

**ALINE PASSOS FICHER**

**APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS NA PREVISÃO DO  
COMPORTAMENTO DO MERCADO DE AÇÕES**

São Paulo/SP  
2018

**ALINE PASSOS FICHER**

**APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS NA PREVISÃO DO  
COMPORTAMENTO DO MERCADO DE AÇÕES**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do título em  
Engenharia Financeira - MBA

Orientador: Prof. Dr. Danilo Zucolli  
Figueiredo

São Paulo/SP  
2018

## **AGRADECIMENTOS**

Aos meus ídolos, meus pais Ademir e Heleneusa, obrigada pelo amor incondicional e pelo exemplo de vida, especialmente à minha mãe que hoje não está mais na terra, mas que nunca deixou de me incentivar mostrar o valor do estudo, e que todo e qualquer esforço em prol do próprio desenvolvimento é necessário e válido. Também sou grata ao meu irmão Henrique, que me ensinou valores importantes e contribuiu com a minha educação. Não posso deixar de agradecer o meu namorado Evandro, que esteve ao meu lado durante todos os meses de elaboração desse trabalho e curso, o qual me ajudou a superar a rotina incessante com seu amor e carinho. Sou grata a todos os professores e colegas que fiz, o quais despertaram em mim um mundo novo cheio de números, nuances e possibilidades, todos foram especialmente essenciais na minha vida acadêmica.

## RESUMO

O mercado de ações é considerado bastante volátil, porém é uma opção de investimento procurada tendo em vista que pode gerar alto retorno. A previsão de tendências nesse mercado não é nada simples, pois além de diversos fatores econômicos e políticos é um mercado bastante suscetível a fatores psicológicos também. Os tradicionais métodos hoje utilizados, que são as análises técnicas e fundamentalistas não se mostram plenamente capazes de identificar relações não lineares entre as variáveis que podem influenciar o preço de uma ação, bem como seus movimentos de alta e baixa, sendo necessário o uso de técnicas mais específicas como as Redes Neurais Artificiais (RNAs) para mapear esses movimentos.

Uma RNA é uma ferramenta que simula habilidades do aprendizado do cérebro humano, possuindo a capacidade de modelar funções não lineares em ambientes complexos e que possuem informação incompletas ou que apresentam ruídos.

Neste trabalho, o objetivo é demonstrar através da utilização das RNAs, o desenvolvimento de modelos capazes de realizar a previsão de tendências, sejam elas de alta ou de baixa no mercado de ações observada sua relação com variáveis macroeconômicas como a taxa de câmbio, inflação e a taxa de juros de longo e curto prazo.

Os modelos estudados apresentaram um elevado grau de acerto na previsão de tendências, sendo possível concluir que esta é uma ferramenta poderosa no auxílio do gerenciamento de uma carteira de investimentos.

Palavras chave: Redes Neurais, RNAs, Previsão

## **ABSTRACT**

The stock market is considered quite volatile but is a very searched option of investment that can provide high return. The prediction of trends in this market is not at all simple, because in addition to several economic and political factors is a very susceptible Market, psychological factors affects as well. The traditional methods used, which are the technical and fundamentalist analyzes, do not show themselves fully capable of identifying non-linear relations between the variables that can compose the price of an action, as well as their movements of high and low, being necessary the use of techniques more specific such as Artificial Neural Networks to map these movements.

RNA is a tool that simulates learning abilities of the human brain, having the ability to model non-linear functions in complex environments that have incomplete or noisy information.

In this work, the objective is to demonstrate through the use of the RNA, the development of models capable of forecasting trends, whether high or low in the stock market, observed its relation with macroeconomic variables such as the exchange rate, inflation and the long and short-term interest rate.

The models studied presented a high degree of accuracy in the forecast of trends, and it is possible to conclude that this is a powerful tool to help the management of an investment portfolio.

**Keywords:** Artificial Neural Network, RNAs, Prevision

## LISTA DE TABELAS

Tabela 1 –.....	37
Tabela 2 –.....	37
Tabela 3 –.....	38
Tabela 4 –.....	38
Tabela 5 –.....	42
Tabela 6 –.....	42
Tabela 7 –.....	42
Tabela 8 –.....	43
Tabela 9 –.....	43
Tabela 10 –.....	43
Tabela 11 –.....	47
Tabela 12 –.....	47
Tabela 13 –.....	48
Tabela 14 –.....	48
Tabela 15 –.....	48
Tabela 16 –.....	49
Tabela 17 –.....	49
Tabela 18 –.....	49

## LISTA DE GRÁFICOS

Gráfico 1 – .....	34
Gráfico 2 – .....	39
Gráfico 3 – .....	39
Gráfico 4 – .....	40
Gráfico 5 – .....	40
Gráfico 6 – .....	43
Gráfico 7 - .....	44
Gráfico 8 - .....	44
Gráfico 9 - .....	45
Gráfico 10 - .....	45
Gráfico 11 - .....	46

## LISTA DE FIGURAS

Figura 1 – Neurônio.....	10
Figura 2 – Modelo McCulloch e Pitts.....	11
Figura 3 – Função Degrau.....	14
Figura 4 – Função Linear.....	14
Figura 5 – Função Sigmóidal.....	15
Figura 6 – Função Sigmóidal Linear.....	15
Figura 7 – Função Tangente Hiperbólica.....	16
Figura 8 – Função Tangente Hiperbólica Linear.....	16
Figura 9 – Rede de uma Única Camada.....	19
Figura 10 – Rede Direta de Múltiplas Camadas.....	20
Figura 11 – Rede Cíclica Simples.....	20
Figura 12 – Modelo não Linear de um Neurônio.....	25



## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1 Motivação.....	1
1.2 Mercado Acionário e os Tipos de Ações .....	2
1.2.1 .....	2
1.2.1.1 Análise Técnica.....	3
1.2.1.2 Teoria de Dow.....	3
1.2.1.3 Análise Fundamental.....	5
1.3 Hipótese do Mercado Eficiente.....	6
1.4 Modelos Neurais de Previsão.....	7
1.5 O Mercado de Ações no Brasil.....	8
1.6 Objetivos.....	9
1.7 Estrutura da Monografia.....	9
 <b>2. REFERENCIAL METODOLÓGICO.....</b>	 <b>10</b>
2.1 As Etapas de Implementação de uma RNA.....	10
2.2 Tipos de Função de Ativação.....	13
2.2.1 Função Degrau.....	14
2.2.2 Função Linear .....	14
2.2.3 Função Sigmóidal.....	14
2.2.3.1 Função Sigmóidal Linear.....	15
2.2.4 Função Tangente Hiperbólica.....	15
2.2.4.1 Função Tangente Hiperbólica Linear.....	16
2.3 Tipos de Rede.....	17
2.4 Tipos de Conexão.....	19
2.5 Tipos de Aprendizagem.....	21
2.6 Indicadores.....	23
2.6.1 Média Móvel.....	23
2.6.2 Volume.....	24
2.6.3 Momento.....	24
2.6.4 Tendência.....	24
2.7 O <i>Back Propagation</i> .....	24
2.7.1 <i>Perceptron</i> .....	25
2.7.2 <i>Perceptron</i> Multicamadas.....	26
2.7.3 O Algoritmo.....	26
2.7.4 Ajustes dos Pesos de Camada de Saída.....	28
2.7.5 Ajustes dos Pesos – Camada Escondida.....	29
 <b>3. METODOLOGIA PARA ESTIMAÇÃO DE RESULTADOS.....</b>	 <b>32</b>
3.1 O modelo de RNA utilizado.....	32

3.2 Topologia e Arquitetura.....	35
3.3 Algoritmo de Treinamento.....	35
3.4 Medida de Performance.....	35
<b>4. ESTIMAÇÕES E RESULTADOS.....</b>	<b>37</b>
<b>5. CONCLUSÕES.....</b>	<b>51</b>
<del>6.</del> <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>52</b>

## 1. INTRODUÇÃO

### 1.1 Motivação

O mercado financeiro é influenciado por diversos fatores, tendo de se levar em consideração condições econômicas e políticas, além fatores psicológicos e sociais que são altamente correlacionados entre si de uma maneira bastante complexa. Existem diversos estudos na área de finanças e economia que tem sido realizados afim de entender melhor a relação entre os preços das ações e as variáveis macroeconômicas, tendo em vista que os preços dos ativos estão diretamente relacionados aos fundamentos macroeconômicos.

É possível observar que os preços das ações apresentam um padrão de oscilação de alternância entre seus movimentos de alta e de baixa ao longo do tempo. Durante a década de 1960 foi desenvolvida a teoria da Hipótese do Mercado Eficiente que pressupõe que os preços exercidos no mercado são definidos por toda informação disponível, onde não há a possibilidade de ganhos extraordinários mesmo que uma outra corrente acredite que seja sim possível extrair altos retornos considerando essas mesmas informações que são públicas a todo o mercado. Nos dias atuais, o desenvolvimento da computação e outras técnicas avançadas têm ajudado em uma melhor compreensão de sistemas denominados complexos e não lineares, tendo em vista que os métodos de análises hoje existentes e os modelos econométricos tradicionais não garantem uma elevada taxa de retorno por um período consistente de tempo (FAMA, 2008). Existe um grau bastante alto de incerteza nas relações que se estabelecem entre as variáveis de interesse, onde o padrão de flutuação de uma delas pode acarretar no desempenho das demais (BRESSAN et al., 2005 e KIM, 2003), onde os métodos não lineares tradicionais não conseguem acompanhar essa volatilidade existente no mercado de ações (CRUZ, 2002). Logo, o estudo de modelos que consigam captar melhor as relações não lineares, surge como uma grande alternativa e oportunidade para a melhor compreensão do comportamento deste mercado tão volátil.

A teoria das Redes Neurais Artificiais (RNAs) surgiu a fim de explorar sistemas complexos, e hoje em dia é aplicada em diferentes áreas como na previsão de clima, na transmissão de energia elétrica, na área da saúde, etc. Os modelos baseados nas RNAs possuem a capacidade de identificar e mapear as relações não lineares entre variáveis, de lidar com dados muitas vezes incompletos ou com ruídos, além de não requererem nenhuma suposição inicial sobre os dados (VELLIDO et al., 1998).

Em economia, Leung et al. (2000) concluíram que as RNAs apresentam melhor desempenho na previsão dos índices de ações comparado com diversos métodos estatísticos. Cruz (2002) realizou uma comparação entre o modelo auto regressivo integrado de médias móveis (ARIMA) e as RNAs na previsão da taxa de câmbio entre o dólar americano e o australiano, sendo que o resultado obtido através das RNAs foi 4 vezes melhor do que o apresentado pelo modelo ARIMA. Dentro desse cenário, é possível verificar significativo interesse do uso de modelos de inteligência artificial, como as RNAs, para auxiliar na tomada de decisões no mercado financeiro.

## **1.2 Mercado Acionário e os Tipos de Ações**

A bolsa de valores é um mercado organizado onde são negociados valores mobiliários, tais como ações e opções sobre ações. O mercado de ações possui a função de negociar esses títulos, que são caracterizados por serem de renda variável e emitidos pelas sociedades, representando a menor parte da fração do capital de uma empresa.

As ações podem ser classificadas como *ordinárias* ou *preferenciais*, sendo respectivamente aquelas que garantem ao acionista o direito de voto em assembleias gerais e proporcionam a participação nos dividendos (resultados) ou então, aquelas que garantem ao acionista uma prioridade no reembolso do capital no caso da dissolução da sociedade e no recebimento dos dividendos e que também podem garantir o direito ao voto.

As ações são conhecidas por serem de renda variável, podendo ser convertidas em dinheiro a qualquer momento, possuem em sua essência uma volatilidade que desafia cotidianamente aqueles que buscam entender o comportamento de seus preços.

## **1.2.1 Análises Técnica e Fundamental**

### **1.2.1.1 Análise Técnica**

A primeira formulação de uma teoria sobre o estudo do movimento dos preços das ações surgiu em 1882 com Charles Henry Dow, juntamente com Edward Davis Jones e Charles Bergstresser, que fundaram os conhecidos *Wall Street Journal* e a *Dow Jones & Company*, sendo ainda hoje, duas gigantes no mercado financeiro.

Na análise técnica se constitui a ideia de que os movimentos sejam eles de tendência para baixa ou alta do preço de uma ação podem ser determinados por padrões que são recorrentes, sendo possível prever estes movimentos. A análise técnica se utiliza de gráficos que mapeiam transações como preços de abertura, fechamento, valor máximo ou mínimo e volume negociado para identificar os padrões que possam auxiliar na decisão de compra ou venda de um determinado tipo de papel.

### **1.2.1.2 Teoria de Dow**

Dow defendia que a criação de índices poderia ajudar os investidores a identificar tendências no mercado financeiro, por isso propôs a criação dos índices *Dow Jones Transportation Average*, obtido a partir dos preços de 20 grandes empresas ferroviárias, e o índice *Dow Jones Industrial Average*, calculado com base nos preços de 30 grandes empresas estadunidenses.

Os estudos de Charles Dow foram interpretados por 6 princípios, sendo eles:

- **Princípio 1:** os preços dos índices são formados por ideias e conceitos de diversos investidores, sendo que alguns se utilizam das melhores informações disponíveis no mercado enquanto outros não, esse pressuposto traz a ideia de que as variações diárias dos preços já incluem em seu preço, possíveis eventos que irão acontecer e que são desconhecidos pela maioria dos investidores;
- **Princípio 2:** pressupõe que existem 3 tipos de tendências de movimento, a primária, secundária e terciária. A primeira se caracteriza por acreditar que o mercado em geral seguirá tendências mais importantes e latentes no momento, chamadas de primárias, que duram em geral um ano ou mais, provocando grandes valorizações e desvalorizações de preço. Ao longo da tendência primária ocorre a tendência secundária, durando de 2 semanas a 4 meses, e a tendência terciária durando de seis a três semanas;
- **Princípio 3:** existem 3 fases de tendências, sendo a primeira de alta, que é estimulada por investidores qualificados e que possuem a melhor percepção do momento. Como nem todos possuem o mesmo nível de conhecimento e informação, isso faz com que os que possuem melhor percepção adquiram os papéis mais baratos. Na segunda fase, as compras aumentam acelerando a tendência, e na terceira fase há grandes altas, reflexo da segurança dos demais agentes de mercado ao observar os movimentos da primeira e segunda fase;
- **Princípio 4:** a tendência pode ser revertida entre índices de composições distintas;
- **Princípio 5:** para Dow, volume e tendência se relacionam de maneira que em uma tendência de alta, é esperado que o volume negociado também aumente com valorização dos ativos, e na baixa esse volume também caia, levando em consideração a desvalorização dos ativos;

- **Princípio 6:** a tendência continua até que surja um sinal de que houve reversão no mercado. Para que a tendência mude completamente, é preciso que essa se mostre definitiva.

Apesar da análise técnica ser criticada por ser subjetiva, onde os agentes podem interpretar de maneiras diferentes os gráficos e requerer certo tipo de conhecimento avançado para sua realização, a análise técnica é a mais utilizada pelos participantes do mercado.

### 1.2.1.3 Análise Fundamental

A análise fundamental procura medir o valor intrínseco de um ativo com a utilização do estudo das informações disponíveis no mercado e da empresa emitente da ação, analisando os ganhos de rentabilidade da mesma, a credibilidade dela no mercado, a expectativa das taxas de juros, as possíveis projeções para o crescimento do setor em que a empresa está inserida, etc. A maior vantagem desse tipo de análise comparada à análise técnica é que a análise fundamental apresenta melhor desempenho se comparada no longo prazo na previsão de estabilidade e crescimento, pois ela preserva a capacidade de prever tendências antes de serem captadas em um gráfico. Porém, ao se tratar de uma análise subjetiva, existe um grande custo para identificar fatores de influenciarem determinado ativo ou prever o tamanho da influência por ela provocada.

