



UM ESTUDO COMPARATIVO PARA PREVISÃO DA COTAÇÃO DE AÇÕES DA BM&FBOVESPA UTILIZANDO REDES NEURAIS ARTIFICIAIS

Trabalho de Conclusão de Curso
Engenharia da Computação

Igor Menezes Marinho de Souza

Orientador: Prof. Dr. Mêuser Jorge Silva Valença



Igor Menezes Marinho de Souza

*UM ESTUDO COMPARATIVO PARA
PREVISÃO DA COTAÇÃO DE AÇÕES DA
BM&FBOVESPA UTILIZANDO REDES
NEURAIS ARTIFICIAIS*

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco

Orientador:

Prof. Dr. Mêuser Jorge Silva Valença

UNIVERSIDADE DE PERNAMBUCO
ESCOLA POLITÉCNICA DE PERNAMBUCO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Recife - PE, Brasil

22 de novembro de 2012

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 13 de 12 de 2012, às 9:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente IGOR MENEZES MARINHO DE SOUZA, orientado pelo professor Mêuser Jorge Silva Valença, sob título Um estudo comparativo para previsão da cotação de ações da BM&FBOVESPA utilizando redes neurais artificiais, a banca composta pelos professores:

Sérgio Galdino

Mêuser Jorge Silva Valença

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,0 (*nove*)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

Sérgio Galdino

SÉRGIO GALDINO

MÊUSER JORGE SILVA VALENÇA

Mêuser Valença

(DEDICATÓRIA) A meus pais.

Agradecimentos

A todos que colaboraram direta ou indiretamente com esse trabalho.

Resumo

Esse trabalho teve como objetivo realizar um estudo comparativo entre previsões realizadas por redes neurais. Preços do mercado de ações foram previstos por redes neurais MLP, Elman e Jordan. Um estudo estatístico foi realizado para descobrir a rede que apresenta a previsão mais acurada.

Palavras-chave: Bolsa de valores, redes neurais, MLP, Elman, Jordan.

Abstract

This work had as objective realize a comparative study among forecast realized by neural networks. Stockes market prices were predicted by MLP, Elman and Jordan neural networks. A statistical study was made to discovery who made the more accurate forecast.

Keywords: stock market, Neural Network, Elman, Jordan, MLP.

Sumário

Lista de Figuras	p. x
Lista de Tabelas	p. xi
Lista de Abreviaturas e Siglas	p. xi
1 Introdução	p. 13
1.1 Problema	p. 13
1.2 Objetivo	p. 14
1.2.1 Objetivo geral	p. 14
1.2.2 Objetivos específicos	p. 14
1.3 Estrutura da Monografia	p. 15
2 Revisão bibliográfica	p. 16
2.1 Análise de investimento	p. 16
2.1.1 Análise fundamentalista	p. 16
2.1.2 Análise técnica	p. 17
2.1.2.1 Princípios da análise técnica	p. 17
2.1.2.2 Análise técnica x análise fundamentalista	p. 18
2.2 Redes Neurais	p. 19
2.2.1 Arquiteturas	p. 20
2.2.1.1 <i>Perceptron</i> e ADALINE	p. 20
2.2.1.2 MLP	p. 21

2.2.1.3	Elman	p. 22
2.2.1.4	Jordan	p. 24
2.2.2	Treinamento	p. 24
2.2.2.1	Algoritmos de treinamento	p. 25
2.2.2.2	Critérios de parada	p. 26
2.3	Tratamento dos dados	p. 28
2.3.1	Normalização	p. 29
2.4	Encog	p. 30
2.5	Trabalhos relacionados	p. 30
2.5.1	Artigos	p. 30
2.5.1.1	<i>Application Study of BP Neural Network on Stock Market Prediction</i>	p. 30
2.5.1.2	<i>Modeling the Behavior of the S&P 500 Index: A Neud Network Appmach</i>	p. 31
2.5.1.3	<i>From an Artificial Neural Network to a Stock Market Day-Trading System: A Case Study on the BM&FBOVESPA</i>	p. 31
2.5.1.4	<i>Artificial Higher Order Neural Networks for Economics and Business</i>	p. 31
3	Metodologia	p. 32
3.1	Processamento dos dados	p. 32
3.1.1	Normalização e tratamento da tendência na prática	p. 32
3.1.2	Os conjuntos de dados	p. 33
3.1.3	Desnormalização	p. 34
3.1.4	Cálculo do erro	p. 34
3.2	Implementação e execução dos treinamentos e testes	p. 35
3.2.1	RNA com Encog	p. 35
3.2.1.1	Treinamento e critério de parada	p. 36

3.3	Experimento	p. 38
3.4	Resumo: parâmetros utilizados	p. 40
4	Análise dos resultados	p. 41
4.1	Teste de normalidade	p. 41
4.2	Teste comparativo	p. 43
4.2.1	Teste de igualdade	p. 43
4.2.2	Comparação das amostras	p. 45
4.3	Previsões	p. 47
5	Considerações Finais	p. 51
5.1	Sugestões para trabalhos futuros	p. 52
	Referências	p. 54

Lista de Figuras

1	Cotação diária do ativo VALE5 entre janeiro de 2007 até outubro de 2012.	p. 18
2	Modelo de uma RNA clássica.	p. 20
3	Ilustração de uma RNA MLP.	p. 21
4	Ilustração de uma RNA Elman.	p. 22
5	Ilustração de uma RNA Jordan.	p. 23
6	Cálculo realizado por um neurônio.	p. 24
7	Evolução do erro durante o treinamento da RNA.	p. 27
8	Passos do critério de parada de validação cruzada.	p. 29
9	Fluxograma do treinamento de uma RNA comparando o erro do conjunto de validação cruzada.	p. 38
10	Valores previstos para a cotação do ativo VALE5 para um dia a frente.	p. 47
11	Comparação da previsão para o primeiro dia a frente com o valor ocorrido para VALE5.	p. 48
12	Valores previstos para a cotação do ativo VALE5 para o quinto dia a frente.	p. 49
13	Comparação da previsão para o quinto dia a frente com o valor ocorrido para VALE5.	p. 50

