

**HÉLDER NEVES SANTA ROSA**

**IMPLEMENTAÇÃO DE UM SIMULADOR DE REDES NEURAIIS  
PARA PREVISÃO DE SÉRIES TEMPORAIS**

Trabalho de conclusão de curso apresentado ao Curso de Ciência da Computação.

UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS

Orientadora: Profa. Lorena Sophia Campos de Oliveira

**BARBACENA**  
**2003**

**HÉLDER NEVES SANTA ROSA**

**IMPLEMENTAÇÃO DE UM SIMULADOR DE REDES NEURAIIS  
PARA PREVISÃO DE SÉRIES TEMPORAIS**

Este trabalho de conclusão de curso foi julgado adequado à obtenção do grau de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Presidente Antônio Carlos.

Barbacena – MG, 03 de dezembro de 2003.

---

Profa. Lorena Sophia Campos de Oliveira - Orientadora do Trabalho

---

Prof. Eduardo Macedo Bhering - Membro da Banca Examinadora

---

Prof. MSc. Élio Lovisi Filho - Membro da Banca Examinadora

## **AGRADECIMENTOS**

*Meus sinceros agradecimentos: a minha família, pelo apoio, carinho e dedicação; aos amigos que estiveram ao meu lado dando força o tempo todo; a Profa. Lorena pela dedicação na execução do trabalho; a todos os professores que nos transmitiram suas experiências e conhecimentos e aos funcionários e amigos da SUPLAN e SEPLAO.*

## RESUMO

O objeto de trabalho desta monografia foi a construção de um simulador para Redes Neurais Artificiais que se adequasse a previsão de Séries Temporais. Para tanto foi feita uma revisão bibliográfica que trouxe subsídios para a escolha da Rede Elman e do algoritmo Backpropagation para a implementação do simulador. Um vez implementado, o simulador foi submetido a testes com dados reais a fim de verificar seu desempenho na realização da tarefa a que se propõe.

**Palavras-chave:** Inteligência Artificial, Redes Neurais Artificiais, Séries Temporais, Backpropagation, Rede de Elman

## SUMÁRIO

<u>FIGURAS.....</u>	<u>7</u>
<u>TABELAS.....</u>	<u>9</u>
<u>1 INTRODUÇÃO.....</u>	<u>10</u>
<u>2 REDES NEURAS ARTIFICIAIS.....</u>	<u>14</u>
<u>3 SÉRIES TEMPORAIS.....</u>	<u>25</u>
<u>4 SIMULADOR.....</u>	<u>31</u>
<u>5 APLICAÇÕES.....</u>	<u>46</u>
<u>6 CONCLUSÕES.....</u>	<u>50</u>
<u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u>	<u>52</u>
<u>ANEXO A – SÉRIES TEMPORAIS UTILIZADAS NA APLICAÇÃO.....</u>	<u>54</u>

## FIGURAS

<b>FIGURA 2.1 NEURÔNIO BIOLÓGICO: OS CONSTITUINTES DA CÉLULA.....</b>	<b>15</b>
<b>FIGURA 2.2 NEURÔNIO ARTIFICIAL (HAYKIN, 2001).....</b>	<b>16</b>
<b>FIGURA 2.3 FUNÇÕES DE ATIVAÇÃO MAIS UTILIZADAS: (A) FUNÇÃO LINEAR; (B) FUNÇÃO DEGRAU; (C) FUNÇÃO SIGMÓIDE (OU LOGÍSTICA); (D) TANGENTE HIPERBÓLICA.....</b>	<b>19</b>
<b>FIGURA 2.4 EXEMPLOS DE REDES FEED-FORWARD: (A) REDE DE CAMADA ÚNICA; (B) REDE MULTICAMADAS.....</b>	<b>20</b>
<b>FIGURA 2.5 REDE RECORRENTE.....</b>	<b>21</b>
<b>FIGURA 3.6 GRÁFICO DA SÉRIE TEMPORAL QUE REPRESENTA OS VALORES DAS EXPORTAÇÕES DO BRASIL.....</b>	<b>26</b>
<b>FIGURA 4.7 REDE DE ELMAN.....</b>	<b>33</b>
<b>FIGURA 4.8 TELA PRINCIPAL DO PREVSIM.....</b>	<b>40</b>
<b>FIGURA 4.9 TELA INSERIR SÉRIE.....</b>	<b>41</b>
<b>FIGURA 4.10 TELA CONFIGURAÇÕES.....</b>	<b>42</b>
<b>FIGURA 4.11 TELA PREVISÕES.....</b>	<b>43</b>
<b>FIGURA 4.12 TELA GRÁFICO.....</b>	<b>44</b>
<b>FIGURA 5.13 GRÁFICO DA SÉRIE EXPORTAÇÕES BRASIL.....</b>	<b>47</b>

**FIGURA 5.14 GRÁFICO DA SÉRIE TEMPERATURA EM BARBACENA.....47**

## TABELAS

<b>TABELA 4.1 NOTAÇÃO UTILIZADA NA DEMONSTRAÇÃO DO ALGORITMO BACKPROPAGATION.....</b>	<b>35</b>
<b>TABELA 5.2 TESTES COM A SÉRIE EXPORTAÇÕES BRASIL.....</b>	<b>48</b>
<b>TABELA 5.3 TESTES COM A SÉRIE TEMPERATURA EM BARBACENA.....</b>	<b>48</b>
Tabela A.1 Valores da série Exportações Brasil.....	53
Tabela A.2 Valores da série Temperatura em Barbacena. ....	54

# 1 INTRODUÇÃO

## 1.1 INTELIGÊNCIA ARTIFICIAL E SÉRIES TEMPORAIS

Inteligência Artificial possui diversas definições, mas talvez a que melhor exprime os esforços da comunidade científica é dada por Elaine Rich (Rich, 1988): Inteligência Artificial é o estudo de como fazer os computadores realizarem tarefas em que, no momento, as pessoas são melhores, apesar de parecer um tanto utópica esta definição devido a atual tecnologia e avanços obtidos pelas pesquisas. Na atualidade as técnicas da Inteligência Artificial vem sendo empregadas na resolução de novos e antigos problemas.

A Inteligência Artificial se divide em duas linhas de pesquisa: simbólica e conexionista. A linha simbólica segue a tradição lógica. Já a conexionista se utiliza de modelos matemáticos para a simulação dos componentes do cérebro e suas interligações, como forma de imitar a inteligência exibida por determinados sistemas biológicos (Bittencourt, 2001; Yioda, 2000).

Dentro da linha conexionista se encontram as Redes Neurais Artificiais uma forma de computação composta por unidades processadoras e interconexões. De inspiração biológica, corresponde a um modelo matemático do cérebro. O interesse por esta forma de computação é devido as características desejáveis apresentadas pelo cérebro: robustez, tolerância a falhas, flexibilidade, capacidade de lidar com informações incompletas e inconsistentes (Bittencourt, 2001; Russell e Norvig, 1995).

As Redes Neurais Artificiais por apresentarem características tais como: capacidade de aprendizado, tomada de decisões e generalização são empregadas na resolução de problemas que apresentam um desempenho insatisfatório tais como: otimização, reconhecimento e classificação de padrões e previsão de séries temporais, quando tratadas por técnicas computacionais convencionais (Bittencourt, 2001).

Assim, as Redes Neurais Artificiais são uma ferramenta utilizada com sucesso na realização de previsão de séries temporais mostrando-se capazes de realizar prognósticos com qualidade e precisão superiores aos modelos tradicionais como: clássico, ARIMA e estrutural (Fernandes *et. al.*, 1995; Tápia *et. al.*, 2000).

## **1.2 OBJETIVO**

O objetivo principal deste trabalho é desenvolver um simulador de redes neurais e uma pequena aplicação. Para tanto é necessário uma revisão dos conceitos de Redes Neurais Artificiais e Séries Temporais.

## **1.3 DESCRIÇÃO DO TRABALHO**

As etapas que constituem o trabalho são: revisão dos conceitos relativos as Redes Neurais Artificiais e Séries Temporais; a implementação do simulador; e a aplicação do simulador na previsão de séries temporais.

A primeira etapa do trabalho, que corresponde a revisão bibliográfica teve como objetivo descrever as Redes Neurais Artificiais com suas principais características que as tornam atraentes na previsão de Séries Temporais. Com relação às Séries Temporais buscou-se elencar seus principais conceitos que julgou-se necessários ao entendimento e prosseguimento do trabalho.

Nesta fase ainda foi feito um estudo mais detalhado do algoritmo Backpropagation, um algoritmo de aprendizado supervisionado para Redes Neurais Artificiais e da Rede Elman que foram escolhidos para a implementação do simulador.

No desenvolvimento do simulador optou-se pelo uso da linguagem Delphi para a implementação do algoritmo Backpropagation e das ferramentas que auxiliam a utilização do simulador.

No simulador é possível configurar a rede de acordo com alguns parâmetros tais como: escolha da função de ativação, função que simula as características do neurônio; valor do coeficiente de aprendizado, ou a velocidade que a rede caminha na curva de erro em busca de uma solução ótima; número de neurônios de entrada e da camada oculta.

Na última etapa duas séries temporais diferentes foram submetidas ao simulador e os resultados obtidos foram descritos.

## **1.4 ORGANIZAÇÃO DA MONOGRAFIA**

A monografia está organizada ainda em mais cinco capítulos, descritos a seguir:

- Capítulo 2: apresenta os principais conceitos das Redes Neurais Artificiais. São descritas razões para utilizá-las, suas principais características, aplicações e um pequeno histórico da sua evolução.
- Capítulo 3: apresenta os principais conceitos envolvidos na análise e previsão de Séries Temporais.

- Capítulo 4: apresenta a solução escolhida para a implementação do simulador. A Rede Elman e o algoritmo Backpropagation são descritos de forma detalhada. Além disso, é descrito o processo de implementação do simulador, suas características básicas e ferramentas de apoio.
- Capítulo 5: apresenta o trabalho realizado na previsão de duas séries temporais diferentes e a avaliação do desempenho do simulador.
- Capítulo 6: apresenta algumas conclusões e sugere possíveis trabalhos futuros.

