Previsão do Mercado de Ações Brasileiro utilizando Redes Neurais Artificiais

Elisângela Lopes de Faria [(a)]
Marcelo Portes Albuquerque [(a)]
Jorge Luis González Alfonso [(b)]
Márcio Portes Albuquerque [(a)]
José Thadeu Pinto Cavalcante [(a)]

[(a)] Centro Brasileiro de Pesquisas Físicas – CBPF
Rua Dr. Xavier Sigaud, 150
22290-180 – Rio de Janeiro, RJ

[(b)] Pontifícia Universidade Católica – PUC-RIO
Marquês de São Vicente, 225
22453-900 – Rio de Janeiro, RJ

Resumo

Esta nota técnica tem como objetivo o estudo das redes neurais artificiais na previsão da série temporal do indicador mais importante do mercado de ações brasileiro, mais precisamente a série Ibovespa, que é o índice da Bolsa de Valores do estado de São Paulo (BOVESPA). Para este estudo será implementada uma rede neural artificial *multilayer perceptrons* com algoritmo *backpropagation* para fazer previsões do índice Ibovespa.

Sumário

1. Introdução

Nos dias atuais o desenvolvimento da computação e outras técnicas avançadas têm proporcionado uma melhor compreensão de sistemas denominados *sistemas complexos e não lineares*. Estes sistemas se caracterizam por apresentar um comportamento não linear onde vários fatores, de diferente natureza, influenciam a evolução no tempo (ou série temporal) do mesmo. Recentemente o surgimento de novas técnicas computacionais tem proporcionado um acompanhamento on-line da estrutura destes sistemas, fornecendo dados experimentais suficientes que tem levado a um melhor entendimento de seu comportamento. Neste sentido as Redes Neurais Artificiais (RNAs) têm encontrado aplicação no estudo e previsão de séries temporais complexas, como por exemplo, clima, transmissão de energia elétrica, séries financeiras, etc. Recentemente, as RNAs começaram a serem usadas na previsão de séries financeiras, como por exemplo, previsão do dólar, balança comercial, mercado de ações e outros. Dentre estes estudos podemos citar em particular o uso de RNAs para prever o mercado de ações e derivativos. Estes mercados têm se desenvolvido muito ultimamente atraindo a atenção de muitos pesquisadores e aplicações bem sucedidas das RNAs e previsões de séries temporais financeiras podem ser encontrados em [CHAK] e também [TANG] entre outros.

Com relação ao mercado de ações, métodos estatísticos foram uma alternativa inicial para se entender e prever os mesmos, porém atualmente as RNAs fazem um grande diferencial no estudo dos mesmos. O que torna as Redes Neurais tão atraentes no estudo de séries financeiras (assim como outros sistemas não lineares) é o fato delas acompanharem mudanças contínuas e bruscas destes sistemas que não podem ser descritas por um modelo matemático pré-definido. Estudos têm mostrado que as RNAs são capazes de acompanhar e seguir estas mudanças extremamente não lineares e imprevisíveis mostrando resultados surpreendentes [DAVI]. Tudo isto tem provocado nos últimos anos o aumento no número de teses e estudos que tentam usar RNAs para prever séries temporais de mercados financeiros.

2. Mercado de Ações

O mercado de ações brasileiro comumente chamado de BOVESPA reúne as ações das principais empresas brasileiras de capital aberto (Vale, Petrobras, etc) e está situado na cidade de São Paulo. Podemos definir ação como a menor parcela em que se divide uma empresa, sendo que todos os dias milhões destes papéis são negociados entre investidores por preços definidos pela lei da oferta e da procura. Os negócios com ações são acompanhados instantaneamente pela internet ou outros meios. O principal indicador do mercado de ações brasileiro é chamado de IBOVESPA e o mesmo mede a lucratividade de uma carteira teórica de ações (as mais negociadas) além de retratar o comportamento do mercado em geral [BOVE]. A previsão deste indicador é o objetivo central deste trabalho. A escolha deste indicador é determinada pelo fato de que o mesmo representa o *estado* do mercado financeiro brasileiro como um todo. A figura 2.1 mostra um gráfico com as cotações diárias do Ibovespa no período de setembro de 1998 a agosto de 2007.

