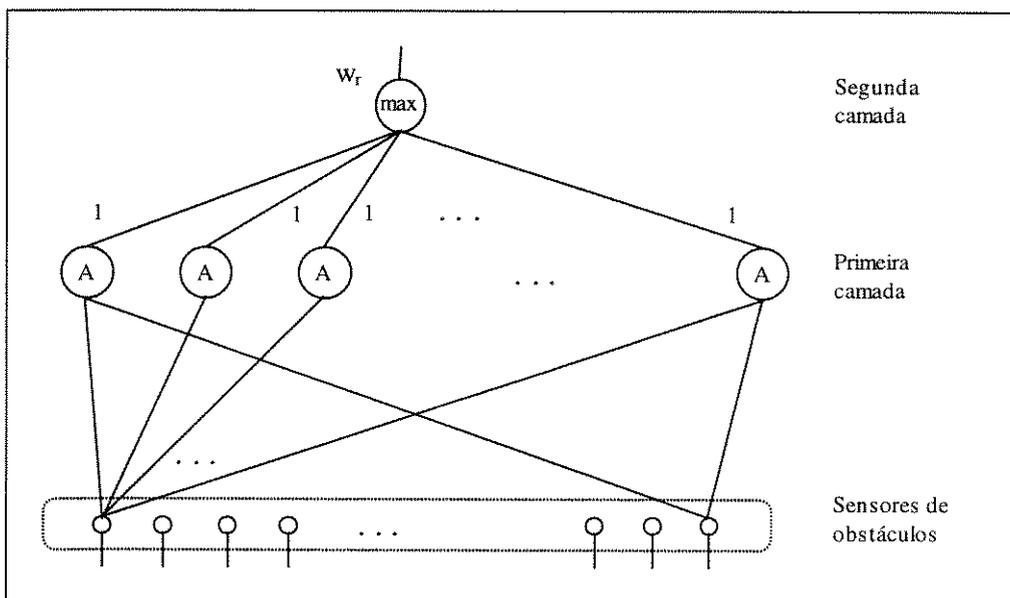


Figura 4.7: Arquitetura da rede neural nebulosa.

O número de neurônios na primeira camada é estabelecido no decorrer do processo de aprendizagem. Cada um destes neurônios conecta-se com todos os sensores de obstáculos, ou seja, o número de entradas de cada neurônio da primeira camada é igual ao número de sensores. O modelo destes neurônios é igual àquele descrito pela equação (4.1). Para aplicação no problema de navegação, as seguintes definições são consideradas: $A(\cdot)$ é uma s -norma máximo, T_2 é a t -norma produto, $f(\cdot)$ é a função identidade e $\Delta(\cdot)$ é a medida de possibilidade dada por:

$$\Delta(G(x), H(x)) = \mathbf{S} \left(\mathbf{T}_1(G(x), H(x)) \right);$$

em que: $S(\cdot)$ é a s-norma máximo, $T_1(\cdot)$ a t-norma mínimo e U o universo de discurso dos conjuntos considerados (ver (2.21) e (2.25)). Além disso, os conjuntos $\Omega_{x_i}(x)$ e $\Phi_{w_i}(w)$ possuem perfis triangulares (isósceles) sendo a largura da base fixa (nos resultados de simulação igual a 3).

Desta forma, a saída y_j correspondente ao j -ésimo neurônio nebuloso da primeira camada é computada de acordo com (4.2):

$$y_j = \bigwedge_{i=1}^N \left(s_{ij} \left(S_{x \in U} \left(T_1 \left(\Phi_{w_{ij}}(x), \Omega_{v_i}(x) \right) \right) \right) \right) \quad (4.2)$$

Um único neurônio compõe a segunda camada. Ele está conectado a todos os neurônios da primeira camada. Ele é definido de acordo com o modelo utilizado pela rede N1 (Figura 3.2). Suas sinapses não alteram as entradas. Elas são modeladas por pesos sinápticos unitários e por um operador sináptico igual ao produto algébrico. As funções de agregação e de ativação são definidas pela s-norma máximo e pela função identidade, respectivamente. Portanto, sua saída w_r (também saída da rede neural nebulosa) corresponde ao maior valor computado pelos neurônios da primeira camada tal como mostra a equação (4.3):

$$w_r = \bigvee_{j=1}^Q (y_j); \quad (4.3)$$

em que: $S(\cdot)$ é a s-norma máximo.

O exemplo apresentado em seguida mostra a motivação por detrás dos mecanismos de ajuste empregados durante a aprendizagem da rede neural nebulosa. Suponha que em sua trajetória o veículo se depare com duas situações distintas. Considere que os sensores de obstáculos tenham detectado padrões de distância representados pelos vetores A e B (Figura 4.8). Os valores baixos relativos às componentes centrais destes vetores indicam que um obstáculo encontra-se muito próximo e à frente do veículo. Considere que o controlador tenha aprendido associar o vetor A como uma situação crítica e, além disso, tenha aprendido como ativar o repertório de colisão para que este evite a colisão iminente. Suponha que o veículo, após passar pela situação caracterizada por A, atinja a posição caracterizada por B. Neste instante a rede neural nebulosa é capaz de reconhecer similaridades relevantes entre os padrões A e B (componentes centrais) sob a perspectiva do desvio de obstáculos. Neste caso, o repertório de coordenação é capaz de ponderar convenientemente as influências dos comportamentos produzidos pelos repertórios CA e TS. Note que as redes do tipo Kohonen classificam padrões A e B como diferentes, conseqüentemente não seriam capazes de reconhecer a situação B como uma situação crítica e provavelmente não impediria que a colisão se sucedesse [Tani96].

A	18	17	9	1	1	1	12	104	105	109
B	109	72	52	2	1	1	2	15	19	18

Figura 4.8: Vetores A e B representando situações de colisão iminente.

Conforme descrito anteriormente, os pesos sinápticos w_{ij} e s_{ij} , correspondentes à sinapse da entrada i e neurônio j , são determinados ao longo da aprendizagem. A estratégia de aprendizagem da rede neural nebulosa, baseada no condicionamento animal, estabelece que os pesos sinápticos são ajustados se ela recebe um sinal não-condicionado emitido pelo sensor de colisão (sempre que uma colisão é detectada).

O algoritmo para aprendizagem da rede neural nebulosa é descrito em seguida (START é uma constante predefinida e utilizada para inicializar o peso sináptico $s_{ij}(t)$ e LIMIAR, também uma constante predefinida, é utilizada como um parâmetro de comparação cuja função é possibilitar ou não o ajuste do peso sináptico):

1- Inicialização: Inicialize o número de colisões e o número de neurônios na primeira camada fazendo $c(0) = 0$ e $N(0) = 0$.

2- Ajuste: Se uma colisão ocorre então faça:

2.1- Se $c(t) = 0$ então adicione o primeiro neurônio na primeira camada da rede neural nebulosa e inicialize seus pesos sinápticos:

$$c(t + 1) = c(t) + 1$$

$$Q(t + 1) = Q(t) + 1$$

Para $i = 1$ até N e $j = Q(t + 1)$ faça:

$$s_{ij}(t + 1) = \text{START}$$

$$w_{ij}(t + 1) = v_i$$

Fim

(Um neurônio relativo à segunda camada também deve ser considerado para compor a arquitetura da rede neural nebulosa.)

Prossiga a aprendizagem em 3.

2.2- Se $c(t) \neq 0$ então faça:

Para $j = 1$ até $Q(t)$ faça: (para cada neurônio da primeira camada)

$$\alpha = c(t) - j + 1$$

Para $i = 1$ até N faça: (para cada sinapse do neurônio)

$$\text{SIMIL} = 0$$

Para $k = 1$ até N faça: (para cada entrada do neurônio)

$$\text{AUX} = \Delta \left(\Phi_{w_{ij}}(x), \Omega_{v_k}(x) \right)$$

Se $\text{SIMIL} < \text{AUX}$ então

$$\text{SIMIL} = \text{AUX}$$

$$\text{COMP} = k$$

Fim

Fim

Se $\text{SIMIL} > \text{LIMIAR}$ então execute os ajustes:

$$w_{ij}(t + 1) = w_{ij}(t) + (v_{\text{COMP}}(t) - w_{ij}(t)) / (1 + s_{ij}(t))$$

$$s_{ij}(t + 1) = s_{ij}(t) + (\text{SIMIL} - s_{ij}(t)) / (\alpha + 1)$$

Fim

Fim

Fim

$$c(t + 1) = c(t) + 1$$

$$Q(t + 1) = Q(t) + 1$$

Para $i = 1$ até N e $j = Q(t + 1)$ inicialize os pesos sinápticos do novo neurônio:

$$s_{ij}(t + 1) = \text{START}$$

$$w_{ij}(t + 1) = v_i$$

Fim

Prossiga a aprendizagem em 3

3- Retorne para 2

Observa-se, a partir do algoritmo de aprendizagem, que os pesos sinápticos w_{ij} e s_{ij} são ajustados se e somente se duas condições são satisfeitas. A primeira refere-se à existência de colisão. A segunda refere-se ao maior valor assumido por AUX que deve ser maior que LIMAR.

