

**UNIVERSIDADE DO VALE DO ITAJAÍ  
CENTRO DE CIÊNCIAS TECNOLÓGICAS DA TERRA E DO MAR  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**REDE NEURAL ARTIFICIAL PARA RECONHECIMENTO DE TABELA DE  
HORÁRIOS DE ARME/DESARME NO SISTEMA SIGMA**

Área de Inteligência Artificial

por

Daniel Falqueto

Anita Maria da Rocha Fernandes  
Orientadora

São José (SC), junho de 2007

**UNIVERSIDADE DO VALE DO ITAJAÍ**  
**CENTRO DE CIÊNCIAS TECNOLÓGICAS DA TERRA E DO MAR**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**REDE NEURAL ARTIFICIAL PARA RECONHECIMENTO DE TABELA DE  
HORÁRIOS DE ARME/DESARME NO SISTEMA SIGMA**

Área de Inteligência Artificial

por

Daniel Falqueto

Relatório apresentado à Banca Examinadora do  
Trabalho de Conclusão do Curso de Ciência da  
Computação para análise e aprovação.  
Orientadora: Anita Maria da Rocha Fernandes,  
Dra .

São José (SC), junho de 2007

# SUMÁRIO

LISTA DE ABREVIATURAS .....	iv
LISTA DE FIGURAS .....	v
LISTA DE EQUAÇÕES .....	vi
RESUMO .....	vii
ABSTRACT .....	viii
1 INTRODUÇÃO .....	1
1.1 PROBLEMATIZAÇÃO.....	1
1.1.1 Formulação do Problema .....	1
1.1.2 Solução Proposta .....	2
1.2 OBJETIVOS.....	2
1.2.1 Objetivo Geral .....	2
1.2.2 Objetivos Específicos .....	2
1.3 METODOLOGIA .....	3
1.4 ESTRUTURA DO TRABALHO.....	3
2 FUNDAMENTAÇÃO TEÓRICA .....	4
2.1 INTRODUÇÃO .....	4
2.2 REDE NEURAL “NATURAL”.....	4
2.3 REDE NEURAL ARTIFICIAL.....	6
2.4 HISTÓRICO.....	7
2.5 NEURÔNIO ARTIFICIAL: MODELO MCP .....	9
2.6 ARQUITETURA .....	10
2.7 TIPOS DE REDES NEURAIIS .....	13
2.7.1 Redes de Auto-Associação .....	13
2.7.2 Redes Neurais Recorrentes.....	14
2.7.3 Redes BSB.....	14
2.7.4 Redes de Contra-Propagação .....	15
2.7.5 Redes de Kohonem.....	15
2.7.6 Redes RBF.....	15
2.8 PERCEPTRON .....	16
2.9 REDES NEURAIIS MULTICAMADAS PERCEPTRON (MLP).....	18
2.10 BACKPROPAGATION.....	20
2.11 ALGORITMO DE APRENDIZADO .....	23
2.12 ETAPAS PARA O DESENVOLVIMENTO DE APLICAÇÕES RE REDES NEURAIIS.....	24
2.12.1 Coleta de dados e separação em conjuntos .....	24
2.12.2 Separação dos Dados.....	25
2.12.3 Configuração da rede .....	25
2.12.4 Treinamento .....	25
2.12.5 Teste .....	26

2.12.6 Integração .....	26
2.13 APLICAÇÕES DE REDES NEURAIS ARTIFICIAIS.....	27
3 DESENVOLVIMENTO .....	29
3.1 ANÁLISE DO SISTEMA SIGMA (DESCRIÇÃO DO SISTEMA).....	29
3.2 DEFINIÇÃO DO TAMANHO DA REDE.....	31
3.3 ETAPAS DE DESENVOLVIMENTO DA REDE.....	32
3.3.1 Coleta e Separação de Dados .....	32
3.3.2 Separação dos Dados em Conjuntos .....	34
3.3.3 Configuração da Rede .....	35
3.3.4 Treinamento .....	35
3.3.5 Testes.....	37
3.3.6 Testes numéricos e resultados obtidos .....	38
4 CONCLUSÕES.....	41
REFERÊNCIAS BIBLIOGRÁFICAS .....	42
ANEXOS.....	45

## LISTA DE ABREVIATURAS

ADALINE	ADaptative LINear Element
ART	Adaptative Resonance Theory
BAM	Bi-directional Associative Memory
BSB	Brain-State-In-a-Box
DM	Doença Meningocócica
LAM	Linear Associative Memories
LRGF	The Local-Recurrent-Global-Feedforward
LVQ	Learning Vector Quantization
MADALINE	Many ADALINE
MCP	McCulloch e Pitts
RBF	Radial Basis Function
MLP	Multi Layer Perceptron
RNA	Redes Neurais Artificiais
SNC	Sistema Nervoso Central

## LISTA DE FIGURAS

Figura 1: Neurônio Biológico .....	6
Figura 2: Neurônio Artificial.....	10
Figura 3: Modelos de Redes (Número de camadas) .....	12
Figura 4: Redes Neurais Artificial. ....	17
Figura 5: Descida do gradiente de uma superfície de erro .....	18
Figura 6: Gráfico da função Sigmoidal .....	19
Figura 7: Operação do algoritmo de backpropagation. ....	21
Figura 8: Backpropagation em modo progressivo (feed-forward).....	22
Figura 9: Treinamento supervisionado.....	24
Figura 10: Tabela de Horários.....	30
Figura 11: Rede Neural .....	32
Figura 12: Dados da Base.....	33
Figura 13: Exemplo de dados extraídos do banco.....	34
Figura 14: Erro quadrático para 150 épocas.....	37
Figura 15: Gráfico de Erro para Fator de Aprendizagem 0,25.....	39

## LISTA DE EQUAÇÕES

Equação 1 .....	10
Equação 2 .....	17
Equação 3 .....	19

## RESUMO

FALQUETO, Daniel. **Rede Neural Artificial para ajuste de tabela de horários de arme/desarme em sistemas de monitoramento de alarme**. São José, 2007. Trabalho de conclusão de curso (Graduação em Ciências da Computação) – Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2007.

Atualmente, um dos grandes problemas das empresas de monitoramento de alarmes é a grande quantidade de eventos (sinais emitidos pelas centrais de clientes) que não representam periculosidade. Dependendo do tipo de política de monitoramento da empresa esses eventos de menor importância podem estar desviando um operador do sistema de algum outro evento de maior importância. O sistema (SIGMA) analisado possui uma tabela de horários pré-configurada, onde o cliente pode definir o comportamento de sua central durante as 24 horas do dia e os 7 dias da semana. A partir do momento em que essa tabela é devidamente configurada e colocada em funcionamento, o sistema verifica de minuto em minuto se a central está se comportando conforme a tabela de horários e gera algum tipo de evento caso a central não esteja seguindo o padrão pré-configurado. A idéia de poder controlar os horários que a central deve estar Armada/Desarmada é excelente, porém a não tolerância de períodos de arme/desarme da central gera no sistema um contingente de eventos muito grande. A solução sugerida para minimizar esse contingente de eventos foi através das redes neurais artificiais, fazer com essas tabelas de horários dos clientes sejam re-configuradas estudando o comportamento dos eventos de ARME/DESARME durante um determinado tempo. A rede neural utilizada foi a MLP, com 48 neurônios na camada de entrada, 15 neurônios na camada escondida e 8 na camada de saída. A Rede Neural Artificial do tipo MLP (*Multi Layer Perceptron*) treinada apresentou uma taxa de 0,001 de erro.

**Palavras-chave:** Redes Neurais. Monitoramento de Alarmes. Horários.



## ABSTRACT

*Nowaday, one of the biggest problems of alarm monitoring companies is the longe quantity os events (signs emited by costumers centrals) that don't represent risk. Depending on the type of the company monitoring politcs, these events of less importance can be removing one of the system operators from same other event of a better importance. The system (SIGMA) to be analysed has a time table previously set, where the costumer can define the behavior of his central during the 24 hours a day, 7 days a week. From the moment the timetable is properly set and working, the system verifies every minute if the central is behaving according to the timetable and criate some kind of event if the central is not following the pattern previously set. The idea of controlling the time table that the central should be on/off is excelent, however the non-tolerance of periods on/off from the central makes a very big contingent of events. The solution suggested to minimize these contingent would be trough Artificial Neural Nets, or, make the costumers timetables reseted studying the behavior of on/off events during a determinated period of time. The neural network used was MLP, with 48 neurons on layer entrance, 15 neurons on hide entrance and 8 neurons on out entrance. The trained network shows a 0.001 rate of error.*

**Keywords:** *Neural Network. alarm monitoring companies. Timetable.*

# **1 INTRODUÇÃO**

Nos últimos tempos, observa-se um grande crescimento do uso de computadores como ferramenta de auxílio nas mais diversas áreas, sendo a área de segurança eletrônica apresenta vários campos para a utilização da informática. Inúmeras aplicações da informática podem ser encontradas na área de segurança citando-se os sistemas de monitoramento de alarmes como uma delas. Como exemplos de aplicações de segurança cita-se os softwares de monitoramento de alarmes Sigma (SEGWARE DO BRASIL, 2006), Condor (FOR SEGURANÇA ELETRÔNICA, 2006) e Iris (W. SECURITY, 2006).

Esses sistemas têm como objetivo auxiliar o profissional da área de segurança na tomada de decisões que normalmente envolve uma grande quantidade de dados e informações, tais como, eventos gerados pelo sistema (armado fora do horário, desarmado fora do horário, auto teste não recebido, etc...), que podem ser armazenadas e processadas muito mais rapidamente por computadores.

O presente trabalho teve como objetivo principal o desenvolvimento de uma Rede Neural Artificial para reconhecimento de padrões de tabelas de horários para eventos de ARME/DESAME no sistema de monitoramento de alarmes.

Para o desenvolvimento deste trabalho foi usado como base o sistema SIGMA desenvolvido pela empresa Segware do Brasil.

## **1.1 PROBLEMATIZAÇÃO**

### **1.1.1 Formulação do Problema**

Atualmente, um dos grandes problemas das empresas de monitoramento de alarmes é a grande quantidade de eventos (sinais emitidos pelas centrais de clientes) que não representam periculosidade. Cita-se como exemplo uma empresa que possui 20.000 (vinte mil) contas monitoradas. Se metade dos seus clientes não desarmarem seus alarmes nos horários pré configurados a empresa de monitoramento receberá 10.000 (dez mil eventos) a cada 30(trinta) minutos.

Para cada evento (sinal emitido pela central) cadastrado no sistema Sigma é definido uma prioridade. Atualmente o sistema trabalha com 3 (três) níveis de prioridade, alta (eventos em vermelho) média ( eventos em verde ) e baixa prioridade ( eventos em azul).

Dependendo do tipo de política de monitoramento da empresa esses eventos de menor prioridade podem estar desviando um operador do sistema de algum outro evento de maior importância.

O sistema (SIGMA) analisado possui uma tabela de horários pré-configurada, onde o cliente pode definir o comportamento de sua central durante as 24h do dia e os 7 dias da semana. A partir do momento em que essa tabela esteja devidamente configurada e colocada em funcionamento o sistema verifica de minuto a minuto se a central está se comportando conforme a tabela de horários e gera algum tipo de evento caso a central não esteja seguindo o padrão pré-configurado. A idéia de poder controlar os horários que a central deve estar ARMADA/DESARMADA é excelente, porém a não tolerância de períodos de ARME/DESARME da central gera no sistema um contingente de eventos muito grande.

### **1.1.2 Solução Proposta**

A solução proposta é o desenvolvimento de uma Rede Neural Artificial para detecção de padrões de tabelas de horários.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo Geral**

Desenvolver uma Rede Neural Artificial de suporte as empresas de monitoramento de alarme que permita reconhecer padrões de comportamentos de eventos de ARME/DESARME com base no conceito de Redes Neurais Artificiais.

### **1.2.2 Objetivos Específicos**

São os objetivos específicos do trabalho:

- Estudar os conceitos e diferentes arquiteturas das RNAs;
- Definir a arquitetura da rede;
- Implementar a rede;

- Testar e validar a rede.

## 1.3 METODOLOGIA

Para efetivação deste trabalho, o mesmo foi dividido em quatro grandes atividades: (i) estudo teórico de redes neurais; (ii) análise do problema a ser resolvido; (iii) escolha da Rede Neural Artificial a ser implementada; (iv) implementação, teste e validação da rede.

No TCC I foi realizado o estudo da teoria referente a Redes Neurais Artificiais através de livros, artigos, teses e dissertações sobre o assunto. Em seguida fez-se a análise dos dados apresentados pelo sistema SIGMA para estabelecer como os conjuntos de treinamento e teste da rede serão organizados; bem como serão compostas as camadas de entrada e saída da rede. Finalmente estabeleceu-se que o algoritmo a ser utilizado pela rede seria o Algoritmo de *Backpropagation*.

No TCC II foi implementada a rede, bem como foram feitas as etapas de validação e testes, que levaram a arquitetura final da rede: 48 neurônios na camada de entrada, 10 neurônios na camada escondida e 8 na camada de saída.

## 1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos. O presente capítulo apresenta os seguintes itens: a introdução do trabalho, a contextualização, o problema, os objetivos, o escopo, os resultados esperados, a justificativa e os aspectos metodológicos.

O Capítulo 2 apresenta as Redes Neurais Artificiais, seus principais conceitos, arquitetura, topologia bem como seu histórico.

O Capítulo 3 apresenta o desenvolvimento do projeto proposto.

No Capítulo 4 tem-se as conclusões do trabalho.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 INTRODUÇÃO**

As Redes Neurais Artificiais (RNA) surgiram da vontade do homem de reproduzir a máquina mais perfeita que existe até hoje, o cérebro humano. Uma máquina com enorme capacidade de raciocínio, onde o menor número de dados possíveis é o suficiente para desenvolver uma resposta exata e rápida (SILVA, 2002).

Uma rede neural é um processador maciço, paralelamente distribuído e constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. (ibidem)

Uma RNA tem seu conhecimento adquirido a partir de processos de aprendizagem, portanto, sua função principal é adquirir conhecimento e, a partir deste momento, reconhecer qualquer computação que tenha semelhanças no seu processamento (SILVA, 2002).

Nas sessões seguintes serão apresentados conceitos sobre redes neurais “naturais” e Redes Neurais Artificiais, seguidas de um breve histórico dos acontecimentos ligados a redes neurais desde o seu princípio até os dias de hoje.