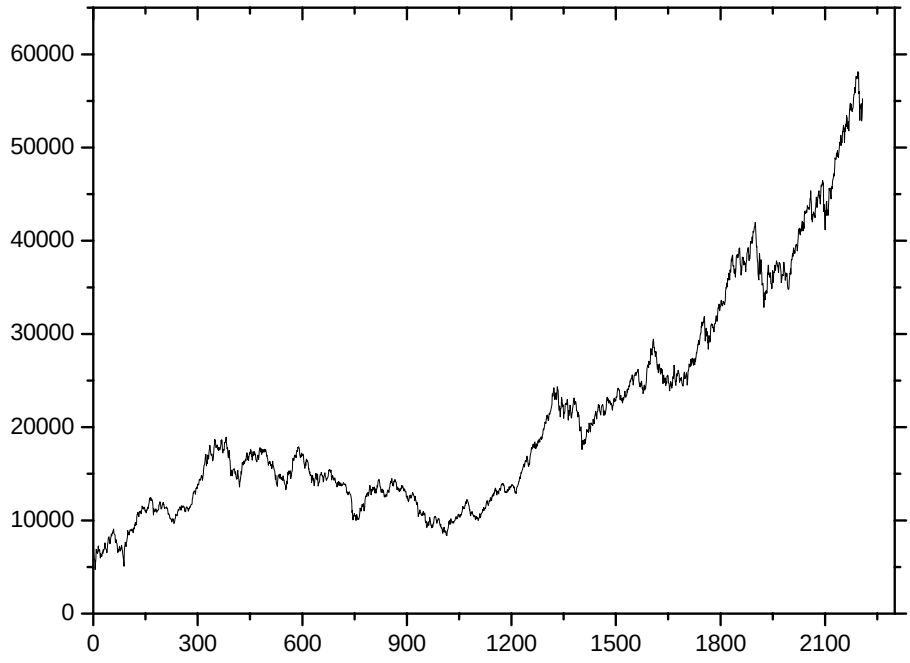

Figura 2.1 Cotações do Ibovespa (Set/98 a Ago/07)

3. Redes Neurais Artificiais

As Redes Neurais Artificiais são formadas por diversos elementos, que tentam imitar o funcionamento dos neurônios. Assim como o cérebro humano as RNAs adquirem o conhecimento através do ambiente externo e tenta encontrar a solução de um determinado problema através de um processo de aprendizado. Atualmente as Redes Neurais Artificiais estão sendo utilizadas em diversas aplicações dentre as quais podemos citar reconhecimento de caracteres manuscritos, diagnósticos médicos, sistemas de segurança, previsões de séries temporais, etc. Características das RNAs como convergência, capacidade de generalização, tolerância a falhas, e uniformidade, entre outras, fazem delas uma poderosa ferramenta computacional.

3.1. Neurônio Artificial

Um neurônio artificial é formado por três elementos básicos: os pesos sinápticos, a função de soma, e a função de transferência também conhecida como função de ativação. A figura 3.1 mostra um exemplo de neurônio com três entradas (cada uma com um peso respectivo), uma função somadora que produz um valor que será processado pela função de transferência gerando assim a saída (y) do neurônio.

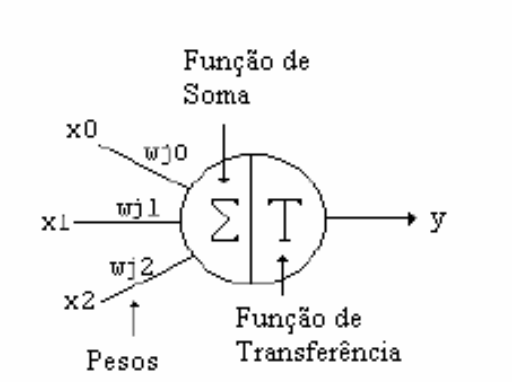

Figura 3.1 Neurônio Artificial

Dentre as principais funções de ativação podemos destacar as funções: linear, degrau e tangente hiperbólica. Outros tipos de funções podem ser usados dependendo do problema específico.

3.2. Topologia

As RNAs podem ser do tipo *Single–Layer Feddforward*, também conhecida como *perceptrons* e que possui apenas uma camada de nós de entrada ligada diretamente à camada de neurônios de saída, ou *Multilayer feedforward*, que assim como a arquitetura anterior, possui camadas de entrada e saída e também uma ou mais camadas intermediárias. Esta última arquitetura é uma das mais utilizadas nas diferentes aplicações, pois o fato de possuir camadas intermediárias aumenta o seu poder computacional [HAYK]. Na figura 3.2 pode ser visualizada a topologia desta arquitetura. Finalmente podemos destacar as Redes Recorrentes que possuem pelo menos um ciclo de feedback, resultando em um comportamento de dinâmica não linear.