Portanto, se as duas condições forem satisfeitas simultaneamente, então valem as expressões de ajuste (4.4) e (4.5), transcritas do algoritmo de aprendizagem descrito anteriormente:

$$w_{ij}(t + 1) = w_{ij}(t) + (v_{\text{comp}}(t) - w_{ij}(t)) / (1 + s_{ij}(t)) \quad (4.4)$$

$$s_{ij}(t + 1) = s_{ij}(t) + (\text{SIMIL} - s_{ij}(t)) / (\alpha + 1) \quad (4.5)$$

As equações de ajuste (4.4) e (4.5) estão coerentes com o papel dos pesos sinápticos w_{ij} e s_{ij} , respectivamente. Em cada ajuste, w_{ij} aproxima-se de uma componente do vetor \mathbf{v} , refletindo o processo de indução definido pelas entradas apresentadas. Já o peso sináptico s_{ij} tende a representar uma média dos valores de comparação SIMIL, refletindo a noção de intensidade da sinapse. Observa-se que o ajuste de w_{ij} pode ser influenciado por qualquer uma das entradas e não apenas por v_i . Este tipo de ajuste é estudado em [Edelman87].

Além dos ajustes, o estímulo não-condicionado provoca alteração na estrutura da rede neural nebulosa. A cada colisão um novo neurônio é adicionado à sua primeira camada. Iguala-se cada novo peso sináptico w_{ij} à respectiva componente v_i do vetor de entradas da rede neural. O peso sináptico s_{ij} é inicializado em START.

Nos resultados de simulações apresentados em seguida, os parâmetros START e LIMIAR são constantes e iguais a 0,1 e 0,5, respectivamente.

Além da rede neural nebulosa, faz parte do repertório de coordenação uma estrutura neural constituída de um único neurônio nebuloso. Este é definido tal qual em (4.2), ou seja, é definido de forma semelhante aos neurônios da primeira camada da rede neural nebulosa. Ele difere daqueles neurônios com respeito a alguns pontos. Ele possui somente uma entrada v_1 , a qual é proveniente do sensor de alvos. Portanto, tem somente dois pesos sinápticos, w_1 e s_1 . Nos resultados de simulação, as bases dos conjuntos nebulosos $\Omega_v(x)$ e $\Phi_{w_1}(x)$ são iguais a 10. O operador de agregação $A(.)$ é definido como a soma algébrica. Sua saída w_s é, portanto (4.6):

$$w_s = s_1 \left(\sum_{x \in U} \left(T \left(\Phi_{w_1}(x), \Omega_{v_1}(x) \right) \right) \right) \quad (4.6)$$

Um ciclo de aprendizagem se inicia sempre que o veículo atinge o alvo e, conseqüentemente, um sinal não-condicionado é emitido pelo sensor de captura. Os pesos sinápticos w_1 e s_1 são ajustados de acordo com o seguinte algoritmo:

1- Inicialização: Inicialize os pesos w_1 e s_1 .

$$s_1 = 0$$

$$w_1 = 0$$

2- Ajuste: Se o veículo alcança o alvo, então faça:

$$\beta = \Delta \left(\Phi_{w_1}(w), \Omega_{v_1}(x) \right)$$

Se $\beta > 0$ então:

$$w_1(t+1) = w(t) + (v_1(t) - w_1(t))/(1 + s_1(t))$$

$$s_1(t+1) = s_1(t) + (1 - s_1(t))/(\beta + 1)$$

Fim

Prossiga em 3.

3- Retorne para 2.

A saída do controlador u é determinada pelo neurônio de saída do repertório (ver Figura 4.2). Ele está conectado às saídas dos repertórios TS e CA, denominadas U_{TS} e U_{CA} , respectivamente. Este neurônio é definido de acordo com o modelo apresentado na Figura 3.2. Para o caso do problema de navegação, são definidos o operador sináptico, o

operador de agregação e a função de ativação como sendo o produto algébrico, a soma algébrica e a função identidade, nesta ordem. Os pesos sinápticos de suas conexões são determinados via repertório de coordenação. Portanto, o sinal de controle u que estabelece o ajuste na direção de movimento do veículo é tal como mostra a equação (4.7):

$$u = (1 - w_r)U_{TS} + (1 - w_s) w_r U_{CA}. \quad (4.7)$$

4.5 Resultados de Simulação

Nesta seção apresentam-se os resultados de simulação que permitem avaliar o desempenho do veículo em diferentes ambientes e situações. Nas figuras que se seguem, os alvos estão representados por pontos pretos e obstáculos são segmentos de reta. Colisões são representadas por (\Rightarrow). Durante as simulações somente um alvo está presente no ambiente. Sempre que o veículo atinge o objetivo corrente, um outro alvo é posicionado em diferente posição.

Os resultados de simulação referentes ao desempenho dos repertórios TS e CA também são considerados em seguida. Tal como já foi relatado, cada rede neural evolui durante um diferente período de aprendizagem. Sua saída é então considerada como a própria saída do controlador, ou seja, o comportamento que gera reflete-se na dinâmica do veículo. Os treinamentos foram executados em ambientes adequados ao tipo de aprendizagem requerido: o comportamento de busca de alvos é desenvolvido em ambiente sem obstáculos (a menos dos seus limites). Para desenvolvimento do comportamento de desvio de obstáculos o ambiente não apresenta alvo. Sob esta estratégia de aprendizagem os repertórios neurais evoluem. Após a evolução, os repertórios CA e TS são capazes de estabelecer os comportamentos instintivos de desvio de obstáculos e busca de alvos, respectivamente. Entretanto, os comportamentos independentes de busca de alvos ou desvio de obstáculos não possibilitam que o veículo desempenhe o comportamento desejado em um ambiente com obstáculos e alvos.

Após a aprendizagem, o repertório TS passa a gerar o comportamento de busca de alvos. A Figura 4.9 ilustra este comportamento. Note que o repertório TS, ao controlar o veículo não evita colisões contra obstáculos (Figura 4.10). De forma semelhante, o repertório CA, após a evolução e no controle do veículo, é eficiente em evitar colisões, mas não é capaz de guiar o veículo ao alvo. Tanto em um ambiente sem alvos (Figura 4.11) ou na presença dele (Figura 4.12) o veículo percorre as mesmas trajetórias.

Para que o veículo possa navegar eficientemente em ambientes com alvos e obstáculos, o repertório de coordenação deve entrar em atividade. Inicialmente a rede neural nebulosa deste repertório passa por um período de aprendizagem. A Figura 4.13 mostra o comportamento do veículo sendo guiado pelo sistema de controle autônomo ainda em fase de aprendizagem. Os repertórios CA e TS são, então, gerenciados pelo repertório de coordenação. Na condição inicial o veículo é posicionado no ambiente de forma que um obstáculo se interponha entre ele e o alvo. Esta estratégia visa garantir que o controlador se depare com situações críticas de colisão permitindo a aprendizagem da rede neural nebulosa. Inicialmente, antes que alguma colisão ocorra, o repertório TS determina a direção de movimento do veículo. Esta estratégia garante que o veículo explore o

ambiente antes que a aprendizagem da rede neural se inicie efetivamente, ou seja, antes que ela receba o primeiro sinal não-condicionado.

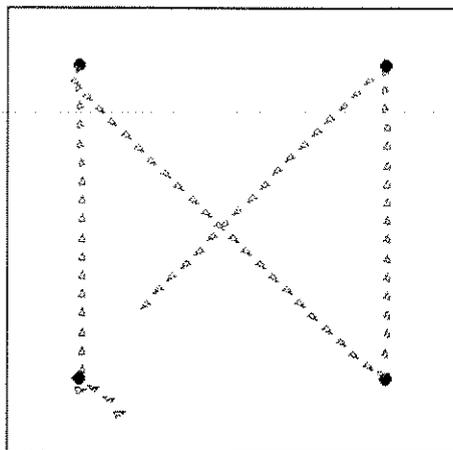


Figura 4.9: Comportamento de busca de alvos gerado pelo repertório TS.

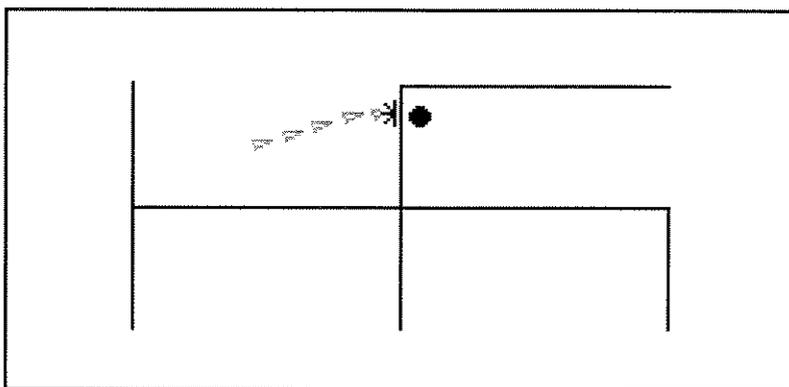


Figura 4.10: O repertório TS leva o veículo a uma colisão inevitável em ambiente com obstáculos.