O principal viés de utilização de qualquer modalidade de análise no mercado financeiro torna-os incertos por natureza devido a serem movidos constantemente por inúmeros participantes que se utilizam de variadas estratégias e táticas em tempos distintos. As análises técnicas e fundamentalistas apesar de serem bastante utilizadas, ainda são também muito subjetivas, onde na prática, muitos autores da literatura afirmam que deixam a desejar comparadas aos modelos de RNAs na previsão de tendências no mercado financeiro em geral.

### 1.3 Hipótese do Mercado Eficiente

A Hipótese do Mercado Eficiente (HME) atesta que, em qualquer período em que é apurado o preço de uma ação, ele captura toda informação disponível em relação à ação. Como essa informação é consumida por todo o mercado, as variações podem ocorrer de maneira aleatória de acordo com a incidência dessas informações e a maneira com que se faz o uso constante dessas pelos agentes de mercado. Este conceito foi se tornando cada vez mais aceito entre o final dos anos 60 e o início dos 70, sob a rubrica *Theory of Random Walk*.

Segundo Fama (2008), que realizou um dos estudos mais importantes nesse sentido à época, intitulado "*Efficient Capital Markets: A Review of theory and Empirical Work*", existem três níveis de eficiência no mercado:

- **Forma Fraca de Eficiência:** os preços das ações já refletem toda informação contida no histórico dos preços e volumes negociados, onde o preço das ações já teria sido ajustado por analistas que tentam prever seu comportamento;
- **Forma Semiforte de Eficiência:** os preços das ações já refletem toda informação disponível aos investidores, não sendo possível superar o mercado, tendo em vista que o preço da ação com base em informações públicas já reflete seu valor real e outros analistas já efetuaram a análise e os preços reajustados com base nessas;
- **Forma Forte de Eficiência:** os preços das ações já refletem toda informação pública e sigilosa, sendo essas informações as que não são sabidas por todo o mercado, podendo influenciar a precificação das ações. Assim não é possível superar o mercado com base nas outras informações, pois outros analistas que também já possuíam essas informações já se utilizaram dessas e os preços já foram reajustados.

Para a teoria da HME, o mercado assume agentes racionais, onde os valores das ações só se alterariam caso alguma informação nova chegasse, porém é



possível observar na prática que algumas ações apresentam volatilidade nos preços que não podem ser explicadas por essa teoria. Para Cruz (2002), os estudos pouco mostram uma correlação entre a chegada de novas informações no mercado e a variação dos preços, existindo uma volatilidade que não pode ser apenas explicada por informações disponíveis no mercado.

A teoria do mercado eficiente pressupõe que os preços das ações não se comportam de maneira previsível e sim aleatória, onde o melhor momento para ser estudado, é o momento atual. A economia comportamental tem tentado mostrar que em observações comportamentais do mercado, os preços não variam de forma tão aleatória quando a teoria do mercado eficiente defende, os preços sofrem muito com a influência da subjetividade e do comportamento humano.

A teoria da HME é importante, pois contradiz todas as outras formas de análise, ou seja, é impossível construir modelos que obtenham retornos extraordinários no mercado. Existe uma contradição na literatura sobre a validade da HME (ZHANG et al., 1998), grande parte dos trabalhos a rejeita tendo em vista que a HME pressupõe que toda informação está disponível a todos os participantes da mesma forma, sem custos. Outro ponto criticado é de que os preços não variam só em função das informações disponíveis tendo de se considerar também os fatores psicológicos dos agentes participantes do mercado, além de que os preços também são formados por agentes que possuem informações privilegiadas e melhores técnicas de previsão.

#### **1.4 Modelos Neurais de Previsão**

As RNAs podem funcionar como aproximadores universais entre as informações disponíveis no mercado acionário. Como a maioria dos modelos se apresentam de forma não linear e com maior número de parâmetros, as RNAs possuem a capacidade de modelar de maneira adequada um maior número de séries do que os modelos lineares, resultado disso é a grande integração dos modelos de RNAs com modelos estatísticos.

### 1.5 O Mercado de Ações no Brasil

Principalmente até o início da década de 90, a estrutura do mercado de ações no Brasil possuía elevado nível de intervenção do governo, considerada uma estrutura fechada. Segundo Grôppo (2004, p.109), existia um problema grande de crescente inflação e um controle excessivo sobre o fluxo de capital. A partir de então, o Brasil começou a adotar uma série de medidas de desregulamentação, redução do papel do estado e modernização, visando abrir um caminho para uma economia mais flexível, objetivando a atração de maiores investimentos externos e, então, logo o mercado de capitais passou a ter uma maior importância em substituição ao mercado do crédito.

Em 1994, com o advento do Plano Real, o Brasil conseguiu controlar a inflação e atingiu certa estabilidade de preços, transformando-se em um ambiente bastante favorável para o amadurecimento da bolsa de valores, sendo que o único agravante retardador desse processo era a incerteza das taxas de juros que perduraria durante um bom tempo (GRÔPPO, 2004).

Os recentes juros mais baixos têm ajudado a atrair a atenção para aplicações em ações, seja por meio direto ou através de fundos de investimento. Para o ano de 2018, os investidores já esperavam seus ativos em patamares de preços mais elevados ocasionados pelo aumento do PIB (Produto Interno Bruto) e uma pequena melhora da economia no final de 2017. Analistas atuais defendem que o período de eleições pode contribuir para que as projeções do índice IBOVESPA ultrapassem 90 mil pontos, o que seria um fato histórico para a Bolsa de Valores no Brasil, mas caso o ajuste fiscal fosse interrompido durante 2018, essa projeção cairia para valores próximos aos 50 mil pontos.

A previsão é de que o ciclo de queda da taxa de juros esteja se encerrando e de que estas sejam taxas altas para longo prazo, levando em consideração outro fato importante que é a volatilidade dos ativos em períodos de eleições, podendo aumentar e trazer riscos para os investidores tornando o cenário de negociações de ações cada vez mais interessante de ser estudado comparado ao cenário

macroeconômico, e uma grande oportunidade de entender melhor como as variáveis macroeconômicas podem influenciar nas decisões de investimento.

## 1.6 Objetivos

O objetivo geral desse trabalho é construir um modelo baseado em RNAs, através de um modelo de rede *feedforward* com o uso do algoritmo MBP (*multiple backpropagation*) para a previsão do comportamento do mercado acionário no Brasil, mais especificamente deseja-se:

- Utilizar RNAs para realizar a previsão das tendências do Índice da Bolsa de Valores de São Paulo, tendo como variáveis de entrada das RNAs algumas variáveis macroeconômicas como a taxa de câmbio, a inflação e a taxa de juros de longo e curto prazo;
- Identificar as variáveis que possuem o papel mais significativo para a realização das previsões;
- Analisar a “arquitetura” da RNA que melhor se aplica na previsão de tendência de preços de ações com base em variáveis macroeconômicas.

## 1.7 Estrutura da Monografia

O restante desta monografia está organizado da seguinte forma. No Capítulo 2 é apresentado o referencial teórico sobre RNAs, no Capítulo 3 é abordada a metodologia empregada nos modelos desenvolvidos neste trabalho seguido então, das estimações e resultados no Capítulo 4. O Capítulo 5 apresenta a conclusão geral da monografia.

## 2. REFERENCIAL METODOLÓGICO

Uma Rede Neural Artificial, ou RNA, é o modelo mais simples de *Rede Biológica*, que procura reproduzir algumas das funções cerebrais. Apesar das RNAs serem modelos simplificados, elas conseguem reproduzir adequadamente determinados processos comportamentais do cérebro humano, como o aprendizado de tarefas simples e o reconhecimento de padrões.

A célula básica do cérebro humano possui cerca de 10 bilhões de neurônios, onde cada um deles está conectado a milhares de outros que processam informações continuamente. São três componentes básicos que compõem o neurônio, conhecidos como soma, dendritos e axônio.

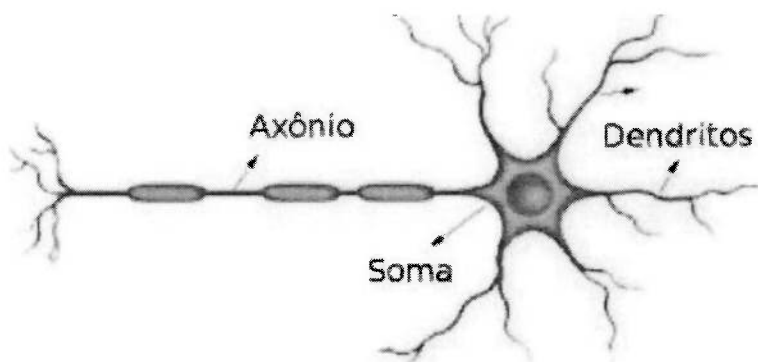


Figura 1 - Neurônio

Os dendritos são responsáveis por receber impulsos nervosos de outros neurônios através de um processo conhecido como *informação*. Essa informação é transportada até o corpo da célula, que é responsável por processá-la gerando novas frequências e novos impulsos. Esses impulsos são transmitidos através do axônio para os dendritos dos próximos neurônios que fazem parte da cadeia de informação através de um processo conhecido como *sinapse*. São as sinapses que controlam as transmissões entre os impulsos, ou seja, entre o fluxo de informações no cérebro, e são através delas que são estabelecidas as redes neurais biológicas, estabelecendo as funções em que o cérebro é responsável, como o aprendizado, as emoções e o pensamento. Destaca-se que cada neurônio está conectado a cerca de  $10^3$  a  $10^4$  outros neurônios.

O primeiro modelo de neurônio artificial, proposto por McCulloch e Pitts em 1943, possuía como entradas as variáveis  $x_1, x_2, \dots, x_n$ , estando associadas aos dendritos. Para que fosse possível a simulação do comportamento de um neurônio, as sinapses tinham pesos associados a elas, sendo esses pesos representados por variáveis  $w_1, w_2, \dots, w_n$ , que podiam assumir valores positivos ou negativos, de modo a representar a amplificação ou a atenuação dos sinais cerebrais. Neste modelo, o neurônio possuía apenas uma saída que podia ser excitado quando a soma dos impulsos ( $\sum x_i \cdot w_i$ ) ultrapassava um limiar de excitação ( $\theta$ ), sendo um mecanismo simples que procurava realizar o somatório e decidir, com base neste, se o neurônio deveria ser ou não excitado.

Uma deficiência observada neste modelo por Minsky (1969) é de que a rede possuía apenas uma camada e os pesos eram fixos, porém a partir deste pressuposto, vários outros modelos foram surgindo tendo como principal modificação o ponderamento de diferentes funções de ativação, produzindo qualquer saída e não apenas uma saída binária (possível excitação ou não).

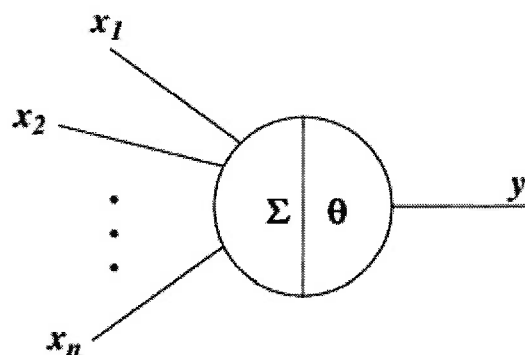


Figura 2 – Modelo McCulloch e Pitts

Frank Rosenbatt em 1958 propôs o modelo nomeado *perceptron*, em que as RNAs possuem sinapses que se ajustam e podem ser treinadas afim de representar padrões e, em 1986, Hopfield publicou seu estudo sobre o algoritmo de treinamento chamado de *backpropagation* – BP (retro-propagação).

Cada “arquitetura” de RNA pode se adaptar melhor dependendo do determinado tipo de problema que está sendo analisado, sendo que os parâmetros que devem ser levados em consideração para a definição da RNA são: a função de ativação

dos neurônios, o número de camadas da rede, os números de neurônios em cada camada e o tipo de conexão entre os neurônios da rede. A seleção destes parâmetros depende do tipo de problema que está sendo abordado (ZHANG et al., 1998).

## 2.1 As etapas de implementação de uma RNA

Após ter sido definido o problema em que a RNA será aplicada e as variáveis que serão utilizadas, são basicamente quatro etapas que constituem a implementação de uma RNA:

- **Coleta e Separação dos Dados:** essa etapa consiste na separação dos dados e sua segregação em dois grupos. O primeiro grupo para o treinamento da rede pretendida e o outro na resolução de problemas que se aproximam do desejado, observado o comportamento desses dois grupos;
- **Seleção de Parâmetros:** são determinados o número de neurônios da camada, os tipos de conexões entre estes neurônios, o tipo de função de ativação e as variáveis de treinamento de rede. Na maior parte dos problemas, esses parâmetros devem ser modificados até que se obtenha o resultado esperado;
- **Treinamento:** é nesta fase que os pesos de uma RNA se modificam, e se ajustam ao ambiente em que estão inseridas;
- **Teste e Validação:** o conjunto que foi separado previamente é submetido à RNA, e então é validado. Caso a RNA não atinja o resultado esperado, deve-se retornar às etapas iniciais.

Em relação aos dados de entrada para a RNA, na literatura não existem muitas discussões sobre a necessidade de tratamento desses dados ou de qual tipo de tratamento utilizar. Shanker et al. (1996) concluiu que o pré-processamento dos

dados escolhidos é benéfico em termos de classificação e erro médio quadrático, porém a utilização do tratamento pode levar a um retardo no tempo de treinamento.

Existem várias maneiras de realizar as transformações dos dados para as RNAs na literatura, as mais utilizadas são (ZHANG. et al., 1998):

- Transformação Linear para  $[0, 1]$ :  $x_n = \frac{(x_0 - x_{min})}{(x_{max} - x_{min})}$  (4.7)
- Transformação linear para  $[a, b]$ :  $x_n = \frac{(b-a)(b_0 - x_{min})}{(x_{max} - x_{min}) + a}$  (4.8)
- Normalização estatística:  $x_n = \frac{(x_0 - \bar{x})}{s}$  (4.9)
- Normalização simples:  $x_n = \frac{x_0}{x_{max}}$  (4.10)

Nas expressões acima,  $x_n$  é o dado normalizado,  $x_0$  é o dado original,  $x_{min}$ ,  $x_{max}$ ,  $\bar{x}$  e  $s$  são o mínimo, o máximo, a média e o desvio padrão de um conjunto de dados, respectivamente.

É importante observar, que após o treinamento realizado com dados normalizados, em geral é necessário voltar os resultados às escalas originais para que se obtenham saídas coerentes (ZHANG et al., 1998).

## 2.2 Tipos de Função de Ativação

A função de ativação ou função de transferência define a relação entre os neurônios de entrada e saída de uma rede. Em geral, a função de ativação introduz certo grau de não-linearidade importante para o desempenho de uma RNA. A função mais utilizada é a sigmoideal podendo se observar bons resultados também com a função hiperbólica como função de ativação (SHARDA et al., 1994). A maioria dos autores se utiliza de função de transferência em todas as camadas para melhores resultados (SHARDA, 1994 e ZHANG et al. 1998).