## 2 REDES NEURAIS ARTIFICIAIS

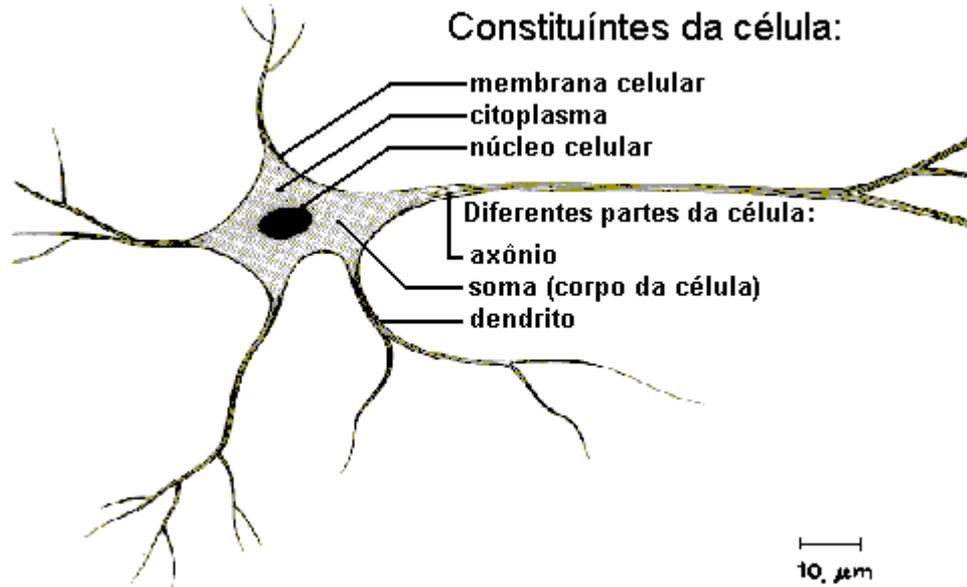
Uma Rede Neural Artificial é um modelo computacional que compartilha algumas das propriedades do cérebro: consiste de muitas unidades simples trabalhando em paralelo sem um controle central. As conexões entre unidades possuem pesos numéricos que podem ser modificados pelo elemento de aprendizado (Russell & Norvig, 1995). A essas unidades damos o nome de neurônio e suas conexões sinapses.

As Redes Neurais Artificiais são semelhantes ao cérebro humano em dois aspectos basicamente (Haykin, 2001):

- A experiência é a fonte do conhecimento adquirido, e
- O conhecimento adquirido é armazenado nas sinapses.

O neurônio, célula constituinte do cérebro, ilustrado pela figura 2.1, é composto por um corpo celular chamado Soma onde se encontra seu núcleo e por axônios e dendritos. O axônio, uma fibra nervosa de superfície lisa com poucas ramificações e maior comprimento é responsável pela transmissão na comunicação com outros neurônios. Os dendritos, que tem

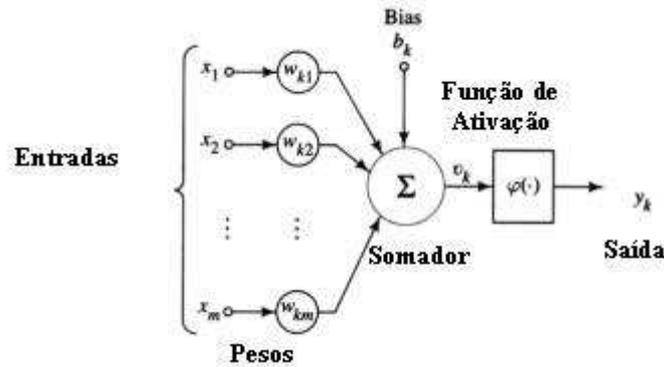
aparência de árvores, possuem superfície irregular e muitas ramificações, atuam como receptores nesta comunicação.



**Figura 2.1 Neurônio biológico: os constituintes da célula.**

Tal comunicação ou interação é chamada sinapse e é caracterizada por um processo químico no qual são liberadas substâncias transmissoras que se difundem pela junção sináptica entre neurônios, o que causa aumento ou queda no potencial elétrico do neurônio receptor. Resumindo, uma sinapse é a conexão entre neurônios o que implica em excitação ou inibição do neurônio receptor (Haykin, 2001).

De forma análoga, um neurônio artificial apresenta as mesmas características. A figura 2.2 mostra um modelo onde podem ser vistos: sinapses representadas pelas entradas e pesos sinápticos, somatório e função de ativação. Cada sinapse é caracterizada por um estímulo de entrada multiplicado pelo seu peso sináptico correspondente. Depois desta multiplicação, cada sinal de entrada é somado e o resultado é então, aplicado a uma função de ativação que restringe a saída do neurônio a um intervalo de 0 a 1 ou de  $-1$  a  $+1$ .



**Figura 2.2 Neurônio artificial (Haykin, 2001).**

O neurônio artificial pode ser descrito pela equação (Haykin, 2001):

$$y_k = \varphi \left( \sum_{i=1}^n x_i w_{ki} \right) \quad (\text{Equação 2.1})$$

2.1)

Onde,  $y_k$  é a saída do neurônio;  $\varphi$  é a função de ativação;  $x_1, x_2, \dots, x_n$  são os sinais de entrada do neurônio; e  $w_{k1}, w_{k2}, \dots, w_{kn}$  são os pesos sinápticos do neurônio em questão (neurônio k).

Portanto, o neurônio artificial imita o funcionamento do neurônio biológico por meio das entradas, representando as sinapses e pela função de ativação que simula o processo químico que libera substâncias químicas que excitarão ou inibirão os próximos neurônios.

## 2.1 RAZÕES PARA USAR REDES NEURAIS ARTIFICIAIS

Alguns problemas complexos, atualmente intratáveis com as técnicas de programação tradicionais são solucionados com as Redes Neurais Artificiais. A razão para tal poder está na sua capacidade de aprender e na sua estrutura paralela. As Redes Neurais contam ainda com as seguintes propriedades e características (Haykin, 2001):

- Não-linearidade: um neurônio artificial pode ser linear ou não-linear. Esta é uma propriedade importante para entradas inerentemente não-lineares, como por exemplo, o sinal de voz.
- Mapeamento de Entrada-Saída: ao realizar o mapeamento de entrada-saída, processo de aprendizagem no qual é apresentado o par entrada-saída para a rede, esta é capaz de alterar seus pesos sinápticos objetivando minimizar o erro obtido da resposta esperada (saída).
- Adaptabilidade: capacidade de se adaptar a modificações do meio ambiente. Tal adaptação consiste em retreinamento para operar nas novas condições.
- Resposta a evidências: capacidade de fornecer informações sobre a tomada de decisão. Consiste na confiança ou crença na escolha feita.
- Informação contextual: cada neurônio é afetado pela atividade de todos os outros neurônios da rede, pois o conhecimento é representado pela estrutura e estado da rede.
- Tolerância a falhas: uma rede neural apresenta tolerância a falhas devido a natureza distribuída do conhecimento pela rede, por exemplo, pequenas falhas como um neurônio ou sinapse danificada fazem que a rede apenas perca em qualidade sendo capaz ainda, de realizar sua tarefa.
- Uniformidade de análise e projeto: a mesma notação é aplicada em todos os domínios de aplicação.
- Analogia Neurobiológica: motivação e prova de que o processamento paralelo do cérebro além de tolerante a falhas é rápido e poderoso.

## 2.2 CARACTERÍSTICAS DAS REDES NEURAIS ARTIFICIAIS

Os diversos modelos de Redes Neurais Artificiais possuem algumas características que são aqui apresentadas:

### 2.2.1 TIPOS DE ENTRADA

Tipo de entrada é a forma do valor de entrada da rede. É dividido em binária e intervalar:

- Binário: entradas discretas (0 e 1).
- Intervalar: qualquer valor numérico como entrada.

### 2.2.2 FUNÇÃO DE ATIVAÇÃO

Cada neurônio realiza um processamento simples: recebe uma entrada e computa um novo nível de ativação (Russel e Norvig, 1995). Este processamento é composto por duas etapas: na primeira, cada entrada  $x_i$  do neurônio é multiplicada pelo peso sináptico correspondente  $w_{ji}$  (peso da entrada  $i$  do neurônio  $j$ ). O resultado de cada multiplicação é então somado. Na segunda etapa a soma é aplicada a uma função de ativação  $f$ , obtendo-se a saída do neurônio ( $y$ ):

$$y = f\left(\sum x_i w_{ji}\right) \quad \text{(Equação 2.2)}$$

A função de ativação deve simular as características não lineares do neurônio biológico (Abelém, 1994). As funções mais utilizadas, ilustradas pela figura 2.3, são:

- Função linear: É uma equação linear da forma:

$$f(x) = ax \quad \text{(Equação 2.3)}$$

- Função degrau: É uma equação utilizada para valores binários e é da forma:

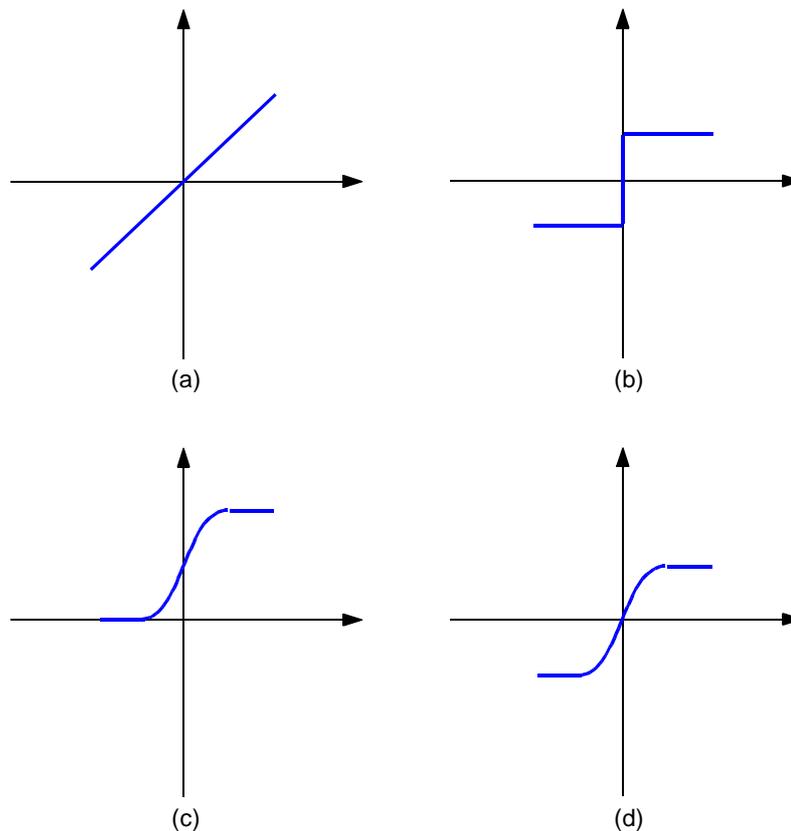
$$f(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \quad (\text{Equação 2.4})$$