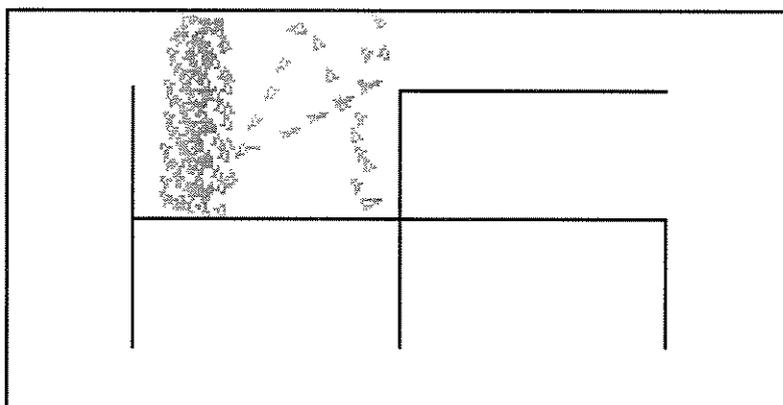


Figura 4.11: Comportamento de desvio de obstáculos desempenhado pelo repertório CA.

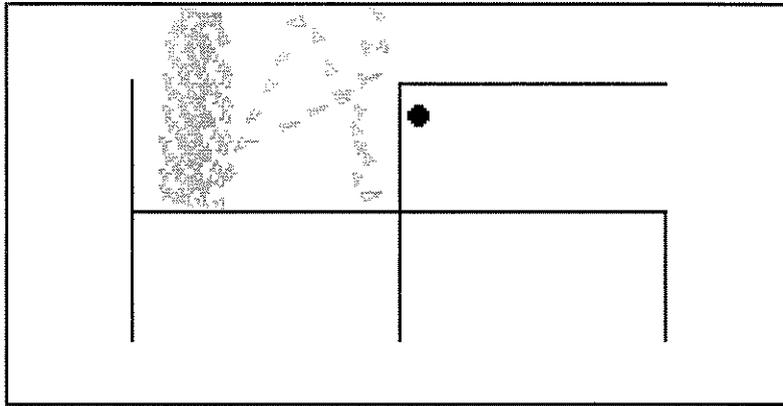


Figura 4.12: O repertório CA não reconhece alvos e o veículo não avança de encontro ao seu objetivo.

Após a primeira colisão, a direção de movimento é ajustada de acordo com (4.7). Desde que o repertório de coordenação não tem experiências em atividades de controle, ele não reconhece possíveis situações de colisão iminente, tal como aquela que o ambiente da Figura 4.13 apresenta. Conseqüentemente, o controlador guia o veículo em direção ao alvo e contra o obstáculo, ocasionando uma primeira colisão. O repertório de coordenação é estimulado por um sinal não-condicionado proveniente do sensor de colisões. Este provoca o reflexo não-condicionado que volta o veículo para uma direção oposta a direção corrente, ou seja, a nova direção é 180° da anterior. Além disso, o mesmo sinal desencadeia um ciclo de aprendizagem da rede neural nebulosa. Ela ajusta seus pesos sinápticos procurando identificar padrões que permitam o reconhecimento de situações de colisão iminente. Se a rede não adquire o completo conhecimento que a capacite ponderar os comportamentos instintivos gerados por CA e TS, então o veículo permanece passível de novas colisões. Em cada colisão é gerado um sinal não-condicionado. Ele é seguido do reflexo não-condicionado correspondente e de um ciclo de aprendizagem da rede neural nebulosa. Após algumas colisões, o repertório de coordenação adquire conhecimento para gerenciar os comportamentos de desvio de obstáculos e busca de alvos. Ele torna-se, então, capaz de guiar o veículo para o alvo perfazendo uma trajetória sem colisões.

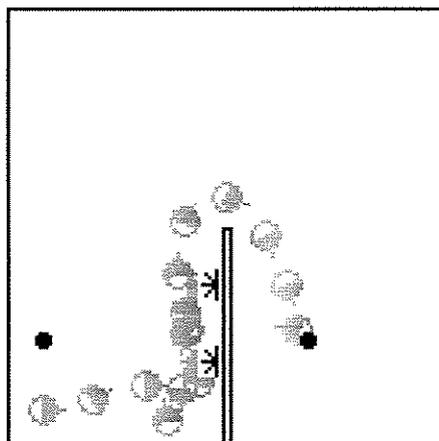


Figura 4.13: Trajetória do veículo sob controle do sistema nebuloso ainda em fase de aprendizagem.

As Figura 4.14 e Figura 4.15 mostram o veículo navegando em ambientes distintos daqueles utilizados durante a aprendizagem. Mesmo nestes casos, nos quais o sistema neural autônomo ainda não desenvolveu qualquer tipo de interação, o veículo navega com sucesso (sem colisões e atingindo o alvo). Esta é uma forte evidência da capacidade de generalização de que é dotado o repertório de coordenação. Isto significa que as informações relevantes para solucionar a tarefa de navegação são capturadas durante sua aprendizagem e implicitamente armazenadas em sua estrutura. Uma importante propriedade desta abordagem refere-se à sua capacidade de aprender uma estratégia de controle geral. Ela se adequa a ambientes quaisquer, ou seja, ela não está restrita a um ambiente específico. Além disso, diferentemente de grande parte das abordagens utilizadas em navegação autônoma de veículos, nenhum conhecimento é inserido durante a fase de projeto do controlador. A aprendizagem das redes neurais ocorre por meio de métodos que ou emulam a evolução natural ou estão fundamentados na teoria clássica do condicionamento animal. Esta característica é de essencial importância, em particular, no caso da navegação autônoma e, de forma mais geral, no caso de controle autônomo. Observa-se ainda, a partir dos resultados de simulação, que dois comportamentos emergem como resultado do processo de aprendizagem: previsão e tangenciar paredes. Note que estes comportamentos não foram considerados em quaisquer das fases da concepção do controlador.

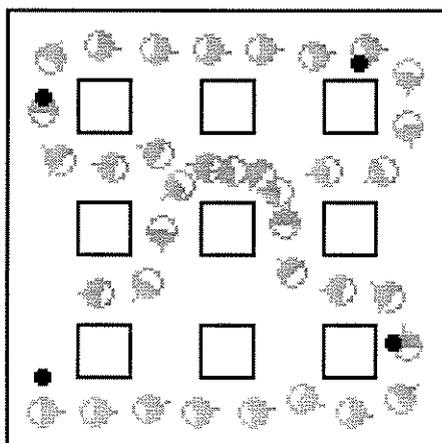


Figura 4.14: O veículo navega eficientemente em um ambiente distinto do utilizado na aprendizagem.

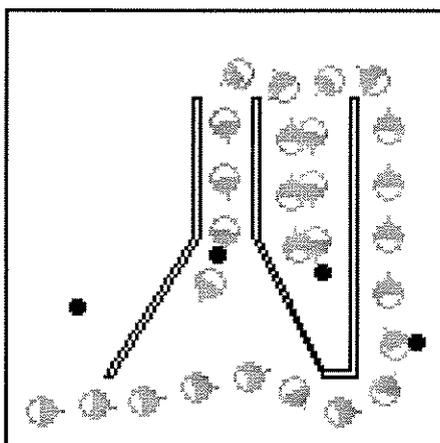


Figura 4.15: As habilidades do veículo reproduzem-se em ambientes complexos.

4.6 Resumo

Neste capítulo descreveu-se uma nova proposta de controlador autônomo para navegação de veículos. A concepção do controlador foi inspirada no comportamento dos animais e fundamentada nos resultados recentes da neurofisiologia. Para o desenvolvimento do modelo foram utilizadas técnicas de inteligência computacional, a saber, redes neurais nebulosas e algoritmos genéticos.

O sistema neural nebuloso que compõe o controlador é hierarquizado. Em um nível inferior, dois repertórios neurais, constituídos por redes neurais nebulosas, geram independentemente comportamentos instintivos de busca de alvos e desvio de obstáculos, respectivamente. Algoritmos genéticos são utilizados no processo de aprendizagem para emular a evolução natural. Durante o aprendizado cada uma das redes determina sua estrutura bem como adquire o conhecimento necessário à reprodução do respectivo comportamento instintivo.

Os comportamentos de busca de alvos e desvio de obstáculos são conflitantes no contexto de navegação de veículos. Para sua coordenação, um terceiro repertório atua em nível superior gerenciando atividades dos outros dois repertórios. O repertório consiste de duas estruturas neurais: rede neural nebulosa e um neurônio nebuloso. A aprendizagem, não-supervisionada, baseia-se na teoria clássica do condicionamento. O repertório aprende a reconhecer padrões captados pelos sensores. Estes padrões indicam situações críticas de colisão, ou de iminência de captura de alvo, e sobre eles desenvolve uma estratégia de controle para navegação do veículo. Ele pondera adequadamente a influência de cada comportamento instintivo sobre a dinâmica do veículo eliminando conflitos de decisão e ao mesmo tempo estabelecendo uma trajetória eficiente de navegação.

