

# PREVISÃO DO PREÇO DA SOJA UTILIZANDO REDES NEURAIS

<sup>1</sup>William Hajime Yonenaga e <sup>2</sup>Reginaldo Santana Figueiredo

<sup>1</sup>Programa de Pós-Graduação em Engenharia de Produção, <sup>2</sup> Departamento de Engenharia de Produção

Universidade Federal de São Carlos - Rod. Washington Luis, km 235

CEP 13565-905 - São Carlos – SP – tel.: (016) 260 8111

william@ldi.dm.ufscar.br, santana@power.ufscar.br

**ABSTRACT:** *This paper presents a neural network modelling study to time series prediction. Owning a database containing the price of soya sack taken monthly, it was built a neural network that predict the price of soya sack in the next month.*

**KEYWORDS:** *neural networks, simulation, prediction.*

**RESUMO:** Este artigo apresenta um estudo de modelagem de redes neurais para previsão de séries temporais. A partir de uma base de dados contendo o preço da saca de soja tomado mensalmente, foi construído uma rede neural que prevê o preço da saca de soja no mês posterior.

## 1.INTRODUÇÃO

As redes neurais são um ramo da inteligência artificial que têm apresentado bons resultados em tarefas que envolvam previsão e classificação. Elas possuem a capacidade de generalização, adaptação, correlação e aprendizado, o que facilita o tratamento com séries temporais. As redes neurais podem extrair as características mais importantes da série, como sazonalidade, tendência, etc. Elas também correlacionam outras variáveis que se mostram importantes para a série. Existem vários exemplos de aplicações de redes neurais para previsão das séries temporais, como: previsão de números de passageiros em aviões, previsão de índices de mercado, previsão de vazão de rios, previsão de vendas de produtos no comércio, previsão de cotação de moeda, etc. Com relação a problemas de

classificação, as redes neurais têm sido usadas em várias aplicações, como: análise de crédito, solvência de empresas, reconhecimento de caracteres, etc.

O objetivo deste trabalho é ilustrar o uso de redes neurais para previsão do preço da saca de soja no estado de São Paulo. Os dados para se realizar os experimentos foram obtidos na publicação AGRIANUAL 96. Este sistema é uma alternativa à abordagem estatística e constitui uma ferramenta para a tomada de decisões. A partir de uma série histórica é possível montar uma rede neural que fornece, com alto grau de certeza, o valor futuro da série. No caso do preço da saca de soja, apresentado neste trabalho, tal informação é importante para os empresários, pois tendo uma boa previsão para o preço no mês posterior, eles podem tomar a decisão apropriada, como vender, comprar ou estocar, por exemplo. Serve também para traçar o comportamento da série de preços no mercado futuro. De posse de uma informação deste tipo, o empresário terá uma vantagem competitiva em relação a seus concorrentes.

Para a realização deste trabalho, foi utilizado o simulador SNNS (Stuttgart Neural Network Simulator) para treinar as redes. Ele é capaz de simular vários tipos de arquiteturas e algoritmos e possui recursos poderosos para obter informações da rede.

## 2. HISTÓRICO

Os primeiros trabalhos na área de redes neurais foram feitos por Warren McCulloch, um psicólogo e Walter Pitts, um matemático. Eles introduziram os conceitos de combinação de neurônios, pesos e *thresholds* (MACCULLOCH e PITTS, 1943). Em 1949, Donald Hebb, um psicólogo da McGill University, definiu a primeira lei de aprendizagem para as redes neurais. Sua premissa era de que se dois neurônios fossem ativados simultaneamente, então o resultado da conexão entre eles deveria ser reforçado (HEBB, 1949).

Em 1958, Frank Rosenblatt introduziu e desenvolveu uma classe de redes neurais denominada *perceptron* (ROSENBLATT, 1962). O tipo mais comum consiste de uma camada de entrada conectada por ligações com pesos a outros neurônios. Os pesos das ligações eram atualizados por um método de ajuste iterativo. Mais tarde descobriu-se que os *perceptrons*, em determinadas situações de aprendizado iterativo, eram limitados. Em 1960, Bernard Widrow e Marcian Hoff desenvolveram o

aprendizado pela regra delta, que ajusta os pesos para reduzir a diferença entre a saída da rede e a saída desejada. Esta diferença é conhecida como erro médio quadrático. Esta regra melhorou a habilidade da rede em generalizar e foi a precursora da regra *backpropagation*, um dos algoritmos mais utilizados.