Lista de Tabelas

1	Dados dos ativos utilizados nesse projeto.	p. 32
2	EMA da previsão um dia a frente das RNA para a ação PETR4.	p. 39
3	Valores dos parâmetros utilizados na simulação.	p. 40
4	Valores do coeficiente W do teste de Shapiro-Wilk da previsão para o 1º dia futuro.	p. 42
5	Valores do coeficiente W do teste de Shapiro-Wilk da previsão para o 5º dia futuro.	p. 42
6	Teste de igualdade para o EMA da previsão do primeiro dia a frente.	p. 44
7	Teste de igualdade para o EMA da previsão do quinto dia a frente.	p. 44
8	Valores teste de Wilcoxon para a previsão do primeiro dia a frente.	p. 45
9	Valores teste de Wilcoxon para a previsão do quinto dia a frente.	p. 46
10	EMA da previsão um dia a frente das RNA para a ação GGBR4.	p. 46
11	Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente.	p. 48
12	Dados estatísticos das 30 amostras do EMA cometido na previsão do quinto dia a frente.	p. 49
13	Resumo do teste comparativo.	p. 51
14	Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente para o ativo GGBR4.	p. 51
15	Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente para o ativo OGXP3.	p. 52

Lista de Abreviaturas e Siglas

RNA	<i>Redes neurais artificiais</i>
ADALINE	<i>Adaptative Linear Neuron</i>
MLP	<i>Multilayer Perceptron</i>
EMQ	<i>Erro Médio quadrático</i>
EMA	<i>Erro Médio Absoluto</i>
RPROP	<i>Algoritmo de Treinamento Resilient Propagation</i>

1 *Introdução*

1.1 Problema

Previsão pode ser considerada como uma expectativa que um evento aconteça, e expectativa indica possibilidade e não certeza. Mas apesar dela não ser algo certo, dependendo do seu nível de maturidade pode indicar, ou alertar, para cenários inesperados o que possibilita gerar um alerta para o estudo da provável situação não prevista e como agir diante dela. Um dos cenários mais interessantes para realizar previsões é a bolsa de valores.

Diante disso, algumas previsões são bastante familiares, tais como: previsão do tempo; previsão do PIB; Previsão da taxa de inflação. E outros são menos, como por exemplo: previsão da vazão de um rio; previsão de crescimento de um nicho de mercado.

Esses tipos de previsões comprovam o interesse existente, seja da parte do governo, de empresas ou do cidadão comum. O tipo de previsão que é de interesse desse trabalho ocorre em um dos cenários mais interessantes: a bolsa de valores.

A bolsa de valores é o lugar onde ações de companhias são compradas e vendidas. Ações são títulos nominativos que representam uma fração do capital social de uma empresa(1). A principal bolsa de valores do Brasil, localizada na cidade de São Paulo, é a BM&FBOVESPA, onde são negociadas, além de ações, derivativos de ações, títulos de renda fixa, títulos públicos federais, derivativos financeiros, moedas à vista e commodities agropecuárias(2). Seu volume financeiro total apenas no mês de agosto de 2012 foi de R\$ 126 bilhões, com as ações da mineradora Vale movimentando R\$ 14,74 bilhões e as da Petrobrás R\$ 12 bilhões (3).

A grande movimentação financeira da bolsa de valores atrai bancos sólidos, empresas, corretoras, investidores profissionais e amadores para atuar nesse mercado. Portanto, uma boa estratégia seguida de disciplina é fundamental para vencer nesse mercado. Desta forma, uma boa ferramenta para auxiliar na elaboração de estratégias ou tomada de decisões pode ser um fator decisivo para o sucesso.

O modo como as ações são precificadas é o que torna a bolsa de valores um local interessante para o estudo da evolução monetária desses ativos. A dinâmica do preço de uma ação ocorre de acordo com a demanda de oferta/procura dos investidores pelo ativo, o que torna o seu preço, de certa forma, imprevisível. Essa imprevisibilidade dos preços desperta o interesse pelo estudo de modelos que possam, dentro de certos limites, oferecer uma previsão aproximada do valor desses títulos. Dentre estes modelos(4–8) encontram-se redes neurais artificiais(RNA).

RNA, como outras técnicas de inteligência artificial, possui a capacidade de se adaptar para chegar ao seu objetivo que no caso desse trabalho é prever o valor da cotação da ação alguns dias à frente. Ao contrário de complexos modelos matemáticos, a RNA facilita o processo, pois ela funciona basicamente como uma caixa preta: oferecemos as entradas necessárias para realização da previsão e obtemos como resposta o provável valor da cotação futura. Isso ocorre porque durante sua fase de aprendizado ela deduz relações entre os valores de entrada e os valores esperados como resposta.

Uma vez escolhida a técnica RNA, é necessário escolher a arquitetura ideal para resolver o problema. A princípio redes neurais como Perceptron(9) e ADALINE(Adaptive Linear Neuron) (10) são logo excluídas por resolverem apenas problemas linearmente separáveis(11). Então, por decisão de projeto, foram escolhidas as arquiteturas *Multilayer Perceptron*(MLP), Jordan(8, 12) e Elman(8, 12), as duas últimas sendo redes com realimentação.

1.2 Objetivo

1.2.1 Objetivo geral

Esse trabalho tem como objetivo geral realizar um estudo comparativo entre o desempenho dos modelos baseados em redes neurais do tipo MLP, Elamn e Jordan na previsão de ações da bolsa de valores.

1.2.2 Objetivos específicos

1. Implementar redes neurais do tipo MLP, Elman e Jordan.

Para alcançar tal objetivo foi desenvolvido um programa utilizando a linguagem de programação java e o *framework* de RNA Encog(13).

2. Realizar simulações com as arquiteturas neurais implementadas e avaliar o desem-

penho destas redes implementadas.

Outro artefato gerado é o resultado de todas as simulações realizadas pelo programa. Esse artefato é um arquivo de texto que contem todos os dados necessários para realização do estudo comparativo entre as arquiteturas de RNA.

No final desse trabalho é apresentada a rede que apresenta o melhor desempenho na resolução do problema já exposto.

1.3 Estrutura da Monografia

Capítulo 1 : Nesse primeiro capítulo foi apresentado uma visão superficial da ambiente onde o desenvolvimento desse trabalho é baseado.

Capítulo 2 : Nesse capítulo são apresentados os conceitos básicos e fundamentais para o entendimento desse trabalho, juntamente com indicações bibliográficas para os que tiverem interesse em aprofundar-se no assunto.

Capítulo 3 : Aqui é exposta a metodologia utilizada no trabalho, com as justificativas para as escolhas tomadas. Também é mostrado o passo a passo do desenvolvimento da simulação e como foram gerados os dados para o estudo comparativo.

Capítulo 4 : Nesse capítulo é realizado o estudo comparativo embasado pelos dados gerado pelo programa do capítulo anterior. Os dados são analisados e comentados

Capítulo 5 : Finalmente é exposta a conclusão do trabalho e indicados possíveis trabalhos futuros.