### **2.2 REDE NEURAL “NATURAL”**

O cérebro humano possui, pelo que se conhece, cerca de 10 bilhões de neurônios. São eles as células mais diferenciadas do organismo, pois apresentam a maior complexidade estrutural e funcional. Durante o período da vida embrionária acontecem as divisões de células nervosas necessárias à vida humana e, graças a esse fato, nenhuma outra célula procriará após o nascimento do bebê, permanecendo, assim, seu número, igual para o resto de sua vida, sofrendo apenas alterações de volume e prolongamento de conexões com outras células (alguns pesquisadores contemporâneos têm defendido que neurônios podem ser reproduzidos) (TAFNER, XEREZ e RODRIGUES FILHO, 1995).

Um neurônio é capaz de criar até 10.000 sinapses, ou seja, até 10.000 conexões com neurônios adjacentes. Considerando que cada conexão responde por um sim ou não, tipo um bit (sim/não), o número máximo de respostas sim/não que o cérebro pode conter é aproximadamente  $10^{10} \times 10^4 = 10^{14}$ , ou 100 trilhões. Para se ter uma idéia da quantidade de conexões, o número de grãos de areia de que a terra inteira poderia conter num volume na base de 10 grãos por  $\text{mm}^3$ , é igual a  $10^{31}$ . Em outras palavras, o cérebro possui uma capacidade de armazenamento

correspondente a cada grão de areia de meio planeta. Simbolicamente, dividindo a terra ao meio, cada metade é um cérebro humano (TAFNER, XEREZ e RODRIGUES FILHO, 1995).

Os neurônios são divididos em três seções: o corpo da célula, os dendritos e o axônio, cada um com funções específicas porém complementares. O corpo do neurônio mede apenas alguns milésimos de milímetros, e os dendritos apresentam poucos milímetros de comprimento. O axônio, contudo, pode ser mais longo e, em geral, tem calibre uniforme (BRAGA, CARVALHO e LUDERMIR, 2000).

Os dendritos têm como função receber as informações, ou impulsos nervosos, oriundas de outros neurônios e conduzi-las até o corpo celular. Aqui, a informação é processada, e novos impulsos são gerados. Estes estímulos são transmitidos a outros neurônios, passando do axônio até os dendritos dos neurônios seguintes. O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro é chamado de sinapse. É pelas sinapses que os nodos (neurônios) se ligam funcionalmente, formando redes neurais. As sinapses funcionam como válvulas, e são capazes de controlar a transmissão de impulsos (fluxo de informação) entre os nodos da rede neural. O efeito da sinapse é variável, e é esta variação que dá ao neurônio capacidade de adaptação. (BRAGA, CARVALHO e LUDERMIR, 2000).

O neurônio, basicamente capta o estímulo nos dendritos e o transmite pelo axônio para que atinja um outro neurônio ou outros tipos de células. A captação do estímulo é realizada pelos receptores sensoriais especializados. Existem diversos tipos desses receptores: mecânicos, térmicos, nociceptores, químicos etc. (TAFNER, XEREZ e RODRIGUES FILHO, 1995)

Os principais componentes dos neurônios segundo Tatibana e Kaetsu ( 200?) (Figura1) são:

- Os dendritos: que têm por função, receber os estímulos transmitidos pelos outros neurônios;
- O corpo do neurônio: também chamado de *somma*, que é responsável por coletar e combinar informações vindas de outros; e
- O axônio: que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outras células.

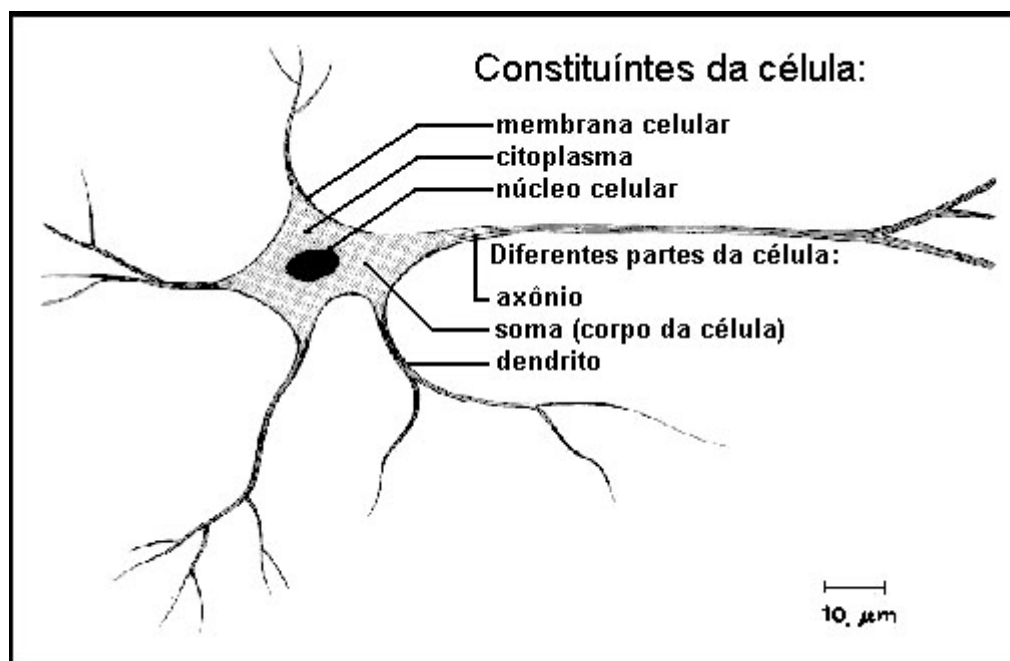


Figura 1: Neurônio Biológico;

Fonte: Tatibana e Kaetsu (200?)

## 2.3 REDE NEURAL ARTIFICIAL

Uma RNA é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades geralmente são conectadas por canais de comunicação que estão associados a determinado peso, e a um conjunto de treinamento. As unidades fazem operações apenas sobre seus dados locais, ou seja, em seu ambiente, que são entradas recebidas pelas conexões. O comportamento inteligente de uma rede neural artificial vem das interações entre as unidades de processamento da rede Muller e Fill (2003, apud FERNANDES, 2003).

As Redes Neurais Artificiais são sistemas físicos que podem adquirir, armazenar e utilizar conhecimentos experimentais, que podem alcançar uma boa performance, devido à sua densa interconexão entre os nós da rede. Elas também são conhecidas por: modelos conexistas, modelos de processamento paralelo distribuído e sistemas neuromorfológicos Lippmann (1997, apud FERNANDES, 2003),

Segundo Kohonen (1972, apud FERNANDES 2003), as Redes Neurais Artificiais são definidas como redes massivamente paralelas e interconectadas, de elementos simples, com organização hierárquica. Estes elementos devem interagir com objetivos do mundo real, da mesma maneira que o sistema nervoso biológico.

Bet (2005, apud FERNANDES 2003) afirma que as Redes Neurais Artificiais são representações computacionais do funcionamento do cérebro humano, sendo inspiradas em modelos

neurais biológicos, ou seja, são compostas por elementos que fazem funções parecidas a um neurônio biológico.

## 2.4 HISTÓRICO

As Redes Neurais Artificiais foram desenvolvidas, originalmente, na década de 40. O primeiro modelo artificial de um neurônio biológico foi fruto do trabalho do psiquiatra e neurofisiologia Warren McCulloch e do matemático Walter Pitts da Universidade de Illinois em 1943. No seu clássico artigo, McCulloch e Pitts descrevem um cálculo lógico das redes neurais que unifica os estudos de neurofisiologia e da lógica matemática. Eles assumiram que o seu modelo formal de um neurônio seguia uma lei “tudo ou nada”. Com um número suficiente dessas unidades simples (neurônios) e com conexões sinápticas ajustadas apropriadamente e operando de forma síncrona, McCulloch e Pitts mostraram que uma rede assim constituída realizaria, a princípio, a computação de qualquer função computável. Este era um resultado muito significativo e com ele é geralmente aceito o nascimento das disciplinas de Redes Neurais Artificiais e inteligência artificial (HAYKIN, 2001).

O próximo desenvolvimento significativo das Redes Artificiais veio em 1949, com a publicação do livro de Hebb *The Organization of Behavior*, no qual é apresentada uma formulação explícita de uma regra de aprendizagem filosófica para a modificação sináptica. Nesse livro Hebb propôs que a conectividade do cérebro é continuamente modificada conforme um organismo vai aprendendo tarefas funcionais diferentes e que agrupamentos neurais são criados por tais modificações. Hebb ainda deu seguimento a uma sugestão anterior de (Ramón y Cajál, 1911 apud HAYKYN, 2001) e apresentou o seu famoso postulado de aprendizagem, que afirma que a eficiência de uma sinapse variável entre dois neurônios é aumentada pela ativação repetida de um neurônio causada pelo outro neurônio, através daquela sinapse. (ibidem)

Nos anos 50, Von Neumann investigou a capacidade das Redes Neurais Artificiais corrigirem seus próprios erros através da utilização de informações redundantes (Fih, 1996 apud FERNANDES 2003). O interesse no mecanismo de aprendizagem foi iniciado por Hebb (Fat, 1995, apud FERNANDES, 2003), que propôs uma regra de aprendizagem que diz que, quando um estímulo de entrada influencia na produção do estímulo de saída, os pesos das conexões entre os neurônios deve ser incrementado. A regra de Hebb é utilizada em vários algoritmos de aprendizagem (FERNANDES, 2003).

Em 1958, Frank Rosenblatt demonstrou com o seu novo modelo, o *perceptron*, que se fossem acrescentadas de sinapses ajustáveis, as RNA's com nodos MCP (McCulloch e Pitts) poderiam



ser treinadas para classificar certos tipos de padrões. Rosenblatt descreveu uma topologia de RNA's, estrutura de ligação entre os nodos e, o mais importante, propôs um algoritmo para treinar a rede para executar determinados tipos de funções. O *perceptron* simples descrito por Rosenblatt possui três camadas: a primeira recebe as entradas do exterior e possui conexões fixas (retina); a segunda recebe impulsos da primeira através de conexões cuja eficiência de transmissão (peso) é ajustável e, por sua vez, envia saída para a terceira camada (resposta). Este tipo de perceptron comporta-se como um classificador de padrões, dividindo o espaço de entrada em regiões distintas para cada uma das classes existentes. O que Rosenblatt buscava era projetar RNA's que fossem capazes de fazer descobertas interessantes sem necessidade de regras (BRAGA, CARVALHO e LUDERMIR, 2000).

Também nos anos 60, Windrow e Hoff (1996, apud VALLE FILHO, 2003) desenvolveram o ADALAINE (ADAPtative LINEar Element) e o MADALAINE (Many ADALAINE) para o reconhecimento de padrões. O algoritmo de aprendizagem de Windrow é conhecido como Regra Delta (FERNANDES, 2003).

Em 1969, Minsky e Papert chamaram a atenção para algumas tarefas que o *perceptron* não era capaz de executar, já que este só resolve problemas lineares separáveis, ou seja, problemas cuja solução pode ser obtida dividindo-se o espaço de entrada em duas regiões através de uma reta. O PERCEPTRON, por exemplo, não consegue detectar paridade, conectividade e simetria, que são problemas não lineares separáveis. O principal argumento de Minsky e Papert era de que o problema do crescimento explosivo, tanto de espaço ocupado como do tempo requerido para solução de problemas complexos afetaria, cedo ou tarde as RNA's, inclusive os PERCEPTRONS (BRAGA, CARVALHO e LUDERMIR, 2000).

O renascimento dos interesses sobre as redes neurais artificiais aconteceu na década de 80, com vários trabalhos, tais como o de Hopfield. Ele elaborou um algoritmo que estabiliza uma rede binária simétrica com realimentação (Hopfield, 1982, apud FERNANDES, 2003). Uma rede de Hopfield é capaz de estabilizar um conjunto de padrões de treinamento e também de realizar auto-associação semelhante a uma memória endereçada por conteúdo (FERNANDES, 2003).

A deficiência das redes *perceptron* na resolução de problemas de associação de padrões para um conjunto de padrões não-lineares foi eliminada por Rumelhart, Hinton e Williams (Carvalho, 1996, apud, VALLE FILHO, 2003). A solução encontrada foi generalização da Regra Delta, conhecida como algoritmo de *Backpropagation* para redes MLP com entradas e saídas analógicas (FERNANDES, 2003).

Um outro tipo de rede neural surgiu com a possibilidade de realizar associação de padrões em memórias óticas associativas (Soffer, 1986, apud VALLE FILHO, 2003) mais tarde esse modelo veio a ser chamado de BAM (Bi-directional Associative Memory) por Kosko (1988, apud FERNANDES, 2003). Este tipo de rede possui um algoritmo de aprendizagem com a regra de Hebb.

As redes do tipo LAM (Linear Associative Memories) são redes *feedforward* de duas camadas que realizam um processo interpolativo de associação de padrões análogos, foram estudadas por Anderson, Kohonem e Nakano (Filho, 1996, apud FERNANDES, 2003).

Redes LVQ (Learning Vector Quantization) definem uma família de algoritmos de quantificação vetorial adaptativa, originalmente criada por Kohonen (HAYKIN, 2001), que podem ser utilizados para a classificação e reconhecimento de padrões, a partir de um conjunto de amostras de treinamento (DIVERIO *et al*, 2005).

As Redes Neurais Artificiais baseadas na Teoria da Ressonância Adaptativa (ART) foram desenvolvidas por Grossberg. Utilizam um sistema de auto-estabilização e auto-organização dos padrões de entrada através de um processo competitivo em uma arquitetura de duas camadas completamente conectadas (BRAGA, CARVALHO e LUDERMIR, 2000).

## 2.5 NEURÔNIO ARTIFICIAL: MODELO MCP

“O modelo foi inicialmente proposto por McCulloch e Pitts e trata-se de uma simplificação sobre o que se sabia à respeito do neurônio biológico naquela época. A sua descrição matemática resultou em um modelo com  $n$  terminais de entrada  $x_1, x_2, \dots, x_n$  (que representam os dendritos), e apenas um terminal de saída  $y$  (representando o axônio). Para emular o comportamento das sinapses os terminais de entrada do neurônio têm pesos acoplados  $w_1, w_2, \dots, w_n$ , cujos valores podem ser positivos ou negativos, dependendo das sinapses correspondentes serem inibitórias ou excitatórias. O efeito de uma sinapse particular  $i$  no neurônio pós-sináptico é dado por:  $x_i w_i$ . Os pesos determinam em que grau o neurônio deve considerar sinais de disparo que ocorrem naquela conexão” (BRAGA, CARVALHO e LUDERMIR, 2000).