Teuvo Kohonen, da Helsinki University of Technology, iniciou suas pesquisas com redes neurais em 1972. Seus trabalhos estão relacionados com memória associativa e mapas de características auto-organizáveis (KOHONEN, 1982). Estes tipos de redes têm sido aplicados no reconhecimento de voz, na solução do problema do caixeiro viajante e na composição musical. James Anderson, da Brown University, desenvolveu redes neurais com memória associativa que têm sido aplicadas em diagnósticos médicos e tabelas de aprendizado de multiplicação.

O método *backpropagation* (BEALE, 1994), desenvolvido na década de 80 por vários pesquisadores, sanou as limitações dos *perceptrons*. Este algoritmo é capaz de treinar uma rede multi-camada e ele repropaga os erros da camada de saída para as camadas anteriores. Kunihiro Fukushima e sua equipe do NHK Laboratories em Tokyo têm desenvolvido redes neurais especializadas em reconhecimento de caracteres (FUKUSHIMA, 1988).

### 3. MODELAGEM DE UMA REDE NEURAL

As redes neurais são novos modelos de computação com características particulares, como: capacidade de aprender, não linearidade, adaptatividade e analogia neurobiológica.

#### 3.1. MODELO DE UM NEURÔNIO

Um neurônio, também conhecido como nó, é uma unidade processadora de informação, fundamental para a operação. Pode-se identificar três elementos básicos em um neurônio (HAYKIN, 1994):

- 1) Um conjunto de conexões, sendo que cada uma delas possui um peso associado,  $w_{kj}$ . E cada entrada  $x_j$  da conexão  $j$  que está ligada ao neurônio  $k$  é multiplicado pelo peso  $w_{kj}$ .

2) Um somador, para somar a multiplicação dos sinais de entrada com seus respectivos pesos de todas as conexões que chegam ao neurônio.

3) Uma função de ativação para limitar a amplitude da saída do neurônio. Geralmente ela limita o intervalo do sinal de saída para  $[0,1]$  ou  $[-1,1]$ . Algumas funções de ativação usadas são a função sigmóide, a função linear e a função passo.

Além desses três elementos, o neurônio possui também um termo *threshold*,  $\theta_k$ , que tem o efeito de diminuir a entrada da rede na função de ativação. Este termo é subtraído da somatória dos pesos, e o resultado é comparado com o zero na função de ativação.

Em termos matemáticos, nós podemos descrever um neurônio  $k$  por meio das seguintes equações:

$$u_k = \sum_{j=1}^n w_{kj} x_j$$

$$y_k = \varphi(u_k - \theta_k)$$

onde  $x_j$  são os sinais de entrada,  $w_j$  são os pesos das conexões,  $y_k$  são as saídas dos neurônios,  $\varphi$  é a função de ativação e  $\theta$  é o *threshold*.

Um modelo de um neurônio pode ser representado da seguinte forma:

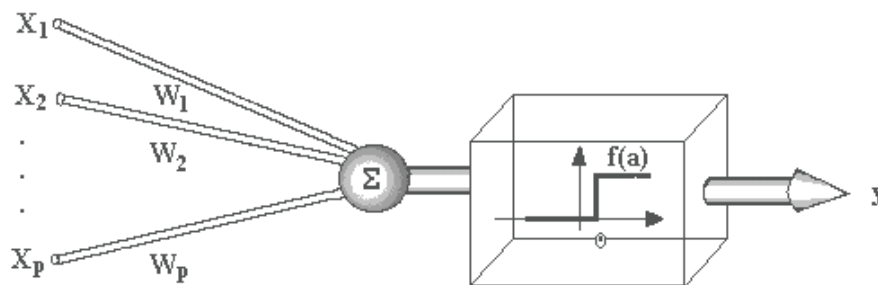


Fig. 1 Modelo de neurônio artificial

Fonte: HAYKIN, 1994

Dispondo-se vários neurônios de uma forma determinada e estabelecendo conexões entre eles, forma-se uma rede neural, que é capaz de aprender.

### 3.2. ARQUITETURAS DE REDES NEURAIS

A arquitetura usada em muitas aplicações de Redes Neurais é a *Multilayer Perceptron*. Neste modelo, os neurônios são dispostos em três camadas; sendo uma de entrada, uma camada conhecida como intermediária ou *hidden*, e uma terceira, chamada camada de saída.