Existem diversos tipos de função de ativação para os neurônios de uma RNA, sendo eles:

### 2.2.1 Função Degrau

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (2.1)$$

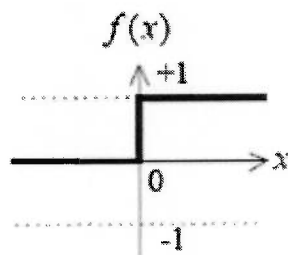


Figura 3

### 2.2.2 Função Linear

$$f(x) = x, \forall x \quad (2.2)$$

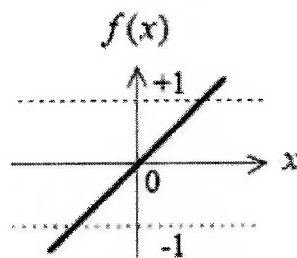


Figura 4

### 2.2.3 Função Sigmoidal

A função sigmoidal é o tipo de função de ativação mais utilizado. Essa função assume valores entre 0 e 1, sendo dada por:



$$f(x) = \frac{1}{1+\exp(-x)} \quad (2.3)$$

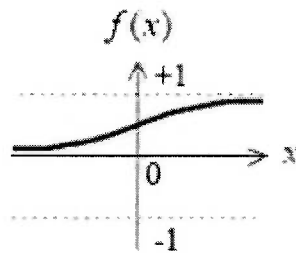


Figura 5

### 2.2.3.1 Função Sigmoidal Linear

Uma variação da função sigmoidal é a função sigmoidal linear:

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

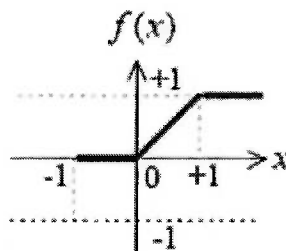


Figura 6

### 2.2.4 Função Tangente Hiperbólica

Também bastante utilizada, a função tangente hiperbólica assume valores entre -1 e 1:

$$f(x) = \frac{2}{1 + \exp(-2x)} - 1 \quad (2.5)$$

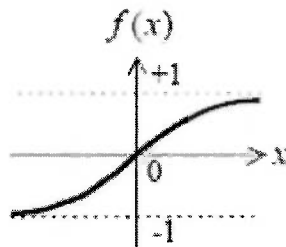


Figura 7

#### 2.2.4.1 Função Tangente Hiperbólica Linear

Uma variação da função tangente hiperbólica, a função tangente hiperbólica linear é dada por:

$$f(x) = \begin{cases} x, & \text{se } -1 \leq x \leq 1 \\ 0, & \text{c. c.} \end{cases} \quad (2.6)$$

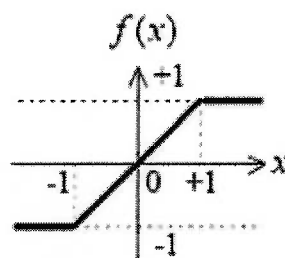


Figura 8

Nos problemas de previsão, é comum que seja aplicada uma normalização dos dados de entrada da RNA, de modo que todas as variáveis fiquem dentro de um mesmo intervalo. Esse intervalo depende em geral da função de transferência da camada de saída. Usualmente é utilizado o intervalo  $[0,1]$  para as funções sigmóides e  $[-1,1]$  para a função tangente hiperbólica (ZHANG et al. 1998 e KIM, 2000).

## 2.3 Tipos de Redes

Dependendo o número de camadas, as RNAs podem ser classificadas como:

- **Redes com uma única camada:** onde apenas uma camada de neurônios faz o processamento de informação entre um vetor de entrada e um de saída;
- **Redes com camadas múltiplas:** existe mais de uma camada de neurônios fazendo o processamento da informação entre a entrada e a saída. Essas camadas adicionais são chamadas de camadas escondidas ou ocultas.

Segundo Haykin (2001, p. 34-41), com a adição de uma ou mais camadas ocultas na rede, a RNA pode modelar aspectos mais complexos do problema a qual está submetida.

O número de neurônios da camada de entrada é correspondente ao número de variáveis que serão utilizadas como vetores iniciais do problema, sendo que a grande maioria dos autores fazem essa escolha de maneira empírica. Esta é uma escolha muito importante e crítica no estudo de uma RNA. Deve-se utilizar de entradas que possuam algum tipo de relação implícita com as variáveis de saída (ZHANG et al., 1998). Além disso, deve-se observar os pesos das camadas após o treinamento da rede, sendo que as entradas que apresentarem maiores pesos possuem maior influência no processamento da rede.

O número de camadas ocultas e o número de neurônios que farão parte delas são fatores decisivos para definir a performance de uma RNA. São nos neurônios da camada oculta que são processados os mapeamentos não lineares entre as camadas de entrada e saída e são identificados os padrões entre os dados. A maioria dos autores utiliza apenas uma camada oculta para previsão (ZHANG et al., 1998), sendo que Sharda et al. (1994) concluíram que o uso de duas camadas ocasiona numa arquitetura mais compacta, e com uma maior eficiência de treinamento. Zhang (1994) observou que as redes que se utilizavam de duas camadas ocultas tinham maiores precisões e Cybenko (1989) concluiu que o uso

de duas camadas são o suficiente para a resolução da grande maioria dos problemas que se utilizam de RNAs, inclusive problemas de previsão.

A maneira mais comum de escolha do número de camadas ocultas é de tentativa e erro. Youngohc (1993) observou que o aumento do número de camadas melhora o desempenho da rede, porém até certo ponto. Diversas regras empíricas foram estabelecidas para determinar o número de componentes de cada camada oculta, mas concluiu-se que não existe um padrão que possa ser seguido para todos os tipos de problemas que podem ser abordados por RNAs (ZHANG et al., 1998).

Geralmente o número de saída é igual ao número de saídas desejadas, sendo uma decisão simples. Na literatura existem duas maneiras de realizar previsões (ZHANG et al., 1998): um período à frente, que se utiliza de um único neurônio na camada de saída e realiza previsão de um único período, e a previsão de múltiplos períodos mais efetuada através do método iterativo – onde para cada saída existe apenas um componente de saída e esta é utilizada como entrada do período seguinte:

$$\bar{x} = f(x_1, x_{t-1}, \dots, x_{t-n}), \quad (4.1)$$

$$\bar{x}_{t+2} = f(x_{t+1}, x_t, x_{t-1}, \dots, x_{t-n+1}), \quad (4.2)$$

.

.

.

$$\bar{x}_{t+k} = f(x_{t+k-1}, x_{t+k-2}, \dots, x_{t+1}, x_t, x_{t-1}, \dots, x_{t-n+k-1}), \quad (4.3)$$

onde  $x_t$  é a observação no período  $t$ ,  $\bar{x}_t$  é a previsão para o período  $t$ ,  $f$  é a função estimada pela RNA.

Ou também pode ser estimada através do chamado método direto, no qual o número de períodos é igual ao número de componentes da camada de saída, e cada componente é responsável por um período:

$$\bar{x}_{t+1} = f(x_t, x_{t-1}, \dots, x_{t-n}), \quad (4.4)$$

$$\bar{x}_{t+2} = f_2(x_t, x_{t-1}, \dots, x_{t-n}), \quad (4.5)$$

.

.

.

$$\bar{x}_{t+k} = f_k(x_t, x_{t-1}, \dots, x_{t+n}), \quad (4.6)$$

onde as  $f_1, \dots, f_k$  são funções determinadas por cada  $k$ -ésimo neurônio. Zhang (1994) observou que os melhores resultados são estimados através do método direto, este é o método que será utilizado nesse trabalho, porém Weigend et al. (1992) e Hill et al. (1992) defendiam que os melhores resultados eram dados pelo método iterativo, o mais utilizado na literatura.

## 2.4 Tipos de Conexão

A seguir são apresentados os tipos de conexões dos neurônios em RNAs:

- **Rede Direta (*feedforward*):** a rede não possui ciclos, uma saída de um neurônio não é utilizada como entrada de uma camada anterior. Esse tipo de rede é a mais utilizada juntamente com a aplicação do algoritmo *backpropagation*.

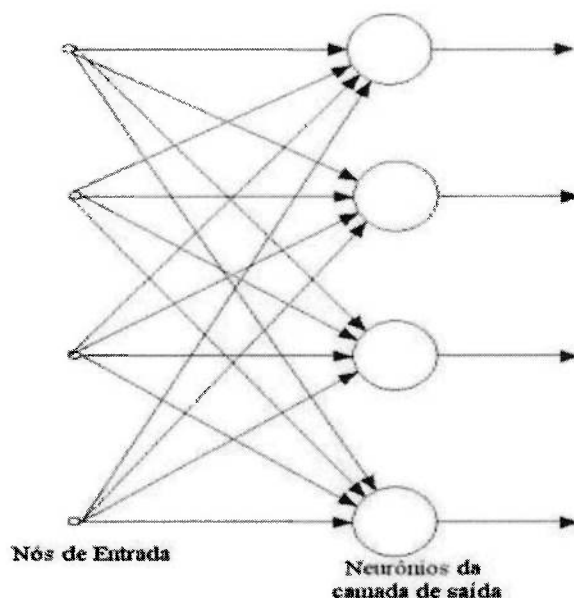


Figura 9 - Rede Direta de Única Camada (Haykin, 2001)

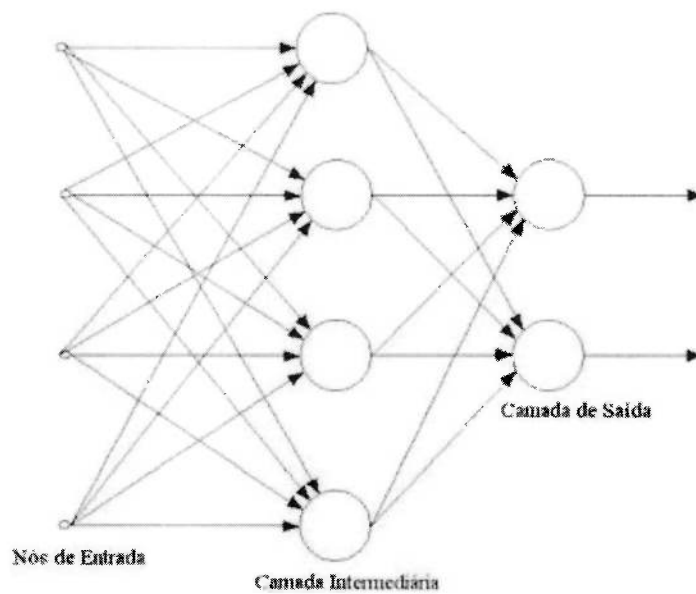


Figura 10 - Rede Direta de Múltiplas Camadas (Haykin, 2001)

- **Rede Cíclica (*feedback*):** a saída de algum neurônio é utilizada como entrada de algum neurônio pertencente à camada anterior. Essa é a rede mais indicada para problemas que possuem processamento temporal.

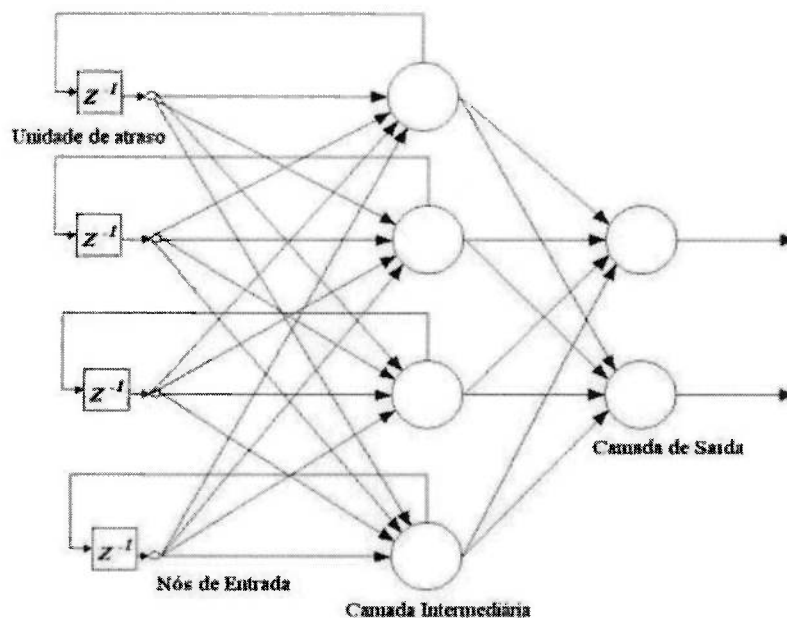


Figura 11 - Rede Cíclica Simples (Haykin, 2001)

## 2.5 Tipos de Aprendizagem

Para Haykin (2001), o tipo de aprendizado é determinado pela maneira pela qual ocorrem mudanças nos parâmetros da RNA. A rede é capaz de criar um padrão do ambiente em que está inserida através do processo iterativo entre os ajustes dos pesos e as conexões entre os neurônios quando estimulados. Este aprendizado permite decidir como os pesos das conexões serão alterados e também, se devem ser alterados e considerados na análise ou descartados para moldar a topologia da rede.

Para a aplicação de uma RNA, é necessário tanto coletar dados relacionados ao problema estudado e quanto segregar esses dados em dois conjuntos distintos. O primeiro conjunto é utilizado no treinamento da rede e o segundo é usado na validação do comportamento dessa rede na resolução dos problemas próximos ao resultado definido como pretendido, podendo se concluir se o comportamento da rede está de acordo com o desejado. É também bastante comum o uso de um terceiro grupo de dados, chamado de validação, utilizado para determinar um ponto de parada no processo de treinamento (WEIGEND et al., 1992).

Na literatura não existe uma conclusão definitiva de como definir o tamanho da amostra a ser coletada, essa dependerá da arquitetura da rede, do algoritmo de treinamento utilizado, complexidade do problema e da quantidade de ruído presente nos dados coletados. Em análises estatísticas, quanto maior o número de dados melhor o desempenho do resultado. Zhang et. Al (1998) e Nam e Schaefer (1998) realizaram testes com diferentes tamanhos de amostra e concluíram que o desempenho da rede melhora conforme aumenta o tamanho da amostra. Para Haykin (1994), uma amostra de tamanho muito elevado pode levar a um *overtraining*, fazendo com que a rede perca sua capacidade de generalização.

Também não existe uma solução dada para a decisão entre a escolha de quais dados farão parte do treinamento e quais irão ser utilizados para teste, porém existe um consenso de que a divisão entre os dados não é o mais importante e

sim o tamanho de cada um deles. É necessário que o conjunto de treinamento seja suficiente para que a RNA atinja uma avaliação satisfatória (HAYKIN, 1994).

O treinamento é a etapa principal no desenvolvimento de uma RNA, pois é nele que a rede aprende sobre o ambiente que está inserida e começa a se adaptar a ele através da variação dos pesos para tentar solucionar o problema proposto (GRÔPPO, 2004). É preciso atribuir valores iniciais aos pesos da rede de acordo com os padrões proposto pelo problema, onde a própria RNA acaba por modificar os pesos e conexões por uma regra pré-estabelecida de aprendizado. Após esta etapa, as RNAs identificam padrões e a capacidade de prever o valor de certas variáveis para situações que não foram apresentadas na etapa inicial de treinamento.