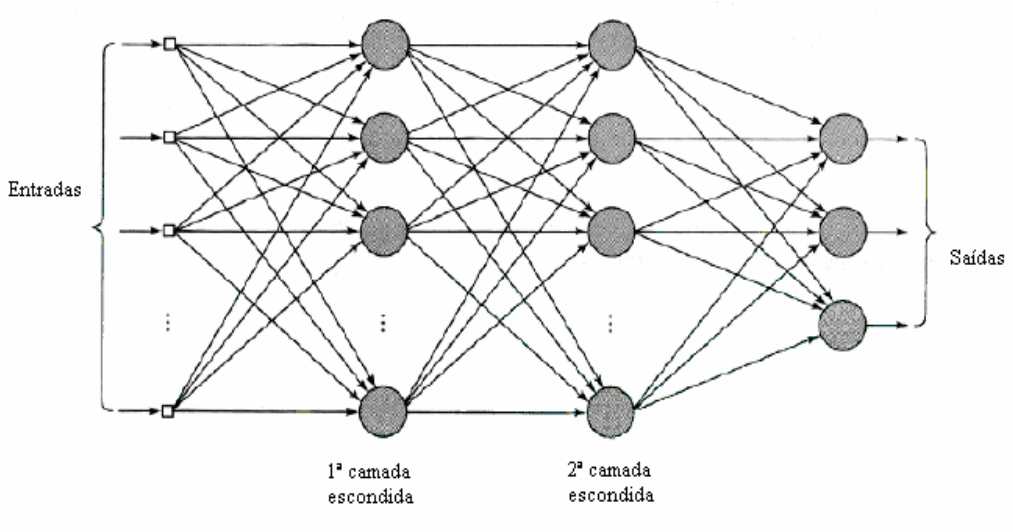

Figura3.2 Arquitetura de uma MLP (Multilayer Perceptrons)

3.3. Treinamento

O treinamento das redes neurais é um processo muito importante e o mesmo pode ser supervisionado ou não supervisionado. Para treinamento supervisionado existe a presença de um professor que indica para a rede um comportamento a seguir, ou seja, a rede recebe um conjunto de exemplo de pares entrada e saída desejada. No treinamento não supervisionado não existe a presença de um professor, a rede aprende através de exemplos de objetos semelhantes para criar grupos ou classes.

3.3.1. Treinamento com Algoritmo Backpropagation

O algoritmo *backpropagation* é do tipo supervisionado, ou seja, utiliza par (entrada e saída desejadas) e é realizado em duas etapas. Na primeira a entrada é propagada pela rede, no sentido da entrada para a saída (fase *forward*), sem que ocorram alterações dos pesos. Já na segunda a resposta de rede é comparada com a resposta desejada e os pesos são reajustados de maneira a minimizar o erro (fase *backward*). Nesta segunda etapa o ajuste de um peso w_{ij} que define seu valor para a próxima iteração é dado por:

$$w_{ij}(n+1) = w_{ij}(n) + \eta\delta_{j}(n)x_{j}(n)$$

Na equação anterior η é a taxa de aprendizagem e δ_j é o gradiente local do erro para o neurônio j. A taxa de aprendizagem η é um parâmetro que geralmente é menor que 1. É preciso ter cuidado com o valor desta taxa, pois quanto menor for o seu valor menor serão as variações dos pesos sinápticos da rede de uma iteração para outra, o que poderá causar uma lentidão na convergência do treinamento. Por outro lado se esta taxa assumir um valor muito alto causando grandes modificações nos pesos sinápticos a rede pode se tornar instável. Já o gradiente local do erro δ_j é determinado através do método do gradiente descendente. Detalhes do algoritmo podem ser encontrados em [HAYK], uma vez que este estudo não faz parte do escopo deste trabalho.

4. Desenvolvimento

Este trabalho visa criar um projeto de rede neural artificial para fazer previsões do índice Ibovespa (índice da Bolsa de Valores de São Paulo). O software utilizado para o projeto foi o Matlab versão 7.1, que é uma ferramenta computacional muito eficiente e também apresenta um toolbox totalmente voltado para redes neurais. Detalhes desta ferramenta podem ser encontrados em [MATL]. O trabalho envolvendo redes neurais é feito de muitas escolhas empíricas, pois não existem na literatura os parâmetros ideais da rede que possam levar a encontrar aquela que apresenta o resultado mais satisfatório. Nosso trabalho foi dividido em duas grandes etapas, a primeira está relacionada à preparação dos dados, nos quais temos a seleção das variáveis (neste trabalho em específico, foi escolhido as cotações do fechamento do Ibovespa) e a separação deste conjunto em outros dois chamados de treinamento (aproximadamente 85 a 90 % dos dados) e teste (aproximadamente 10 a 15%). A segunda etapa está relacionada ao projeto da rede na qual temos a definição do modelo, da arquitetura e dos parâmetros necessários da rede.

4.1. Preparação dos dados (Treinamento e Teste)

Os dados utilizados durante este trabalho são reais e referentes aos valores das cotações diárias do fechamento da série Ibovespa no período de Setembro de 1998 a agosto de 2007. Na base de dados, o fechamento corresponde ao último valor assumido pelo Ibovespa em um determinado dia. O conjunto de dados (mostrado nas figuras 4.1 e 4.2) foi dividido em duas partes, uma correspondente ao treinamento da rede (de 01/09/98 a 28/12/06) totalizando 2058 pontos, e a segunda parte chamado de conjunto teste (de janeiro/07 a agosto/07) totalizando 150 pontos. Antes de serem apresentados à rede os dados passaram por um pré-processamento sendo normalizados no intervalo [-1,1], e posteriormente sendo desnormalizados. Para a normalização e desnormalização dos dados foram utilizadas duas rotinas disponíveis no software matlab chamadas

respectivamente premnmx e postmnmx. A utilização destas rotinas no matlab é feita da seguinte forma:

Normalização: [pn, minp, maxp] = premnmx(p). A função premnmx recebe o conjunto utilizado para o treinamento (p) como parâmetro e retorna o conjunto treinamento normalizado (pn) e o valor mínimo (minp) e máximo (maxp) do conjunto de treinamento.

