

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

**SISTEMAS ESPECIALISTAS CONEXIONISTAS:
IMPLEMENTAÇÃO POR REDES DIRETAS E
BIDIRECIONAIS**

Rudimar Luís Scaranto Dazzi

Dissertação Submetida à Universidade Federal de Santa Catarina para
a obtenção do grau de Mestre em Ciência da Computação

Prof. Jorge Muniz Barreto, D.Sc.A.
Orientador

Florianópolis, Setembro de 1999

**SISTEMAS ESPECIALISTAS CONEXIONISTAS: IMPLEMENTAÇÃO
POR REDES DIRETAS E BIDIRECIONAIS**

RUDIMAR LUÍS SCARANTO DAZZI

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO
TÍTULO DE

MESTRE EM CIÊNCIA DA COMPUTAÇÃO

ESPECIALIDADE SISTEMAS DE CONHECIMENTO E APROVADA EM SUA
FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO.

Prof. JORGE MUNIZ BARRETO, D.Sc.A.
Coordenador do Curso

Banca Examinadora:

Prof. JORGE MUNIZ BARRETO, D.Sc.A.
Orientador

Prof. SILVIA MODESTO NASSAR, Dra
Co-orientadora

Prof. LOURDES MATTOS BRASIL, Dra

Prof. MAURO ROISENBERG, Dr

DEDICATÓRIA

A meus pais, meu irmão e minha
esposa por toda a força, carinho
e apoio em todos os momentos,
suportando minha ausência em
momentos importantes.
Obrigado pela compreensão.

AGRADECIMENTOS

Agradeço, em primeiro lugar, a meus pais e irmão, que me incentivaram quando larguei tudo para mudar de cidade e iniciar o mestrado.

Ao meu amor, Márcia, que encarou o casamento no meio da minha correria do mestrado e soube como ninguém aceitar a minha ausência, sempre dando total apoio e carinho. Passando horas dos finais de semana ao meu lado, enquanto escrevia minha dissertação.

Ao meu orientador por sua dedicação indicando sempre os melhores caminhos para que pudesse chegar até o fim, com sucesso.

A minha co-orientadora por seus conselhos, correções e incentivo.

À “dinda” Eliane por todo apoio que tive desde minha chegada à Florianópolis.

Aos amigos Otávio e Gilberto pelo auxílio nos momentos de aperto.

À Universidade do Contestado por viabilizar minha vinda à Florianópolis para iniciar o mestrado.

À Universidade do Vale do Itajaí por me acolher durante todo esse processo.

A todos que, direta ou indiretamente, colaboraram comigo nessa caminhada.

Sumário

Sumário	v
Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Abreviaturas	x
1 Introdução	3
1.1 Motivação	3
1.2 Objetivos	4
1.2.1 Objetivo Geral	4
1.2.2 Objetivos Específicos	4
1.3 Organização do Trabalho	5
2 Inteligência Artificial	6
2.1 Histórico da Inteligência Artificial	6
2.1.1 Alguns Precursores	8
2.1.2 As fases da IA	11
3 Redes Neurais Artificiais	14
3.1 Introdução	14
3.2 Fundamentos Biológicos	15
3.3 Organização Hierárquica do Cérebro	18
3.4 Modelo do Neurônio Artificial	19

3.4.1	Modelo de McCulloch e Pitts	19
3.4.2	Modelo Geral	20
3.5	Topologias de Redes Neurais Artificiais	23
3.5.1	Redes Diretas (“ <i>Feedforward</i> ”)	25
3.5.2	Redes com realimentação ou ciclos (“ <i>Feedback</i> ”)	28
3.5.3	Redes de Memórias Associativas Bidirecionais	29
3.6	Caracterização das RNA	30
3.7	Aprendizado em RNA	32
3.7.1	Aprendizado Supervisionado	32
3.7.2	Aprendizado não Supervisionado	33
3.8	Regras de Aprendizado	34
3.8.1	Regra de Hebb	34
3.8.2	Regra Delta	35
3.8.3	Regra de Retropropagação (“ <i>Backpropagation</i> ”)	36
3.9	Diferenças entre RNA e RNB ou RNN	37
4	Sistemas Especialistas	41
4.1	Introdução	41
4.2	Histórico	42
4.3	Alguns SE desenvolvidos na Universidade Federal de Santa Catarina (UFSC)	44
4.4	Definição de Sistemas Especialistas	54
4.5	Áreas de Utilização de Sistema Especialista	55
4.6	As duas Abordagens dos Sistemas Especialistas	55
4.6.1	Sistemas Especialistas Simbólicos (SES)	55
4.6.2	Sistemas Especialistas Conexionistas (SEC)	56
4.6.3	Uma Comparação Simplificada entre a Estrutura de um SES e de um SEC	58
5	Metodologia Proposta	61
5.1	Passos para o Desenvolvimento	61

5.2	Base de Exemplos	62
5.3	Considerações Importantes	64
6	Escolha de Ferramentas para Implementar o Sistema Especialista	67
6.1	PDP - “Parallel Distributed Processing”	67
6.2	MatLab	68
6.3	NeuralWorks	69
6.4	BrainMaker	72
6.5	Neural Model	73
6.6	Ferramenta Escolhida	75
7	Implementação da Rede Direta	76
8	Crítica de Utilização da BAM	81
8.1	Introdução	81
8.2	Implementação da BAM	81
8.2.1	Teoria de Estabilidade	82
8.2.2	A instabilidade na BAM	95
9	Conclusões	98
9.1	Rede BAM	98
9.2	Rede Direta	99
9.3	Análise Geral	99
10	Anexos	101

Lista de Figuras

3.1	Neurônio e suas conexões.	16
3.2	Modelo de neurônio de McCulloch e Pitts.	19
3.3	Figura do neurônio geral.	20
3.4	Algumas funções de ativação	21
3.5	RNA Direta de uma camada.	27
3.6	RNA com 3 camadas.	27
3.7	RNA com ciclos.	28
3.8	RNA Simétrica com camada interna.	29
3.9	RNA com as conexões ressaltadas.	31
3.10	Aprendizado supervisionado.	33
3.11	Aprendizado não supervisionado.	34
3.12	Fluxograma Back-Propagation (Pandya página 95 [68])	38
4.1	Componentes básicos de um SES.	58
4.2	Componentes básicos de um SEC.	59
6.1	Tela básica do PDP com a rede IAC	68
6.2	Exemplo de uma aplicação em MatLab	70
6.3	Exemplo de uma tela do Neural Works	71
6.4	Tela de abertura do NetMaker	72
6.5	Tela de abertura do BrainMaker	73
6.6	Tela do Neural Model	74
7.1	Janela do NeuralWorks com parâmetros	77

7.2	Rede treinada com 24 casos	78
7.3	Gráfico de evolução da rede Direta com 5 neurônios	79
7.4	Gráfico de evolução da rede Direta com 3 neurônios	80
8.1	Uma função definida positiva	93

Lista de Tabelas

5.1	Exemplos para o treinamento (início)	64
5.2	Exemplos para o treinamento (continuação)	65
5.3	Exemplos para os testes (início)	66
5.4	Exemplos para os testes (continuação)	66

Lista de Abreviaturas

ACM	- <i>Association for Computing Machines</i>
BA	- Base de Aprendizado / Treinamento
BAM	- <i>Bidirectional Associative Memory</i>
BP	- <i>Backpropagation</i>
BT	- Base de Testes
DEC	- <i>Digital Eletronic Corporation</i>
DOS	- Disc Operating System
EPS	- Engenharia de Produção e Sistemas
GPEB	- Grupo de Pesquisa em Engenharia Biomédica
GPS	- <i>General Problem Solver</i>
IA	- Inteligência Artificial
IAC	- Inteligência Artificial Conexionista
IAS	- Inteligência Artificial Simbólica
NW	- Neural Works
PDP	- Parallel Distributed Processing
PPGCC	- Program de Pós-Graduação em Ciência da Computação
RNA	- Rede Neural Artificial
RNB	- Rede Neural Biológica
RNN	- Rede Neural Natural
SBC	- Sistemas Baseados em Conhecimento
SE	- Sistemas Especialistas
SEC	- Sistemas Especialistas Conexionistas
SES	- Sistemas Especialistas Simbólicos
SMP	- <i>math expert system</i>
UFSC	- Universidade Federal de Santa Catarina

Resumo

Este trabalho apresenta a implementação de Sistemas Especialistas Conexionistas (SEC), a qual utiliza a topologia de Redes Neurais Artificiais (RNA) Diretas. Após a apresentação dos conceitos básicos de Inteligência Artificial e suas abordagens simbólica e conexionista, apresentar-se-ão os fundamentos dos SEC.

A análise dos resultados do aprendizado com a inserção de novos exemplos para o treinamento das redes diretas será mostrado com o suporte de gráfico de acompanhamento, que apresenta a curva de desempenho da rede, representada pelos seguintes eixos: número de exemplos utilizados no treinamento pelo percentual de acertos obtidos nos testes.

Apresenta-se, também, os resultados obtidos com a tentativa de implementação da rede BAM que, em virtude de problemas de instabilidade encontrados e demonstrados no desenvolvimento desse, tornou-se impossível obter as conclusões, bem como os testes.

Por fim, faz-se uma breve análise dos resultados obtidos com o término desse trabalho, ressaltando os pontos mais significativos encontrados no decorrer de seu desenvolvimento.

Abstract

This work presents the implementation of Connectionist Expert Systems (CES), which using the topology of Feedforward Artificial Neural Networks (ANN). After the presentation of Artificial Intelligence basic concepts and its symbolic and connectionist approaches, the foundations of CES will be presented.

Learning results analysis with the insertion of new examples for the feedforward nets training will be shown with support of graphical that presents the network performance graph, represented by the following axes: number of examples used in the training versus percentile of successes obtained in the tests.

The results obtained by the attempt for implementing the BAM net are showed. However problems found about the net instability and demonstrated in the development of that, it was impossible to obtain the conclusions about the tests.

Finally, it is presented a brief analysis of the results obtained in the end of this work, showing the most significant points found during its the development.

Capítulo 1

Introdução

1.1 Motivação

Após o renascimento do interesse por redes neurais, no início dos anos 80, costumava-se falar da dicotomia: sistemas especialistas e sistemas conexionistas. A referência a sistemas especialistas dizia respeito, com efeito, à metodologia usada para implementá-los: uso de técnicas simbólicas. Por esse motivo, não se falava em sistemas especialistas conexionistas, mas sim em Redes Neurais. Este termo foi consolidado a partir de 1988 com Gallant [34], quando publicou no *Communications of the Association for Computing Machines* (ACM) o célebre artigo “*Connectionist expert systems*” onde mostrou como seria possível implementar um sistema especialista usando uma rede neural direta e dela extrair explicações, o que na época era considerado um dos grandes inconvenientes da abordagem conexionista para sistemas especialistas.

Mais recentemente, outro tipo de arquitetura foi proposta para uso de sistemas especialistas: o das redes bidirecionais (Memórias Associativas Bidirecionais ou *Bidirectional Associative Memory* (BAM)). Um primeiro estudo comparativo destas duas abordagens foi feito por de Azevedo [20]. Este estudo, no entanto, deu apenas noções preliminares, devendo ser completado de modo a que se tenha um conjunto de heurísticas capazes de indicar qual das duas abordagens é a mais adaptada em cada caso particular. Nos testes realizados, com resultados favoráveis às BAM, foi utilizado na implementação a relação entre casos clínicos e seus diagnósticos (além dos sintomas e

diagnósticos), conhecimento este não existente nas Redes Diretas, que são treinadas apenas através de exemplos, onde aparece apenas o conhecimento entre sintomas e diagnósticos.

Este estudo derivou da necessidade de informações a respeito do desempenho e aplicações das redes BAM. Com efeito, não se conhece estudos que indiquem as reais vantagens ou desvantagens das Redes BAM, comparadas às Redes Diretas. As conversas informais, entre os pesquisadores mostram que, na maioria das vezes, se usa Redes Diretas, mesmo sem ter uma fundamentação maior, no que diz respeito as vantagens ou desvantagens de se usar essa topologia de rede. Como informalmente se diz, existe um modismo, ou costume de usar as Redes Diretas, pois estas já são bem conhecidas e testadas, dessa forma sabe-se de antemão, que devem funcionar.

Com isso, pretende-se concluir com mais clareza e fundamentação, qual a topologia de rede é mais indicada para as características de um determinado problema. Dando assim aos usuários dessa técnica, condições para optar pelo melhor método na resolução dos seus problemas.

1.2 Objetivos

1.2.1 Objetivo Geral

Este estudo tem o objetivo de comparar as topologias de Redes Diretas e Redes Bidirecionais (BAM) na implementação de Sistemas Especialistas (SE).

1.2.2 Objetivos Específicos

Para atingir o objetivo geral, deverão ser alcançados os seguintes objetivos específicos:

1. Apresentar a implementação de um SE Conexionista (SEC) usando redes diretas como proposto por Gallant [34];
2. Apresentar a implementação de um SE Conexionista (SEC) usando a BAM como proposto por de Azevedo [20];

1.3 Organização do Trabalho

O primeiro capítulo é este que está sendo apresentado aqui.

O segundo capítulo é introdutório, apontando as principais características, assim como um breve histórico da evolução da IA.

O terceiro capítulo mostra os conceitos básicos necessários para compreensão da topologia das Redes BAM e das Redes Diretas. Aproveita-se para fazer um resumo dos conceitos de RNA necessários à compreensão deste trabalho, tais como: Fundamentação biológica, o modelo de neurônio artificial, topologias das RNA, características das RNA, regras de aprendizagem e as principais diferenças entre RNA e Redes Neurais Biológicas (RNB).

O quarto capítulo trata dos Sistemas Especialistas (SE), conceituando-os e mostrando um histórico desde os primeiros SE apresentados, assim como, um apanhado dos trabalhos realizados na Universidade Federal de Santa Catarina (UFSC) em seus diversos centros e departamentos.

O quinto capítulo apresenta a metodologia de comparação a ser utilizada no trabalho, incluindo a lista de exemplos a serem utilizados para o treinamento das redes e a lista de exemplos para os testes das mesmas.

Para a dissertação, serão implementadas (utilizando ferramentas específicas) as duas redes propostas, uma rede direta e uma de memórias associativas. Com as duas redes em funcionamento, iniciará a fase de treinamento da rede direta e inserção de exemplos (para o aprendizado) da BAM. Serão utilizados exatamente os mesmos exemplos para ambas as redes.

Após essa fase, terá início a fase de testes das redes, utilizando nesse momento os exemplos reservados para os testes, que também serão os mesmos para as duas redes. Além de testar as redes, objetiva-se montar uma tabela de crescimento de ganho de conhecimento das redes, com a apresentação de novos exemplos.

Capítulo 2

Inteligência Artificial

Inteligência Artificial (IA) é o estudo da inteligência, preocupado em implementá-la em computadores para que estes exibam comportamento inteligente quando da solução de problemas propostos, podendo dessa forma dar respostas à perguntas não explicitamente programadas anteriormente. Para isso usa informações que fazem parte do mesmo domínio do problema [7][80].

As definições para IA, encontradas na literatura, utilizam diversos pontos de vista, conforme a visão que o autor quer passar, a respeito do texto que está sendo explorado. Muitas delas, podem ter interpretações diferentes, dependendo da forma como se visualiza a mesma. Por exemplo, quando se utiliza a idéia de *comportamento inteligente* para explicar a IA, pode-se perguntar: O que é *Comportamento Inteligente*? Dependendo dessa resposta, pode-se ter uma visão diferenciada de IA.

Aqui, utiliza-se apenas uma idéia, que é a da IA sendo um, “software ou hardware” capaz de responder a perguntas ou gerar ações as quais ele não foi explicitamente programado para responder ou fazer.

2.1 Histórico da Inteligência Artificial

Quando apareceram os primeiros computadores, criou-se uma grande admiração pela nova máquina, capaz de realizar tarefas até então desenvolvidas apenas por humanos. Este fato, propiciou a que muitos começassem a considerar estas máquinas maravilhosas

como “cérebros eletrônicos”. Criava-se, com isso, um mito, gerando um clima de desconfiança a respeito das capacidades destes equipamentos, que sob esse enfoque, poderiam até substituir os humanos em suas tarefas. Esse clima perdurou até o final da década de 50 [7].

Em função disso, as fábricas de computadores não conseguiam mais clientes, até que, encabeçados pela IBM, as fábricas começaram uma propaganda maciça com o objetivo de desmistificar os computadores, tentando com isso conquistar clientes [7].

A década de 60, trouxe, por sua vez, a popularização dos computadores, que, passaram a ser vistos como máquinas desprovidas de qualquer capacidade de raciocínio. Com isso, os computadores deixaram de ser considerados uma ameaça aos seus empregos, uma vez que não eram providos da capacidade de “pensar”, ou executar tarefas, as quais não tinham sido explicitamente programados para executar, começando assim a desfazer o mito anteriormente criado.

Estes fatos apresentaram uma nova realidade para os computadores, que foram se tornando extremamente úteis na realização de tarefas corriqueiras e repetitivas dos escritórios e empresas, tais como controle de estoque, contabilidade, etc. Tarefas estas, desagradáveis e desgastantes quando realizadas por humanos, mas realizadas com muita agilidade e precisão pelos computadores.

Desde então os computadores foram caracterizados como máquinas “burras”, mas extremamente úteis nas tarefas que não necessitassem nenhum tipo de raciocínio. Com essa nova abordagem, os computadores começaram a ser cada vez mais vendidos e utilizados.

Neste mesmo período, um pequeno grupo de pesquisadores continuava buscando maneiras de dotar os computadores de características inteligentes. Aos poucos, eles começaram a obter alguns resultados significativos nesta área, hoje conhecida por IA. Porém, a repercussão junto ao cidadão comum, veio com o retumbante anúncio do projeto japonês da quinta geração de computadores [27], cuja pretensão era construir uma nova geração de computadores inteligentes, divulgando assim uma importante aplicação de IA: SE.

Ainda na década de 60, alguns pesquisadores de IA, perceberam que, para se ter

algo de uso prático, precisavam limitar significativamente o campo de aplicação do programa de IA, este deveria ter o conhecimento correspondente a um especialista humano que pudesse imitá-lo em seu domínio de trabalho [7].

2.1.1 Alguns Precursores

Esta seção tem como objetivo apresentar um panorama dos principais pesquisadores e precursores na área de IA, demonstrando quais foram suas principais conclusões a respeito dos estudos neste assunto e suas principais contribuições para o desenvolvimento da IA.

Leonardo TORRES Y QUEVEDO, entre outros, em 1911, construiu uma máquina para jogar o final do jogo de xadrez, de rei contra rei e torre, que foi chamada de automata. Vindo a construir outra em 1922. Na construção dessa máquina, ele usou técnicas de resolução de problemas, como ainda é usado em muitas realizações da IA. Apesar disso, sua máquina não foi dotada de inteligência, pois tudo dependia de regras escolhidas pelo construtor [78].

Dentre os precursores da IA podemos ainda destacar os préstimos realizados por George Boole [11] e Lewis Carrol [15]. Boole inventou a *álgebra booleana* e suas idéias se incorporam como base da matemática, filosofia e constituem a base para o projeto de circuitos eletrônicos conhecidos como digitais. Finalmente suas idéias inspiraram o paradigma da programação em lógica. Carrol, por sua vez, mostrou como a lógica encerra raciocínios pouco usuais, podendo ser considerado, mesmo sem que ele soubesse, um precursor da importância da imprecisão em IA.

Mais recentemente, Alain Turing[7] propôs um teste para decidir se um computador exibe inteligência, baseado em um jogo de salão, composto por um homem, uma mulher e um interrogador que fica em local separado dos outros dois. Ganha o interrogador se descobrir, fazendo perguntas a cada um, quem é o homem (Y) e a mulher (X). Ganha a dupla (YX) se conseguirem enganar, com suas respostas, o interrogador. Turing propôs como critério de inteligência este jogo em que um dos elementos da dupla é substituído por um computador, que será considerado inteligente se conseguir ganhar o jogo, não dando ao interrogador, durante um tempo razoável, argumentos convincentes de quem

é humano e quem é a máquina.

Apesar de restringir inteligência a seres humanos, descartando outros animais por não se comunicarem como os humanos e usar frases mal definidas, (como por exemplo, quanto é um tempo razoável de interação entre os participantes para saber a hora de parar e declarar o ganhador), este critério é largamente aceito como teste de inteligência [7].

O primeiro encontro organizado de IA ocorreu nos Estados Unidos, no “Darthmouth College”, em 1956. Lá estavam os pesquisadores: McCarthy, Minsky, Newell e Simon, para o primeiro encontro de duas décadas de estudos. Deste grupo de estudos e do livro intitulado “*Automata Studies*” [84], surgiu o primeiro artigo tratando redes neurais como um paradigma da arquitetura computacional. Nascendo aí os dois paradigmas da IA, que são: “Inteligência Artificial Simbólica (IAS)” e “Inteligência Artificial Conexionista (IAC)” (na época chamado de neural) [7].

Na IAS, tenta-se simular (ou copiar) um comportamento inteligente (considerando os resultados), sem se preocupar com os mecanismos que levam os seres a este comportamento. A preocupação é em chegar aos resultados, não importando, de que forma os problemas são resolvidos (naturalmente) pelos seres dotados de inteligência ou capacidade de fazer analogias. Assim, a IAS se baseia no princípio do sistema simbólico:

“Um sistema simbólico é capaz de manifestar um comportamento inteligente.”

Como a IAS foi alvo de estudos por algum tempo, enquanto a IAC ficava esquecida nos escritos de Rosenblatt, essa já deu muitos frutos, tão importantes que muitas vezes são esquecidos por se tornarem comuns, como por exemplo os programas de jogar xadrez, monitoração, previsões financeiras, diagnósticos, etc. Usam especialmente ferramentas tais como lógica, regras de produção, técnicas de extração de conhecimento de especialistas e sua representação. Além disso, muitos aspectos dos mecanismos de cognição são melhor compreendidos hoje graças aos avanços da IAS [7].

Os primeiros esforços na área de aprendizagem mecânica, tentaram imitar a aprendizagem animal, em nível neural. Métodos esses bem diferentes dos métodos de manipulação simbólica. Imitando o mecanismo de aprendizagem animal, os pesquisadores

esperavam conseguir criar máquinas de aprendizagem, com peças muito simples, o que não foi possível [80].

O progresso da neurobiologia, por volta dos anos 1940, permitiu a construção de modelos matemáticos de neurônios como o primeiro modelo de neurônio artificial proposto por McCulloch e Pitts (1943) [57]. Estes mostraram alguns resultados, com a idéia de criar máquinas inteligentes com neurônios artificiais.

Em 1949, Hebb [39] propôs algumas regras de como as redes neurais aprendiam [32]. Outros pesquisadores seguiram esse caminho nas duas décadas seguintes. Alguns exemplos foram: Ashby(1952) [4], Minsky (1954) [59], Minsky e Selfridge(1961) [59], Block(1962)[10] e Rosenblatt(1962) [81]. Por volta da década de 1970, os estudos com redes neurais foram se retraindo, uma vez que, as redes usadas na época mostraram-se muito fracas computacionalmente [80].

No começo dos anos 1980, ressurgiu o interesse por essa técnica, com a descoberta de algoritmos de aprendizagem mais poderosos¹. Hopfield (1982), introduziu a ideia de minimização de energia, em Física, dentro de redes neurais. O termo conexionista primeiramente utilizado por Eduard Li Thorndike, em sua teoria denominada conexionista na qual a conexão entre as impressões, os sentidos e as ações de impulso como sendo a conexão neural entre o estímulo e as respostas [40], tornando-se popular com Feldman e Ballard (1982) [32].

A IAC foi buscar no cérebro humano inspiração, pois acredita-se que construindo máquinas que imitem (seja por *software* ou *hardware*) a estrutura do cérebro, elas apresentarão inteligência. A abordagem da IAC se baseia no seguinte axioma [7]:

“Se for construído um modelo suficientemente preciso do cérebro, este modelo apresentará um comportamento inteligente. Se apenas uma pequena parte do cérebro for reproduzida, a função exercida por esta parte emergirá do modelo.”

A IAC é mais jovem, pois só recentemente atraiu a atenção de profissionais das mais variadas áreas, alguns resultados já foram obtidos e outros estão sendo obtidos a cada dia.

¹Algoritmos de aprendizagem são apresentados no capítulo 3.

O livro Perceptron [60] de Minsky e Papert, de 1969, fez com que progressivamente as duas correntes para IA separassem-se e as pesquisas em redes neurais (corrente conexionista) andassem lentamente, enquanto a corrente da manipulação simbólica acelerou-se.

Atualmente, as duas abordagens estão cada vez mais sendo usadas em conjunto, uma completando a outra, formando o que se conhece como Inteligência Artificial Híbrida (IAH) [13]. A arquitetura ótima para sistemas inteligentes no futuro, pode bem envolver sua integração de uma maneira ou de outra [32].

2.1.2 As fases da IA

Segundo Barreto [7], pode-se dividir a história da IA em algumas fases, inspiradas nas fases da evolução da humanidade. Nesse divisão, foram propostas 7 (sete) épocas, quais sejam:

1. *Época Pré-histórica*: Até 1875, quando Camillo Golgi visualizou o neurônio.

Nessa época, desejava-se criar seres e mecanismos que apresentassem comportamento inteligente, para tal, foram utilizados, mecânica de precisão, desenvolvida nos autômatos. As principais limitações eram a complexidade dos mecanismos e as dificuldades de construção dos mesmos.

2. *Época Antiga*: De 1875 até 1943, quando McCulloch e Pitts publicaram seu trabalho modelando o neurônio.

Nesse período, o objetivo era entender a inteligência humana, para esse entendimento foi necessário o estudo da psicologia e da neurofisiologia, porém esses estudos limitavam-se pela grande distância entre as conquistas da psicologia e da neurofisiologia “*Trabalho de Norbert Wiener de Cibernetica publicado logo após a guerra, por problemas de sigilo*”.

3. *Época Romântica*: De 1943 até 1956, marcada pela reunião no Dartmouth College.

Essa fase baseava-se em simular inteligência humana em situações pré-definidas. Inspirando-se na natureza e na biologia, apareceram os primeiros mecanismos imitando o funcionamento de redes neurais e os primeiros programas imitando o comportamento inteligente. A barreira aqui foi a limitação da capacidade dos computadores. Também foram realizados muitos trabalhos na linha simbólica; ex: GPS - General Problem Solver

4. *Época Barroca*: De 1956 até 1969, quando foi publicado o livro “Perceptrons”.

Nesse período, o objetivo era expandir ao máximo as aplicações da IAS e da IAC. Nessa época surgiram os primeiros SE utilizando a abordagem simbólica. A limitação, aqui, era as dificuldades nas técnicas de aprendizado de redes complexas. Isto fez com que as pesquisas se concentrassem no paradigma simbólico.

5. *Época das Trevas*: De 1969 até 1981, com o anúncio dos planos para a quinta geração de computadores feito pelos japoneses.

O objetivo dessa fase foi encontrar aplicações práticas para a IA, pois eram aplicadas, principalmente, nos laboratórios de pesquisa, não sendo aplicados ainda às empresas. O que limitou esse período foram os interesses econômicos. (*Prolog* adotado como linguagem. Grande sucesso da abordagem simbólica).

6. *Época do Renascimento*: De 1981 até 1987, marcada pela primeira conferência internacional em Redes Neurais.

Nessa época renasceu a IAS e a IAC, retomando o crescimento dos estudos da IA, com a proliferação de computadores que suportavam ferramentas para IA, ainda houve uma popularização da linguagem Prolog e o crescimento da importância da Lógica. No começo do período, houve uma redução significativa no número de pesquisadores interessados em Redes Neurais Artificiais (RNA). Voltando a crescer, principalmente, após os trabalhos de Hopfield e do grupo *Parallel Distributed Processing (PDP)*. A limitação principal se deu justamente pela separação dos estudos da IAS e da IAC.

7. *Época Contemporânea*: A partir de 1987.

Essa época objetivou o alargamento das aplicações de RNA, tendo com as RNA bons resultados com problemas mal definidos, Redes Diretas como aproximador universal e o surgimento do paradigma de programação conexionista.

Nesta época novas idéias foram incorporadas, no entanto não serão tratadas em detalhes por fugirem da finalidade do presente trabalho. As principais são: Lógica Nebulosa, Computação Evolucionária e suas variantes (Algoritmos Evolucionários [29] [45], Estratégias Evolucionárias [83], Algoritmos Genéticos [42] [43]).

Capítulo 3

Redes Neurais Artificiais

3.1 Introdução

As RNA são inspiradas nos sistemas nervosos biológicos. Se tenta reproduzir (pelo menos em parte) o funcionamento do cérebro humano em computadores, seja por “*hardware*” ou “*software*”. Objetiva-se, com isso, dotar os computadores da capacidade de “aprender”, para resolver problemas aos quais ele não foi explicitamente programado para resolver, mas que fazem parte de um mesmo domínio já conhecido.

Atualmente, as arquiteturas de redes estão sendo intituladas arquiteturas “conexionistas” (*por usarem conexões e pesos associados a elas* [32]). Porém, pode-se encontrar em literaturas diversas denominações correlatas, tais como: Sistema neural artificial, Rede conexionista, Conexionismo, Neurocomputadores, Sistemas de processamento paralelo distribuído (“*Parallel distributed processors systems*”), entre outros [68] [64].

O principal objetivo dessas arquiteturas não é duplicar o funcionamento do cérebro humano, mas receber inspiração de fatos conhecidos do funcionamento deste. Estas arquiteturas são caracterizadas por possuírem [80]:

1. Grande número de elementos de processamento parecidos com os neurônios, tentando simular seu funcionamento;
2. Um grande número de conexões entre os elementos (similar às conexões sinápticas).

Elas apresentam valores variáveis, os quais codificam o conhecimento armazenado de uma rede;

3. Ênfase na aprendizagem automática de representações internas.

O estudo das redes neurais é um campo extremamente interdisciplinar, tanto no desenvolvimento quanto nas aplicações. As redes neurais podem ser aplicadas nas mais diversas áreas, indo desde atividades de pesquisa até aplicações comerciais e industriais.

Sistemas Neurais Artificiais são sistemas celulares físicos que podem adquirir, armazenar e utilizar aprendizado por experiência. As seguintes características de redes neurais tem disputado um importante papel em uma grande variedade de aplicações [68]:

- **Adaptativo:** Poderosos algoritmos de aprendizagem e regras de auto-organização, permitindo auto adaptação, tanto mediante necessidade, como continuamente mudando o ambiente;
- **Processamento não-Linear:** Habilidade para realizar tarefas que tenham afinidades não-lineares, é um bom candidato para classificação e prognóstico;
- **Processamento Paralelo:** Arquiteturas, com uma grande quantidade de unidades de processamento, aumentam a interconectividade para processamento concorrente tão bem como armazenagem de informações paralela e distribuída.

3.2 Fundamentos Biológicos

O descobridor da célula nervosa ou neurônio foi o anatomista italiano Camilo Golgi em 1875, através de sua técnica de tingimento dos neurônios. Este tingimento mostra apenas uma pequena parte dos neurônios, permitindo visualizá-los, individualmente, em toda sua forma (ramificações). Cada neurônio possui um corpo celular, ou soma, dentro do qual encontra-se a maioria dos seus orgânulos. Do soma de cada neurônio partem prolongamentos que podem ser funcionalmente divididos em dendritos e axônios, como visto na figura 3.1 [61] [41].

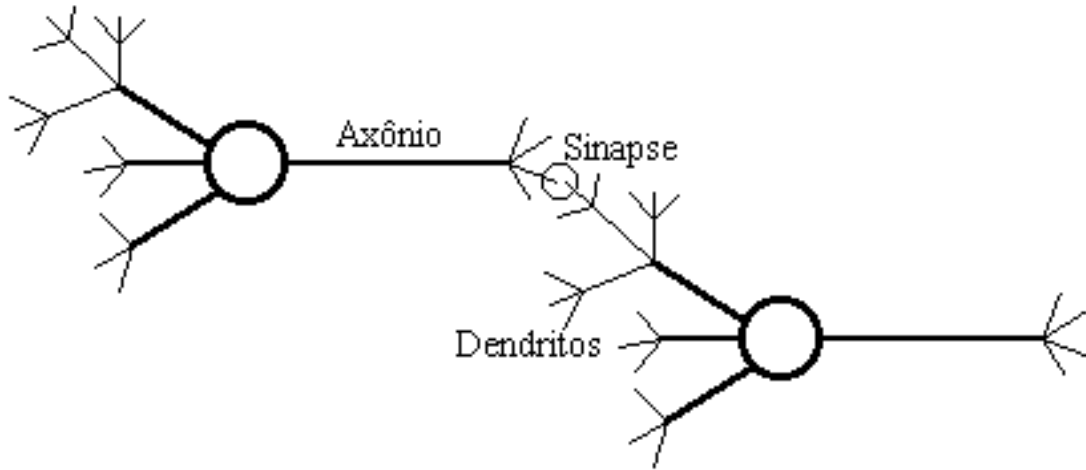


Figura 3.1: Neurônio e suas conexões.

Outro pesquisador, o neurologista espanhol Santiago Ramón Y Cajal, adotou a noção de sistema nervoso, postulando sobre a comunicação entre os neurônios, através das sinapses. A sinapse conecta um axônio de um neurônio a um dendrito de outro. Outra grande conclusão alcançada por Cajal foi que a interconexão dos neurônios não é feita aleatoriamente (ao acaso), pois é uma estrutura altamente específica.

O neurônio é uma célula altamente especializada. Como qualquer célula biológica, é delimitado por uma fina membrana celular que, além de sua função de separar interior do exterior da célula, possui determinadas propriedades que são essenciais para o funcionamento elétrico da célula nervosa. O neurônio recebe impulsos elétricos a partir de seus dendritos. Esses impulsos são processados no soma (corpo celular) e retransmitidos pelo seu axônio, para os dendritos de um ou vários outros neurônios [38].