Os métodos de aprendizado aplicados às RNAs são classificados como descrito abaixo:

- **Aprendizado Supervisionado:** a topologia da rede é moldada afim de obter um comportamento definido para cada situação. São calculados erros entre os valores obtidos pela RNA e os valores esperados para cada situação, sendo que o algoritmo de aprendizado tem como objetivo minimizar os erros obtidos. Essa operação é realizada em cada etapa do treinamento, buscando uma solução, caso ela exista. A soma dos erros quadráticos costuma ser utilizada como índice de desempenho da rede e como função objetivo a ser minimizada pelo algoritmo de treinamento (BRAGA et al., 2000). A limitação do aprendizado supervisionado é o aprendizado de novas estratégias que não estejam no escopo do treinamento inicial.
- **Aprendizado Não-Supervisionado:** a rede neste caso não possui saídas desejadas a priori, utilizando-se apenas de vetores de entrada, tendo como objetivo diferenciar classes de padrões diferentes dentre os dados apresentados à rede. Isto é possível apenas quando existe redundância nos dados de entrada para que a rede possa identificar possíveis padrões ou características no problema analisado.



- **Aprendizado Híbrido:** é uma mistura dos dois tipos de aprendizado anteriormente mencionados, sendo uma parte dos pesos da RNA definida pelo aprendizado supervisionado, enquanto a outra é determinada pelo não-supervisionado.

O mais importante para medir a performance de uma RNA é a acurácia após a etapa de treinamento. Esse grau de acurácia geralmente é medido em termos do erro entre o valor desejado e o valor obtido na previsão (ZHANG et al. 1998). As medidas mais utilizadas são:

- Desvio absoluto médio (MAD):  $\frac{\sum |e_t|}{N}$  (4.11)

- Soma dos erros quadráticos (SSE):  $\sum (e_t)^2$  (4.12)

- Erro médio quadrático (MSE):  $\frac{\sum (e_t)^2}{N}$  (4.13)

- A raiz do erro médio quadrático (RMSE):  $\sqrt{MSE}$  (4.14)

- Erro percentual absoluto (MAPE):  $\frac{1}{N} \sum \left| \frac{e_t}{y_t} \right| (100)$  (4.15)

Nas definições acima,  $e_t$  é o erro individual de cada previsão,  $y_t$  é o valor atual e  $N$  é o número de períodos avaliados. A medida mais usada para acurácia na literatura é o erro médio quadrático (MSE).

## 2.6 Indicadores

### 2.6.1 Média Móvel

As médias móveis consistem em médias que são extraídas de dados sequenciais de um determinado intervalo de tempo, ela propõe informar o início ou o fim de uma tendência de preços. Uma média que apresenta 8 períodos, mostrará o preço médio do fechamento desses 8 períodos. No dia subsequente substitui-se o preço de fechamento do primeiro dia apurado pelo preço de fechamento do dia subsequente e assim sucessivamente, criando uma chamada “média móvel”.

$$MME_{hoje} = \frac{2 \times PF_{hoje} + (N-1) \times MME_{ontem}}{N+1} \quad (2.7)$$

onde,  $MME_{hoje}$  é a média móvel exponencial de hoje,  $PF_{hoje}$  é o preço de fechamento de hoje,  $N$  é o número de dias da média móvel exponencial e  $MME_{ontem}$  é a média exponencial de ontem.

### 2.6.2 Volume

É uma medida que expressa o valor do montante financeiro negociado durante um pregão, podendo se referir ao mercado como um todo ou a um ativo específico.

### 2.6.3 Momento

O momento representa a diferença dos preços de fechamento de determinado período e a evolução de tendência desses preços.

O momento de  $x$  dias é dado por:

$$Mx = PF_{hoje} - PF(hoje - x) \quad (2.8)$$

### 2.6.4 Tendência

Refere-se à direção dos preços representados graficamente, podendo assumir um padrão ascendente, descendente ou indefinido. A permanência da direção em determinado padrão é denominada tendência.

## 2.7 O Backpropagation

O *backpropagation* é um modelo não linear - representado esquematicamente na Figura 12 - formado por um conjunto de entradas  $X_n$ , um conjunto de pesos correspondentes representado por  $W_n$  e possuindo em sua estrutura um agente somador que adiciona as entradas já ponderadas pelos pesos junto a uma

função de transferência que limita a amplitude da saída do neurônio a um valor finito. Além disso, Hopfield (1982) denominou de “*bias*” um fator que tem efeito de aumentar ou diminuir a ação da função de ativação.

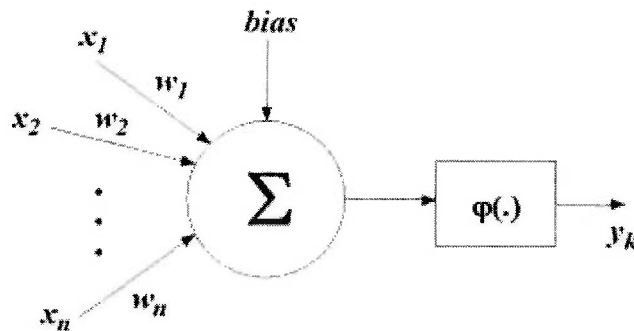


Figura 12 - Modelo Não Linear de um Neurônio (Hopfield, 1982)

Onde,

$$U_k = \sum_{j=1}^n (w_{kj} \cdot x_j) \quad (2.9)$$

$$Y_k = \varphi(U_k + H_k) \quad (2.10)$$

Sendo:

$x_1, x_2, \dots, x_n$  são os sinais de entrada do neurônio ( $k$ );

$w_{k1}, w_{k2}, \dots, w_{kn}$  são os pesos do neurônio ( $K$ );

$U_k$  é o resultado do somador;

$H_k$  sendo o bias;

$\varphi(.)$  a função de ativação e,

$Y_k$  o sinal de saída do neurônio  $K$ .

### 2.7.1 Perceptron

Em 1958, Rosenbalt concluiu uma forma simples de rede neural capaz de resolver problemas linearmente separáveis, que acabou se tornando pioneira segundo Haykin (1994). Também era utilizada para resolver problemas de classificação e capaz de convergir para uma superfície de decisão formada por um outro plano entre essas classes de classificação denominada *perceptron*.

### 2.7.2 *Perceptron* Multicamadas

O *Perceptron* que foi proposto por Rosenbalt tinha a capacidade de resolver apenas problemas que eram linearmente separáveis e, então, em 1986, Rumelhart propôs um algoritmo de treinamento, o *backpropagation*, que seria possível realizar o treino de redes multicamadas diretas, introduzindo o conceito de camadas ocultas e redes neurais que ficaram conhecidas como *perceptron multicamadas* (MLP – *Multi Layer Perceptrons*). Estes são os modelos de RNAs que são mais utilizados em aplicações práticas atualmente.

### 2.7.3 O Algoritmo

O processo de treinamento de uma RNA que é utilizada juntamente com o auxílio do *backpropagation* e envolve basicamente duas fases: A propagação do erro e a retro propagação do erro, que compara a saída obtida com a saída desejada onde os pesos são modificados enquanto forem apuradas diferenças.

As redes que utilizam o *Backpropagation* para o treinamento, geralmente trabalham com a regra delta generalizada, que é baseada na regra delta desenvolvida por Windrow e Hoff em 1960 para treinar o neurônio conhecido como Adaline. A regra delta consiste em um método de gradiente descendente para minimizar o erro quadrado total para a função de ativação linear, porém se a superfície de erro for complexa não há garantias de solução ótima, levando o algoritmo a convergir para uma solução estável (mínimos locais) (BRAGA, et al. 2000). A regra delta generalizada se utiliza de uma função de ativação semi-linear, no caso a sigmoide que é diferencial e não decrescente. É com essa função que o neurônio identifica o nível de atividade de sua entrada e a partir disso, é definida sua saída. É comum o algoritmo chegar a convergir para os mínimos locais, e para contornar esse problema algumas técnicas são utilizadas, como por exemplo, adotar a taxa de treinamento decrescente, adicionar nós intermediários e ruído aos dados. Essas soluções fazem com que a rede acelere o algoritmo *Backpropagation* reduzindo a incidência dos mínimos locais. (ARAÚJO. et al., 2012).

Todo processo de treinamento é baseado em função do sinal obtido no resultado do erro calculado, tendo em vista que este é responsável pela atualização dos pesos (SILVA, 2010). O sinal do erro na saída do neurônio definida como  $k$  (considerando a  $k$  - *ésima* amostra de treinamento), o sinal do erro é obtido através da diferença entre a saída desejada e a saída de ativação do neurônio  $k$  (saída do neurônio  $k$  na interação  $n$ ).

$$e_k(n) = d_k(n) - y_k(n) \quad (2.11)$$

O termo  $e_k$  corresponde ao erro calculado na saída do neurônio  $k$ , onde  $d_k(n)$  é a saída definida como desejada e a resposta dada para o neurônio  $k$  é  $y_k(n)$ .  $y_k(n)$  é definido por:

$$y_k(n) = \phi(v_k(n)) \quad (2.12)$$

Onde a função de ativação  $\phi$  que é associada ao neurônio  $k$  é definida pela sigmoide:

$$\frac{1}{1+e^{-v_k(n)}} \quad (2.13)$$

Neste caso a direção da curva e sua tendência é determinada pelo fator  $\alpha$  e a orientação sigmoide é dada pela direção do vetor peso  $w$ , ou seja, define a posição da função sigmoide com ao eixo da ordenada (BRAGA, et al. 2000). O sinal usado na função de ativação é dado por:

$$v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n) \quad (2.14)$$

O peso é dado por  $w_{kj}$  da sinapse que é conectada a saída  $j$  e a entrada do neurônio  $k$ .  $y_j(n)$  corresponde a saída do neurônio  $j$ ,  $m$  representa o número total de entradas excluindo o bias. Com a propagação do erro o treinamento é iniciado,  $x_i(n)$  representa os estímulos de entrada, nessa fase, os pesos das sinapses são iniciados com valores aleatórios e permanecem inalterados para

serem propagados pelas camadas intermediárias até a camada de saída (CARRARA, 1997).

$$y_j(n) = \phi(v_j(n)), \text{ onde} \quad (2.15)$$

$y_j(n)$ , é a saída do neurônio  $j$ , ou então,

$$y_j(n) = \phi(\sum_{i=0}^m w_{ji}(n)x_i(n)) \quad (2.16)$$

O desempenho local para ser medido exige a utilização de uma função custo, que neste caso é o erro quadrático (SILVA, 2010), através dessa função todos os erros de cada neurônio são processados:

$$E(n) = \frac{1}{2} \sum_{j \in c} e_{jk}^2(n) \quad (2.17)$$

$$E(n) = \frac{1}{2} \sum_{j=1} (d_k(n) - y_k(n))^2 \quad (2.18)$$

A cada novo ciclo que se inicia a função assume um novo valor em função da atualização dos pesos, podendo se verificar como os erros vão decrescendo na fase de treino.

#### 2.7.4 Ajustes dos Pesos da Camada de Saída

Quando o treino entra na fase de retro propagação do erro, os bias e o peso atribuído aos ajustes são baseados no gradiente e da função do erro quadrático. O cálculo do gradiente entra em um processo de repetição para cada neurônio, e faz com que os pesos do erro na superfície caminhem de maneira a diminuir na direção contrária ao gradiente da função custo (CARRARA, 1997).

O algoritmo *backpropagation* aplica uma correção “proporcional” à derivada parcial do erro quadrático para realizar o ajuste dos pesos sinápticos da camada

de saída. A relação  $\frac{\partial E(n)}{\partial w_{kj}}$ , estabelece a direção que se dá a busca no espaço da sinapse  $w_{kj}$ .

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E}{\partial y_j} \frac{\partial y_k}{\partial v_k} \frac{\partial v_k}{\partial w_{kj}} \quad (2.19)$$

Diferenciando as equações obtém-se:

$$\frac{\partial y_k}{\partial v_k} = -e_k \phi'(v_k) y_j \quad (2.20)$$

Pela regra delta obtém se a sinapse corrigida, onde  $\mu$  é a taxa de aprendizagem:

$$\Delta w_{kj}(n) = -\mu \frac{\partial E}{\partial w_{kj}} \quad (2.21)$$

Substituindo 2.18 em 2.19, onde  $\delta_k$  é o gradiente local:

$$\Delta w_{kj}(n) = \mu e_k \phi'(v_k) y_j \quad (2.22)$$

$$\Delta w_{kj}(n) = \mu \delta_k y_j \quad (2.23)$$

Após a apuração da correção sináptica o valor do peso atualizado é dado por:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (2.24)$$

### 2.7.5 Ajustes dos Pesos - Camadas Escondidas

Nesse caso não existe uma resposta que é desejada para o erro, apenas uma estimativa deste. O ajuste do peso  $w_{ji}(n)$  em relação à retro propagação do erro é dada por:

$$\frac{\partial E(n)}{\partial w_{ji}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial w_{kj}} \quad (2.25)$$

Na diferenciação das equações obtém-se:

$$\frac{\partial y_j}{\partial v_j} = x_i \quad (2.26)$$

$$\frac{\partial v_{kj}}{\partial w_{kj}} = \phi'(v_j) \quad (2.27)$$

O cálculo da derivada parcial  $\frac{\partial E}{\partial y_j}$  é dada por:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial v_k} \frac{\partial v_k}{\partial y_j} \quad \text{onde,} \quad (2.28)$$

$$\frac{\partial v_k}{\partial y_j} = \sum_k \frac{\partial (w_{kj} y_j)}{\partial y_j} = w_{kj} \quad (2.29)$$

O cálculo do termo  $\frac{\partial E}{\partial v_k}$ :

$$\frac{\partial E}{\partial v_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial v_k} = -(d_k - y_k) \phi'(v_k) \quad (2.30)$$

Dessa forma o gradiente local pode ser definido como:

$$\delta_k = -(d_k - y_k) \phi'(v_k) \quad (2.31)$$

Substituindo 2.27 e 2.29 em 2.26:

$$\frac{\partial E}{\partial y_j} = -\sum_k w_{kj} \delta_k \quad (2.32)$$

Substituindo as equações 2.24, 2.25 e 2.28 em 2.23, obtemos:



$$\frac{\partial E}{\partial w_{ji}} = -(\sum_k w_{kj} \delta_k) x_i \phi'(v_j) \quad (2.33)$$

O ajuste do peso  $w_{ji}$  deve ser feito de modo em direção contrária ao gradiente afim de minimizar o erro:

$$\Delta w_{ji}(n) = -\mu \frac{\partial E}{\partial w_{ji}} \quad (2.34)$$

$$\Delta w_{ji}(n) = -\mu (\sum_k w_{kj} \delta_k) x_i \quad (2.35)$$

E então o gradiente local é dado por:

$$\delta_j = (\sum_k w_{kj} \delta_k) \phi'(v_j) \quad (2.36)$$

A correção que é aplicada à sinapse:

$$\Delta w_{ji}(n) = \mu \delta_j x_i \quad (2.37)$$

Portanto, o valor atualizado do peso da primeira camada é dado por:

$$w_{ji}(n+1) = \Delta w_{ji}(n) + w_{ji}(n) \quad (2.38)$$

Os pesos atribuídos inicialmente à rede são inicializados com base em valores aleatórios, e somente depois da propagação dos primeiros valores de entrada e os desvios estejam calculados é que os pesos são atualizados com os valores relativos aos dados do problema que está sendo apresentado.

### 3. METODOLOGIA PARA ESTIMAÇÃO DOS RESULTADOS

O objetivo do estudo apresentado nessa monografia é medir a capacidade de uma RNA realizar previsão dos valores e tendências de alta ou de baixa do índice IBOVESPA com base no desempenho de variáveis macroeconômicas. Também se deseja entender quais variáveis macroeconômicas possuem maior peso sobre a estimação dos resultados desse índice e se há correlação entre essas variáveis.