- Função sigmóide: Também chamada de função logística, é uma função contínua que permite a transição gradual entre os dois estados. É dada por:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (\text{Equação 2.5})$$

- Função tangente hiperbólica: É uma função sigmóide que varia entre  $-1$  e  $+1$ . É dada por:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (\text{Equação 2.6})$$

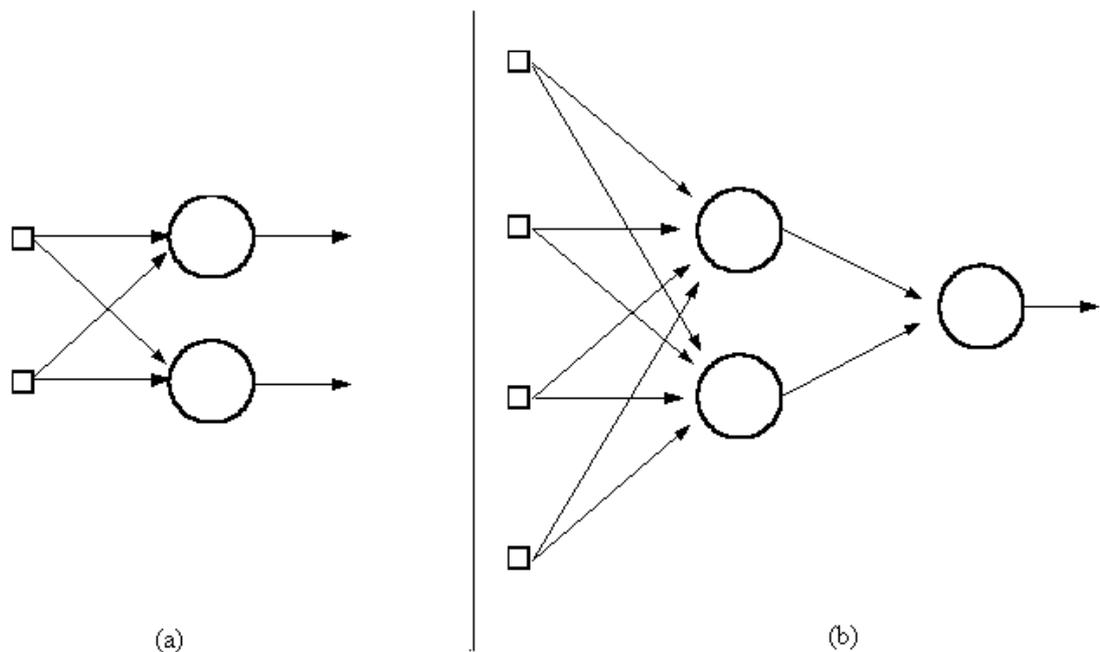


**Figura 2.3 Funções de ativação mais utilizadas: (a) função linear; (b) função degrau; (c) função sigmóide (ou logística); (d) tangente hiperbólica**

### 2.2.3 TOPOLOGIA DA REDE

Existe uma grande variedade de redes cada uma produzindo diferentes resultados. Elas podem ser classificadas basicamente em alimentadas a frente (feed-forward) e recorrentes (Russell & Norvig, 1995).

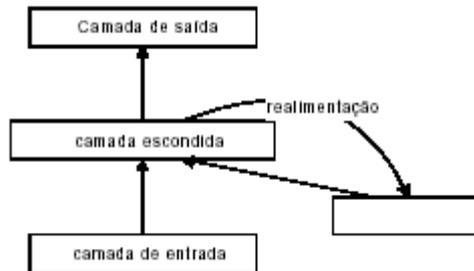
Nas redes feed-forward os neurônios estão dispostos em camadas, podendo haver redes com uma única camada e redes com múltiplas camadas. As redes feed-forward são inerentemente acíclicas, ou seja, o sinal é propagado somente da entrada para a saída da rede. Também são chamadas redes sem memória. As redes multicamadas se distinguem das redes de camada única pela presença de uma ou mais camadas ocultas. As entradas de um neurônio são as saídas dos neurônios da camada anterior, portanto não há ligação entre neurônios de uma mesma camada. Uma rede feed-forward é representada pela notação  $e-o_1-o_2-\dots-on-s$ , onde  $e$  representa o número de neurônios nas camadas de entrada,  $o_1, o_2, \dots, o_n$  representam o número de neurônios nas camadas ocultas e  $s$  o número de neurônios na camada de saída. A figura 2.4 ilustra uma rede feed-forward de camada única (a) e uma rede multicamada 4-2-1 (b) (Haykin, 2001; Russell & Norvig, 1995).



**Figura 2.4 Exemplos de redes feed-forward: (a) rede de camada única; (b) rede multicamadas.**

Perceptron e o MLP (Multi-Layer Perceptron) são, respectivamente, exemplos de modelos de rede de camada única e rede de múltiplas camadas.

Ao contrário das redes feed-forward as redes recorrentes possuem laços de realimentação, ou seja, a saída de um neurônio pode ser entrada para outro de uma camada precedente ou, no caso de auto-realimentação, para o próprio neurônio. As redes recorrentes, chamadas de redes com memória, não possuem organização rígida e seus neurônios têm



liberdade para se ligar a qualquer outro neurônio (Abelém, 1994; Haykin, 2001; Russell & Norvig, 1995). A figura 2.5 ilustra uma rede recorrente.

**Figura 2.5 Rede recorrente.**

Como exemplos de modelos de redes recorrentes temos a rede de Elman, rede de Hopfield, rede de Jordan e NARX.

## 2.2.4 TIPOS DE APRENDIZADO

O processo de aprendizagem ocorre através de um processo iterativo de ajuste dos parâmetros livres, pesos sinápticos, por estimulação do ambiente (Haykin, 2001).

Os paradigmas de aprendizado são: aprendizado supervisionado e aprendizado não supervisionado e serão descritos a seguir.

- **Aprendizado Supervisionado**, também chamado de aprendizado com professor. Esta forma de aprendizado se baseia em um conjunto de exemplos de entrada-saída que é apresentado a rede. A partir da entrada a rede realiza seu processamento e a saída obtida é comparada com a saída esperada. Caso não sejam iguais, um processo de ajuste de pesos é aplicado buscando-se um

erro mínimo ou aceitável. O algoritmo de aprendizado supervisionado mais comum é o Backpropagation (Haykin, 2001).

- Aprendizado não supervisionado: é caracterizado pela ausência de algum elemento externo supervisor, ou seja, um padrão de entrada fornecido permite que a rede livremente escolha o padrão de saída a partir das regras de aprendizado adotadas. Possui duas divisões: aprendizado por reforço que consiste no mapeamento entrada-saída através da interação com o ambiente, e aprendizagem não-supervisionada ou auto-organizada onde, a partir de métricas de qualidade do aprendizado ocorre a otimização dos parâmetros livres da rede. Pode, por exemplo, ser utilizada a regra de aprendizagem competitiva. Os algoritmos de aprendizado não supervisionado mais importantes são: Algoritmo de Hopfield e Mapas de Kohonen (Haykin, 2001).

## **2.3 APLICAÇÕES**

As Redes Neurais Artificiais por suas características já descritas aqui, tais como, capacidade de aprender, tolerância a falhas, entre outras, características estas inviáveis de serem obtidas através de técnicas de programação convencionais, são empregadas na resolução de alguns problemas.

Atualmente as Redes Neurais Artificiais são aplicadas em: reconhecimento de padrões, previsão de séries temporais, composição musical, processamento de sinais digitais e imagens, otimização, visão computacional, robótica e controle (Braga *et. al.*, 1998).

## **2.4 HISTÓRICO**

Um pequeno histórico da evolução das Redes Neurais Artificiais será apresentado (Russell e Norvig 1995):

- 1943: Warren Culloch e Walter Pitts introduziram a idéia de analisar a atividade neural. Sugeriram em artigos a construção de uma máquina inspirada no cérebro humano.
- 1949: Donald Hebb em seu livro “The Organization of Behavior” perseguia a idéia de que o condicionamento psicológico, presente nos animais, é propriedade de neurônios individuais. Hebb foi o pioneiro na proposição de uma lei de aprendizagem para as sinapses dos neurônios. As sinapses deveriam ter seus pesos alterados dinamicamente de acordo com o uso, ou seja, as Redes Neurais Artificiais deveriam ser capazes de aprender.
- 1951: Marvin Minsky e Dean Edwards criaram a primeira rede neural artificial e Minsky em 1954 em sua dissertação de doutorado continuou a exploração das redes neurais. As idéias apresentadas por Minsky e Edwards serviam de inspiração para pesquisas futuras.
- 1956: no Dartmouth College nascia oficialmente a Inteligência Artificial. Um workshop de dois meses foi organizado por John McCarthy, Marvin Minsky, Claude Shannon e Nathaniel Rochester no verão de 1956. O workshop em Dartmouth introduziu o termo Inteligência Artificial, campo dominado pelos próximos 20 anos por pesquisadores da Carnegie Mellon University, MIT, Stanford e IBM.
- 1957: Frank Rosenblatt inventou o “Perceptron” um estilo de rede neural composto de unidades capazes de serem treinadas, organizadas em uma única camada.
- 1960: Widrow e Hoff inventaram dispositivos semelhantes chamados “Adalines” (Adaptative Linear Network).
- 1962: Widrow desenhou máquinas multicamadas chamadas “Madalines” (Many Adalines) indo no sentido oposto das pesquisas desenvolvidas na época que tinham foco em uma única camada.

- 1969: Marvin Minsky e Seymour Papert publicaram “Perceptrons” fazendo severas críticas aos modelos Perceptron e Adaline. Tarefas linearmente separáveis como o ou-exclusivo não eram capazes de serem resolvidas pelos perceptrons. O prestígio de Minsky deu credibilidade ao seu trabalho e a comunidade científica abandonou as pesquisas na área durante a década de 70 e início da década de 80, embora alguns poucos pesquisadores continuaram a pesquisar e desenvolver trabalhos na área.
- 1982: Carpenter e Grossberg criam o modelo ART1 e depois surgem os modelos ART2, ART3, Fuzzy ART, ARTMAP, etc.
- 1986: Rumelhart, Hinton e Williams redescobrem o algoritmo de Backpropagation inventado por Bryson e Ho em 1969 e redescoberto também por Werbos em 1974, Parker e também Le Cun em 1985. Com a descoberta do Backpropagation, algoritmo que resolve o problema apresentado por Minsky e Papert, o interesse pelas pesquisas na área voltou a crescer.