2 Revisão bibliográfica

Esse capítulo tem como objetivo apresentar, de maneira superficial, alguns conceitos básicos relacionados ao entendimento desse trabalho. Uma abordagem mais detalhada sobre cada tópico desse capítulo poderá ser encontrada nas referências.

2.1 Análise de investimento

A análise de investimentos para operações em mercados de bolsa de valores pode ser agrupada em correntes de pensamento que seguem duas escolas, a escola fundamentalista e a escola técnica(1). A primeira está relacionada à análise empresa e da economia, enquanto a segunda é relacionada com a análise do comportamento dos preços das ações.

2.1.1 Análise fundamentalista

A análise fundamentalista foca em avaliar o potencial da empresa. Essa avaliação leva em consideração fatores como o nicho de mercado da empresa, o seu patrimônio, taxa de inflação, taxa básica de juros, cotação do câmbio, os balanços da empresa, a qualificação profissional dos colaboradores, qualidade da administração, competitividade da empresa, etc...

Levando em conta esses fatores, ou boa parte deles, a análise fundamentalista avalia até que ponto uma variação de cenário pode afetar uma empresa e como a empresa pode suportar, ou superar, cenários adversos. Desse modo o preço da ação é avaliado para descobrir se a ação está sobrevalorizada ou subvalorizada para o cenário esperado.

Em resumo, a escola fundamentalista verifica o potencial de uma ação avaliando o potencial da empresa e do mercado que ela atua para achar ações interessantes do ponto de vista da valorização e distribuição de dividendos(parcela do lucro apurado pela empresa, que é distribuída aos acionistas por ocasião do encerramento do exercício social(1)). É uma análise para um investimento de longo prazo.Pelos dados expostos no primeiro parágrafo

dessa seção fica evidente a complexidade da análise fundamentalista em realizar uma análise desse tipo. É um tipo de análise que leva tempo e deve ser feita com cautela por um, ou vários, especialistas na área de atuação da empresa.

2.1.2 **Análise técnica**

Como exposto em (14), análise técnica é o estudo dos preços, volumes e contratos em aberto no mercado, principalmente através do uso de gráficos, visando prever tendências futuras de preços. Em outras palavras, é o estudo da dinâmica do mercado através da análise dos sinais que o próprio mercado emite(15) através da precificação das ações.

Para o analista técnico, mesmo que alguém conheça todos os fatores fundamentais que afetam o preço de um ativo, tais como catástrofes ambientais, greves, decisões políticas, fatores de demanda, ele ainda assim não terá todos os dados necessários para compreender a formação dos preços. Isso porque, segundo os analistas técnicos, o que afeta os preços é a maneira como os participantes do mercado reagem a esses e outros fatores(1).

2.1.2.1 **Princípios da análise técnica**

Charles Dow e Edward Jones fundaram a *Dow Jones & Company* em 1882 que publicou uma série de artigos no *Wall Street Journal*(14). Dow é considerado o criador dos índices do mercado de ações(1) e os artigos que ele escreveu serviram de base para que estudiosos do mercado utilizassem suas ideias para desenvolver os princípios da análise técnica, onde alguns são descritos abaixo(1, 14).

1. **O mercado desconta tudo:** A análise técnica prega que qualquer coisa que possa influenciar no preço de um ativo é imediatamente refletida no preço desse ativo. Desse modo, não preocupa-se em responder "por quê" os preços se movem, mas sim "como" ou "para onde".
2. **Os preços se movem em tendências:** Apesar da oscilação do preço das ações, ao longo do tempo, eles tendem a seguir uma direção. Isto é, uma tendência de alta ou de baixa. Desse modo, o analista técnico deve primeiro descobrir a tendência para, assim, assumir uma posição no mercado.

Na Figura 1 podemos observar várias tendências de alta e de baixa. Entre maio e outubro de 2008 vemos uma tendência de baixa, esse foi o período da crise mundial. Já entre outubro de 2008 até março de 2010 temos uma tendência de alta, mas

dentro dessa tendência principal de alta houveram outras tendências menores baixa como entre junho e agosto de 2009.

3. **A história se repete:** O que esse princípio quer dizer é que os preços se movem em ciclos, mas esses ciclos apesar de não serem idênticos guardam certas semelhanças, entre si. Isso fica evidente no estudo dos padrões como triângulos(14), bandeiras(14), engolfo de alta(15), *gaps*(15), *harami*(15).

Um outro exemplo de padrão é o conceito de suporte(1) e resistência(1). Se é traçada uma reta no valor de aproximadamente R\$33,00 na Figura 1, veremos que em agosto de 2007 o preço teve uma certa dificuldade para passar esse valor, em outras palavras, sofreu uma resistência. Já em janeiro de 2008, junho de 2010 e agosto de 2011, a reta serviu como um suporte para segurar o preço da ação. Já em outros momentos como agosto de 2008 e agosto de 2009 o suporte e resistência, respectivamente, praticamente não foram sentidos. O que faz com que esse princípio seja melhor exposto como: a história tende a se repetir.