Ainda neste capítulo foram apresentados resultados de simulação que confirmam as características do sistema proposto, tanto no que se refere aos métodos de aprendizagem quanto à sua concepção. Após a aprendizagem, o controlador neural nebuloso torna-se capaz de guiar eficientemente o veículo pelo ambiente. Além disso, o conhecimento adquirido em um ambiente particular é generalizado para diferentes tipos de ambiente. Nota-se também que um comportamento diferente dos instintivos é aprendido pelo controlador. Para que seja capaz de gerar trajetórias sem colisões, o controlador faz o veículo tangenciar obstáculos.

As principais características do controlador o distinguem de outras propostas presentes em literatura. O controlador não necessita do conhecimento da configuração do ambiente para que determine a trajetória do veículo. O controlador proposto neste trabalho é do tipo reativo, ou seja, ele determina suas ações de controle de acordo com a situação corrente do veículo no ambiente; portanto quaisquer alterações na configuração do ambiente são sempre consideradas. Ele é capaz de aprender sem supervisão a tarefa de navegação de veículos. Conseqüentemente, não é necessário que qualquer base de conhecimento seja previamente construída e inserida no controlador. Além disso, o controlador proposto é constituído por um sistema neural nebuloso concebido integralmente segundo progressos da neurofisiologia e inspirado em teorias da evolução natural e comportamento animal. Esta sua característica o faz significativamente importante para a compreensão da inteligência em geral e do fenômeno da aprendizagem em particular.

CAPÍTULO 5

CONCLUSÕES

5. Conclusões

5.1 Introdução

No curso do seu desenvolvimento, a sociedade moderna tem se deparado com problemas de complexidade crescente. Técnicas tradicionais mostram-se freqüentemente impotentes em solucioná-los. Entre outras dificuldades encontradas estão: critérios rígidos de desempenho, presença de ruídos ou alterações na dinâmica dos sistemas.

A pesquisa por abordagens alternativas capazes de superar tais dificuldades tem sido intensa. Algumas, por exemplo, redes neurais artificiais, sistemas nebulosos e algoritmos genéticos, são inspiradas na natureza e usufruem do poder computacional dos atuais computadores para implementar suas soluções. As respectivas teorias, que constituem objeto de pesquisa na área de inteligência computacional, têm sido reconhecidas pelo sucesso dos resultados alcançados. Além disso, pela simplicidade tanto de compreensão quanto de implementação, estas abordagens estão sendo disseminadas pelos diferentes segmentos de atividades científicas e também tecnológicas.

Uma das atuais tendências de pesquisa em inteligência computacional visa investigar a integração de suas diferentes abordagens. As redes neurais nebulosas são fruto da integração da teoria de redes neurais com a teoria da lógica e conjuntos nebulosos. Elas unificam vantagens e cobrem deficiências individuais de cada uma das teorias que as formam. De um lado as redes neurais nebulosas incorporam a capacidade de tratamento da linguagem natural presente nos sistemas nebulosos e de outro as habilidades de adaptação e aprendizagem exibidas pelas redes neurais. A possibilidade de usufruir destas características unificadamente é um fator que desperta grande motivação para o estudo e pesquisa das redes neurais nebulosas.

O principal foco deste trabalho concentrou-se no desenvolvimento de novas classes de rede neurais nebulosas com aplicação em problemas de modelagem nebulosa e navegação autônoma de veículos.

5.2 Contribuições

A modelagem através de sistemas nebulosos é feita a partir de um conjunto de regras do tipo se-então. Estas regras permitem fácil compreensão e estudo da natureza do objeto modelado. Infelizmente, poucos são os métodos sistemáticos para a construção de tais modelos. As redes neurais nebulosas surgem como uma alternativa de grande potencial devido às suas características.

Para aplicação em modelagem foram desenvolvidas duas classes de redes neurais nebulosas (denominadas N1 e N2 respectivamente) que permitem codificação direta de regras nebulosas do tipo se-então em suas estruturas.

A rede neural nebulosa N1 aprende em duas fases. Inicialmente ela se auto-organiza estabelecendo regras e funções de pertinência relativas aos seus antecedentes. Em seguida, ela define conseqüentes utilizando uma abordagem supervisionada. Suas características mais importantes são: 1) fácil inserção e extração de conhecimento na forma de regras

nebulosas; 2) o processamento neural emula o mecanismo de raciocínio nebuloso; 3) funções de pertinência são totalmente determinadas a partir dos dados de treinamento sem que nenhuma informação predefinida seja necessária; 4) regras são geradas automaticamente sem qualquer procedimento inicial ou complementar; 5) a partição do espaço de entradas garante que as regras cubram todo o domínio de operação do modelo (observado em resultados de simulação) e 6) N1 é um aproximador universal.

As propriedades de convergência do método de aprendizagem e da capacidade de aproximação universal foram caracterizadas formalmente por meio de teoremas.

A rede N2 compartilha várias das características de N1 enumeradas anteriormente, tais como: 1, 2, 4, 5 e 6. Sua aprendizagem não limita-se a um período de treinamento. Ela aprende enquanto e sempre que seu desempenho não atingir o índice especificado. A aquisição de conhecimento implica na alteração de sua estrutura correspondendo à inserção de regras nebulosas e ajuste das funções de pertinência.

As redes N1 e N2 reúnem um conjunto de características difícil de ser reproduzido. Além disso, cada qual se sobressai em um aspecto de maior relevância. Em N1 desponta a capacidade de determinar independente e completamente as funções de pertinência que compõem as regras nebulosas. O método de aprendizagem empregado em N2, do tipo construtivo, também a distingue da maioria das propostas. Ela permanece em constante adaptação, o que possibilita compensar degradações de desempenho provocadas por variações nos parâmetros do modelo, ruídos ou quaisquer outras perturbações.

Tanto N1 quanto N2 são capazes de projetar ou auxiliar na determinação de modelos nebulosos, especificando parâmetros diversos e mantendo uma completa transparência no que se refere às suas formas de representação e processamento de conhecimento. É importante notar que elas são capazes de estreitar a distância entre a representação do conhecimento numérico e lingüístico, sendo capazes de traduzi-los de uma forma a outra em ambos os sentidos.

A rede N1 foi avaliada em problemas de aproximação de funções. Dois tipos de experimentos são considerados: aprendizagem em ambiente livre de perturbação e aprendizagem em presença de ruído. Em ambos os casos, comparações com outras abordagens mostram que N1 atinge índices de desempenho competitivos e muitas vezes superiores, gerando um número reduzido de regras (compromisso desejável entre complexidade e precisão).

A aplicação correspondente à rede N2 refere-se à modelagem de um sistema dinâmico, especificamente à modelagem de um forno (problema extensamente considerado em literatura para avaliação de desempenho). Os modelos obtidos a partir de simulações, incluindo os relativos a propostas alternativas, são apresentados. A análise dos resultados demonstram que a rede N2 oferece vantagens tanto no que se refere à precisão quanto à simplicidade de seus modelos.

Ambas as redes confirmaram, após confronto contra outras propostas, as expectativas em torno de suas potencialidades. Muito embora as aplicações escolhidas tenham se referido a problemas de modelagem, elas são igualmente capazes de solucionar outros tipos de

problemas, em particular problemas de controle, além de proporcionar vantagens relativas à sua compatibilidade com o modo de raciocinar e de se expressar do ser humano.

A outra aplicação considerada neste trabalho refere-se à navegação autônoma de veículos. Em sua forma mais usual considera-se um ambiente onde o veículo deve alcançar o alvo estabelecendo uma trajetória livre de colisões contra obstáculos. Durante o percurso pelo ambiente o veículo depara-se com distintas situações. Para cada uma delas uma ação de controle adequada deve ser gerada. A grande multiplicidade de situações possíveis em um ambiente geral é a origem da dificuldade no projeto de controladores para navegação autônoma de veículos.