### 3 SÉRIES TEMPORAIS

Uma série temporal é um conjunto de observações de uma mesma grandeza, tomadas em intervalos consecutivos e iguais (Downing & Clark, 2003; Spiegel, 1993).

São exemplos de séries temporais:

- Valores das exportações de um país;
- Valores diários da temperatura de uma cidade;
- Índices diários da Bolsa de Valores;
- Cotação de uma moeda;
- Índice pluviométrico de uma região.

A figura 3.1 ilustra graficamente uma série temporal que representa os valores em milhões de dólares das exportações do Brasil no período de 1974 a 2002.



**Fonte:** Fundação Centro de Estudos do Comércio Exterior (Funcex)

**Figura 3.6** Gráfico da série temporal que representa os valores das exportações do Brasil

### 3.1 COMPONENTES DAS SÉRIES TEMPORAIS

As séries temporais possuem quatro componentes principais (Spiegel, 1993; Stevenson, 1981):

- **Tendência:** se refere ao movimento ao longo do tempo dos dados da série temporal. Este movimento pode ser para cima ou para baixo e pode ser representado por uma reta ou uma curva.
- **Cíclico:** se refere a movimentos com certo grau de regularidade sobre a reta ou curva da tendência.
- **Sazonalidade:** se refere a variações que parecem obedecer padrões durante mesmas épocas de anos sucessivos.

- Irregular: se refere a movimentos esporádicos ou casuais.

### 3.2 ANÁLISE DAS SÉRIES TEMPORAIS

O processo de análise das séries temporais consiste na estimação da cada componente que se apresenta na série temporal. O modelo clássico de análise de séries temporais apresenta duas variantes (Stevenson, 1981):

- Modelo Aditivo: onde a série temporal é o resultado da soma de cada componente.

$$Y = T + C + S + I \quad (\text{Equação 3.1})$$

- Modelo Multiplicativo: onde a série temporal é o resultado do produto de cada componente

$$Y = T \times C \times S \times I \quad (\text{Equação 3.2})$$

Onde:

Y = valor da série temporal

T = valor do componente tendência

C = valor do componente cíclico

S = valor do componente sazonalidade

I = valor do componente irregular.

#### 3.2.1 DETERMINAÇÃO DO COMPONENTE TENDÊNCIA

A determinação do componente tendência de uma série temporal pode ser feita de quatro maneiras distintas, quais sejam (Spiegel, 1993):

- Mínimos quadrados: método que consiste em ajustar uma reta, parábola ou outra curva a um conjunto de dados de modo que, esta seja a melhor forma de ajustamento, evitando-se assim, a utilização de critérios individuais que proporcionam curvas e equações diferentes para observadores diferentes. Para um conjunto de dados a melhor curva de ajustamento é a que apresenta o menor valor da soma dos quadrados dos erros, que para cada  $x$  da equação que define a curva é a diferença entre o  $Y$  da série temporal e o valor correspondente obtido pela curva (Spiegel, 1993).

A equação da reta que, por exemplo, ajusta-se melhor a um conjunto de dados é dada por:

$$Y = a_0 + a_1x \quad (\text{Equação 3.3})$$

E as constantes  $a_0$  e  $a_1$  são determinadas pela resolução do sistema (Spiegel 1993):

$$\begin{cases} \sum Y = a_0n + a_1 \sum X \\ \sum XY = a_0 \sum X + a_1 \sum X^2 \end{cases} \quad (\text{Equação 3.4})$$

O mesmo raciocínio pode ser empregado para o ajuste de uma curva dada por uma equação de grau maior que 1.

- Sentimento: consiste no traçado de uma reta ou curva apenas pela observação do gráfico, ou seja, pelo uso de critérios individuais, que, como já foi dito, resulta em equações diferentes para observadores diferentes.
- Médias móveis: método em que os componentes cíclico, sazonal e irregular podem ser removidos restando somente a tendência. Uma média móvel de ordem  $n$  é obtida pela seqüência de médias (Spiegel 1993):

$$\frac{y_1 + y_2 + \dots + y_n}{n}, \frac{y_2 + y_3 + \dots + y_{n+1}}{n}, \frac{y_3 + y_4 + \dots + y_{n+2}}{n}, \dots$$

Após a determinação das médias móveis, uma reta é traçada passando por todos os pontos obtidos. Esta reta representa a tendência.

- Semimédias: neste método o conjunto de dados é dividido ao meio e, para cada parte a média é calculada. A tendência é determinada traçando-se uma reta passando nestes dois pontos (Spiegel, 1993).

### **3.2.2 DETERMINAÇÃO DO COMPONENTE SAZONAL**

O processo de determinação do componente sazonal consiste em determinar a maneira como os valores de um conjunto de dados variam em determinados períodos em anos sucessivos. Dispomos de quatro maneiras para realização de tal processo (Spiegel, 1993):

- Percentagem média: o conjunto de dados de um determinado período é expresso em percentagens da média do ano inteiro. Isto feito é empregado a média ou a mediana em períodos correspondentes de anos diferentes. Se a soma dos valores encontrados não for 1200%, estes devem ser ajustados multiplicando-os por  $1200/\text{soma encontrada}$ .
- Tendência ou relação percentual: o conjunto de dados é expresso em percentagens dos valores da tendência. Neste método o valor  $Y$  da equação 3.2 é dividido por  $T$ , produzindo  $CSI$ , o que representa desvantagem caso o valor de  $C$  e  $I$  forem grandes.
- Média móvel percentual: é calculada a média móvel de 12 meses para o conjunto de dados. Após isso, é calculada novamente a média móvel agora de 2 meses, chamada média móvel centrada de 12 meses. Agora o conjunto de dados pode ser expresso em função da percentagem da média móvel centrada de 12 meses. Então, calcula-se a média dos meses correspondentes.
- Elos relativos: consiste em expressar o valor de cada mês e percentagem do mês anterior. Feito isso é calculado uma média dos elos relativos dos meses correspondentes.

### **3.2.3 DETERMINAÇÃO DO COMPONENTE CÍCLICO**

De acordo com a equação 3.2, ao dividir Y por S e por T temos CI. Para suavizar o componente irregular é calculado uma média móvel de pequena ordem e ímpar. Se algum efeito cíclico existir, agora está evidente, uma vez que os outros três componentes foram eliminados ou atenuados.

### **3.2.4 DETERMINAÇÃO DO COMPONENTE IRREGULAR**

O componente irregular é obtido dividindo-se o segundo membro da equação 3.2 por S, T e C.

## **3.3 PREVISÃO**

A previsão de séries temporais pode ser obtida realizando-se os passos descritos na seção 3.2, Análise das séries temporais. Uma vez cada componente tomado individualmente podem ser feitas avaliações e considerações sobre comportamentos futuros. A junção dos componentes novamente após a análise, provê a previsão para os valores futuros da série temporal..

## 4 SIMULADOR

Para a implementação do simulador, buscou-se na literatura soluções que apresentavam os melhores resultados. Tal busca consistia em escolher um modelo de rede e um algoritmo de treinamento que, baseado em trabalhos empíricos de pesquisadores da área, obtivessem melhores resultados quando comparados, nas mesmas condições, com outras soluções para o mesmo problema.

Para tanto, foram lidos artigos, principalmente (Fernandes *et. al.*, 1995; Soto *et. al.*, 2000; Tápia *et. al.*, 2000;), que descrevem a previsão de séries temporais usando Redes Neurais Artificiais. Nos dois primeiros artigos foi utilizada a rede MLP com o algoritmo Backpropagation. Em ambos, os resultados obtidos foram comparados com técnicas tais como: Box & Jenkins e modelo estrutural, obtendo melhores resultados. Já em (Soto *et. al.*, 2000) artigo que descreve os resultados obtidos por Claver Pari Soto em sua dissertação de mestrado (Soto, 1999), foram utilizadas as redes FIR (Finite-duration Impulse Response) com o algoritmo Backpropagation alterado para atender as características da rede e a Rede Elman com o algoritmo Backpropagation convencional. Os resultados obtidos na previsão foram comparados com as técnicas de Holt-Winters, Box & Jenkins e MLP + Backpropagation obtendo-se, também, melhores resultados. As redes FIR e Elman tiveram desempenhos

parecidos, com a rede Elman obtendo maiores erros que a rede FIR, mesmo assim menores que as outras técnicas.

Ainda com o intuito de escolher a melhor solução, foi lida a dissertação de Abelém (Abelém, 1994) onde foi utilizada a rede MLP com o algoritmo Backpropagation que obteve bons resultados quando comparada com a técnica de previsão Box & Jenkins.

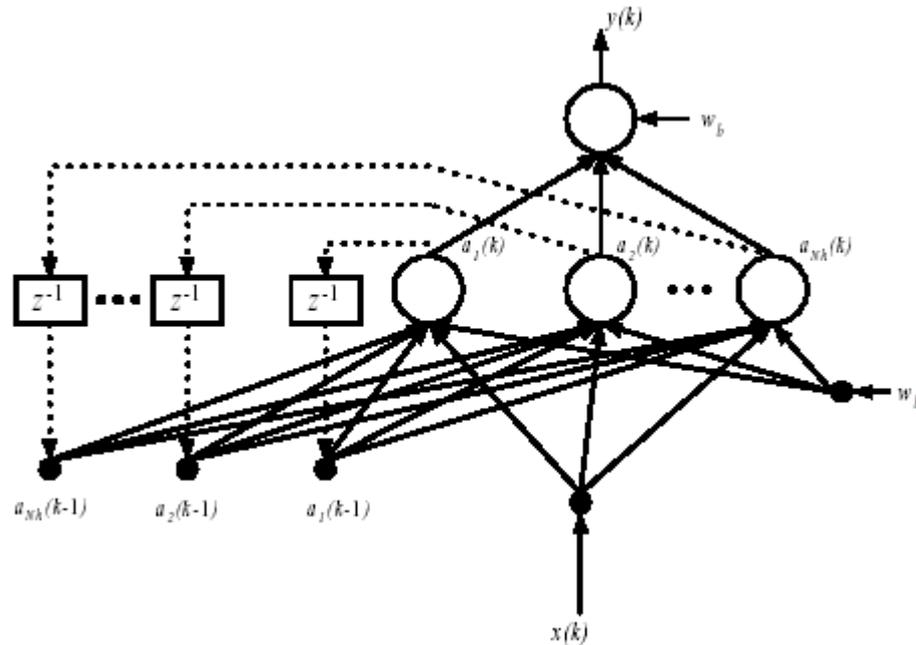
Diante do exposto, a rede Elman em conjunto com o algoritmo Backpropagation foram escolhidos para a implementação do simulador. Os motivos para tal escolha foram os resultados obtidos em (Soto, 1999, Soto *et. al.*, 2000), melhores que as técnicas tradicionais. E a facilidade de implementação em comparação com a rede FIR que necessita alterar o algoritmo de aprendizado e calcular parâmetros inerentes a rede, tais como: comprimento de filtros e número de neurônios.