Desnormalização: [p] = postmnmx (pn, minp, maxp). A função postmnmx faz a condição inversa à função anterior. Ela recebe o valor normalizado (pn) e o valor mínimo e máximo do conjunto (p) e retorna o valor desnormalizado.

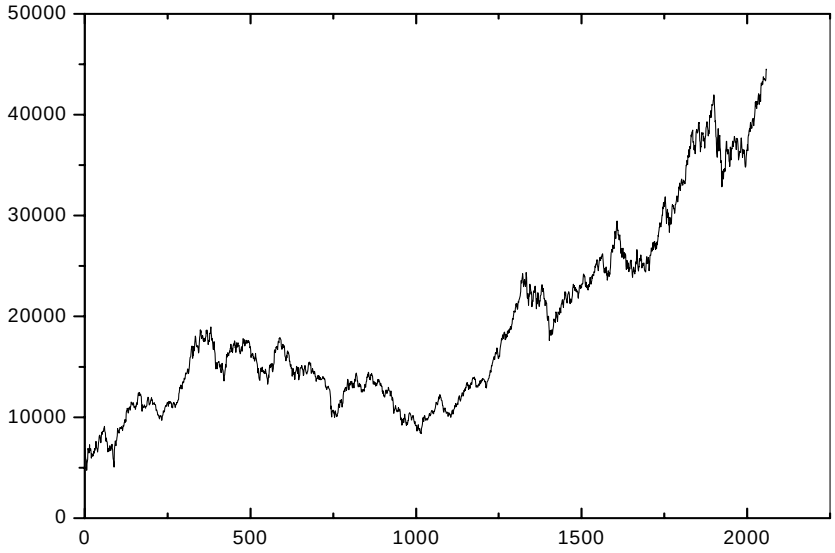

Figura 4.1 Conjunto treinamento (Set/98 a Dez/06)

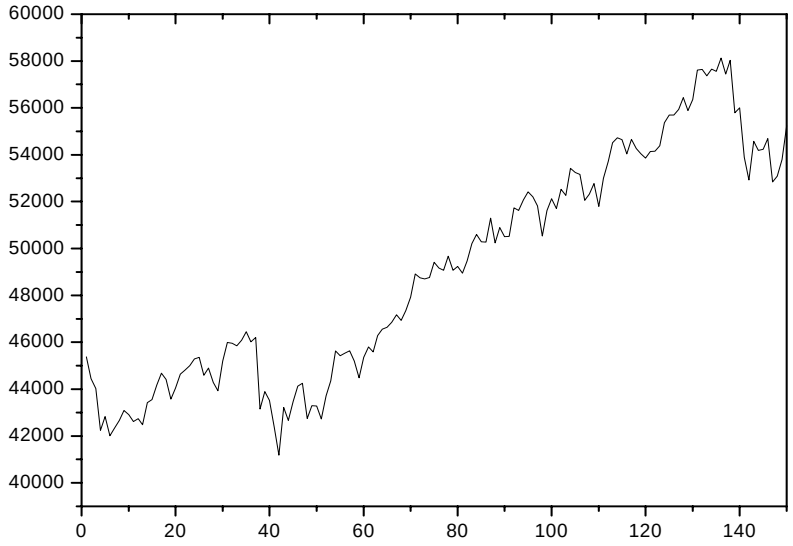

Figura 4.2 Conjunto Teste (Jan/07 a Ago/07)

4.2. Projeto da rede

O modelo escolhido para fazer as previsões do Ibovespa é um modelo de rede neural *feedforward* com algoritmo *backpropagation*. Os parâmetros da rede foram alterados ao longo do treinamento com a finalidade de se encontrar um modelo com um resultado mais satisfatório. Para criar e simular todos estes parâmetros foi criado um programa no matlab que permitiu alterar diferentes parâmetros da simulação. Devido ao grande número de redes criadas e ao tempo gasto pelas redes na convergência das mesmas o programa foi executado no *Cluster Linux SSolar* do CBPF. No programa proposto os parâmetros alterados foram os seguintes:

- ✓ Quantidade de camadas escondidas: foram adotadas arquiteturas com até duas camadas escondidas.
- ✓ Número de neurônios da primeira camada escondida: variou entre 3 e 30. Mais precisamente foram criadas redes com 3, 5, 10, 15, 20 e 30 neurônios.
- ✓ Número de neurônios da segunda camada escondida: 5, 12 e 20 neurônios.

- ✓ Tamanho da janela: número de cotações utilizado para treinamento. Exemplo: tamanho da janela igual a cinco, seriam utilizados cinco dias de cotações do Ibovespa, ou seja, dias 1, 2, 3, 4, 5 para a previsão do sexto dia. Esta técnica será detalhada na próxima seção. Os tamanhos de janela escolhidos para este trabalho foram de 5, 10, 15, 20, 25, 30, 40, 50 e 60.