Neste trabalho um sistema neural nebuloso para controle autônomo de veículos foi proposto. A concepção geral do controlador têm origem na observação dos animais e estão fundamentadas nos recentes resultados da neurofisiologia. Em sua implementação foram utilizadas abordagens provenientes da inteligência computacional, especificamente, redes neurais nebulosas e algoritmos genéticos. Elas permitem emular de forma natural alguns conceitos envolvidos, por exemplo, a evolução natural, a teoria do condicionamento e os comportamentos instintivos e emergentes. O sistema neural nebuloso é hierarquizado. Em um nível inferior duas redes neurais nebulosas geram, independentemente, comportamentos instintivos de busca de alvos e desvio de obstáculos, respectivamente. A aprendizagem de cada uma das redes é realizada via algoritmos genéticos e emula o processo de evolução natural. Durante a aprendizagem a rede determina sua estrutura bem como adquire o conhecimento necessário à reprodução do respectivo comportamento instintivo. Devido aos diferentes objetivos de cada repertório e dependendo da configuração do ambiente, suas decisões podem se tornar conflitantes. Por este motivo, estes repertórios necessitam de uma coordenação. Atuando em um nível superior um terceiro repertório, constituído por uma rede neural nebulosa e um neurônio nebuloso, estabelece esta coordenação. O método de aprendizagem utilizado por este repertório é não-supervisionado e baseia-se na teoria clássica do condicionamento animal. Ele aprende a reconhecer quais padrões indicam situações críticas de colisão ou de iminência de captura de alvo. Além disso, desenvolve estratégia de controle para navegação do veículo. Suas saídas ponderam adequadamente a influência de cada comportamento instintivo sobre a dinâmica do veículo, eliminando conflitos de decisão e, ao mesmo tempo, estabelecendo uma trajetória eficiente.

Resultados de simulação apresentados comprovam as características do sistema proposto tanto no que se refere aos métodos de aprendizagem quanto à sua concepção. O controlador neural nebuloso adquire o conhecimento necessário à navegação do veículo enquanto estabelece suas primeiras interações com o ambiente. Após a aprendizagem, o sistema torna-se capaz de encontrar uma trajetória livre de colisões e de conduzir o veículo até um alvo estabelecido. O conhecimento adquirido durante a aprendizagem é generalizado para diferentes tipos de ambiente. Além disso, o controlador aprende uma estratégia diferente das instintivas. Para evitar colisões e atingir o alvo, o controlador guia o veículo de forma que este tangencie paredes.

A abordagem proposta neste trabalho contribui em vários sentidos para a pesquisa na área de controle autônomo, em particular em navegação autônoma de veículos. O controlador autônomo é capaz de guiar o veículo em ambientes desconhecidos independente de qualquer informação a respeito de sua topologia. As decisões do controlador são tomadas

de acordo com a situação corrente do veículo. Portanto, o controlador é robusto no que se refere a alterações que possam ocorrer no ambiente (mudança na posição de alvos ou disposição de obstáculos). Ele é capaz de aprender sem supervisão a tarefa de navegação de veículos reduzindo esforços ou possíveis falhas de projeto. A concepção geral do controlador é marcada pela coerência nos diversos estágios de seu projeto. Sua inspiração, fundamentação e abordagens adotadas para implementação têm como ponto comum a modelagem da natureza da maneira mais verossímil possível.

5.3 Trabalhos Futuros

Os resultados de simulação não só confirmaram as virtudes das propostas apresentadas mas também possibilitaram a constatação de suas deficiências e limitações. Em seguida, alguns dos aspectos desfavoráveis mais preponderantes de cada proposta são brevemente discutidos.

Algumas das falhas presentes nas redes neurais nebulosas N1 e N2 podem ser suprimidas se suas características forem integradas num só modelo neural. De um lado observa-se que ao conjunto de características da rede N1 falta a capacidade de adaptação a mudanças no ambiente. Esta característica é exibida pela rede N2. Por outro lado N2 mostra-se dependente de informações adicionais para estabelecer funções de pertinência, ao contrário da rede N1 que é independente neste sentido. Um possível tema para investigação futura diz respeito ao desenvolvimento de um modelo que unifique tais características de N1 e N2, visando corrigir suas deficiências individuais. Inclusive para este novo modelo, também é importante corrigir a falha de aprendizagem da rede N2 referente ao excessivo número de neurônios adicionados em sua estrutura (método MT2). De acordo com o algoritmo de aprendizagem, sempre que o desempenho da rede não satisfaz os índices requeridos, ela expande sua estrutura visando adquirir mais conhecimento e melhorar seu desempenho. Infelizmente, ela não reduz sua estrutura de forma a eliminar conhecimentos desnecessários e/ou redundantes. Esta estratégia torna-se ineficaz se a estrutura atinge um limite máximo de expansão.

Nota-se em alguns dos experimentos realizados que as aproximações obtidas pela rede N1 não apresentam uma suavidade comparável àquelas exibidas por outras abordagens. Durante a aprendizagem, a rede neural N1 compõe um conjunto de regras nebulosas. Cada uma delas mapeia um subespaço do espaço de entradas no subespaço de saída através do seu conseqüente. Para que a aproximação apresente-se suave, as regras nebulosas devem concordar nos limites de suas regiões de influência. Esta concordância é fortemente dependente de como os conseqüentes das regras nebulosas são definidos. Outras definições podem ser experimentadas visando melhorar a suavidade das aproximações. Esta característica não foi alvo de estudo no presente trabalho e merece ser investigada em maior profundidade futuramente.

A rede neural nebulosa que compõe o repertório de coordenação do controlador para navegação autônoma de veículos apresenta uma deficiência semelhante àquela apresentada pela rede N2. O método de aprendizagem utilizado nesta rede é do tipo construtivo. A rede expande sua estrutura para adquirir novos conhecimentos, mas não a reduz quando o conhecimento prescindível pode ser eliminado. Esta questão pode ser considerada

compartilhadamente com os trabalhos sugeridos para a rede N2. Além disso, o sistema neural nebuloso proposto para esta aplicação é do tipo reativo, ou seja, suas ações de controle consideram apenas a situação corrente do veículo. Esta sua característica cria certas limitações quanto à sua capacidade de navegação. Certas configurações do ambiente, e.g., tipo rua sem saída, podem aprisionar o veículo. Uma vez que o veículo entre na rua, o veículo pode iniciar uma trajetória cíclica, tornando-se incapaz de escapar pela única saída e de alcançar o alvo. Esta limitação também causa problemas quando a capacidade de manobras do veículo exige, por exemplo, que ele afaste-se de seu alvo, refaça sua estratégia de abordagem e retorne ao alvo. Transpor esta limitação é uma questão imprescindível não só por dotar o veículo de habilidades mais poderosas para navegação como também por permitir um melhor entendimento da capacidade de inteligência e formação de memória. Ela constitui-se um tema que deve ser profundamente investigado e explorado.

Referências Bibliográficas

- [Abe95] Abe, S. e Lan, M. "A Method for Fuzzy Rules Extraction directly from Numerical Data and Its Application to Pattern Classification", IEEE Transactions on Fuzzy Systems, vol. 3, nº. 1, pp. 18-28, fevereiro, 1995.
- [Abe96] Abe, S, "Convergence Acceleration of The Hopfield Neural Network by Optimising Integration Step Sizes", IEEE Transactions on System, Man, and Cybernetics, Part B, vol. 26, nº. 1, pp. 194-201, fevereiro, 1996.
- [Adachi92] Adachi, M., Aihara, K. e Kotani, M., "Nonlinear Associative Dynamics in a Chaotic Neural Networks", Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks, pp. 947-950, Iizuka, Japão, 1992.
- [Aihara90] Aihara, K., Takabe, T. e Toyoda, M., "Chaotic Neural Networks", Physics Letters A, vol. 144, 6,7, pp. 333-340, março, 1990.
- [Almassy92] Almassy, N. e Vershure, P., "Optimization Self-Organizing Control Architectures with Genetic Algorithms: The interaction between natural selection and ontogenesis", em R. Manner e B. Manderick, Parallel Problem Solving from Nature, 2, Elsevier Science Publishers, 1992.
- [Al-Sakaf95] Al-Sakaf, O. e Hawbrich, H. "An Intelligent Load Forecasting System-based on Kohonen's Self-Organizing Feature Maps", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, pp. 1540-1545, Aachen, Germany, setembro, 1995.
- [Anderson95] Anderson, J., An Introduction to Neural Networks, MIT Press, Cambridge, Massachusetts, USA, 1995.
- [Ash89] Ash, T., "Dynamic Node Creation in Backpropagation Neural Networks", Connection Science, vol. 1, nº.4, pp. 365-375, 1989.
- [Baglio94] Baglio, S., "Neuro-Fuzzy Networks to Predict Urban Traffic", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, pp. 824-828, Aachen, Germany, setembro, 1994.
- [Bastian95] Bastian, A. "Towards a Fuzzy System Identification Theory", Proceedings of Sixth International Fuzzy Systems Association World Congress, pp. 69-72, São Paulo, Brasil, 1995.
- [Beom92] Beom, H.R. e Cho, H.S., "Path Planning for Mobile Robot Using Skeleton of Free Space", Proceedings of IFAC Symposium on Information Control Problems for Manufacturing, Toronto, Canada, pp. 509-513, maio, 1992.
- [Beom95] Beom, H.R. e Cho, H.S., "A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning", IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, nº.3, março, 1995.
- [Box70] Box, G. e Jenkins, G., Time Series Analysis, Forecasting and Control, Holden Day, San Francisco, EUA, 1970.
- [Brooks91] Brooks, R., "Intelligence Without Reason", Proc. IJCAI, pp. 569-595, agosto, 1991.