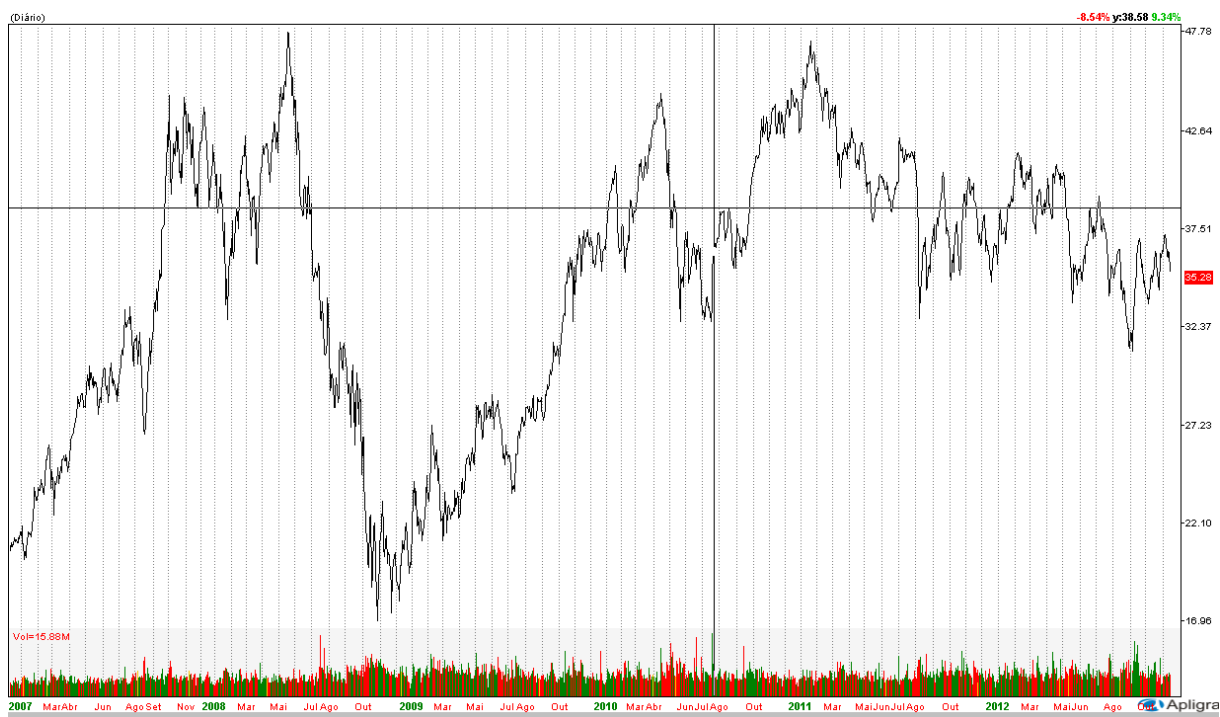


Figura 1: Cotação diária do ativo VALE5 entre janeiro de 2007 até outubro de 2012.

2.1.2.2 Análise técnica x análise fundamentalista

Como foi dito, a análise técnica pretende responder ao "como" ou "para onde", enquanto a análise fundamentalista tenta responder o "porque". Isso faz com que uma

análise fundamentalista para investimentos de curto prazo não seja prática, já que os fatores que ela analisa geralmente fazem efeito depois de certo tempo. O ideal é que ela seja para meses ou anos. Já o "para onde" da análise técnica é ideal para análises de períodos mais curtos tais como: minutos, horas, dias, semanas ou poucos meses. Já que, basicamente, ela tenta descobrir a inércia da movimentação dos preços.

2.2 Redes Neurais

As redes neurais artificiais (RNA) são modelos que tem como metáfora o funcionamento do cérebro com suas redes neurais biológicas(11). São formadas por várias unidades simples de processamentos interconectados chamados de neurônios(16). Nas redes utilizadas nesse trabalho os neurônios são organizados em camadas. Onde, os neurônios de uma camada estão ligados aos neurônios de outra camada que não a sua.

A Figura 2 apresenta um exemplo de RNA clássica, com os neurônios X_t fazendo parte da camada de entrada e os neurônios y_i pertencendo a camada de saída. Cada neurônio da camada de entrada está conectado aos neurônios da camada seguinte, e essas conexões estão associadas a pesos representados por W_{it} . Onde, t é um número natural que representa a quantidade de entradas variando de 0 até n e i pode assumir os valores 1 ou 2, que representam o neurônio de destino.

Os neurônios da camada de entrada representam os estímulos de entrada, enquanto que os da camada de saída representam a resposta da rede a esse estímulo. As RNA mais utilizadas geralmente apresentam pelo menos uma camada intermediária, como as redes mostradas nas Figuras 3, 4 e 5.

No geral o neurônio y possui uma função de ativação f que é utilizada para calcular a resposta do neurônio ao estímulo. Onde o estímulo é soma de todos os sinais que chegam ao neurônio multiplicado pelo respectivo peso associado. Matematicamente temos que a resposta do neurônio y_i é dada pela seguinte fórmula : $Y_i = f(\sum_{t=0}^n w_{it}x_t)$. A mudança da função de ativação, a adição de mais uma camada de neurônios, ou uma mudança nas ligações entre os neurônios pode indicar uma arquitetura diferente de RNA.

Um exemplo para o cálculo da resposta de um neurônio pode ser visto na Figura 6. O valor da resposta Y é calculado por $Y = f(0, 3)$, onde f é função de ativação. Caso f seja a função degrau, o valor de Y será 1. Se a função de ativação for a função linear, o valor de Y será 0,3.

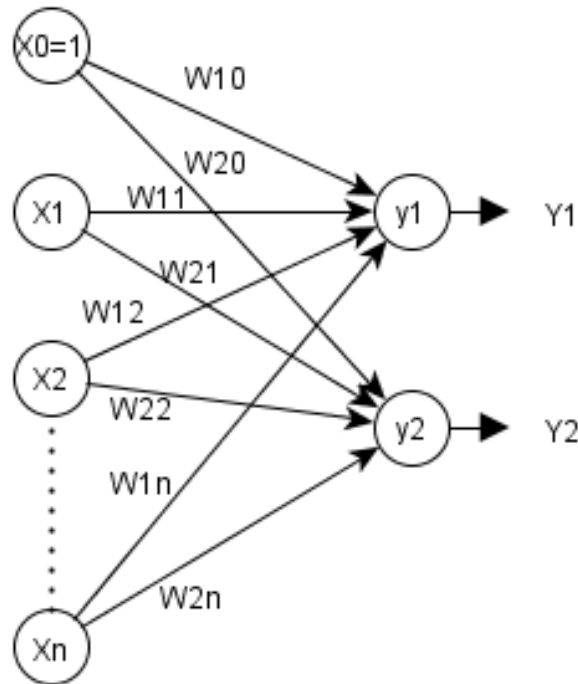


Figura 2: Modelo de uma RNA clássica.

As RNA podem ser utilizadas na resolução de problemas que não podem ser expressos em uma série de passos, como reconhecimento de padrões, classificação, mineração de dados e previsão de séries temporais(17). Elas são excelentes em reconhecimento de padrões e aproximação de funções matemáticas(7). Mas para tal, ela precisa ser treinada por algum algoritmo de treinamento, onde durante a fase de treinamento ela vai ajustar os pesos para chegar ao valor desejado. No geral, no fim do treinamento a RNA funciona como uma caixa preta que recebe uma entrada, processa ela e entrega uma resposta.

2.2.1 Arquiteturas

Esta seção tem como objetivo descrever as arquiteturas de RNA utilizadas nesse trabalho e as que serviram de base para elas.

2.2.1.1 *Perceptron* e ADALINE

Perceptron é a arquitetura mais simples de rede neural. Pode ter forma mostrada na Figura 2 e o que a caracteriza é ter a função degrau como função de ativação. Já a rede ADALINE se diferencia do *Perceptron*, pela possibilidade de utilizar uma função de ativação diferente da função degrau, desde que essa função seja contínua.