“Um neurônio biológico dispara quando a soma dos impulsos que ele recebe ultrapassa o seu limiar de excitação (*threshold*). O corpo do neurônio, por sua vez, é emulado por um mecanismo simples que faz a soma dos valores  $x_i w_i$  recebidos pelo neurônio (soma ponderada), e decide se o neurônio deve ou não disparar (saída igual a 1 ou 0) comparando a soma obtida ao limiar ou *threshold* do neurônio. No modelo MCP (McCulloch e Pitts), a ativação do neurônio é obtida através da aplicação de uma função de ativação, que ativa a saída ou não dependendo do

valor da soma ponderada de suas entradas. Na descrição original do modelo MCP, a função de ativação é dada pela função de limiar conforme a Equação 1 e a Figura 2.

$$\sum_{i=1}^n x_i w_i \geq 0 \quad \text{Equação 1}$$

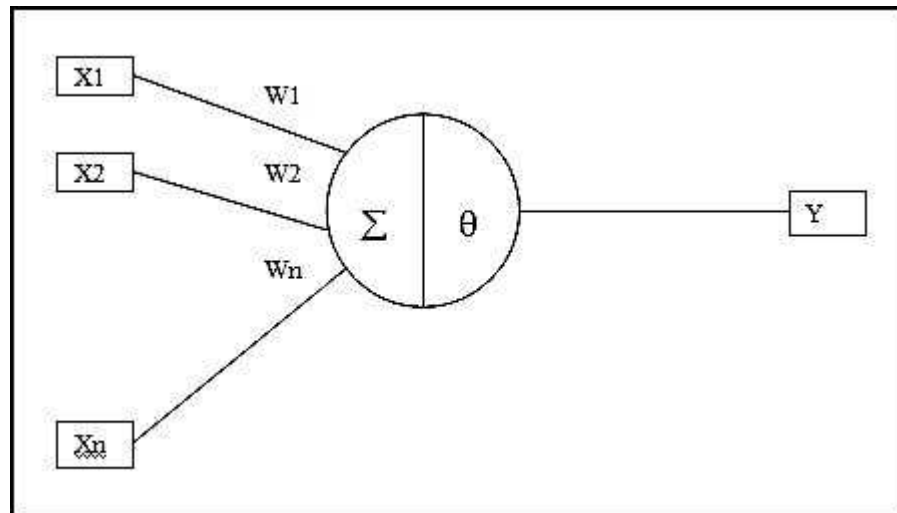


Figura 2: Neurônio Artificial;

Fonte: Haykin (2001)

Onde  $n$  é o número de entradas do neurônio,  $w_i$  é o peso associado à entrada  $x_i$ , e  $\theta$  é o limiar (*threshold*) do neurônio”.

“McCulloch e Pitts simplificaram seu modelo assumindo que os nodos em cada camada da rede disparam sincronamente, isto é, todos os nodos são avaliados ao mesmo tempo e também que as entradas em um instante de tempo  $t$  produzem a sua saída no tempo  $t+1$ . Em sistemas biológicos, sabe-se que não existe um mecanismo para sincronizar as ações dos nodos, nem há restrição para que as suas saídas sejam ativadas em tempos discretos como no modelo MCP. Sabe-se também que o valor da próxima saída dos nodos biológicos depende enormemente das ativações dos estados anteriores, já que até mesmo os neuro-transmissores liberados anteriormente levam algum tempo para se recombinarem, influenciando, assim, as ativações seguintes” (BRAGA, CARVALHO e LUDERMIR, 2000).

## 2.6 ARQUITETURA

A definição da arquitetura de uma RNA é um parâmetro importante na sua concepção, uma vez que ela restringe o tipo de problema que pode ser tratado pela rede (BRAGA, CARVALHO e LUDERMIR, 2000).

A maneira pela qual os neurônios de uma RNA estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede (HAYKIN, 2001).

Fazem parte da definição da arquitetura os seguintes parâmetros: número de camadas da rede, número de nodos em cada camada, tipo de conexão entre os nodos e topologia da rede (BRAGA, CARVALHO e LUDERMIR, 2000).

Número de camadas: As camadas intermediárias da rede são como detectores de características, as quais serão representadas, internamente, através dos pesos sinápticos.

Número de nodos em cada camada: Para a escolha do número de unidades em cada camada, deve-se considerar:

- O número de exemplos de treinamento;
- A quantidade de erro presente nos exemplos;
- A complexidade da função a ser aprendida pela rede; e
- A distribuição estatística dos dados de treinamento.

Ao determinar o número de neurônios em cada camada, os seguintes cuidados devem ser tomados:

- Não utilizar um número de unidades maior que o necessário: número alto de unidades pode fazer com que a rede memorize os dados do treinamento; com isto ela torna-se incapaz de generalizar e, portanto, reconhecer padrões não vistos durante o treinamento. Isto é chamado de *overfitting*.
- Não utilizar um número de unidades inferior ao número necessário: isto pode fazer com que a rede gaste muito tempo para aprender, podendo não alcançar os pesos adequados, ou seja, a rede pode não convergir, ou generalizar demais os padrões de entrada (BRAGA, CARVALHO e LUDERMIR, 2000).

Redes com uma única camada de nodos MCP, por exemplo, só conseguem resolver problemas lineares separáveis. Redes recorrentes, por sua vez, são mais apropriadas pra resolverem problemas que envolvem processamento temporal. (ibidem).

Em geral, identifica-se três classes de arquiteturas de redes fundamentalmente diferentes (HAYKIN, 2001); conforme Figura 3.

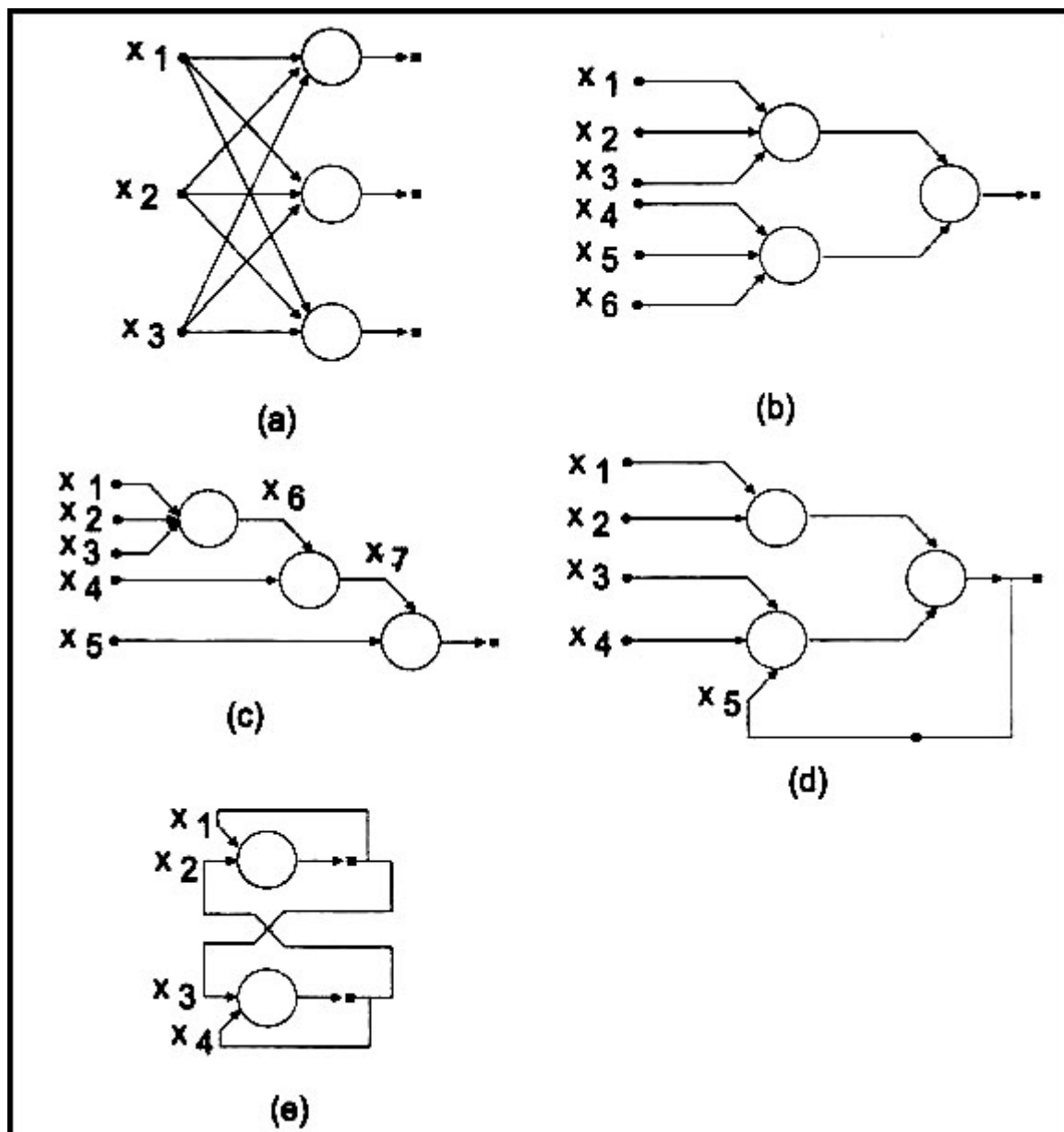


Figura 3: Modelos de Redes (Número de camadas)

Fonte: Braga, Carvalho, Ludemir (2000)

Quanto ao número de camadas, tem-se:

1. Redes de camada única – Só existe um nó entre qualquer entrada e qualquer saída da rede (Figura 3a, 3e) (BRAGA, CARVALHO e LUDERMIR, 2000);
2. Redes de múltiplas camadas – Existe mais de um neurônio entre alguma entrada e alguma saída da rede (Figura 3b, 3c, 3d) (ibidem);
3. Redes recorrentes – Uma rede neural recorrente se distingue de uma rede de múltiplas camadas por ter pelo menos um laço de realimentação (Figura 3e e 3d) (HAYKIN, 2001).

Os nodos podem ter conexões do tipo:

1. *feedforward*, ou acíclica – A saída de um neurônio na  $i$ -ésima camada da rede não pode ser usada como entrada de nodos em camadas de índice menor ou igual a  $i$  (Figura 3a, 3b, 3c);
2. *feedback*, ou cíclica – A saída de algum neurônio na  $i$ -ésima camada da rede é usada como entrada de nodos em camadas de índice menor ou igual a  $i$  (Figura 3d, 3e).

Redes cuja saída fina (única) é ligada às entradas comportam-se como autômatos reconhecedores de cadeias, onde a saída que é realimentada fornece o estado do autômato (Figura 3d, 3e)

Se todas as ligações são cíclicas, a rede é denominada auto-associativa. Estas redes associam um padrão de entrada com ele mesmo, e são particularmente úteis para recuperação ou “regeneração” de um padrão de entrada (Figura 3e).

Finalmente, as RNA's podem ser classificadas quanto a sua conectividade:

1. Rede fracamente (ou parcialmente) conectada, onde nem todas as entradas e saídas da rede são conectadas a todos os neurônios da próxima camada (Figura 3b, 3c, 3d);
2. Rede completamente conectada onde todos os neurônios se comunicam (Figura 3a, 3e);

## **2.7 TIPOS DE REDES NEURAIS**

A seguir serão apresentados alguns tipos de Redes Neurais Artificiais: redes de auto-associação; redes neurais recorrentes; redes BSB; redes de contra propagação; redes de Kohonem e redes RBF.

### **2.7.1 Redes de Auto-Associação**

Uma das principais características das RNA's é a capacidade de aprender através de exemplos, sem a necessidade de serem explicitamente programadas. Redes que utilizam paradigma de aprendizado supervisionado, em que um supervisor externo fornece informações sobre respostas desejadas para os padrões de entrada, limitam o seu campo de aplicações onde estes dados são conhecidos. Existem várias aplicações em que é necessário que a rede seja treinada sem um supervisor externo ou punição/recompensa. Nestes casos, a única informação fornecida está no conjunto de padrões de entrada. Desta forma, estas redes definem seus parâmetros por si próprias, sem auxílio externo. Esta é a idéia central dos modelos *self-organizing*, os quais utilizam o

paradigma de aprendizado não supervisionado. Estes algoritmos utilizam um conjunto de regras de natureza local para ajuste dos pesos (BRAGA, CARVALHO e LUDERMIR, 2000).

### 2.7.2 Redes Neurais Recorrentes

Redes neurais recorrentes são usualmente utilizadas para lidar com processos que produzem seus resultados baseados na entrada presente e nos estados passados de um dado processo dinâmico. Em 1991, Hertz apontou dois tipos de redes neurais recorrentes. Uma rede neural totalmente recorrente tem arquitetura na qual uma dada unidade de processamento pode realimentar qualquer outra unidade. Uma rede neural parcialmente recorrente é definida como uma rede formada por maioria de conexões de alimentação direta (*feedforward connections*) e conexões de realimentação (*feedback connections*). As ligações diretas são predominantes, contudo as realimentações são fundamentais para se lidar com seqüências temporais.

As redes com um tipo de arquitetura denominada recorrência local e alimentação direta global (*the local-recurrent-global-feedforward - LRGF - architectures*) constituem um caso particular das redes neurais parcialmente recorrentes. Neste caso, as conexões de realimentação ocorrem exclusivamente de uma unidade para ela mesma (TSOI e BACK, 1994). Estes modelos apresentam três diferentes tipos de realimentações. A realimentação local de ativação (*the local activation feedback*) permite combinações de versões atrasadas do somatório ponderado de cada unidade de processamento. A realimentação local de saída (*the local output feedback*) consiste da entrada em uma unidade de versões atrasadas de sua própria saída. A realimentação local de sinapse (*the local synapse feedback*) compreende uma estrutura de realimentação em cada sinapse (HERTZ, 1991).

### 2.7.3 Redes BSB

A rede *Brain-State-In-a-Box* (BSB), desenvolvida por Anderson, em 1977, é considerada atualmente uma precursora da rede de Hopfield mais simples, por demonstrar robustez superior, e com a qual mantém algumas similaridades conceituais (LOESCH, 1996).

Como a rede Hopfield, BSB efetua revocação auto-associativa dos vetores de entrada. Seus padrões de entrada são codificados com valores binários +1 e -1. Para ser efetiva, seus padrões de entrada devem ser ortogonais ou aproximadamente ortogonais. Sua capacidade de memória (o número de padrões ortogonais armazenáveis) é igual a dimensão da rede. Ao contrário da rede Hopfield, estados espúrios nesta rede são raros. (ibidem)

Em seu desenvolvimento, Anderson considerou como principal conceito o fato de que uma rede singular de interconexões pode possuir múltiplos estados estáveis. Considerou os possíveis valores extremos de atividade neural como sendo -1 e +1, idênticos aos dois únicos possíveis valores de entrada, porém admitindo todos os valores lineares intermediários de atividade durante o processo iterativo (LOESCH, 1996).