#### **4.1 REDE ELMAN**

A Rede Elman, ou rede recorrente simples, foi descrita em (Elman, 1990) e é uma rede onde a realimentação capacita a realização de tarefas que se estendem no tempo (Haykin, 2001).

A Rede Elman, ilustrada pela figura 4.1, é uma rede recorrente onde a realimentação se dá da saída de cada neurônio da camada oculta para todos neurônios da mesma camada. Uma camada, chamada Camada de Contexto, também oculta, composta de Unidades de Contexto simula a memória da rede. O processamento da rede consiste nos eventos: No instante  $t$  (inicial) o sinal é propagado pela rede e as unidades de contexto, inicializadas com a saída da camada oculta com o valor 0, não influenciarão na saída da rede, ou seja, na primeira iteração a rede se comportará como uma rede feed-forward. Ainda na primeira iteração os neurônios ocultos ativarão os neurônios da camada de contexto e esses armazenarão a saída desta iteração que será utilizada no próximo ciclo. O algoritmo Backpropagation é então aplicado para a correção dos pesos sinápticos, com exceção as sinapses recorrentes que são fixas em 1. No instante  $t+1$  o processo é repetido. A diferença é

que a partir de agora os neurônios ocultos serão ativados pelas unidades de entrada e pelas unidades de contexto que possuem o valor de saída dos neurônios ocultos no instante  $t$  (Elman, 1990).



**Figura 4.7 Rede de Elman**

A equação 4.1 define a saída de um neurônio da camada oculta.

$$a_i(t) = f\left(\sum w_{ei} x_i(k) + \sum a_i(k-1) + w_{bi}\right) \quad (\text{Equação 4.1})$$

onde:  $a_i(t)$ , a saída do neurônio oculto  $i$ , no instante  $t$ , dada pela função de ativação  $f$  que recebe a soma do produto  $w_{ei}$ , peso sináptico associado do neurônio  $e$  com o neurônio  $i$  multiplicado pela entrada  $x_i(t)$ , ou a saída do neurônio  $e$ ,  $a_e$ , somado ao somatório das saídas dos neurônios ocultos no instante  $t - 1$ , somado ainda, a bias.

## 4.2 ALGORITMO BACKPROPAGATION

O algoritmo Backpropagation é um algoritmo de aprendizado supervisionado baseado na correção de erros. Foi publicado por Rumelhart em 1986 no livro Parallel

Distributed Processing, embora tenha sido proposto por pesquisadores como Bryson e Ho em 1969, Werbos em 1974, Parker em 1985 e Minsky e Papert em 1988 para diversos propósitos (Braga *et. al.*, 1998; Russel & Norvig, 1995).

O objetivo do algoritmo Backpropagation é encontrar na superfície de erro um mínimo global, ou seja, valores para os pesos sinápticos que minimizam o erro da rede.

Como um algoritmo de aprendizado supervisionado, o Backpropagation utiliza pares entrada-saída para ajustar os pesos da rede. A frequência com que estes ajustes são feitos depende do tipo de treinamento, que pode ser: por padrão (on-line) onde a atualização é realizada após cada exemplo apresentado; por ciclo (batch ou lote) onde a atualização é realizada após todos os exemplos do conjunto de treinamento serem apresentados a rede (Braga *et. al.*, 1998; Haykin, 2001).

Os critérios de parada, embora ainda não existam critérios bem sucedidos para terminar o processo de aprendizado, podem ser: quando, percorrendo-se a superfície de erro, for encontrado um mínimo; a variação do erro for infinitamente pequena sendo insuficiente para melhorar a performance da rede; e ao aplicar o conjunto de treinamento um número determinado de vezes, ou seja, baseado no número de épocas. É comum se observar na prática a combinação dos critérios acima (Braga *et. al.*, 1998; Haykin, 2001).

#### **4.2.1 O ALGORITMO**

A tabela abaixo apresenta a notação utilizada na demonstração do algoritmo.

Notação	Significado
$w_{ij}(n)$	Peso sináptico da saída do neurônio $j$ para o neurônio $i$ , na iteração $n$ .
$y_i(n)$	Saída do neurônio $i$ na iteração $n$ .
$v_i(n)$	Sinal de ativação para o neurônio $i$ na iteração $n$ .
$d_i(n)$	Saída esperada para o neurônio $i$ na iteração $n$ .
$e_i(n)$	Erro do neurônio $i$ na iteração $n$ .
$f(\cdot)$	Função de ativação
$f'(\cdot)$	Derivada da função de ativação
$\xi$	Erro quadrático médio
$\eta$	Taxa de aprendizagem.
$\delta$	Gradiente local

**Tabela 4.1 Notação utilizada na demonstração do algoritmo Backpropagation.**

O erro na saída do neurônio é dado por:

$$e_j(n) = d_j(n) - y_j(n) \quad (\text{Equação 4.2})$$

E o objeto que se deseja minimizar é a função custo, dada por:

$$\xi = \frac{1}{2} e_j^2(n) \quad (\text{Equação 4.3})$$

Para minimizar o erro, o algoritmo Backpropagation aplica uma correção  $\Delta w_{ji}$  ao peso sináptico de acordo com a Regra Delta de Widrow e Hoff (Equação 4.4). Para fazer esta minimização é utilizado o gradiente descendente que é dado pela derivada  $\partial \xi(n) / \partial w_{ji}(n)$  (Haykin, 2001):

$$\Delta w_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}(n)} \quad (\text{Equação 4.4})$$

A derivada  $\partial \xi(n) / \partial w_{ji}(n)$  determina a direção da busca no espaço de pesos para o peso  $w_{ji}$ . Pela regra da cadeia, esta derivada pode ser expressa em (Haykin, 2001):

$$\frac{\partial \xi(n)}{\partial w_j(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (\text{Equação 4.5})$$

Derivando as equações 4.3 em relação a  $e_j(n)$ ; 4.2 em relação a  $y_j(n)$  temos:

$$\frac{\partial \xi(n)}{\partial e_j(n)} = \frac{\partial 1/2 e_j^2(n)}{\partial e_j(n)} = e_j(n) \quad (\text{Equação 4.6})$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = \frac{\partial d_j(n) - y_j(n)}{\partial y_j(n)} = -1 \quad (\text{Equação 4.7})$$

Derivando  $\partial y_j(n)/\partial v_j(n)$  e  $\partial v_j(n)/\partial w_{ji}(n)$ , temos

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \frac{\partial f(v_j(n))}{\partial v_j(n)} = f'(v_j(n)) \quad (\text{Equação 4.8})$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = \frac{\partial \sum w_{ji}(n) y_j(n)}{\partial w_{ji}(n)} = y_j(n) \quad (\text{Equação 4.9})$$

Substituindo-se as equações 4.6, 4.7, 4.8 e 4.9 no segundo membro da equação 4.5

temos:

$$\frac{\partial \xi(n)}{\partial w_j(n)} = -e_j(n) f'(v_j(n)) y_j(n) \quad (\text{Equação 4.10})$$

Portanto, usando a equação 4.10 em 4.4,  $\Delta w_{ji}(n)$  é dado por:

$$\Delta w_{ji}(n) = \eta e_j(n) f'(v_j(n)) y_j(n) \quad (\text{Equação 4.11})$$

Podemos definir o gradiente local  $\delta_j(n)$  como:

$$\delta_j(n) = e_j(n) f'(v_j(n)) \quad (\text{Equação 4.12})$$

E a equação 4.11 pode ser reescrita:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_j(n) \quad (\text{Equação 4.13})$$

Como o termo  $e_j(n)$  é fator chave no cálculo para o ajuste dos pesos, existem duas situações que devem ser tratadas (Haykin, 2001):

1ª Situação: O neurônio está localizado na camada de saída. Como para um neurônio de saída o sinal de erro  $e(n)$  é calculado diretamente da resposta esperada. Assim o novo peso é dado por:

$$w(n+1) = w(n) + \eta \delta_j(n) y_j(n) \quad (\text{Equação 4.14})$$

2ª Situação: O neurônio está localizado na camada oculta. Não existe uma resposta esperada para este neurônio. Então, através de recursão, o sinal de erro é determinado em função dos neurônios da camada seguinte aos quais este neurônio está conectado.

O gradiente local, definido pela derivada parcial  $\partial \xi(n) / \partial v_j(n)$  pode ser reescrito pela regra da cadeia como:

$$\delta_j(n) = - \frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \quad (\text{Equação 4.15})$$

Utilizando a equação 4.8 na equação 4.15 temos

$$\delta_j(n) = - \frac{\partial \xi(n)}{\partial y_j(n)} f'(v_j(n)) \quad (\text{Equação 4.16})$$

O termo  $\xi(n)$  da equação 4.16 é igual ao erro médio quadrado de todos os neurônios da camada k, camada de saída (Haykin, 2001):

$$\xi(n) = \frac{1}{2} \sum_k e_k^2(n) \quad (\text{Equação 4.17})$$

Portanto, diferenciando a equação 4.17 temos:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \quad (\text{Equação 4.18})$$

Aplicando a regra da cadeia mais uma vez para diferenciar a equação 4.18 obtemos:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (\text{Equação 4.19})$$

Usando a equação 4.2 na equação 4.19 podemos calcular a derivada  $\partial e_k(n)/\partial v_k(n)$ :

$$\frac{\partial (d_k(n) - y_k(n))}{\partial v_k(n)} = \frac{\partial (d_k(n) - f(v_k(n)))}{\partial v_k(n)} = -f'(v_k(n)) \quad (\text{Equação 4.20})$$

Para o neurônio k podemos definir  $v_k(n)$  como:

$$v_k(n) = \sum w_{kj}(n) y_j(n) \quad (\text{Equação 4.21})$$

Assim podemos diferenciar  $\partial v_k(n)/\partial y_j(n)$ :

$$\frac{\partial v_k(n)}{\partial y_j(n)} = \frac{\partial (\sum w_{kj}(n) y_j(n))}{\partial y_j(n)} = w_{kj}(n) \quad (\text{Equação 4.22})$$