Foram realizadas em torno de 100 simulações de forma empírica, onde se comprovou o informado pela literatura de que os melhores resultados foram obtidos através de redes diretas utilizando o algoritmo *backpropagation*. A implementação computacional das RNAs foi realizada utilizando o software MBP (*multiple backpropagation*).

Foram selecionados 8 modelos para análise, de modo a demonstrar a evolução do estudo, cujos resultados constam no próximo capítulo.

#### 3.1 O modelo de RNA utilizado

Os dados escolhidos como entradas para as RNAs são as séries mensais de variáveis macroeconômicas, para o período de Janeiro de 1998 a Dezembro de 2006. A escolha desse intervalo de datas foi feita de modo que todas as variáveis de interesse estavam disponíveis.

O conjunto total de variáveis de entrada das RNAs tem 1.440 amostras e essas variáveis foram separadas na proporção geralmente proposta pela literatura, tal que 90% desses dados foram separados para teste e 10% para validação de maneira aleatória. Optou-se pelo escalonamento de algumas variáveis de entrada além do uso de variáveis percentuais.

As variáveis de entrada escolhidas foram a Taxa de Juros de Longo Prazo (TJLP), Taxa Selic, Taxa de Câmbio Efetiva Real - INPC e o Índice Nacional de Preços ao Consumidor Amplo (IPCA). Essas variáveis foram escolhidas tendo

em vista o interesse que há de entender como elas podem afetar a previsão do índice IBOVESPA, e entender quais dessas variáveis possuem maior peso para a formação do preço deste índice, que será adotado como variável de saída buscando entender se há relação entre todas essas variáveis.

Todos os dados foram uniformizados de maneira linear se limitando a um máximo de 1, e um mínimo de 0 e também de maneira sigmoide, afim de estabelecer um limite máximo entre os números de 1 e um mínimo de -1. Conforme defendido por muitos autores, após os treinamentos, os dados linearizados ente -1 e 1 se mostram mais eficazes no tocante ao desempenho da rede. Para os modelos aqui propostos, usaremos esses dados que foram uniformizados utilizando a expressão:

$$x^* = \frac{2x - [\max(x) + \min(x)]}{\max(x) - \min(x)} \quad (4.16)$$

No Gráfico 1 constam os valores das variáveis de entrada e a variável de saída esperada já normalizados.

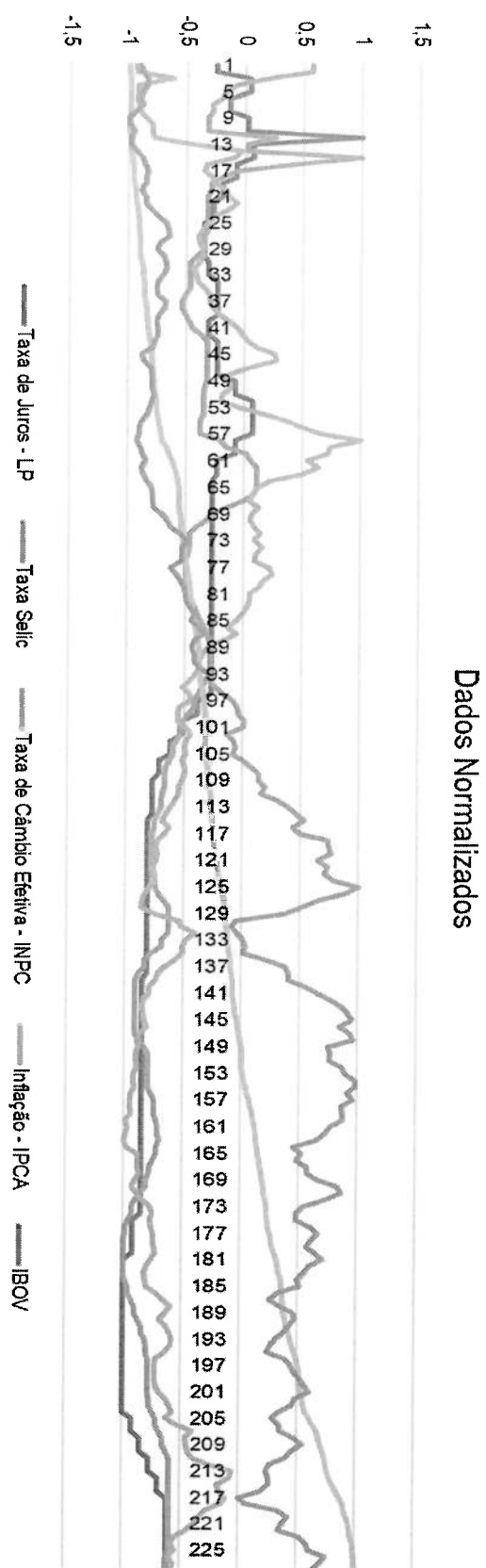


Gráfico 1

### 3.2 Topologia e Arquitetura

A arquitetura escolhida para o treinamento das redes neste trabalho foi uma rede *feedforward* (direta), com o aprendizado supervisionado, pois após todo o apanhado teórico feito para o desenvolvimento deste trabalho, entende-se que esse se apresenta como o melhor modelo com finalidade de previsão.

Optou-se pelo uso de duas camadas ocultas, em fase de teste, se observou que bons resultados eram obtidos através de apenas uma camada ou mais camadas além de duas, mas esses eram ainda melhores quando utilizados duas camadas, tendo em vista que a diferença entre os erros das amostras de teste e treino se apresentam menores, foi adotada a sugestão inicial do software MBP para o número de elementos em cada camada.

Foram utilizadas as funções de ativação tangente hiperbólica e sigmoidal, e efetuada a comparação entre essas em cada modelo. No próximo capítulo são apresentados os desempenhos obtidos.

### 3.3 Algoritmo de Treinamento

O algoritmo utilizado foi o backpropagation (BP). A taxa de aprendizagem utilizada foi de 0,7 e incluído uma taxa de *momentum* igual a 0,7 onde a probabilidade de cruzamento adotada foi de 0,9 e a probabilidade de mutação de 0,01. Foram utilizadas 1.000 iterações em cada treinamento e como critério de parada foi adotado o erro quadrático médio (MSE) menor que 0,01. Todos os pesos iniciais da RNA foram escolhidos aleatoriamente pelo software.

### 3.4 Medida de Performance

Para medida de performance foram usadas as seguintes variáveis:

- Erro absoluto mínimo;
- Erro absoluto máximo;

- MSE – Erro quadrático médio:  $MSE = \frac{\sum_{i=0}^N (d_i - x_i)^2}{N}$  (4.17)

- MAE – Erro absoluto máximo:  $MAE = \frac{1}{N} \sum_{i=1}^N |x_i - d_i|$  (4.18)

- $r$  – Coeficiente de correlação linear:  $r = \frac{\frac{\sum (x_i - \bar{x})(d_i - \bar{d})}{N}}{\sqrt{\frac{\sum (d_i - \bar{d})^2}{N}} \sqrt{\frac{\sum (x_i - \bar{x})^2}{N}}}$  (4.19)

Nas expressões acima,  $d$  é o valor desejado,  $N$  é o número de amostras e  $x$  é a saída rede. Os pesos também foram analisados afim de definir quais variáveis foram mais importantes para a tomada de decisão da RNA.

No software utilizado, as variáveis foram definidas conforme listado abaixo:

Entradas:

1<sup>th</sup> input → Taxa de Juros (Longo Prazo)

2<sup>nd</sup> input → Taxa de Juros (Curto Prazo)

3<sup>rd</sup> input → Taxa de Câmbio

4<sup>th</sup> input → Inflação

5<sup>th</sup> input → IBOV

#### 4. ESTIMAÇÕES E RESULTADOS

Neste capítulo são apresentados os resultados da aplicação de 8 modelos distintos de RNAs na previsão do comportamento do índice IBOVESPA (IBOV).

Inicialmente foi decidido realizar a previsão do valor atual do IBOV observando o desempenho das variáveis macroeconômicas envolvidas. Foram desenvolvidos dois modelos distintos:

**Modelo 1:** O objetivo deste modelo foi a previsão do valor médio de fechamento mensal do índice IBOV levando em consideração as variáveis macroeconômicas no mesmo período. Foram levados em consideração dois conjuntos de dados composto por todas as 1.140 amostras, sendo o primeiro composto 965 amostras, que correspondem ao período de Janeiro de 2006 a Janeiro de 2014, que farão parte do treinamento, enquanto as 175 amostras restantes que correspondem aos períodos de Fevereiro de 2014 a Dezembro de 2016 para serem usadas como teste (validação).

RMS (Erro Quadrático Médio) after the addition of noise to the:								
Noise (Barulho)	Training Data				Testing Data			
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação
0%	0.0473453734	0.0473453734	0.0473453734	0.0473453734	0.1892767891	0.1892767891	0.1892767891	0.1892767891
2%	0.0487362460	0.0474807047	0.0479852869	0.0483365838	0.1892761558	0.1892767873	0.1892775715	0.1892772541
5%	0.0553913652	0.0481842108	0.0512284870	0.0532020874	0.1892727281	0.1892767766	0.1892815636	0.1892796167
10%	0.0695627518	0.0500058497	0.0590951365	0.0641306807	0.1892627029	0.1892767388	0.1892913109	0.1892852541
always set to minimum	0.1310291029	0.0735276056	0.1670333650	0.0730305387	0.1978346497	0.1900913995	0.1832928707	0.1865680112
always set to mean	0.1115662152	0.0585036643	0.1249825051	0.1178797090	0.1936882562	0.1897707407	0.1855084868	0.1910366351
always set to maximum	0.1586404180	0.0962569709	0.2688225164	0.2414929744	0.1886174045	0.1890999387	0.1908759626	0.1896534979

Tabela 1 – Modelo 1

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	0.659648	-447.966	-142.375	312.024	-39.801
2th neuron	-173.984	377.835	116.974	0.809849	3.18

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	377.669	-387.645	-590.638

Tabela 2 – Modelo 1

**Modelo 2:** Este modelo tem o mesmo objetivo e estrutura do modelo 1, porém não foi utilizado todo o conjunto de dados. Foram levados em consideração dois conjuntos de dados compostos por todas as 535 amostras, sendo o primeiro composto 425 amostras que correspondem ao período de Janeiro de 2008 a Janeiro de 2015, que farão parte do treinamento e enquanto as 110 amostras restantes que correspondem aos períodos de Fevereiro de 2015 a Dezembro de 2016 para serem usadas como teste.

RMS (Erro Quadrático Médio) after the addition of noise to the:								
Noise (Barulho)	Training Data				Testing Data			
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação
0%	0.0250190826	0.0250190826	0.0250190826	0.0250190826	0.0415600551	0.0415600551	0.0415600551	0.0415600551
2%	0.0251406263	0.0259176084	0.0273310158	0.0255980446	0.0415590676	0.0415611983	0.0415460631	0.0415634275
5%	0.0257685620	0.0301866454	0.0369573177	0.0284382906	0.0415539037	0.0415676400	0.0414816668	0.0415810881
10%	0.0273665397	0.0392854775	0.0533300280	0.0348664055	0.0415403088	0.0415885641	0.0414121085	0.0416278285
always set to minimum	0.0299429190	0.0913977386	0.1164376316	0.2174205513	0.0416723771	0.0415794329	0.0469302874	0.2793326835
always set to mean	0.0488584555	0.1130868746	0.0853066019	0.1109155753	0.0410515816	0.0427612218	0.0415541277	0.0905557450
always set to maximum	0.0752669842	0.1759631173	0.0717816510	0.1957954228	0.0591837905	0.0474686522	0.0422925228	0.0423800179

Tabela 3 – Modelo 2

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-139.781	0.193096	0.467063	-0,394404	-1,84256
2th neuron	-602.988	204.874	-345.846	-8,27752000	-4.16367

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	103.531	-990.514	290.493

Tabela 4 – Modelo 2

Como variáveis de entrada foram consideradas as variáveis macroeconômicas e como variável de saída, a cotação do IBOV no mesmo período. Nos dois modelos apresentados foi utilizada a função sigmoidal.



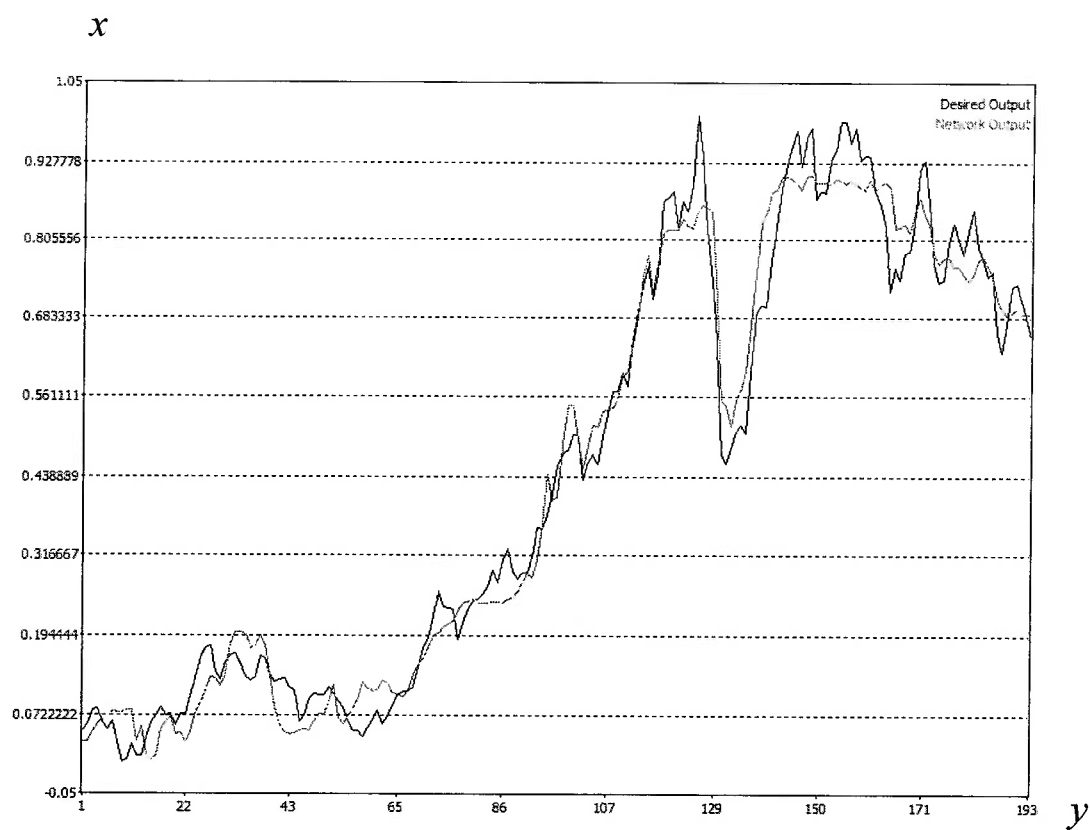


Gráfico 2 – Modelo 1 (Treino)