- [Broomhead89] Broomhead, D. S., Lowe, D. e Webb, A. K., "A Sum Rule Satisfied By Optimized Feedforward Layered Networks", RSRE Memorandum, 4341, Royal signal and Radar Establishment, Malvern, UK, 1989.
- [Campos96] Campos, T.P., "Connectionist Modeling for Arm Kinematics Using Visual Information", IEEE Transactions on System, Man, and Cybernetics, Part B, vol. 26, nº. 1, pp. 89-99, fevereiro, 1996.
- [Carpenter87] Carpenter, G. e Grossberg, S., "Associative Learning, Adaptive Pattern Recognition, and Competitive Decision Making by Neural Networks", in H. Szu (ed.) Optical and Hybrid Computing, SPIE, 634, pp. 218-247, 1987.
- [Carpenter95] Carpenter, G., Grossberg, S. e Reynolds, J., "Fuzzy ART MAP Nonparametric Probability Estimator for Nonstationary Pattern Recognition Problems", IEEE Transactions on Neural Networks, vol. 6, nº. 6, pp. 1330-1336, novembro, 1995.
- [Castro95] Castro, J. "Fuzzy Logic Controllers Are Universal Approximators", IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, nº. 4, pp. 629-635, 1995.
- [Chakrabarti95] Chakrabarti, S., Bindal, N. e Theagarajan, K., "Robust Radar Target Classifier Using Artificial Neural Networks", IEEE Transactions on Neural Networks, vol. 6, nº. 3, pp. 760-766, maio, 1995.
- [Chang92] Chang, T. e Abdel-Ghaffar, K., "A Universal Neural Net with Guaranteed Convergence to Zero System Error", IEEE Transactions on Signal Processing, vol. 40, nº. 12, pp. 3022-3031, dezembro, 1992.
- [Chen95^a] Chen, T., "Fuzzy Neural Network Applications in Medicine", Proceedings of The International Joint Conference of The Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium", pp. 627-634, Yokohama, Japão, 1995.
- [Chen95^b] Chen, T., Chen, H., "Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks", IEEE Transactions on Neural Networks, vol. 6, nº. 4, pp. 904-910, julho, 1995.
- [Chen96] Chen, S. e Stockman, G.C. "SO Dynamic Deformation for Building of 3-D Models", IEEE Transactions on Neural Networks, vol. 7, nº. 2, pp. 374-387, março, 1996.
- [Colombetti96] Colombetti, M., Dorigo, M. e Borghi, G., "Behavior Analysis and Training - A Methodology for Behavior Engineering", IEEE Transactions on Systems, Man, and Cybernetics, part B Cybernetics, vol. 26, nº. 3, pp. 365-380, junho, 1996.
- [Courant79] Courant, R., Calculus and Analitic Geometry, Goodyear Publishing Co., Inc., EUA, 1979.
- [Cybenko89] Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function", Mathematics of Control, Signal, and Systems, 2, pp. 303-314, 1989.
- [Delgado95] Delgado, M. "Generating Fuzzy Rules Using Clustering Based Approach", Proceedings of European on Intelligent Technique and Soft Computing, pp. 810-814, Aachen, Alemanha, 1995.
- [Denoeux95] Denoeux, T., "Application of Intelligent Techniques to River Quality Monitoring", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, Aachen, Alemanha, pp. 1603-1609, setembro, 1995.

- [Dingle93] Dingle, A., Andreae, J. e Jones, R., "A Chaotic Neural Unit", Proceedings of 1993 International Conference on Neural Networks, pp. 335-340, San Francisco, EUA, 1993.
- [Doersan94] Doersan, T. e Fatikow, S., "Neurofuzzy-Based Force Control of a Multifingered Robot Grip", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, pp. 829-834, Aachen, Germany, setembro, 1994.
- [Edelman87] Edelman, G., Neural Darwinism: The Theory of Neuronal Group Selection, Basic Books, EUA, 1987.
- [Egusa95] Egusa, Y., et al. "An Application of Fuzzy Set Theory for an Electronic Video Camera Image Stabilizer", IEEE Transactions on Fuzzy Systems, vol. 3, 3, pp. 351-356, agosto, 1995.
- [Fahlman90] Fahlman, S.E., Leiere, C., "The Cascade Correlation Learning Architecture", in D.S. Tourezky (ed.), Advances in Neural Information Processing Systems 2, pp. 524-532, San Mateo, CA, Morgan Kaufman Publishers, 1990.
- [Faller95] Faller, W.E., e Shreck, S. J., "Real-time Prediction of Unsteady Aerodynamics: Adaptation for Aircraft Control and Maneuverability Enhancement", IEEE Transactions on Neural Networks, vol. 6, n^o. 6, pp. 1461-1468, novembro, 1995.
- [Fang96] Fang, Y. e Kincaid, T. G., "Stability Analysis of Dynamical Neural Networks", IEEE Transactions on Neural Networks, vol. 7, n^o. 4, pp. 996-1006, julho, 1996.
- [Fausett94] Fausett, L., "Fundamentals of Neural Networks; Architectures, Algorithms, and Applications, Prentice Hall International, Inc. New York, USA, 1994.
- [Figueiredo93] Figueiredo, M.; Gomide, F. Rocha, A. e Yager, R., "Comparison of Yager's Level Set Method for Fuzzy Logic Control with Mamdani's and Larsen's Methods, IEEE Transactions on Fuzzy Systems, vol. 1, n^o. 2, pp. 156-159, maio, 1993.
- [Figueiredo95^a] Figueiredo, M., Gomide, F. e Pedrycz, W., "Fuzzy Neurons and Networks: Models and Learning", Proceedings of 3rd European Congress on Intelligent Techniques and Soft Computing, pp. 332-335, Aachen, Alemanha, 1995.
- [Figueiredo95^b], Figueiredo, M. , Gomide, F. e Pedrycz, W., "A Fuzzy Neural Network: Structure and Learning", em Fuzzy Logic and Its Applications, Information Sciences and Intelligent Systems, eds. Z. Bien e K. Min, pp. 177-186, Kluwer Academic Publishers, Holanda, 1995.
- [Figueiredo96], Figueiredo, M. e Gomide, F., "Evolving Neurofuzzy Networks for Basic Behaviors and a Recategorization Approach for Their Coordination", em Genetic Algorithms and Soft Computing, eds. Herrera, F. e Verdegay, J., pp. 533-552, Physica-Verlag, Itália, 1996.
- [Figueiredo97^a] Figueiredo, M. e Gomide, F. "Adaptive Neuro Fuzzy Modeling", Proceedings of The 6th IEEE International Conference on Fuzzy Systems, pp. 1567-1572, Barcelona, Espanha, 1997.
- [Figueiredo97^b] Figueiredo, M. e Gomide, F. "A Neural Fuzzy Approach for Fuzzy System Design", Proceedings of The International Conference on Neural Networks (ICNN'97), pp. 420-425, Huston, EUA, 1997.
- [Fukumoto95] Fukumoto, S. et al., "A Destructive Learning Method of Fuzzy Inference Rules", Proceedings of The International Joint Conference of The Fourth IEEE

- International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium”, pp. 687-694, Yokohama, Japão, 1995.
- [Funahashi89] “On the Approximate Realization of Continuous Mappings by Neural Networks”, *Neural Networks*, 2, pp. 183-192, 1989.
- [Goldberg89] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Weley, Reading, 1989.
- [Gomide92] Gomide, F. e Rocha, A., “A Neurofuzzy Controller”, *Proceedings of 2nd International Conference on Fuzzy Logic and Neural Networks - IIZUKA’92*, pp. 213-216, Fukuoka, Japão.
- [Gorinevsky95] Gorinevsky, D. “On the Persistency of Excitation in Radial Basis Function Network Identification of Nonlinear Systems”, *IEEE Transactions on Neural Networks*, vol. 6, nº. 4, pp. 859-879, julho, 1995.
- [Grader95] Grader, P., Keller, J. e Cai, J., “A Fuzzy Logic System for the Detection and Recognition of Handwritten Street Numbers”, *IEEE Transactions on Fuzzy Systems*, vol. 3, nº. 1, pp. 83-95, fevereiro, 1995.
- [Grossberg87] Grossberg, S., “Competitive Learning, From Interactive Activation to Adaptive Resonance”, *Cognitive Science*, 11, pp. 23-63, 1987.
- [Haykin94] Haykin, S., *Neural Networks*, Prentice Hall, 1994, New Jersey, EUA.
- [Hebb49] Hebb, D., *Organization Behavior*, John Wiley & Sons, New York, 1949.
- [Hentz91] Hentz, J., Krogh, A. e Palmer, R.G., *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishers Company, New York, USA, 1991.
- [Hesselroth94] Hesselroth, T., Sarkar, K., Smagt, P. e Schulten K., "Neural Network Control of a Pneumatic Robotic Arm", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, nº. 1, pp. 28-38, janeiro, 1994.
- [Hetch-Nielsen89] Hetch-Nielsen, R., *Neurocomputing*, Addison Wesley Publishing Company, 1989, San Diego, EUA.
- [Hiraga95] Hiraga, I. et al. “An Acquisition of Operator’s Rules for Collision Avoidance Using Fuzzy Neural Networks”, *IEEE Transactions on Fuzzy Systems*, vol. 3, nº. 3, pp. 280-287, agosto, 1995.
- [Holden83] Holden, A. e Winlow, W., "Neuronal Activity as the Behavior of a Differential System", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, nº. 5, pp. 711-719, setembro/outubro, 1983.
- [Hopfield82] Hopfield, J.J., “Neural Network and Physical Systems with Emergent Collective Computational Abilities”, *Proc. Nat. Acad. Sci.*, vol. 79, abril, pp. 2554-2558, 1982.
- [Hopfield85] Hopfield, J. e Tank, D., “Neural Computation of Decisions in Optimization Problems”, *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- [Hornik89] Hornick, K., Stinchcombe, M. e White, H., “Multilayer Feedforward Networks Are Universal Approximators”, *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [Hush93] Hush, D. e Horne, B., “Progress in Supervised Neural Networks”, *IEEE Signal Processing Magazine*, pp.8-39, janeiro, 1993.