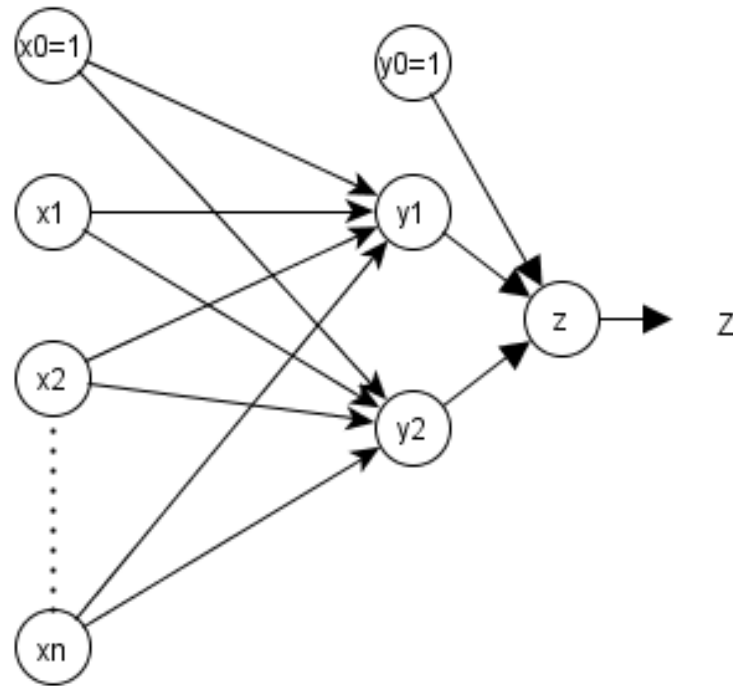


Figura 3: Ilustração de uma RNA MLP.

Outra diferença entre as redes é o algoritmo de treinamento. A RNA ADALINE utiliza a regra delta(11) para o treinamento que é um caso mais geral que o algoritmo de treinamento do *Perceptron*. Uma limitação de ambas as redes, é a incapacidade de resolver problemas não linearmente separáveis(12).

2.2.1.2 MLP

A arquitetura MLP, também conhecida como *Multi-Layer Perceptron(MLP)*, é uma das mais populares arquiteturas de RNA. Essa rede é do tipo *feedforward*, isto é, os sinais se propagam em um único sentido, ou seja, da camada de entrada para a camada de saída. Essa arquitetura apresenta pelo menos três camadas: camada de entrada, camada escondida (uma ou mais) e a camada de saída. Na Figura 3 vemos uma ilustração genérica de uma rede MLP, onde os neurônios que começam com x , y e z fazem parte da camada de entrada, escondida e de saída, respectivamente.

Na camada de entrada cada neurônio representa uma variável que é dada como entrada para o problema. Os dados de entrada estão ligados aos neurônios da camada escondida através de arestas que possuem um peso associado. A soma dos produtos de cada entradas pelo seu respectivo peso da aresta ligada ao neurônio é aplicada a uma função de ativação e esse valor é a resposta do neurônio as entradas recebidas. O valor calculado por cada

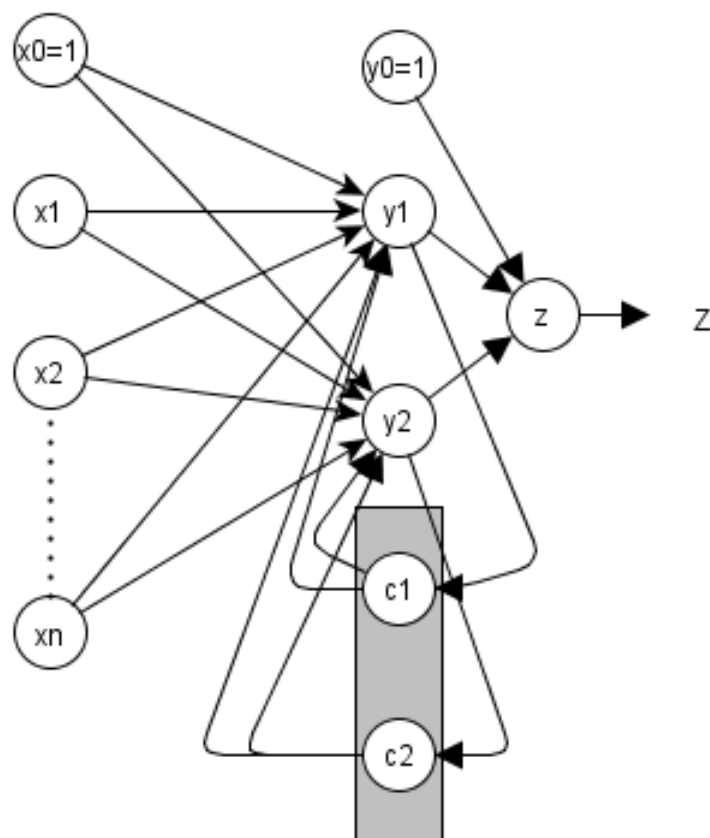


Figura 4: Ilustração de uma RNA Elman.

neurônio da camada escondida servirá como entrada para os neurônios da camada de saída, que calcula o valor da sua saída de forma análoga aos neurônios da camada escondida, sendo que tem como entrada os dados de saída dessa camada em vez dos da camada de entrada.

A MLP não tem a limitação de resolver apenas problemas linearmente separáveis. E essa habilidade deve-se a adição da camada escondida por isso, são excelentes em reconhecimento de padrões, classificação, previsão e aproximação de funções matemáticas.

2.2.1.3 Elman

Em uma rede MLP os sinais se propagam apenas em uma direção mas, existem redes em que a propagação do sinal também ocorre no sentido contrário com o objetivo de realimentar a rede. Essas redes são conhecidas como rede neurais recorrentes.

A RNA Elman é uma rede recorrente que possui uma camada de contexto, como mostrado na Figura 4. A camada de contexto da rede Elman tem a função de guardar o

último valor calculado pela camada escondida, para na próxima iteração esse valor servir como entrada da camada escondida. Dessa forma, a quantidade de neurônios na camada de contexto é idêntica a quantidade de neurônios da camada escondida. Em resumo, além dos dados da camada de entrada e seus respectivos pesos, a camada de escondida também terá como entrada o resultado calculado por ela na execução anterior ligado por pesos.

Uma das diferenças para as redes *feedforward*, é que nas redes recorrentes a ordem em que os dados são apresentados importa, não só no treinamento. A saída de uma rede *feedforward* é calculada apenas em função da suas entradas, enquanto que para as redes recorrentes é calculada em função das entradas e do estado atual, isto é, considerando também as entradas anteriores. Ou seja, a rede *feedforward* está para um circuito combinacional, assim como uma rede recorrente está para um circuito sequencial.