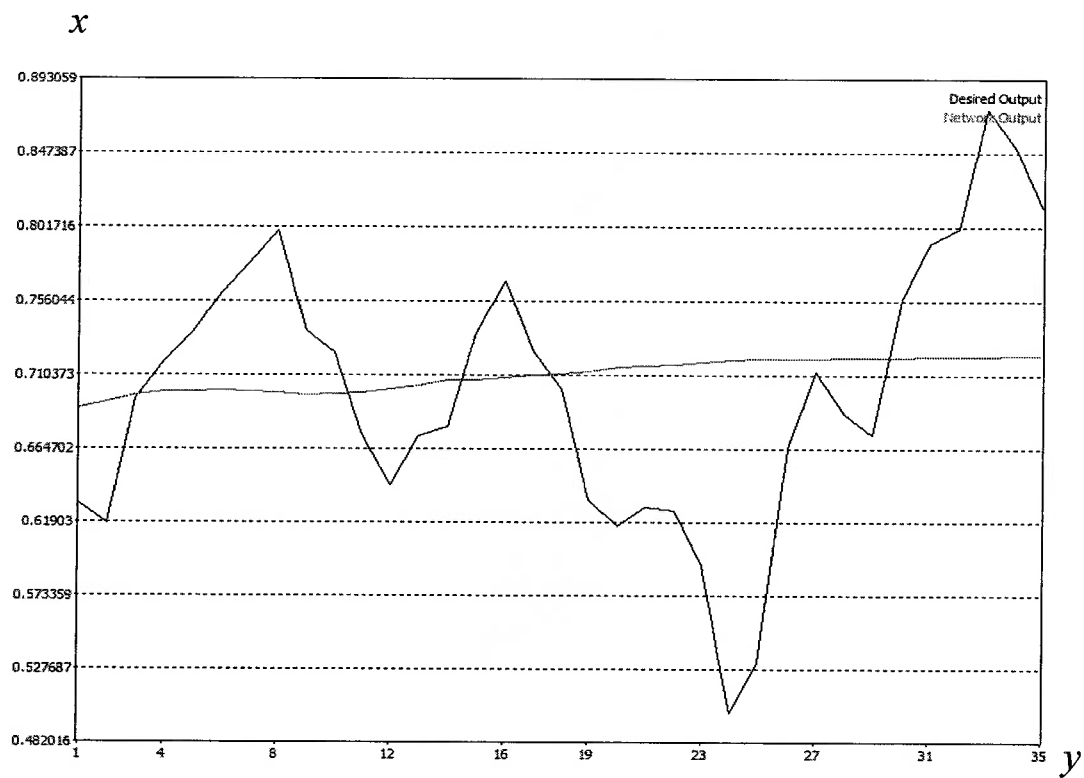


Gráfico 3 – Modelo 1 (Teste)

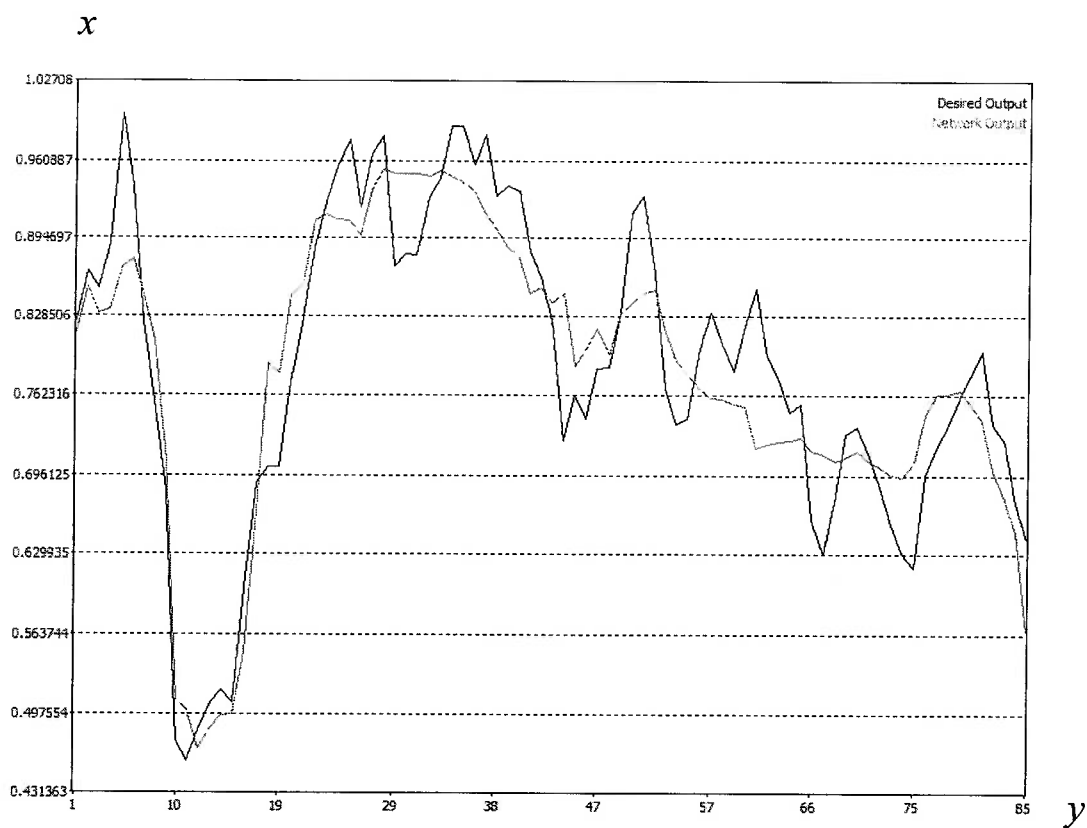


Gráfico 4 – Modelo 2 (Treino)

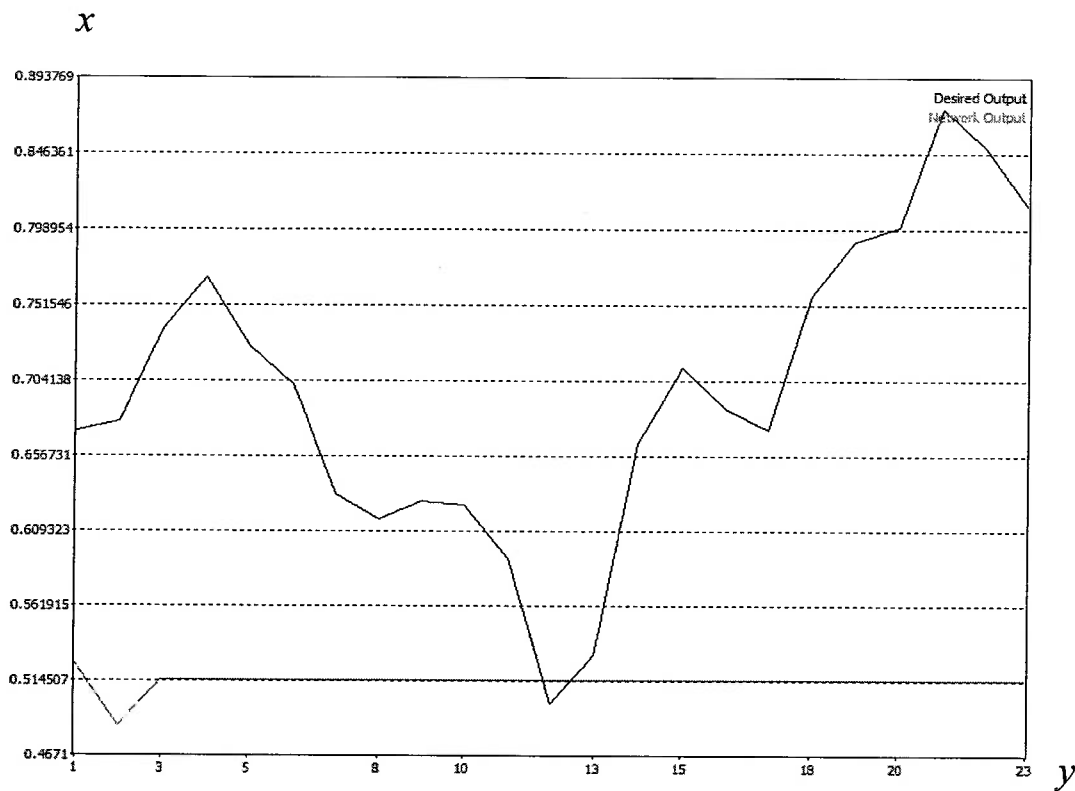


Gráfico 5 – Modelo 2 (Teste)

É possível observar através dos dados e gráficos que o valor do IBOV (*Desired Output*) apresenta elevada taxa de crescimento no início do período coletado. Como o conjunto de treinamento do primeiro modelo é formado pelos primeiros períodos, a RNA só consegue apresentar bons resultados quando o valor da saída desejada é próximo aos valores de treinamento.

No modelo 2 foram desconsiderados os últimos períodos visando corrigir essa oportunidade. O problema foi reduzido e este apresentou menores erros em relação ao modelo anterior (os valores dos erros quadráticos se apresentam menores nesse modelo), porém a rede ainda apresentou uma defasagem em relação ao valor real.

Em ambos os modelos todas as variáveis macroeconômicas são consideradas importantes, com destaque em comum para a taxa de câmbio e inflação. É possível observar essa importância à medida que são adicionados ruídos (*noise*) à rede, sendo que as variações entre os valores de erro encontrados são maiores que as observadas nas demais variáveis. No modelo 1 a Taxa de Juros (Longo Prazo) se mostrou mais significativa que no modelo 2, tendo maior peso a Taxa de Juros (Curto Prazo).

Em seguida, foram desenvolvidas RNAs com objetivos voltados a uma aplicação mais prática: a previsão do IBOV no próximo dia, tendo em vista que o interesse dos investidores tipicamente é prever valores futuros de preços de ações.

Também é possível identificar que as redes conseguiram modelar boa parte dos movimentos de subida e descida das variáveis e ambos os modelos apresentaram um bom índice de correlação linear com destaque para o primeiro modelo.

**Modelo 3:** O objetivo deste modelo foi medir a capacidade de previsão do valor do IBOV no período  $(t+1)$ , onde foram utilizados escalonamentos nas variáveis de entrada e a inclusão do IBOV escalonado também como uma variável de entrada, para fornecer uma memória de curto prazo da rede (últimos três

períodos). O conjunto de dados foi formado por todas as 1.140 amostras, sendo 965 parte do treinamento e as 175 restantes correspondentes às amostras de teste.

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Noise (Barulho)	Training Data					Testing Data				
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.0356709424	0.0356709424	0.0356709424	0.0356709	0.0356709	0.1029569881	0.1029569881	0.1029569881	0.10295698	0.10295698
2%	0.0359307174	0.0356854598	0.0358004452	0.0367510	0.0364194	0.1029580975	0.1029577990	0.1029763728	0.10301837	0.10299272
5%	0.0372671670	0.0357615822	0.0364725041	0.0419696	0.0401189	0.1029639206	0.1029620563	0.1030785160	0.10334415	0.10318161
10%	0.0406319379	0.0359638077	0.0382045176	0.0533331	0.0486160	0.1029794403	0.1029734145	0.1033539292	0.10424033	0.10369540
always set to minimum	0.1818529338	0.0450255649	0.0552000867	0.1255587	0.2469716	0.1106228164	0.1001870329	0.0985219927	0.14318824	0.09971175
always set to mean	0.0803751760	0.0384811827	0.0548357006	0.1158652	0.1296393	0.1069007541	0.1011561258	0.0983915794	0.20496677	0.11300140
always set to maximum	0.0403226951	0.0516072528	0.0927244643	0.1322009	0.1790584	0.1036328460	0.1027569002	0.0987821213	0.10693280	0.19182204

Tabela 5 – Modelo 3

to the 1th hidden layer		from the input layer				
		bias	1th neuron	2th neuron	3th neuron	5th neuron
1th neuron		-233.124	-293.781	0.811064	105.591	-704.709
2th neuron		-125.659	-0.440865	0.371951	181.293	-323.356

to the output layer		from the 1th hidden layer	
		bias	1th neuron
1th neuron		-0.132139	-438.099

Tabela 6 – Modelo 3

**Modelo 4:** Apresenta o mesmo objetivo do modelo 3, porém com a redução do conjunto de dados para 950 sendo, 902 utilizados para treino e 48 para teste.

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Noise (Barulho)	Training Data					Testing Data				
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.0323677540	0.0323677540	0.0323677540	0.03236775	0.0323677	0.1123514512	0.1123514512	0.1123514512	0.1123514	0.1123514512
2%	0.0330552190	0.0323795741	0.0326361748	0.03563165	0.0331590	0.1123514790	0.1123524805	0.1123890832	0.1124360	0.1123878452
5%	0.0364252099	0.0324415508	0.0340105128	0.04883066	0.0370265	0.1123516252	0.1123578858	0.1125887880	0.1128909	0.1125809116
10%	0.0440018779	0.0326061780	0.0374258170	0.07084264	0.0457062	0.1123520154	0.1123723131	0.1131386683	0.1141888	0.1131119719
always set to minimum	0.2350860328	0.0401336582	0.0574172444	0.18630275	0.2403767	0.1114131273	0.1098424932	0.1088047585	0.1620688	0.1107606184
always set to mean	0.1084868670	0.0342572143	0.0666064455	0.12961982	0.1218660	0.1117861293	0.1106387912	0.1086687445	0.2487917	0.1399569264
always set to maximum	0.0407745888	0.0415616153	0.1129212955	0.12522092	0.1701367	0.1120903317	0.1121376894	0.1073374344	0.1250959	0.2237795579

Tabela 7 – Modelo 4

		from the input layer				
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron
1th neuron	-325.663	-369.478	0.433138	0.76175	-875.491	-246.952
2th neuron	107.577	-0.078075	-0.477333	-288.362	432.009	-283.554

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	332.502	-486.846	-34.366

Tabela 8 – Modelo 4

**Modelo 5:** Similar ao modelo anterior, com o uso do mesmo conjunto de treinamento do modelo 3. Entretanto, eliminando o IBOV escalonado como variável de entrada.