Os demais parâmetros da rede como função de ativação e método de treinamento não foram modificados ao longo do treinamento e receberam as mesmas funções e métodos para todas as configurações de rede possíveis, ou seja, para a função de ativação foi utilizada a função tangente hiperbólica (no matlab chamada de tansig), para o método de treinamento foi utilizado o algoritmo resilient backpropagation (no matlab chamado de trainrp). Sendo assim os modelos de rede foram estruturados em uma camada de entrada, onde o número de neurônios foi definido pelo tamanho da janela, uma ou duas camadas escondidas com quantidades variadas ao longo do treinamento, e com apenas um neurônio na camada de saída, que representa a previsão do Ibovespa no dia seguinte. Uma das possíveis arquiteturas propostas pode ser visualizada na figura 4.3 abaixo onde temos uma rede que trabalha com uma janela de entrada de 5 dias, uma camada escondida e uma saída correspondente ao Ibovespa do sexto dia.

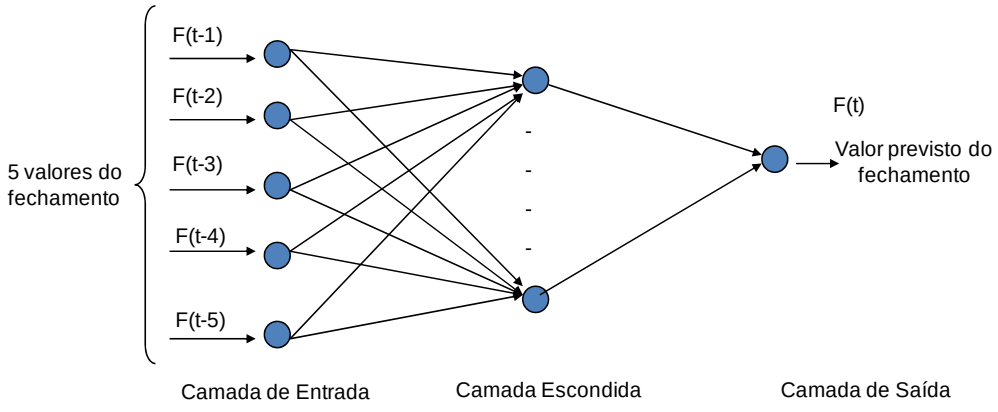

Figura 4.3 Arquitetura proposta.

4.2.1. Técnica de Janelamento

Conforme apontado anteriormente foram definidas duas janelas, uma de entrada (J^I) e outra de saída (J^0) de tamanhos fixos E (entrada) e S (saída) respectivamente, e deslocá-las no tempo com um passo fixo de tamanho *p*. Sendo assim para um determinado tamanho de janela, a suposição é que a seqüência de valores $J^I_0,\cdots,J^I_n$ está de alguma forma relacionada com a seqüência $J^O_0,\cdots\cdots,J^O_n$ e esta relação, embora desconhecida, está definida completamente no conjunto de dados [REFE]. No nosso caso a utilização deste método foi feita utilizando os pares de janela J^I e J^O como vetores de treinamento, onde os dados das janelas J^I's e J^O's formarão respectivamente os vetores de entrada formados pelas cotações do Ibovespa (fechamento) com tamanho fixo *E* (entrada) conforme especificado acima e os vetores alvos correspondentes que formam os dados de saída com tamanho fixo *S=1* (saída), ou seja, o fechamento do Ibovespa um dia à frente. Já o passo de tamanho *p* foi fixado em 1. Na figura abaixo 4.4 pode ser visualizada a criação de uma janela de previsão com entrada E = 5 (5 dias de cotações do Ibovespa) e saída S = 1 (dia de cotação do Ibovespa do 6 º dia).