Segundo Laurene Fausett [26] e M. M. Nelson [64], o neurônio biológico possui três partes particularmente importantes para o estudo dos neurônios artificiais, são elas: *Dendritos, Soma e Axônio*.

- Os dendritos tem a função de receber as informações (impulsos nervosos) de outros neurônios e conduzi-los ao soma para que sejam processados. Possuem uma forma similar as raízes de árvores;

- O soma, ou corpo celular, possui uma forma piramidal ou cilíndrica e este é o núcleo da célula nervosa. A informação vinda de muitos dendritos é processada, gerando, assim, novos impulsos que serão retransmitidos. O soma age como um processador de informações, somando o potencial elétrico vindo dos dendritos. Ele é o responsável por prover as funções de suporte necessárias do neurônio [68];
- O axônio de um neurônio, geralmente é único e pode se prolongar por distâncias comparáveis as dimensões do organismo. Também é o responsável por carregar os impulsos nervosos gerados no soma do neurônio até a sinapse. O axônio pode ser visto simplesmente como uma via de transporte, ou uma linha de transmissão. Eles tem uma forma lisa, com poucas ramificações comparados com os dendritos, os quais possuem uma forma irregular;
- Uma sinapse consiste num terminal pré-sináptico, por onde chega um estímulo proveniente de uma outra célula e o terminal pós-sináptico, que é a do dendrito. Nesta região intersináptica, o estímulo nervoso que chega à sinapse é transferido à membrana dendrital através de substâncias conhecidas como neurotransmissores. O resultado desta transferência é uma alteração no potencial elétrico da membrana pós-sináptica. Dependendo da natureza dos neurotransmissores e pela natureza dos receptores químicos que os recebem no outro lado da sinapse, a conexão será excitatória ou inibitória. [88]

Em uma sinapse inibitória, a transmissão sináptica provoca uma pequena hiperpolarização no potencial elétrico do neurônio afetado. Isto faz com que seja mais difícil para este neurônio se despolarizar e propagar seu próprio potencial de ação ao longo do seu axônio [88]. Já em uma sinapse excitatória, ocorre uma pequena despolarização no neurônio afetado, rebaixando seu potencial elétrico em direção a um ponto mínimo crítico, onde ele repentinamente pode se despolarizar a ponto de iniciar a transmissão de seu próprio potencial de ação, através do seu axônio.

A transmissão do sinal de um neurônio em particular é dotado por um potencial de ação, resultado da diferença de concentração de íons entre os dois lados da membrana do neurônio. A maior parte dos íons são diretamente envolvidos por potássio, sódio e

cloro [26].

Em várias partes do cérebro, existem uma grande variedade de neurônios, cada um com forma e tamanho diferentes. Com isso, existe um grande número de tipos diferentes de junções sinápticas entre os neurônios.

O cérebro humano possui cerca de 100 bilhões (10^{11}) células nervosas ou neurônios, estes, por sua vez, possuem aproximadamente 100 trilhões (10^{14}) de conexões (sinapses) tendo uma densidade de 1000 conexões por neurônio em média. Isto resulta num estonteante número de neurônios e sinapses, pois o cérebro é uma estrutura extremamente complexa [68].

3.3 Organização Hierárquica do Cérebro

A organização estrutural do cérebro, com suas diferentes funções, tomando lugar nos maiores e menores níveis do cérebro, foram reveladas através das pesquisas e análises das regiões do cérebro [68]. A hierarquia dos níveis de organização no cérebro pode ser apresentada com a seguinte sequência:

- 1. Sistema Nervoso Central**
- 2. Circuitos Inter-regionais**
- 3. Circuitos Locais**
- 4. Neurônios**
- 5. Microcircuitos Neurais**
- 6. Sinapses**
- 7. Membranas, Moléculas e Íons**

3.4 Modelo do Neurônio Artificial

3.4.1 Modelo de McCulloch e Pitts

Os precursores na modelagem neural foram McCulloch e Pitts que em 1943 [57] propuseram o primeiro modelo de neurônio artificial tentando imitar a realidade biológica. Seu modelo de neurônio funcionava como sendo um circuito binário, conforme a Figura 3.2

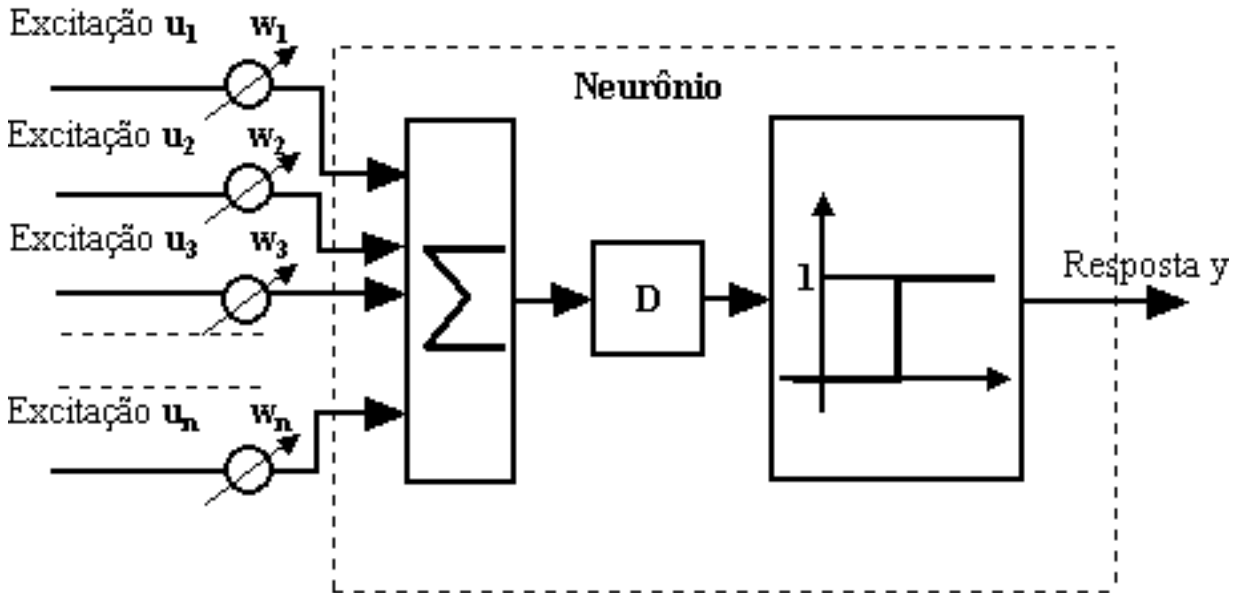


Figura 3.2: Modelo de neurônio de McCulloch e Pitts.

Onde as entradas do neurônio, que também são binárias, são combinadas por uma operação de *confluência* (soma ponderada), produzindo a entrada efetiva do neurônio:

$$u_{efetivo} = \sum w_i y_i \quad (3.1)$$

Na equação, w_i é o valor da conexão sináptica do neurônio considerado e entrada y_i . O resultado da entrada efetiva, passa por um retardo D (que pode ser desprezado) e servirá de entrada para uma função de transferência, gerando a saída do neurônio. No modelo em questão, a saída será uma resposta binária 0 ou 1.

3.4.2 Modelo Geral

Uma generalização do modelo de McCulloch e Pitts é conhecida como o modelo geral de neurônio, mostrado na Figura 3.3.

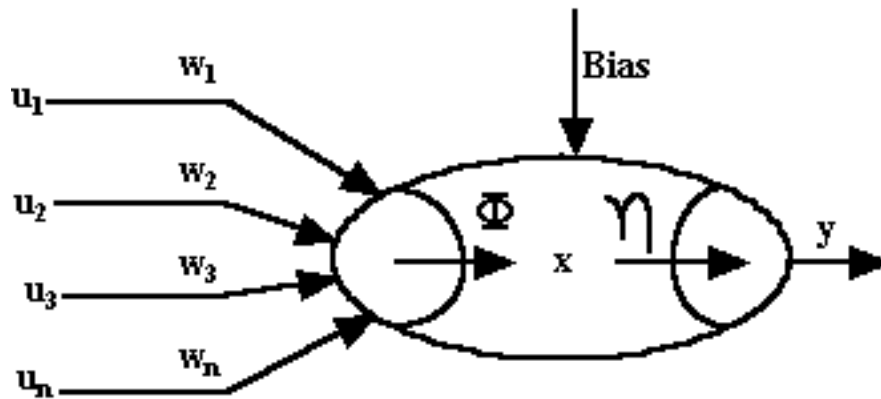


Figura 3.3: Figura do neurônio geral.

As entradas desse neurônio u_i , podem estar derivando de outros neurônios ou do exterior e são combinadas por um valor de peso (força de conexão) $w_1..w_n$, que podem ser excitatórias se $w_n > 0$ ou inibitórias se $w_n < 0$, geralmente variando de -1 a 1. A combinação das entradas com os pesos, passam por uma função de ativação, geralmente não linear, que irá produzir a saída do neurônio. Um valor de “bias” é geralmente usado para representar uma polarização, abaixo do qual a saída é nula.

Pode-se considerar três fases no núcleo do neurônio: a primeira, com a função soma, que recebe todas as entradas, gerando um único valor. Este valor é alterado pela função de ativação, responsável pela ativação ou não do neurônio. Por fim, a função de transferência repassa o resultado da função de ativação para o próximo neurônio.

Além dos valores de saída do neurônio poderem ser diferentes de 0 e 1, como no caso do modelo de McCulloch e Pitts, no modelo geral também podem ser usadas funções de ativação diferentes, tais como: a função linear, a função rampa, a função degrau ou a função sigmoideal, que estão representadas na Figura 3.4 [24] [64] [68] [26].

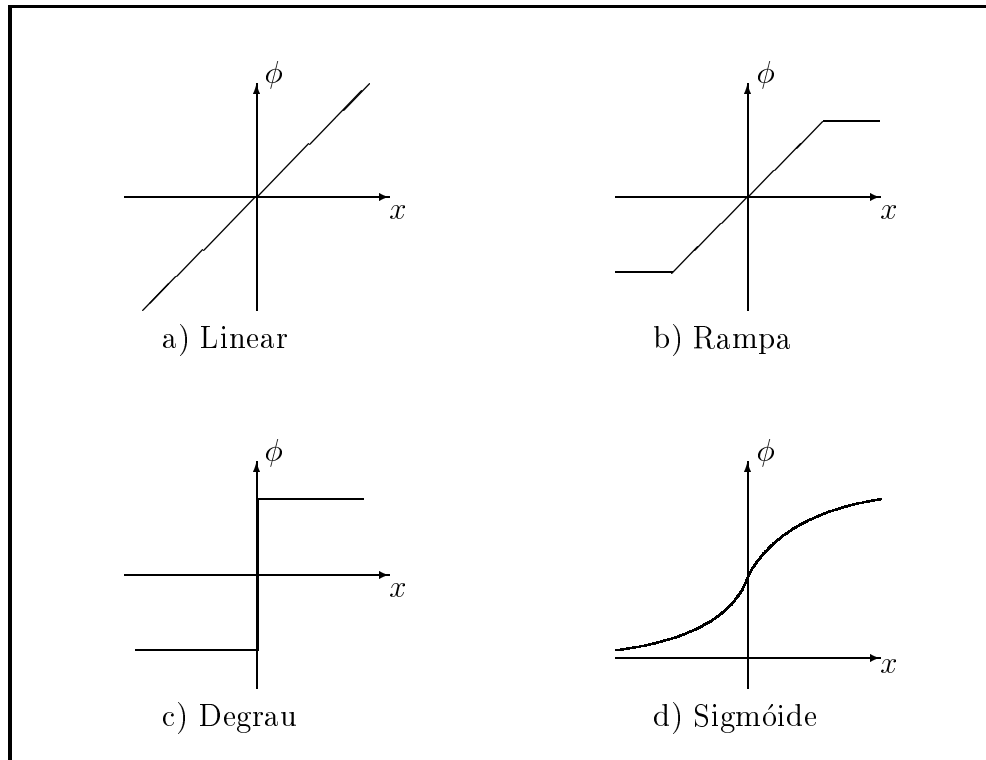


Figura 3.4: Algumas funções de ativação

A função de ativação linear é definida matematicamente por:

$$\phi_i(x) = \alpha x \quad (3.2)$$

Onde α é um número real, que define o valor do ganho de ϕ para os valores da ativação interna do neurônio x .

A função de ativação rampa é definida matematicamente por:

$$\phi_i(x) = \begin{cases} +\gamma & \text{iff } x \geq \gamma \\ x & \text{iff } |x| < \gamma \\ -\gamma & \text{iff } x \leq -\gamma \end{cases} \quad (3.3)$$

Os valores máximo e mínimo da saída são γ e $-\gamma$, respectivamente. A função rampa é geralmente usada como uma função não-linear simplificada.

A função de ativação degrau é definida matematicamente por:

$$\phi_i(x) = \begin{cases} \gamma & \text{iff } x > 0 \\ -\gamma & \text{iff } x \leq 0 \end{cases} \quad (3.4)$$

Esta função produz como saída o valor γ se $x > 0$ e $-\gamma$, caso contrário.

A função de ativação sigmoideal pode ser definida de várias formas, dependendo da área de aplicação. Uma das funções sigmoideais mais importantes é a função logística, definida por:

$$S_i(x) = \frac{1}{1 + e^{-\frac{x}{T}}} \quad (3.5)$$

Onde o parâmetro T determina o formato da curva.

Na biologia, os neurônios estão interligados, formando uma rede, denominada Rede Neural Natural (RNN) ou Rede Neural Biológica (RNB). Os neurônios artificiais sozinhos, não tem quase nenhum poder computacional. Sendo assim, mais uma vez aproveita-se a inspiração biológica para juntar os neurônios formando as redes neurais artificiais, aumentando assim o seu poder computacional. Uma vez que, interligados

os neurônios passam a ter um comportamento emergente, lembrando em alguns casos, os biológicos.

3.5 Topologias de Redes Neurais Artificiais

Pode-se dividir as RNA, de acordo com sua topologia, em dois tipos distintos: redes sem realimentação (as redes diretas, não recursivas, direcionais) e redes com realimentação. Nas redes sem realimentação, a informação flui em alguma direção, de modo que, ao ser processada por um neurônio, não retorne a ele. Já nas redes com realimentação, a informação processada pelo neurônio poderá ser novamente utilizada por ele, ou por outro neurônio de uma camada anterior (vide Figura 3.7).

Independente de serem com ou sem realimentação, estas redes podem ter conexões simétricas ou assimétricas. Seja a rede neural N com n neurônios indicados pelas letras a, b, c, \dots, n e sejam w_{ij} as conexões entre os neurônios, neste caso os neurônios i e j . Excluindo as conexões de entrada e saída da rede, ela é dita “Simétrica” se:

$$\forall i, j \in N \tau w_{ij} = w_{ji} \quad (3.6)$$

Caso contrário, se estas condições não forem satisfeitas, as conexões são ditas “Assimétricas” [32]¹.

As conexões entre as camadas de uma rede podem ter várias estruturas diferentes, de acordo com a composição das conexões. Pode-se ter cada neurônio de uma camada ligado a apenas um outro neurônio na outra camada, assim como pode-se ter todos os neurônios de uma camada conectados a todos os neurônios da outra camada (sem realimentação). Também pode-se ter a saída de um neurônio, conectada a entrada do mesmo neurônio ou a outro neurônio de uma camada anterior, neste caso tem-se uma rede com realimentação. Ainda, pode-se ter a conexão entre neurônios de camadas alternadas, tanto para frente (direta, sem realimentação), como para trás (realimentadas) [91].

¹A matriz W de elementos w_{ij} é a conhecida como matriz de conectividade, que é uma matriz composta pelo valor das conexões entre os neurônios (sinapses). Esta matriz é quadrada e possui o tamanho (número de linhas e colunas) igual ao número de neurônios da RNA.

Quanto ao número de camadas e de neurônios por camada, as redes neurais podem aparecer de diversas formas. Sendo que, globalmente falando, pode-se considerar apenas três camadas, são elas:

- *Camada de Entrada:* Responsável por receber os impulsos de entrada da rede. Terá um número de neurônios relativo a quantidade de entradas possíveis (um neurônio para cada sinal possível de entrada);
- *Camada de Saída:* Responsável por apresentar os resultados decorridos do processamento da rede. O número de neurônios dessa camada depende da quantidade de saídas esperadas para o sistema;
- *Camada Intermediária:* Esta camada pode conter de 1 a n camadas, dependendo da forma de rede escolhida, ou da necessidade do problema em questão. Assim como, estas camadas podem conter um número de neurônios variáveis. Geralmente escolhidos por métodos estatísticos, ou por tentativas de aproximação do resultado. Os neurônios desta camada são chamados *neurônios intermediários* como tradução do inglês.

Tomando-se aqui, como exemplo, a rede “*NETtalk*”, proposta por McCord em seu livro “*A practical Guide to Neural Nets*” [64]. A *NETtalk* tem 203 neurônios de entradas, para representar 7 (sete) caracteres de posições na janela e os 29 caracteres entre: alfabéticos, vírgula, espaço e o período ($7 \times 29 = 203$). Da mesma forma, tem-se 26 neurônios de saída; 1 (um) para cada um dos 23 (vinte e três) diferentes sons e 3 (três) para acentuar (stress). A quantidade de neurônios da camada intermediária não é óbvia como as camadas de entrada e saída. Neste exemplo, foram usados 80 (oitenta) neurônios para fala contínua.

Quando a estrutura do padrão de entrada é completamente diferente do padrão de saída, as unidades intermediárias são necessárias para criar uma representação interna do sinal de entrada. A capacidade de processar informações aumenta na proporção do número de camadas da rede. Este é um poderoso aspecto. Padrão de entrada pode sempre ser recodificado a fim de adequar a saída se há suficientes unidades intermediárias [64].

3.5.1 Redes Diretas (“*Feedforward*”)

Redes Diretas (*Feedforward*)² são aquelas que podem ser representadas por um dígrafo acíclico. Estas redes só possuem uma memória de curtíssimo prazo, sendo as únicas informações armazenadas relativas ao processamento imediatamente anterior de cada neurônio. Seus sinais se propagam somente num sentido e as saídas só dependem dos sinais que estão chegando dos outros neurônios. Uma vez treinadas, estas redes produzem sempre o mesmo resultado de saída, para uma mesma entrada. Estas redes geralmente são organizadas em camadas e, neste caso, são chamadas redes em camadas (“*multilayer*”). Neurônios que recebem sinais de excitação são chamados de camada de entrada, ou primeira camada. Neurônios que tem sua saída, como saída da rede, pertencem a camada de saída ou última camada. Neurônios que não pertencem nem a camada de entrada nem a de saída, são neurônios internos à rede e pertencem a uma ou mais camadas internas (“*hidden layers*”).

O primeiro paradigma usado na implementação de sistemas especialistas com redes neurais usava redes diretas e foi proposto por Gallant [18]. Baseia-se em que as redes diretas são aproximadores universais de funções. Assim, apresenta-se na entrada da rede os dados e treina-se a rede para a saída representar o parecer (resultado) do sistema especialista.

Esta rede geralmente é treinada de modo supervisionado e, superficialmente falando, pode ser descrito como: dado um vetor com o padrão a ser reconhecido. Insere-se esse vetor na camada de entrada da rede, que repassará os valores para as camadas intermediárias que a seguem e finalmente para a camada de saída através dos pesos das conexões, obtendo-se dessa forma o resultado. Cada neurônio da rede opera realizando a soma dos pesos de entrada e repassando o resultado para a próxima camada, ou para a saída da rede, através de uma função de ativação não-linear.

Como apresentado matematicamente:

$$out_i = f \left(\sum_j W_{ij} out_j + \theta_i \right) \quad (3.7)$$

²Não se pretende que o termo em português seja a tradução de “*feedforward*” mas de “*direct networks*” termo inicialmente usado.

Onde:

out_i - Saída do i – *esimo* neurônio da camada inferior

out_j - Saída do j – *esimo* neurônio precedendo a camada ativa

f - Função de ativação não-linear (na maioria das vezes é usada uma função sigmóide. Como já visto na Equação 3.5)

Estas redes são muito utilizadas atualmente, por serem bastante conhecidas e por existirem métodos de aprendizado fáceis de usar, como o de retropropagação que é o mais difundido para esse tipo de rede³.

Outra característica destas redes é a capacidade de aproximar com maior ou menor precisão, qualquer função não-linear, dependendo do número de neurônios da rede.

Como estas redes não possuem ciclos (realimentação), elas são estáticas. Mesmo assim é possível introduzir dinâmica nessas redes, através da utilização de neurônios dinâmicos (equação diferencial de primeira ordem ou a uma diferença finita). Apesar dessa possibilidade, sua dinâmica é limitada, não sendo possível representar todos os sistemas dinâmicos [7].

A Figura 3.5 mostra uma rede direta com uma única camada, sendo composta por 3 (três) neurônios na camada de entrada e 3 (três) na camada de saída. Estas redes tem a limitação de não resolverem problemas classificados como linearmente não separáveis, como demonstrado por Minsky e Papert.

Outra possibilidade é a mostrada na Figura 3.6. Uma rede com uma ou mais camadas internas, chamada rede multi-camadas. Os neurônios das camadas internas extraem características do neurônios da camada de entrada, transmitindo-as para os neurônios da camada de saída.

³O algoritmo de treinamento retropropagação será apresentado com mais detalhes no decorrer desse capítulo, no ítem 3.8.3.

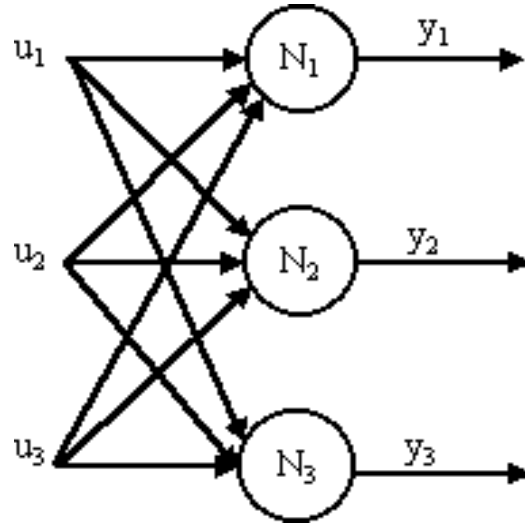


Figura 3.5: RNA Direta de uma camada.

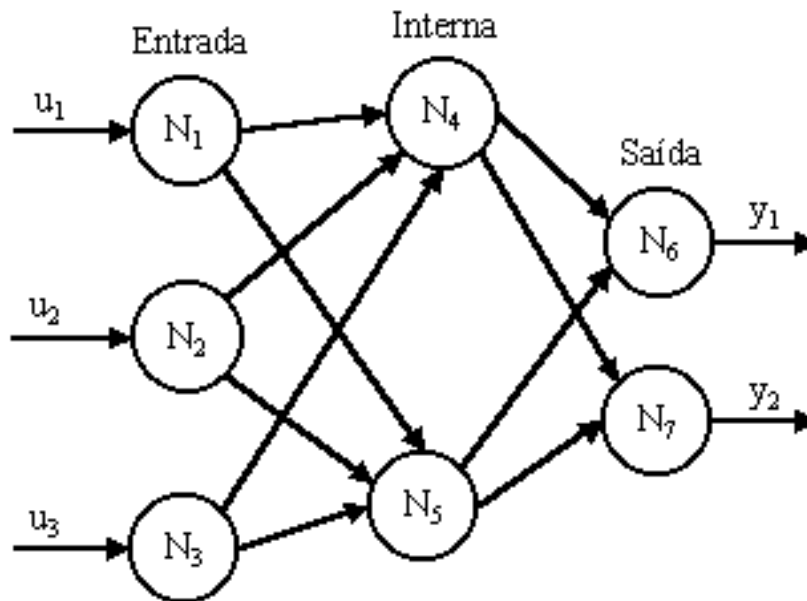


Figura 3.6: RNA com 3 camadas.

3.5.2 Redes com realimentação ou ciclos (“*Feedback*”)

As redes com realimentação são aquelas que podem ser representadas por dígrafos com ciclos (pelo menos um ciclo), conforme a Figura 3.7. Quando estas redes também envolvem neurônios dinâmicos (a tempo discreto) são chamadas de recorrentes.

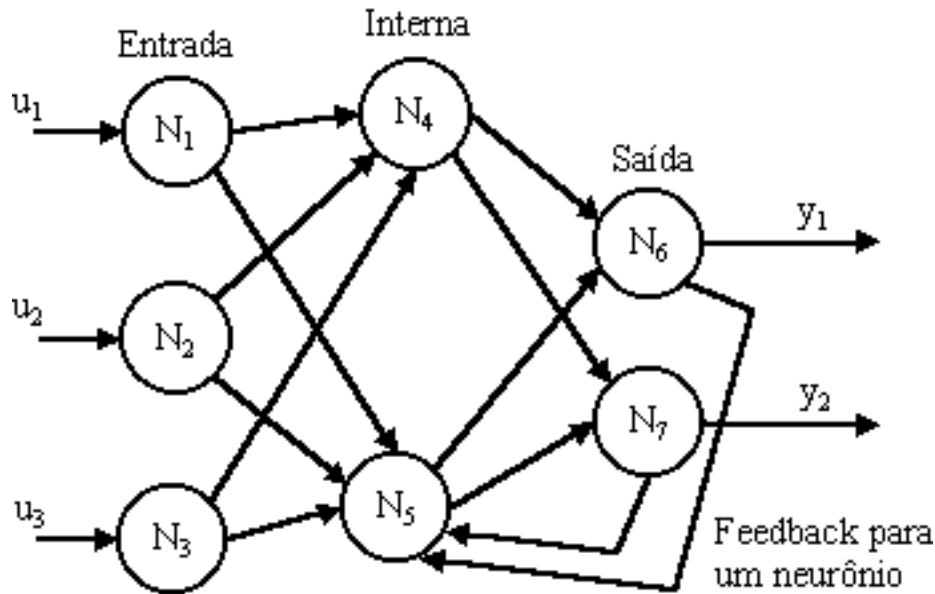


Figura 3.7: RNA com ciclos.

Duas destas redes tem particular importância: as redes propostas por Hopfield [30, 31] e as redes bi-direcionais [33] que podem ser usadas em um dos dois principais paradigmas de SE, são elas: treinamento com exemplos de uma rede direta e representação do conhecimento de modo localizado e uso de rede com ciclos, geralmente uma rede simétrica.

O segundo paradigma usa redes bidirecionais, caso particular das redes com ciclos, contendo neurônios dinâmicos [5] [20]. Neste caso, tanto os dados, como os possíveis diagnósticos do especialista são representados pela ativação de neurônios, o conhecimento sendo representado pelos valores das intensidades de conexões sinápticas. Uma consulta é feita excitando neurônios representativos dos sintomas presentes no caso,

deixando a rede evoluir até atingir um ponto de equilíbrio. A excitação de algum neurônio representando o diagnóstico será a resposta do sistema.

Existe um caso particular das redes com ciclos, que são as redes simétricas. Uma rede simétrica é aquela cuja matriz de conectividade é uma matriz simétrica. Os SE mencionados anteriormente e que usam rede com ciclos, também usam redes simétricas. A Figura 3.8 representa uma rede simétrica com neurônios internos.

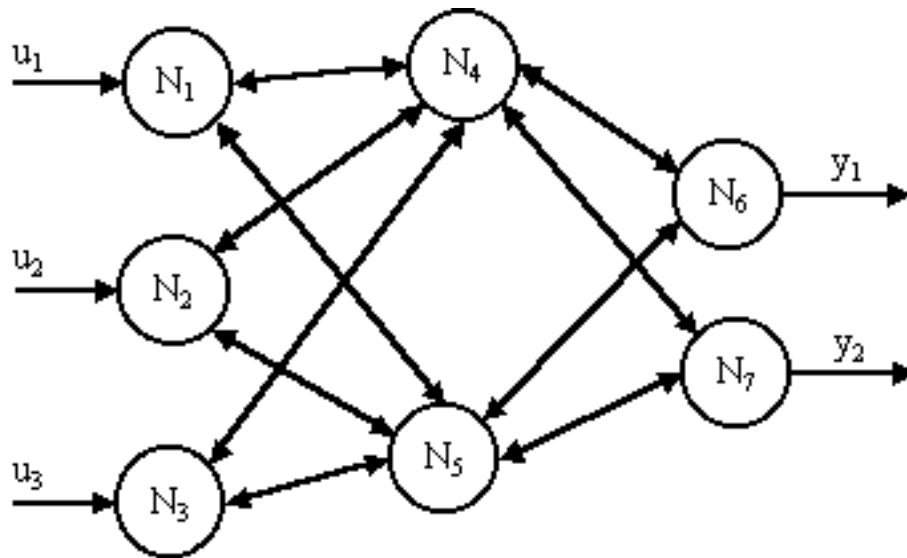


Figura 3.8: RNA Simétrica com camada interna.

3.5.3 Redes de Memórias Associativas Bidirecionais

Redes BAM são implementadas sem a necessidade de treinamento posterior, pois o treinamento (que aqui não existe) é embutido no próprio corpo da rede. É na matriz de pesos e conexões, que se inclui os dados referentes ao conhecimento da rede BAM. Para cada conjunto de conceitos, existe um aglomerado de neurônios correspondente. Deve-se inicialmente, organizar estes conceitos em conjuntos de conceitos semelhantes. Tomando como exemplo o diagnóstico médico. Têm-se os seguintes conjuntos de conceitos: doenças, diagnósticos e pacientes [6]. A conexão dos objetos é feita, utilizando o conhecimento que se tem a respeito da relação entre esses objetos (geralmente va-

riando entre 1 e -1). Utilizando o exemplo médico, supondo que um paciente tenha um sintoma, coloca-se a conexão entre o sintoma e o paciente, com o valor referente a quão grave é este sintoma. Por outro lado, tem-se a conexão entre os sintomas e as doenças, que neste caso seria, quão provável esse sintoma gera determinada doença. Esses valores das conexões são subjetivos, e podem variar de um especialista para outro. Apesar dessa subjetividade, a rede consegue generalizar o suficiente, para responder corretamente.

A matriz sináptica é simétrica, com suas dimensões, dependendo do número de objetos utilizados no problema. Os valores contidos nessa matriz correspondem as conexões existentes entre os objetos, como citado acima, variam de -1 a 1, sendo que os valores positivos representam excitação, os negativos inibição e o valor nulo ou zero, para irrelevância ou inexistência de conexão.

O funcionamento dessa rede, pode ser expresso matematicamente, por:

$$y_i = f(x_i \cdot w_{ij}) \quad (3.8)$$

Onde:

y_i = vetor de saída da rede

f = função de atribuição

x_i = vetor de entrada da rede

w_{ij} = matriz de conexões (pesos)

Esta rede deve ser implementada com ciclos, onde $x_{i+1} = y_i$. O resultado será obtido quando a rede atingir o equilíbrio, com a entrada igual a saída, ou seja quando $y_i = x_i$ [50].

3.6 Caracterização das RNA

Como nas RNN, aqui os neurônios estão interligados por conexões sinápticas, sendo que os neurônios podem estar em três diferentes situações, dependendo das suas conexões. Se receber estímulos do exterior, são os neurônios de entrada da rede; os neurônios responsáveis por emitir os resultados da rede (para o exterior), são os neurônios de

saída da rede e os demais neurônios, os que recebem e enviam estímulos para outros neurônios, são os neurônios intermediários. Os neurônios intermediários são de grande importância, principalmente pelo fato de que, sem eles, não é possível para uma RNA resolver problemas classificados como linearmente não separáveis.

Para facilitar a compreensão, basta visualizar a representação gráfica na Figura 3.9, que possui três neurônios de entrada, dois de saída e dois internos. As conexões sinápticas estão destacadas para ressaltar o fato de que as conexões são externas aos neurônios. A função das conexões é transmitir a saída de um neurônio para a entrada de outro, sem interferir internamente no neurônio. Comumente, estas conexões são simplesmente representadas pela seta direcional de um neurônio para outro.

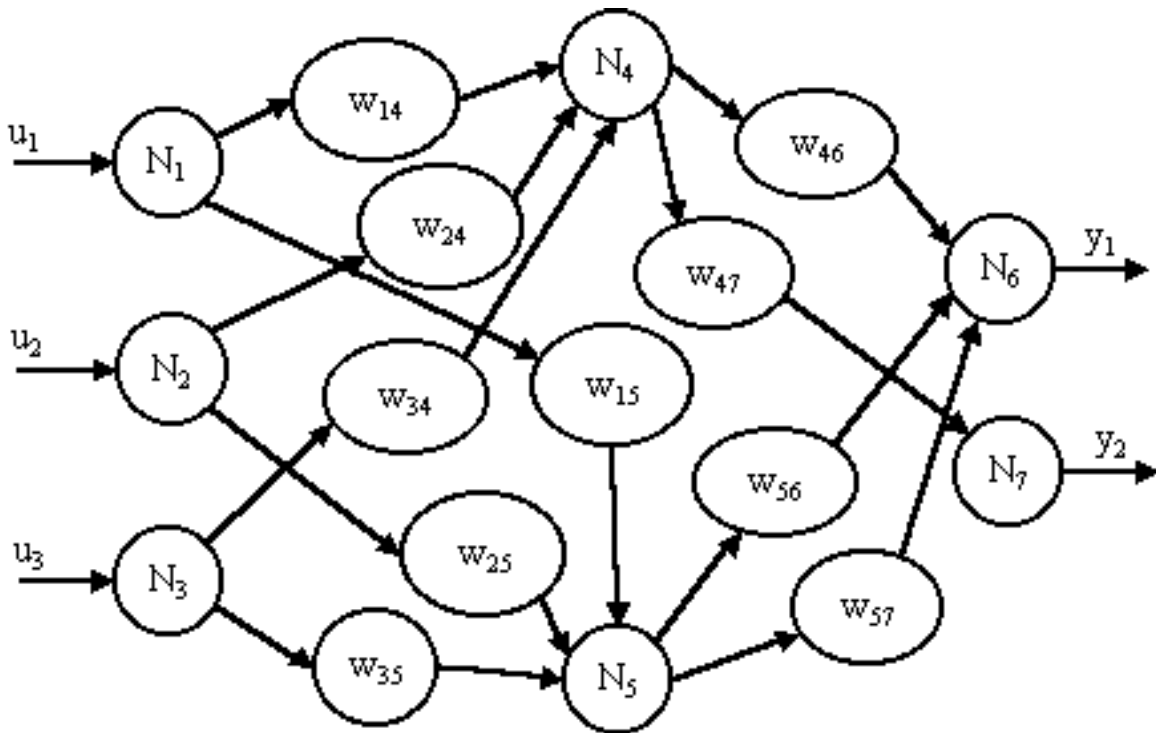


Figura 3.9: RNA com as conexões ressaltadas.