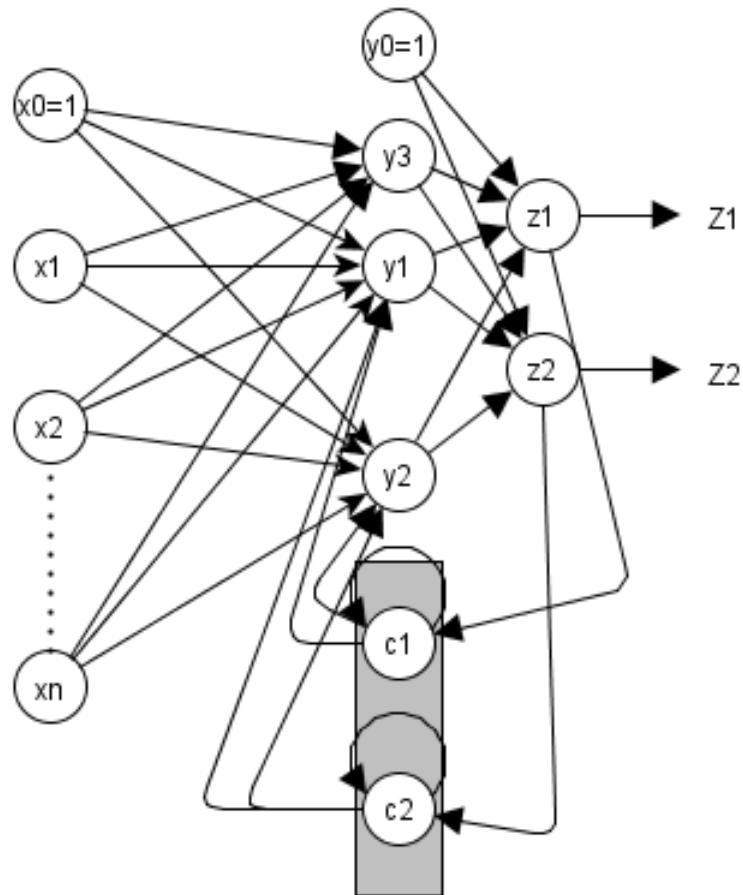


Figura 5: Ilustração de uma RNA Jordan.

2.2.1.4 Jordan

Assim como as redes Elman, as redes Jordan(18) também possuem uma camada de contexto. No caso das redes Jordan a camada de contexto é realimentada pela camada de saída, em vez da camada escondida. A camada de contexto continua alimentando a camada escondida, juntamente com a camada de entrada.

Um outro detalhe das redes Jordan é que sua camada de contexto também é realimentada, sendo que por ela própria. Uma ilustração da arquitetura de uma RNA Jordan pode ser vista na Figura 5. A quantidade de neurônios na camada de contexto deve ser a mesma da camada de saída.

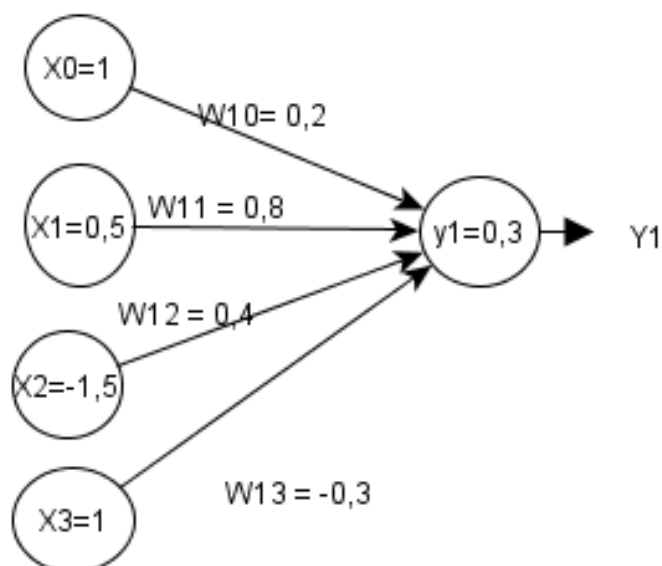


Figura 6: Cálculo realizado por um neurônio.

2.2.2 Treinamento

Uma das habilidades mais importantes das redes neurais e que as torna tão interessantes é a capacidade de aprender. No caso das RNA utilizadas nesse trabalho aprender pode ser definido como a capacidade que a rede tem de melhorar seu desempenho aproximando, cada vez mais, suas respostas geradas, em resposta a um conjunto de entrada, das respostas esperadas para esse mesmo conjunto de entrada. Isso ocorre através do treinamento da rede que consiste no ajuste dos pesos da RNA.

O aprendizado das redes neurais utilizadas nesse trabalho é do tipo supervisionado(11). No aprendizado supervisionado a cada resposta da rede neural, é oferecido como *feedback*

o valor que ela deveria ter dado como resposta. Desse modo, a rede pode ajustar seus pesos para aproximar-se mais da resposta esperada.

O treinamento de uma RNA é composto por vários ciclos ou iterações onde, um ciclo ocorre quando todos os exemplos de treinamento foram mostrados a ela. O ajuste dos pesos da rede neural pode ocorrer quando um exemplo é calculado por ela, ou quando um ciclo termina. No primeiro caso, dizemos que o ajuste é feito *on-line*(8). Já no outro caso temos o ajuste feito em *batch*, também conhecido como em lote(8).

Durante o treinamento o aprendizado da rede é guiado por uma função objetivo. A cada interação o algoritmo de treinamento ajustará os pesos da rede neural para adequar-se a função objetivo. Geralmente a função objetivo utilizada é a função erro médio quadrado.

A fórmula do erro médio quadrático é:

$$E(n) = \frac{\sqrt{\sum_{i=1}^n (d_i - y_i)^2}}{n}, \quad (2.1)$$

onde n é o tamanho da amostra, d_i é o valor desejado, y é o valor calculado e E é o erro médio quadrático.

Na Figura 7 é verificada a evolução do treinamento de uma RNA com atualização dos pesos em lote. Considerando apenas o conjunto de treinamento, observa-se que o erro médio absoluto, função objetivo utilizada no treinamento da RNA que gerou os dados para essa figura, diminui com o passar das iterações, o que indica o aprendizado da RNA.

2.2.2.1 Algoritmos de treinamento

Existem diversos algoritmos de treinamento para uma RNA, e os que serão aqui brevemente descritos se diferenciam apenas no critério de atualização de pesos. Em linhas gerais, as entradas são processadas normalmente pela RNA e seu resultado é comparado com o resultado esperado e o erro cometido pela rede é calculado (fase *forward*).