#### **2.7.4 Redes de Contra-Propagação**

A rede de Contra-propagação (*Counterpropagation*) foi proposta por Robert Nielsen em 1987. É muito utilizada em classificação de padrões, aproximação de funções, análise e aplicações que requerem compressão de dados (BARRETO, 1999).

A primeira camada funciona com o aprendizado competitivo e a segunda com aprendizado supervisionado. O mais interessante é que este paradigma permite obter valores das conexões sinápticas de modo algorítmico, sem iterações. (ibidem)

Um vetor de entrada é aplicado na camada de entrada, pré-processado e propagado para a camada intermediária. Na camada intermediária, cada neurônio calcula o seu net (produto escalar de um vetor de entrada pelo vetor de pesos) e compete com outros neurônios da mesma camada para ver quem tem o maior valor de net. Apenas a unidade vencedora é que manda o seu sinal para os neurônios da camada de saída (BARRETO, 1999).

#### **2.7.5 Redes de Kohonen**

O pioneiro no desenvolvimento da teoria das redes Competitivas foi Teuvo Kohonen, e por esta razão os neurônios de uma rede competitiva são muitas vezes chamados de neurônios de Kohonen. Do mesmo modo, as redes auto-organizáveis são também conhecidas como redes de Kohonen ou Mapas topológicos de características auto-organizativas, nos quais as características do objeto a ser mapeado é preservado.

A rede de Kohonen é uma estrutura de duas camadas de neurônios. A primeira camada é a de entradas e seus neurônios Kohonen (1987, apud FERNANDES, 2003).

#### **2.7.6 Redes RBF**

Redes RBF (*Radial Basis Function*) têm dupla origem histórica: como sistema neural de inspiração biológica e aprendizado rápido e como método de aproximação de funções. Do ponto de vista de inspiração biológica, baseia-se no modelo de campos sobrepostos de recepção, presentes em regiões do córtex cerebral ligadas a processamento de informações sensoriais. O caráter de



atividade local de seus processadores traz vantagens em rapidez de aprendizado (JANG, 1992). De maneira independente, vários estudos indicaram a boa capacidade de aproximação em sistemas com funções de base radiais (ROMARIZ, 1995).

O principal interesse neste tipo de rede na área de sistemas conexionistas está no caráter local do processamento dos neurônios constituintes. Cada ponto de entrada excita principalmente um pequeno subconjunto dos neurônios da primeira camada. Sendo assim, durante o treinamento, pontos distantes da área de atuação de uma unidade não influenciam o ajuste de seus parâmetros. Este caráter local do treinamento faz com que novos aspectos do problema possam ser aprendidos sem prejuízo do que já se conhece, característica dificilmente obtida no Perceptron multicamada. Sendo assim, esta rede presta-se melhor, a princípio, para adaptação *on-line*. Como vantagem adicional, durante a operação da rede (após treinamento), uma saída de baixo valor em todos os neurônios indica uma “novidade” em relação aos exemplos usados para aprendizado, o que serve de alerta no caso de operação do sistema em condições radicalmente diferentes das previstas (NARENDRA, 1996).

## 2.8 PERCEPTRON

O modelo de RNA's *perceptron* foi o primeiro a ser desenvolvido. Possui um conceito de neurônio artificial que ainda hoje é usado. Cada neurônio computa uma soma ponderada de suas entradas, e passa esta soma em uma função não-linear com limiarização. Perceptrons tomam decisões, determinam se um padrão de entrada se encaixa ou não em um certo padrão (Loesch, 1996 apud ZAPAROLI, 2002). Segundo Lemes (2005), arquiteturas neurais são tipicamente organizadas em camadas. Usualmente as camadas são classificadas em três grupos:

- a) Camada de entrada: onde os padrões são apresentados à rede;
- b) Camadas Intermediárias: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras ou moduladoras de características;
- c) Camada de Saída: onde o resultado final é concluído e apresentado – Figura 4

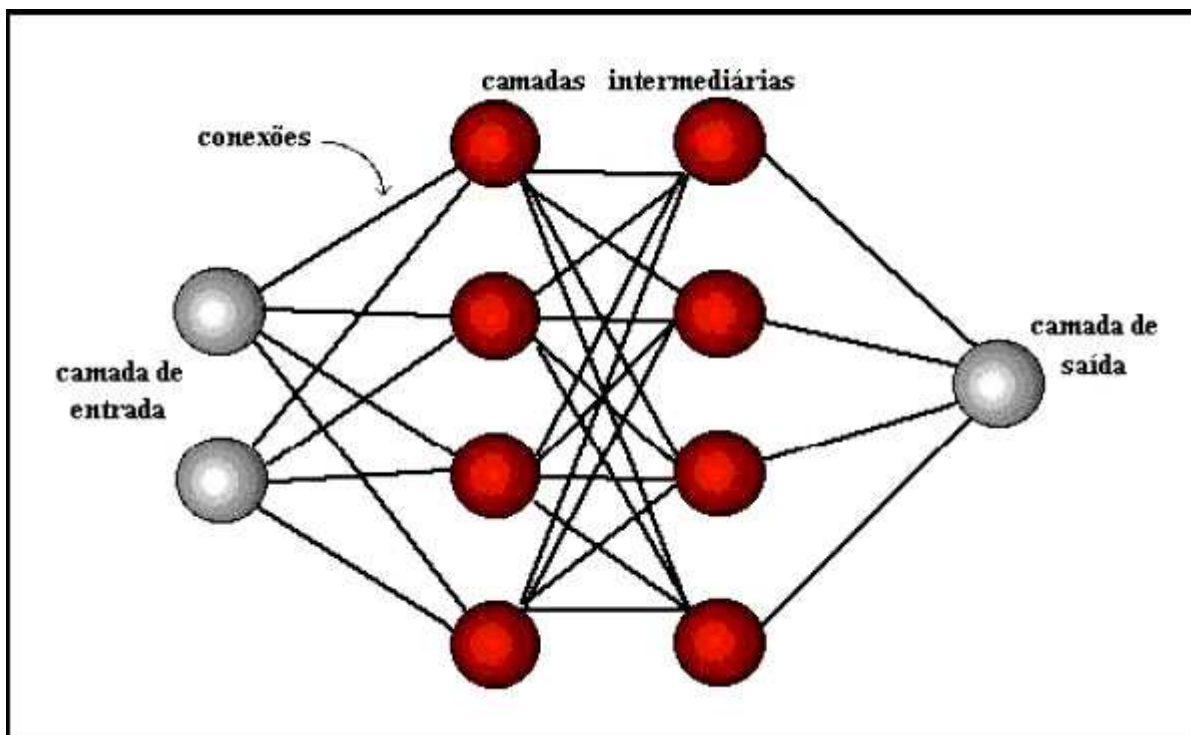


Figura 4: Redes Neurais Artificial.

Fonte: Lemes (2005).

O *Perceptron* trabalha da seguinte forma: quando um neurônio artificial é ativado, ele recebe os valores obtidos nas entradas, realiza a soma ponderada, envia o resultado para a função de transferência, e assim se obtém o valor de saída. O valor de saída obtido é então comparado com a saída desejada, e o erro na saída é calculado diminuindo o valor da saída desejado pelo valor de saída obido. Este erro é utilizado para o cálculo do ajuste dos pesos, dado pela Equação 2:

$$Peso_{\text{Novo } i} = Peso_{\text{Antigo } i} + \frac{\beta \times Erro_i \times Entrada_i}{|Entrada_i|} \quad \text{Equação 2}$$

onde *Entrada* é o valor recebido na entrada, *Erro* é o erro calculado na saída e  $\beta$  é o passo, que é um valor entre 0 e 1 que determina a velocidade do processo de aprendizado.

Quando uma Rede Neural está sendo treinada, existe uma curva de erro a ela associada, como por exemplo, a curva mostrada na Figura 5. Esta figura representa que para certos valores de pesos o erro é menor e para outros é maior, e conforme se altera os valores dos pesos para cima ou para baixo o erro vai sendo alterado de acordo com a curva de erro do problema que está sendo analisado. Cada problema tem a sua curva de erro própria, sendo que a curva da figura apenas um exemplo ilustrativo (HEINEN, 2002).

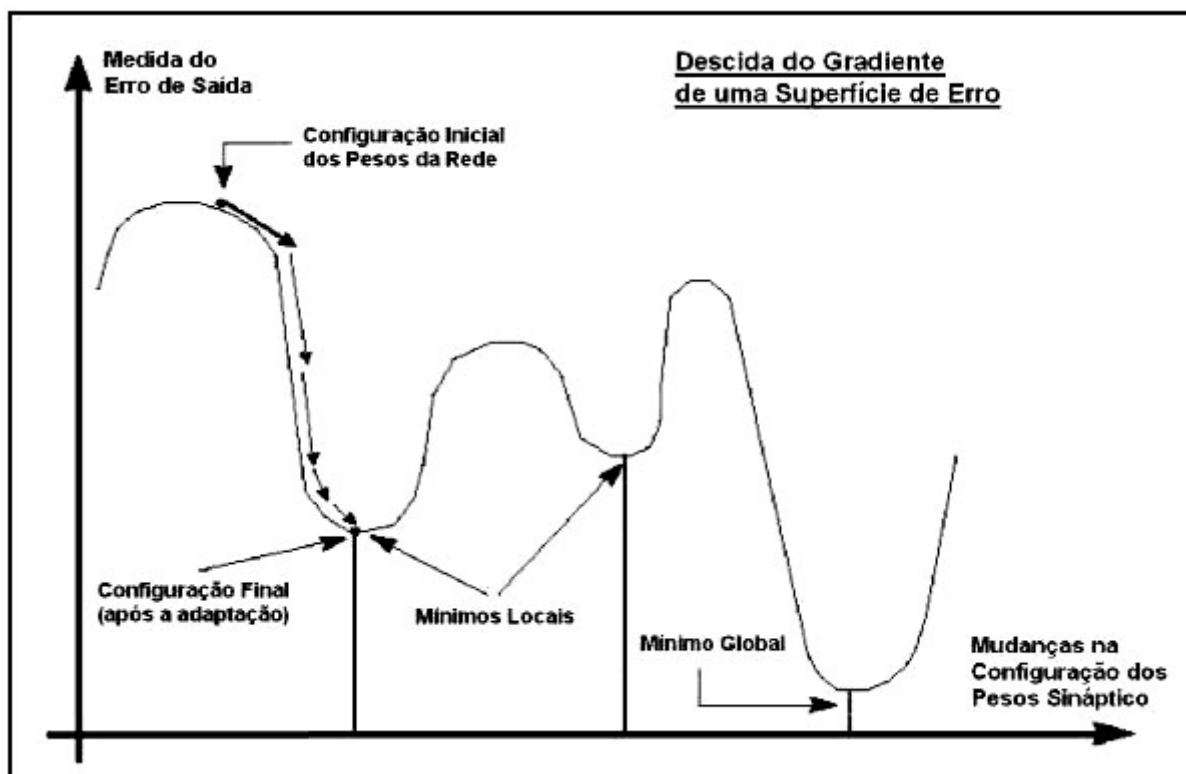


Figura 5: Descida do gradiente de uma superfície de erro.

Fonte: adaptado de Heinen (2002).

Na Figura 5 pode-se notar que existem espécies de “vales” na curva onde o erro é mais baixo que nos pontos vizinhos. Como os pesos são inicializados de forma aleatória, o erro inicial se situa em algum ponto da curva. A medida que os pesos vão sendo ajustados, vai se descendo na curva de erro, até que se atinja um ponto mínimo, de onde qualquer mudança que seja feita para baixo ou para cima fará com que o erro seja aumentado. Em uma curva de erro existe um mínimo global, que é o ponto onde ocorre o menor erro possível, e vários mínimos locais, que são mostradas na Figura 5. O mínimo global é muito difícil de ser atingido, e mesmo que já se esteja nele não será possível de saber, pois é difícil conhecer o formato de toda a curva de erro do problema que está sendo analisado (HEINEN, 2002).

## 2.9 REDES NEURAIIS MULTICAMADAS PERCEPTRON (MLP)

O modelo de rede neural escolhido para este trabalho foi o MLP. O modelo de rede neural multicamadas mais simples é aquele em que a camada de entrada, não realiza processamento, se conectando a camada de saída. Este modelo, com apenas um neurônio na camada de saída recebe a denominação de Perceptron, tem seu uso limitado pois somente é capaz de solucionar problemas linearmente separáveis. A interconexão das entradas à camada de saída por pelo menos uma camada

de neurônios intermediária, tipicamente denominada de camada escondida, remove a limitação do Perceptron e amplia o leque de problemas solucionados por estas redes. Definido um vetor de entradas  $X = [x_0, x_1, x_2, x_3, \dots, x_N]$  e um vetor de saídas  $Y = [y_0, y_1, y_2, \dots, y_N]$ , uma rede neural multicamadas realiza um mapeamento complexo  $y = \Psi(w, x)$ , da entrada  $x$  na primeira camada a saída  $y$ , parametrizado pelos pesos sinápticos  $w$ . Redes multicamadas são, portanto, ferramentas poderosas. Dado um número suficiente de neurônios elas conseguem aproximar qualquer função linearmente contínua.

Uma rede MLP – “Multilayer Perceptron” – apresenta, segundo (BARRETO, 1999) 3 características principais:

1. O modelo de cada neurônio ou elemento processador da rede possui uma função de ativação não-linear. A função sigmoidal atende esta exigência e é muito utilizada em redes MLP.