3.7 Aprendizado em RNA

Uma RNA pode aprender de várias formas. A principal finalidade é de que dado um determinado conjunto de entradas, se produza um conjunto esperado na saída. Para realizar tal tarefa, existem alguns métodos para determinar a força das conexões da rede. Uma maneira é definir e atribuir explicitamente os pesos a cada uma das conexões da RNA, usando o conhecimento de um especialista, como no caso das rede bidirecionais. Outra forma é treinar a rede através de um algoritmo de treinamento, que irá automaticamente ajustar os pesos da rede.

Para esse treinamento, existem basicamente duas técnicas, a do aprendizado supervisionado e aprendizado não supervisionado. O que distingue as duas técnicas é a presença ou não de um professor ou supervisor. O professor pode ser um conjunto de dados de treinamento, ou um observador que qualifica o desempenho.

3.7.1 Aprendizado Supervisionado

Para o aprendizado supervisionado, apresenta-se à rede sempre pares de entradas e saídas, onde as entradas levarão a rede a produzir uma saída, que será comparada com a saída desejada. Caso a resposta produzida pela rede seja diferente da resposta esperada, a rede gera um erro, que será usado para calcular o ajuste dos pesos sinápticos. Os pesos das conexões são alterados para aproximar os resultados das saídas esperadas. Esse processo se repete com todo o conjunto de treinamento (entradas e saídas), até que a rede atinja um nível de erros aceitável, ou nenhum erro. Este é o caso em que as respostas geradas pela rede coincidem com as respostas esperadas. Outra opção, ao final do treinamento, é quando a o aprendizado da rede não converge. Neste caso, pode ser aumentado o número de repetições (épocas) do treinamento, ou alterada a configuração da rede (alterando o número de neurônios de cada camada, ou o número de camadas da rede).

A Figura 3.10, mostra o esquema de funcionamento do aprendizado supervisionado.

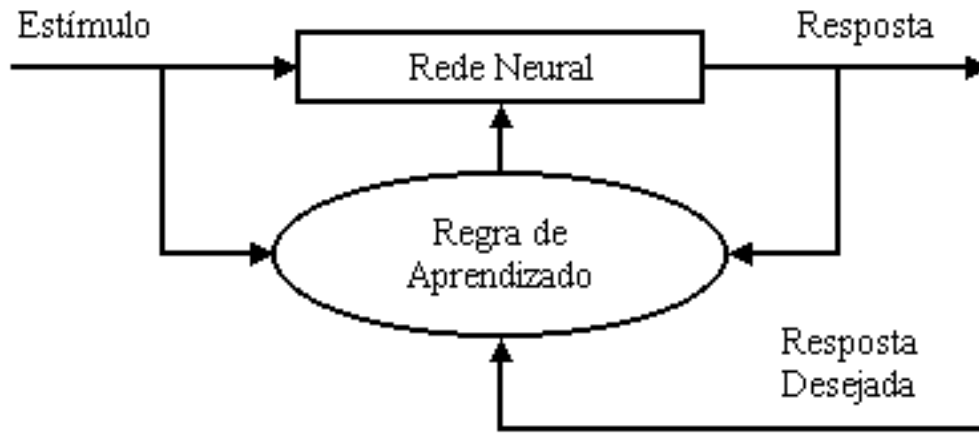


Figura 3.10: Aprendizado supervisionado.

Um exemplo de algoritmo de treinamento supervisionado é o algoritmo de Retropropagação, um dos algoritmos de treinamento mais utilizados atualmente. Este algoritmo será apresentado no item 3.8.3.

3.7.2 Aprendizado não Supervisionado

Quando se tem aprendizado não supervisionado, apresenta-se para a rede apenas as entradas, deixando de lado as saídas desejadas. Utiliza-se um esquema, tal que, para exemplos de coisas semelhantes, a rede responde de modo semelhante. A rede se encaminha iterativamente, num processo interno, próprio de categorização da informação de entrada, também conhecido como “data clustering”, onde os dados de entrada são classificados em categorias nas saídas. Aprendizado não supervisionado, também é conhecido como descobridor de regularidade, ou redes auto-organizadas, devido à propriedade básica de seu funcionamento, ver Figura 3.11.

O algoritmo de aprendizado não supervisionado pode enfatizar a cooperação entre os grupos de neurônios. Neste método, os grupos trabalhariam juntos, tentando estimular um ao outro. Se uma entrada externa ativar algum neurônio no conjunto, o conjunto seria ativado em sua totalidade. Da mesma forma, se uma entrada externa inibir algum neurônio do conjunto, isso poderá ter um efeito inibidor no conjunto inteiro.

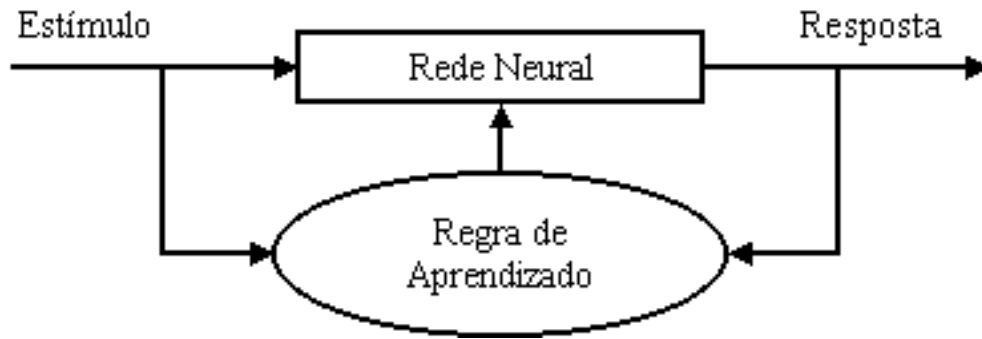


Figura 3.11: Aprendizado não supervisionado.

A competição entre os neurônios também pode formar a base do aprendizado. Treinando grupos competitivos poderiam ampliar as respostas de grupos específicos para incentivos específicos e poderiam associar esses grupos com cada outro e com uma resposta apropriada específica [64].

Esse processo de aprendizado possui uma especial capacidade de aprender a discriminar estímulos, ocorrendo em partes espacialmente diferentes. Muitos pesquisadores usam esse tipo de rede como detector de características [64].

3.8 Regras de Aprendizado

Aprender em RNA, consiste em encontrar os pesos das conexões sinápticas, de forma a resolver determinado problema. Em alguns casos, quando já se tem o conhecimento do especialista, o aprendizado é inserido diretamente na matriz de conectividade, como no caso das rede bidirecionais. Nesse caso não existe propriamente um aprendizado. Em outros casos, quando os valores das conexões são desconhecidos, utiliza-se um algoritmo para encontrá-las. A este, dá-se o nome de algoritmo de aprendizagem ou treinamento [6].

3.8.1 Regra de Hebb

Esta regra estabelece que, quando um axônio de uma célula A está suficientemente próximo para excitar uma célula B e repetidamente ou persistentemente, ocorre o seu disparo, algum processo de crescimento ou uma mudança metabólica acontece em uma

ou ambas as células, tal que a eficiência de A, como uma das células disparadoras de B, é aumentada. Neste caso, a força sináptica, conhecida como peso W , entre a célula A e B, é modificada de acordo com um grau de atividade correlacionada entre entrada e saída.

Outra lei enunciada, posteriormente por Hebb, afirma que quando o neurônio tenta excitar o outro e não consegue, existe um mecanismo que enfraquece a conexão.

Pode-se definir a regra de Hebb como:

$$\Delta w_{ij} = \eta \cdot x_i \cdot o_j \quad (3.9)$$

onde:

w_{ij} = peso da conexão entre os neurônios i e j

Δw_{ij} = ajuste dos pesos da conexão

η = taxa de aprendizado

x_i = estado de ativação do neurônio i

o_j = saída do neurônio j

3.8.2 Regra Delta

A regra delta ou Regra de Widrow Hoff, por ter sido desenvolvida por Bernard Widrow e Ted Hoff, pode ser considerada uma variação da regra de Hebb. A proposta de Widrow foi de obter o ponto de mínimo através de um processo de iteração local, utilizando cada exemplo do conjunto de treinamento por vez [50]. Esta regra baseia-se na diferença entre o valor da excitação real do neurônio i e o valor que seria desejado.

Esta regra pode ser expressa matematicamente por:

$$\Delta w_{ij} = \eta(d_i - x_i)o_j \quad (3.10)$$

onde:

w_{ij} = peso da conexão entre os neurônios i e j

Δw_{ij} = ajuste dos pesos da conexão

η = taxa de aprendizado

d_i = estado de ativação desejado do neurônio i

x_i = estado de ativação real do neurônio i

o_j = saída do neurônio j

3.8.3 Regra de Retropropagação (“*Backpropagation*”)

Retropropagação (BP) pode se considerada como a generalização da Regra Delta para redes diretas com mais de duas camadas [6]. Nesse caso, existe pelo menos uma camada interna. A Regra BP, ou o Algoritmo BP, começou a ser largamente utilizado quando, em 1986, Rumelhart et al. [82] desenvolveu a forma atualmente aplicada.

O algoritmo BP consiste basicamente em apresentarmos a entrada da rede um exemplo, que propagado por toda a rede, até que seja apresentada uma saída. Essa saída é comparada com a saída desejada, dessa comparação resulta um erro. É calculado o gradiente desse erro, com relação aos valores sinápticos da camada de saída, que é atualizada por um passo escolhido. Agora pode-se calcular o erro da penúltima camada, e assim conseqüentemente, até chegar a camada de entrada. Daí, a origem do nome retropropagação, propagação para trás, pois o erro é atualizado de trás para frente [6].

Para se fazer esses ajustes pelo algoritmo BP, utiliza-se o seguinte: [24]

O ajuste dos pesos w_{ij} da camada de saída, do neurônio j é dado por:

$$\Delta w_{ij} = \eta \delta_j o_i \quad (3.11)$$

onde:

η = Taxa de aprendizagem

o_i = Saída do neurônio i

δ_j = erro entre a saída apresentada e a desejada do neurônio j

O erro da camada de saída, pode ser calculado por:

$$\delta_j = o_j(d_j - o_j)(1 - o_j) \quad (3.12)$$

onde:

d_j = saída desejada do neurônio j

O ajuste dos pesos w_{ki} da camada interna do neurônio i , que está ligado a um neurônio k da camada anterior, é dado por:

$$\Delta w_{ki} = \eta \delta'_i o_k \quad (3.13)$$

onde:

η = Taxa de aprendizagem

o_k = Saída do neurônio k

δ'_i = erro relativo do neurônio i

O erro em uma unidade interna é dado por:

$$\delta'_i = o_i(1 - o_i) \sum_j w_{ij} \delta_j \quad (3.14)$$

Pandya propõe o fluxograma mostrado na Figura 3.12 como a seqüência de passos necessários para a implementação do algoritmo de treinamento BP [68].

É importante salientar ainda, que o algoritmo BP pode ser usado para redes recorrentes ou com ciclos. Sempre que a função de ativação for derivável, a rede pode ser treinada, não garantindo com isso, que o contrário seja verdade.

3.9 Diferenças entre RNA e RNB ou RNN

Com o propósito de uma melhor compreensão das funções cognitivas do cérebro, os pesquisadores das Redes Neurais Biológicas (experimentalistas) tem acumulado um grande número de informações sobre as propriedades moleculares e celulares dos neurônios e de circuitos neurais do cérebro [76].

Por outro lado, os pesquisadores de modelos de RNA (modelistas) tem se esmerado na análise das propriedades decorrentes do elevado grau de paralelismo e da distribuição do processamento em elementos processadores (neurônios) simples [82]. McCulloch e Pitts mostraram que tudo que pode ser computado em uma máquina de Turing, pode ser computado com o auxílio de seu modelo formal de neurônio.

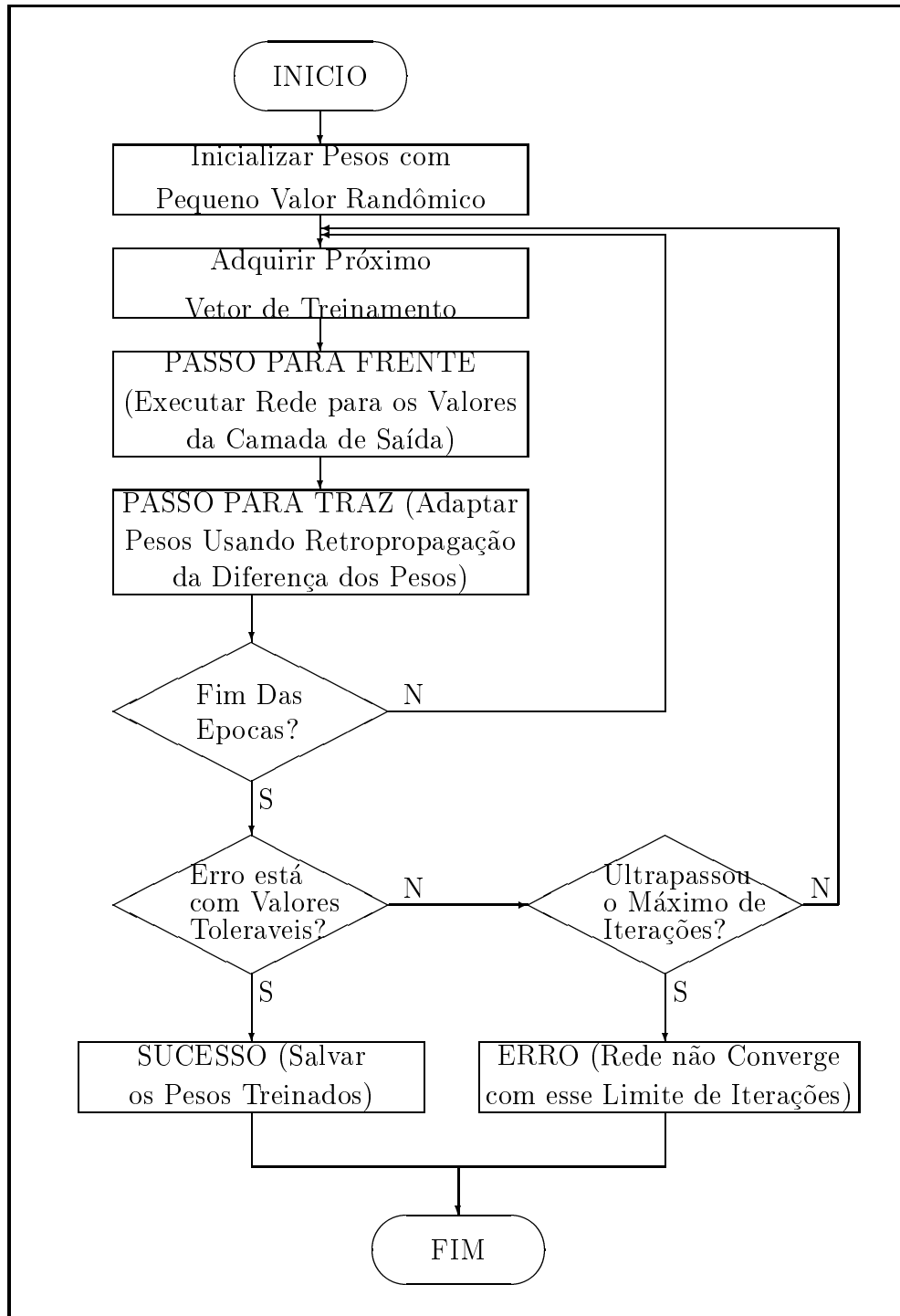


Figura 3.12: Fluxograma Back-Propagation (Pandya página 95 [68])

O cérebro é um computador extremamente sofisticado, um dos principais desafios é entender como ele realiza o processamento de informações. De modo a transpor a distância existente entre as teorias computacionais e os dados biológicos, é necessário primeiro entender como computações elementares podem ser realizadas no “hardware” do cérebro [47].

Os modelos de Redes Neurais consideram a existência de apenas uma conexão entre cada dois elementos processadores, o que equivale a um modelo biológico representando somente o que ocorre entre o axônio de um neurônio e o soma de outro. Shepherd mostrou que grande parte do processamento ocorre, não no soma ou no axônio dos neurônios e sim, em seus dendritos [86].

Os dendritos estão normalmente formando ou compreendidos em uma estrutura denominada árvore dendrítica devido as suas ramificações semelhantes a arbustos. Rall aplicou a teoria de cabos na árvore dendrítica para verificar os efeitos desta topologia [76].

O processo de divergência do sinal elétrico, que é enviado do axônio de um neurônio para os dendritos de um outro neurônio, é da ordem de apenas 10% à 20%, e não de 100% (total), como nos modelos artificiais.

O processo de convergência de diversos sinais recebidos de neurônios distintos por um único neurônio não é somente temporal, mas também é espacial, dependendo da topologia das sinapses. Muitos neurônios realizam uma soma não-linear de suas entradas, podendo haver, inclusive, processamento lógico (E,OU,NÃO) dentro da árvore dendrítica.

As árvores dendríticas não são lisas, ao contrário, apresentam pequenas estruturas da ordem de 1 a 2 micra, denominadas spines. Ao que tudo indica os “*spines*” são responsáveis por grande parte das sinapses e conseqüentemente, pelo processamento de informações. Hoje em dia, acredita-se que os “*spines*” são responsáveis pelas memórias de longo-termo (LTM) e pela plasticidade das eficiências sinápticas [76].

Os estímulos externos regulam a quantidade de neurotransmissores nas sinapses, possibilitando transformar eventos externos em informações neurais. Diferentes neurotransmissores agrupados em diferentes combinações são regulados em diferentes instantes

de tempo em diferentes sinapses.

Todo modelo de Rede Neural baseia-se na premissa de que o neurônio é a unidade básica de processamento da informação. Considera a sinapse como sendo a verdadeira unidade básica de organização dos circuitos neurais biológicos.

Um dos princípios gerais da biologia é o de que o comportamento de um organismo depende de níveis hierárquicos de organização [87]. Podemos começar pela própria sinapse onde mecanismos moleculares realizam a comunicação. As sinapses formam pequenos circuitos envolvendo algumas formas de interação entre as sinapses, denominados micro-circuitos. Estes, normalmente, encontram-se nos dendritos e seu comportamento é determinado pela distribuição na árvore dendrítica. O próprio neurônio representa um outro nível de organização, e as interações entre neurônios de propriedades similares ou distintas formam circuitos locais. Estes interligam-se em circuitos interregionais, envolvendo múltiplas regiões em diferentes partes do cérebro.

O futuro da neuro-computação poderá ser altamente beneficiado pelas pesquisas biológicas. Estruturas encontradas nos sistemas biológicos podem, e devem, inspirar novas arquiteturas para modelos de Redes Neurais [85]. Do mesmo modo que o desenvolvimento de Redes Neurais podem ser beneficiadas pelos estudos biológicos, alguns modelos ou aplicações podem ilustrar e auxiliar a explicar diversas características dos sistemas biológicos. Atualmente, a anatomia e fisiologia das redes existentes em regiões específicas do cérebro são ainda pouco conhecidas [52].

Capítulo 4

Sistemas Especialistas

4.1 Introdução

Ao implementar SE, tenta-se imitar o processo humano de raciocínio [46]. Tentando-se assim, criar sistemas computacionais que possam realizar tarefas antes só realizadas por humanos, tendo um desempenho similar aos mesmos. Uma das dificuldades iniciais no desenvolvimento de SE, vem logo nos primeiros passos, quando deve-se obter as informações do especialista. O principal problema nesse momento, é a comunicação entre o projetista e o especialista (do domínio da aplicação). Pois ambos falam sua própria linguagem técnica (utilizando jargões) que não são compreensíveis para quem não é técnico desse assunto.

Uma grande vantagem dos SE, ao contrário dos sistemas tradicionais, é possuir algumas facilidade, que aumentam sua flexibilidade e eficiência. Em SE pode-se construir regras, tomar decisão lógica sob impressisão, ou na falta de informação. Nos sistemas tradicionais as pesquisas e as respostas são baseadas num conhecimento anteriormente codificados. Com isso, qualquer novo conhecimento que apareça, exige que se reescreva o código para esse novo conhecimento. Já os SE podem recuperar novos fatos e regras e usá-los sem modificar a estratégia de busca [28].

Pode-se dizer que um SE nunca ou dificilmente (sendo otimista, considerando grandes avanços tecnológicos no futuro), atingirá a capacidade cognitiva do especialista humano, porém na sua ausência é uma ferramenta válida de resolução de problemas.

4.2 Histórico

Os SE realmente começaram a aparecer e ser explorados na década de setenta, com uma revolução ocorrida nessa época na área computacional. Os pesquisadores de IA, queriam desenvolver computadores que pudessem “pensar, raciocinar”, ou ainda ter um comportamento inteligente (similar ao que os humanos conseguem fazer) [14].

Um breve passeio por eventos importantes da história dos SE, para situar em datas os principais fatos ocorridos, ampliados de Giarratano e Riley [36]:

- 1943 - Regras de produção; Modelo de Neurônio - McCulloch and Pitts.
- 1954 - Algoritmos para controle de execução de regras - Markov.
- 1956 - Reunião do Dartmouth College; Logic Theorist; Pesquisa heurística; O aparecimento do termo IA.
- 1957 - Criação do Perceptron por Rosenblatt; GPS (General Problem Solver) Solução de Problemas Genéricos, iniciados por Newell, Shaw e Simon.
- 1958 - Demonstração da função *eval* (origem do LISP) como equivalente a Máquina de Turing - Steve Russell [16].
- 1960 - LISP linguagem de IA - McCarthy, Russell.
- 1962 - Princípios de neurodinâmica em percepção - Rosenblatt.
- 1965 - Lógica nebulosa para raciocínio de objetos nebulosos - Zadeh; Início dos trabalhos no DENDRAL - Feigenbaum, Buchanan et.al.; Teorema da Resolução - Robinson.
- 1968 - Redes Semânticas; Modelo de Memórias Associativas - Quillian.
- 1969 - MACSYMA, SE matemático - Martin e Moses.
- 1970 - Início dos trabalhos com PROLOG - Colmerauer, Roussel, et.al.
- 1971 - HEARSAY I, reconhecimento de fala.

- 1973 - MYCIN, SE para diagnóstico médico - Shortliffe, et.al.; EMYCIN, primeira shell - Van Melle, Shortliffe e Buchanan.
- 1975 - Representação de conhecimento - Minsky.
- 1976 - AM (Matemática Artificial), descobrimento de conceitos matemáticos - Lenat; Início dos trabalhos com PROSPECTOR, SE para exploração mineral - Duda, Hart, et.al.
- 1977 - OPS, Shell de SE (Forgy), usado no XCON/R1.
- 1978 - Início dos trabalhos com XCON/R1 (McDermott, DEC) para configurar o sistema computacional DEC ("*Digital Electronic Corporation*"); Meta-DENDRAL, meta regras e regras de indução - Buchanan.
- 1980 - Simbolismo; LMI fundado para fabricar máquinas LISP.
- 1982 - INTERNIST-I, SE para diagnóstico médico criado por Miller, Pople e Myers; SMP ("*math expert system*"), SE matemático; Redes Neurais de Hopfield; Projeto Japonês de computadores de quinta geração para desenvolver computadores inteligentes.
- 1983 - KEE, ferramenta para SE - IntelliCorp.
- 1985 - CLIPS, ferramenta para SE - NASA.

Os principais SE na história, começam com o DENDRAL, considerado o primeiro SE (1965), que ficou restrito ao meio acadêmico. O mais conhecido dos SE foi o MYCIN (1976), que apesar de toda a sua popularidade, também ficou restrito aos meios acadêmicos. Os SE saíram das universidades em 1982, com o desenvolvimento do XCON, que ganhou o interesse da indústria. Atualmente o setor manufatureiro e empresarial, cada vez mais estão utilizando os recursos de SE.

Mais recentemente, na Universidade Federal de Santa Catarina (UFSC), começou a haver interesse em SE, tendo sido desenvolvidos estudos e diversos trabalhos que foram defendidos em diversos departamentos. Alguns desses trabalhos serão citados a seguir.

4.3 Alguns SE desenvolvidos na Universidade Federal de Santa Catarina (UFSC)

Não se pretende neste estudo fazer um histórico completo, mas mostrar um pouco do que está sendo feito nessa universidade. Tomando por base os departamentos de Ciência da Computação, Engenharia Elétrica e Engenharia de Produção e Sistemas, para não alongá-lo muito. Porém sabe-se que outros departamentos vêm realizando trabalhos nessa área como, por exemplo, a Engenharia Civil.

Inicialmente, serão citados alguns trabalhos do Programa de Pós-Graduação em Ciência da Computação (PPGCC), ao qual este trabalho está vinculado.

- *Anita Maria da Rocha Fernandes*: defendeu sua dissertação de Mestrado em fevereiro de 1996, com o título de “Sistema Especialista Difuso Aplicado ao Processo de Análise Química Qualitativa de Amostras de Minerais” [19].

O objetivo principal desse SE é propor um sistema inteligente, baseado em lógica difusa, para o tratamento de incertezas relacionadas ao processo de análise química qualitativa de amostras de minerais. Esse processo envolve a análise de características tais como, dureza, brilho, densidade, etc., que são de natureza imprecisa.

- *Angelita Maria De Ré*: defendeu sua dissertação de Mestrado em março de 1995, com o título de “Sistemas Conexionistas Adaptativos Aplicados a Problemas de Controle de Tráfego Urbano” [79].

Neste trabalho é proposta uma abordagem através de RNA para ampliar o escopo de controle, possibilitando a atualização do conjunto de regras da máquina de inferência do sistema de controle. Por meio dessa abordagem, melhorias são efetivadas nas máquinas de inferência de sistemas inteligentes.

- *Cristiane Koehler*: defendeu sua dissertação de Mestrado em novembro de 1998, com o título de “Uma Abordagem Probabilística para Sistemas Especialistas” [48].

A presente pesquisa realizou um estudo sobre o raciocínio probabilístico em SE,

tendo como domínio de aplicação à área médica. Foi desenvolvida uma aplicação para auxiliar na avaliação do estado nutricional em crianças de 0 a 2 anos de idade com base nos sinais e sintomas e dados antropométricos.

- *Maria Aparecida Fernandes Almeida*: defendeu sua dissertação de Mestrado em setembro de 1999, com o título de “Aprender Atividade Inteligente: e se esta Inteligência for Parcialmente Artificial” [1].

Este trabalho analisa as perspectivas da IA no processo de Ensino-Aprendizagem com o computador. São observados alguns impactos da aplicação do computador no processo de ensino perante a sociedade contemporânea.

- Cabe citar aqui, os livros escritos por professores desse departamento, relacionados à área em questão:

Inteligência Artificial escrito por Renato A. Rabuske, publicado em 1995 [74].

Inteligência Artificial no Limiar do Século XXI, escrito por Jorge M. Barreto, em 1997 [7].

Programa de Pós-Graduação em Engenharia Elétrica no Grupo de Pesquisa em Engenharia Biomédica (GPBE). Aqui os estudos começaram, com o artigo escrito por Lima e Barreto, intitulado *Inteligência Artificial* em 1987 [53]. A partir desse artigo, vieram outros estudos e diversos trabalhos defendidos, pode-se citar os trabalhos de:

- *Júlio César Nievola*: defendeu sua dissertação de Mestrado em abril de 1988, com o título de “Sistema Especialista para Auxílio ao Diagnóstico Médico de Icterícia” [65].

Esse SE serve para auxiliar o especialista da área de gastroenterologia no diagnóstico em pacientes ictericos, sendo utilizadas uma Base de Conhecimentos, onde estão contidos os principais pontos em que o especialista se baseia para poder determinar as prováveis causas da icterícia. Este conhecimento é então utilizado por um mecanismo de inferência, que procura estabelecer os graus de possibilidade de cada uma das doenças causadoras de icterícia.

- *Aurora Trinidad Ramirez Pozo*: defendeu sua dissertação de Mestrado em dezembro de 1991, com o título de “Sistema Inteligente para Treinamento do Controle da Infecção Hospitalar” [73].

Este sistema para computador, utiliza técnicas de Inteligência Artificial e um banco de dados para o processamento de índices de controle da infecção hospitalar. Com isso, o sistema detecta automaticamente o nível de conhecimento do estudante, guiando-o a partir daí, pelos diferentes tópicos com maior ou menor aprofundamento. São abordados dois pontos fundamentais: treinamento do pessoal (sistema de ensino) que permitirá uma ação preventiva e ação de controle mais ágil facilitada pela análise de informações através do banco de dados.

- *Renato Garcia Ojeda*: defendeu sua tese de Doutorado em julho de 1992, com o título de “Técnicas de Inteligência Artificial Aplicadas ao Apoio à Decisão Médica na Especialidade de Anestesiologia” [67].

Esse sistema destina-se ao auxílio no ensino de anestesiologia, assim como para apoio aos médicos. Foi utilizado para isso um SE integrado, onde cada conhecimento é representado pela técnica que melhor se adapte às características do domínio de aplicação. Essa estrutura integrada, com processamento semântico-numérico, reúne redes neurais, regras de produção, modelos clínicos e bases de dados, atuando sinergisticamente para dar apoio à decisão médica na proposta e na avaliação de procedimentos anestésicos.

- *Lourdes Mattos Brasil*: defendeu sua dissertação de Mestrado em maio de 1994, com o título de “Aquisição de conhecimento Aplicada ao Diagnóstico de Epilepsia” [12].

Essa dissertação realizou uma análise das técnicas de elicitação de conhecimento aplicada ao diagnóstico de epilepsia. Para tal, foi escolhida Epilepsia do Lobo Temporal e utilizado técnicas de elicitação e representação do conhecimento.

- *Fernando Passold*: defendeu sua dissertação de Mestrado em março de 1995, com o título de “Sistema Especialista Híbrido em Anestesiologia para Pacientes

Críticos / Problemáticos” [69].

Este trabalho apresenta uma expansão introduzida no sistema especialista híbrido em anestesiologia, PROVANES [67]. Este sistema busca auxiliar o processo de tomada de decisão dos anestesiológicos, para procedimentos cirúrgicos em cavidade abdominal, vasos e membros. Sua principal base de conhecimentos é formada por dez redes neurais que imitam o planejamento anestésico realizado pelo especialista da área.

- *Gisele Faffe Pellegrini*: defendeu sua dissertação de Mestrado em Setembro de 1995, com o título de “Proposta de uma Metodologia de Avaliação de Sistema Especialista na Área Médica” [70].

Este trabalho propõe uma metodologia para avaliação de SE utilizando a metodologia de desenvolvimento por protótipo. Esta metodologia consiste em um sistema que é continuamente re-implementado. Através de um protótipo operacional, os constantes retornos obtidos pelos usuários irão garantir um SE de melhor qualidade e maior aceitabilidade, mais voltado para as reais necessidades dos usuários.

- *Kathya Silvia Collazos Linares*: defendeu sua dissertação de Mestrado em fevereiro de 1997, com o título de “Sistema Especialista Nebuloso para Diagnóstico Médico” [54].

Neste trabalho é proposto um método para tratar a imprecisão nos dados clínicos utilizando a lógica nebulosa, para tentar acabar a diferença existente entre os sistemas de computador convencionais (lógicos - certo ou errado) e a realidade (com toda sua faixa aceitável entre o certo e o errado).

- *Nancy Akemi Sigaki*: defendeu sua dissertação de Mestrado em fevereiro de 1997, com o título de “Estudo da Influência de Bases de Casos em Redes IAC (“*Interactive Activation and Competition*”) na Implementação de Sistema Especialista para Diagnóstico Médico” [88].

Neste trabalho foi estudado o problema do aprendizado (baseado em casos) em redes conexionistas IAC (“*Interactive Activation and Competition*”). Foi desen-

volvido um “*software*” para a implementação da rede IAC, contendo as versões de Rumelhart e McClelland [82], de Grossberg [82] e a versão proposta por de Azevedo [20]. Para validar o sistema e exemplificar o estudo da abordagem, foi construído um Sistema de Apoio à Decisão na Área Médica, utilizando um problema de Diagnóstico Diferencial em Reumatologia. A seguir foi estudado o comportamento deste tipo de rede quando considerados casos reais clínicos diagnosticados por um especialista da área em comparação com uma rede contendo somente as relações entre doenças e sintomas. Discute-se, finalmente, o desempenho comparativo das duas redes: aquela contendo somente as relações doenças-sintomas e aquela contendo também os casos clínicos reais.

- *Rocío Liliana Callupe Pérez*: defendeu sua dissertação de Mestrado em fevereiro de 1997, com o título de “Sistema Especialista de Apoio à Decisão Médica com Metodologia de Aprendizagem Simbólica” [72].

Este trabalho tem por objetivo apresentar os métodos de aprendizagem simbólica em sistemas especialistas na área médica. Tais métodos foram analisados com um critério de classificação por aquisição de conhecimento dos sistemas. Estes métodos refletem o tipo de inferência desempenhado pela máquina para obter o conhecimento desejado.

- *Jurandir Paz de Oliveira*: defendeu sua dissertação de Mestrado em fevereiro de 1997, com o título de “Redes Neurais Artificiais para representação de conhecimento Médico em Anestesiologia” [25].