Os nós da camada de entrada da rede formam o padrão que será usado para treinar a rede. Estes neurônios passam pela função de ativação e fornecem a entrada para a camada intermediária. Da mesma forma, as saídas da camada intermediária são usadas como entradas na terceira camada. De uma forma geral, os nós de cada camada têm como suas entradas os sinais de saída da camada anterior. A camada intermediária pode ser considerada como uma extratora de características e a camada de saída apresenta o resultado final.

Um modelo de rede neural é apresentado na Fig. 4.

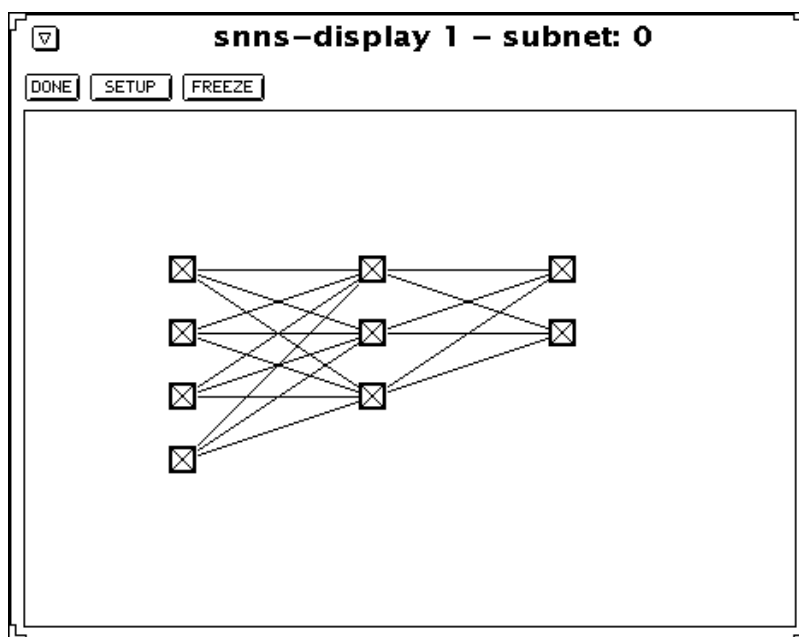


Fig. 2 Rede *multi-layer perceptron*

Fonte: Rede projetada pelo simulador SNNS.

## **4.DESENVOLVIMENTO DE UMA APLICAÇÃO**

A partir de uma base de dados é possível modelar uma rede neural para diversas aplicações. De posse das informações relevantes do problema e com um simulador, como foi feito neste trabalho, ou com um programa feito em uma linguagem de programação qualquer que contenha um algoritmo de treinamento, é possível desenvolver uma rede neural seguindo os passos descritos abaixo (CARVALHO, 1990).

### **4.1.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 teste. 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 cobrir 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.

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.

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.

### **4.2. CONFIGURAÇÃO DA REDE**

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

- 1) Seleção do paradigma neural apropriado à aplicação.

2) Determinação da topologia da rede a ser utilizada - o número de camadas, o número de unidades em cada camada, etc.

3) Determinação de parâmetros do algoritmo de treinamento e funções de ativação. Este passo tem um grande impacto na performance do sistema resultante.

### **4.3. TREINAMENTO**

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

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 (número de vezes que o conjunto de padrões é apresentado à rede). Mas também devem ser considerados a taxa de erro médio por ciclo, e a capacidade de generalização da rede.

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, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

### **4.4. 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.

### **4.5. 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.

## **5.VANTAGENS DAS REDES NEURAIS**

As redes neurais constituem uma técnica poderosa para resolver muitos problemas do mundo real. Elas têm a habilidade de aprender a partir das experiências, melhorando seus desempenhos e adaptando-se às mudanças no ambiente. Além disso, elas são capazes de lidar com informações incompletas ou dados com ruídos e podem ser muito eficientes em situações onde não é possível definir as regras ou passos que levem à solução do problema. As vantagens e características atrativas das redes neurais são listadas abaixo:

- Habilidade de aprender por meio de exemplos e generalizar a partir dos padrões de treinamento, fornecendo saídas apropriadas aos padrões que não foram vistos durante o treinamento.
- Paralelismo natural.
- Representação distribuída das informações, sendo que o conhecimento adquirido pela rede não fica centralizado em um único neurônio.
- Habilidade de lidar com informações incompletas, verificando-se a capacidade da rede em fornecer uma saída satisfatória, mesmo desconhecendo determinados dados de entrada.
- Podem ser usadas em muitos problemas em que não haja regras ou passos que levem à solução.
- Não é necessário informações sobre o ambiente a priori, pois o aprendizado é feito através de exemplos apresentados à rede.
- Pode-se lidar com uma grande quantidade de dados e automaticamente descobrir relações não lineares e complexas entre a entrada e a saída da rede.