Função Logística Sigmoidal: é a função de ativação mais utilizada em redes neurais artificiais. Definida como uma função monotônica crescente que apresenta propriedades assintóticas e de suavidade. Um exemplo de função sigmoidal é a função logística definida pela Equação 3 e representada pela Figura 6.

$$f(x) = \frac{1}{1 + e^{-ax}}$$

Equação 3

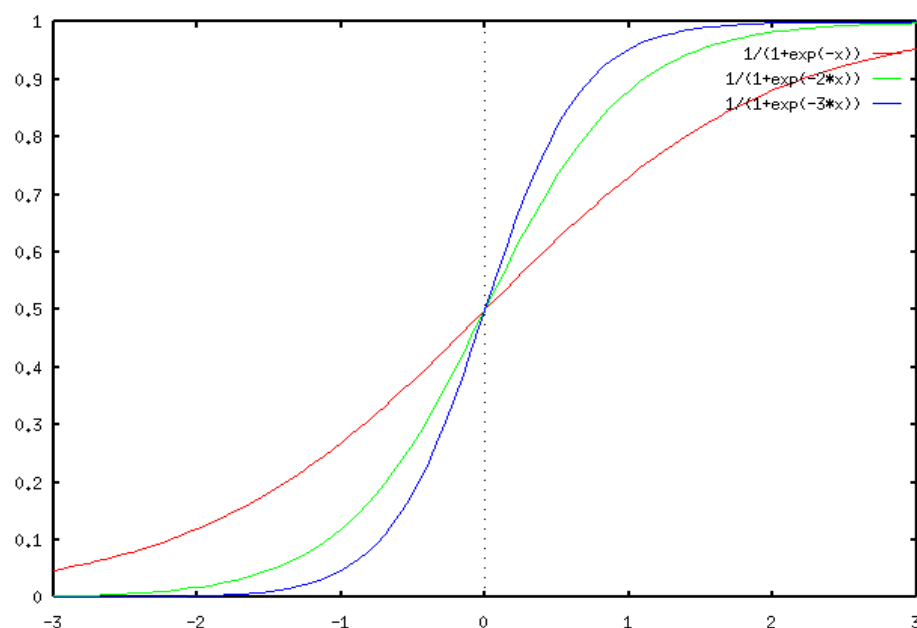


Figura 6: Gráfico da função Sigmoidal

2. A rede possui pelo menos uma camada oculta de processamento com neurônios que não fazem parte da entrada ou da saída.

3. A rede possui alto grau de conectividade entre seus elementos processadores. Esta conectividade é definida pelos pesos sinápticos.

## 2.10 BACKPROPAGATION

O conhecimento das Redes Neurais Artificiais (RNA) se dá através da ponderação que os pesos da conexão entre os neurônios de diferentes camadas trocam entre si. Ou seja, encontrar solução para um determinado problema utilizando RNA seria, a grosso modo, encontrar a melhor topologia de rede, bem como ajustar corretamente os pesos das conexões entre os neurônios (VIEIRA e ROISENBERG, 2005).

O treinamento utilizando o algoritmo de *backpropagation* é dividido em 2 fases. Na primeira fase (Figura 7), um padrão é apresentado à camada de entrada da rede. O resultado é propagado através da rede, camada por camada, até que a resposta seja comparada à saída produzida pela última camada, camada de saída. Na segunda fase, a saída obtida é comparada à saída desejada para esse padrão apresentado. Se a saída obtida não estiver correta, o erro é calculado sendo propagado a partir da camada de saída até a camada de entrada e os pesos das conexões das unidades de camadas internas vão sendo modificados conforme o erro é retropropagado (Carvalho, 2000 apud TONSIG, 2000).

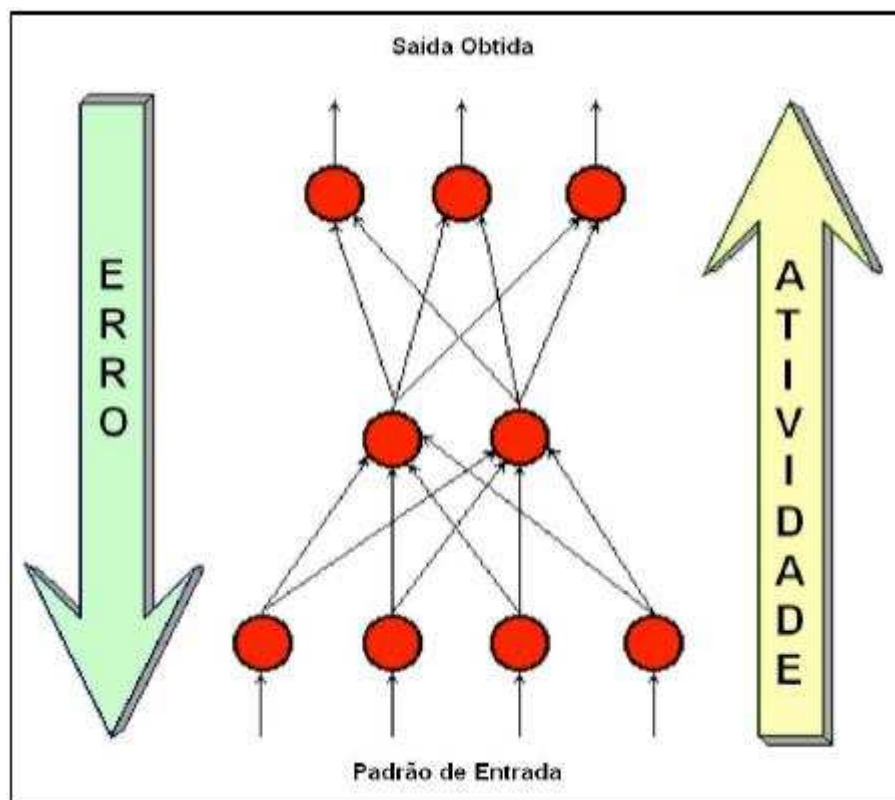


Figura 7: Operação do algoritmo de backpropagation.

Fonte: adaptado de Tonsig (2000).

Depois que a rede estiver treinada e o erro estiver em um nível satisfatório, ela poderá ser utilizada como uma ferramenta para classificação de novos dados.

Para isto, a rede deverá ser utilizada apenas no modo progressivo (*feed-forward*). Ou seja, novas entradas são apresentadas à camada de entrada, são processadas nas camadas intermediárias e os resultados são apresentados na camada de saída, como no treinamento, mas sem a retropropagação do erro. A saída apresentada é o modelo dos dados, na interpretação da rede Figura 8 (Carvalho, 2000 apud TONSIG, 2000).

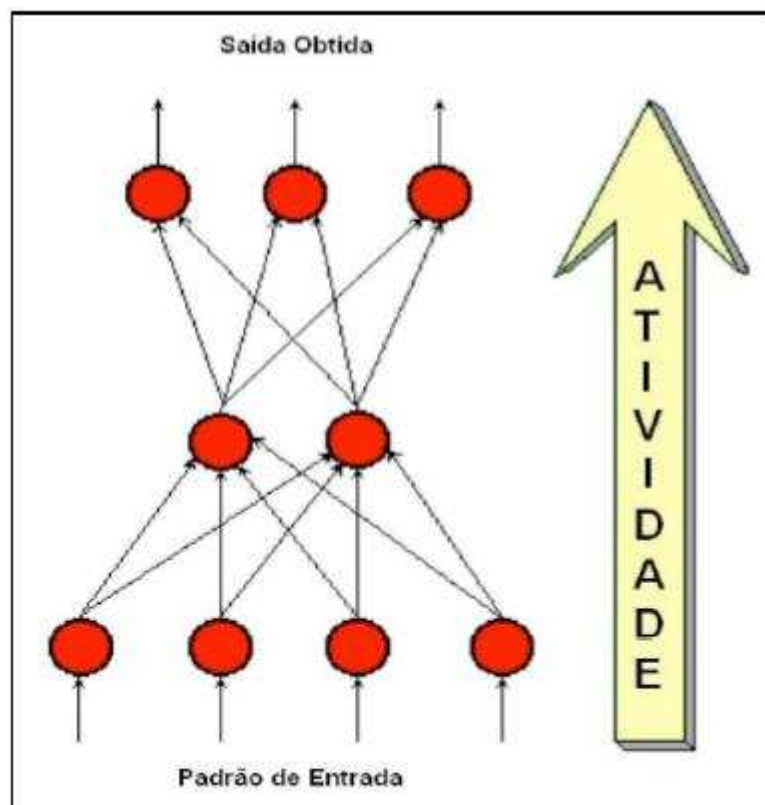


Figura 8: Backpropagation em modo progressivo (feed-forward).

Fonte: Tonsig (2000).

Segundo Fahlman (1988 apud HEINEN, 2002), o algoritmo de *Backpropagation* possui diversos problemas e limitações como:

- a) definição da arquitetura: no *Backpropagation*, é preciso definir manualmente o número de neurônios da Rede Neural, o número de camadas e as interconexões entre os neurônios destas camadas, o que exige que sejam feitas várias simulações até que se consiga chegar a uma arquitetura ideal para a solução de determinado problema;
- b) dependência da inicialização dos pesos: como os pesos são iniciados de forma aleatória, nunca se sabe exatamente em que ponto da curva de erro o aprendizado irá iniciar, o que faz com que cada simulação realizada traga resultados diferentes, algumas vezes não sendo possível se chegar a valores satisfatórios;
- c) minimização do erro lenta e incerta: são necessárias muitas épocas de aprendizado para que o aprendizado ocorra, e nunca se sabe com antecedência se o aprendizado irá ocorrer de forma satisfatória com a configuração e a topologia utilizada; e
- d) plasticidade e elasticidade: depois que uma Rede Neural convergiu para uma solução, é muito difícil adicionar novos exemplos e continuar o aprendizado do ponto onde parou, e ao

se tentar fazer isto, corre-se o risco de fazer com que a Rede Neural esqueça todos os conhecimentos adquiridos anteriormente (esquecimento catastrófico).

As Redes Neurais que utilizam *backpropagation* ainda não tem uma justificativa científica e matemática na qual justifiquem a razão por qual cheguem em um resultado, assim como muitos outros tipos de redes neurais artificiais, podem ser vistas como “caixas pretas”. Assim sendo, muitos estudos vêm sendo realizados para tentar justificar esse comportamento da rede em determinadas situações (TONSIG, 2000).

Com a finalidade de tentar resolver a maioria destes problemas, Fahlman propôs um novo modelo de Redes Neurais, chamado de *Cascade Correlation*, que corrige muitas das deficiências presentes no modelo *Backpropagation* (Fahlman, 1988 apud HEINEN, 2002).

## **2.11 ALGORITMO DE APRENDIZADO**

Existem diversas maneiras de se classificar uma rede neural artificial. Uma das mais importantes é a classificação pelo processo ou algoritmo de aprendizado empregado para treiná-la, que pode ser supervisionado ou não-supervisionado.

No aprendizado supervisionado, representado na Figura 9, a rede recebe uma série de padrões ou vetores de entradas com sua respectiva resposta ou saída desejada. A comparação entre a saída desejada e a saída real gerada pela rede é utilizada para alterar os parâmetros internos (ou pesos sinápticos) da mesma, de modo a aproximar sua resposta a saída desejada. Este procedimento é repetido até que a diferença entre as saídas geradas pela rede para os diversos padrões apresentados e a resposta esperada seja menor que um determinado limiar predefinido. O processo de treinamento ou aprendizado supervisionado de uma rede neural consiste, essencialmente, em minimizar o erro entre a saída da rede para um determinado padrão de entrada e a resposta esperada para aquele mesmo padrão.



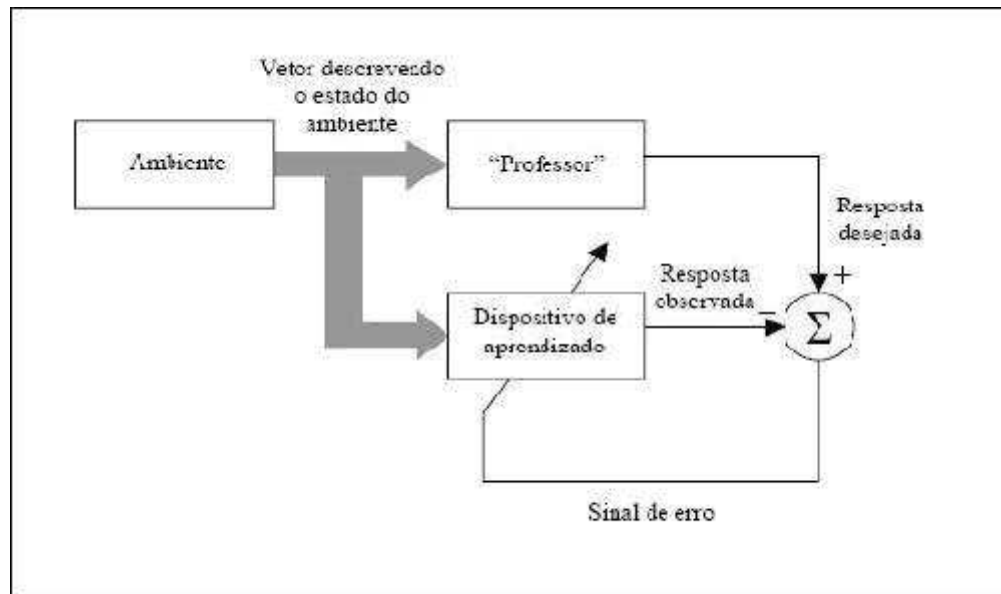


Figura 9: Treinamento supervisionado

Pontos chaves a serem destacados para o aprendizado supervisionado:

1. A rede neural é estimulada pelo ambiente através da apresentação de padrões na sua camada de entrada.
2. Sua estrutura interna é alterada em função destes padrões e da saída desejada.
3. Sua resposta futura ao estímulo do ambiente será diferente devido a iteração da estrutura interna.

Diversos algoritmos para ajuste dos parâmetros da rede, baseados neste conceito, existem como : adaline, madaline, retropropagação do erro "*backpropagation*" ( apresentado anteriormente e utilizado neste trabalho ) dentre outros.

## 2.12 ETAPAS PARA O DESENVOLVIMENTO DE APLICAÇÕES RE REDES NEURAI

Segundo Gurney ( apud FERNANDES, 2003) as etapas de desenvolvimento de aplicações de redes neurais são: Coleta de dados e separação em conjuntos, Separação dos dados, Configuração da rede, Treinamento, Teste, Integração.

### 2.12.1 Coleta de dados e separação em conjuntos

Os dois primeiros passos do processo de desenvolvimento de redes neurais artificiais são a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um

conjunto de testes. Esta tarefa requer uma análise cuidadosa sobre o problema para minimizar ambiguidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrirem amplamente o domínio do problema; não devem cobrir apenas as operações normais ou rotineiras, mas também as exceções e as condições nos limites do domínio do problema.

### **2.12.2 Separação dos Dados**

Normalmente, os dados coletados são separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede e dados de teste, que serão utilizados para verificar sua performance sob condições reais de utilização. Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um conjunto de validação, utilizado para verificar a eficiência da rede quanto a sua capacidade de generalização durante o treinamento, e podendo ser empregado como critério de parada do treinamento.

Depois de determinados estes conjuntos, eles são geralmente colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

### **2.12.3 Configuração da rede**

O terceiro passo é a definição da configuração da rede, que pode ser dividido em três etapas:

- Seleção do paradigma neural apropriado à aplicação.
- Determinação da topologia da rede a ser utilizada - o número de camadas, o número de unidades em cada camada, etc.
- Determinação de parâmetros do algoritmo de treinamento e funções de ativação. Este passo tem um grande impacto na performance do sistema resultante.