Substituindo as equações 4.20 e 4.22 em 4.19, temos:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} = \sum_k e_k - f'(v_k(n)) w_{kj}(n) \quad (\text{Equação 4.23})$$

Reescrevendo a equação 4.23 em função da equação 4.12:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = -\sum_k \delta_k(n) w_{kj}(n) \quad (\text{Equação 4.24})$$

Substituindo a equação 4.24 em 4.16 obtemos a fórmula do gradiente local para neurônios da camada oculta (Haykin, 2001):

$$\delta_j(n) = f'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (\text{Equação 4.25})$$

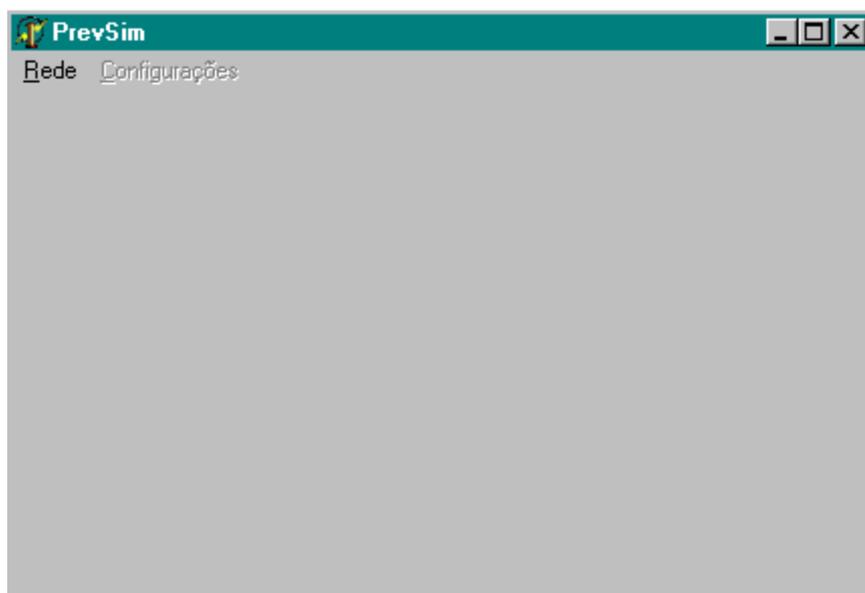
O algoritmo Backpropagation, representado pelas equações descritas anteriormente pode ser resumido em dois tipos de processamento (Haykin, 2001):

- Processamento para frente, onde o sinal é propagado e aparece na camada de saída;
- Processamento de retropropagação, onde o erro é corrigido camada a camada, começando pela camada de saída até a primeira camada oculta.

### 4.3PREVSIM

Pensando em integrar todas as tarefas envolvidas no processo de simulação foi desenvolvido, em Delphi, um simulador – PrevSim – que em um único módulo apresenta: criação e configuração da rede e parâmetros de treinamento, treinamento da rede, validação dos resultados.

A figura 4.2 apresenta a tela inicial do simulador.

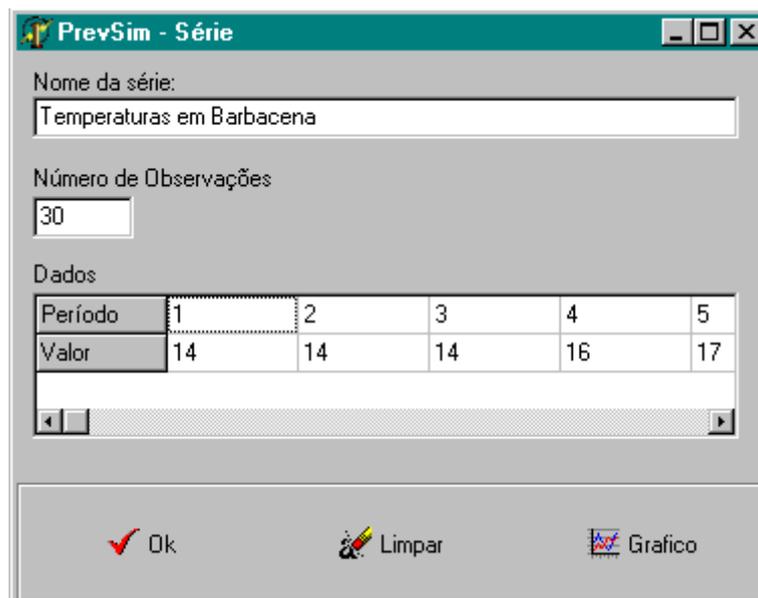


**Figura 4.8 Tela principal do Prevsim.**

A correta utilização do simulador prevê a seqüência de passos que devem ser seguidos: inserção da série, configuração da rede, treinamento e validação do treinamento. Nas seções seguintes será descrita a utilização do simulador para cada um desses passos.

#### **4.3.1 INSERÇÃO DA SÉRIE**

Primeiro passo na utilização do simulador, a inserção da série temporal consiste em: inserir um nome para tal série, o número de dados que a compõem e a série propriamente dita, que é composta de dois dados: o período e o valor correspondente. A figura 4.3 ilustra a janela onde a série é inserida.



Nome da série:  
Temperaturas em Barbacena

Número de Observações:  
30

Dados

Período	1	2	3	4	5
Valor	14	14	14	16	17

Ok    Limpar    Gráfico

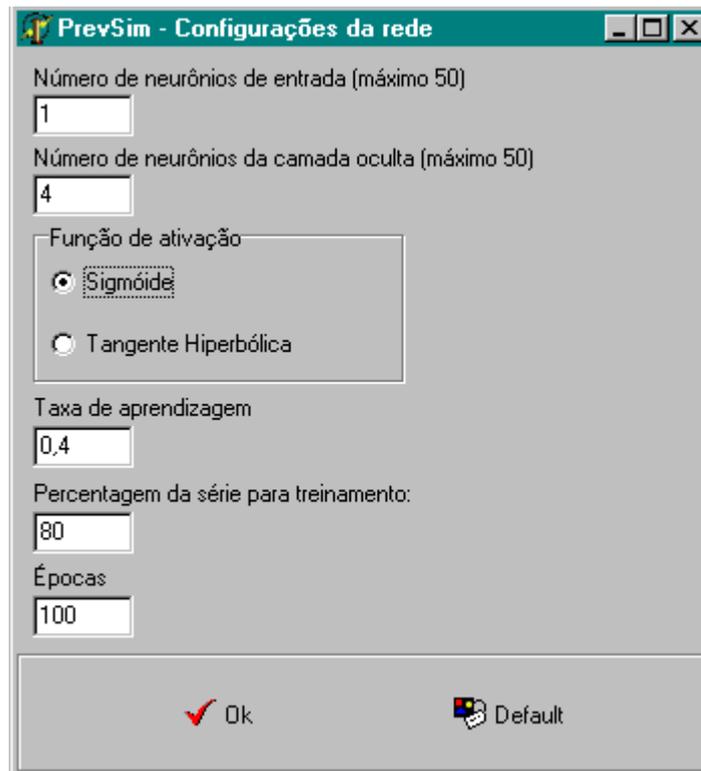
**Figura 4.9 Tela Inserir Série.**

### 4.3.2 CONFIGURAÇÃO DA REDE

Imediatamente após a inserção da série, uma janela se abrirá, figura 4.4, permitindo a configuração da rede. A configuração da rede consiste em estabelecer valores para os parâmetros da rede, quais sejam:

- Número de neurônios de entrada: permite estabelecer o número de neurônios de entrada da rede. Pode variar entre 1 e 50.
- Número de neurônios da camada oculta: permite estabelecer o número de neurônios da camada oculta, como para os neurônios da entrada, variam entre 1 e 50. No simulador foi implementada apenas uma camada oculta, pois esta é uma característica da Rede Elman (Soto, 1999).
- Função de ativação: permite a escolha entre a função Sigmóide e Tangente Hiperbólica para a ativação de todos os neurônios da camada oculta e de saída.
- Taxa de aprendizagem: permite definir a taxa com a qual os pesos sinápticos serão alterados.

- Percentagem da série para treinamento: define como será dividido a série: qual percentagem dos dados será utilizada no conjunto de treinamento, enquanto o restante, será usado no validação.
- Épocas: permite definir o número de vezes que o conjunto de treinamento será apresentado a rede.



**Figura 4.10 Tela Configurações**

O simulador permite ainda a utilização de um conjunto de parâmetros definidos como default: número de neurônios de entrada: 1; número de neurônios na camada oculta: 4; função de ativação: Sigmóide; Taxa de aprendizagem: 0,4; Percentagem da série para treinamento: 80%, e Épocas: 100.

### 4.3.3 TREINAMENTO

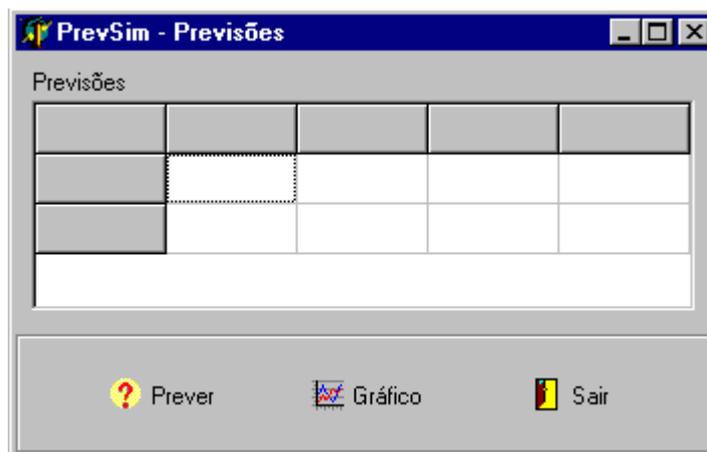
O treinamento consiste na aplicação do algoritmo Backpropagation para a correção dos pesos, cuja frequência é após a apresentação de cada par entrada-saída. O critério de parada é o número de épocas, definido pelo usuário.