## 6. O ALGORITMO BACKPROPAGATION

O método *backpropagation* (BEALE, 1994), utilizado neste trabalho, pode ser usado para várias aplicações de redes neurais. Ele é um método de treinamento do tipo supervisionado, onde o padrão de entrada é apresentado à rede juntamente com a sua saída desejada. O objetivo é treinar a rede e atingir um meio-termo entre a habilidade de responder corretamente aos padrões de entrada que são usados para o treinamento (memorização) e a habilidade de fornecer uma resposta coerente a uma entrada que é similar, mas não idêntica, às utilizadas no treinamento. O método *backpropagation* envolve 3 estágios: a propagação dos padrões de entrada, a retropropagação do erro associado e o ajuste de pesos.

O desenvolvimento do algoritmo de aprendizagem *backpropagation* foi um dos mais importantes avanços da área de redes neurais nos últimos tempos.

Neste algoritmo, usamos a função sigmóide como função de ativação:

$$f(x) = 1 / (1 + e^{-(x - \theta)}) \text{ sendo } \theta \text{ o threshold.}$$

Uma das razões de se usar a função sigmóide como função de ativação é que ela fornece uma informação mais precisa quando o threshold e a soma dos pesos têm valores parecidos. Ou seja, o valor de saída de um neurônio não é simplesmente 1 ou 0, mas varia no intervalo de 0 a 1. Esta informação é importante para o ajuste de pesos.

Os passos do algoritmo são os seguintes (BEALE, 1994):

- 1) Inicializar os pesos e os thresholds com valores randômicos pequenos;
- 2) Apresentar as entradas  $X_p = x_0, x_1, x_2, \dots, x_{n-1}$  e a saída desejada  $T_p = t_0, t_1, t_2, \dots, t_{m-1}$ , onde  $n$  é o número de nós de entrada e  $m$  é o número de nós de saída. Atribua a  $w_0$  o valor  $-\theta$  e a  $x_0$  o valor 1. Para a associação do padrão,  $X_p$  e  $T_p$  representam os padrões a serem associados. Para a classificação,  $T_p$  é definido como 0, exceto para um dos nós, definido como 1 que corresponde a classe que  $X_p$  pertence;
- 3) Calcular a saída atual.

Para cada camada, calcule  $y_p = f[\sum w_i x_i]$ ,

e passe este resultado para a próxima camada. A camada final fornece a saída  $o_{pj}$ ;

4) Adaptar os pesos.

Comece pela camada de saída e trabalhe de trás para frente.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj}, \quad (1)$$

onde  $w_{ij}$  representa o peso do nó  $i$  para o nó  $j$  no instante  $t$ ,  $\eta$  é o termo de ganho e  $\delta_{pj}$  é um termo de erro para o padrão  $p$  no nó  $j$ .

Para neurônios de saída,

$$\delta_{pj} = k o_{pj} (1 - o_{pj})(t_{pj} - o_{pj}), \quad (2)$$

E para neurônios intermediários,

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} w_{jk} \quad (3)$$

onde a soma é sobre os  $k$  nós na camada acima do nó  $j$ ;

5) Repetir os passos 2 a 4 até a rede convergir, ou seja, até que o erro decline a um nível aceitável.