Este trabalho apresenta uma análise das topologias e algoritmos de treinamento de RNAs para representar conhecimento médico especializado em anestesiologia. Estas RNA foram previamente implementadas na construção de um sistema especialista híbrido em anestesiologia - PROVANES, com o objetivo de propor e avaliar planos de anestesia. Foram desenvolvidos e implementados algoritmos para análise da sensibilidade das RNA com objetivo de extração e obtenção de regras de conhecimento armazenadas nos pesos das redes.

- *Marlise Vidal Montello*: defendeu sua dissertação de Mestrado em fevereiro de

1999, com o título de “Sistema Especialista para Predições e Complicações Cardiovasculares Integrado a um Sistema de Controle de Pacientes Portadores de Diabetes Mellitus” [62].

Este trabalho tem por objetivo a implementação de um sistema de gerenciamento de dados, o SCDM (Sistema de Controle de Pacientes Portadores de Diabetes Mellitus), capaz de gerir todo o processo de atendimento a pacientes diabéticos realizado no Ambulatório de Endocrinologia do Hospital Universitário da Universidade Federal de Santa Catarina de forma a torná-lo mais ágil, eficiente e confiável. Desenvolveu-se um módulo especialista que, através de técnicas de Inteligência Artificial, é capaz de identificar se um determinado paciente apresenta ou não algum tipo de predisposição para desenvolver complicações cardiovasculares baseado na Análise dos Fatores de Risco.

- *Júlio César Nievola*: defendeu sua tese de Doutorado em julho de 1995, com o título de “Sistema Inteligente para auxílio no Ensino em Traumatologia Crânio-Encefálica” [66].

Este trabalho concentra-se no estudo e especificação de um protótipo básico de um ICAI (Instrução Inteligente Auxiliada por Computador) para Emergência Hospitalar, a fim de possibilitar o treinamento de residentes para atender pacientes em ambiente hospitalar, a ser utilizado em Hospitais Universitários ou Hospitais Escola.

- *Sílvia Modesto Nassar*: defendeu sua tese de Doutorado em dezembro de 1995, com o título de “Sistema Estatístico Inteligente para Apoio a Pesquisas Médicas” [63].

O desenvolvimento de sistemas especialistas estatísticos permite o uso adequado de procedimentos estatísticos no processo de análise de dados de pesquisas quantitativas, evitando conclusões errôneas devido à falta de conhecimentos estatísticos do pesquisador. Esta tarefa foi implementada utilizando o paradigma quadro negro (“*blackboard*”). Este paradigma é uma técnica de IA distribuída, onde é proposto um modelo de fontes de conhecimento independentes, assíncronas e

anônimas que, na forma original do modelo, se comunicam entre si através de uma base de dados multinível, chamada de quadro negro. Foi implementado um protótipo de um sistema estatístico inteligente para apoio à pesquisas médicas no Hospital Universitário da UFSC.

- *João da Silva Dias*: defendeu sua tese de Doutorado em fevereiro de 1999, com o título de “Sensibilidade Paramétrica como Guia para o Treinamento Híbrido de Redes Neurais” [2].

A técnica híbrida de treinamento de RNA através do algoritmo genético e do algoritmo de retropropagação apresenta limitações de convergência é restrita a pequenas redes. Este trabalho discute as razões destas deficiências e, baseado nos resultados obtidos, propõe a análise de sensibilidade como forma de guiar o algoritmo genético para uma condição do espaço de busca mais promissora.

- *Lourdes Mattos Brasil*: defendeu sua tese de Doutorado em fevereiro de 1999, com o título de “Proposta de Arquitetura para Sistema Especialista Híbrido e a Correspondente Metodologia de Aquisição do conhecimento” [13].

Este trabalho tem como meta propor uma metodologia para o desenvolvimento de um SE usando uma arquitetura híbrida. Este SE tem, como característica principal, a capacidade de aprender a extrair conhecimento a partir de uma base de conhecimento inicial e de um conjunto de exemplos. Desta forma, espera-se contribuir para a solução de um dos problemas de IA, que consiste na extração de conhecimento do especialista de domínio e que, nos sistemas simbólicos, é conhecido como uma das etapas da Aquisição de Conhecimento.

- *Roberto Célio Limão de Oliveira*: defendeu sua tese de Doutorado em março de 1999, com o título de “Modelo de Redes Neurais Artificiais com Dinâmica Interna em Sistemas de Controle” [3].

De modo a facilitar o uso de modelos de Redes Neurais Artificiais com dinâmica interna, desenvolveu-se um modelo que apresenta dinâmica interna linear. A linearidade da dinâmica interna facilita bastante o estudo da estabilidade da rede,

uma vez que é necessário apenas calcular os autovalores da matriz de dinâmica dos neurônios. Para implementar o aprendizado neste modelo, com dinâmica linear, foi desenvolvido um algoritmo de aprendizado, do tipo “*backpropagation*”, de fácil implementação.

Para mostrar a grande importância que os SE vem tendo dentro da UFSC, podemos apresentar também o trabalho desenvolvido no programa de pós-graduação em medicina interna do Hospital Universitário (HU), onde o médico Li Shih Min defendeu sua tese, denominada *Sistema Baseado em Conhecimento para Detecção e Classificação de Crises Epilepticas* [58]

Muito próximo a este departamento, também deve-se destacar, o departamento de Engenharia de Produção e Sistemas (EPS), que tem produzido alguns trabalhos nessa área. Aqui citam-se alguns exemplos mais recentes.

- *Ana Paula Soares Fernandes*: defendeu sua dissertação de Mestrado em janeiro de 1997, com o título de “Sistema Especialista Difuso de Apoio ao Aprendizado do Traumatismo Dento-Alveolar Utilizando Recursos Multimídia” [28].

É necessária experiência prática para determinar corretamente o tipo de tratamento indicado nos traumatismos dento-alveolar. Existem vários fatores relevantes para indicar o tratamento, tais como: tipo de dentição, tempo decorrido entre o trauma e o atendimento, entre outros. Alguns fatores imprecisos, quando corretamente considerados, podem indicar ou não o sucesso do tratamento. Esse SE tem o objetivo de facilitar o ensino do traumatismo dento-alveolar em aulas práticas e teóricas, basendo-se em lógica difusa para o tratamento de incertezas relacionadas aos traumatismos.

- *Marcelo Hugo*: defendeu sua dissertação de Mestrado em setembro de 1995, com o título de “Uma Interface de Reconhecimento de Voz para o Sistema de Gerenciamento de Central de Informação de Fretes” [44].

Este trabalho procura demonstrar a viabilidade e potencialidade dos sistemas comandados por voz, construindo um protótipo de “*software*” capaz de responder

em tempo real a um comando falado por um usuário. Ele se utiliza das técnicas de RNA para realizar o reconhecimento das palavras faladas. A interface de voz construída para reconhecer as palavras, que operam um protótipo do Sistema Gerenciador de Central de Informação de Fretes (SGCIF) em “*Windows*”, aplica o modelo de rede neural Kohonen, alcançando uma taxa média de acerto de 84,84% no reconhecimento.

- *David Lemos*: defendeu sua dissertação de Mestrado em maio de 1996, com o título de “A Utilização de Sistemas Especialistas para o Diagnóstico do Uso do Solo e Seus Limites de Ocupação” [51].

Através dos SE podemos fazer uma análise lógica do problema apresentado, à nível do especialista humano, utilizando a inferência de conhecimentos, via regras de produção. Além da abordagem da utilização da tecnologia dos SE nesta área de domínio, apresenta-se um SE para o diagnóstico do uso do solo e seus limites de ocupação, denominado UsoSolo, a nível de um protótipo de demonstração [94].

- *Raimundo Celeste Ghizoni Teive*: defendeu sua dissertação de Mestrado em março de 1997, com o título de “Planejamento da Expansão da Transmissão de Sistemas de Energia Elétrica Utilizando Sistemas Especialistas” [92].

Neste trabalho é proposto que a experiência do especialista seja utilizada na solução do problema de definição das alternativas de expansão, e conseqüentemente, na determinação do plano de expansão de mínimo custo. Este conhecimento pode ser representado como regras de produção em um SE, e o mecanismo de inferências deste sistema aplica estas regras de maneira a imitar o raciocínio do especialista humano (planejador).

- *Ronaldo Fernandes Ramos*: defendeu sua dissertação de Mestrado em fevereiro de 1995, com o título de “Sistemas Especialistas - uma Abordagem Baseada em Objetos com Prototipagem de um Seleccionador de Processo de Soldagem” [77].

O objetivo deste trabalho é desenvolver um protótipo de SE em seleção de processos de soldagem. Foram feitas verificação da viabilidade de construção de um SE

em soldagem, bem como da adequabilidade do uso do paradigma da orientação a objetos e uma estrutura de aplicação no desenvolvimento de SE. Os resultados foram satisfatórios em todos os seus aspectos.

- *Malcon Anderson Tafner*: defendeu sua dissertação de Mestrado em janeiro de 1996, com o título de “Reconhecimento de Palavras Faladas Isoladas Usando Redes Neurais Artificiais” [90].

O objetivo desta pesquisa foi tratar exatamente desse problema, da possibilidade de reconhecimento de palavras faladas com o uso de RNA. Entretanto, o reconhecimento apenas com o uso de redes neurais apresenta certas dificuldades, uma vez que a rede encontra obstáculos para aprender a reconhecer o sinal, visto a complexidade da voz humana. Dessa forma, procurou-se associar o processamento da rede neural com um rigoroso pré-processamento do sinal de voz, cuja finalidade foi reduzir a quantidade de sinais e aumentar a sua representatividade, acentuando os principais pontos.

- *José Leomar Todesco*: defendeu sua dissertação de Mestrado em dezembro de 1995, com o título de “Reconhecimento de Padrões Usando Redes Neurais Artificiais com uma Função de Base Radial: Uma Aplicação na Classificação de Cromossomos Humanos” [93].

Neste trabalho apresenta-se o uso de uma RNA com uma função de base radial para classificar os cromossomos humanos em 24 classes. No estudo explorou-se diferentes combinações de características extraídas de cromossomos bem como diferentes topologias e parâmetros para a rede.

- *Roberto Tadeu Raittz*: defendeu sua dissertação de Mestrado em dezembro de 1997, com o título de “Free Associative Neurons - FAN uma Abordagem para Reconhecimento de Padrões” [75].

Aresenta-se aqui um novo método de aprendizagem: Free Associative Neurons (FAN). FAN é constituído por neurônios independentes com capacidade autônoma de aprendizado. O poder de aprendizado de FAN está baseado na granularidade

na representação da informação. O método é embasado na lógica difusa e nas noções de redes neurais. Apresentam-se os princípios que regem FAN e também são discutidos os resultados obtidos pelo método e sua comparação com outros algoritmos conhecidos de redes neuronais. Na conclusão são apresentadas propostas para trabalhos futuros.

4.4 Definição de Sistemas Especialistas

SE são sistemas implementados em computador, para resolver problemas específicos, onde tenta-se dotar o SE de um comportamento semelhante ao que o especialista (humano) teria para resolver os seus problemas. Para realizar tais tarefas, é necessário antes de mais nada, obter o conhecimento do especialista, que deve ser armazenado convenientemente (de acordo com o objetivo, ou a técnica utilizada).

Os SE são conhecidos, também, como Sistemas Baseados em Conhecimento (SBC). Durante algum tempo, as abordagens *Simbólica* e *Conexionista* ficaram muito afastadas, cada qual andando por si, e nesse momento, só se falava em SE (abordagem Simbólica) e Redes Neurais, esquecendo-se, em alguns momentos, que os SE também podem ser implementados por RNA. Sendo assim, quando se fala SE apenas, geralmente, entende-se como SE Simbólicos.

Os SE e SBC diferem em alguns aspectos. O desenvolvimento dos SE inicia com a extração do conhecimento de um especialista humano, por um engenheiro de conhecimento, que codificará em regras e fatos esse conhecimento do especialista. De forma similar, o desenvolvimento de SBC tem seu conhecimento vindo de outras fontes, incorporando tópicos de assuntos que não requerem aptidão ou educação especial, como aqueles encontrados em livros ou manuais. Desta forma, pode-se considerar os SE como um caso particular de SBC, e os SBC, a generalização dos SE [13].

Esse trabalho trata de Sistemas Especialistas Conexionistas, ou seja, sistemas baseados em exemplos (inspirados no cérebro humano).

4.5 Áreas de Utilização de Sistema Especialista

Os SE estão sendo utilizados cada vez mais e nas mais diversas áreas de aplicação, algumas delas serão citadas a seguir conforme Pereira [71] e Fernandes [28].

- **Interpretação:** Interpretar e analisar informações tal como: Fotos de satélite (para sensoriamento remoto);
- **Predição ou Diagnóstico:** Inferir as conseqüências de uma dada situação inicial. Diagnóstico médico, por exemplo;
- **Síntese:** Configurar sistemas ou objetos a partir de um conjunto de especificações;
- **Planejamento:** Estabelecer uma seqüência de ações que atinjam determinada meta. Planejamento de trajetória de robôs, por exemplo;
- **Monitoração:** Acompanha a evolução de determinado sistema, como na supervisão de processos industriais;
- **Correção de Falhas:** Propor medidas corretivas para falhas em sistemas, como por exemplo, em manutenção de aeronaves;
- **Instrução:** Propor problemas e acompanhar sua solução pelo treino. Por exemplo para treinamento de operadores de processos;
- **Controle:** Impor ao sistema certo comportamento desejado, como por exemplo, controle de processos industriais.

4.6 As duas Abordagens dos Sistemas Especialistas

4.6.1 Sistemas Especialistas Simbólicos (SES)

São os SE baseados no *Simbolismo*, ou seja *Base de conhecimento*, *Motor de Inferência*, *Regras*, etc.

Dois pontos importantes a serem considerados são:

Os fatos, dados, sintomas ou características do domínio do problema (as informações relevantes para que se possa resolver o problema). Ex: Quando seu carro não funciona corretamente, você descreve para o mecânico (especialista), o que está ocorrendo (Motor está falhando, temperatura alta, etc). E as regras que o especialista usa para resolver o problema. As analogias que ele faz para chegar a uma possível resposta para o problema em questão.

Isso é cuidadosamente analisado pelo projetista do sistema (*engenheiro de conhecimento*), o qual irá inserir na base de dados do sistema todo esse conhecimento, toda essa experiência retirada do especialista. O trabalho do *engenheiro de conhecimento* é extremamente importante, uma vez que o especialista têm a capacidade de resolver problemas difíceis, explicar os resultados obtidos, aprender, reestruturar o conhecimento e determinar as suas características relevantes. Porém, geralmente o especialista não consegue explicar o seu modo de raciocínio de uma maneira analítica [19]. A maioria dos SES que nasceram nos anos 60 eram organizados ao redor dessa estrutura.

4.6.2 Sistemas Especialistas Conexionistas (SEC)

Esse são os SE baseados (ou inspirados) no cérebro humano. Aqui o foco de interesse são as RNA, com suas mais diversas topologias, como apresentado no capítulo 3.

Os SEC são especialmente indicados, quando o domínio do problema em questão não é bem definido, quando existem muitas exceções as regras, de forma que não se consiga definir com precisão as regras para a extração do conhecimento. Como os SEC, são baseados em exemplos e não em regras, pode-se resolver problemas sem o conhecimento de como tomar as decisões, bastando para isso ter exemplos de casos reais. Barreto [7] divide o domínio de atuação dos SE em dois tipos: o das coisas criadas pelo homem e as criadas pela natureza. Quando se tem os feitos pelo homem, se conhece bem o domínio e é fácil conseguir todos os detalhes do assunto. Mas nos naturais o sistema é desconhecido ou pouco conhecido e seu funcionamento pode ser conhecido através do uso de modelos e a estrutura do sistema conhecida através de exemplos. Neste segundo caso, fica claro a indicação do uso de SEC para resolver o

problema.

SEC implementados por Redes Diretas

Quando se implementa um SEC por Redes Diretas, faz-se criando a rede direta conforme explicado no capítulo 3 item 3.5.1. As Redes Diretas, precisam ser treinadas. Para tal, utiliza-se um algoritmo de treinamento como por exemplo o de Retropropagação já mencionado no capítulo 3 item 3.8.3. Para treinar a rede, também é necessário um conjunto de exemplos que deve cobrir os casos mais típicos do domínio considerado. Este conjunto de exemplos para o treinamento será utilizado pelo algoritmo de treinamento.

Após esse processo, a rede já está treinada e pronta para ser utilizada, fornecendo os resultados aos quais ela foi treinada para responder. Mas como em qualquer sistema computacional, não se pode deixar de realizar os testes, para validar o sistema.

Talves a parte mais complicada dos SEC implementados por Redes Diretas, seja a escolha ou a definição do número de neurônios e de camadas a serem utilizados pela rede. Não existe uma fórmula exata, para calcular a quantidade ideal de neurônios em cada camada da rede.

SEC implementados por Memórias Associativas Bidirecionais

Esta abordagem de SEC através de Memórias Associativas foi estudada e apresentado com muita propriedade nos trabalhos de Fernando de Azevedo [8], [9], [20], [21], [23].

Neste caso, não existe uma fase de treinamento, nem tão pouco utiliza-se algum tipo de algoritmo de treinamento, conseqüentemente não existe a fase de aprendizado. Aqui os exemplos são utilizados para fixar os valores das conexões sinápticas, representados por uma matriz sináptica simétrica.

A intensidade das conexões é um número no intervalo $[-1 ; 1]$ representando o valor da relação entre os conceitos representados pelos dois neurônios. Os valores do intervalo $[-1 ; 0]$ representam inibição do neurônio e o intervalo $[0 ; 1]$ representam excitação, sendo que o valor $[0]$ representa a indiferença ou a falta de relação entre os neurônios.

Para se obter respostas nessa implementação, dá-se a entrada excitando os neurônios

correspondentes aos dados do caso. Esta excitação pode variar de intensidade de acordo com a força que se deseja para esta situação (sempre respeitando o intervalo $[0 ; 1]$).

4.6.3 Uma Comparação Simplificada entre a Estrutura de um SES e de um SEC

Pode-se considerar que um SES possua cinco componentes básicos, como apresentado na Figura 4.1.

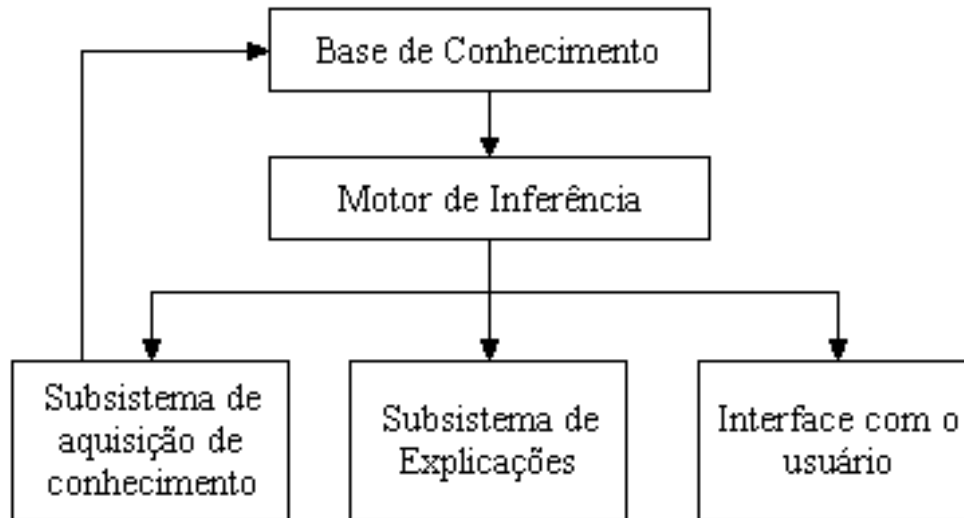


Figura 4.1: Componentes básicos de um SES.

A *Base de Conhecimento* é formada pelas regras e procedimentos que o especialista humano usa na solução de problemas. Como esta é separada do motor de inferência, é mais fácil alterar o conhecimento contido na base. Pode-se incluir, alterar ou excluir as regras e procedimentos contidos na base de conhecimento.

O *Motor de inferência* é o mecanismo que procura as respostas na base de conhecimento, buscando os fatos e as regras necessárias a serem avaliadas e compara os fatos com a informação fornecida pelo usuário. Funciona como um supervisor, que dirige a operação sobre o conhecimento contido no SE.

O *Subsistema de Aquisição de Conhecimento* é o responsável por introduzir novos conhecimentos e alterar ou eliminar conhecimentos antigos. Fazendo assim toda a movimentação das regras e fatos da base de conhecimento.

O *Subsistema de Explicações* é quem explica ao usuário a linha de raciocínio que o SE utilizou para chegar à conclusão. Isso é necessário, para mostrar ao usuário a forma de raciocínio usada para chegar às respostas, quando o usuário precisa convencer-se ou não da resposta apresentada pelo SE.

A *Interface com o Usuário* é uma parte do sistema utilizada para estabelecer um meio de comunicação entre o SE e o usuário. A interface exhibe as entradas (perguntas) e saídas (respostas) do SE, independente de quais recursos visuais são utilizados para isso.

A estrutura do SEC, pode ser apresentada com três componentes básicos, da forma como apresentado na Figura 4.2 [35].

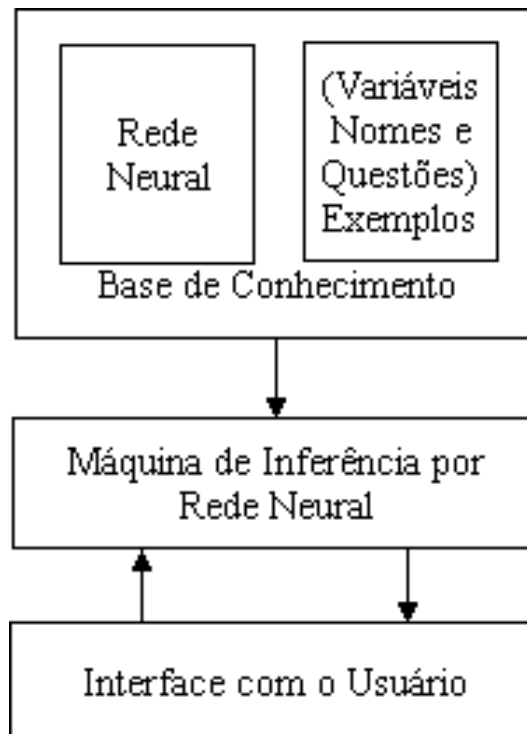


Figura 4.2: Componentes básicos de um SEC.

A *Base de Conhecimento* é composta dos exemplos usados pela rede neural para o seu treinamento (onde a rede adquire conhecimento). Aqui entra a fase de treinamento (redes diretas) ou a fase de inserção dos pesos na matriz de conexões (memórias associativas).

O *Motor de inferência* é o próprio mecanismo da rede neural, o qual recebe as entradas vindas da interface com o usuário, interage e retorna a resposta (os neurônios ativados da saída), para a mesma.

A *Interface com o Usuário* é uma parte do sistema utilizada para estabelecer um meio de comunicação entre o SE e o usuário. Pode-se considerar a mesma explicação dos SES.

Outra diferença a se considerar, é o fato das BAM poderem utilizar dois tipos de conhecimento. O conhecimento do especialista e o conhecimento de casos (exemplos) simultaneamente. As Redes Diretas não utilizam o conhecimento causal do especialista diretamente. Para o treinamento das redes diretas utiliza-se apenas os exemplos. Acredita-se com isso, que os SE implementados por BAM tenham um desempenho superior, conforme resultados apresentados por de Azevedo [20].

Capítulo 5

Metodologia Proposta

5.1 Passos para o Desenvolvimento

Para a realização deste trabalho, será montada uma base de exemplos de casos de reumatologia, que servirão para realizar o treinamento e os testes da Rede Direta, necessários para as verificações e análises sugeridas nesse trabalho.

Será selecionado um conjunto de exemplos. Este conjunto será particionado, de modo aleatório em dois subconjuntos disjuntos, um que encerra o conhecimento a ser colocado na rede (Base de Aprendizado (BA) apresentados nas Tabelas 5.1 e 5.2) e o outro para avaliar a qualidade da rede munida deste conhecimento (Base de Testes (BT) apresentados nas Tabelas 5.3 e 5.4). Para a rede direta, será criado um arquivo com uma parte da BA, o qual será submetido ao algoritmo de treinamento de Retropropagação, para o devido treinamento desta. Pretende-se começar o treinamento com cerca de 6 (seis) exemplos, para obter os resultados dos primeiros testes. Os testes serão feitos utilizando os exemplos das BT. Todos os exemplos serão utilizados, verificando quantos acertos e quantos erros ocorrerão. Após a obtenção dos resultados, repete-se novamente o treinamento, mas agora com a inserção de 3 (três) novos exemplos no arquivo de exemplos para o treinamento, para que se possa gerar gráficos mostrando a evolução da rede com um incremento pequeno de casos no treinamento. Este processo sera repetido, até esgotar toda a base de exemplos BA. A cada teste realizado, armazena-se os resultados em um arquivo, para posterior análise da evolução dos resultados.

Na rede de memórias associativas, não existe a fase de treinamento, como é o caso da rede direta. No entanto, é necessário inserir os exemplos na matriz de conexões. Apesar dessa diferença, o procedimento permanece o mesmo usado para a rede direta. Os primeiros testes serão realizados, inserindo-se 6 (seis) exemplos da base BA na BAM (acrescentando 3 (três) a cada teste) e testando com os exemplos da base BT. Da mesma forma, os resultados obtidos serão armazenados em um arquivo, para que possam ser analisados.

Será verificado o nível de acerto, em cada etapa dos testes, montando tabelas e gráficos que demonstrem a curva de aprendizado da rede.

5.2 Base de Exemplos

Os exemplos a serem utilizados para as redes, tanto para treinamento como para os testes, foram obtidos de casos clínicos reais da área de reumatologia, vindos da *Clínica EMONT GODINE* da Universidade Católica de Lovain na Bélgica, através da professora *Lycia Epprecht* [22]. Foram omitidos os nomes dos pacientes a fim de preservar suas identidades.

Os exemplos são compostos por: 3 (três) patologias a diagnosticar, 50 casos clínicos (pacientes) e 17 (dezessete) exames/sintomas. Os dados para os diagnósticos estão divididos em 2 (dois) grupos, os *Sintomas* e os *Exames Laboratoriais*.

Estes exemplos foram divididos como citado anteriormente, em 2 (dois) grupos de forma aleatória (por sorteio). Onde foram retirados 32 (trinta e dois) casos para o treinamento e 18 (dezoito) casos para os testes. Conforme discriminados abaixo:

Patologias

1. Artrite de Gota (AG)
2. Artrite Reumática (AR)
3. Ankylosing Spondylitis (AS)

Sintomas

1. Artrite (Art)
2. Rigidez Matinal (RM)
3. Bursite (Bur)
4. Tophi (Tof)
5. Sinovite (Sin)
6. Artralgia (Atg)
7. Nódulos Reumatóides (NR)
8. HLA-B27
9. Deformação nas Juntas (DJ)
10. Dor Lombar (DL)
11. rigidez na Coluna (RC)
12. Deformação na Coluna (DC)
13. Mobilidade (Mob)
14. Dor ao Toque no Sacroilíaco (DTS)

Exames Clínicos

1. Inflamação Laboratorial (IL)
2. Evidências Radiológicas (ER)
3. Tomografia Computadorizada / Sintigraphy Evidence (TCSE)

Tabela 5.1: Exemplos para o treinamento (início)

Caso	DL	RC	DC	Mob	DTS	IL	ER	TCSE
1	sim	sim	sim	limitado	sim	ausente	moderado	moderado
2	não	não	não	normal	não	ausente	moderado	não
4	não	não	não	limitado	não	ausente	moderado	não
5	não	não	não	normal	não	moderado	importante	moderado
6	não	não	não	limitado	não	muito importante	muito importante	importante
7	não	não	não	limitado	não	moderado	muito importante	não
8	sim	não	sim	normal	sim	moderado	importante	importante
11	sim	sim	sim	limitado	sim	moderado	importante	não
13	não	não	não	limitado	não	muito importante	ausente	não
14	não	não	não	limitado	não	importante	ausente	leve
16	oui	não	não	normal	não	leve	muito importante	não
18	não	não	não	normal	não	leve	importante	não
20	sim	não	não	normal	não	moderado	ausente	não
21	não	não	não	limitado	não	importante	ausente	moderado
24	sim	sim	não	normal	sim	ausente	moderado	leve
25	sim	não	sim	normal	não	ausente	importante	não
26	não	não	não	normal	não	importante	ausente	não
27	sim	sim	não	normal	não	leve	importante	importante
29	não	não	não	limitado	não	moderado	moderado	moderado
30	não	não	não	limitado	não	importante	importante	moderado
32	não	não	não	normal	não	importante	importante	não
36	sim	sim	não	normal	sim	ausente	importante	importante
37	não	não	não	normal	não	importante	moderado	não
38	não	não	não	limitado	não	moderado	ausente	não
40	não	não	não	normal	não	importante	importante	não
41	não	não	não	limitado	não	importante	moderado	moderado
42	sim	não	não	normal	não	moderado	importante	não
44	não	não	não	limitado	não	ausente	ausente	não
46	sim	sim	não	limitado	sim	leve	importante	não
47	não	não	não	limitado	não	importante	moderado	não
49	sim	sim	não	limitado	não	importante	moderado	não
50	sim	não	não	normal	não	ausente	moderado	moderado

5.3 Considerações Importantes

A Metodologia Proposta teve algumas alterações em sua forma inicialmente prevista, visto que problemas ocorreram quando da implementação da rede BAM. Constatou-se, no momento de sua implementação, que problemas de instabilidade estavam ocorrendo, conforme será apresentado no capítulo 8, impossibilitando os testes comparativos inicialmente propostos, uma vez que não se completou com sucesso os testes da referida rede. Com isso não foi possível extrair resultados dessa rede.

Em função do ocorrido, foram feitos apenas os testes com a rede Direta, extraíndo dessa, resultados significativos conforme será apresentado nos capítulos 7 e 9, Resultados Obtidos e Conclusões, dessa dissertação.

Aproveitou-se, também, devido aos problemas encontrados, para apresentar os motivos através dos quais não se conseguiu a estabilidade da rede BAM, mostrando os

Tabela 5.2: Exemplos para o treinamento (continuação)

Caso	Diagnóstico	ART	RM	Bur	Tof	Sin	ATG	NR	HLA-B27	DJ
1	Ankylosing Spondylitis	não	não	não	não	não	não	não	neg	não
2	Artrite de gota	sim	não	não	não	não	sim	não	não	não
4	Artrite de gota	não	não	não	sim	não	sim	não	não	não
5	Artrite Reumática	não	não	não	não	não	sim	não	não	não
6	Artrite de gota	sim	não	não	sim	sim	sim	não	não	não
7	Artrite Reumática	sim	não	não	não	sim	sim	não	não	não
8	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
11	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
13	Artrite de gota	sim	não	não	não	sim	sim	não	não	não
14	Artrite Reumática	sim	sim	não	não	sim	sim	não	neg	não
16	Ankylosing Spondylitis	não	não	não	não	não	não	não	neg	não
18	Artrite Reumática	sim	sim	não	não	sim	sim	não	não	não
20	Ankylosing Spondylitis	sim	não	não	não	sim	sim	não	pos	não
21	Artrite Reumática	sim	sim	não	não	sim	sim	não	não	sim
24	Ankylosing Spondylitis	não	sim	não	não	não	não	não	pos	não
25	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
26	Artrite Reumática	não	não	sim	não	não	sim	sim	não	não
27	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
29	Artrite de gota	sim	não	não	não	não	sim	não	não	não
30	Artrite de gota	sim	não	sim	sim	sim	sim	não	não	não
32	Artrite de gota	sim	sim	sim	sim	não	sim	não	não	não
36	Ankylosing Spondylitis	não	sim	não	não	não	sim	não	pos	não
37	Artrite de gota	sim	não	sim	sim	sim	sim	não	não	não
38	Artrite de gota	não	não	sim	sim	não	não	não	não	não
40	Artrite Reumática	sim	não	não	não	sim	sim	não	neg	não
41	Artrite Reumática	sim	não	não	não	sim	sim	não	não	não
42	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
44	Artrite Reumática	sim	não	não	não	sim	sim	não	neg	não
46	Ankylosing Spondylitis	não	sim	não	não	não	não	não	neg	não
47	Artrite Reumática	sim	sim	não	não	sim	sim	não	não	não
49	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
50	Artrite de gota	não	não	não	não	não	sim	não	não	não

testes realizados, a forma como esses foram feitos e onde se encontrou o problema em questão.