Existem metodologias, "dicas" e "truques" na condução destas tarefas. Normalmente estas escolhas são feitas de forma empírica. A definição da configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas.

### **2.12.4 Treinamento**

O quarto passo é o treinamento da rede. Nesta fase, seguindo o algoritmo de treinamento escolhido, serão ajustados os pesos das conexões. É importante considerar, nesta fase, alguns aspectos tais como a inicialização da rede, o modo de treinamento e o tempo de treinamento.

Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido. A escolha errada destes pesos pode levar a uma saturação prematura.

Quanto ao modo de treinamento, na prática é mais utilizado o modo padrão devido ao menor armazenamento de dados, além de ser menos suscetível ao problema de mínimos locais, devido à pesquisa de natureza estocástica que realiza. Por outro lado, no modo *batch* se tem uma melhor estimativa do vetor gradiente, o que torna o treinamento mais estável. A eficiência relativa dos dois modos de treinamento depende do problema que está sendo tratado.

Quanto ao tempo de treinamento, vários fatores podem influenciar a sua duração, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo *backpropagation* não é bem definido, e geralmente é utilizado um número máximo de ciclos. Pode ocorrer que em um determinado instante do treinamento a generalização comece a degenerar, causando o problema de *over-training*, ou seja a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização.

O treinamento deve ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for suficientemente pequena, ou seja menor que um erro admissível. Assim, busca-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

### **2.12.5 Teste**

O quinto passo é o teste da rede. Durante esta fase o conjunto de teste é utilizado para determinar a performance da rede com dados que não foram previamente utilizados. A performance da rede, medida nesta fase, é uma boa indicação de sua performance real.

Devem ser considerados ainda outros testes como análise do comportamento da rede utilizando entradas especiais e análise dos pesos atuais da rede, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas (*prunning*). De modo inverso, valores substantivamente maiores que os outros poderiam indicar que houve *over-training* da rede.

### **2.12.6 Integração**

Finalmente, com a rede treinada e avaliada, ela pode ser integrada em um sistema do ambiente operacional da aplicação. Para maior eficiência da solução, este sistema deverá conter

facilidades de utilização como interface conveniente e facilidades de aquisição de dados através de planilhas eletrônicas, interfaces com unidades de processamento de sinais, ou arquivos padronizados. Uma boa documentação do sistema e o treinamento de usuários são necessários para o sucesso do mesmo.

Além disso, o sistema deve periodicamente monitorar sua performance e fazer a manutenção da rede quando for necessário ou indicar aos projetistas a necessidade de retreinamento. Outras melhorias poderão ainda ser sugeridas quando os usuários forem se tornando mais familiares com o sistema, estas sugestões poderão ser muito úteis em novas versões ou em novos produtos.

## 2.13 APLICAÇÕES DE REDES NEURAIIS ARTIFICIAIS

A seguir serão apresentados alguns exemplos de aplicação de RNA's.

- **Aplicações de Redes Neurais na classificação de imagens de alta resolução espacial e dados do laser scanner, usando uma abordagem orientada a regiões:** Este estudo aborda o tema da integração de dados derivados de um levantamento laser scanner com imagens de satélite de alta resolução espacial, Quickbird, para o mapeamento de áreas urbanas. Para isto, uma metodologia orientada à análise de regiões na imagem é proposta. Inicialmente, o ganho obtido com a inclusão dos dados do laser scanner é comparado com o ganho resultante do uso de parâmetros espaciais derivados das imagens, através da análise da separabilidade das classes e a classificação de áreas de treinamento. Os resultados mostram que a contribuição da informação altimétrica é muito maior do que aquela atribuída aos descritores de forma. Em uma segunda fase, a utilização de redes neurais artificiais como ferramenta para a integração dos dados espectrais e espaciais foi avaliada, sendo que esta metodologia comprovou ser mais eficiente do que outras abordagens tradicionais. O uso de redes neurais e os dados do laser scanner aumentam a qualidade do mapa temático em regiões onde a informação espectral não é suficiente para discriminar objetos diferentes (BOTELHO, 200?).
- **Aplicação de Redes Neurais Artificiais à Engenharia de Estrutura:** Neste trabalho é estudada a aplicação de redes neurais artificiais (RNA) à engenharia de estrutura. RNA constituem hoje uma forte ferramenta computacional, utilizada em diferentes áreas da ciência. A utilização desta ferramenta tem como objetivo a exploração de métodos

alternativos para a análise estrutural, que forneçam resultados com rapidez e segurança (LASTIRI, PAULETTI, 2004).

- **Aplicação de Redes Neurais Artificiais e Sensoriamento Remoto na Caracterização Ambiental do Parque Estadual Morro do Diabo:** As características espaciais, espectrais, radiométricas e temporais das imagens de sensoriamento remoto permitem sua utilização em uma ampla variedade de estudos ambientais. A aplicação de estruturas de redes neurais artificiais a essas imagens, muitas vezes integradas a dados temáticos, permite estabelecer relações de síntese entre as variáveis que representam o meio físico e fenômenos ambientais acessíveis a diferentes níveis analíticos (GALO, 2000).
- **Aplicação de Redes Neurais para o Diagnóstico Diferencial da Doença Meningocócica:** Nas infecções do Sistema Nervoso Central (SNC), como no caso da Doença Meningocócica (DM), é essencial que o médico atue rapidamente no sentido de obter um diagnóstico específico e instrua uma terapia apropriada. A avaliação inicial deve levar em conta tanto a urgência em se introduzir um tratamento adequado, quanto o risco potencial envolvido. Este trabalho propõe uma da técnica da Inteligência Artificial, conhecida como Redes Neurais Artificiais (RNA), para simular o processo de decisão médica.
- **Aplicação de Redes Neurais Artificiais no Reconhecimento On-Line de Assinaturas:** Este trabalho apresenta uma abordagem baseada em Redes Neurais Artificiais voltada para a autenticação de assinaturas. O trabalho proposto utiliza um tablet para a aquisição on-line das assinaturas, que são processadas de modo a verificar se pertencem ou não a um usuário específico previamente cadastrado. Um protótipo do reconhecedor de assinaturas vem sendo desenvolvido, onde seu desempenho tem sido plenamente satisfatório (HEINEN, 200?).
- **Aplicações de Redes Neurais para Identificação de Feições Espaciais:** O presente trabalho investigou a utilização de Redes Neurais Artificiais (RNAs) na identificação de feições espaciais em imagens digitais oriundas do sensoriamento remoto (BOTELHO, 200?).
- **Aplicações de Redes Neurais Artificiais em Simulações:** Considerando a simulação de situações reais como elemento importante na otimização de processos, o presente artigo tem por objetivo a apresentação de um sistema especialista desenvolvido para modelagem

computacional de simulação de dados de consulta, utilizando a técnica de redes neurais artificiais, que consiste em um método para solucionar problemas de Inteligência Artificial (IA) (BILOBROVEC, 2004).

### **3 DESENVOLVIMENTO**

Este capítulo apresenta o desenvolvimento da RNA proposta. Primeiramente apresenta-se a análise do sistema SIGMA, a fim de esclarecer como serão as entradas e saídas da rede. Em seguida fala-se das redes MLP e do algoritmo de *backpropagation* que foram utilizados neste trabalho.

Posteriormente serão apresentadas as características da rede desenvolvida, o treinamento e os testes.

#### **3.1 ANÁLISE DO SISTEMA SIGMA (DESCRIÇÃO DO SISTEMA)**

Segware do Brasil é a primeira indústria especializada no desenvolvimento de Software e Hardware exclusivamente para Empresas de Segurança Privada e de Monitoramento de Alarmes no país. Seu nome deriva de SEGurança e hardWARE/softWARE. Constituída por profissionais de reconhecida competência e talento na área de segurança, a Segware é fruto de uma bagagem superior a 30 anos em Segurança Privada.

O SIGMA está integrado à Internet através do WEBALARME ([www.webalarme.com.br](http://www.webalarme.com.br)), disponibilizando dados para os clientes finais, podendo estar disponível no Web Site da própria Empresa de Monitoramento. Desta forma, em qualquer lugar, o cliente da operadora SIGMA poderá acessar seu histórico de atendimento e ser informado em que momento sua empresa foi aberta e por quem, dentre outros dados relevantes.

O SIGMA possui recursos exclusivos e inéditos:

- Integração à Internet;
- Comunicação Automática para Celular;
- Serviço de Atendimento a Clientes;
- Controle de Horários ( foco desse trabalho ).

Todos os controles de horários das centrais são feitos na tela representada pela Figura 8. Nela temos 48 posições, correspondendo cada uma a meia hora. Em cada uma dessas frações

coloca-se o estado da central naquela meia hora. Para isto, utiliza-se de duas letras maiúsculas e minúsculas na seguinte disposição:

- A - Central deve estar Armada.
- D - Central deve estar Desarmada.
- a - Tolerância para arme.
- d - Tolerância para desarme.

Para cada faixa de horários criada deve-se designar o dia da semana e os dias que essa faixa será válida. O SIGMA compreende uma condição de exclusão por importância, ou seja, uma faixa para o dia específico sobrepõe uma faixa para a semana ou para todos os dias. O sistema primeiro procura uma faixa de horários específica para o dia. Se não encontrar e for um dia entre segunda e sexta, procurará se existe uma faixa para segunda-sexta, não encontrando e estando dentro de segunda a sábado procura uma faixa para segunda - sábado, não encontrando procura uma faixa para todos os dias. Caso não encontre nenhuma dessas faixas considerará que a central deve estar, durante todo o dia, ARMADA e gera os eventos de anormalidade X2 (Não Armado), X3 (Não Desarmado), X4 (Armado Fora do Horário) e X5 (Desarmado fora do Horário) a cada vez que a central armar ou desarmar de acordo com cada caso. Na Figura 10 encontramos, também, o campo Feriado, que fornece a opção de armar e desarmar a central em feriados. Para ativar os horários dos dias, basta clicar na caixa que melhor o satisfaz.

A Figura 10 ilustra a configuração da tabela de horário onde existe 48 campos de preenchimento obrigatório, cada campo representa a configuração que a central deve estar no horário referenciado pelo mesmo.

**Copiar de:**  
Coloque ao lado o código da central que será usada como original.  
Ao clicar no botão OK, o SIGMA vai apagar todos os horários da central atual e copiar os horários da central de origem.

☐ 1 - Domingo  
☐ 2 - Segunda  
☐ 3 - Terça  
☐ 4 - Quarta  
☐ 5 - Quinta  
☐ 6 - Sexta  
☐ 7 - Sábado  
☐ 8 - Seg-Sex.  
☐ 9 - Seg-Sab.  
☒ 10 - Todos  
☐ 11 - Feriado

A - Central Armada  
D - Central Desarmada  
a - Tolerância para arme  
d - Tolerância para desarme

Figura 10: Tabela de Horários

Fonte: Segware do Brasil (2006).

Para a escolha das classes de entrada da rede neural foi consultado no banco de dados do software SIGMA a tabela dbRamo. Essa tabela contém todos os ramos de atividades das centrais cadastradas no sistema. O preenchimento do ramo de atividade é obrigatório para a efetivação do cadastro das centrais.

### **3.2 DEFINIÇÃO DO TAMANHO DA REDE**

Projetar uma rede neural MLP para um problema específico envolve determinar quantas camadas e, principalmente, quantos neurônios artificiais ocultos a rede deve possuir. O tamanho das camadas de entrada e saída será sempre determinado pela natureza do próprio problema. Como exemplo prático e ligado ao problema aqui estudado, uma rede neural MLP desenvolvida para classificar as tabelas de horários terá 48 entradas, correspondentes aos quarenta e oito campos existentes na tabela de horários apresentada na Figura 10 e apenas 8 neurônios na camada de saída, correspondente aos 8 ramos de atividades. Determinar, quantos neurônios ocultos e quantas camadas uma rede que procure solucionar com precisão este problema é tarefa que não possui resposta exata. Existem, entretanto, soluções aproximadas, heurísticas, que procuram estimar estas variáveis. Estas heurísticas expõem sempre o compromisso entre a convergência e a generalização da rede (BARRETO, 2000).

Segundo WASSERMAN, (1989), convergência e generalização são:

**Convergência:** é a capacidade da rede de aprender todos os padrões de entrada usados no seu treinamento. Uma rede muito pequena, em relação ao problema em análise, não será capaz de aprender os dados de treinamento do problema, ou seja, a rede não possuirá parâmetros ou pesos sinápticos suficientes.

**Generalização:** é a capacidade da rede neural responder adequadamente a padrões jamais vistos. Uma rede muito grande, com número de neurônios muito superior aos necessários para o problema em análise, não responderá corretamente a padrões nunca vistos, perderá a capacidade de generalizar. Isto ocorrerá pois os pesos sinápticos da rede aprenderão os vetores de entrada e também o ruído presente nos dados de treinamento.

A capacidade de generalização de uma rede neural é afetada por três fatores :

1. Tamanho e eficiência dos dados de treinamento.
2. Arquitetura da rede e número de processadores nas camadas ocultas.



### 3. Complexidade do problema.

Normalmente não se tem controle sobre o 3º fator apresentado (complexidade do problema). O problema pode ser investigado de duas maneiras:

1. Fixa-se arquitetura e tamanho da rede e determina-se qual o tamanho dos dados de treinamento necessário.
2. Tamanho do conjunto de treinamentos é fixo e deve se achar a melhor arquitetura e tamanho da rede neural para o problema em questão.

No presente trabalho foi definido com base em testes realizados que a melhor arquitetura da RNA para resolver o problema em questão teria 15 neurônios na camada intermediária como mostra a Figura 11.

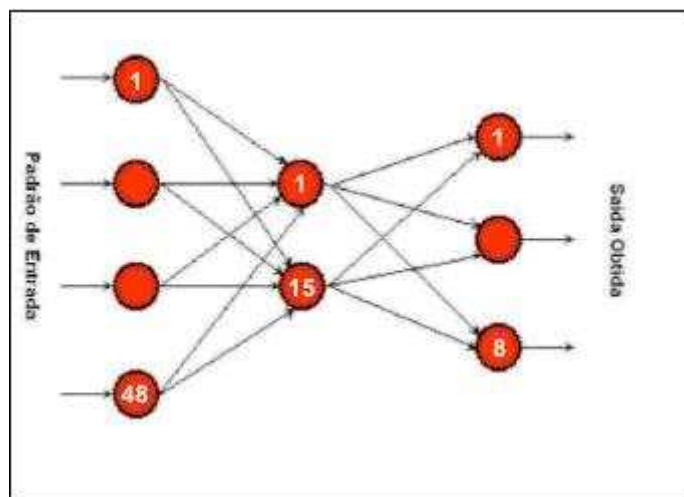


Figura 11: Rede Neural

## 3.3 ETAPAS DE DESENVOLVIMENTO DA REDE

Para o desenvolvimento da rede, foram consideradas as seguintes etapas: coleta e separação dos dados; separação dos dados em conjuntos; configuração da rede, treinamento e testes. Cada uma dessas etapas será descrita à seguir.