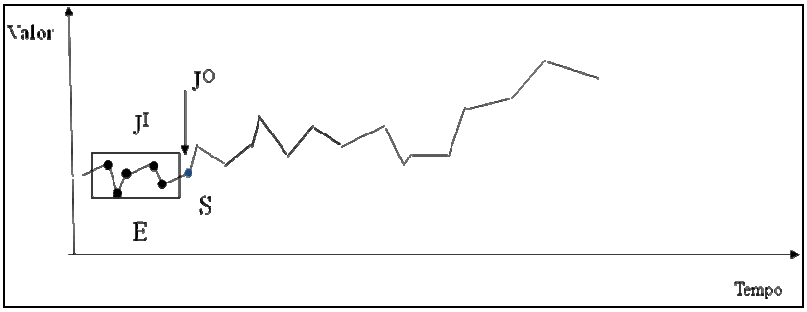

Figura 4.4 Criação da Janela de Previsão

4.2.2. Treinamento e Simulação

Foram criadas 240 redes com as diferentes configurações possíveis. O algoritmo criado em matlab primeiramente carrega o conjunto de treinamento, ou seja, o conjunto contendo todas as 2058 cotações diárias do fechamento do Ibovespa. Em seguida é feita a normalização dos dados conforme citado na seção 4.1. No próximo passo é feito a criação da matriz de treinamento (entradas da rede) com um determinado número de cotações do fechamento do Ibovespa (janela de entrada). Como o algoritmo utilizado backpropagation é supervisionado foi criada a matriz resposta para a rede. Com as duas matrizes (entrada e resposta desejada) já construídas, elas foram então apresentadas à rede formando um par entrada-saída de treinamento. Toda vez que um conjunto todo é apresentado à rede para treinamento temos uma época (epoch). Como o aprendizado no treinamento ocorre através da apresentação repetitiva do conjunto de exemplos, o número de épocas foi estipulado em 500000. Este número alto de épocas foi escolhido propositalmente para obrigar a rede a convergir quando atingisse o erro mínimo desejável e não o número máximo de épocas. Valores do erro do treinamento usado na simulação foram da ordem de 10^{-3}. Cada possível configuração utilizada foi apresentada para treinamento um total de 50 vezes, pois como a rede sempre inicia seu treinamento com pesos aleatórios, é possível de se obter um bom resultado com uma determinada configuração que não necessariamente fornece a melhor solução.

Com a rede treinada foi feita a previsão utilizando os valores de entrada que a rede nunca viu antes, ou seja, o conjunto de teste. O primeiro passo para a simulação foi carregar o conjunto de teste e normalizar os dados no intervalo [-1,1]. Foi criado então um padrão de entrada para a rede com a mesma quantidade de cotações (fechamento Ibovespa) utilizadas na matriz de treinamento. Com o primeiro padrão construído foi feita a simulação. A resposta obtida da simulação que corresponde à primeira previsão é armazenada e um segundo padrão é então construído e apresentado para a simulação, e assim sucessivamente até que as 150 previsões (tamanho do conjunto teste) foram concluídas. A figura 4.5 mostra a estrutura do algoritmo proposto.

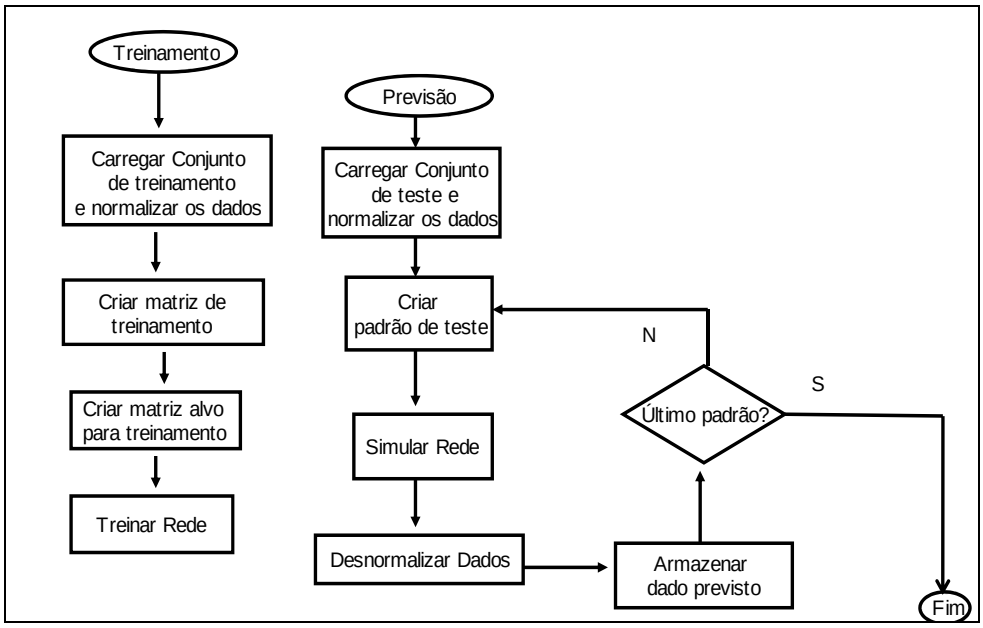

Figura 4.5 Estrutura do algoritmo

4.3. Resultados obtidos

A tabela 4.1 visualizada abaixo apresenta as redes com melhores desempenhos, juntamente com sua arquitetura, o parâmetro tamanho da janela, e os respectivos *Root Mean Square Erro* (RMSE) obtidos.

	Arquitetura	Tamanho da Janela	RMSE
Rede1	10x3x1	JE = 10	0,0188
Rede2	15x3x1	JE = 15	0,0191
Rede3	5x5x1	JE = 5	0,0165

Rede4	5x10x1	JE = 5	0,0181
Rede5	15x10x1	JE = 15	0,0192
Rede6	5x15x1	JE = 5	0,0172
Rede7	5x20x1	JE = 5	0,0164
Rede8	5x30x1	JE = 5	0,0172
Rede9	5x3x5x1	JE = 5	0,0177
Rede10	5x5x5x1	JE = 5	0,0170
Rede11	5x20x5x1	JE = 5	0,0168
Rede12	5x30x5x1	JE = 5	0,0176
Rede13	5x10x12x1	JE = 5	0,0173
Rede14	5x15x5x1	JE = 5	0,0168
Rede15	5x3x20x1	JE = 5	0,0180

Tabela 4.1 Tabela com melhores resultados: parâmetros e respectivos erros.