Ao final do treinamento, a janela ilustrada pela figura 4.5 exibirá o número de épocas, a taxa de aprendizagem utilizada para o treinamento. Ainda apresenta o erro, MAPE (Mean Absolute Percentage Error), Erro Percentual Médio, uma métrica de avaliação do aprendizado, dado pela equação 4.26 (Soto, 1999):

$$MAPE = \frac{\sum_{i=1}^n \left| \frac{Saída\_esperada - Saída\_obtida}{Saída\_esperada} \right|}{n} \times 100 \quad (\text{Equação 4.26})$$

### 4.3.4 VALIDAÇÃO

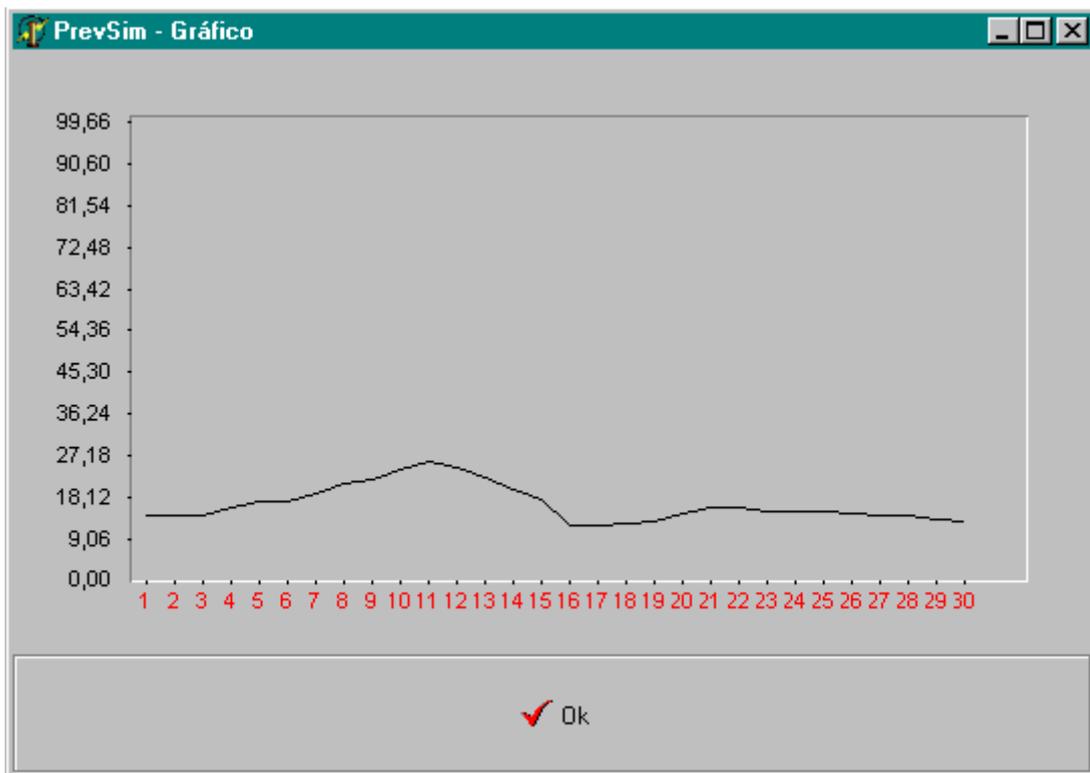
A função de validação permite que o usuário faça previsões com a rede já treinada e compare com o conjunto de testes que ficou reservado no momento de configuração. A figura 4.5 ilustra a janela onde são apresentados os resultados.



**Figura 4.11 Tela Previsões**

### 4.3.5 FUNÇÕES EXTRAS

No simulador foram, ainda, implementadas algumas funções para facilitar a previsão de séries temporais tais como: gerar gráficos, figura 4.6, no momento em que a série é inserida e quando é feita a validação do treinamento; realizar o pré-processamento e pós-processamento de forma transparente para o usuário e permite salvar e recuperar séries que já foram trabalhadas.



**Figura 4.12 Tela Gráfico**

Com relação ao pré-processamento, atividade realizada antes do treinamento, consiste em normalizar os dados, ou seja, são aplicadas transformações nas entradas de modo que a média destas seja igual a 0 (Haykin, 2001). No simulador foi utilizada a transformação descrita pela equação 4.27, para que a série tenha média 0 e variância 1 (Soto, 1999):

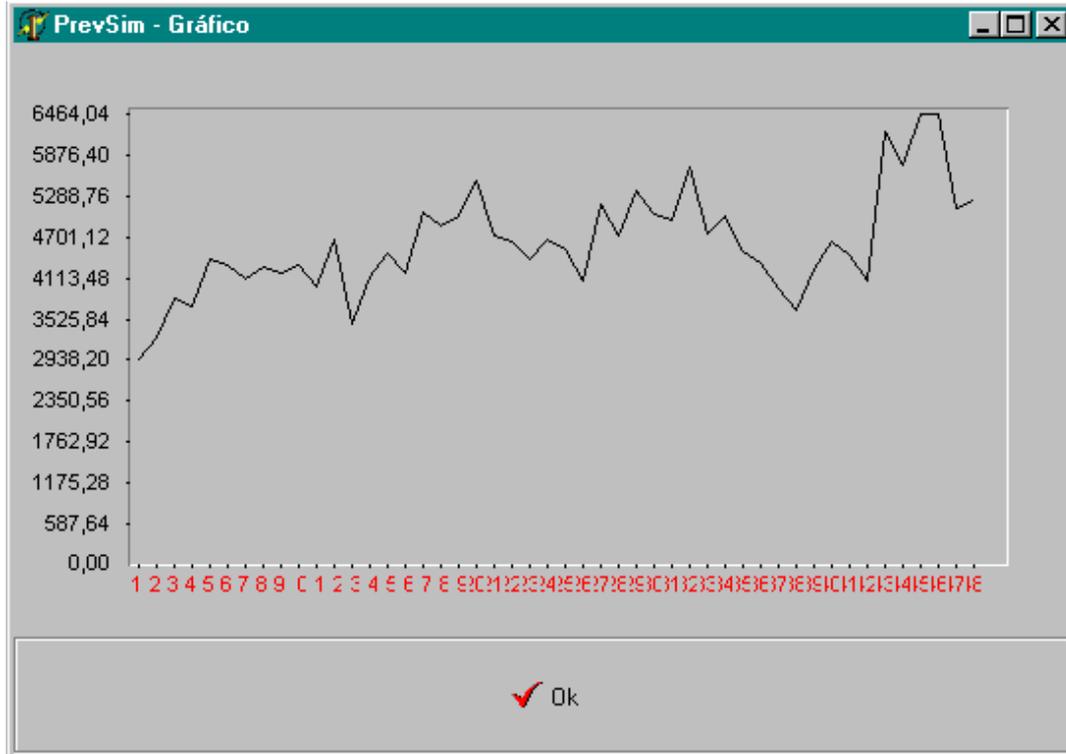
$$x_i = \frac{s_t - \bar{s}}{\sigma_s} \quad (\text{Equação 4.27})$$

O pós-processamento consiste em transformar a saída da rede na série prevista e é dada pela equação 4.28 (Haykin, 2001; Soto, 1999):

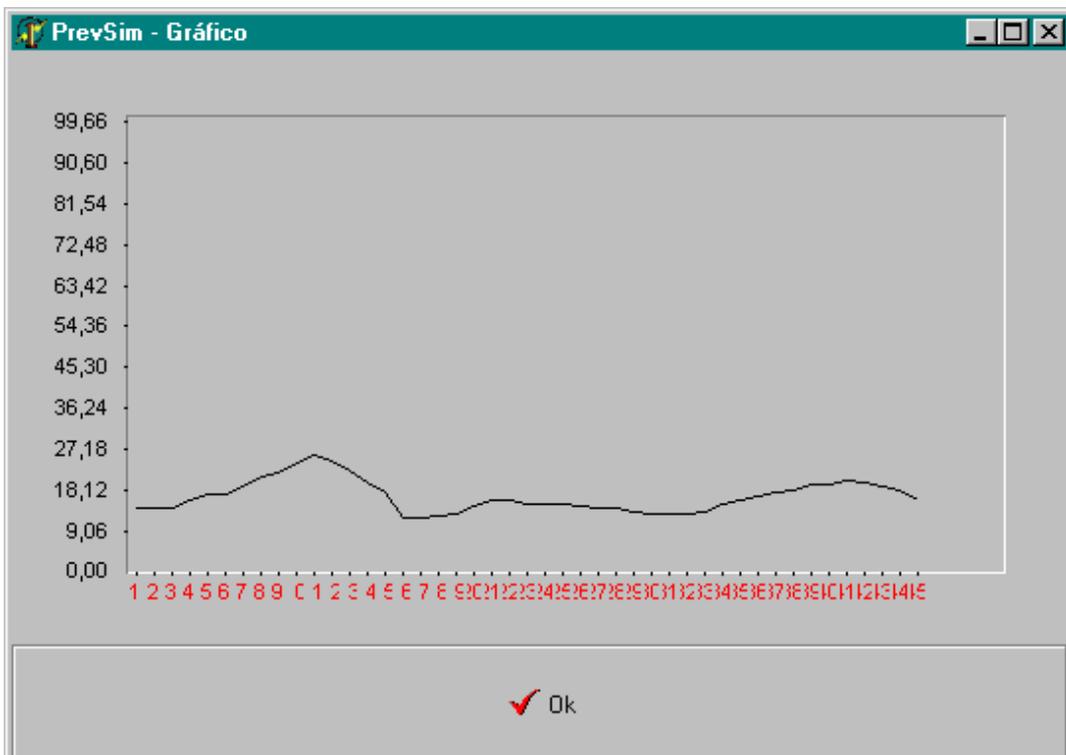
$$s = x_i \sigma_s + \bar{s} \quad (\text{Equação 4.28})$$

## 5 APLICAÇÕES

Com o intuito de avaliar o desempenho do simulador na previsão, foram utilizadas duas séries temporais reais: série 1, Exportações Brasil no período de 01/1999 a 12/2002, figura 5.1, perfazendo um total de 48 observações obtidas em (IPEADATA, 2003); e a série 2, série Temperatura em Barbacena nos dias de 31/08/2003 a 02/09/2003 com 45 registros obtidos pela Aeronáutica no Aeroporto de Barbacena, ilustrada pela figura 5.2. As duas séries são apresentadas no Anexo A.



**Figura 5.13 Gráfico da série Exportações Brasil**



**Figura 5.14 Gráfico da série Temperatura em Barbacena**

## 5.1 TESTES

As duas séries foram inseridas no simulador e foi realizada uma bateria de testes para cada uma, buscando determinar as configurações que obtivessem os melhores resultados, ou seja que minimizassem o erro.