### 3.3.1 Coleta e Separação de Dados

A primeira etapa do processo foi definir quais dados eram importantes e quais poderiam ser utilizados para classificar as centrais. Como o preenchimento de todos os 48 campos da tabela de

horário é obrigatório não houve necessidade de completar nenhuma tabela. O banco de dados utilizado foi a junção de 3 bancos totalizando 800 centrais cadastradas e 1,8 anos de eventos. Com o intuito de acelerar e minimizar a possibilidade de erros durante o processo, foram utilizados algumas ferramentas administrativas que gerenciam os bancos de dados SQL-Server. Esses dados foram conseguidos junto a equipe de desenvolvimento do Software SIGMA.

Os dados utilizados neste trabalho são reais e foram extraídos a partir do banco de dados fornecido pela empresa Segware do Brasil e organizados no Notepad para posterior uso na rede neural. Durante a extração dos dados foi detectado que em muitas centrais as tabelas de horários se repetiam, por esse motivo houve uma redução significativa nos dados utilizados para treinamento da rede conforme mostra a Figura 12.

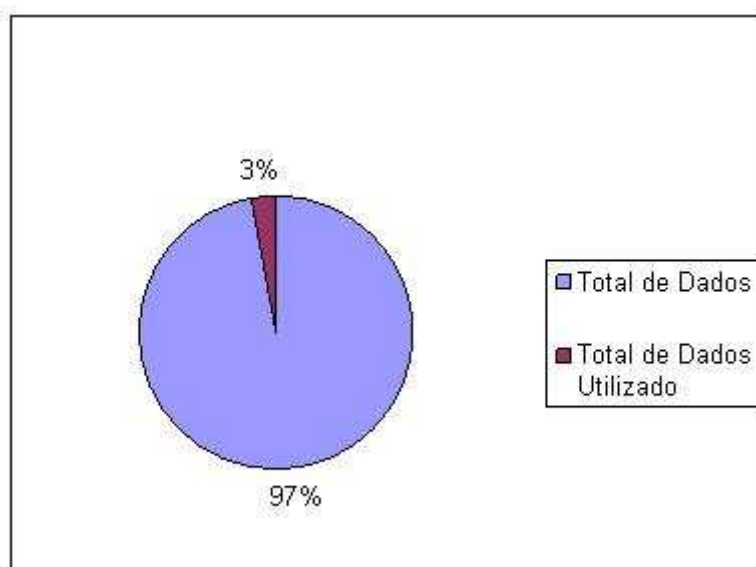


Figura 12: Dados da Base

O Sigma é um software que fornece um grande número de informações. Um exemplo de dados extraídos do banco de dados pode ser visualizado na Figura 13.

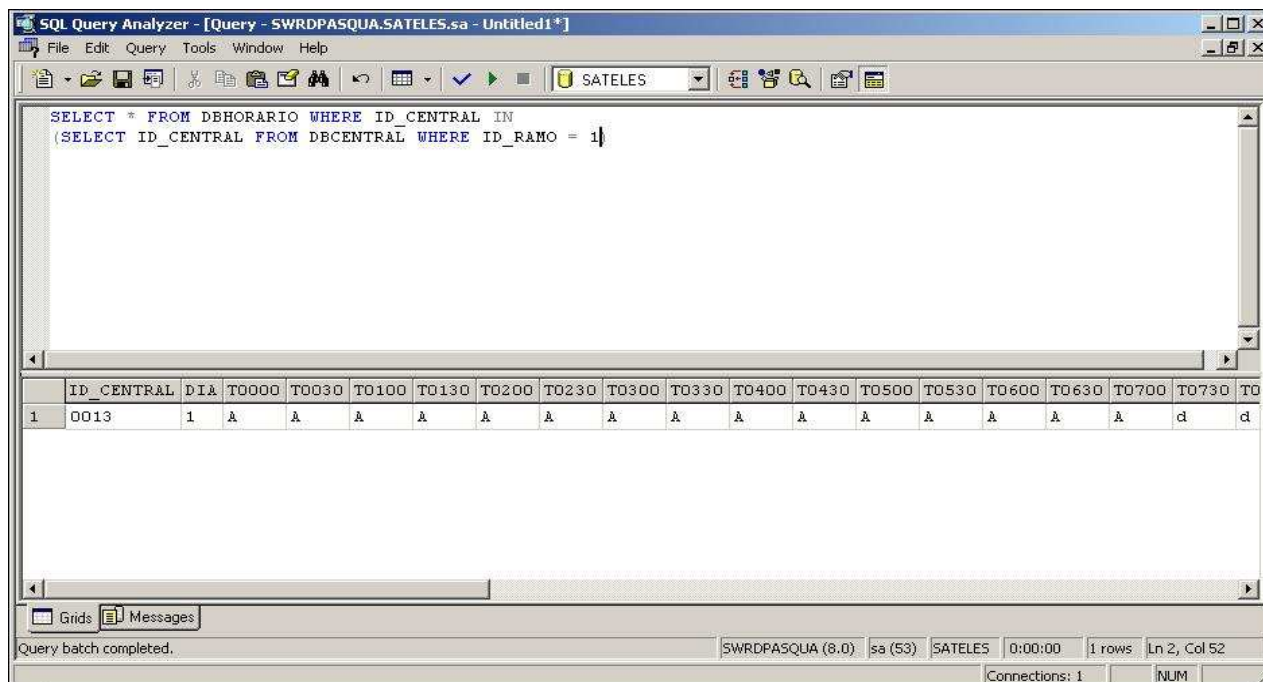


Figura 13: Exemplo de dados extraídos do banco

### 3.3.2 Separação dos Dados em Conjuntos

O conjunto de dados foi dividido em três partes: a primeira parte totaliza 24 exemplos sendo 3 exemplos de cada padrão de ramo de atividade utilizados para o processo de treinamento da rede. Para as fases de validação e testes foram retirados dados do mesmo banco de dados de forma a contemplar todos os ramos de atividades.

Sendo assim, os valores representados no Anexo I ficaram distribuídos da seguinte forma:

- Linha 1 à 24 – Treinamento.
- Linha 25 à 49 – Teste.

Antes de serem apresentados à rede, os dados ainda passaram por um pré-processamento, pois os dados das tabelas de horários são armazenados no banco em forma de letras conforme exemplo mostrado na Figura 13. Para podermos trabalhar com esses dados foi necessário fazer a conversão para formato numérico. As letras foram trocadas dentro de dois valores. Os dados utilizados na entrada da rede foram trocados para os valores [0,1] da seguinte forma:

Todas as letras A/a foram trocados pelo valor 1 e as letras D/d por 0.

Já os valores das respostas desejadas para a rede foram substituídos por [0,1] igualmente. O objetivo da substituição é poder trabalhar com operações matemáticas utilizadas nas Redes Neurais Artificiais.

Os dois conjuntos de dados, de treinamento e de teste, são armazenados em um mesmo arquivo txt da seguinte maneira: cada linha do arquivo de dados representa um padrão de entrada e cada linha do arquivo de padrões também representa o resultado esperado para a entrada correspondente.

### 3.3.3 Configuração da Rede

A utilização de uma rede neural artificial exige que uma série de escolhas, não triviais, sejam feitas na busca de um modelo que apresente um resultado considerado satisfatório. Dentre as principais escolhas estão: a topologia da rede, o algoritmo de aprendizagem, a função de ativação, a taxa de aprendizagem e o número ideal de épocas.

Optou-se pelo modelo de rede neural MLP, freqüentemente utilizado em aplicações financeiras, com algoritmo de treinamento *backpropagation*, no qual os pesos são ajustados baseados na regra de aprendizagem por correção de erro. A rede, modelada e executada no software Matlab, foi estruturada em três camadas, ou seja, apenas uma camada oculta foi utilizada, pois, de acordo com FAUSETT (1994), duas ou mais camadas ocultas podem beneficiar determinadas aplicações, mas uma única camada oculta é considerada suficiente.

A camada de entrada é formada por quarenta e oito neurônios, um para cada variável de entrada da rede. O número de neurônios na segunda camada ( oculta ) foi definido com 15 neurônios, a explicação para esse numero de neurônios na camada de teste se dará na fase de testes. Na terceira camada há 8 neurônios que representam as variáveis de saída da rede.

A função Logística Sigmoidal foi definida como a função de ativação.

### 3.3.4 Treinamento

Considerando que o objetivo da rede neural neste trabalho é realizar o reconhecimento de padrões, seu treinamento foi realizado da maneira descrita a seguir.

Tendo definido o número de épocas para o treinamento, realiza-se a apresentação dos 23 pares entrada-saída de treinamento N vezes para que a rede seja treinada até que o erro se torne aceitável levando-se em consideração a natureza da aplicação. Após o treinamento, é feito o teste com o primeiro padrão do conjunto de teste e guarda-se o resultado obtido (a resposta da rede e o

erro gerado) em um vetor. Adiciona-se o padrão testado ao conjunto de treinamento e treina-se novamente a rede N vezes, agora com um padrão a mais.

Depois deste treinamento realiza-se o teste com o próximo padrão do conjunto de teste. Adiciona-se o novo resultado ao vetor de resultados, adiciona-se o padrão ao conjunto de treinamento voltando a treinar a rede N vezes, com o novo conjunto de treinamento. Este processo repete-se até que todos os padrões de teste tenham sido testados.

No treinamento, alguns aspectos devem ser considerados, tais como a inicialização da rede, modo de treinamento e o tempo de treinamento. No estudo optou-se em inicializar os pesos sinápticos com uma forma randômica em um intervalo definido. O intervalo escolhido foi  $[-1;1]$ . O modo de treinamento poderia ser de duas formas: padrão, onde os pesos são modificados a cada padrão apresentado na rede, sendo este menos susceptível aos mínimos locais; batch, onde a correção é realizada uma vez por ciclo (época). O modo de treinamento escolhido foi o modo Padrão. O tempo de treinamento, ou seja a especificação de um critério de parada pode ser feito considerando dois aspectos: número de ciclos e/ou a taxa de erro ser menor que um erro aceitável.

O número de ciclos da rede é muito importante, pois se for estabelecido um parâmetro incorreto, a rede poderá perder a sua generalização ficando esta especializada em um determinado conjunto de dados. Este fato é chamado de *overtraining*. O critério de parada aqui adotado foi a taxa de erro representado pela Figura 14.

No treinamento foi utilizado a Regra Delta que tem como objetivo determinar um conjunto de pesos (W) que minimize o erro quadrático sobre determinado número de amostras de treinamento. A obtenção de um ponto mínimo é alcançado através de um processo de iteração local, utilizando um conjunto de amostras.

Regra Delta:

Erro = saída esperada - saída obtida

Peso novo = Peso antigo + (Erro \* Entrada \* Fator de Aprendizagem)

O critério de parada utilizado para o treinamento da rede foi o MSE (*Mean Squared Error*) onde a saída produzida pela rede é comparada com a saída desejada. Caso a saída produzida não esteja dentro do padrão desejado os parâmetros da rede devem ser modificados.

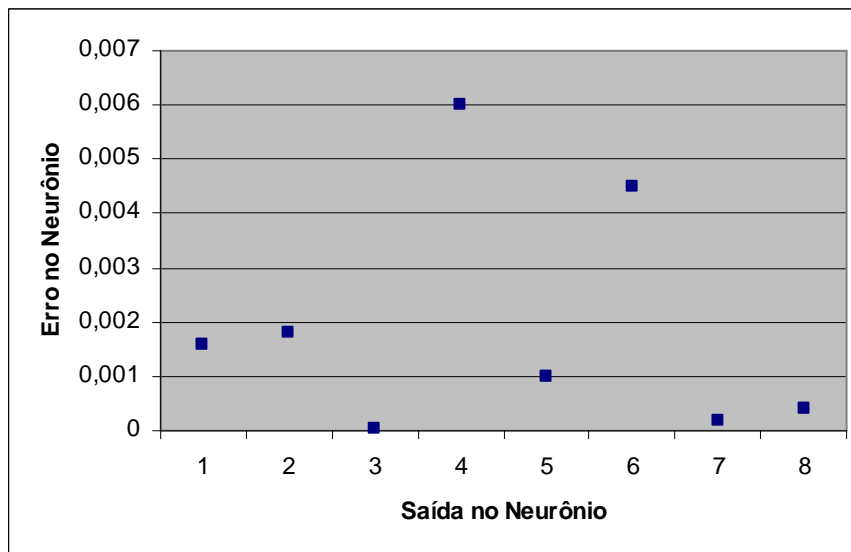


Figura 14: Erro quadrático para 150 épocas

### 3.3.5 Testes

A fase de teste foi responsável pela determinação do desempenho da rede. Nesta fase também foi realizada a análise do erro gerado ao final para verificar se a estrutura da rede teria que ser modificada, tanto na parte de camadas quanto no número de épocas de treinamento. A RNA utilizada nessa pesquisa foi baseada no aprendizado supervisionado, obtendo em cada época um erro quadrado médio que visa ajustar os pesos sinápticos da rede para um melhor aprendizado. O vetor de pesos sinápticos foi iniciado com valores randomizados entre -1 e 1. Também foi estabelecido um número de épocas e iterações, evitando assim o excesso de treinamento, que seria prejudicial para o desempenho final da rede. Inicialmente, teste foram realizados fixando-se a função de transferência Logarítmica Sigmoidal e variando-se os padrões de entrada e a taxa de aprendizagem da rede. Os resultados obtidos podem ser observados na Tabela 1. Desta forma testes posteriores foram realizados para analisar a melhor arquitetura. O software utilizado foi o Matlab® através da Neural Network Toolbox e o Turbo Pascal. O computador utilizado foi um AMD Athlon 900 Mhz.

Na fase de treinamento e validação, o processo somente é interrompido por dois mecanismos pré-determinados. O primeiro diz respeito a um padrão mínimo de erro na saída, determinado no caso em foco, na fase de retro propagação, pelo erro quadrático médio, e o segundo

mecanismo de interrupção é devido ao número máximo de iterações (épocas). Foi pré-definido que para todas as redes testadas seria fixado o valor para o erro médio, 0,001 e um limite de 1.500 iterações. No tocante à arquitetura da rede, foi utilizada apenas uma camada oculta, tendo em vista que, esta topologia mais simples, já é suficiente para possibilitar a aproximação de funções contínuas e não-lineares das entradas.