Tabela 5.3: Exemplos para os testes (início)

Caso	DL	RC	DC	Mob	DTS	IL	ER	TCSE
3	não	não	não	limitado	não	moderado	moderado	não
9	sim	sim	não	limitado	sim	importante	muito importante	importante
10	não	não	não	limitado	não	muito importante	ausente	moderado
12	não	não	não	limitado	não	leve	moderado	não
15	não	não	não	normal	não	não	não	não
17	não	não	não	limitado	não	moderado	moderado	não
19	sim	sim	sim	limitado	não	moderado	muito importante	moderado
22	não	não	não	normal	não	leve	importante	não
23	sim	não	não	normal	não	ausente	moderado	moderado
28	sim	sim	não	normal	não	ausente	ausente	não
31	sim	sim	sim	normal	não	ausente	importante	não
33	sim	não	não	normal	sim	moderado	importante	não
34	não	não	não	limitado	não	leve	importante	não
35	não	não	não	limitado	não	leve	moderado	não
39	não	não	não	limitado	não	moderado	importante	não
43	não	não	não	normal	não	importante	importante	não
45	sim	não	não	normal	não	leve	importante	não
48	não	não	não	normal	não	moderado	importante	não

Tabela 5.4: Exemplos para os testes (continuação)

Caso	Diagnóstico	ART	RM	Bur	Tof	Sin	ATG	NR	HLA-B27	DJ
3	Artrite de gota	sim	sim	sim	sim	sim	sim	não	não	não
9	Rheumatoid Arthristis	sim	sim	não	não	sim	sim	não	não	não
10	Rheumatoid Arthristis	sim	sim	não	não	sim	sim	sim	não	não
12	Rheumatoid Arthristis	sim	não	não	não	sim	sim	não	não	não
15	não	não	não	não	não	não	não	não	não	não
17	Artrite de gota	sim	não	não	sim	não	sim	não	não	não
19	Ankylosing Spondylitis	sim	não	não	não	não	sim	não	pos	sim
22	Artrite de gota	sim	não	não	sim	sim	sim	não	não	não
23	Ankylosing Spondylitis	não	não	não	não	não	sim	não	pos	não
28	Rheumatoid Arthristis	não	sim	não	não	não	sim	não	não	não
31	Ankylosing Spondylitis	não	não	não	não	não	não	não	pos	não
33	Ankylosing Spondylitis	não	sim	não	não	não	não	não	pos	não
34	Rheumatoid Arthristis	sim	sim	não	não	sim	sim	não	não	não
35	Rheumatoid Arthristis	sim	sim	não	não	sim	sim	não	não	não
39	Rheumatoid Arthristis	sim	não	não	não	sim	sim	não	não	não
43	Rheumatoid Arthristis	não	sim	não	não	não	sim	não	não	não
45	Artrite de gota	sim	não	não	não	sim	sim	não	não	não
48	Rheumatoid Arthristis	não	sim	não	não	não	sim	não	não	não

Capítulo 6

Escolha de Ferramentas para Implementar o Sistema Especialista

Neste capítulo, será feita uma breve apresentação das principais ferramentas encontradas e disponibilizadas, para que fosse possível realizar a análise que segue, através da qual será escolhida uma ferramenta adequada para a execução dessa dissertação.

6.1 PDP - “Parallel Distributed Processing”

O PDP [56] é uma ferramenta para DOS, logo não dispõe de uma interface agradável e interativa com o usuário, tornando assim seu uso um pouco mais complicado aos iniciantes, principalmente àqueles que não possuem conhecimentos de lógica de programação e do funcionamento de arquivos do DOS, uma vez que para se criar novas bases é necessário alterar diretamente seus arquivos de dados, ou criar novos com a mesma estrutura.

Um certo domínio dos comandos e instruções é indispensável para viabilizar o uso desta ferramenta, tendo em vista que seu mecanismo não é intuitivo, requerendo, antes de mais nada, um período de treinamento, para só então pensar em utilizar seus recursos.

O PDP permite o uso de diversas topologias de redes neurais, tais como IAC (*“Interactive Activation and Competition”*), BP (*“Backpropagation”*), que são o objetivo

principal desse estudo.

A Figura 6.1 mostra a tela inicial do PDP com o aplicativo de Rede IAC. Pode ser constatado uma interface bastante simples e com poucos recursos, nada auto-explicativa. O formato de apresentação é texto, deixando o aplicativo pouco convidativo para usuários acostumados a ambientes gráficos.

```
file name (return to abort): mai.txt
disp/ exam/ get/ save/ set/ clear cycle do input log quit reset run
test

0 Jets -10 0 Art -10 0 Phil -10 0 _Art -10 0 _Phil -10 cycle 1
0 Sharks -10 0 Al -10 0 Ike -10 0 _Al -10 0 _Ike -10
0 Sam -10 0 Nick -10 0 _Sam -10 0 _Nick -10
0 in20s -10 0 Clyde -10 0 Don -10 0 _Clyde -10 0 _Don -10
0 in30s -10 0 Mike -10 0 Ned -10 0 _Mike -10 0 _Ned -10
0 in40s -10 0 Jim -10 0 Karl -10 0 _Jim -10 0 _Karl -10
0 Greg -10 ** Ken 34 0 _Greg -10 0 _Ken -10
0 JH -10 0 John -10 0 Earl -10 0 _John -10 0 _Earl -10
0 HS -10 0 Doug -10 0 Rick -10 0 _Doug -10 0 _Rick -10
0 College-10 0 Lance -10 0 Ol -10 0 _Lance -10 0 _Ol -10
0 George -10 0 Neal -10 0 _George-10 0 _Neal -10
0 Single -10 0 Pete -10 0 Dave -10 0 _Pete -10 0 _Dave -10
0 Married-10 0 Fred -10 0 _Fred -10
0 Divorce-10 0 Gene -10 0 _Gene -10
0 Ralph -10 0 _Ralph -10
0 Pusher -10
0 Burglar-10
0 Bookie -10
```

Figura 6.1: Tela básica do PDP com a rede IAC

6.2 MatLab

O MatLab [55] é uma ferramenta extremamente poderosa, pois é praticamente uma linguagem de programação. Esta ferramenta foi desenvolvida para trabalhar com matrizes, permitindo que o tratamento de matrizes seja extremamente simplificado, facilitando em muito todo o tipo de problema que utilize matrizes. Como redes neurais são baseadas em operações com matrizes, esta é uma ótima opção.

O ponto forte do MatLab é a grande flexibilidade que se tem ao trabalhar com ele, pois com uma linguagem de programação pode-se fazer tudo que se desejar. Porém,

mais uma vez, é fundamental para usar esta o conhecimento de lógica de programação, pois sem este conhecimento fica inviável seu uso. Outra vantagem é que acompanhando o MatLab há uma série de funções prontas, entre elas, algumas para redes neurais, como por exemplo, o algoritmo de treinamento Retropropagação. As fontes de todas estas funções estão disponíveis nos “*ToolBox*” do MatLab, permitindo, com isso, que se conheça os arquivos fonte e altere-os, se for necessário. Estas funções podem ser testadas, pois ele também possui um menu que permite o acesso a demonstração das mesmas.

Sua interface não é um ponto forte da ferramenta, mesmo sendo desenvolvido para Windows, como pode ser observado na Figura 6.2. Apesar disso, pode-se trabalhar a interface do aplicativo a ser gerado, programando-a conforme a necessidade. O visual e a ergonomia do produto que se está desenvolvendo depende muito da habilidade e conhecimento do programador.

A necessidade de programar o MatLab poderá ser considerado por alguns um ponto fraco, porém para outros, esta é uma de suas vantagens. Ao mesmo tempo que a programação o deixa extremamente flexível, permitindo que um bom programador o utilize com muita eficiência, também pode inviabilizar seu uso para usuários que não dominem a lógica de programação.

6.3 NeuralWorks

Por ser uma ferramenta gráfica e um aplicativo específico para RNA, o NeuralWorks (NW) [37] possui uma interface mais fácil de ser compreendida e utilizada, pois com o clicar do botão do mouse, acessa-se os seus menus com as diversas opções disponíveis, conforme mostra a Figura 6.3. Ele permite que se crie novas redes apenas escolhendo as opções desejadas e preenchendo alguns campos solicitados com as devidas informações. Pode-se treinar a rede diversas vezes apenas mudando seus parâmetros, tais como: número de neurônios, de camadas, função de transferência, regra de treinamento, entre outros. Tudo isso apenas selecionando opções dos menus e preenchendo com as

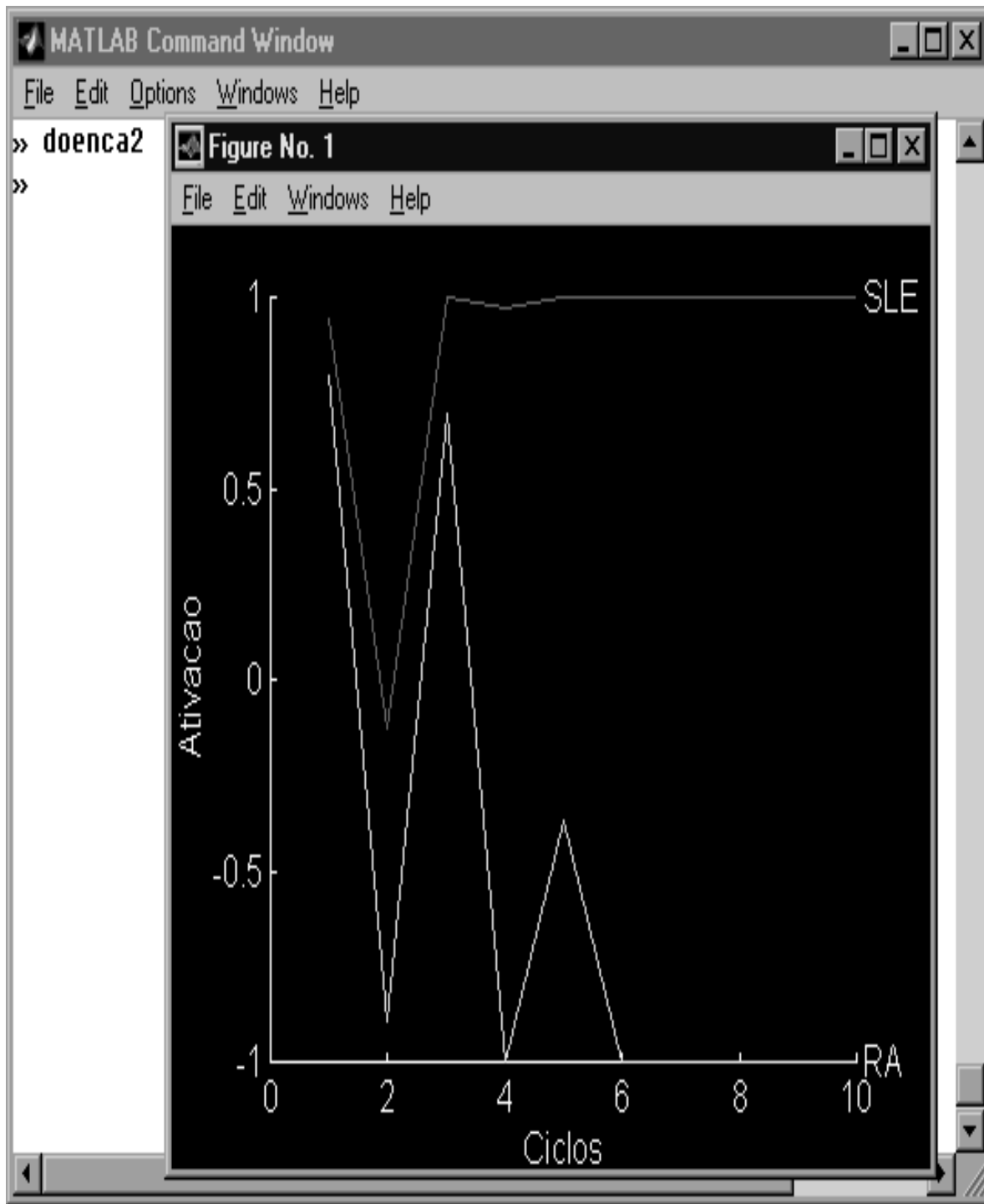


Figura 6.2: Exemplo de uma aplicação em MatLab

informações correspondentes quando necessário. Dessa forma pode-se fazer testes e escolher a melhor opção, que será a rede que obteve melhor índice de acertos.

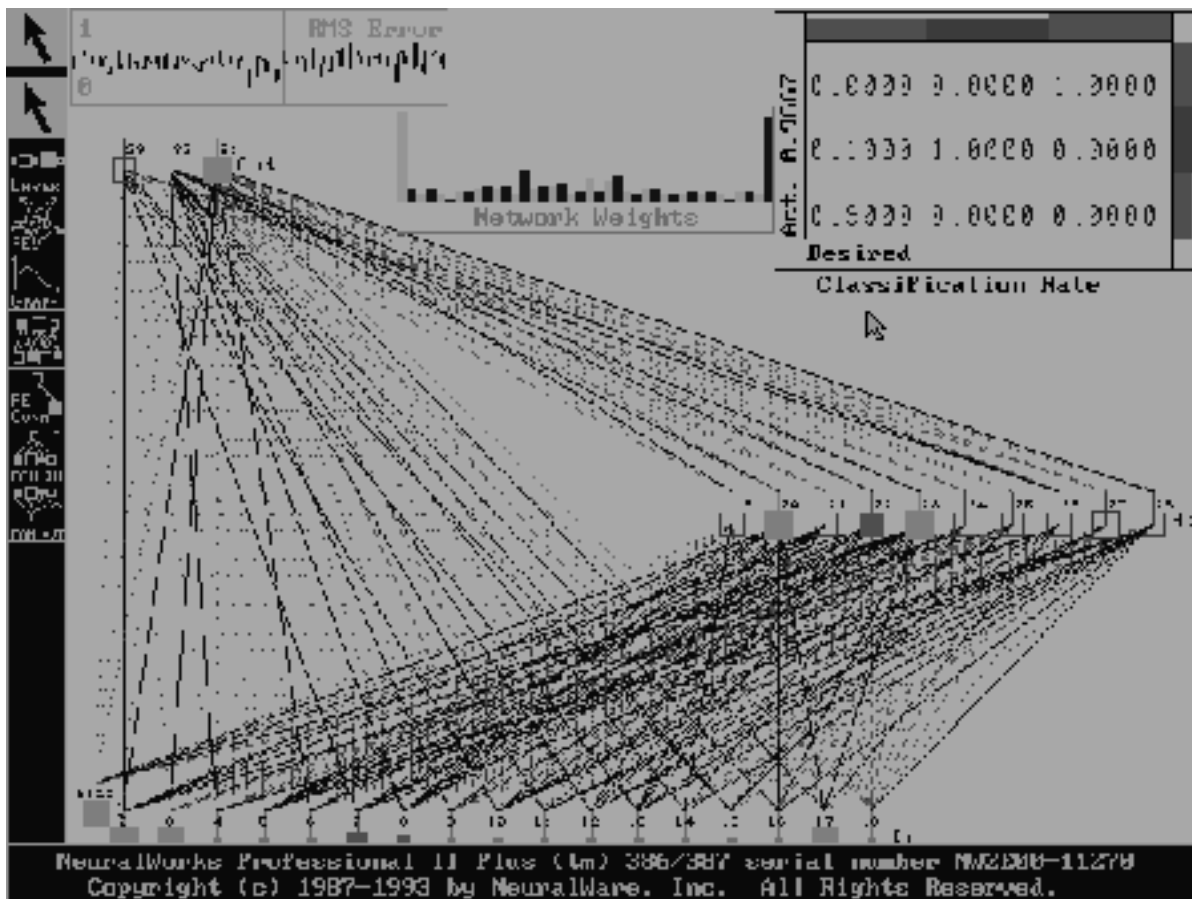


Figura 6.3: Exemplo de uma tela do Neural Works

Um ponto fraco dessa ferramenta é a montagem das bases de dados de teste e treinamento. Esta deve ser montada no formato exigido para que o NW reconheça e efetue o treinamento (treinamento supervisionado). Após o treinamento é gerado um arquivo com a estrutura da rede treinada. A partir dessa fase, pode-se começar a utilizar a rede.

Outro ponto fraco é o fato de sua interface ser muito pobre, apesar da sua apresentação na forma gráfica. Isso se explica, em parte, pelo fato de ser um aplicativo de 1993, época em que ainda não se dispunha de todos os recursos visuais conhecidos atualmente.

6.4 BrainMaker

O BrainMaker [89] é um “*software*” para implementar RNA, que trabalha no padrão DOS, porém com uma aparência razoável para este padrão. Este aplicativo possui dois módulos separados, o NetMaker e o BrainMaker.

O NetMaker é o aplicativo usado para criar os arquivos e manipular os dados que irão formar a RNA. A Figura 6.4 apresenta a tela de abertura do NetMaker, com suas diversas opções.

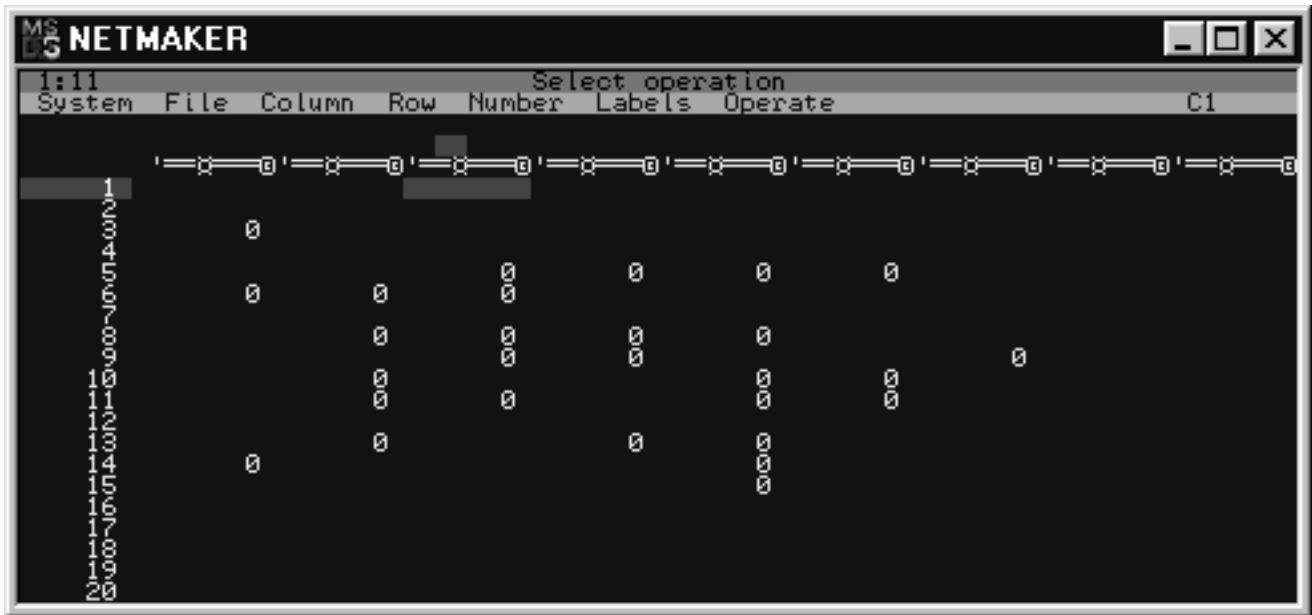


Figura 6.4: Tela de abertura do NetMaker

O Módulo BrainMaker possui outra tela de abertura, a qual solicita inicialmente alguns parâmetros derivados do módulo anterior, como por exemplo o arquivo de definições, que já deve estar criado. A figura 6.5 apresenta a tela de abertura do BrainMaker.

Por essa ser uma ferramenta para DOS, sua interface deixa a desejar se comparado a aplicativos gráficos. Seu ponto positivo é o fato de ter o aplicativo para manipular os

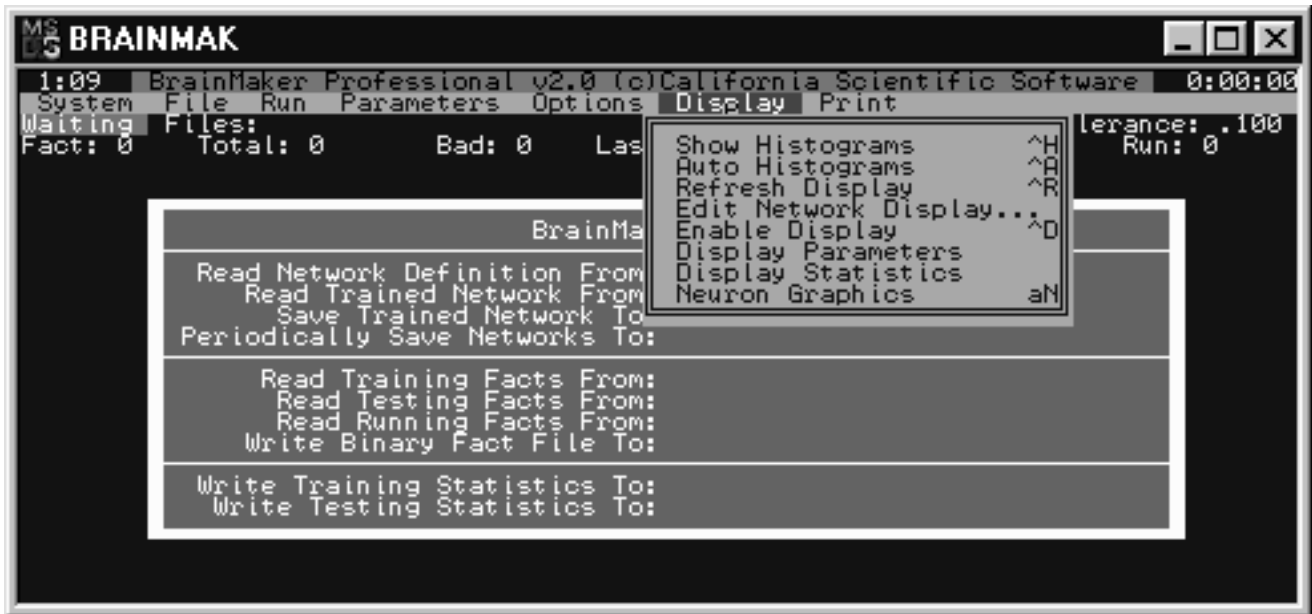


Figura 6.5: Tela de abertura do BrainMaker

dados dos arquivos de treinamento. Um ponto fraco é a pouca flexibilidade em relação as opções para gerar a rede, assim como para as funções de transferência e o algoritmo de treinamento, que só pode ser o de Retropropagação.

6.5 Neural Model

Esta é uma ferramenta mais moderna e totalmente para Windows, com isso, possui uma interface bastante atrativa e fácil de manipular, uma vez que é idêntica a maioria dos aplicativos para Windows. A Figura 6.6 apresenta uma tela do Neural Model.

O Neural Model parece ser uma ferramenta bastante poderosa, porém, ela foi pouco explorada até o momento. De qualquer forma vale a pena investir um pouco de tempo para verificar seu potencial, pois pode vir a ser uma boa opção.

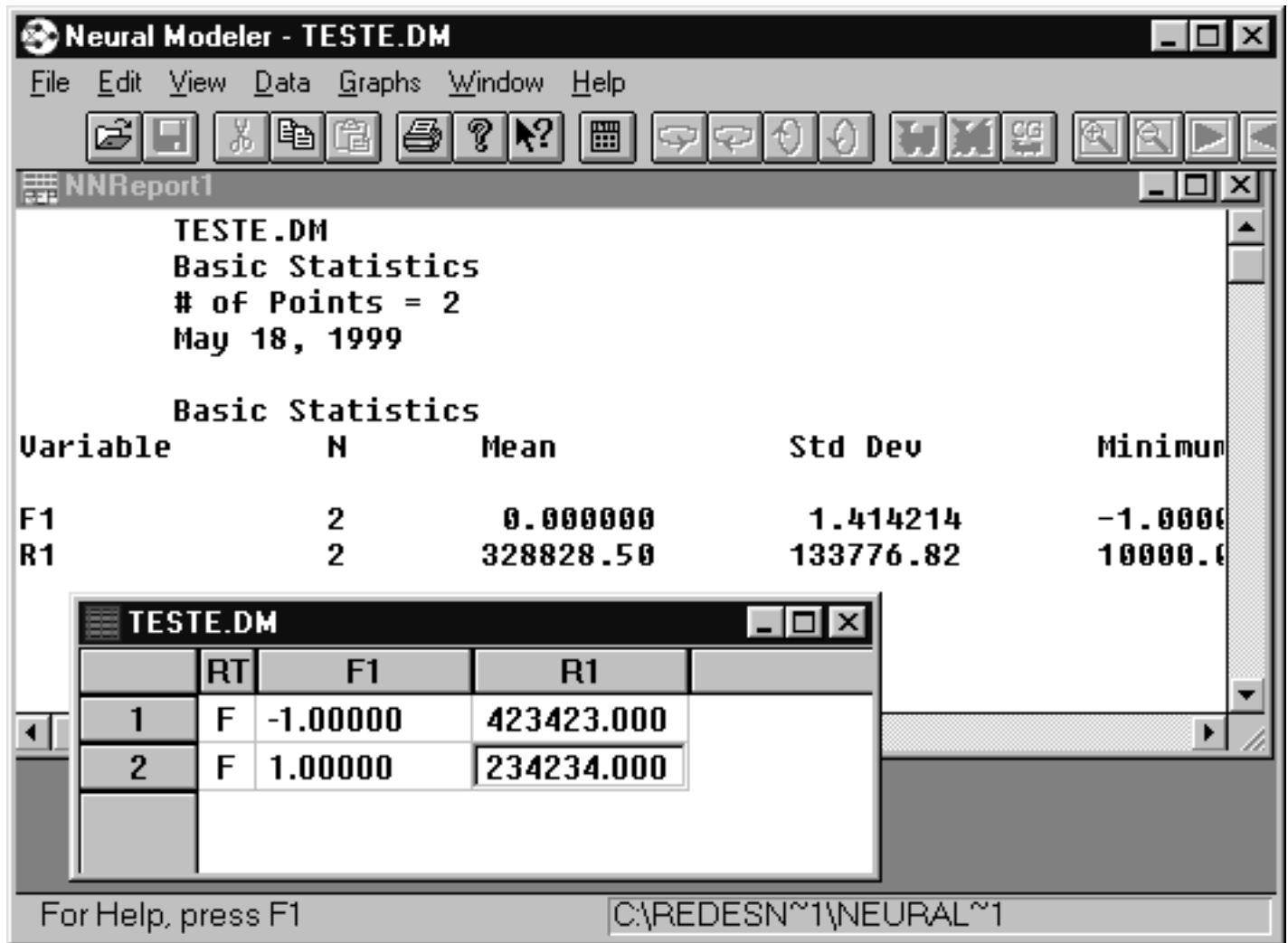


Figura 6.6: Tela do Neural Model

6.6 Ferramenta Escolhida

Basedo no exposto até aqui, optou-se por duas ferramentas. O MatLab para implementar a rede BAM, fazendo uso de seus recursos de programação e de sua flexibilidade, para que se tenha total liberdade na forma de implementação. Também optou-se por ele devido as demais ferramentas não permitirem a criação da BAM na forma proposta nesse trabalho. A única exceção é o PDP, que permite o uso de rede IAC, mas foi deixado de lado por ser extremamente pobre de recursos.

A outra ferramenta escolhida foi o NeuralWorks para implementar a rede Direta. Essa opção foi feita, por ser essa uma ferramenta bastante prática e fácil de ser utilizada. Por isso, tornou a implementação da rede extremamente rápida, que no momento era de singular importância. Como o NW é totalmente parametrizado, pode-se dar todas as características desejadas à rede gerada.

Capítulo 7

Implementação da Rede Direta

A Rede Direta foi implementada na ferramenta “*NeuralWorks Professional II Plus*” pois é uma ferramenta fácil e rápida para se implementar RNA desse tipo.

Utilizou-se como parâmetros para implementar tal rede o seguinte:

1. Três camadas foram utilizadas na rede:
 - A de Entrada, com 17 (desessete) neurônios, correspondendo a cada um dos sintomas utilizados na base de dados;
 - A de Saída, com 3 (três) neurônios, correspondendo as doenças a serem diagnosticadas;
 - Uma camada interna, com 5 (cinco) neurônios, escolhidos eurísticamente e após alguns testes com diversos valores.
2. O algoritmo de treinamento utilizado foi o de retropropagação, com uma base de 32 (trinta e dois) casos clínicos para treinar a rede.
3. Utilizou-se também a função de transferência Tangente Hiperbólica, para manter-se os valores sempre entre 1 e -1.
4. Foram utilizados 20.000 (vinte mil) passos para o treinamento. Para todos os testes utilizou-se esse parâmetro, porém a rede costumava estabilizar o treinamento com pouco menos de 10.000 (dez mil) passos.

Na Figura 7.1 apresenta-se os principais parâmetros utilizados para configurar a rede em questão.



Figura 7.1: Janela do NeuralWorks com parâmetros

Apresenta-se na Figura 7.2, a forma da rede após ter sido treinada com 24 (vinte e quatro) casos.

Os testes realizados foram seguindo a seqüência prevista na metodologia, porém pequenas alterações foram feitas, como no número de exemplos de treinamento utilizados para começar os testes. O primeiro teste foi executado com o treinamento da rede numa base de apenas 2 (dois) casos, por ter-se percebido um bom índice de acertos no treinamento com os 6 (seis) casos previstos para o primeiro teste na metodologia proposta. Também alterou-se o incremento no número de casos dos testes a cada

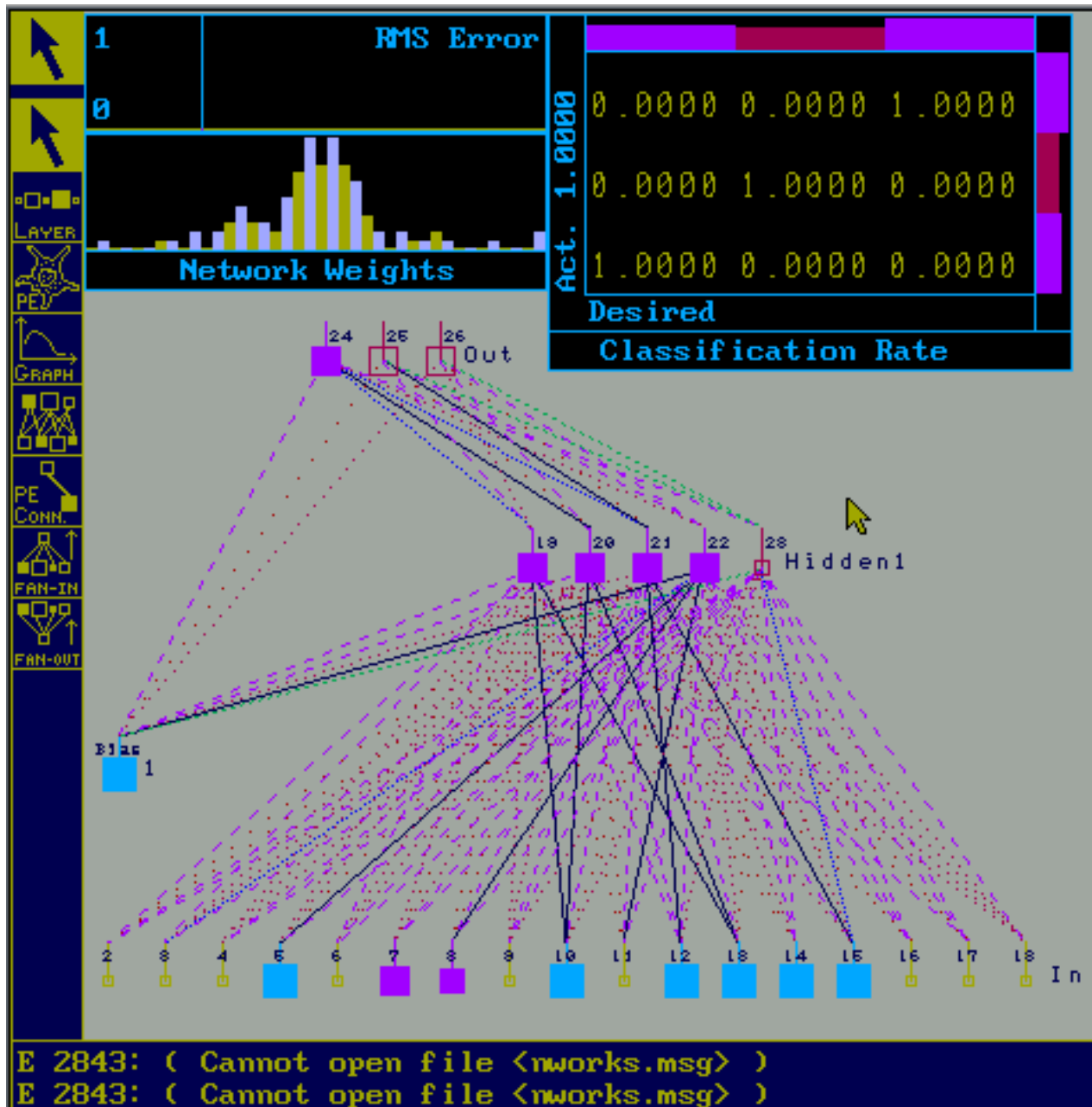


Figura 7.2: Rede treinada com 24 casos

novo teste, utilizando-se aqui um incremento de 2 (dois) casos, ao invés dos 3 (três) propostos.

Desta forma, o primeiro teste foi executado com a rede treinada com 2 (dois) casos, o segundo com 4 (quatro), o terceiro com 6 (seis) e assim por diante, até atingirmos o total de casos da base de treinamento, isto é 32 (trinta e dois) casos.

A cada treinamento efetuado, fez-se o teste com os 18 (dezoito) casos separados na base de testes. Casos estes, não utilizados na base de treinamento.

Tabelas foram geradas com os resultados obtidos em cada um desses testes, para que fosse possível montar o gráfico apresentado na Figura 7.3, relacionando o percentual de acertos da rede com o número de casos utilizados em seu treinamento.