O resultado dos testes para série 1 é apresentado na tabela 5.1, onde são vistos a arquitetura da rede, função de ativação utilizada no treinamento, taxa de aprendizagem, número de épocas usadas no treinamento e o erro obtido. A configuração que obteve o melhor resultado para a série 1, na última linha em negrito, possui arquitetura 18-36-1, ou seja, 18 neurônios na entrada, 36 neurônios na camada oculta e 1 neurônio na camada de saída; função de ativação Sigmóide, taxa de aprendizagem 0.5, 1000 épocas no treinamento e obtendo um erro de 293,8949.

Arquitetura	Função de ativação	Taxa de aprendizagem	Épocas	Erro (MAPE)
20-20-1	Sigmóide	0.5	500	551,9465
20-20-1	Sigmóide	0.5	1000	390,7700
12-24-1	Sigmóide	0.5	1000	407,3139
12-36-1	Sigmóide	0.5	1000	331,0970
<b>18-36-1</b>	<b>Sigmóide</b>	<b>0.5</b>	<b>1000</b>	<b>293,8949</b>

**Tabela 5.2 Testes com a série Exportações Brasil**

Como para a série 1, a tabela 5.2 apresentada os resultados obtidos no treinamento da série 2. A configuração que obteve o menor erro foi: arquitetura 12 neurônios de entrada, 24 neurônios na camada oculta e 1 neurônio na camada de saída; função de ativação sigmóide; taxa de aprendizagem 0.4; épocas 300 obtendo um erro de 12,6305.

Arquitetura	Função de ativação	Taxa de aprendizagem	Épocas	Erro (MAPE)
6-12-1	Sigmóide	0.4	300	17,4148
6-18-1	Sigmóide	0.4	300	16,5830
10-15-1	Sigmóide	0.4	300	15,2652
10-20-1	Sigmóide	0.4	300	14,6483
<b>12-24-1</b>	<b>Sigmóide</b>	<b>0.4</b>	<b>300</b>	<b>12,6305</b>

**Tabela 5.3 Testes com a série Temperatura em Barbacena**

## 5.2 AVALIAÇÕES

Pode-se notar pelos dados das tabelas acima, que o simulador não apresenta bons resultados na previsão, produzindo erros percentuais médios na ordem de centenas para as melhores configurações de rede para a primeira série, enquanto que, para a segunda, apresenta erros cerca de 10 vezes menores para os melhores casos.

Uma explicação para tal diferença pode ser a magnitude dos dados de entrada, diferentes nas duas séries. Na primeira série os dados de entrada estão na faixa de 1000 a 9999, e na segunda, de 10 a 99

Outra explicação seria a diferença entre os maiores e menores dados presentes nas séries na primeira, o menor dado é 2949 e o maior 6492. Para a segunda série o menor e o maior dados são 12 e 26, respectivamente. Embora a mesma normalização dos dados de entrada foi empregada nas duas séries: os dados foram transformados para terem média 0 e variância 1.

Na literatura encontra-se casos de sucesso na previsão de séries temporais, com erros percentuais na faixa de 1 a 5% como em (Soto, 1999) erros esses, menores que técnicas tradicionais da Estatística, o que comprova a qualidade de das Redes Neurais Artificiais na previsão de Séries Temporais.

## 6 CONCLUSÕES

Embora o desempenho do simulador implementado não tenha sido o esperado, acredita-se que o resultado do trabalho desenvolvido ao longo do tempo, que culminou na implementação do simulador, foi de grande valia.

Foram feitos estudos na literatura buscando subsídios para a escolha de modelos de rede e algoritmos de treinamento que em conjunto obtivessem bons resultados. Nestes estudos constatou-se que a tarefa de implementação de um simulador não é uma tarefa simples, é uma tarefa que depende muito da experiência do implementador, pois muitas das configurações e escolhas a serem tomadas não possuem metodologias definidas ou aceitas pela comunidade científica. O que se encontra são trabalhos de pesquisadores que obtiveram sucesso com alguma configuração ou técnica, mas que não podem ser estendidas para todos os casos.

O estudo desenvolvido neste trabalho possibilitou um maior conhecimento das Redes Neurais Artificiais, dos seus conceitos básicos , técnicas e principais problemas, como também das Séries Temporais.

As principais dificuldades encontradas se referem às configurações encontradas na implementação do simulador, tais como: escolha das funções de ativação para cada neurônio, como também de seus coeficientes, escolha das transformações realizadas no pré-processamento dos dados de entrada adequadas os dados disponíveis.

## **6.1 TRABALHOS FUTUROS**

Alguns rumos que podem ser seguidos no sentido de dar continuidade ao trabalho desenvolvido destacam-se:

- Pelas dificuldades encontradas, elencadas acima, pode se proceder um estudo de como definir estes parâmetros de forma a melhorar o desempenho do simulador.
- Usar outras configurações para a rede incluindo camadas ocultas adicionais;
- Utilizar algoritmos genéticos com o intuito de escolher num universo de configurações da rede aquela que apresenta os melhores resultados.
- Como neste trabalho foi utilizado o algoritmo Backpropagation padrão, pode-se utilizar técnicas como taxa de aprendizagem dinâmica, termo de momentum e outras que melhorem o desempenho do algoritmo.
- Utilizar outras métricas para avaliar o desempenho do simulador.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABELÉM, Antônio Jorge Gomes. **Redes Neurais Artificiais na previsão de séries temporais**. 1994. Tese (Mestrado em Engenharia Elétrica) - Curso de Mestrado em Engenharia Elétrica, PUC-Rio, Rio de Janeiro.
- BITTENCOURT, Guilherme. **Inteligência Artificial. Ferramentas e Teorias**. Florianópolis, SC: Editora da UFSC, 2001.
- BRAGA, Antônio de Pádua, CARVALHO, André Ponde de Leon F., LUDERMIR, Teresa Bernarda. **Fundamentos de Redes Neurais Artificiais**. 11ª Escola de Computação. Rio de Janeiro, RJ
- DOWNING, Douglas e CLARK, Jeffrey. **Estatística aplicada**. São Paulo, SP: Saraiva, 2003.
- ELMAN, Jeffrey L. **Finding Structure in Time**. Cognitive Science, 14, 179-211. 1990.
- FERNANDES, Luiz Gustavo L., PORTUGAL, Marcelo Savino, NAVAUX, Philippe O. A. **Previsão de Séries de Tempo: Redes Neurais Artificiais e Modelos Estruturais**. Disponível em: <http://www8.ufrgs.br/ppge/pdf/msp/95-09.pdf>. Acesso em: 16 de junho de 2003.
- HAYKIN, Simon. **Redes Neurais. Princípios e prática**. Porto Alegre, RS: Bookman, 2001.
- IPEADATA. **IPEADATA**. <<http://www.ipeadata.gov.br>> Acesso em: 14 de novembro de 2003
- RICH, Elaine. **Inteligência Artificial**. São Paulo, SP: McGraw-Hill, 1988.
- RUSSELL, Stuart Jonahan e NORVIG. Peter. **Artificial Intelligence: a modern approach**. New Jersey: Prentice-Hall Inc., 1995

SOTO, Claver Pari. **Redes Neurais Temporais para o tratamento de sistemas variantes no tempo**. 1999. Tese (Mestrado em Engenharia Elétrica: Sistemas de Computação) - Curso de Mestrado em Engenharia Elétrica, PUC-Rio, Rio de Janeiro.

SOTO, Claver Pari, VELLASCO, Marley Maria Bernardes Rebuzzi, PACHECO, Marco Aurélio Cavalcanti. **Previsión en Series Temporales de Carga Eléctrica usando Redes Neuronales Temporales**. Disponível em: [www.ica.ele.puc-rio.com.br/publicacoes/download/cnf\\_0118.pdf](http://www.ica.ele.puc-rio.com.br/publicacoes/download/cnf_0118.pdf). Acesso em: 25 de junho de 2003.

SPIEGEL, Murray R. **Estatística**. São Paulo, SP: Makrin Books, 1993

STEVENSON, William J. **Estatística aplicada à Administração**. São Paulo, SP: Harper & Row do Brasil, 1981

TÁPIA, Milena, ROISENBERG, Mauro, BARRETO, Jorge Muniz. **Aplicação de Redes Neurais Diretas na previsão de preços de ovos**. Disponível em: [http://200.201.9.33/artigos/pdf/info\\_022.pdf](http://200.201.9.33/artigos/pdf/info_022.pdf). Acesso em: 30 de março de 2003.

YIODA, Eduardo Masato. **Inteligência Computacional no projeto automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas**. 2000. Tese (Mestrado em Engenharia Elétrica) - Curso de Mestrado em Engenharia Elétrica, UNICAMP, Campinas.

## ANEXO A – SÉRIES TEMPORAIS UTILIZADAS NA APLICAÇÃO

Série Exportações Brasil:

Mês	Valor	Mês	Valor	Mês	Valor	Mês	Valor
01/1999	2949	01/2000	3453	01/2001	4538	01/2002	3972
02/1999	3267	02/2000	4123	02/2001	4083	02/2002	3658
03/1999	3829	03/2000	4472	03/2001	5167	03/2002	4261
04/1999	3707	04/2000	4181	04/2001	4730	04/2002	4641
05/1999	4386	05/2000	5063	05/2001	5367	05/2002	4441
06/1999	4313	06/2000	4861	06/2001	5042	06/2002	4079
07/1999	4117	07/2000	5003	07/2001	4965	07/2002	6224
08/1999	4277	08/2000	5519	08/2001	5727	08/2002	5751
09/1999	4187	09/2000	4724	09/2001	4755	09/2002	6492
10/1999	4304	10/2000	4638	10/2001	5003	10/2002	6474
11/1999	4002	11/2000	4390	11/2001	4500	11/2002	5126
12/1999	4673	12/2000	4659	12/2001	4346	12/2002	5243

**Tabela A.1 Valores da série Exportações Brasil.**

Os valores da tabela acima estão em milhões de dólares (US\$).

## Série Temperatura em Barbacena:

Medição	Temperatura (°C)	Medição	Temperatura (°C)	Medição	Temperatura (°C)
1	14	16	12	31	13
2	14	17	12	32	13
3	14	18	12,5	33	13,4
4	16	19	13	34	14,8
5	17	20	14,5	35	16
6	17	21	16	36	16,7
7	19	22	16	37	17,5
8	21	23	15,2	38	18,2
9	22	24	14,8	39	19,2
10	24	25	15	40	19,4
11	26	26	14,5	41	20,4
12	24,5	27	14,2	42	19,6
13	22,5	28	14	43	18,9
14	19,6	29	13,4	44	18
15	17,4	30	13	45	15,8

**Tabela A.2 Valores da série Temperatura em Barbacena.**