## 7.SIMULAÇÃO

Para se treinar a rede foram utilizados dados sobre o preço em dólares verificado mensalmente da saca de 60 kg de soja no Estado de São Paulo no período de janeiro de 1975 a julho de 1995, totalizando 247 observações. O experimento proposto tem como objetivo prever o preço da saca de soja no mês posterior. Para tanto, a topologia da rede foi definida com 2 neurônios na camada de entrada, correspondendo ao preço nos meses  $t-2$  e  $t-1$  e um neurônio na camada de saída, indicando o preço no mês  $t$ . Deve-se ressaltar que é possível formatar os padrões com mais neurônios na camada de entrada, contendo valores do preço da soja em meses anteriores, alterando o desempenho da rede, bem como fazer a previsão para mais de um mês à frente. Foram feitos vários experimentos variando-se a quantidade de neurônios da camada intermediária entre 3 e 15 neurônios. O critério de parada utilizado foi o número de ciclos. Foram feitos experimentos variando-se este item. Para formatar os padrões apresentados à rede é preciso normalizar o valor dos preços no intervalo entre 0 e 1. Esta normalização é feita com a seguinte equação:

$$X_{NORM} = \frac{X - MÍNIMO}{MÁXIMO - MÍNIMO}$$

Depois de normalizados, os dados devem formar os padrões que treinarão a rede. E estes padrões devem ser colocados no formato aceito pelo simulador, apresentado a seguir:

SNNS pattern definition file V3.2

generated at Mon Feb 1 21:03:40 1999

No. of patterns : 202

No. of input units : 2

No. of output units : 1

# Input pattern 1:

0.315531 0.208011

# Output pattern 1:

0.125791

# Input pattern 2:

0.208011 0.125791

# Output pattern 2:

0.127899

Os padrões foram divididos entre padrões de treinamento e de teste. Os padrões de treinamento correspondem aos dados de janeiro de 1975 a dezembro de 1991, totalizando 202 padrões. E os padrões de teste foram feitos com os dados restantes, de janeiro de 1992 a julho de 1995, com 41 padrões. O desempenho da rede na fase de teste pode ser medido através do erro médio quadrático, que é definido pela fórmula :

$$MSE = \frac{\sum_i (o_i - d_i)^2}{n}$$

Também pode-se usar o arquivo de resultados gerado pelo simulador para verificar o desempenho da rede. Este arquivo fornece os valores da saída atual e desejada, e é possível traçar um gráfico comparativo.

## 8.RESULTADOS OBTIDOS

A seguir são apresentados os resultados de treinamentos contendo o número de neurônios na camada intermediária, o número de ciclos e o valor do MSE, uma tabela com os preços reais e previstos no arquivo de teste referente à rede 8 , além da diferença e a média das diferenças. Por fim, é apresentado um gráfico ilustrando a diferença entre o preço real e a previsão . Os valores da tabela e do gráfico já são apresentado sem a normalização, em dólares.

Rede	Nro neurônios	Nro de ciclos	MSE
1	3	1000	0.00325
2	3	3000	0.00320
3	3	5000	0.00331
4	3	10000	0.00317
5	5	1000	0.00327
6	5	3000	0.00324
7	5	5000	0.0034
8	5	10000	0.00313

9	10	1000	0.00425
10	10	3000	0.00324
11	10	5000	0.00338
12	10	10000	0.00324
13	15	1000	0.00316
14	15	3000	0.00324
15	15	5000	0.00321
16	15	10000	0.00322

Tabela 1: Erro médio quadrático do arquivo de teste.

Tabela comparativa entre preço real da saca de soja e a previsão  
fornecida pela rede neural

mês	PREÇO REAL (US\$)	PREVISÃO DA REDE (US\$)	DIFERENÇA (módulo)
MAR/92	10,18998	10,10517	0,084811
ABR/92	9,410038	10,13306	0,723026
MAI/92	9,719968	9,261619	0,458348
JUN/92	10,28006	9,963727	0,316333
JUL/92	10,22001	10,63055	0,410536
AGO/92	11,02002	10,2913	0,728718
SET/92	12,44999	11,45816	0,991831
OUT/92	11,74006	12,99571	1,255655
NOV/92	11,86002	11,35386	0,506161
DEZ/92	11,79001	11,86258	0,072573
JAN/93	12,37002	11,70889	0,661126
FEV/93	11,06997	12,51559	1,445626