- [Ichikawa92] Ichikawa, Y. e Sawa, T., "Neural Network Application for Direct Feedback Controllers", IEEE Transactions on Neural Networks, vol. 3, nº. 2, pp. 224-231, março, 1992.
- [Ishibuchi95] Ishibuchi, H. et al., "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms", IEEE Transactions on Fuzzy Systems, vol. 3, nº. 3, pp. 260-270, agosto, 1995.
- [Jagota95] Jagota, A., "Approximating Maximum Clique with a Hopfield Network", IEEE Transactions on Neural Networks, vol. 6, nº. 3, pp. 724-735, maio, 1995.
- [Jain96] Jain, A. e Mao, J., "Artificial Neural Networks: A Tutorial", Computer, pp. 31-44, março, 1996.
- [Jim96] Jim, K., Giles, L. e Horne, B.G., "An Analysis of Noise in Recurrent Neural Networks: Convergence and Generalization", IEEE Transactions on Neural Networks, vol. 7, nº. 6, pp. 1424-1438, novembro, 1996.
- [Kandell91] Kandell, E.; Schwartz, J. e Jessell, T., Principles of Neural Science, Elsevier, New York, 1991.
- [Kaufmann75] Kaufmann, A. "Introduction to the Theory of Fuzzy Subsets", Academic Press, New York, EUA, 1975.
- [Keller92] Keller, J., Yager, R., e Tahani, H., "Neural Network Implementation of Fuzzy Logic", Fuzzy Sets and Systems, vol. 45, pp 1-12, 1992.
- [Keuthen95] Keuthen, M., Seeliger, A. e Thornton, S., "Neural Net for Process Condition Prediction in Steelmaking", Proceedings of 3rd European Congress on Intelligent Techniques and Soft Computing, pp. 304-309, Aachen, Alemanha, setembro, 1995.
- [Klir95] Klir, G. e Yuan, B., Fuzzy Sets and Fuzzy Logic, Englewood Cliffs, New Jersey, Prentice Hall, 1995.
- [Kohonen82] Kohonen, T., "Self-Organized Formation of Topologically Correct Feature Maps", Biological Cybernetics, 43, pp.59-69, 1982.
- [Kohonen90] Kohonen, T., "The Self-Organizing Map", Proceedings of IEEE, vol. 78, nº. 9, pp. 1464-1480, setembro 1990.
- [Kolen91] Kolen, J. e Goel, A., "Learning in Parallel Distributed Processing Networks: Computational Complexity and Information Content", IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, nº. 2, pp.359-367, março/abril, 1991.
- [Kosko92] Kosko, B., Neural Networks and Fuzzy Systems, Prentice-Hall Inc., New Jersey, 1992.
- [Koza93] Koza, J. Genetic Programming, The MIT Press, Cambridge, EUA, 1993.
- [Krishnakumar94] Krishnakumar, S., Sauheney, S. e Wai, R., "Neuro-Controllers for Adaptive Helicopter Hover Training", IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, nº. 8, pp. 1142-1152, agosto, 1994.
- [Lee90] Lee, C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controllers, Part I" e, "Fuzzy Logic in Control Systems: Fuzzy Logic Controllers, Part II" IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, nº. 2, pp. 404-435, março/abril, 1990.

- [Leurg94] Leurg, C., "Optimum Learning for Bidirectional Associative Memory in the Sense of Capacity", IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, nº. 5, pp. 791-796, maio, 1994.
- [Li94] Li, W. "Fuzzy logic-based perception-action behavior control of a mobile robot in uncertain environments", Proc. 3rd IEEE International Conference on Fuzzy Systems, Orlando, EUA, pp. 1626-1631, 1994.
- [Lin91] Lin, C.T. e Lee, C.S.G., "Neural-Network-Base Fuzzy Logic Control and Decision System", IEEE Transactions on Computers, vol.40, nº 12, pp. 1320-1336, dezembro, 1991.
- [Lin95] Lin, Y. e Cunningham III, G.A., "A New Approach to Fuzzy-Neural System Modeling", IEEE Transactions on System, Man, and Cybernetics, vol. 3, nº. 2, pp. 190-197, maio, 1995.
- [Linc96] Lin, C. e Chen, H. "An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents", IEEE Transactions on Systems, Man, and Cybernetics, vol. 26, nº. 1, pp. 28-38, fevereiro, 1996.
- [Linj96] Lin, C.J. e Lin, C.T. , "Reinforcement Learning for ART-Based Fuzzy Adaptive Learning Control Networks", IEEE Transactions on Neural Networks, vol. 7, nº. 3, pp. 709-731, maio, 1996.
- [Lippmann87] Lippmann, R., "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, pp. 4-22, abril, 1987.
- [Liu92] Liu, W., Andreou, A. e Goldstein, M., "Voice-Speech Representation by an Analog Silicon Model of the Auditory Periphery", IEEE Transactions on Neural Networks, vol. 3, nº. 3, pp. 477-487, setembro, 1992.
- [Malsburg73] Malsburg, C. "Self-Organization of Orientation Sensitive Cells in the Striate Cortex", Kybernetik, 14, pp.85-100.
- [Manoranjan95] Manoranjan, V.S. et al. "A Systematic Approach to Obtaining Fuzzy Sets for Control Systems", IEEE Transactions on System, Man, and Cybernetics, vol. 25, nº. 1, pp. 206-213, janeiro, 1995.
- [Mataric92] Mataric, M., "Behavior-based systems: key properties and implications", Proc. IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems, Nice, França, maio, pp. 46-54, 1992.
- [MATLAB95] MATLAB User's Guide, The MATH WORKS, Inc., Natick, EUA, agosto, 1995.
- [Mauer95] Mauer, G. "A Fuzzy Logic Controller for an ABS Breaking System", IEEE Transactions on Fuzzy Systems, vol. 3, 4, pp. 381-388, novembro, 1995.
- [McCulloch43] McCulloch, W. e Pitts, W., "A Logical Calculus of the Ideas Imminent in Nervous Activity", Bull. Math. Biophys., vol.5, pp. 115-133, 1943.
- [Minsky69] Minsky, M. e Papert, S., Perceptrons, MIT Press, Cambridge, 1969.
- [Mitra94] Mitra, S., e Pal, S.K., "Self Organizing Neural network As A Fuzzy Classifier", IEEE Transactions on Systems, man and Cybernetics, Vol. 24, nº. 3, pp. 385-399, 1994.