A partir disso o gradiente do erro é calculado. O valor do gradiente é calculado como o produto do valor da derivada da função de ativação multiplicado pelo erro(13), e isso é feito no sentido contrário ao usual da propagação do sinal(fase *backward*). Esse valor é então usado de algum modo para ajustar os pesos. O modo como ele é usado é dado pelo algoritmo de treinamento.

- **Algoritmo Backpropagation** : é um dos mais antigos e tradicionais métodos de treinamento para redes *feedforward*(8). E usa em conjunto com o gradiente, anteriormente mencionado, um parâmetro chamado: taxa de aprendizado. A taxa de aprendizado indica quanto o gradiente vai afetar a atualização do peso(13).

Um dos problemas desse algoritmo é a possibilidade de ficar preso em mínimos locais o que pode levar a convergência prematura. Por esse motivo é utilizado um outro parâmetro chamado de momento. Sua função é colocar inercia na atualização dos pesos para evitar o problema citado. Mais detalhes sobre esse algoritmo podem ser encontrados em (8, 11, 12).

- **Algoritmo Resilient Propagation Training(RPROP)** : é um algoritmo que utiliza apenas o sinal do gradiente descendente para atualização dos pesos(13). O RPROP não utiliza constantes globais para atualização dos pesos. Em vez disso ele possui uma variável associada a cada peso que determina a magnitude de sua atualização. Essa variável é atualizada a cada iteração da seguinte maneira:

1. Caso valor do erro cometido após os ajustes dos pesos tenha mantido o sinal o fator de atualização é aumentado, já que a atualização do peso relacionado ocorreu na direção que minimiza o erro.
2. Se o sinal do erro cometido tiver sido alterado, então a taxa de ajuste foi exagerada e deve ser decrementada.

Essa abordagem é equivalente a cada peso ter sua própria taxa de aprendizado. Logo, a vantagem deste algoritmo é de que não se precisa definir uma taxa de aprendizagem constante, já que ela é calculada em função do gradiente descendente. O valor de sua inicialização também não tem grandes impactos no treinamento(19). Mais detalhes sobre esse algoritmo de treinamento podem ser encontrados em (13, 19).

2.2.2.2 Critérios de parada

Como pode ser visto na Figura 7 , a cada iteração o resultado calculado pela RNA aproxima-se mais dos valores do conjunto de treinamento. Portanto, enquanto o treinamento continuar é provável que o erro do conjunto de treinamento continue a diminuir, mas isso pode afetar a capacidade de generalização da RNA. Então, precisa-se estabelecer um ou vários critérios de parada para a fase de treinamento.

Dois critérios de paradas clássicos são descritos abaixo:

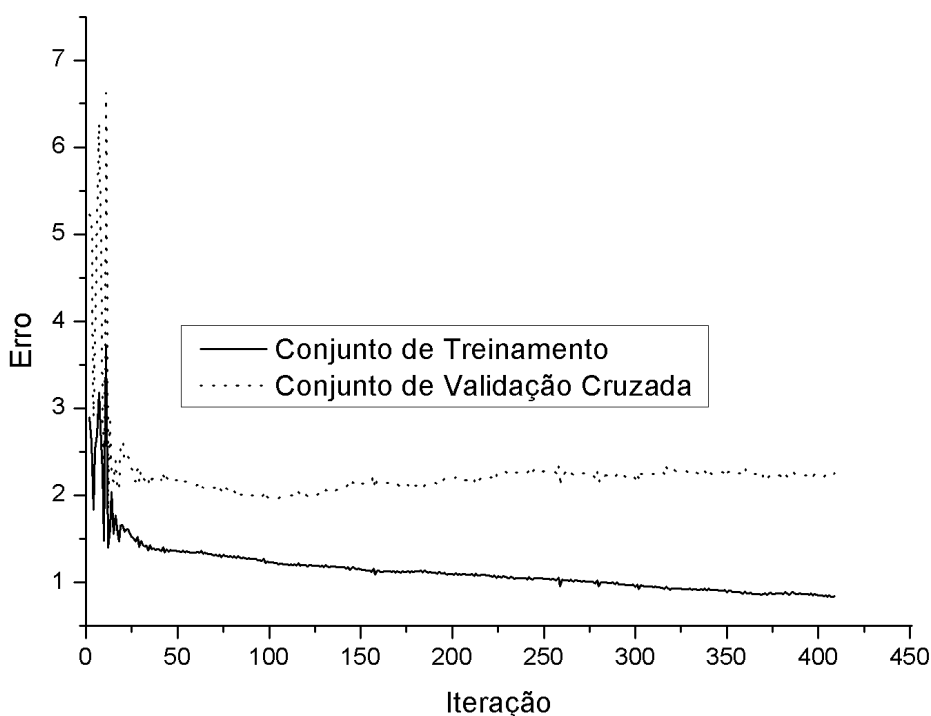


Figura 7: Evolução do erro durante o treinamento da RNA.

Taxa de erro desejada : nesse caso é definida a taxa de erro desejada e quando essa taxa é alcançada o treinamento para e a rede está treinada.

Número máximo de iterações : é definida por quantas iterações a rede deve ser treinada. Quando chegar na iteração definida, o treinamento acaba.

Uma combinação dos dois casos pode ser aplicada para ter um critério mais robusto, como por exemplo: o treinamento deve parar quando o erro médio percentual absoluto do conjunto de teste for 0.1% ou ocorrer 10.000 iterações.

Esses critérios apresentam a desvantagem do treinamento parar antes ou depois do ideal. Prolongar demais o treinamento pode provocar um superajustamento (*overfitting*) da rede que a faria ter um desempenho satisfatório apenas sobre o conjunto de treinamento, pois ela memorizaria esses dados. Por outro lado caso o treinamento pare antes do ideal, a RNA terá um desempenho abaixo do que poderia ser obtido que é conhecido como *underfitting*.

Para evitar esses problemas existe o critério de parada conhecido como **validação cruzada**. Na validação cruzada, além de um conjunto de dados de treinamento é consi-

derado outro conjunto de dados, o conjunto de validação cruzada, onde os dados devem ser estatisticamente representativos.

Esse critério funciona da seguinte forma(11):

1. No final de cada iteração é calculado o erro do conjunto de treinamento e do conjunto de validação cruzada.
2. O valor dos erros de ambos os conjuntos são comparados. Enquanto a rede estiver aprendendo corretamente o valor das taxas de erro decrescerão juntas e o treinamento deve continuar.
3. Quando o erro do conjunto de validação cruzada passar a aumentar enquanto a do conjunto de treinamento continuar a diminuir, é o sinal de que a rede agora está sofrendo um superajustamento e o treinamento deve ser parado.