MAR/93	10,07003	10,51101	0,440988
ABR/93	9,649956	9,763369	0,113413
MAI/93	9,79994	9,605985	0,193955
JUN/93	9,859991	9,975538	0,115548
JUL/93	12,53993	9,996314	2,543613
AGO/93	12,81001	13,64503	0,835016
SET/93	11,76994	12,71894	0,948999
OUT/93	11,89004	11,22194	0,668098
NOV/93	12,70998	11,88805	0,821925
DEZ/93	12,92997	12,90635	0,023622
JAN/94	12,77003	12,79094	0,020918
FEV/94	12,15002	12,47646	0,326436
MAR/94	11,52006	11,74134	0,221277
ABR/94	10,58003	11,20473	0,624697
MAI/94	10,91002	10,24064	0,669379
JUN/94	11,63006	11,12475	0,505307
JUL/94	11,49004	11,96518	0,47514
AGO/94	11,52006	11,4159	0,104164
SET/94	11,9	11,52561	0,374391
OUT/94	12,38994	12,02608	0,363861
NOV/94	12,62004	12,30029	0,319748
DEZ/94	12,49994	12,61591	0,115975
JAN/95	12,48002	12,27596	0,204058
FEV/95	12,44003	12,30897	0,131058
MAR/95	9,649956	12,26628	2,616328
ABR/95	9,139953	8,84838	0,291573

MAI/95	9,710007	9,116331	0,593676
JUN/95	9,529997	10,06348	0,533483
JUL/95	9,200003	9,585067	0,385064

Média das diferenças
US\$ 0,566743

Tabela 2 : Comparação entre preço real e a previsão da rede neural

Fonte: Arquivo de resultados do SNNS

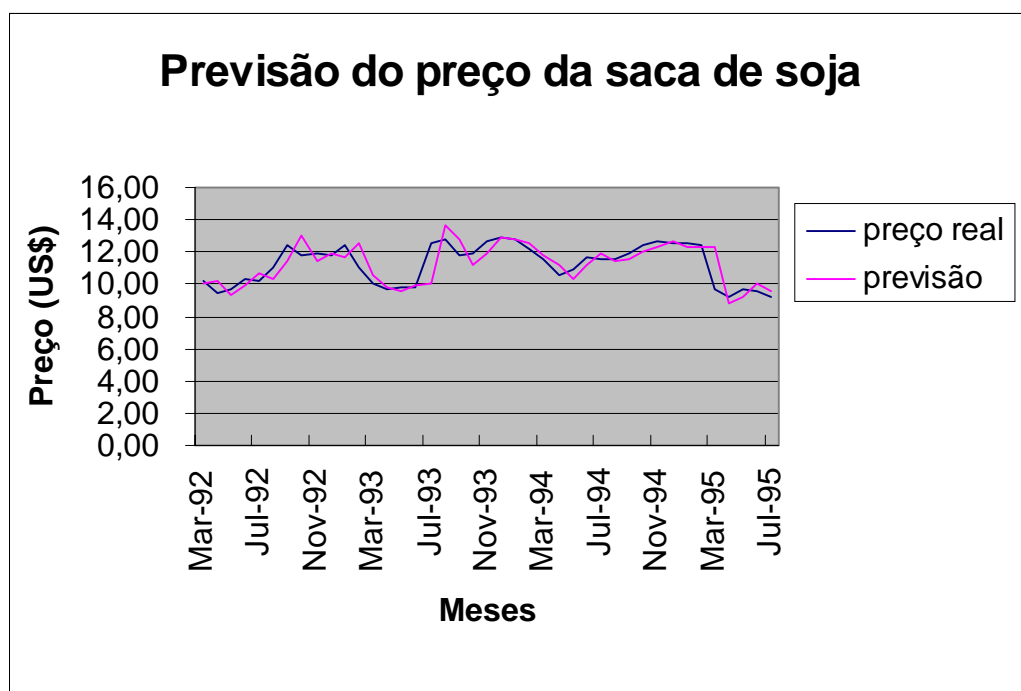


Fig 3: Gráfico comparativo entre as saídas desejada e atual na fase de teste da rede treinada

Fonte: Arquivo de resultados do simulador SNNS

## 9.CONCLUSÃO

O trabalho realizado mostrou que as redes neurais são uma alternativa aos métodos estatísticos para o problema de previsão de séries temporais, apresentando bons resultados, como mostra a tabela com os preços reais e previstos. Pela tabela 2, nota –se que a diferença entre o preço real e o previsto é pequena e o gráfico mostra que a curva da previsão da rede acompanha a tendência do preço real. Foi mostrado que, a partir de um banco de dados, as redes neurais podem ser utilizadas para várias aplicações e elas podem comportar muitas variáveis e descobrir relações entre elas. Em relação à aplicação apresentada, os resultados obtidos na fase de teste foram satisfatórios, sendo que um sistema feito a partir dessa rede treinada para definir o preço da soja no próximo mês pode ser útil na tomada de decisão por parte dos administradores.