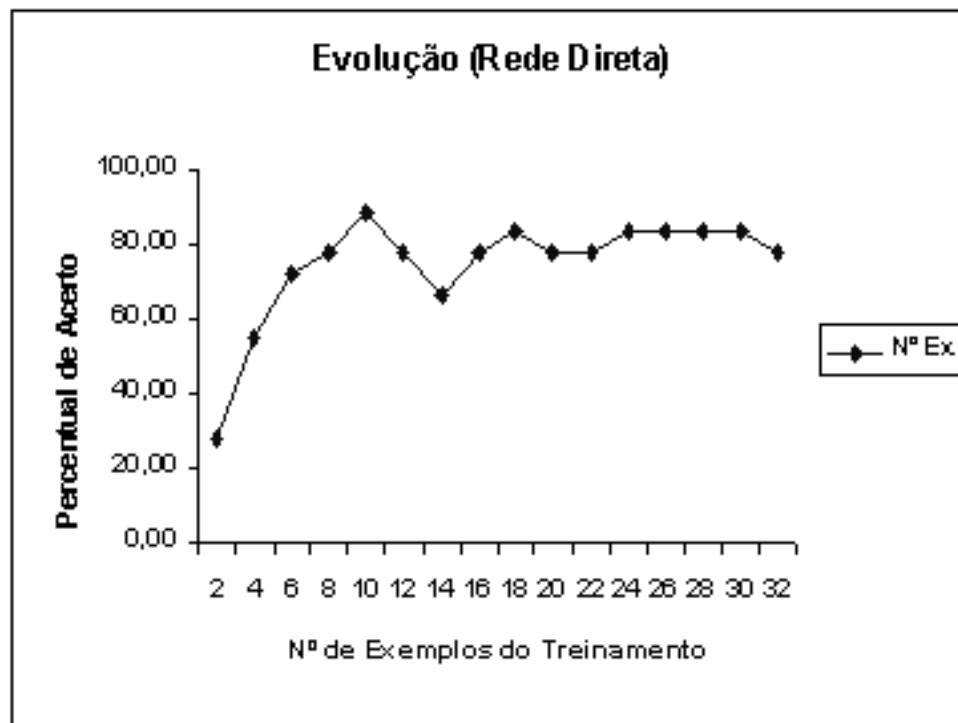


Figura 7.3: Gráfico de evolução da rede Direta com 5 neurônios

Outros testes foram realizados, um deles alterando o número de neurônios da camada interna para 3 (três) ao invés dos 5 (cinco) do teste anterior. O gráfico apresen-

tado na Figura 7.4, apresenta o resultado para esse novo teste. Novamente relacionando o percentual de acertos da rede com o número de casos utilizados em seu treinamento.

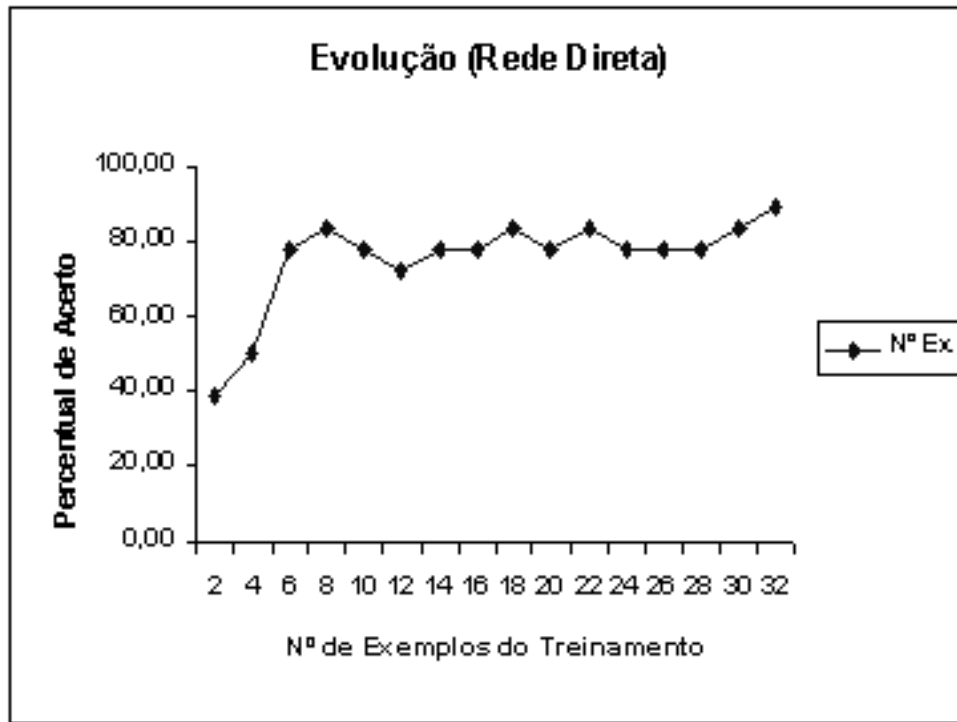


Figura 7.4: Gráfico de evolução da rede Direta com 3 neurônios

Com estes teste, observou-se que a rede direta conseguiu um bom desempenho, mesmo com poucos exemplos no treinamento. Os índices de acerto estabilizaram-se em torno dos 80 (oitenta) por cento, já com poucos exemplos, como pode ser visto nos gráficos.

Capítulo 8

Crítica de Utilização da BAM

8.1 Introdução

Em função da maioria das ferramentas de RNA verificadas não apresentarem o código ou a forma (explícita) de como são implementadas, tornando-se “caixas-pretas” para quem as está utilizando, optou-se por implementar a RNA. A ferramenta escolhida para tal tarefa, foi o MATLAB, por apresentar as condições necessárias e uma grande facilidade na manipulação de matrizes, essencial para a implementação de RNA. Além disso, também possui uma série de funções prontas, que auxiliam na geração dos códigos em tal tarefa.

8.2 Implementação da BAM

Segundo Cohen/Grossberg, uma BAM alcançará a estabilidade se sua matriz de conexões for simétrica e sua diagonal principal nula [49, 17].

Baseado na proposta de Cohen/Grossberg, assim como nos trabalhos De Azevedo [20] e, posteriormente, Sigaki [88], resolveu-se implementar a RNA BAM utilizando, para isso, o *MatLab*. Verificou-se, então, que a rede não apresentava estabilidade, ficando sempre numa variação cíclica entre os valores máximos e mínimos das excitações dos neurônios.

Em virtude desse fato, partiu-se para uma verificação mais detalhada a respeito do motivo que levara a RNA a tal instabilidade. Após diversos testes e tentativas de

alterações nos pesos da rede, como por exemplo, utilizando apenas valores variando entre 0 e 1 ou entre -1 e 1 ou ainda 0.95 e -0.95 para tentar que não saturasse nos extremos, confirmou-se que realmente não estava aí o problema. O problema não era com o nível de excitação dos neurônios, mas supunha-se que estivesse relacionado aos pesos da matriz de conexões.

Verificou-se detalhadamente esta matriz, para constatar se não havia passado despercebidamente alguma inconsistência em seus valores ou na montagem da mesma, uma vez que, esta é montada automaticamente, partindo da BA. Como essa deve ser uma matriz simétrica, poderia estar aí o motivo da inconsistência. Mais uma tentativa frustrada ocorreu, pois os dados estavam corretamente apresentados na matriz, sem nenhum problema com simetria ou qualquer outro problema correlato.

Tentou-se novamente, verificar se a implementação estava realmente seguindo a forma e metodologia corretas. Após ter sido verificado com detalhes a maneira como foi implementada a BAM, constatou-se não haver erros, pois apresentava-se um vetor (u) com os dados de entrada, excitando ou não os neurônios correspondentes. Submetia-se ao produto desse vetor com a matriz de pesos (W) passando o seu resultado por uma função de ativação tangente hiperbólica (f), obtendo-se como saída um vetor (y) num tempo posterior ao da entrada ($k + 1$).

$$y(k + 1) = f(W.u(k)) \quad (8.1)$$

Se a rede é recursiva, seu modelo será de uma equação de estado discreta, ou seja;

$$x(k + 1) = F(x(k), u(k)) \quad (8.2)$$

Para entender melhor os problemas de estabilidade, recordou-se os conceitos de estabilidade com seus vários conceitos correlatos e que são válidos para sistemas lineares e não lineares. Esse estudo iniciou com a revisão da teoria de estabilidade de Liapunov.

8.2.1 Teoria de Estabilidade

O estudo de estabilidade de sistemas não-lineares, por considerações de energia, achou uma grande aplicação no campo de RNA. Em particular, o segundo método de Lia-

punov parece ser uma certa aproximação geral. A idéia básica se preocupa com o estudo de uma função V , análoga, mas não precisamente igual, para a energia armazenada do sistema, sendo examinada como uma função de tempo. Para provar estabilidade é suficiente mostrar que a função V aproxima zero como aproximações de tempo infinitas. A função V é sempre utilizada em sistemas lineares mas, às vezes, é difícil encontrá-la em sistemas não lineares.

O Conceito de Estabilidade

Informalmente, estabilidade refere-se a um sistema que tem seu funcionamento inalterado quando está sofrendo alguma interferência externa. Nestes conceitos existem vários pontos que devem ser discutidos antes que uma definição precisa de estabilidade possa ser apresentada. São eles:

1. Como estabilidade envolve a idéia de uma possível troca no estado ou saída do sistema, o sistema deve ser pelo menos um sistema temporal para fazer sentido falar de estabilidade. Em outras palavras, um sistema estático não pode ser estável ou instável;
2. O funcionamento preciso, deve ser declarado. Por exemplo, as duas declarações que seguem, referem-se a conceitos diferentes:
 - A temperatura de um mamífero é estável. Neste caso é um valor estável. Está se lidando com a estabilidade de um ponto de equilíbrio;
 - Observando as tomadas de pressão intra-ventricular, observa-se que os sinais apresentam um comportamento estável. Neste caso este não é um valor estável, mas a forma de um sinal de tempo ou uma trajetória que é estável. Está se lidando com a estabilidade de uma trajetória;
3. A importância da interferência externa. Considerando os dois exemplos apresentados anteriormente, é conhecido que a temperatura de um mamífero é estável mas perde sua estabilidade se o animal é exposto à temperaturas muito baixas, e

no caso de pressão intra-ventricular e contrações extra-sistólicas podem perturbar a estabilidade do funcionamento;

4. A duração da perturbação também é importante. É completamente diferente falar sobre estabilidade durante a presença de uma perturbação ou se está interessado em saber se depois de ocorrida uma perturbação, o sistema é capaz de recuperação. Como um exemplo, suponha um mamífero sendo mergulhado na água. A perturbação é a falta de respiração. O resultado é completamente diferente seguindo o tempo, o mamífero não pode respirar!

Existem várias condições diferentes sob as quais é possível falar de estabilidade admitindo ter vários conceitos relacionados à estabilidade.

Começa-se apresentando a definição proposta por Liapunov em sua tese de doutorado (1892). Esta definição é válida para sistemas lineares e não lineares e se refere à estabilidade de um ponto de equilíbrio.

Definição 8.2.1 Ponto de Equilíbrio: *Suponha um sistema onde sua entrada é nula para todo o tempo maior que um certo tempo $t = T$. O valor de estado do sistema ($x_0 \in X$) é um ponto de equilíbrio do sistema se $\forall A > T$ tem-se $x(T) = x_0 \rightarrow x(t) = x_0$. Existem Conceitos análogos referentes à saídas de sistemas.*

Definição 8.2.2 Trajetória de Equilíbrio: *Suponha um sistema onde sua entrada é nula para todo o tempo maior que um certo tempo $t = T$. Se o estado do sistema é periódico $\forall t > T$, diz-se que os valores do estado do período constituem uma trajetória de equilíbrio. Existem Conceitos análogos e se refere à saída do sistema.*

Definição 8.2.3 Estabilidade de Liapunov: *Considera-se um sistema com entrada u e saída y . O sistema é dito estável se dado um número positivo ϵ , é possível achar outro η positivo tal que se o sistema tem entrada nula $\forall t > 0$, se as condições iniciais do estado estão dentro de uma esfera de raio ϵ , lá existe um T para qual o estado $\forall t > T$ pertence a uma esfera de raio η .*

Definição 8.2.4 Estabilidade Local: *Um ponto de equilíbrio (uma trajetória) é dito ser um local estável se existe um conjunto aberto incluindo o ponto de equilíbrio no qual, todas as condições iniciais correspondem a um comportamento estável do sistema.*

Definição 8.2.5 Domínio de Estabilidade: *O domínio de estabilidade de um ponto de equilíbrio (uma trajetória) é o conjunto de ponto que corresponde as condições iniciais para as quais o sistema é estável.*

Definição 8.2.6 Estabilidade Global: *Um sistema é dito globalmente estável em uma região, se é estável para todas as condições iniciais pertencendo a pontos nesta região.*

Definição 8.2.7 Estabilidade de uma Trajetória: *A definição de estabilidade de uma trajetória segue a definição de estabilidade de um ponto de equilíbrio proposta por Liapunov. De um modo semelhante, é possível ter os conceitos de estabilidade local, domínio de estabilidade e estabilidade global que se referem a uma trajetória.*

Nas seções que seguem, serão apresentados o primeiro e o segundo métodos de Liapunov. O primeiro método também é chamado Método Indireto porque para concluir sobre a estabilidade de um sistema é necessário seguir um caminho indireto - pela solução do sistema. Semelhantemente o segundo método também é chamado Método Direto porque a conclusão de estabilidade do sistema é obtida sem passar pela solução, de um modo indireto.

Primeiro Método de Liapunov

Liapunov divide o problema geral analisando a estabilidade de sistemas não-lineares em duas classes. A primeira classe consiste em todos esses métodos nos quais a equação diferencial do sistema é resolvida em geral ou em particular. A solução é examinada e a estabilidade ou instabilidade é notada. Liapunov não teve nada a dizer de particular importância relativo à solução destas equações diferenciais não lineares, com talvez uma exceção. Ele mostrou que freqüentemente a solução deve ser buscada na forma de

uma série e deve ser provada por meio do segundo método que em muitos sistemas a solução apenas da primeira aproximação se dará usando informação de estabilidade.

Uma aproximação de primeira ordem, ou seja o desenvolvimento em série de Taylor da equação 8.2 resulta em:

$$x(k+1) = Ax(k) + Bu(k) \quad (8.3)$$

Se a rede for de neurônios contínuos, tem-se um sistema de equações diferenciais e um sistema análogo de equações, só que agora se trata de um sistema normal.

Definição 8.2.8 Estabilidade de James: *Um sistema linear é estável se para toda entrada limite corresponde uma saída limite. A palavra limite quer dizer uma função de tempo para qual existe um número real suficientemente grande tal que, o módulo do valor da função sempre é mais baixo que este número.*

Definição 8.2.9 Estabilidade de Bode: *Um sistema linear é estável se seu comportamento autônomo (solução quando a entrada é nula para todo o valor de tempo) não contém qualquer modelo cuja amplitude aumente com tempo.*

1. A solução de um sistema linear quando a entrada é nula pode ser expressa como a soma de várias funções, cada uma chamando um modelo do sistema.

As definições de James e Bode, se referem a sistemas lineares. Porém as condições são diferentes: James considera um sistema com entrada não nula (chamou um sistema forçado) e Bode um sistema autônomo (similarmente chamou um sistema não forçado). Esta diferença faz alguns sistemas serem considerado diferentemente estáveis ou instáveis seguindo James ou Bode. Estes sistemas são sistemas que possuem um modo oscilatório exterior sufocado - se eles recebem uma entrada com a mesma frequência do modo, a saída se torna uma oscilação crescente até a destruição do sistema, um comportamento tipicamente instável. Um exemplo é a possibilidade de quebrar um vidro por um som.

2. Em alguns manuais do exército era recomendado para não cruzar pontes caminhando ritmicamente porque é possível que a frequência seja igual a da ponte.

Esboço do Segundo Método

Às vezes é possível obter informação de estabilidade sobre um sistema sem resolver as equações diferenciais que descrevem este fato. O critério de Routh-Hurwitz provê uma aproximação para sistemas lineares. Por certas manipulações dos coeficientes da equação característica do sistema é possível determinar, sem resolver a equação de fato, se os pólos do sistema estão a direita ou meio a esquerda do plano. Às vezes é possível formar funções do sistema de equações e de tempo que possui certas propriedades desejáveis.

A teoria de equilíbrio de Lagrangian, para o qual o negativo da energia potencial do sistema está em um máximo, provê um exemplo. Tem-se estabilidade se a função de potencial-energia tem certas propriedades. Liapunov escreveu o agrupamento dos métodos nos quais uma função especial é gerada e é examinada para estabilidade em vez de solução das equações do sistema para a sua segunda classe. Sua contribuição principal foi o desenvolvimento de vários teoremas que, quando aplicada a uma função de Liapunov ou função V do sistema, indicará estabilidade ou instabilidade.

Porque esta aproximação gera informação de estabilidade sem a necessidade de resolver as equações de sistema, às vezes é chamado o método direto. O principal problema para aplicar o segundo método para sistemas de controle automáticos não-lineares é a construção da função V . Há algumas regras gerais para este processo, embora as pesquisas atuais aumentem as classes disponíveis. Agora é possível incluir todos os sistemas de uma só volta (loop) com um valor simples não-linear em uma classe geral, e algum progresso foi feito com a não-linearidade de tipo-derivado. Para fazer uma análise serve como um grande escopo de possibilidade, é desejável adotar um padrão ou forma canônica.

Este conceito era usado com sistemas descritos por um conjunto de equações diferenciais lineares no capítulo precedente e provará com sucesso nos sistemas não lineares. Sempre é possível expressar as equações de um “nth” autônomo - sistema não lineares ordenados na forma das “n” primeiras equações não lineares ordenadas em lugar de, talvez, formas mais comuns de uma única (nth) equação diferencial não-linear ordenada. Este conjunto de “n” equações pode ser expressa em termos de divergências sobre algum

ponto operacional estático.

$$\begin{aligned} \frac{dx_1}{dt} &= X_1(x_1, \dots, x_n) \\ \frac{dx_2}{dt} &= X_2(x_1, \dots, x_n) \\ &\text{---} \quad \text{---} \quad \text{---} \\ \frac{dx_n}{dt} &= X_n(x_1, \dots, x_n) \end{aligned} \tag{8.4}$$

As funções X_1, \dots, X_n podem ser, em geral, qualquer função não linear que satisfaça a condição

$$\begin{aligned} X_1 = X_2 = \dots = X_n = 0 \\ \text{para} \\ x_1 = x_2 = \dots = x_n = 0 \end{aligned} \tag{8.5}$$

Por certo, operações como a aplicação do teorema de Krasovskii será requerido posteriormente, que as funções não lineares sejam diferenciáveis. As variáveis x_1, \dots, x_n são variáveis de estado. A função V será construída usando as variáveis de estado x_1, x_2, \dots, x_n ou as funções X_1, X_2, \dots, X_n das variáveis de estado. Na aplicação dos teoremas de Liapunov, as condições de estabilidade obtidas a partir da função particular V são, em geral, suficientes mas não necessárias. Além disso a função de Liapunov para um sistema particular não é única. Estes fatos conduzem à várias conclusões:

1. Deveria uma função particular V falhar para mostrar que um sistema particular é estável ou instável. Não há garantia que qualquer outra função não poderia ser encontrada que demonstra estabilidade ou instabilidade. Em outras palavras, fracasso do teorema de estabilidade ou instabilidade sob qualquer condição determinada não implica a conversão;
2. Deveria uma função particular V demonstrar a estabilidade de um sistema não linear, e a partir desta função V certamente limites de estabilidade dos parâmetros do sistema tem sido derivadas. Não há garantia que excedendo estes limites na verdade causarão ao sistema instabilidade. Em outras palavras as exigências de estabilidade são, quase sem exceção, rigorosas demais.

Adicionalmente, deveria ser notado que, como sempre em sistemas não-lineares, a estabilidade é definida aqui para um sistema não forçado. (Uma exceção é um caso especial no qual Malkin provou estabilidade total em uma classe de sistemas estáveis assintoticamente globais.) As relações de Eq. 8.4 deste modo representam um sistema de ordem “n” sob qualquer condição inicial, mas sem forçar função e parâmetros estacionários.

Com ajuda dos teoremas demonstrados abaixo, o exame de estabilidade de um sistema de controle pode tomar lugar, não só para pequenos, mas também para grandes divergências de equilíbrio. A aplicabilidade destes teoremas está limitado só pelo alcance definido de aplicabilidade dos sistemas de equações.

O Conceito de Definição de Sinal

Defina uma função das variáveis de estado como:

$$V = V(x_1, x_2, \dots, x_n) \quad (8.6)$$

As variáveis x_1, \dots, x_n são coordenadas linearmente independentes do espaço de estados n-dimensional. Quando $n = 2$, tem-se um plano de estado, ou fase do plano, se uma das coordenadas é derivada da outra. A função V tem um valor definido a cada ponto neste espaço. Com certeza valores das coordenadas da função V são zero (por exemplo a $x_1 = x_2 = \dots = x_n = 0$), e em uma certa região sobre estes pontos é contínuo.

Definição 8.2.10 *A função V é conhecida como definida positiva ou definida negativa em uma determinada região sobre a origem se todos os pontos nesta região tem o mesmo sinal (positivo ou negativo), e exceto à origem, não está em nenhuma parte zero.*

Definição 8.2.11 *A função V é conhecida como semidefinida se tem o mesmo sinal ao longo da região exceto em certos pontos nos quais é zero, além disso deve ser zero na origem. (Nós desejamos $V(0) = 0$ de forma que a origem é um minimax.)*

Definição 8.2.12 *A função V é conhecida como indefinida se em determinada região sobre a origem assume sinais variados.*

Estas são as definições habituais achadas em livros de álgebra linear. Porém, são úteis na ordem para simplificar o uso em casos práticos para restringir um pouco mais o conceito habitual de definição. As restrições são:

- Primeiro, para ser uma função definida deve ser zero na origem;
- Segundo, definição em um sentido global é compreendida a menos que, caso contrário, declarada;
- Terceiro, uma função V definida globalmente positiva (negativa) tem que ir para o máximo (mínimo) infinitamente, quando qualquer ou todos dos seus argumentos tendem a infinito;
- Finalmente, para definições finitas, a região na qual a função V é definida deve ser fechada e deve incluir a origem.

Serão dados exemplos de todos os três tipos de funções. Deixe $n = 2$ e $V = x_1^2 + x_2^2$. Esta é uma função positivo-definida desde que qualquer lugar no plano x_1x_2 a função V é positiva, exceto quando $x_1 = x_2 = 0$. Similarmente, para qualquer n arbitrário,

$$V = x_1^2 + \dots + x_n^2 \quad (8.7)$$

é positivo definida e

$$V = -(x_1^2 + x_2^2) \quad (8.8)$$

é negativo definida. Está claro que o negativo de uma função positiva definida é uma função definida negativa.

Considere a função

$$V = x_1^2 + x_2^2 \quad (8.9)$$

para $n = 3$. Isto não está definido porque é nulo para um x_3 arbitrário se $x_1 = x_2 = 0$, isto é, ao longo do eixo x_3 . Isto ocorrendo, então, a função é positiva semidefinida. Finalmente, a função,

$$V = x_1 + x_2 \quad (8.10)$$

é indefinida até mesmo para $n = 2$, desde que seja positiva para todos os pontos à direita da linha $x_1 = -x_2$ e negativo para todos os pontos à esquerda desta linha.

Primeiro Teorema de Liapunov

O estudo de sistemas lineares e o critério de estabilidade deles são interessantes até mesmo em casos não lineares devido ao teorema de Liapunov em estabilidade local. Este teorema de estados que estuda o comportamento de um sistema não linear perto de um ponto de equilíbrio, pode ser dito que o sistema não-linear é estável de instável, se o sistema linearizado é assintoticamente estável (a resposta livre vai zerar quando o cronômetro incrementar) ou instável. No caso de um sistema linearizado estável mas não assintoticamente (ex: o caso de um centro 0, nada pode ser concluído sobre o comportamento do sistema não-linear estudando somente a aproximação linearizada disto.

A prova deste teorema é muito simples. Considera-se um sistema não linear definido pela equação de estado abaixo e F uma função analítica.

$$\frac{dx}{dt} = F(x, u(k)) \quad (8.11)$$

Sob estas condições é possível escrever a expansão da série de Taylor perto da origem desta função. Tem-se:

$$\frac{dx}{dt} = F(0) + \frac{F'(0)}{1!} + \frac{F''(0)}{2!} + \dots \quad (8.12)$$

Como o denominador dos vários termos desta expansão aumenta tão rápido quanto o fatorial dos naturais, pode ser esperado que o termo dominante seja o primeiro. Isto conclui a prova.

Também há um dito método direto de Liapunov.

Método Direto dos Teoremas de Liapunov na Estabilidade e Instabilidade de Sistemas não-Lineares.

O método direto de Liapunov inclui o teorema de estabilidade e instabilidade.

Qualquer função

$$V = V(x_1, x_2, \dots, x_n) \quad (8.13)$$

de sinal definido é chamada uma função de Liapunov se é zero para $x_1 = x_2 = \dots = x_n = 0$ e se todos esses argumentos x_1, x_2, \dots, x_n , são as soluções do sistema de Eq. 8.4.

A expressão geral durante o tempo derivado de uma função V de n argumentos é:

$$\frac{dV}{dt} = \frac{\partial V}{\partial x_1} \frac{dx_1}{dt} + \frac{\partial V}{\partial x_2} \frac{dx_2}{dt} + \dots + \frac{\partial V}{\partial x_n} \frac{dx_n}{dt} \quad (8.14)$$

Usando a equação de estado, definindo a dinâmica do sistema da Eq. 8.4, tem-se:

$$W = \frac{\partial V}{\partial x_1} X_1 + \frac{\partial V}{\partial x_2} X_2 + \dots + \frac{\partial V}{\partial x_n} X_n \quad (8.15)$$

Onde X_1, \dots, X_n , são funções conhecidas de x_1, x_2, \dots, x_n . Assim, a função de W é, como é V , uma função de variáveis de estado e vai zerar à origem do espaço de estado. O conceito de definição de sinal assim aplica-se igualmente W . O teorema de estabilidade pode ser declarado como segue:

Teorema 8.2.1 *Para um sistema da ordem de “nth” descrito pelas suas equações de estado. Se uma função V de sinal definido pode ser selecionada tal que seu tempo W derivado também está definido e de sinal oposto, o sistema determinado é estável assintoticamente em uma região suficientemente pequena que inclui a origem. Se a função W somente é semidefinida e de sinal oposto, o sistema é estável mas não necessariamente estável assintoticamente.*

Pode ser possível obter tal como uma função W somente sobre uma região finita; neste caso a prova de estabilidade estende-se somente sobre esta região.

Este teorema é suscetível a uma demonstração geométrica bastante simples. Assuma para simplicidade um sistema de segunda ordem. Para $n = 2$ a equação de estado pode ser escrita como:

$$\begin{aligned}\frac{dx_1}{dt} &= X_1(x_1, x_2) \\ \frac{dx_2}{dt} &= X_2(x_1, x_2)\end{aligned}\tag{8.16}$$

Deixa-se levar como função V uma família de elipses, um para cada valor de V , com parâmetros ' a ' e ' b '. Assim

$$V = a^2 x_1^2 + b^2 x_2^2\tag{8.17}$$

Os parâmetros ' a ' e ' b ' são coeficientes. Isso deve ser determinado de tal modo que a função V alcance valores constantes sucessivamente maiores. Para $V = 0$ nós temos a origem do espaço de estado. Para os outros valores elipses concêntricas. Para ter estabilidade é suficiente que a trajetória definida pelas equações de estado atravesse cada uma das elipses na direção da origem, ou em outras palavras, seu tempo derivativo deve ser negativo.

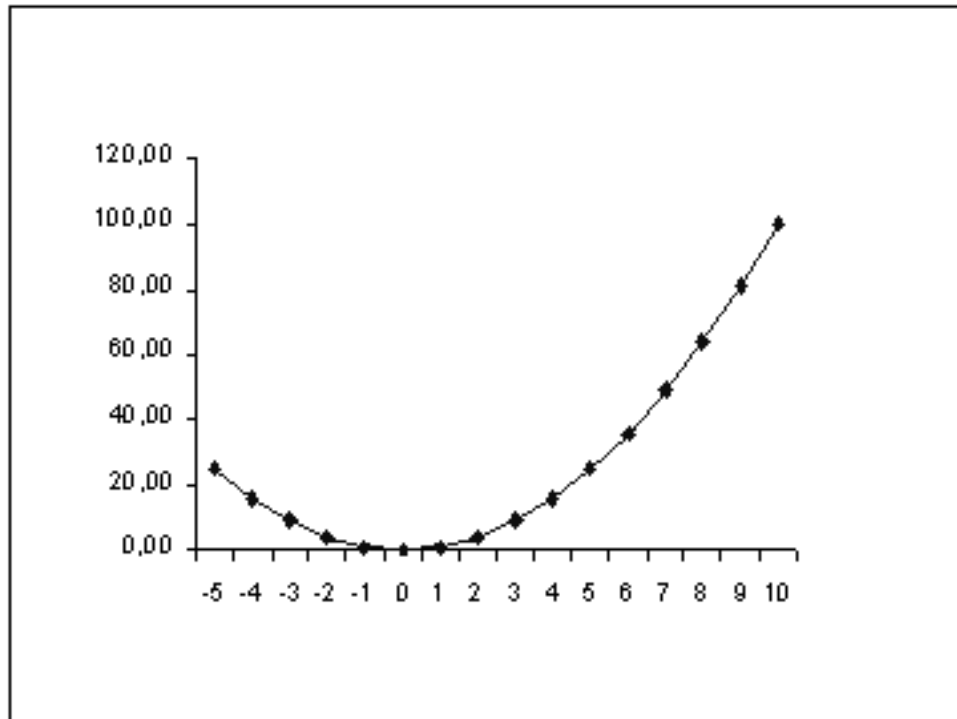


Figura 8.1: Uma função definida positiva

A primeira destas funções descreve a origem do espaço de estado, e as funções restantes descrevem superfícies elipsoidais no estado Figura 8.1. Superfícies de constantes V para constantes sucessivamente maiores, $V = 0, C_1, C_2, C_3, \dots$

Se W é definido negativo isto é, se $dV/dt < 0$, o ponto de estado doente para qualquer condição inicial deve mover-se na direção de valores menores de V , isto é, para a origem. Isto significa que divergências de ponto de equilíbrio dado no curso transitório, e o sistema é assintoticamente estável. Por analogia esta demonstração pode ser estendida à sistemas de ordem 'n'. Porém, se a função W não está definida mas só semidefinida, a trajetória do ponto de estado não corta alguma superfície particular $V = C_i$ mas antes pode tornar-se tangente a isto, nos pontos ao qual $W = 0$. Torna-se necessário demonstrar que o ponto de estado não pode permanecer no ponto ao qual $W = 0$, mas deve continuar em direção a origem, se estabilidade assintótica e não mera estabilidade será provado.

Isto não é difícil de fazer por argumentos físicos, ou outros meios, em muitos casos práticos; ou, o segundo método de Liapunov pode prover esta ajuda. Se um teorema de Liapunov somente prova estabilidade assintótica, uma função W definida pode ser encontrada.

O segundo teorema de Liapunov, ou denominado teorema de instabilidade, pode ser declarado como segue:

Teorema 8.2.2 *Para um sistema de ordem 'n' descrito pelas equações de estado, se existir uma função V de real-valor contínuo (x_1, \dots, x_n) com tempo derivado $dV/dt = W$ definido negativo, então:*

1. O sistema é instável na região finita para aquele V que não é positivo semidefinido;
2. A resposta do sistema é ilimitada como $t \rightarrow \infty$ se V não é nenhum semidefinida globalmente positiva. Invertendo os sinais ao longo da definição não muda o significado;

Por causa dos negativos envolvidos, este teorema não é talvez tão transparente quanto o primeiro, mas novamente suas correções podem ser mostradas graficamente.

Para um determinado sistema de segunda-ordem uma certa função V indefinida pode ser encontrada, tal que seu W derivado é definido positivo. As linhas de $V(x_1, x_2) = C_i$ pode ser organizado no plano de estado como mostrado na Figura 8.1. As linhas íngremes correspondem ao valor $V = 0$ e separa o plano de estado em regiões nas quais $V < 0$ e $V > 0$.

Considere uma condição arbitrária inicial representada na Figura 8.1 através do ponto M. Desde que em todos os lugares $W = dV/dt > 0$, o ponto de estado M muda continuamente na direção de valores crescentes de C_i ; assim, pode aproximar a origem durante um tempo, mas eventualmente tende para infinito. O sistema é instável.

Deveria ser notado que o teorema de instabilidade é de certo modo mais poderoso que o teorema de estabilidade, desde que sempre é possível selecionar uma função W definida negativa. Então, se uma função V formal pode ser obtida e se as condições para instabilidade não estão satisfeitas, as condições para estabilidade são automaticamente encontradas, considerando que fracasso para conhecer o critério de estabilidade não implica a conversão. Infortunadamente, às vezes, é muito difícil de achar a função V formal para um determinado W . Porém, esta técnica pode ser usada no caso especial de sistemas lineares.

8.2.2 A instabilidade na BAM

Após essa revisão da teoria de estabilidade, partiu-se para os testes referentes à matriz de conexões, com o intuito de provar ou não a teoria de Cohen/Grossberg quanto a estabilidade do sistema, verificando se este é ou não estável.

Pode-se verificar se um sistema é instável ou estável verificando por método indireto se a solução aumenta ou não. Se a solução cresce com o tempo indefinidamente é porque o sistema é instável, ao modo que se a solução não cresce com o tempo, é porque o sistema é estável. Ou seja, se as variáveis de estado do sistema forem diminuindo progressivamente, o sistema é estável.

Se dado um sistema de tempo discreto $k = 1, k = 2, k = 3$, onde $k \in [N]$ a cada instante existe um vetor estado do sistema. Esse vetor deve ter suas componentes diminuindo progressivamente. Se colocar uma entrada ele não vai tender à zero, mas

vai tender para um ponto.

Para verificar se o modelo de um vetor é menor que outro, multiplica-se esse por uma matriz que faça com que ele se reduza. Vamos supor que esse vetor seja expresso por uma base tal que essa matriz fique diagonal. Neste caso a primeira componente não é mais a excitação do primeiro neurônio, mas ele pode ser obtido através de uma transformação de coordenadas.

Se o sistema é discreto e os valores próprios forem menores do que um, a componente seguinte será menor do que um. Valores próprios são os que aparecem quando se diagonaliza a matriz. Estabilidade de sistema discreto tem valor próprio em módulo menor do que um.

Em sistemas não-lineares acontece de ser instável em uma região mas tem um limite de estabilidade, quando chega num certo ponto ele não continua, é limitado. Isso é, toda não-linearidade tem uma saturação.

Isso está relacionado à sistemas lineares. Quer dizer estabilidade do ponto de equilíbrio de acordo com o primeiro teorema de Liapunov, que diz que estabilidade de um sistema não linear em um ponto de equilíbrio é dado pela estabilidade do sistema linearizado em torno desse ponto.