Na tabela acima: Arquitetura 10x3x1 significa 10 neurônios na camada de entrada, 3 neurônios na camada escondida, e apenas 1 neurônio na camada de saída. JE: número de dias na janela de entrada, os tamanhos da janela de saída e o passo a cada iteração não constam na tabela acima, pois eles receberam o valor 1 para todas as arquiteturas e RMSE (*Root Mean Square Erro*) métrica utilizada para verificar a eficiência das previsões.

5. Resultados e Discussão

Dentre as diversas técnicas utilizadas para avaliar os resultados obtidos com RNAs podemos citar: comparação gráfica, diagramas de dispersão, raiz do erro médio quadrático, erro percentual médio absoluto, coeficiente de U-Theil, entre outros. Neste trabalho, para avaliar os resultados obtidos foram utilizados algumas das técnicas citadas acima como comparação gráfica e a raiz do erro médio quadrático (RMSE). A equação do cálculo do RMSE pode ser visualizada abaixo:

$$RMSE = \sqrt{\frac{\sum_{K=1}^{N}\left(\frac{a_k - y_k}{a_k}\right)^2}{N}}$$

Onde N é o número de padrões, neste trabalho N é igual a 150, a_k representa o valor real no instante t, e y_k representa o valor previsto no instante t.

5.1 Métrica RMSE

O histograma dos erros (RMSE) em todas as simulações com todas as redes criadas neste trabalho está apresentado na figura abaixo, onde o valor médio da distribuição é de 2%.

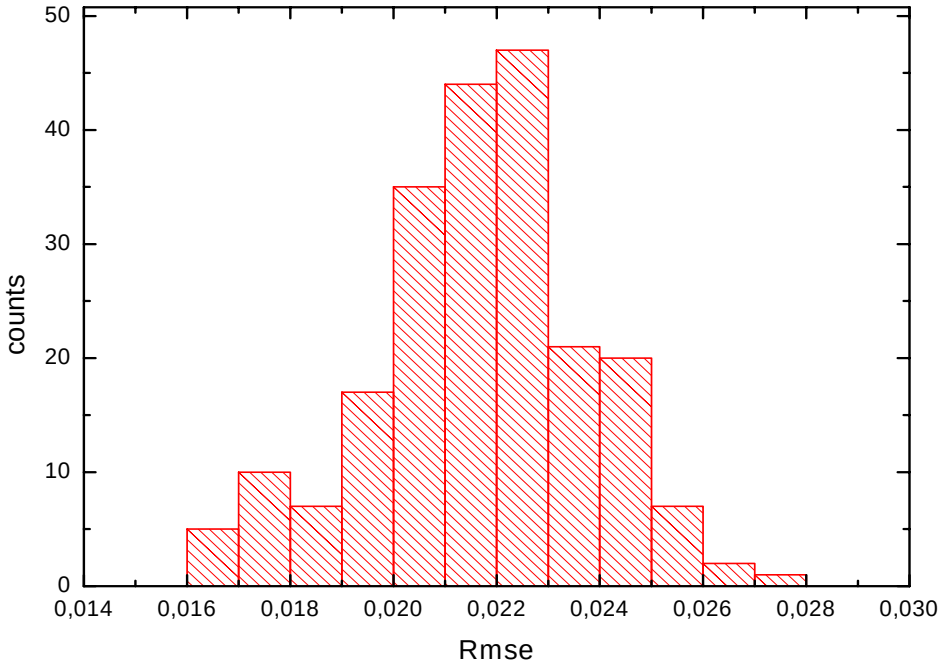

Figura 5.1 Histograma de todos os erros das possíveis configurações

5.2 Métrica Comparação Gráfica

Através da comparação gráfica é possível ter uma melhor visualização dos resultados. É uma métrica bem simples que não quantifica os resultados, mas permite fazer uma boa análise visual. Na figura 4.7 pode ser visualizado o gráfico com os valores alvos e as respectivas previsões feitas pelas redes neurais que apresentou o menor *Root Mean Square Erro* (RNA7).