## 10. REFERÊNCIAS BIBLIOGRÁFICAS

AGRIANUAL 96.

BEALE, R. & JACKSON, T. Neural Computing: an introduction . IOP Publishing Ltda, 1994.

BOX, G. E. P. , JENKINS, G. M., REINSEL, G. C. , Time series analysis: forecast and control, Prentice Hall, 1994.

CARVALHO, A. C. P. L. F. Um novo modelo de neurônio para o reconhecimento de sequências . Tese de Mestrado, Universidade Federal de Pernambuco, Recife, 1990.

DÍBIO, L. B., Martins, W., IV Simpósio Brasileiro de Redes Neurais. Universidade Federal de Goiás , 1997.

FAUSETT, L. Fundamentals of neural networks. Prentice-Hall, Inc, 1994.

FUKUSHIMA, K. , Neocognitron: a hierarchical neural network capable of visual pattern recognition , Neural Networks I, 1988.

HAYKIN, S. Neural Networks. A Comprehensive Foundation. Macmillan College Publishing Company, 1993.

HEBB, D. O., The Organization of Behavior, John Wiley, 1949.

KOHONEN, T., Self-organized formation of topological correct feature maps, Biological Cybernetics 43, 1982.

McCULLOCH, W., PITTS, W., A logical calculus of the ideas immanent in neurons activity, Bulletin of Mathematical Biophysics 5, 1943.

ROSENBLATT, F. , Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism, Spartan, 1962.

ZELL, A. *et al.* SNNS - Stuttgart Neural Network Simulator, User Manual, versão 4.1. IPVR, Alemanha, 1995.



# ANEXO I

## Preço da saca de 60 kg de soja em dólares

ANO	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
1975	11,83	10,30	9,13	9,16	9,31	9,19	9,33	9,98	10,17	9,99	9,48	9,21
1976	9,12	8,72	8,26	7,79	7,96	9,30	10,43	10,46	11,64	12,42	12,66	12,14
1977	12,26	12,28	13,95	15,13	14,82	13,57	10,86	9,80	9,33	9,61	10,10	10,36
1978	10,80	10,85	11,39	12,08	11,92	11,63	11,33	11,02	11,34	12,12	12,87	12,92
1979	12,59	12,71	13,02	12,76	12,16	12,24	13,71	14,64	14,65	15,18	14,85	11,82
1980	10,85	11,18	10,85	10,34	10,08	9,95	10,21	10,96	11,34	12,36	13,04	13,36
1981	12,77	12,61	11,87	11,53	11,41	11,19	10,95	10,65	11,08	11,65	11,48	10,79
1982	10,87	11,73	11,55	11,55	12,08	12,24	11,53	10,92	9,85	9,63	9,45	10,97
1983	11,18	9,88	8,47	9,43	9,35	9,88	9,88	11,28	21,57	19,65	17,98	16,96
1984	16,71	14,77	14,19	14,07	13,96	13,57	11,57	10,70	11,76	12,53	13,09	12,20
1985	11,29	11,01	10,18	9,82	8,98	8,55	9,39	9,60	9,10	9,68	10,86	10,38
1986	9,90	9,94	9,33	9,34	9,24	9,24	9,16	9,29	9,32	9,35	9,45	9,57
1987	9,00	7,74	7,78	7,34	7,84	8,37	8,62	10,26	11,98	11,97	12,02	14,12
1988	13,59	12,59	10,92	11,59	12,13	16,13	15,56	14,67	15,15	15,11	15,89	16,32
1989	15,08	14,15	13,95	13,89	13,31	10,91	10,49	8,42	9,72	9,91	10,25	11,06
1990	12,63	11,04	10,86	9,77	10,55	9,85	9,94	10,10	9,86	9,90	10,16	9,32
1991	9,19	10,02	10,03	9,95	9,85	9,62	9,12	10,10	11,10	11,98	10,05	9,56
1992	11,75	10,55	10,19	9,41	9,72	10,28	10,22	11,02	12,45	11,74	11,86	11,79
1993	12,37	11,07	10,07	9,65	9,80	9,86	12,54	12,81	11,77	11,89	12,71	12,93
1994	12,77	12,15	11,52	10,58	10,91	11,63	11,49	11,52	11,90	12,39	12,62	12,50
1995	12,48	12,44	9,65	9,14	9,71	9,53	9,2					

Fonte: Agrianual 96.