Assim, com o objetivo de encontrar a melhor arquitetura, ou grupo de redes que adquiram as melhores capacidades de aprendizado, partiu-se de 10 (dez) neurônios na camada oculta, verificando-se o comportamento das redes, tanto na fase de treinamento e validação, acompanhando o parâmetro de erro, quanto na fase de resultados dos testes realizados. Em relação ao parâmetro de erro, foi utilizado, como técnica para mensuração a relação entre os valores da saída da rede e de treinamento, para isto, foi plotado, a saída da rede e o conjunto de validação (valor esperado). Os menores valores de erros encontrados determinam as melhores redes.

A última etapa do processo de implementação de redes neurais proposto neste trabalho é a Etapa de Integração. Esta etapa ainda não foi realizada, devendo ser executada em trabalhos futuros.

Os Anexos I, II e III apresentam respectivamente os dados analisados a implementação da rede no MATLAB e a implementação da rede em PASCAL.

### 3.3.6 Testes numéricos e resultados obtidos

Nesta seção são apresentados alguns resultados obtidos com a rede *backpropagation* aplicada às tabelas de horários. Além dos testes propriamente ditos (realizados com os dados de teste), foram realizados testes com os dados de treinamento para verificar se a rede realmente estava aprendendo.

Os testes foram realizados para cada para cada padrão com o objetivo de encontrar valores para os parâmetros que combinados realizassem a melhor previsão. Os parâmetros não tiveram que ser alterados pois os resultados obtidos foram satisfatórios.

O reconhecimento dos padrões só foi testado com o treinamento sendo feito em 150 épocas pois somente nesse momento a rede atinge o erro previamente estabelecido representados pela Quadro 1.

Neurônios na Camada Oculta	Erro	Épocas	Fator Aprendizagem	Padrões Reconhecidos
15	0,001059	150	0,25	8
15	0,1432	1000	0,25	Padrões não verificados

15	0,122	5000	0,25	Padrões não verificados
15	0,103	10000	0,32	Padrões não verificados
15	0,115	10000	0,12	Padrões não verificados
15	0,08337	10000	0,25	Padrões não verificados
15	0,12	25000	0,25	Padrões não verificados
10	0,746	100	0,25	Padrões não verificados
10	0,097	50	0,32	Padrões não verificados
10	0,097	500	0,32	Padrões não verificados
10	0,0968	5000	0,32	Padrões não verificados
10	0,0633	5000	0,25	Padrões não verificados
10	0,0253	10	0,25	Padrões não verificados
10	0,1002	500	0,25	Padrões não verificados
10	0,703	100	0,25	Padrões não verificados

Quadro 1: Erro x Épocas

O gráfico abaixo mostra o erro para algumas épocas as quais foram testada com a melhor arquitetura encontrada, 48 neurônios na camada de entrada, 15 neurônios na camada intermediária e 8 neurônios na camada de saída representados pela Figura 15.

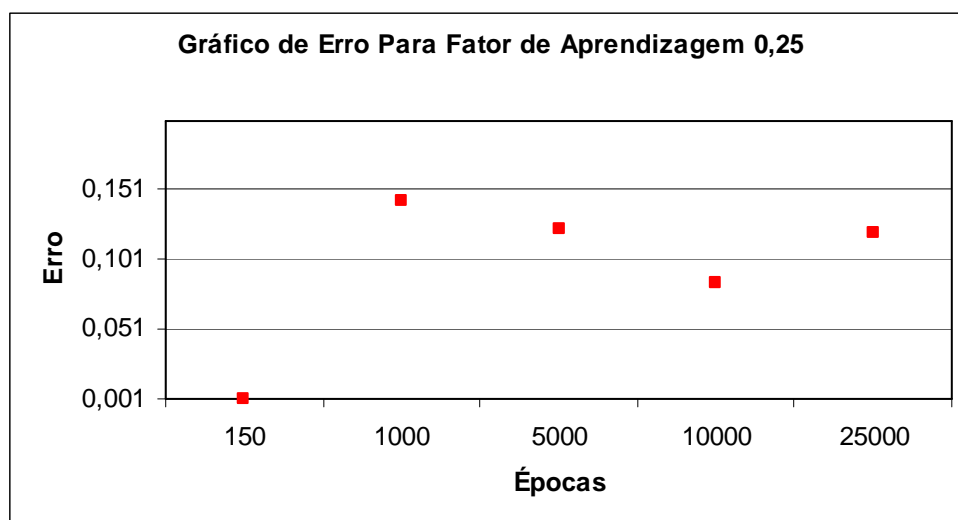


Figura 15: Gráfico de Erro para Fator de Aprendizagem 0,25





## **4 CONCLUSÕES**

Uma rede neural artificial foi implementada para classificar tabela de horários em sistemas de monitoramento de alarme. Os resultados obtidos demonstraram sua eficiência nessa classificação levando-se em consideração os resultados obtidos.

A RNA implementada e treinada se demonstrou capaz de reconhecer corretamente 8 tipos de centrais ( padrões ) para as quais foi treinada anteriormente.

Mesmo que com um número pequeno de épocas se consiga chegar a um resultado satisfatório, outras arquiteturas e parâmetros devem ser testado pois a prática revela que com poucas épocas de treinamentos as RNA's se limitam-se a reconhecer somente os padrões para os quais ela foi treinada.

Como trabalhos futuros, fica a proposta de desenvolvimento de um sistema que através dos eventos de ARME/DESARME consiga montar uma tabela de horários.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARRETO, Jorge Munis. **Inteligência Artificial no Limiar do Século XXI**, 2a. Ed., Florianópolis: Duplic Edições, 1999.

BILOBROVEC, Marcelo; MARÇAL, Rui Francisco Martins e PILATTI, Luiz Alberto. Aplicações de redes neurais em artificiais em simulações. *In*: SIMPEP. XI, 2004, Bauru, SP. **Anais...** Bauru, 2004. p. 1-7.

BOTELHO, Mosar Faria. Aplicação de redes neurais para identificação de feições espaciais. I WORKSHOP SOBRE GEOPROCESSAMENTO DA UFV. **Anais...** Viçosa, MG, 2002.

BOTELHO, Mosar Faria. Aplicação de redes neurais na classificação de imagens de alta resolução espacial e dados do laser scanner, usando uma abordagem orientada a regiões. *In*: Vários. **Boletim ciência geod..** Curitiba, Paraná, 2004

BRAGA, A. P., CARVALHO, A. C. P. L. F., LUDERMIR, T. B. **Redes Neurais Artificiais: teoria e aplicações**. Ed LTC. 2000.

CARVALHO FILHO, E. C. B., Modelagem, Aplicações e Implementações de Redes Neurais. IV ESCOLA REGIONAL DE INFORMÁTICA DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, Itajaí, **Anais.....** 1996.

CASTRO, L. N. de. **Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais**. Dissertação (Mestrado) — Universidade Estadual de Campinas - UNICAMP, Campinas, SP, Setembro 1998

DIVERIO, T, V.; SHINABUKURO, Y, E.; FORMAGGIO, A, R.; SILVA, J, D, S. Sistema Neural Integrado para a Identificação de Áreas Desflorestadas por Corte Raso na Floresta Amazônica In: XII Simpósio Brasileiro de Sensoriamento Remoto. Goiânia, abril de 2005. **Anais.....** Disponível em: <http://marte.dpi.inpe.br/col/ltid.inpe.br/sbsr/2004/11.19.17.24/doc/4053.pdf#search=%22Sistema%20Neural%20Integrado%20para%20a%20Identifica%C3%A7%C3%A3o%20de%20C3%81reas%20Desflorestadas%22>. Acessado em: 15 de setembro de 2006.

FAUSETT, L. **Fundamentals of neural networks: architectures, algorithms and applications**. Ed. Prentice Hall. New Jersey, 1994.

FERNANDES, Anita M. da Rocha. **Inteligência Artificial Noções Gerais**, Florianópolis, Visual Books, 2003

FOR SEGURANÇA ELETRÔNICA, Condor, Disponível em: <http://www.softwarecondor.com.br>. Acessado em: 29 agosto 2006.

GALO, Maria de Lourdes Bueno Trindade. **Aplicação de redes neurais artificiais e sensoriamento remoto na caracterização ambiental do parque estadual Morro do Diabo**. 2000. 17 f. Tese (Doutorado em Ciência de Engenharia Ambiental) – Escola de Engenharia de São Carlos, São Carlos, 2000.

HAYKIN, Simon. **Redes Neurais. Princípio e prática**. Tradutor: Paulo Martins Engel. 2 edição. Porto Alegre: Bookman. 2001.

HEINEN, Milton Roberto ; OSÓRIO, Fernando Santos . Aplicação de Redes Neurais Artificiais no Reconhecimento On-Line de Assinaturas. WORKCAP - WORKSHOP DE COMPUTAÇÃO APLICADA DO PIPCA, 2002, São Leopoldo - RS. **Anais...** Unisinos, 2002. v. 1. p. 3-4.

HEINEN, Milton Roberto. **Autenticação On-line de assinaturas utilizando Redes Neurais**. Trabalho de Conclusão de Curso (Bacharelado em Informática), Universidade do Vale do Rio dos Sinos, São Leopoldo, 2002.

HERTZ J. A. Krogh, and R. G. Palmer, **Introduction to the Theory of Neural Computing**, Addison-Wesley, 1991.

HOHONEN, T. - **Self-Organization and Associative Memory**, Berlin: Springer-Verlag, 1987.

JANG, J.S., SUN, C.T.: **Functional equivalence between radial basis function networks and fuzzy inference systems**. Preprint (Internet). Berkeley, EUA: University of California. Department of Electrical. Engineering and Computer Science, 1992.

LASTIRI, Cayo Pardos e PAULETII, Ruy Marcelo de Oliveira. Aplicação de redes neurais artificiais à engenharia de estruturas. JORNADAS SUD-AMERICANAS DE INGENIERÍA ESTRUCTURAL. XXXI, 2004, Mendoza, Argentina. **Memórias...** Argentina, 2004. p. 1-16.

LEMES, Neslon H.T. **Redes Neurais: o Perceptron**. 2005. Disponível em: <http://www.usuarios.unincor.br/nhtlemes/cpu/perceptron.pdf>. Acesso em: 20 out 2006.

LOESCH, Cláudio. **Redes neurais artificiais, fundamentos e modelos**. Blumenau: Editora da FURB, 1996. 166 p.

MULLER, F.M. et al. **Métodos Heurísticos e Meta-heurística**. II Simpósio Nacional de Informática, Centro Universitário Franciscano, Santa Maria, RS, **Anais.....** 1997.

MUSSOI, I. V. L. de. **Uma Abordagem Simplificada de Detecção de Intrusão Baseada em Redes Neurais Artificiais**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina - UFSC, Florianópolis, SC, Fevereiro 2006

NARENDRA, K.S.: Neural Network for control: Theory and Practice, **Proceedings IEEE**, Vol. 84, Nr. 10, pp. 1385- 1406, Oct. 1996.

OSÓRIO, F. ; BITTENCOURT, J. R. Sistemas inteligentes baseados em RNAs aplicados ao processamento de imagens. WORKSHOP DE INTELIGÊNCIA ARTIFICIAL, **Anais....**Santa Cruz do Sul: UNISC, 2000.

QUINTELA, Daniela Mendes; OLIVEIRA, Marcio Ferreira. **Tutorial de Redes Neurais: Aplicações em Bioinformática**. Disponível em: [http://www.lncc.br/~labinfo/tutorialRN/frm4\\_perceptronMultiCamadas.htm](http://www.lncc.br/~labinfo/tutorialRN/frm4_perceptronMultiCamadas.htm). Acessado em Outubro de 2006.

ROMARIZ, A.R.S: **Representação e Aquisição de Regras em Sistemas conexionistas**. Dissertação de Mestrado. Universidade Estadual de Campinas, Campinas, SP, Brasil, 1995.

SEGWARE DO BRASIL, Sigma , Disponível em: <http://www.segware.com.br>. Acessado em: 29 agosto 2006.

SILVA, E. M. **Redes Neurais Artificiais**. WORKSHOP DE VERÃO DE FUNDAMENTOS DA COMPUTAÇÃO WFC 2002. Pelotas, novembro de 2002, **Anais.....**. Disponível em: <http://descartes.ucpel.tche.br/WFC/2002/07-rna.pdf>. Acessado em Setembro de 2006.

SILVA FILHO, A. C. R., **Redes Neurais Artificiais**. Disponível em: <http://www.ffclrp.usp.br/disciplinas/fisicaematematica/5915755-724.html>. Acessado em: 20 agosto 2006.

TAFNER, M. A.; XEREZ M.; RODRIGUES FILHO I. W. R. - **Redes Neurais Artificiais: Introdução e princípios de neurocomputação**. Ed. EKO. 1995.

TATIBANA, Cássia Yuri, KAETSU, Deisi Yuki. **Uma Introdução às Redes Neurais**. Disponível em: <http://www.din.uem.br/ia/neurais>. Acessado em: 07 setembro 2006.

TONSIG, Sérgio Luiz. **Redes Neurais Artificiais Multicamadas e o Algoritmo de Backpropagation**. Campinas, 2000. Disponível em: <http://209.123.181.8/~archives/tutoriais/1243.zip>. Acesso em 21/10/2006.

TSOI, A. C., A. D. BACK, Locally recurrent and globally feedforward networks: A critical review of architectures. **IEEE Transactions on Neural Networks**, vol. 5, 1994.

VALLE FILHO, Adhemar Maria. **Modelo para Implementação de Consciência em Robôs Móveis**, Tese (Doutorado) – Universidade Federal de Santa Catarina - UFSC Florianópolis, 2003.

VELLASCO, M. M. **Redes neurais**. Núcleo de Pesquisa em Inteligência Computacional Aplicada PUC Rio. 2001. Disponível em: <http://www.ele.puc-rio.br/labs/ica/icahome.html>. Acesso em: 29 Agosto de 2006

VIEIRA, R. C.; ROISENBERG, M. **Redes neurais artificiais : um breve tutorial**. Universidade Federal de Santa Catarina. Florianópolis, 2005.

W. SECURITY, Íris, Disponível em: <http://www.sistemairis.com.br>. Acessado em: 29 agosto 2006

WASSERMAN, Philip, D. “Neural Computing: Theory and Practice”, VNR, New York, 1989

ZAPAROLI, Alexsandra. **Protótipo de Software para Controle de Acesso de Funcionários Utilizando Redes Neurais Artificiais para Identificação de Impressão Digital**. Blumenau. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação), Universidade Regional de Blumenau, 2002

## **ANEXOS**