Partindo-se do princípio de que uma rede BAM deve ser estável se sua matriz de conexões for simétrica ($W_{ij} = W_{ji}$) e sua diagonal principal for nula ($W_{ii} = 0$), conforme apresentado por Cohen [49, 17], não havia motivos para estar ocorrendo a instabilidade da rede.

Começou-se então a fazer testes para tentar confirmar se realmente a matriz de pesos era estável. Para isso, verificou-se os seus valores próprios. Os valores próprios de uma matriz quadrada, em módulo devem ser menores do que 1 (um) para que esta seja estável.

Porém, esse teste nos mostrou que essa matriz possui valores próprios, em módulo maiores do que 1, chegando a aproximadamente 13 (treze) para alguns valores, longe do esperado, que seriam valores menores do que 1 para que a rede permanecesse estável.

Mostra-se aqui um exemplo provando esse fato. Não será utilizada a matriz originalmente testado, como matriz de conexões da rede BAM apresentada nesse trabalho,

pois esta é muito grande e seria difícil representá-la em sua totalidade em uma página.

A matriz A apresentada na Equação 8.18 satisfaz as condições do teorema de Grossberg; porém, seus valores próprios mostrados na Equação 8.19, não satisfazem como valores próprios nem para valores de uma sistema contínuo, nem para um sistema discreto. Como esses valores próprios crescem ultrapassando em módulo o valor um, constatou-se que realmente a rede não poderia estabilizar.

$$A = \begin{bmatrix} 0.0 & -0.9 & -0.3 & 0.5 \\ -0.9 & 0.0 & -0.2 & 0.8 \\ -0.3 & -0.2 & 0.0 & -0.5 \\ 0.5 & 0.8 & -0.5 & 0.0 \end{bmatrix} \quad (8.18)$$

Os valores próprios da matriz A calculados pelo *MatLab* são (Equação 8.19):

$$eig(A) = \begin{bmatrix} 0.8296 \\ 1.0570 \\ -0.4071 \\ -1.4795 \end{bmatrix} \quad (8.19)$$

Baseado nesses testes, e contradizendo o Teorema de Cohen/Grossberg, constatou-se que não basta a matriz ser simétrica e sua diagonal principal ser nula para que a rede BAM seja estável. Com isso, verifica-se que deve haver mais alguma condição não explícita no Teorema de Cohen/Grossberg, para que se alcance a estabilidade da rede.

Capítulo 9

Conclusões

9.1 Rede BAM

Com respeito a implementação da rede BAM, não foi possível implementá-la com sucesso, tendo em vista a incapacidade de mantê-la estável. Foram feitos testes com o objetivo de confirmar a estabilidade da rede, contudo os resultados obtidos foram uma rede totalmente instável, oscilando entre os valores máximos e mínimos das excitações dos neurônios, quando o esperado era obter valores que tendessem a uma solução. Ou seja, preferencialmente um dos neurônios com seu valor crescendo tendendo a 1, enquanto os demais estabilizando em níveis de ativação mais baixos, obtendo, com isso, uma resposta.

As simulações feitas levaram a concluir que os escritos de Cohen/Grossberg possuem condições necessárias, mas não suficientes para implementar a rede BAM de forma estável. Feitos os testes com a matriz de conexões da rede, verificou-se que seus valores próprios, em módulo, ultrapassam o limite de 1, gerando assim instabilidade a rede. Pois para a rede estabilizar, seus valores próprios em módulo deveriam ser menores do que 1 para redes de neurônios funcionando de modo assíncrono e a tempo discreto. Concluindo-se assim que, apenas implementando a rede com a metodologia apresentada pelos referidos autores, não foi suficiente para que esta estabilize e apresente os resultados desejados.

9.2 Rede Direta

Com os testes efetuados com a rede direta, constatou-se que essa, surpreendentemente, apresentou um treinamento (aprendizado) extremamente rápido utilizando um número muito pequeno de exemplos. Esperava-se, baseado nos trabalhos de De Azevedo, um resultado menos animador. Segundo estudos realizados, para obter bons resultados seria necessário treinar esta rede com um número maior de exemplos de treinamento. Porém, com menos de 20 exemplos utilizados já se conseguiu uma taxa de aprendizado em torno dos 80% (oitenta por cento), fato esse que não era esperado.

Um dos fatores observados, que contribuíram para o resultado do rápido aprendizado, foi o fato de a base de exemplos ser composta por dados que possuem seus resultados de forma mutuamente exclusivos, facilitando assim a distinção um do outro.

9.3 Análise Geral

Devido a impossibilidade de efetuar a conclusão por meio da comparação entre as redes, BAM e Direta, pelos motivos gerados pela instabilidade da primeira, como proposto no início deste estudo, limitando-o assim a uma análise simples do desenvolvimento e das dificuldades das redes.

Decorrente do problema apresentado, parte desta dissertação destinou-se a apresentar as dificuldades de gerar um rede estável para o exemplo proposto, tratando dos motivos que não permitiram a implementação com sucesso da rede BAM, motivos estes que deixaram esta rede instável.

No que tange a aplicação da Rede Direta, foi obtido sucesso, mais do que a expectativa estudada nos autores do assunto, devido a rapidez na geração de resultados. O trabalho apresentou a implementação da rede, os testes executados com a mesma, assim como, a análise dos resultados encontrados.

As dificuldades de implementar a rede BAM foram decorrentes da falta de apresentação de detalhamento nos trabalhos desenvolvidos até então, relacionados a estabilidade de Rede BAM. Levando-se a trabalhar a respeito de um assunto que, apesar dos estudos já realizados, merece ainda várias pesquisas antes da utilização prática do

assunto.

Detectou-se que o teorema de Cohen/Grossberg é geralmente apresentado com suposições implícitas e que uma matriz real de conexões simétrica com diagonal principal nula é condição necessária mas não suficiente para a estabilidade de uma rede BAM.

Por fim, fica clara a intensão desse trabalho em realizar uma análise comparativa da aplicabilidade das redes BAM e Direta no exemplo de casos clínicos de reumatologia, impedido pela surpresa da incapacidade de obtermos uma rede BAM estável. Não espera-se deste trabalho resultados conclusivos sobre o assunto, no entanto, a contribuição apresentada revelou a necessidade de intensificar os estudos que viabilizarão a implementação estável das redes BAM. Gerando, com isso, um foco de debate no assunto proposto para este trabalho, ficando esta questão a ser estudada mais profundamente no decorrer dos trabalhos acadêmicos.

ANEXO 1

ARQUIVOS DE TREINAMENTO E TESTES DA REDE DIRETA (NEURALWORKS)

Arquivo TESTE.NNA: Arquivo com a base de testes da rede com 18 (dezoito) exemplos. Esta é a base de dados de teste no formato texto como exigido pelo NeuralWorks.

- Cada uma das 17 colunas corresponde respectivamente aos sintomas/exames relacionados na metodologia.
- A linha que inicia com o símbolo & corresponde aos 3 diagnósticos possíveis correspondendo respectivamente as 3 doenças relacionadas na metodologia.
- Os pesos variam de -0.9 a 0.9 significando graus de inibição e ativação.

```
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 -0.9 0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
0.9 0.9 -0.9 0.5 0.9 0.6 0.8 0.6 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.8 -0.9 0.4 0.9 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.4 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9
& 0.0 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 -0.9 0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
0.9 0.9 0.9 0.5 -0.9 0.4 0.8 0.4 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 0.9 0.9 0.9 -0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
0.9 -0.9 -0.9 0.9 0.9 0.4 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.6 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.4 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
```

Arquivo TRIN32.NNA: Arquivo com a base de treinamento da rede com 32 (trinta e dois) exemplos. Esta é a base de dados de treinamento completa no formato texto como exigido pelo NeuralWorks.

- Cada uma das 17 colunas corresponde respectivamente aos sintomas/exames relacionados na metodologia.
- A linha que inicia com o símbolo & corresponde aos 3 diagnósticos possíveis correspondendo respectivamente as 3 doenças relacionadas na metodologia.
- Os pesos variam de -0.9 a 0.9 significando graus de inibição e ativação.

```
0.9 0.9 0.9 0.5 0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 -0.9 0.4 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.8 0.8 0.6 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.8 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 -0.9 0.9 0.9 0.9 0.4 0.6 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
0.9 0.9 0.9 0.5 0.9 0.4 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.8 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.6 -0.9 0.2 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.8 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.4 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.6 -0.9 0.4 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.9
& 0.0 0.9 0.0
0.9 0.9 -0.9 0.9 0.9 -0.9 0.4 0.2 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
0.9 -0.9 0.9 0.9 -0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 0.9 -0.9 0.9 -0.9 0.2 0.6 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 0.4 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.6 0.6 0.4 0.9 -0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 0.9 0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
0.9 0.9 -0.9 0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.4 -0.9 0.9 -0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9
```

```

& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 -0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.6 0.4 0.4 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 0.9 -0.9 0.5 0.9 0.2 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9
& 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.6 0.4 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9
& 0.0 0.9 0.0
0.9 0.9 -0.9 0.5 -0.9 0.6 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9
& 0.0 0.0 0.9
0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9
& 0.9 0.0 0.0

```

Arquivo TESTE32.NNR: Arquivo com os resultados obtidos após o treinamento com a base de 32 exemplos.

O NeuralWorks fornece os resultados no formato texto da seguinte forma:

- As 3 primeiras colunas correspondem aos resultados esperados e as 3 últimas aos resultados obtidos.

0.900000	0.000000	0.000000	0.898163	0.003381	-0.001418
0.000000	0.900000	0.000000	-0.007206	0.010845	0.895921
0.000000	0.900000	0.000000	-0.049032	0.949077	-0.001726
0.000000	0.900000	0.000000	0.213736	0.687695	-0.001225
0.000000	0.000000	0.000000	0.003665	0.018879	0.874617
0.900000	0.000000	0.000000	0.913741	-0.013051	-0.000870
0.000000	0.000000	0.900000	0.061997	-0.029010	0.900694
0.900000	0.000000	0.000000	0.897088	0.004253	-0.001142
0.000000	0.000000	0.900000	0.002013	0.000304	0.897276
0.000000	0.900000	0.000000	0.466241	-0.047424	0.898075
0.000000	0.000000	0.900000	-0.000939	0.000197	0.900459
0.000000	0.000000	0.900000	-0.000709	-0.000333	0.900476
0.000000	0.900000	0.000000	-0.043971	0.943818	0.001114
0.000000	0.900000	0.000000	-0.042545	0.942358	0.001034
0.000000	0.900000	0.000000	0.052514	0.849396	-0.001756
0.000000	0.900000	0.000000	0.000299	0.707634	0.187089
0.900000	0.000000	0.000000	0.304138	0.395549	0.054510
0.000000	0.900000	0.000000	-0.000413	0.763900	0.134200

Existem outros arquivos gerados com os resultados dos teste variando o número de casos de 2 (dois) a 32 (trinta e dois) incrementando de 2 (dois) casos para cada novo teste.

A tabela a seguir mostra os resultados de todos os teste realizados com a rede nas configurações mostradas no desenvolvimento.

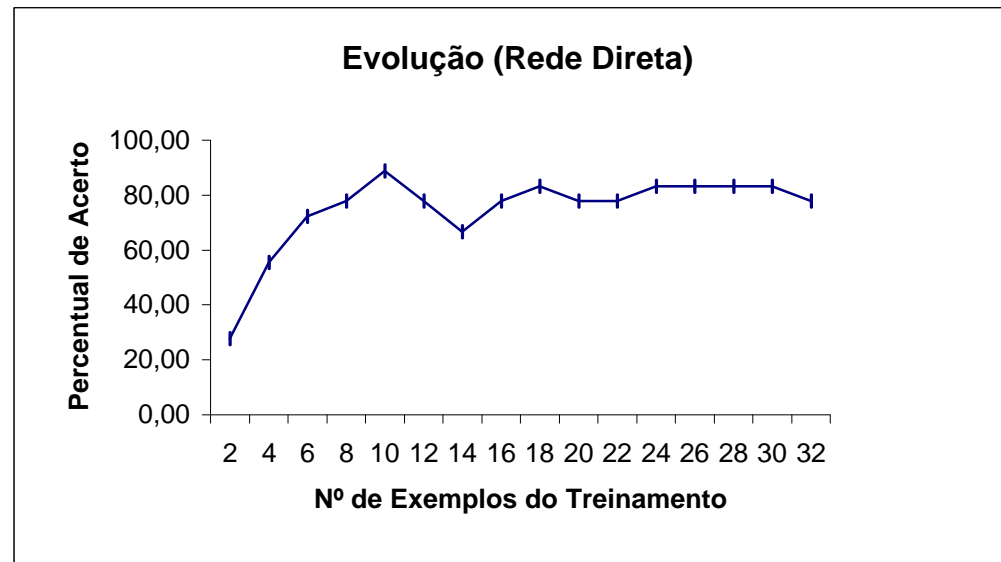
Saídas Experadas			02 Exemplos			04 Exemplos			06 Exemplos			08 Exemplos		
0.900000	0.000000	0.000000	0.887952	0.000552	0.022705	0.914689	-0.013672	-0.002949	0.897250	-0.000057	0.002218	0.895835	0.025684	-0.013870
0.000000	0.900000	0.000000	0.473912	-0.014996	0.517730	0.068537	0.015944	0.797770	0.092380	0.068038	0.653354	0.000491	0.069344	0.829755
0.000000	0.900000	0.000000	0.884194	0.006043	0.020156	0.446529	0.633761	-0.014299	0.592483	0.612070	-0.036251	0.265394	0.763244	-0.021009
0.000000	0.900000	0.000000	0.891094	0.001073	0.011707	0.891690	0.014334	-0.003747	0.038737	0.859730	0.000627	0.037674	0.872469	-0.004382
0.000000	0.000000	0.000000	0.814936	0.006310	0.059648	0.387532	0.343917	0.039887	0.450688	0.139526	0.131653	0.640834	0.028624	0.162388
0.900000	0.000000	0.000000	0.884251	0.004032	0.023091	0.909692	-0.006828	-0.002465	0.908054	-0.006475	-0.001732	0.912572	-0.018019	0.010262
0.000000	0.000000	0.900000	0.642508	-0.010520	0.399151	0.197681	0.030057	0.635598	0.319649	0.029940	0.460335	0.081192	-0.007239	0.833006
0.900000	0.000000	0.000000	0.902220	-0.011329	0.017460	0.904831	-0.004290	0.000856	0.906235	-0.007872	0.001573	0.915214	-0.024835	0.023951
0.000000	0.000000	0.900000	0.826723	-0.010204	0.081076	0.057419	0.747207	0.043420	0.407620	0.131258	0.172028	0.313031	0.038849	0.472943
0.000000	0.900000	0.000000	0.793161	-0.015120	0.155033	0.612685	0.015612	0.221480	0.196041	0.012319	0.655536	0.279605	0.010290	0.608828
0.000000	0.000000	0.900000	0.346873	-0.010897	0.607324	0.096283	0.008687	0.782240	0.024753	0.001865	0.874151	0.021944	-0.011853	0.896116
0.000000	0.000000	0.900000	0.562597	-0.008530	0.367677	0.008995	0.341791	0.542857	-0.039041	0.695573	0.562838	0.003754	0.013455	0.886190
0.000000	0.900000	0.000000	0.896423	-0.001981	0.007138	0.901352	0.003817	-0.005946	0.041560	0.850237	0.003525	0.041477	0.879678	-0.006352
0.000000	0.900000	0.000000	0.896410	-0.001247	0.006668	0.901681	0.003257	-0.005626	0.060444	0.835873	0.000481	0.058102	0.869798	-0.007626
0.000000	0.900000	0.000000	0.891357	0.000696	0.011749	0.880666	0.033228	-0.006604	0.007572	0.894209	-0.001677	0.007269	0.894713	-0.001310
0.000000	0.900000	0.000000	0.883240	0.003107	0.004943	0.156934	0.833103	-0.024780	-0.035225	0.925515	0.030808	-0.016517	0.908510	0.014415
0.900000	0.000000	0.000000	0.898507	-0.015348	0.024112	0.845398	0.023254	0.019532	0.100400	0.668473	0.053649	0.121406	0.341708	0.248071
0.000000	0.900000	0.000000	0.882719	0.002570	0.005488	0.201923	0.808545	-0.025575	-0.032770	0.922643	0.027572	-0.012358	0.904368	0.014144

Saídas Experadas			10 Exemplos			12 Exemplos			14 Exemplos			16 Exemplos		
0.900000	0.000000	0.000000	0.654021	0.322249	0.021454	0.576166	0.365789	-0.010222	0.149950	0.729607	0.005017	0.400368	0.612709	0.007547
0.000000	0.900000	0.000000	-0.044503	0.711078	0.560129	-0.013944	0.910770	-0.001802	-0.017059	0.906835	0.012421	-0.025368	0.920263	0.004617
0.000000	0.900000	0.000000	0.018622	0.896522	-0.009489	0.000947	0.895985	-0.004403	0.004432	0.898676	-0.003423	0.010404	0.894790	-0.003440
0.000000	0.900000	0.000000	0.171681	0.720207	0.005195	0.087695	0.817544	0.002876	0.059855	0.852022	-0.006470	0.151896	0.758569	-0.002089
0.000000	0.000000	0.000000	0.833454	-0.025372	0.135809	0.491176	-0.047192	0.865701	0.551813	-0.047328	0.880999	0.129605	-0.036854	0.892535
0.900000	0.000000	0.000000	0.915172	-0.015835	-0.008148	0.922195	-0.019654	-0.006111	0.914972	-0.022672	0.012749	0.903037	-0.002402	0.001173
0.000000	0.000000	0.900000	0.024059	-0.002462	0.879889	0.063197	0.823661	0.030576	-0.042379	0.771145	0.485884	0.020572	0.440671	0.469489
0.900000	0.000000	0.000000	0.916125	-0.014326	-0.007963	0.913629	-0.012289	-0.000001	0.933770	-0.049241	0.507330	0.888853	-0.001994	0.013529
0.000000	0.000000	0.900000	0.136864	0.210869	0.450535	0.020133	0.885366	0.012515	-0.040398	0.744208	0.479000	0.006244	0.196237	0.722576
0.000000	0.900000	0.000000	0.112270	0.121140	0.638949	0.021965	0.887685	0.008345	0.167328	0.106736	0.513635	0.040858	0.045555	0.798225
0.000000	0.000000	0.900000	0.003376	-0.001072	0.897096	0.002287	-0.002643	0.901188	-0.001598	-0.002635	0.904212	-0.000528	-0.000508	0.901844
0.000000	0.000000	0.900000	-0.040260	0.476104	0.697294	-0.011288	0.007077	0.895498	-0.005885	0.001529	0.904090	-0.002524	0.000940	0.901581
0.000000	0.900000	0.000000	-0.035520	0.936709	0.001741	-0.013644	0.910516	-0.001903	-0.008565	0.910523	-0.002935	-0.029009	0.924480	-0.002014
0.000000	0.900000	0.000000	-0.035152	0.935926	0.002724	-0.013318	0.910257	-0.001929	-0.008452	0.910474	-0.002996	-0.027947	0.923539	-0.001967
0.000000	0.900000	0.000000	0.045402	0.853166	0.001555	0.023461	0.879245	0.002247	0.017861	0.885015	-0.003084	0.033422	0.866812	-0.000438
0.000000	0.900000	0.000000	-0.034727	0.938311	-0.006812	-0.012096	0.909232	-0.000461	-0.011769	0.905911	0.006207	-0.025234	0.916934	0.005174
0.900000	0.000000	0.000000	0.533754	0.117947	0.154270	0.043834	0.860866	0.015928	0.015669	0.114274	0.749542	0.019929	0.054644	0.820472
0.000000	0.900000	0.000000	-0.034723	0.938165	-0.006399	-0.011892	0.909139	-0.000412	-0.010891	0.906192	0.004778	-0.025078	0.917988	0.003604

Saídas Experadas			18 Exemplos			20 Exemplos			22 Exemplos			24 Exemplos		
0.900000	0.000000	0.000000	-0.001439	0.911513	-0.010085	0.083904	0.794406	0.008524	0.476661	0.375587	0.012796	0.904681	0.004704	0.000226
0.000000	0.900000	0.000000	-0.013497	0.891542	0.019897	-0.002572	0.872109	0.031145	0.002255	0.397969	0.492289	-0.000686	0.759155	0.150085
0.000000	0.900000	0.000000	-0.002852	0.902876	-0.000086	-0.028605	0.933059	-0.010214	-0.045571	0.946723	-0.016120	-0.001274	0.906156	-0.002840
0.000000	0.900000	0.000000	0.049559	0.861752	-0.005805	0.030531	0.872819	-0.002658	0.086918	0.816103	-0.000826	0.044377	0.855100	-0.002260
0.000000	0.000000	0.000000	0.206244	-0.036648	0.810878	0.573906	-0.048226	0.897241	0.087947	-0.032151	0.896368	0.131200	-0.037439	0.892984
0.900000	0.000000	0.000000	0.925714	-0.018133	-0.006016	0.933728	-0.035575	0.006197	0.913215	-0.013917	0.001174	0.916527	-0.010508	-0.017429
0.000000	0.000000	0.900000	0.078400	0.114722	0.734363	-0.000150	0.407072	0.492200	0.016679	0.010151	0.870871	0.023751	-0.006066	0.885511
0.900000	0.000000	0.000000	0.907805	-0.019167	0.018981	0.876489	0.015821	0.005869	0.895091	-0.027252	0.066188	0.900564	-0.008652	0.012711
0.000000	0.000000	0.900000	0.014437	0.035759	0.858493	-0.000397	0.028153	0.872325	-0.000311	0.023130	0.877178	-0.001734	0.015300	0.885315
0.000000	0.900000	0.000000	0.019908	0.537098	0.344882	-0.014418	0.071264	0.861781	0.000029	0.028253	0.871598	-0.000427	0.160903	0.740751
0.000000	0.000000	0.900000	-0.002049	-0.000192	0.901679	-0.000861	-0.000539	0.901354	-0.000791	0.000105	0.900517	-0.000232	0.000681	0.900344
0.000000	0.000000	0.900000	-0.001151	-0.009378	0.909369	-0.001068	-0.000280	0.901304	-0.000800	0.000117	0.900513	0.001398	-0.009284	0.908862
0.000000	0.900000	0.000000	-0.004564	0.902824	0.002084	-0.027576	0.932772	-0.011643	-0.045571	0.946913	-0.018018	-0.002167	0.905751	-0.002472
0.000000	0.900000	0.000000	-0.004549	0.902948	0.001913	-0.028014	0.933075	-0.011553	-0.045010	0.946491	-0.017574	-0.002235	0.905569	-0.002393
0.000000	0.900000	0.000000	0.016127	0.887700	-0.001031	0.011002	0.892537	-0.003460	0.032986	0.867395	0.000029	0.012033	0.886090	-0.001177
0.000000	0.900000	0.000000	-0.008652	0.897090	0.010278	-0.041906	0.924747	0.093071	-0.027096	0.919096	0.015221	-0.001674	0.899886	0.003960
0.900000	0.000000	0.000000	0.057207	0.074738	0.791201	-0.001069	0.510399	0.393512	0.002627	0.085945	0.807878	-0.001846	0.031708	0.867711
0.000000	0.900000	0.000000	-0.008032	0.898082	0.008871	-0.042243	0.930465	0.068164	-0.028234	0.923019	0.009948	-0.001738	0.901296	0.002454

Saídas Experadas			26 Exemplos			28 Exemplos			30 Exemplos			32 Exemplos		
0.900000	0.000000	0.000000	0.899626	0.001139	-0.000993	0.902292	0.000365	-0.002163	0.870567	0.022740	0.009534	0.898163	0.003381	-0.001418
0.000000	0.900000	0.000000	-0.015675	0.913269	0.002601	0.093312	0.793424	0.064106	-0.007796	0.088153	0.860363	-0.007206	0.010845	0.895921
0.000000	0.900000	0.000000	-0.012753	0.914790	-0.003160	-0.005672	0.905719	-0.001980	-0.004634	0.917493	-0.016664	-0.049032	0.949077	-0.001726
0.000000	0.900000	0.000000	0.089214	0.806615	0.004674	0.011073	0.890203	0.000637	0.009064	0.888170	0.004816	0.213736	0.687695	-0.001225
0.000000	0.000000	0.000000	0.018515	0.006913	0.870801	-0.013487	0.045392	0.864855	0.007298	0.007651	0.885836	0.003665	0.018879	0.874617
0.900000	0.000000	0.000000	0.949562	-0.049435	-0.024893	0.949790	-0.049794	-0.007100	0.904117	-0.014624	0.002011	0.913741	-0.013051	-0.000870
0.000000	0.000000	0.900000	0.018723	0.011861	0.864282	0.046025	0.018822	0.841462	0.004099	0.028895	0.858221	0.061997	-0.029010	0.900694
0.900000	0.000000	0.000000	0.886442	0.004282	0.004830	0.891358	0.008626	0.002033	0.901225	-0.001983	0.001290	0.897088	0.004253	-0.001142
0.000000	0.000000	0.900000	0.001874	-0.001588	0.899242	0.000532	0.013112	0.887250	-0.001364	0.003263	0.899742	0.002013	0.000304	0.897276
0.000000	0.900000	0.000000	0.068082	0.128071	0.589599	-0.029157	0.253767	0.713938	0.001088	0.013277	0.894386	0.466241	-0.047424	0.898075
0.000000	0.000000	0.900000	0.001743	-0.004631	0.902576	0.000721	-0.003136	0.902565	-0.001022	0.000727	0.901101	-0.000939	0.000197	0.900459
0.000000	0.000000	0.900000	0.000787	-0.003830	0.902687	0.000631	-0.003112	0.902607	-0.001024	0.000741	0.901103	-0.000709	-0.000333	0.900476
0.000000	0.900000	0.000000	-0.013275	0.915304	-0.003277	-0.005884	0.906497	-0.002343	-0.000505	0.898626	-0.003212	-0.043971	0.943818	0.001114
0.000000	0.900000	0.000000	-0.012285	0.914257	-0.002957	-0.005880	0.906430	-0.002282	-0.001084	0.900139	-0.003742	-0.042545	0.942358	0.001034
0.000000	0.900000	0.000000	0.025794	0.874018	0.001709	0.013461	0.887870	0.001516	0.011232	0.885273	0.005638	0.052514	0.849396	-0.001756
0.000000	0.900000	0.000000	-0.009561	0.907747	0.001317	-0.007976	0.896865	0.009800	0.014035	0.759536	0.075811	0.000299	0.707634	0.187089
0.900000	0.000000	0.000000	-0.007698	0.158546	0.767334	0.010185	0.292809	0.594760	-0.014401	0.235307	0.787920	0.304138	0.395549	0.054510
0.000000	0.900000	0.000000	-0.006550	0.904651	0.001234	-0.008053	0.897509	0.009124	0.015772	0.742525	0.082954	-0.000413	0.763900	0.134200

Nº Ex.	Acertos	%
2	5	27,78
4	10	55,56
6	13	72,22
8	14	77,78
10	16	88,89
12	14	77,78
14	12	66,67
16	14	77,78
18	15	83,33
20	14	77,78
22	14	77,78
24	15	83,33
26	15	83,33
28	15	83,33
30	15	83,33
32	14	77,78



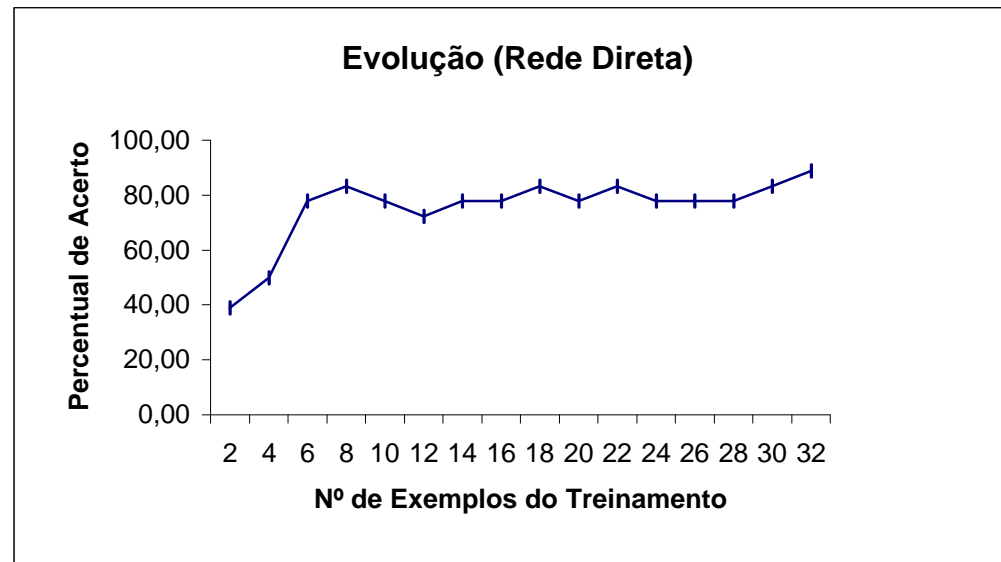
Saídas Experadas			02 Exemplos			04 Exemplos			06 Exemplos			08 Exemplos		
0.900000	0.000000	0.000000	0.921716	0.529010	0.005971	0.937685	0.019571	0.007110	0.865744	0.033636	0.000452	0.900046	-0.000021	-0.000000
0.000000	0.900000	0.000000	0.010262	0.054353	0.902842	-0.047483	0.793782	0.875900	-0.033367	0.848283	0.210660	0.000845	0.823619	0.041877
0.000000	0.900000	0.000000	0.923605	0.297649	-0.011641	0.220071	0.899153	-0.015726	-0.002158	0.902125	0.000042	-0.007836	0.902607	-0.000151
0.000000	0.900000	0.000000	0.906952	0.023386	-0.004754	0.900429	0.001968	-0.000044	0.008351	0.891574	0.000078	0.001108	0.899687	0.000020
0.000000	0.000000	0.000000	0.885842	-0.018425	0.012223	0.829079	-0.005237	0.083411	0.674821	0.219349	0.001449	0.308138	0.441516	-0.000031
0.900000	0.000000	0.000000	0.908776	0.033350	-0.005778	0.901371	0.000233	-0.000959	0.900795	-0.000870	0.000090	0.900068	-0.000031	-0.000000
0.000000	0.000000	0.900000	0.213692	0.000687	0.692093	-0.046155	0.601609	0.854557	-0.011893	0.391970	0.576081	0.000229	0.082027	0.756740
0.900000	0.000000	0.000000	0.922432	0.120251	-0.018668	0.903253	0.002640	-0.000880	0.900685	-0.000831	0.000175	0.900068	-0.000031	-0.000000
0.000000	0.000000	0.900000	0.888816	-0.010064	0.011391	-0.024984	0.867824	0.107894	0.265891	0.629520	0.000443	0.815957	0.040059	0.003235
0.000000	0.900000	0.000000	0.821708	-0.013122	0.077570	0.148484	-0.021740	0.834086	-0.033545	0.398209	0.744650	-0.022452	0.882214	0.000080
0.000000	0.000000	0.900000	0.079030	-0.008574	0.815988	0.046165	-0.025239	0.901634	-0.045764	0.347339	0.899350	-0.003023	0.000738	0.899013
0.000000	0.000000	0.900000	0.400781	0.021057	0.531965	-0.048006	0.797914	0.837469	-0.048591	0.717783	0.868634	-0.024380	0.006311	0.898141
0.000000	0.900000	0.000000	0.919905	0.118559	-0.014609	0.904926	0.006731	0.000814	-0.000892	0.900842	0.000061	0.001073	0.899698	0.000019
0.000000	0.900000	0.000000	0.919941	0.119778	-0.014582	0.905771	0.006264	0.000969	-0.000329	0.900277	0.000062	0.001064	0.899701	0.000019
0.000000	0.900000	0.000000	0.909015	0.030943	-0.006545	0.898729	0.004168	-0.000313	0.000761	0.899184	0.000064	0.001107	0.899688	0.000020
0.000000	0.900000	0.000000	0.918111	0.082620	-0.014550	0.057542	0.886487	-0.019813	-0.003687	0.903646	0.000053	-0.028299	0.911715	-0.000710
0.900000	0.000000	0.000000	0.907197	0.019691	-0.005840	0.872325	0.024624	0.005543	0.094100	0.805532	0.000168	0.001098	0.899689	0.000020
0.000000	0.900000	0.000000	0.916591	0.071068	-0.013214	0.083880	0.866995	-0.018802	-0.003613	0.903573	0.000052	-0.027859	0.911461	-0.000694