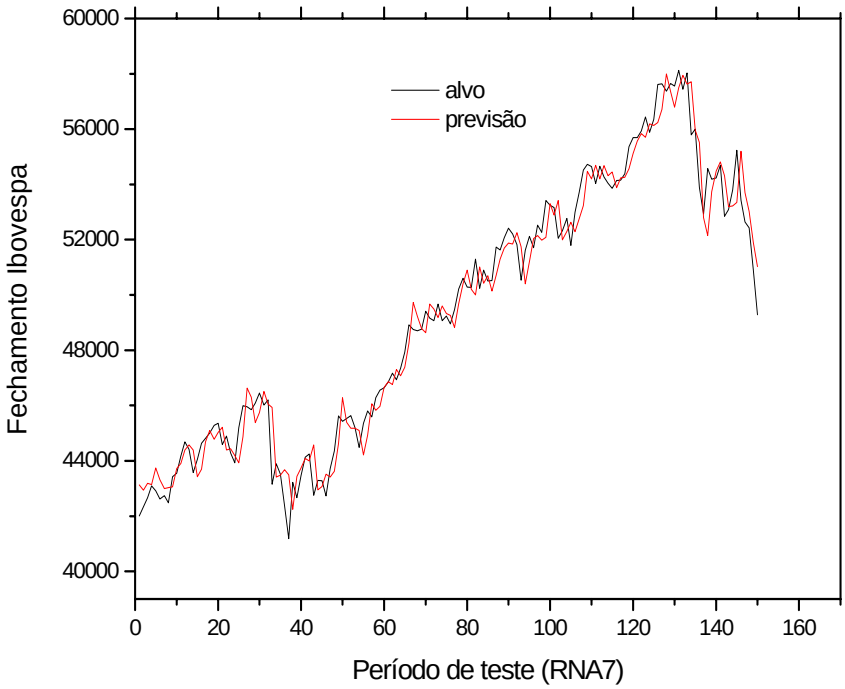

Figura 4.7 Gráfico índice Ibovespa e sua previsão.

5.3 Análise dos Parâmetros da rede e comparação da métrica

Analisando a tabela 4.1 é possível perceber alguns pontos relevantes que serão destacados a seguir.

- Os menores erros (RMSE) encontrados durante as simulações possuem tamanho de janela igual a 5 ou valores próximos do mesmo. Isto nos parece lógico uma vez com o tamanho de janela igual a cinco é possível de se construir um conjunto de treinamento mais representativo, o que não acontece quando usamos janela de tamanho igual a 60, por exemplo.

- Quanto ao número de camadas escondidas para se obter bons resultados basta uma única camada escondida. A utilização de duas camadas escondidas não acrescentou nenhuma melhora nos resultados. Este é um ponto importante uma

vez que pode otimizar o tempo gasto no treinamento com a utilização de apenas uma camada escondida.

- Quanto ao erro no treinamento todas as redes criadas convergiram ao atingir o erro mínimo desejável no treinamento (ordem de 10^{-3}).

- Quanto ao número máximo de épocas utilizadas para atingir o erro desejável no treinamento (ordem de 10^{-3}) ficou na faixa de 5000 a 10000 épocas. Sendo que como já era esperado, tamanhos de janelas maiores implica em um número maior de épocas para a convergência.

- Quanto à quantidade de vezes que uma determinada configuração foi apresentada ao treinamento (total de 50 vezes) foi possível constatar que não houve muita diferença entre um treinamento e outro. O que nos leva a concluir que os diferentes pesos sinápticos utilizados na inicialização da rede não alteram significativamente os resultados.

- Quanto às métricas utilizadas é possível destacar que com os parâmetros utilizados neste trabalho os erros (RMSE) encontrados ficam na faixa de 1.5 a 2%. E que analisando graficamente os resultados pode-se perceber que os valores de previsão encontrados conseguem acompanhar os valores reais da série.

6. *Conclusões Gerais*

Neste trabalho foi estudado o mercado de ações brasileiro através do uso de redes neurais artificiais. Foram criadas e simuladas diferentes configurações de RNAs com o objetivo de tentar prever o indicador mais importante da Bolsa de Valores de São Paulo, o Ibovespa. Os resultados obtidos permitiram concluir que:

- O indicador Ibovespa pode ser previsto com erros médios em torno de 2 %.
- A técnica de janelamento permitiu otimizar os resultados mostrando-se uma ferramental útil para a previsão. O tamanho de janela igual a cinco produziu os melhores resultados.
- Apenas uma camada escondida foi suficiente para a previsão do indicador estudado.
- O erro no treinamento foi igual a 10^{-3}. Este valor foi suficiente para obter previsões razoáveis do indicador estudado.

Por fim destacamos que estudos envolvendo redes neurais artificiais envolvem muito tempo na busca de parâmetros para a rede que melhor representa os resultados almejados. Nós conferimos que estes parâmetros são obtidos através da experimentação conforme um modelo de tentativa e erro. Neste ponto é importante destacar que o *Cluster Linux SSolar* do CBPF foi fundamental para tais simulações, pois permitiu a otimização do tempo.

7. Bibliografia

[BOVE] – http://www.bovespa.com.br

[HAYK] – Haykin, S. **Redes Neurais, Princípios e Prática**. 2.ed. Porto Alegre: Bookman, 2001.

[CHAK] - Chakraborty, K. Forecasting the behavior of multivariate Time Séries Using Neural Networks. Neural Networks, E.U.A., n. 5, p. 961-970, 1992.

[MATL] - Matlab Neural Network Toolbox User's Guide 1992-2002 by The MathWorks, Inc – Versão 4.

[REFE] - Refenes, A. N. Francis, G, 1992. Currency exchange rate prediction and neuralnetwork design strategies. Neural computing & Aplications Journal. Londres, v. 1, n.1, p. 46-58.

[TANG] – Tang, Z.,. Time séries forecasting using neural network vs. Box- Jenkins methodology. Artificial Neural Network: forecasting time series. E.U.A., IEEE Press, p. 20-27, 1994.