RMS (Erro Quadrático Médio) after the addition of noise to the:								
Noise (Barulho)	Training Data				Testing Data			
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação
0%	0.0490708671	0.0490708671	0.0490708671	0.0490708671	0.1063030147	0.1063030147	0.1063030147	0.1063030147
2%	0.0492153887	0.0503980655	0.0493324321	0.0493769180	0.1063286517	0.1064801618	0.1063566781	0.1063698530
5%	0.0499670467	0.0567644380	0.0506990357	0.0509707616	0.1064636613	0.1074296764	0.1066402300	0.1067235752
10%	0.0519146757	0.0702422205	0.0542758057	0.0550950312	0.1068270819	0.1101154687	0.1074111607	0.1076897128
always set to min	0.0985939066	0.1230015319	0.1055165772	0.1263588087	0.0984216164	0.0983241000	0.2314911824	0.2334352138
always set to max	0.0776728827	0.1129903502	0.0845589298	0.0857990450	0.0984359124	0.0983899996	0.1287316092	0.2295897931
always set to mean	0.0580794467	0.2713173328	0.1260745818	0.1313066025	0.0985491663	0.1028912008	0.0996120122	0.1573729580

Tabela 9 – Modelo 5

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-888.536	191.494	502.473	-277.848	-309.774
2th neuron	290.558	-0.500238	0.462297	306.464	290.531

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	292.884	-928.705	-304.812

Tabela 10 – Modelo 5

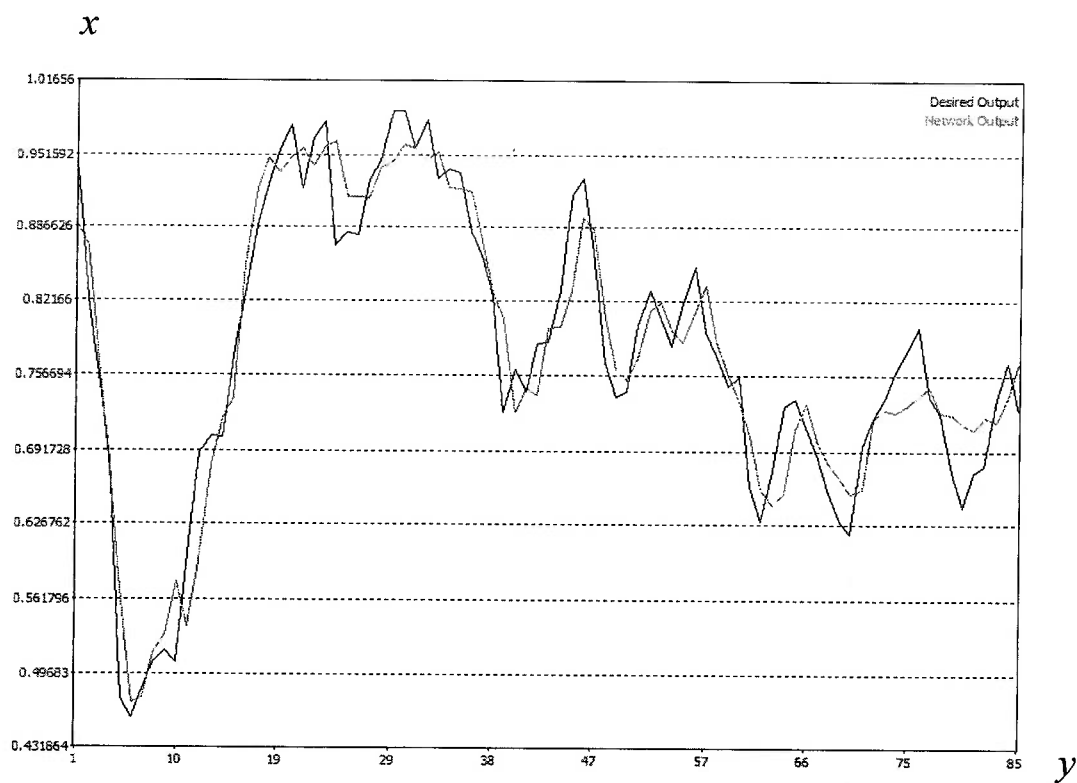


Gráfico 6 – Modelo 3 (Treino)

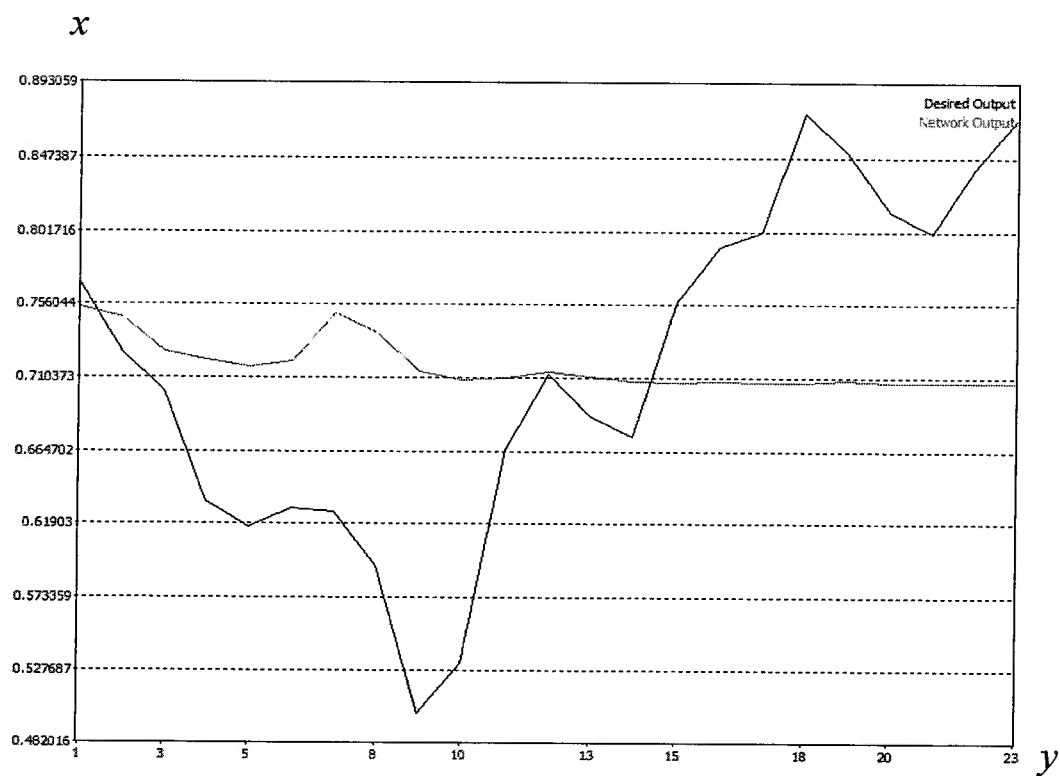


Gráfico 7 – Modelo 3 (Teste)

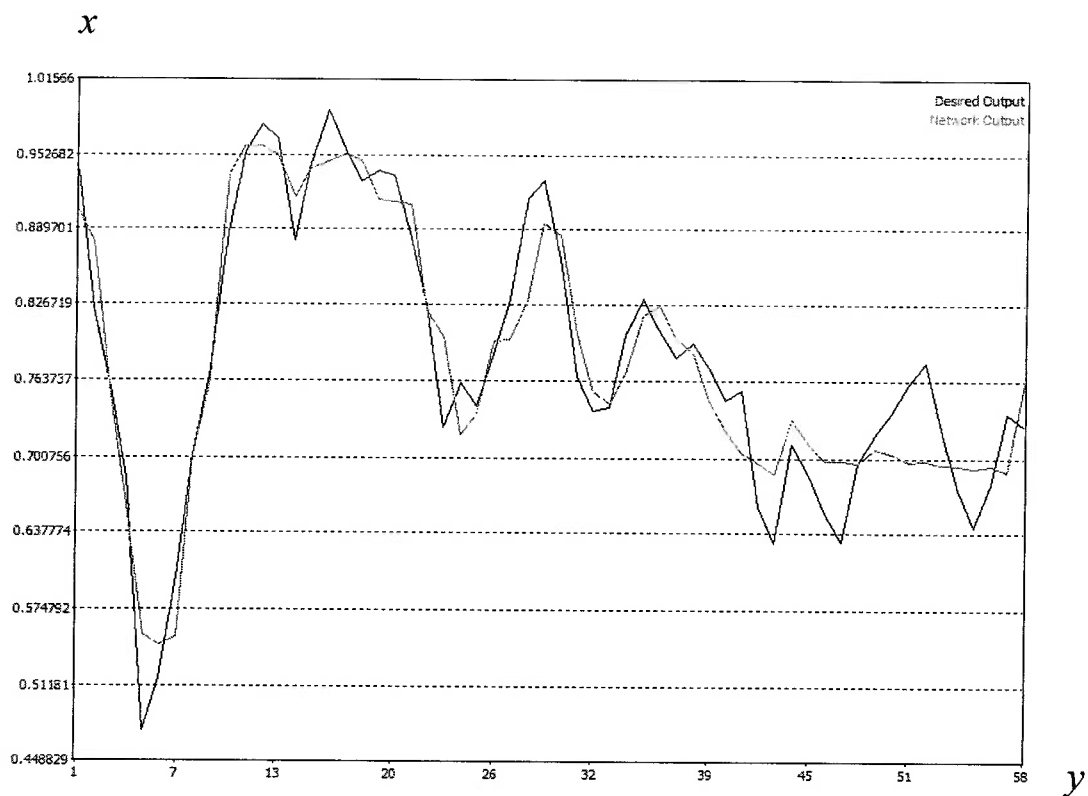


Gráfico 8 – Modelo 4 (Treino)

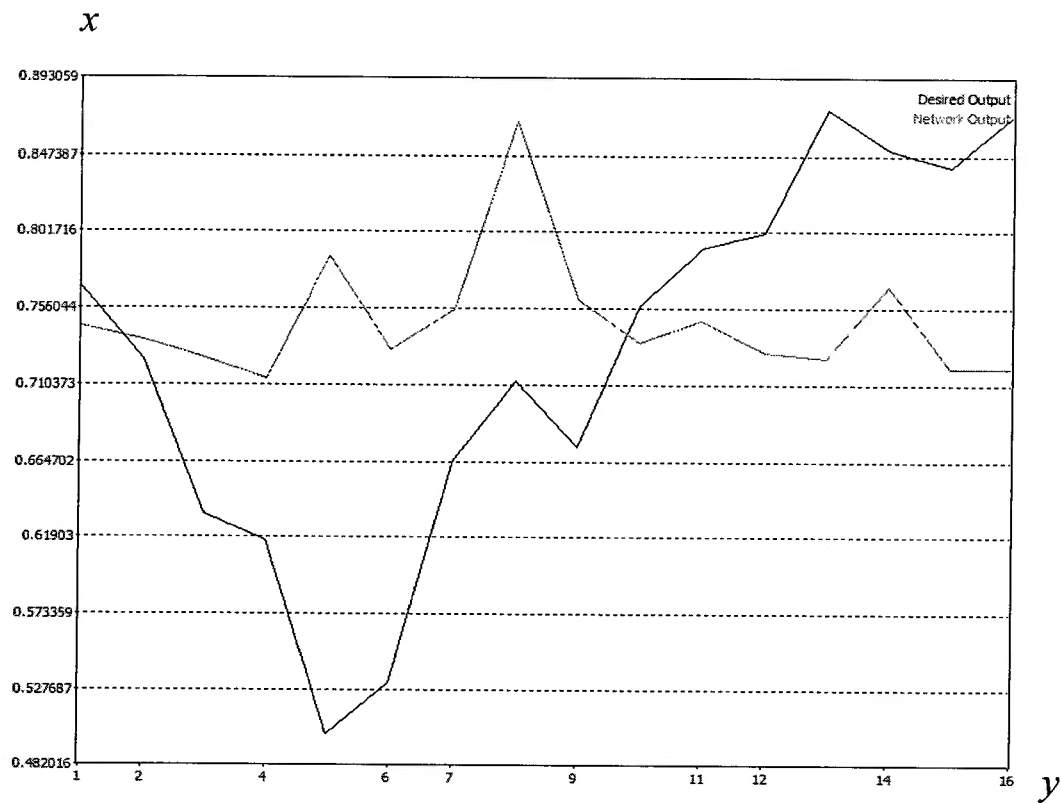


Gráfico 9 – Modelo 4 (Teste)

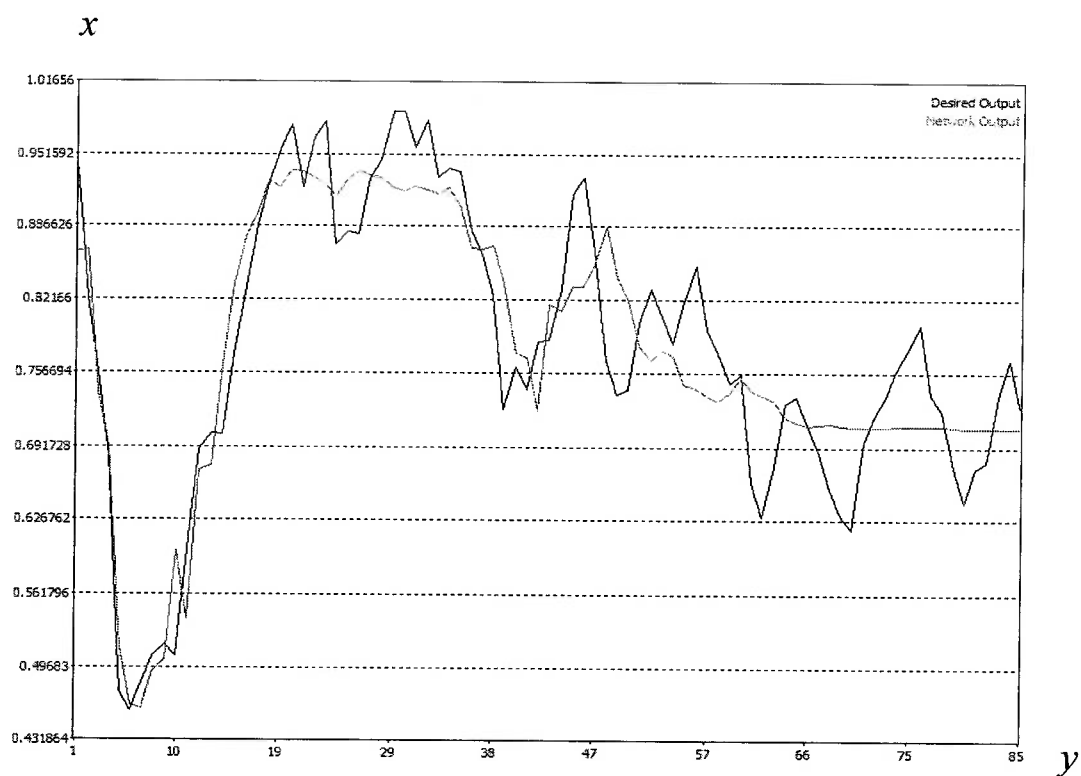


Gráfico 10 – Modelo 5 (Treino)

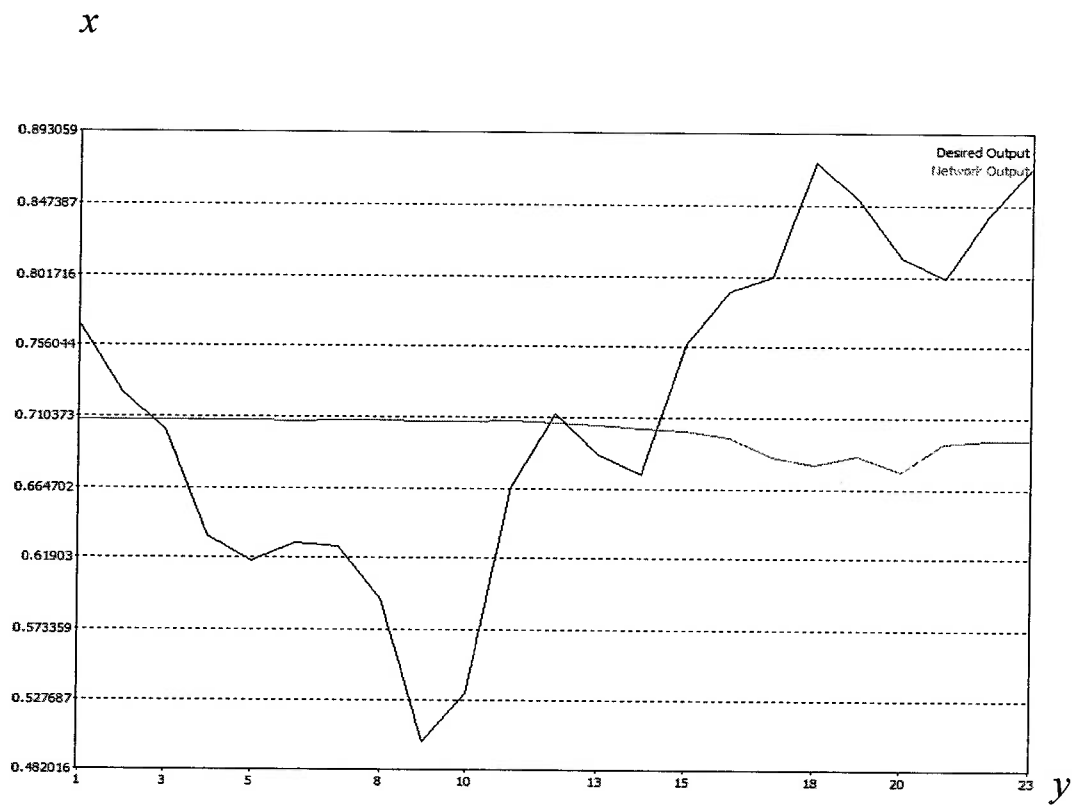




Gráfico 11 – Modelo 5 (Teste)

É possível observar que a introdução do IBOV escalonado ajudou na majoração dos valores da rede, porém este obteve um peso bastante considerável juntamente com a inflação principalmente no modelo 3. Percebeu-se que melhores resultados eram obtidos sem esta variável (comparando o modelo 3 ao modelo 5, onde foram utilizados os mesmos dados onde é possível observar que os valores dos erros se apresentam maiores no modelo 3), a inclusão do IBOV escalonado melhora a percepção de alta e baixa do IBOV, porém os erros se mostram menores com a retirada dessa variável, isto podendo ser observado nos valores de erro máximo e mínimo apresentados.