Saídas Experadas			10 Exemplos			12 Exemplos			14 Exemplos			16 Exemplos		
0.900000	0.000000	0.000000	0.766972	0.938875	-0.049867	0.484351	0.555458	-0.043216	0.190421	0.370159	-0.015052	0.626262	0.274026	0.000031
0.000000	0.900000	0.000000	-0.001558	0.899975	0.001630	-0.000715	0.901202	-0.001134	-0.001370	0.902490	0.000006	-0.044203	0.675135	0.723099
0.000000	0.900000	0.000000	0.050410	0.911328	-0.031387	-0.000714	0.901201	-0.001135	0.002158	0.892361	-0.000688	-0.003639	0.903686	-0.000051
0.000000	0.900000	0.000000	0.022444	0.890588	-0.009174	0.038468	0.845339	0.045162	0.124348	0.674249	-0.001356	0.064946	0.834892	0.000111
0.000000	0.000000	0.000000	0.948124	-0.048818	0.027094	0.024326	-0.014481	0.887017	0.036408	-0.028937	0.889751	0.478353	-0.047548	0.900028
0.900000	0.000000	0.000000	0.948268	-0.048863	0.024242	0.901427	-0.001209	-0.000230	0.902121	-0.001622	-0.000332	0.906754	-0.006639	-0.000108
0.000000	0.000000	0.900000	-0.010607	0.015087	0.898547	0.149446	0.640548	0.224795	0.264447	0.635632	0.012465	-0.047380	0.461540	0.900041
0.900000	0.000000	0.000000	0.948275	-0.048867	0.024222	0.901456	-0.001161	-0.000281	0.883758	-0.008087	0.014235	0.906605	-0.006490	-0.000108
0.000000	0.000000	0.900000	-0.020372	0.038133	0.896907	0.115039	0.646776	0.229371	0.014869	0.008710	0.863786	-0.000242	0.000010	0.899845
0.000000	0.900000	0.000000	0.762640	0.938764	-0.049862	-0.006337	0.890879	0.009823	-0.011477	0.799567	0.073976	-0.000133	0.000027	0.899967
0.000000	0.000000	0.900000	0.042205	-0.025317	0.903435	0.000685	-0.001093	0.901031	0.001378	-0.001129	0.900276	-0.011914	0.014842	0.900119
0.000000	0.000000	0.900000	-0.001779	0.897611	0.004311	-0.043074	0.393436	0.549601	-0.051098	0.373291	0.863089	-0.045918	0.350841	0.900232
0.000000	0.900000	0.000000	0.001269	0.900951	-0.002170	-0.000715	0.901202	-0.001134	-0.001330	0.902377	-0.000002	-0.003431	0.903465	-0.000036
0.000000	0.900000	0.000000	0.004522	0.901993	-0.006002	-0.000715	0.901202	-0.001134	-0.001257	0.902167	-0.000017	-0.002600	0.902631	-0.000031
0.000000	0.900000	0.000000	0.002763	0.897083	0.000144	0.006364	0.891626	0.007597	0.008467	0.879062	-0.000735	0.011134	0.888814	0.000054
0.000000	0.900000	0.000000	0.025622	0.907252	-0.022075	-0.000734	0.901172	-0.001102	-0.002001	0.901550	0.000939	-0.003311	0.903227	0.000090
0.900000	0.000000	0.000000	0.256566	0.517438	0.033201	0.006945	0.890644	0.008565	-0.050630	0.386807	0.836571	-0.044231	0.430326	0.857060
0.000000	0.900000	0.000000	0.110027	0.917781	-0.040888	-0.000732	0.901175	-0.001105	-0.001792	0.901864	0.000626	-0.002609	0.902512	0.000103

Saídas Experadas			18 Exemplos			20 Exemplos			22 Exemplos			24 Exemplos		
0.900000	0.000000	0.000000	0.374940	0.176291	-0.008129	0.900020	0.000208	-0.000233	0.901132	-0.000932	-0.000155	0.900546	-0.000391	-0.000127
0.000000	0.900000	0.000000	-0.001558	0.901032	0.000008	-0.049755	0.949754	0.000249	-0.001359	0.901165	0.000152	-0.000197	-0.000948	0.901130
0.000000	0.900000	0.000000	-0.001515	0.901088	-0.000072	-0.000104	0.900110	-0.000007	-0.001382	0.901183	0.000155	-0.000812	0.902067	-0.001265
0.000000	0.900000	0.000000	0.092446	0.786173	-0.000959	0.181023	0.719236	-0.000066	0.021132	0.881005	-0.001254	0.027800	0.871018	0.000725
0.000000	0.000000	0.000000	0.344845	-0.048990	0.891075	0.878931	-0.049788	0.899889	0.280247	-0.044765	0.902339	0.900894	-0.049845	0.895999
0.900000	0.000000	0.000000	0.906551	-0.005667	0.000098	0.900071	0.000138	-0.000215	0.901137	-0.000936	-0.000156	0.900335	-0.000028	-0.000306
0.000000	0.000000	0.900000	0.003201	-0.004363	0.899466	-0.048389	0.948196	0.005686	0.877011	-0.049802	0.903376	0.016697	-0.012848	0.900069
0.900000	0.000000	0.000000	0.906543	-0.005667	0.000102	0.900105	-0.001472	0.001399	0.901136	-0.000935	-0.000155	0.900350	-0.006455	0.006958
0.000000	0.000000	0.900000	-0.000474	0.000651	0.900032	0.193863	-0.041462	0.899822	0.006856	-0.006613	0.900390	-0.000598	0.000452	0.900001
0.000000	0.900000	0.000000	-0.043806	0.094432	0.894136	0.006787	0.002071	0.890813	-0.001062	0.005429	0.895566	-0.000198	-0.000787	0.900971
0.000000	0.000000	0.900000	-0.000457	0.000628	0.900031	0.000940	-0.001089	0.900119	-0.001033	0.000777	0.900109	-0.001328	0.001248	0.899957
0.000000	0.000000	0.900000	-0.000508	0.000670	0.900030	-0.049675	0.844291	0.900315	-0.001068	0.000804	0.900116	-0.000197	-0.000958	0.901141
0.000000	0.900000	0.000000	-0.001514	0.901088	-0.000072	-0.003497	0.903500	-0.000004	-0.001352	0.901161	0.000149	-0.000811	0.901753	-0.000947
0.000000	0.900000	0.000000	-0.001514	0.901088	-0.000072	-0.000868	0.900873	-0.000006	-0.001332	0.901146	0.000144	-0.000811	0.901770	-0.000964
0.000000	0.900000	0.000000	0.013321	0.888420	0.000153	0.018287	0.881734	-0.000009	0.007764	0.893892	-0.001181	0.002877	0.895762	0.001282
0.000000	0.900000	0.000000	-0.001557	0.901037	0.000002	-0.000108	0.900093	0.000015	-0.001383	0.901181	0.000158	-0.000811	0.901708	-0.000901
0.900000	0.000000	0.000000	-0.000762	0.000812	0.900014	-0.038586	0.907313	0.118916	-0.001006	0.897467	0.003574	-0.000204	-0.000908	0.901096
0.000000	0.900000	0.000000	-0.001557	0.901036	0.000003	-0.000107	0.900082	0.000024	-0.001383	0.901181	0.000158	-0.000811	0.901717	-0.000911

Saídas Experadas			26 Exemplos			28 Exemplos			30 Exemplos			32 Exemplos		
0.900000	0.000000	0.000000	0.899354	-0.000621	-0.000211	0.883254	0.019334	-0.001565	0.901838	-0.000694	-0.000436	0.899901	0.003010	-0.000110
0.000000	0.900000	0.000000	-0.001512	0.878115	0.024293	0.000502	0.860645	0.038290	-0.001886	0.898095	0.003116	0.024288	0.868542	-0.000082
0.000000	0.900000	0.000000	-0.001275	0.901785	-0.000017	0.000928	0.899282	-0.000265	-0.001477	0.901255	-0.000314	0.005154	0.888871	-0.000081
0.000000	0.900000	0.000000	0.450152	0.440023	0.000066	0.001298	0.898915	-0.000267	0.004377	0.897992	-0.000022	0.005163	0.888862	-0.000081
0.000000	0.000000	0.000000	0.895662	-0.049859	0.900348	0.038879	-0.022626	0.899778	-0.000083	0.000128	0.899998	-0.049747	0.024536	0.900136
0.900000	0.000000	0.000000	0.900540	-0.000538	-0.000363	0.949738	-0.049730	-0.000421	0.901838	-0.000695	-0.000436	0.887498	0.022235	-0.000021
0.000000	0.000000	0.900000	0.001179	-0.000493	0.900008	0.949692	-0.049999	0.899763	0.768168	-0.049711	0.900097	0.011491	-0.049580	0.899217
0.900000	0.000000	0.000000	0.900523	-0.000539	-0.000360	0.883957	0.018604	-0.001563	0.901837	-0.000699	-0.000431	0.899903	0.003008	-0.000110
0.000000	0.000000	0.900000	-0.002917	-0.000331	0.900491	-0.002665	0.002143	0.899939	-0.000055	0.000134	0.899978	0.022371	-0.049686	0.900122
0.000000	0.900000	0.000000	0.000314	-0.000425	0.900099	-0.002665	0.002143	0.899939	-0.003299	0.001852	0.899911	0.010701	-0.049591	0.900193
0.000000	0.000000	0.900000	0.000748	-0.000066	0.900011	0.012757	-0.010706	0.899868	-0.000044	0.000109	0.899998	-0.049801	0.051221	0.900224
0.000000	0.000000	0.900000	-0.000842	-0.000179	0.900215	-0.002665	0.002144	0.899939	-0.000044	0.000109	0.899998	-0.049802	0.051621	0.900224
0.000000	0.900000	0.000000	-0.001095	0.901603	-0.000017	0.000884	0.899326	-0.000265	-0.001482	0.901237	-0.000294	0.005154	0.888871	-0.000081
0.000000	0.900000	0.000000	-0.000689	0.901192	-0.000016	0.000884	0.899326	-0.000265	-0.001482	0.901237	-0.000294	0.005154	0.888871	-0.000081
0.000000	0.900000	0.000000	0.025976	0.874299	-0.000007	0.001165	0.899047	-0.000266	0.004407	0.897980	-0.000026	0.005158	0.888868	-0.000081
0.000000	0.900000	0.000000	-0.001275	0.901759	0.000009	0.000594	0.871904	0.027042	-0.001556	0.900977	0.000008	-0.004659	0.876455	0.011617
0.900000	0.000000	0.000000	-0.004473	-0.000140	0.900399	-0.002617	0.002234	0.899801	-0.001951	0.001164	0.899894	0.899905	0.003006	-0.000110
0.000000	0.900000	0.000000	-0.001275	0.901760	0.000009	0.000648	0.877862	0.021097	-0.001559	0.900964	0.000023	-0.000468	0.882306	0.006067

Nº Ex.	Acertos	%
2	7	38,89
4	9	50,00
6	14	77,78
8	15	83,33
10	14	77,78
12	13	72,22
14	14	77,78
16	14	77,78
18	15	83,33
20	14	77,78
22	15	83,33
24	14	77,78
26	14	77,78
28	14	77,78
30	15	83,33
32	16	88,89



ANEXO 2

LISTAGEM DOS ARQUIVOS FONTES DOS PROGRAMAS GERADOS EM MATLAB PARA A REDE BAM

1. Programa que gerencia o funcionamento da BAM, chamando todas as funções necessárias para executar suas rotinas.

```
% PROGRAMA PRINCIPAL DA BAM

clear; % LIMPA AREA DE TRABALHO
% DECLARACAO DAS CONSTANTES DE INICIALIZACAO
ExpTrein = 32; % NUMERO DE EXEPLoS PARA TREINAMENTO
Sintomas = 17; % NUMERO DE SINTOMAS UTILIZADOS
Diagnost = 3; % NUMERO DE DIAGNOSTICOS A RECONHECER
ExpTeste = 18; % NUMERO DE EXEPLoS PARA TESTES
Ciclos = 30; % NUMERO DE CICLOS A GERAR

Treiname = converte('D:\MATLAB\RNA\TESE\DadTrein.txt');
MatTrein = geramat(Treiname,ExpTrein,(Sintomas+Diagnost));
Pesos = gerapeso(MatTrein,ExpTrein,(Sintomas+Diagnost));

Testes = converte('D:\MATLAB\RNA\TESE\DadTeste.txt');
MatTeste = geramat(Testes,ExpTeste,(Sintomas+Diagnost));

Entrada(1,Sintomas+Diagnost+ExpTrein) = 0;
Entrada(1,[ExpTrein+1:ExpTrein+Sintomas]) = MatTeste(4,[1:Sintomas]);

Resultados = procbam(Pesos,Entrada,Ciclos);

% MONTAGEM DO GRAFICO DOS RESULTADOS

hold on
plot(Resultados);
hold off
xlabel('Ciclos');
ylabel('Ativacao');
text(Ciclos,Resultados(Ciclos,1),' AR');
text(Ciclos,Resultados(Ciclos,2),' AG');
text(Ciclos,Resultados(Ciclos,3),' AS');
```

2. Função CONVERTE para transformar o arquivo com a base de teste e exemplos em um vetor com todos os dados do arquivo.

```
% FUNCAO PARA LER O ARQUIVO DE EXEMPLOS E/OU TESTES GERANDO UM VETOR
function elemento = converte(ARQUIVO);

% LE O ARQUIVO E GERA UM VETOR COM OS EXEMPLOS LIDOS (TODO ARQUIVO)
arq = fopen(ARQUIVO,'r');
[elemento,count] = fscanf(arq,'%f');
status = fclose(arq);
```

3. Função GERAMAT transforma o vetor gerado do arquivo da base de exemplos ou teste em uma matriz, onde cada linha corresponde a um caso clínico.

```
% FUNCAO PARA GERAR A MATRIZ DOS EXEMPLOS OU TESTES

function Matriz = GeraMat(CASOS,linhas,colunas);

% ROTINA PARA TRANSFORMAR O VETOR GERADO EM UMA MATRIZ, ONDE CADA LINHA
% CORRESPONDE A UM CASO DO ARQUIVO
ind=1;
for lin=1:linhas;
    for col=1:colunas;
        Matriz(lin,col) = CASOS(ind);
        ind = ind + 1;
    end;
end;
```

4. Função GERAPESO monta a matriz de pesos da rede simetricamente, deixando sua diagonal principal nula.

```
% ROTINA PARA MONTAR A MATRIZ DOS PESOS

function Pesos = GeraPeso(Matriz,linhas,colunas);

tam = linhas+colunas;           % DEFINE O ORDEM DA MATRIZ PESOS
Pesos = zeros(tam,tam);        % CRIA MATRIZ PESOS COM 0(ZERO)
lin = 1:linhas;                % VETOR COM O NUM.DE LINHAS DA MATRIZ
col = 1:colunas;               % VETOR COM O NUM.DE COLUNAS DA MATRIZ
Pesos(lin,col+linhas) = Matriz(lin,col); % INSERE NAS LINHAS OS
EXEMPLOS DA MATRIZ
Pesos(col+linhas,lin) = Matriz(lin,col)'; % INSERE NAS COLUNAS OS
EXEMPLOS DA MATRIZ
```

5. Função PROCBAM responsável pelo processamento da rede BAM. Recebe o vetor de entrada da rede e gera um vetor de saída multiplicando a entrada pela matriz de pesos, passando esse por uma função tangente hiperbólica. Repete pelo quantidade de ciclos. Também gera uma matriz com todos os vetores resultantes de cada ciclo de processamento para que possa ser gerado o gráfico de acompanhamento dos resultados.

```
% FUNCAO PARA FAZER O PROCESSAMENTO DA BAM

function Result = ProcBAM(MPesos,VEntrada,Ciclos);

Bias = 0.0
VEntrada = VEntrada';
fprintf('%6.3f ',VEntrada([50:52]));
for ind = 1 : Ciclos;
    VSaida = (tanh(MPesos * VEntrada)+Bias);
    VEntrada = VSaida;
    fprintf('\n %ld \n',ind);
    fprintf('%6.3f ',VSaida([50:52]));
    Result(ind,:) = VSaida([50:52]);
end;
```

Arquivo (DADTREIN.TXT) tipo texto com a base de dados de treinamento com 32 (trinta e dois) casos utilizada para a montagem da matriz de pesos. As 17 (dezesete) primeiras colunas correspondem aos sintomas e as 3 (três) ultimas aos diagnósticos.

```
0.9 0.9 0.9 0.5 0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 -0.9 0.4 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.8 0.8 0.6 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.8 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 -0.9 0.9 0.9 0.9 0.4 0.6 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
0.9 0.9 0.9 0.5 0.9 0.4 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.8 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.6 -0.9 0.2 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.8 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.4 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.6 -0.9 0.4 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.0 0.9 0.0
0.9 0.9 -0.9 0.9 0.9 -0.9 0.4 0.2 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
0.9 -0.9 0.9 0.9 -0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 0.9 -0.9 0.9 -0.9 0.2 0.6 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 0.4 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.6 0.6 0.4 0.9 -0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
0.9 0.9 -0.9 0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.4 -0.9 0.9 -0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 -0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.4 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 0.9 -0.9 0.5 0.9 0.2 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.5 -0.9 0.6 0.4 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 0.9 -0.9 0.5 -0.9 0.6 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
```

Arquivo (DADTESTE.TXT) tipo texto com a base de dados de testes com 18 (dezoito) casos utilizada como entrada da rede. As 17 (dezesete) primeiras colunas correspondem aos sintomas e as 3 (três) ultimas aos diagnósticos.

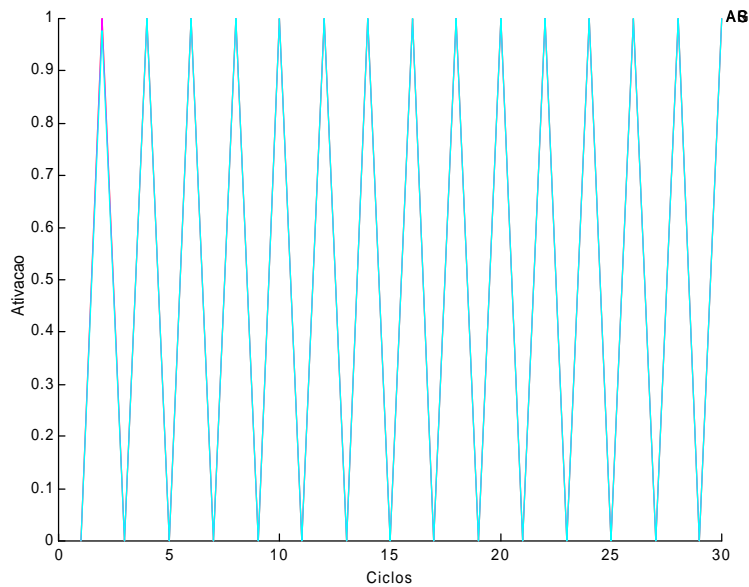
```
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 -0.9 0.9 0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
0.9 0.9 -0.9 0.5 0.9 0.6 0.8 0.6 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.8 -0.9 0.4 0.9 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.4 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.0 0.0 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.4 -0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
0.9 0.9 0.9 0.5 -0.9 0.4 0.8 0.4 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 0.9 0.0 0.0 0.9
-0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 -0.9 -0.9 0.9 0.9 0.9 -0.9 -0.9 -0.9 0.9 0.0 0.0
0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.4 0.4 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
0.9 0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 0.9 0.9 0.9 -0.9 -0.9 0.6 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
0.9 -0.9 -0.9 0.9 0.9 0.4 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 -0.9 -0.9 0.9 -0.9 0.0 0.0 0.9
```

```

-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.6 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.2 0.4 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.5 -0.9 0.4 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.6 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.0 0.9 0.0
0.9 -0.9 -0.9 0.9 -0.9 0.2 0.6 -0.9 0.9 -0.9 -0.9 -0.9 0.9 0.9 -0.9 -0.9 0.9 0.0 0.0
-0.9 -0.9 -0.9 0.9 -0.9 0.4 0.6 -0.9 -0.9 0.9 -0.9 -0.9 -0.9 0.9 -0.9 -0.9 0.0 0.9 0.0

```

Gráfico gerado pela rede com 30 (trinta) ciclos, apresentando claramente os problemas de instabilidade. Os resultados de todos os diagnósticos se sobrepõem e oscilam entre 0 e 1.



Os valores próprios da matriz de pesos (52x52) gerada com os dados acima apresentados, gerados pelo MatLab são:

```

» eig(Pesos)
0.0000    0.0000    0.8729    2.6314   -4.5810
0.0000   -0.6709    0.9683    2.5921    4.5810
0.0000   -0.8729    1.2257    2.8397   -5.8766
0.0000   -0.9683    1.4396   -3.4523    5.8766
0.0000   -0.3370    1.7358   -3.3291  -10.0890
0.0000   -1.2257    1.8074    3.4523   10.0890
0.0000   -1.4396   -2.1927    3.3291  -13.1267
0.0000   -1.7358    2.1927   -3.9934   13.1267
0.0000   -1.8074   -2.6314   -4.1208
0.0000    0.3370   -2.5921    3.9934
0.0000    0.6709   -2.8397    4.1208

```

Bibliografia

- [1] ALMEIDA, M. A. F. Aprender atividade inteligente: e se esta inteligência for parcialmente artificial. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Computação, UFSC, Florianópolis, Santa Catarina, setembro 1999.
- [2] AO DA SILVA DIAS, J. *Sensibilidade Paramétrica como Guia para o Treinamento Híbrido de Redes Neurais*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1999.
- [3] AO DE OLIVEIRA, R. C. L. *Modelo de Redes Neurais Artificiais com Dinâmica Interna em Sistemas de Controle*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, março 1999.
- [4] ASHBY, W. R. *Design for a Brain*. Wiley, New York, 1952.
- [5] BARRETO, J. M. Uma definição de otimização de sistemas. *Rev.Circ.Eng.Mil.* (1976), 76:S179–183.
- [6] BARRETO, J. M. *Redes neurais artificiais, fundamentos, aplicações e implementações*. Florianópolis, 1996.
- [7] BARRETO, J. M. *Inteligência Artificial no Limiar do Século XXI*, primeira ed. Duplic, Florianópolis, 1997.

- [8] BARRETO, J. M. & DE AZEVEDO, F. M. Connectionist expert systems as medical decision aid. *Artificial Intelligence in Medicine* 5 (1993), 515–523.
- [9] BARRETO, J. M.; DE AZEVEDO, F. M.; ZANCHIN, C. I. & EPPRECHT, L. R. Associative memories in medical diagnostics. In *In K. P. Adlassnig, G. Grabner, S. Engtsson, R. Hansen, editor, Medical Informatics Europe 91*, Berlin, August 1991, p. 348–352.
- [10] BLOCK, H. D. The perceptron: A model for brain functioning. *Reviews of Modern Physics* (1962), 34(1):123–135.
- [11] BOOLE, G. *The Laws of Thought*. McMillan, Reprinted in 1958 by Dover, 1854.
- [12] BRASIL, L. M. Aquisição de conhecimento aplicada ao diagnóstico de epilepsia. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, maio 1994.
- [13] BRASIL, L. M. *Proposta de Arquitetura para Sistema Especialista Híbrido e a Correspondente Metodologia de Aquisição de Conhecimento*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1999.
- [14] BRONZINO, J. D. *The Biomedical Engineering Hand Book*. CRC Press, IEEE PRESS, USA, 1995.
- [15] CARROL, L. *Symbolic Logic*. MacMillan and Co., reimpresso pela Dover em 1985, Londres, 1958.
- [16] CHARNIAC, E. & MCDERMOTT, D. *Introduction to Artificial Intelligence*. Addison-Wesley, Massachusetts, 1985.
- [17] COHEN, M. & GROSSBERG, I. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. In *IEEE Transactions on Systems, Man and Cybernetics*. IEEE PRESS, 1983, p. 815–826.

- [18] CRICK, F. The recent excitement about neural networks. *Nature* (Janeiro 1989), 337(12):129–132.
- [19] DA ROCHA FERNANDES, A. M. Sistema especialista difuso de aplicado ao processo de análise química qualitativa de amostras de minerais. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Computação, UFSC, Florianópolis, Santa Catarina, fevereiro 1996.
- [20] DE AZEVEDO, F. M. *Contribution to the Study of Neural Networks in Dynamical Expert Systems*. Ph.d. thesis, Institut d'Informatique, FUNDP, Namur, Bélgica, 1993.
- [21] DE AZEVEDO, F. M. Estudo do aprendizado de redes conexionistas iac considerando pool de casos reais. *III Congresso Brasileiro de Redes Neurais* (julho 1997), 238–243.
- [22] DE AZEVEDO, F. M.; BARRETO, J. M.; EPPRECHT, E. K.; EPPRECHT, L. R. & DE LIMA, W. C. Two approaches in case-based connectionist expert systems. In *ANNI'91 Artificial Intelligence and Neural Networks*, Calgary, Anahaim, julho 1991, p. 13–17.
- [23] DE AZEVEDO, F. M.; BARRETO, J. M. & LIMA, W. C. Teaching medical diagnosis with associative memories. In *In Proceedings on XI Systems Science International Conference*, Wroclaw, Poland, setembro 1992, p. 28–29.
- [24] DE B. CARVALHO FILHO, E. C.; LUDERMIR, T. B. & DE OLIVEIRA JUNIOR, W. R. Modelagem, aplicações de redes neurais. *I Congresso Brasileiro de Redes Neurais*, Itajubá - MG, Outubro 1994.
- [25] DE OLIVEIRA, J. P. Redes neurais artificiais para representação de conhecimento médico em anesthesiologia. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1997.

- [26] FAUSETT, L. *Fundamental of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall International Inc, New Jersey, 1994.
- [27] FEIGENBAUM, E. A. *The Fifth Generation*. Addison-Wesley, 1983.
- [28] FERNANDES, A. P. S. Sistema especialista difuso de apoio ao aprendizado do traumatismo dento-alveolar utilizando recursos multimídia. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, janeiro 1997.
- [29] FOGEL, D. B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, 1995.
- [30] FOGEL, D. B. & PORTO, V. W. Evolutionary programming for training neural networks. In *Improceedings of the International Joint Conference on Neural Networks*, June 1990, p. 601–605.
- [31] FOGEL, D. B. & PORTO, V. W. Evolutionary methods for training neural networks. In *Improceedings of Conference on Neural Networks for Occan Engineering*, Washington, 1991, p. 317–328.
- [32] FU, L. *Neural Networks in Computer Intelligence*. McGraw-Hill, New York, 1994.
- [33] FUKUSHIMA, K. Cognitron: A self-organising multilayered neural networks. *Biol. Cubernet.* (1975).
- [34] GALLANT, S. I. Conexionist expert systems. *Communication of the ACM* (1988), 152–169.
- [35] GALLANT, S. I. *Neural Network Learning and Expert Systems*. The MIT Press, 1993.
- [36] GIARRATANO, J. C. & RILEY, G. *Expert Systems - Principles and Programming*. PWS Publishing Company, Boston, 1994.

- [37] GROUP, T. P. *Using NeuralWorks - NeuralWorks Professional II/Plus and NeuralWorks Explorer*. NeuralWare, 1993.
- [38] HEBB, D. *Psicologia*. Livraria Ateneu, Rio de Janeiro, 1979.
- [39] HEBB, D. O. *The Organization of Behavior*. Wiley, New York, 1949.
- [40] HERGENHAHN, B. R. *An Introduction to Theories of Learning*. Prentice-Hall, Cambridge, Massachusetts, 1982.
- [41] HINTON, G. E. *Connectionist learning procedures*. Artificial Intelligence, 1989.
- [42] HOLLAND, J. H. *Adaptation in natural and artificial systems*. The University of Michigan Press, 1981.
- [43] HOLLAND, J. H. Genetic algorithms. *Scientific American* (1992), 267(1):66–72.
- [44] HUGO, M. Uma interface de reconhecimento de voz para o sistema de gerenciamento de central de informação de fretes. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, setembro 1995.
- [45] J., L.; FOGEL, A. J. O. & WALSH, M. J. *Artificial Intelligence Through Simulated Evolution*. John Wiley, 1966.
- [46] KANDEL, A. *Fuzzy Expert Systems*. CRC Press, Florida, USA, 1992.
- [47] KOCH, C. & SEGEV, I. *Methods in Neuronal Modeling: From Synapses to Networks*. The MIT Press, Cambridge, Massachusetts, 1989.
- [48] KOEHLER, C. Uma abordagem probabilística para sistemas especialistas. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Computação, UFSC, Florianópolis, Santa Catarina, novembro 1998.
- [49] KOSKO, B. *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, New Jersey, 1992.

- [50] KOVÁCS, Z. L. *Redes Neurais Artificiais - Fundamentos e Aplicações*. Edição Acadêmica São Paulo, São Paulo, 1996.
- [51] LEMOS, D. A utilização de sistemas especialistas para o diagnóstico do uso do solo e seus limites de ocupação. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, maio 1996.
- [52] LEVINE, R. I. & ET AL. *Inteligência Artificial e Sistemas Especialistas*. McGraw Hill, São Paulo, 1988.
- [53] LIMA, W. C. & BARRETO, J. M. Inteligência artificial. *Ciência Hoje* (1987), 7(38):50–56.
- [54] LINARES, K. S. C. Sistema especialista nebuloso para diagnóstico médico. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1997.
- [55] MATHWORKS. *MatLab User's Guide*. The MathWorks Inc, 1994.
- [56] MCCLELLAND, J. L.; RUMELHART, D. E. & PDP GROUP. *Parallel Distributed Processing, vol 3: Exercices*. The MIT Press, 1986.
- [57] MCCULLOCH, W. S. & PITTS, W. H. A logical calculus of ideas immanent in nervous activity. *Bull. of Mathematical Biophysics* (1943).
- [58] MIN, L. S. Sistema baseado em conhecimento para detecção e classificação de crises epilépticas. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Medicina Interna do Hospital Universitário, UFSC, Florianópolis, Santa Catarina, novembro 1994.
- [59] MINSKY, M. & SELFRIDGE, O. G. Learning in neural nets. In *Proceedings of Fourth London Symposium on Information Theory*, New York, 1961, Academic Press.

- [60] MINSKY, M. L. & PAPERT, S. A. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, Massachusetts, 1969.
- [61] MOISÉS, H. N. & SANTOS, T. H. F. *Santos Biologia*. Nova Cultural, São Paulo, 1993.
- [62] MONTELLO, M. V. Sistema especialista para predições e complicações cardiovasculares integrado a um sistema de controle de pacientes portadores de diabetes mellitus. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1999.
- [63] NASSAR, S. M. *Sistema Estatístico Inteligente para Apoio a Pesquisas Médicas*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, dezembro 1995.
- [64] NELSON, M. M. & ILLINGWORTH, W. T. *A Practical Guide to Neural Nets*. Addison-Wesley Publishing Company, Inc, Massachusetts, 1991.
- [65] NIEVOLA, J. C. Sistema especialista para auxílio ao diagnóstico médico de icterícia. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, UFSC, Florianópolis, Santa Catarina, abril 1988.
- [66] NIEVOLA, J. C. *Sistema Inteligente para auxílio no Ensino em Traumatologia Crânio-Encefálica*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica,, UFSC, Florianópolis, Santa Catarina, julho 1995.
- [67] OJEDA, R. G. *Técnicas de Inteligência Artificial Aplicadas ao Apoio á Decisão Médica na Especialidade de Anestesiologia*. Tese de doutorado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica,, UFSC, Florianópolis, Santa Catarina, julho 1992.

- [68] PANDYA, A. S. & MACY, R. B. *Pattern recognition with Neural Networks in C++*. CRC Press, Boca Raton, Florida, 1996.
- [69] PASSOLD, F. Sistema especialista híbrido em anesthesiologia para pacientes críticos/problemáticos. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, março 1995.
- [70] PELLEGRINI, G. F. Proposta de uma metodologia de avaliação sistema especialista na área médica. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, setembro 1995.
- [71] PEREIRA, C. G. Análise de crédito bancário: Um sistema especialista com técnicas difusas para os limites das agências. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, 1995.
- [72] PÉREZ, R. L. C. Sistema especialista de apoio à decisão médica com metodologia de aprendizagem simbólica. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, fevereiro 1997.
- [73] POZO, A. T. R. Sistema inteligente para treinamento do controle da infecção hospitalar. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, dezembro 1991.
- [74] RABUSKE, R. A. *Inteligência Artificial*. UFSC, Florianópolis, 1995.
- [75] RAITTZ, R. T. Free associative neurons - fan uma abordagem para reconhecimento de padrões. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, dezembro 1997.

- [76] RALL, W. *Cable Theory for Dendritic Neurons*. 1992.
- [77] RAMOS, R. F. Sistemas especialistas - uma abordagem baseada em objetos com prototipagem de um selecionador de processo de soldagem. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, fevereiro 1995.
- [78] RANDELL, B. *The Origins of Digital Computers: selected papers*. Springer-Verlag, 1975.
- [79] RÉ, A. M. D. Sistemas conexionistas adaptativos aplicados a problemas de controle de tráfego urbano. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Computação, UFSC, Florianópolis, Santa Catarina, março 1995.
- [80] RICH, E. & KNIGHT, K. *Inteligência Artificial*. Makron Books, São Paulo, 1993.
- [81] ROSEMBLATT, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, 1962.
- [82] RUMELHART, D. E.; G.E.HILTON & R.J.WILLIAMS. Learning internal representations by error propagation. In *Parallel Distributed Processing*, D. E. Rumelhart, J. L. McClelland, & PDP Group, Eds., vol. 1, Foundations. The MIT Press, Cambridge, Massachusetts, 1986, p. 318–362.
- [83] SCHEWFEL, H. P. Collective phenomena in evolutionary systems. In *31st Annu. Meet. Inter'l Soc. for General System Research*, Budapest, 1987, p. 1025–1033.
- [84] SHANNON, C. E. & (EDT.), J. M. *Automata Studies*. Princeton University Press, New Jersey, 1956.
- [85] SHEPHERD, G. M. *Microcircuits in the Nervous System*. Scientific American, vol.238, no.2, New York, February 1978.
- [86] SHEPHERD, G. M. *Neurobiology*. Oxford University Press, New York, 1983.

- [87] SHEPHERD, G. M. *The Significance of real neuron architectures for neural network simulation* Computation Neuroscience. The MIT Press, Cambridge, Massachusetts, 1990.
- [88] SIGAKI, N. A. Estudo da influência de base de casos em redes iac na implementação de sistemas especialistas. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Elétrica, área Engenharia Biomédica, UFSC, Florianópolis, Santa Catarina, 1997.
- [89] SOFTWARE, C. S. *Brain Maker - User's Guide and Reference Manual*. California Scientific Software, december 1990.
- [90] TAFNER, M. A. Reconhecimento de palavras faladas isoladas usando redes neurais artificiais. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, janeiro 1996.
- [91] TAFNER, M. A. & ET AL. *Redes Neurais Artificiais: Introdução e Princípios de neurocomputação*. Editora da FURB, Blumenau, 1995.
- [92] TEIVE, R. C. G. Planejamento da expansão da transmissão de sistemas de energia elétrica utilizando sistemas especialistas. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, março 1997.
- [93] TODESCO, J. L. Reconhecimento de padrões usando redes neurais artificiais com uma função de base radial: Uma aplicação na classificação de cromossomos humanos. Dissertação de mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, UFSC, Florianópolis, Santa Catarina, dezembro 1995.
- [94] WATERMAN, D. A. *A Guide to Expert System*. Addison-Wesley Publishing Company, USA, 1986.