- [Mitra96] Mitra, S., e Pal, S.K., "Fuzzy Self-Organization, Inferencing, and Rule Generation", IEEE Transactions on Systems, man and Cybernetics, Part A: Systems and Humans, Vol. 26, n^o. 5, pp. 608-620, 1996.
- [Narendra90] Narendra, K.S., e Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, 1, pp. 4-27, 1990.
- [Nie95] Nie, J. "Constructing Fuzzy Model by Self-Organizing Counterpropagation Network", IEEE Transactions on Neural Networks, vol. 25, n^o. 6, pp. 963-970, junho, 1995.
- [Oliveira94] Oliveira, M., Figueiredo, M. e Gomide, F., "A Neurofuzzy Approach to Autonomous Control", Proceedings of The 3rd International Conference on Fuzzy Logic, Neural Networks and Soft Computing, pp. 597-598, Iizuka, Japão, 1994.
- [Oliveira95^a] Oliveira, M., Figueiredo, M. e Gomide, F., "Evolving a Neurofuzzy Network in a Competitive Environment", Proceedings of The International ICSC Symposium on Fuzzy Logic, pp. B10-B16, Zurique, Suíça, 1995.
- [Oliveira95^b] Oliveira, M., Figueiredo, M. e Gomide, F., "Neurofuzzy Navigation and Neural Group Selection", Proceedings of Sixth International Fuzzy Systems Association World Congress, vol.1, pp. 73-76, São Paulo, Brasil, 1995.
- [Park95] Park, S., Park, L. e Park,C., "A Neuro-Genetic Controller for Nonminimum Phase Systems", IEEE Transactions on Neural Networks, vol. 6, n^o. 5, pp. 1297-1300, setembro, 1996.
- [Pedrycz93] Pedrycz, W., "Fuzzy Control and Fuzzy Systems", John Wiley & Sons, Inc., New York, EUA, 1993.
- [Pedrycz95] Pedrycz, W., Jam, P.C.F. e Rocha, A.F., "Distributed Fuzzy System Modeling", IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, n^o. 5, pp. 769-780, maio,1995.
- [Phola96] Phola, V. e Oldham, W., J., "Image Recovery and Segmentation Using Competitive Learning in a Layered Network", IEEE Transactions on Neural Networks, vol. 7, n^o. 4, pp. 843-856, julho,1996.
- [Poggio90] Poggio, T, e Girosi, F., "Networks for Approximation and Learning", Proceedings of the IEEE, 78, pp. 1481-1497, 1990.
- [Poggio90] Poggio, T. e Girosi, F. "Networks for Approximation and Learning", Proceedings of IEEE, vol.78, n^o. 9, pp. 1481-1497, setembro, 1990.
- [Reeke90] Reeke, G., Sporns, O. e Edelman, G., "Synthetic Neural Modeling: The "Darwin Series of Recognition Automata", Proceedings of IEEE, vol. 78, n^o. 9, pp. 1498-1530, setembro/outubro, 1990.
- [Rocha92] Rocha, A., Neural Nets, Springer Verlag, 1992.
- [Rosenblatt58] Rosenblatt, F., "The Perceptron; A Probabilistic Model for Information Storage and Organization in The Brain", Psychological Review, vol. 65, pp. 386-408, 1958.
- [Rosenblatt62] Rosenblatt, F., "Principles of Neurodynamics", Spartan, New York, USA, 1962.

- [Rovatti96] Rovatti, R. e Guerrieri, R. "Fuzzy Sets of Rules for System Identification", IEEE Transactions on Fuzzy Systems, vol. 4, nº. 2, pp. 89-102, maio, 1996.
- [Rubio95] Rubio, F., Berenguel, M. e Cancacho, E., "Fuzzy Logic Control of a Solar Power Plant", IEEE Transactions on Fuzzy Systems, vol. 3, nº. 4, pp. 459-468, novembro, 1995.
- [Rumelhart86] Rumelhart, D.E., Hinton, G. e Williams, R.J., "Learning Internal Representations by Error Propagation", em D.E. Rumelhart e J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, pp. 318-362, MIT Press, Cambridge, MA, 1986.
- [Sackingen92] Sackingen, E. et al., "Application of the ANNA Neural network Chip to High-Speed Character Recognition", IEEE Transactions on Neural Networks, vol. 3, nº. 3, pp. 498-505, setembro, 1992.
- [Simpson90] Simpson, P.K., Artificial Neural Systems, Pergamon Press, New York, 1990.
- [Sokol95] Sokol, R. e Mercier, G. "Recognition of Phonetic Features Using Neural-Fuzzy Network", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, Aachen, pp. 1751-1755, setembro, Alemanha, 1995.
- [Steels94] Steels, L. "The Artificial Life Roots of Artificial Intelligence", Artificial Life Journal, vol. 1, nº 1, 1994.
- [Suzuki95] Suzuki, Y., "Self-Organizing QRS-Wave Recognition in ECG Using Neural Network", IEEE Transactions on Neural Networks, vol. 6, nº. 6, pp. 1469-1477, novembro, 1995.
- [Takagi85] Takagi, T. e Sugeno, M. "Fuzzy Identification of Systems and Its Applications to Modeling and Control", IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, nº. 1, pp. 116-132, janeiro/fevereiro, 1985.
- [Takagi90] Takagi, H., "Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future Directions, Proceedings of International Conference on Fuzzy Logic & Neural Networks, pp. 13- 26, Iizuka, Japão, 1990.
- [Tanaka95] Tanaka, K., Sano, M. e Watanabe, H. "Modeling and Control of Carbon Monoxide Concentration Using a Neuro-Fuzzy Technique", IEEE Transactions on Fuzzy Systems, vol. 3, nº. 3, pp. 271-279, agosto, 1995.
- [Tani96] Tani, J. "Model-Based Learning for Mobile Robust Navigation from the Dynamical Systems Perspective", IEEE Transactions on Systems, Man, and Cybernetics, part B Cybernetics, vol. 26, nº. 3, pp. 421-436, junho, 1996.
- [Thiesing95] Thiesing, F., Middelberg, U. e Vornberger, O., "A Neural Network Approach for Predicting the Sale of Articles in Supermarkets", Proceedings of The European Congress on Intelligent Techniques and Soft Computing, pp. 315-320, Aachen, Germany, setembro, 1995.
- [Tsuji96] Tsuji, T., Ito, K. e Morasso, P., "Neural Network Learning of Robot Arm Impedance in Operational Space", IEEE Transactions on System, Man, and Cybernetics, Part B. Cybernetics, vol. 26, nº. 2, pp. 290-298, abril, 1996.
- [Verschure93] Verschure, P., "Formal Minds and Biological Brains", IEEE Expert, pp. 66-75, outubro, 1993.

- [Vershure92] Vershure, P.; Krose, B. e Pfeifer, R., "Distributed adaptive control: the self-organization of structured behavior", *Robotics and Autonomous Systems*, Vol. 9, pp. 181-196, 1992.
- [Vinod94] Vinod, V., Chaudhury, S. e Mudherjee, J., "A Connectionist Approach for Clustering with Application in Image Analysis", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, nº. 3, pp. 365-384, março, 1994.
- [Vuorimaa95] Vuorimaa, P., Jukarainen, T. e Karpanja, E. "A Neuro-Fuzzy System for Chemical Agent Detection", *IEEE Transactions on Fuzzy Systems*, vol.3, 4, pp. 415-424, novembro, 1995.
- [Wakabayashi95] Wakabayashi, T., Itoh, K. e Ohuchi, A. "A Method for Constructing of System Models by Fuzzy Flexible Interpretative Structural Modeling", *Proceedings of The International Joint Conference of The Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, pp. 913-918, Yokohama, Japão, 1995.
- [Walt96] Walt, T. e Reventer, J., "Process Modeling with the Regression Network", *IEEE Transactions on Neural Networks*, vol. 6, nº. 1, pp. 78-93, janeiro, 1996.
- [Wang94] Wang, T., Zhuang, X. e Xing, X., "Designing Bidirectional Associative Memories with Optimal Stability", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, nº. 5, pp. 778-790, maio, 1994.
- [Werbos90] Werbos, P.J., "Backpropagation Through Time: What It Does and How to Do It", *Proceedings of IEEE*, vol.78, nº. 10, pp. 1550-1560, outubro, 1990.
- [Whitehead96] Whitehead, B. A., "Genetic Evolution of Radial Basis Function Coverage Using Orthogonal Niches", *IEEE Transactions on Neural Networks*, vol. 7, nº. 6, pp. 1424-1438, novembro, 1996.
- [Widrow90] Widrow, B. e Lehr, M.A., "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", *Proceedings of the IEEE*, vol. 78, nº. 9, pp. 1415-1442, setembro, 1990.
- [Xia96] Xia, Y., "A Neural Network for Solving Linear and Quadratic Programming Problems", *IEEE Transactions on Neural Networks*, vol. 7, nº. 6, pp. 1544-1547, novembro, 1996.
- [Yager94] Yager, R.R. e Filev, D.P., *Essential of Fuzzy Modeling and Control*, John Wiley & Sons Inc., New York, EUA, 1994.
- [Yshikawa91] Ishidawa, S., "A Method of Indoor Mobile Robot Navigation by Fuzzy Control", *Proceedings of International Conference on Intelligent Robots and Systems*, Osaka, Japão, pp. 1013-1018, novembro, 1991.
- [Zadeh65] Zadeh, L., "Fuzzy Sets", *Information and Control*, nº. 8, pp. 338-353, 1965.
- [Zak91] Zak, M., "An Unpredictable-Dynamics Approach to Neural Intelligence", *IEEE Expert*, pp.4-10, agosto, 1991.
- [Zhang96] Zhang, C.,Q. e Fadali, M., S., "Nonlinear System Identification Using a Gabor/Hopfield Network", *IEEE Transactions on System, Man, and Cybernetics, Part B*, vol. 26, nº. 1, pp. 124-134, fevereiro, 1996.

[Zimmermann87] Zimmermann, H., Fuzzy Sets, Decision Making and Expert Systems, Kluwer-Nihoft, Boston, 1987.

[Zuben96] Zuben, F., "Modelos Paramétricos e Não Paramétricos de Redes Neurais Artificiais e Aplicações", Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1996.

[Zurada94] Zurada, J.; MarksII, R. e Robinson, C., (Eds.), Computational Intelligence: Imitating Life, IEEE Press, New York, USA, 1994.