**Modelo 6:** O objetivo deste modelo foi prever se o índice IBOV irá cair ou subir do período ( $t+1$ ), utilizando como valor desejado o teste lógico 0 = cair e 1 = subir. Foram utilizados os mesmos conjuntos de treinamento e teste do modelo 3.

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Noise (Barulho)	Training Data					Testing Data				
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.0004244982	0.0004244982	0.0004244982	0.0004244982	0.0004244982	0.0007812220	0.0007812220	0.0007812220	0.0007812220	0.0007812220
2%	0.0004245370	0.0004245389	0.0004245441	0.0004245551	0.0004246655	0.0007812293	0.0007812029	0.0007812056	0.0007811961	0.0007812285
5%	0.0004247411	0.0004247524	0.0004247852	0.0004248539	0.0004255441	0.0007812681	0.0007811032	0.0007811198	0.0007810614	0.0007812631
10%	0.0004252857	0.0004253210	0.0004254270	0.0004256489	0.0004278884	0.0007813715	0.0007808374	0.0007808913	0.0007807004	0.0007813559
always set to minimum	0.0004582624	0.0003709631	0.0003555086	0.0003603449	0.0002843390	0.0008500457	0.0006221393	0.0006166011	0.0005237711	0.0007120758
always set to mean	0.0004266275	0.0004235661	0.0004416114	0.0004352002	0.0004049301	0.0008000515	0.0007173747	0.0007077475	0.0006583824	0.0007855714
always set to maximum	0.0004005765	0.0005147696	0.0005626928	0.0005589380	0.0005447324	0.0007817845	0.0007808924	0.0007751473	0.0007563184	0.0008627568

Tabela 11 – Modelo 6

from the input layer						
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron
1th neuron	0.431427	-0.244052	-0.729624	-0.673287	-0.836951	-0.626326
2th neuron	191.295	0.652927	0.374651	0.0923666	0.313591	-0.66252

from the 1th hidden			
to the output layer	bias	1th neuron	2th neuron
1th neuron	443.036	158.214	196.235
2th neuron	380.772	180.756	257.417

Tabela 12 – Modelo 6

**Modelo 7:** Possui a mesma proposta do modelo anterior, apenas com a mudança da função de ativação para linearsig.

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Noise (Barulho)	Training Data					Testing Data				
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.000474926	0.000474926	0.000474926	0.000474926	0.000474926	0.0011347518	0.0011347518	0.0011347518	0.0011347518	0.0011347518
2%	0.0004749314	0.0004749634	0.0004749316	0.0004749385	0.0004749404	0.0011348423	0.0011382024	0.0011352285	0.0011356464	0.0011359898
5%	0.0004749597	0.0004751603	0.0004749611	0.0004750042	0.0004750166	0.0011353173	0.0011563937	0.0011377321	0.0011403471	0.0011425009
10%	0.0004750356	0.0004756892	0.0004750398	0.0004751798	0.0004752198	0.0011365831	0.0012055375	0.0011444194	0.0011529188	0.0011599595
always set to minimum	0.0005357887	0.0004569899	0.0004611493	0.0004407269	0.0004904207	0.0009455209	0.0004628208	0.0006303257	0.0004903003	0.0007331217
always set to mean	0.0004726197	0.0004722008	0.0004759960	0.0004752943	0.0004771821	0.0010181338	0.0005510835	0.0008211767	0.0006033043	0.0011880421
always set to maximum	0.0004260525	0.0005312605	0.0004974074	0.0005269483	0.0004842366	0.0010136462	0.0009998183	0.0012242282	0.0008622809	0.0030464852

Tabela 13 – Modelo 7

to the 1th hidden layer		from the input layer				
		bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron		0.0464861	107.357	0.0177283	-0.212954	-0.645925
2th neuron		256.843	-0.0972837	-0.631838	-0.231137	-0.309326

to the output layer		from the 1th hidden layer	
		bias	1th neuron
1th neuron		332.787	0.189969
2th neuron		323.859	0.184927

Tabela 14 – Modelo 7

**Modelo 8:** Muda o teste lógico e a função de ativação em relação ao modelo 6. O teste lógico passa a ser -1 = cair, 1 = subir. Foram utilizadas funções de transferência sigmóide e lineartanh.

### Sigmoide

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Noise (Barulho)	Training Data					Testing Data				
	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.0004663673	0.0004663673	0.0004663673	0.0004663673	0.0004663673	0.0005678888	0.0005678888	0.0005678888	0.0005678888	0.0005678888
2%	0.0004663846	0.0004664637	0.0004663749	0.0004663843	0.0004663794	0.0005679102	0.0005680075	0.0005678972	0.0005679117	0.0005679051
5%	0.0004664757	0.0004669699	0.0004664147	0.0004664738	0.0004664429	0.0005680224	0.0005686301	0.0005679412	0.0005680318	0.0005679905
10%	0.0004667184	0.0004683193	0.0004665208	0.0004667123	0.0004666122	0.0005683214	0.0005702894	0.0005680585	0.0005683520	0.0005682183
always set to minimum	0.0005394751	0.0003472087	0.0004888194	0.0004193051	0.0005126128	0.0009004889	0.0003228826	0.0006568142	0.0004432578	0.0006230564
always set to mean	0.0004689860	0.0004664679	0.0004680609	0.0004665309	0.0004677711	0.0007720060	0.0004154171	0.0006104801	0.0004857966	0.0005629279
always set to maximum	0.0004117289	0.0006529145	0.0004557037	0.0005288379	0.0004351482	0.0006640707	0.0005638130	0.0005739513	0.0005425364	0.0005177747

Tabela 15 – Modelo 8

		from the input layer				
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron
1th neuron	126.227	0.237203	-0.557036	0.31491	-0.309325	0.31724
2th neuron	-0.212218	0.191707	-0.457702	-0.314847	-0.0091773	-0.144672

		from the 1th hidden	
to the output layer	bias	1th neuron	2th neuron
1th neuron	495.311	190.464	119.468
2th neuron	482.862	225.578	0.884807

Tabela 16 – Modelo 8

### Lineartanh

RMS (Erro Quadrático Médio) after the addition of noise to the:										
Training Data						Testing Data				
Noise (Barulho)	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV	Taxa de Juros LP	Taxa de Juros CP	Taxa de Câmbio	Inflação	IBOV
0%	0.0003891479	0.0003891479	0.0003891479	0.000389147	0.0003891479	0.0003527845	0.0003527845	0.0003527845	0.0003527845	0.0003527845
2%	0.0003891548	0.0003891669	0.0003891672	0.000389163	0.0003891746	0.0003527885	0.0003527866	0.0003527936	0.0003527860	0.0003528008
5%	0.0003891912	0.0003892669	0.0003892687	0.000389246	0.0003893145	0.0003528095	0.0003527975	0.0003528415	0.0003527941	0.0003528867
10%	0.0003892881	0.0003895340	0.0003895396	0.000389466	0.0003896890	0.0003528655	0.0003528266	0.0003529693	0.0003528157	0.0003531165
always set to minim	0.0004138450	0.0004288461	0.0003917727	0.000425683	0.0004502216	0.0004087245	0.0003740326	0.0003386443	0.0003736429	0.0003760541
always set to mean	0.0003899830	0.0003865967	0.0003848620	0.000387538	0.0003919184	0.0003795286	0.0003613865	0.0003408537	0.0003623547	0.0003482968
always set to maxim	0.0003738119	0.0003646852	0.0004007388	0.000367061	0.0003652252	0.0003613735	0.0003529174	0.0003483723	0.0003546343	0.0003368489

Tabela 17 – Modelo 8

		from the input layer				
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron
1th neuron	199.883	0.206253	0.114196	-0.308547	0.0934586	0.418166
2th neuron	204.591	0.0852473	0.361554	0.287205	0.332899	0.138023

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	299.288	22.956	203.688
2th neuron	27.446	261.543	197.552

Tabela 18 – Modelo 8

É possível se observar elevada correlação entre as variáveis e os erros que se apresentam pequenos, sendo que o desempenho da rede aumenta consideravelmente utilizando a função lineartanh e o uso de -1 para indicar

possíveis movimentos de queda, nota-se pequenas diferenças entre os erros apresentados para as variáveis. Para tentativa de previsões, onde não há informação futura, o desempenho da rede deve ser medido através da correlação entre as variáveis, provando ser possível com a combinação dessas chegar a um consenso de volatilidade futura aceitável.

## 5. CONCLUSÕES

Após observar os modelos estudados, é possível concluir que as redes neurais artificiais (RNAs) são ferramentas que podem ser consideradas para uso de investidores na tomada de decisão de compra e venda de ações.

As RNAs estudadas nessa monografia não apresentaram resultados satisfatórios ao se tentar prever as cotações reais de preços, pois as RNAs apresentaram oportunidades de majoração de valores que não foram apresentados em seu treinamento. Porém nos casos de previsão de tendências de dia futuro, ao se utilizar de testes lógicos ( $-1$  = baixa e  $+1$  = alta) os resultados se mostraram excelentes. Ao considerar a topologia de 2 camadas ocultas, juntamente com a função de ativação hiperbólica linear foram obtidos níveis de acerto de 90% com uma confiança de 95%.

Observou-se que duas camadas na arquitetura das redes foram suficientes em todos os modelos estudados nessa monografia. O aumento do número de camadas para além desse valor não melhorou consideravelmente o desempenho da rede, sendo que o tempo para processamento do treinamento se tornou maior.

Os dados históricos do IBOV se mostraram de fundamental importância para as redes na previsão de tendências. Outro ponto interessante é que com a adição das variáveis escalonadas, que representam de alguma forma diferenças entre períodos passados da entrada a ser prevista, ajudou muito no desempenho das RNAs.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

B3. **Portal da B3 - Brasil, Bolsa, Balcão.** Disponível em: [http://www.b3.com.br/pt\\_br/](http://www.b3.com.br/pt_br/). Acesso em: 29 out. 2018.

VELLIDO, A.; LISBOA, P. J. G. e VAUGHAN, J. **Neural networks in business: a survey of applications.** a School of Computing and Mathematical Sciences. Liverpool John Moores University - Byrom St., 1998.

ARAÚJO, A. B. et al. **Aproximação de Funções de Dados Meteorológicos usando Redes Neurais Backpropagation.** Belém, 2012.

BRAGA, A. P.; CARVALHO, A. P. L. F. e LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações.** Rio de Janeiro, 2000.

ALMEIDA, C.; BAUGH, C.; LACEY, C.; FRENK, C.; GRANATO, G.; SILVA, L. e BRESSAN, A. **Modelling the dsty universe i: Introducing the artificial neural network and first applications to luminosity and colour distributions.** Monthly Notices of the Royal Astronomical Society, 2010. 402: 544–564 p.

CARRARA, V. **Análise e Controle de Sistemas Lineares,** 2013.

CARRARA, V. **Redes Neurais Aplicadas ao Controle de Atitude de Satélites com Geometria Variável.** Tese de Doutorado em Mecânica Espacial e Controle, INPE. São José dos Campos, 1997.

CRUZ, A. J. O.; RAPOSO, R. C. T. e MENDES, S. B. T. **Using Previous Knowledge for Stock Market Prediction Based on Fundamental Analysis with Fuzzy-Neural Networks,** 2002. 2211-2216 p.

CYBENKO, G. **Approximation by Superpositions of a Sigmoidal Function.** Math Control Signals Systems, 1989. 2 v, 303-314 p.

FAMA E. F. "Random Walks In Stock Market Prices". **Financial Analysts Journal**, Fall 2008.

F. ROSENBLATT. **The perceptron, a perceiving and recognizing automaton Project**, 1957.

GRÔPPO, G. S. **Causalidade das variáveis macroeconômicas sobre o Ibovespa**. Piracicaba, 2004.

HAYKIN, S. **Redes neurais: princípios e prática**. trad. Paulo Martins Engel. Porto Alegre, ed. 2, 2001.

HAYKIN, S. **Advances in Spectrum Analysis and Array Processing**, Vol. III, Prentice-Hall, 1994.

HORNIK, K., STINCHCOMBE, M. e WHITE, H. **Multi-layer feedforward networks are universal approximators**. **Neural Networks**, v. 2, 1989.

Receita Federal – **Portal da Receita Federal** - Disponível em  
<<http://idg.receita.fazenda.gov.br/orientacao/tributaria/pagamentos-e-parcelamentos/taxa-de-juros-de-longo-prazo-tjlp>> Acesso em 29 out. 2018

IPEA - **Portal do IPEA** - Disponível em  
<<http://www.ipeadata.gov.br/ExibeSerie.aspx?serid=1688012623>> Acesso em 29 out. 2018.

MINSKY, M. L. e PAPERT, S. A., **Perceptrons: an introduction to computational geometry**. The MIT Press, Cambridge, Massachusetts, US. 1969.

ODOM, M. e SHARDA, R. A. **Neural Network for Bankruptcy Prediction**. **International Joint Conference on Neural Networks**, 1993.

RUMELHART, D. E., HINTON, G. E. e WILLIAMS, R. J. **Learning internal representations by error propagation.** In: **Parallel Distributed Processing**, MIT Press, Cambridge, Massachusetts, USA, 1986.

SHANKER, M.; HU, M. Y. e HUNG, M. S. **Effect of data standardization on neural network training.** Elsevier, 1996.

SILVA, I. N.; da SPATTI, D. H. e FLAUZINO, R. A. **Redes Neurais Artificiais: Para engenharia e ciências aplicadas.** São Paulo: Artliber, 2010.

ZHANG, G. Q. e Hu, M. Y. **Neural Network forecasting with artificial neural networks: the state of art.** *International Journal of Forecasting*, 1998.

MCCULLOCH, W. S. e PITTS, W. **A logical calculus of the ideas immanent in nervous activity.** *The bulletin of mathematical biophysics*, 1943.

WEIGEND, A. S.; HUBERMAN, B. e RUMELHART, D. **Generalization by weight-elimination with application to forecasting advances in neural information processing systems**, 1992.

WILSON, R. L. e SHARDA. **Bankruptcy Prediction Using Neural Networks.** *Decision Support Systems*, 1994.