Os passos descritos anteriormente estão ilustrados no fluxograma da Figura 8. Considerando esses fatos, vemos que o momento da parada do treinamento utilizando esse critério na Figura 7 seria por volta da centésima iteração pois, a partir desse momento, o erro do conjunto de validação cruzada para de diminuir e passa a aumentar.

2.3 Tratamento dos dados

Em trabalhos com dados é necessário de maneira geral realizar um pré-processamento deles. Durante esse pré-processamento algumas vezes se faz necessário excluir alguns dados devido a sua falta de qualidade, bem como preencher, com estimativas, dados faltosos. No caso da série de preços de uma ação da bolsa de valores de São Paulo não existem problemas relacionados à qualidade ou a frequência dos dados.

Para trabalhar com redes neurais é necessário que os dados estejam em certo formato a depender de alguns fatores, como, principalmente da escolha da função de ativação. Algumas funções de ativação, como a sigmoide logística, possuem como saída um valor limitado. A função sigmoide logística é dada por:

$$y(x) = \frac{1}{1 + e^{-x}}. \quad (2.2)$$

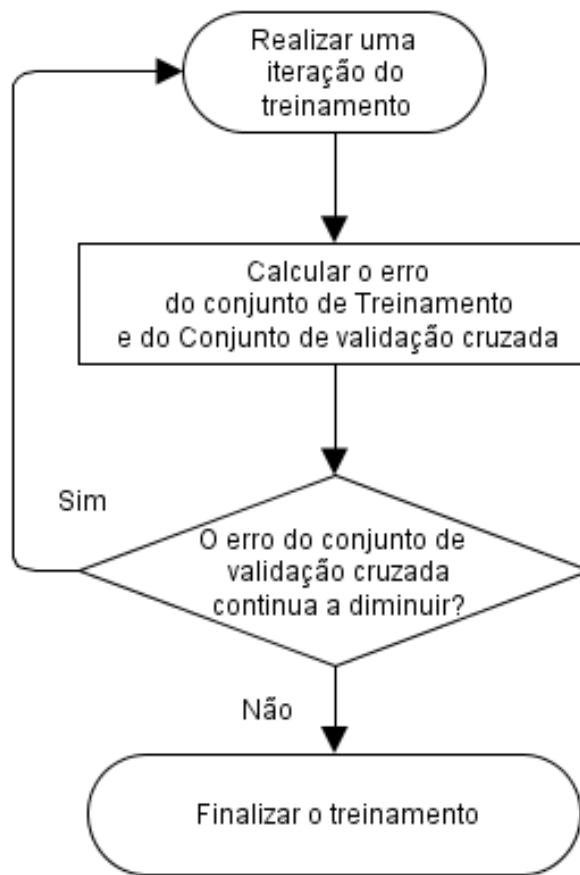


Figura 8: Passos do critério de parada de validação cruzada.

2.3.1 Normalização

A normalização é a responsável por garantir a mesma importância a dados que tenham ordem de grandezas diferentes. Isso evita que a RNA durante o treinamento venha a valorizar variáveis de maior magnitude. A normalização pode ser realizada pela seguinte fórmula:

$$y(x) = \frac{(b - a)(x - x_{min})}{x_{max} - x_{min}} + a, \quad (2.3)$$

onde: y é o valor normalizado; x o valor a ser normalizado; x_{min} e x_{max} o valor mínimo e máximo do conjunto, respectivamente; a e b o menor e o maior valor possível para y , respectivamente.

2.4 Encog

Encog(13) é um *framework* que implementa vários tipos de redes neurais. Ele faz parte de um projeto *open source* e provê as ferramentas necessárias para trabalhar com redes neurais distintas. Além de poder ser utilizado para o desenvolvimento em Java, o Encog possui uma interface gráfica que possibilita o seu uso sem programação.

Além da implementação das redes neurais, ele oferece suporte ao tratamento de dados, aos seus algoritmos de treinamento e diversas funções de ativação. Também implementa outras técnicas de aprendizado de máquina, como algoritmos genéticos, modelo oculto de Markov, entre outros. Durante o treinamento a atualização dos pesos ocorre somente após a apresentação de todos os exemplos a rede, ou seja, treinamento em lote(13)(11).

Apesar de ser uma ótima ferramenta, o Encog deixa a desejar em alguns pontos como o critério de parada do treinamento e o tratamento de dados com tendência, pelo menos até sua versão 3.1.0. No primeiro caso, os critérios de parada disponíveis são os por número de iterações, ou taxa de erro . Quanto a consideração de tendência, não havia um bom tratamento para a séries não estacionárias.

2.5 Trabalhos relacionados

A pesquisa de trabalhos relacionados a esse foi em sua maior parte realizada no portal IEEE Xplore. Foram encontrados diversos trabalhos relacionados, dos quais alguns são descritos na em seguida.

2.5.1 Artigos

2.5.1.1 *Application Study of BP Neural Network on Stock Market Prediction*

O artigo (4) de 2009 analisa os princípios da previsão em bolsa de valores baseadas em redes neurais MLP. Com o algoritmo de treinamento que é uma variação do *backpropagation* a rede foi treinada para realizar previsões semanais. Um detalhe importante é que o critério de parada utilizado no treinamento foi o da taxa de erro desejada. Desse modo, houve apenas dois conjuntos de dados: o de treinamento e o de testes.

2.5.1.2 *Modeling the Behavior of the S&P 500 Index: A Neud Network Approach*

Em (6) de 1994, utilizou redes neurais para mostrar que o comportamento do índice não era aleatório e sim de dinâmica caótica. Um sistema de dinâmica caótica possui uma relação complexa entre suas variáveis, daí fez-se o uso de RNA para validar essa hipótese.

Através do treinamento a RNA poderia achar uma relação entre as variáveis e realizar uma previsão o que rejeitaria a hipótese de aleatoriedade completa. Sua conclusão afirma que a fortes evidências de os preços seguirem algum tipo de modelo determinístico, incluindo o caótico, em vez de um modelo aleatório.

2.5.1.3 *From an Artificial Neural Network to a Stock Market Day-Trading System: A Case Study on the BM&FBOVESPA*

O interessante desse artigo é que as suas ações avaliadas (Petrobras PN e Vale PNA) fazem parte da BM&FBOVESPA, e ambas estão incluídas nessa monografia. Mas o foco da previsão utilizado em (5) eram as negociações que se iniciavam e terminavam no mesmo dia. Ele teve como objetivo desenvolver um sistema utilizando redes neurais que indicasse decisões.

A proposta do artigo foi de integrar as previsões realizadas pelas redes neurais e integra-las com um *trading system* para converter as previsões em decisões. Para o treinamento foi utilizado o algoritmo backpropagation com o critério de parada por número de iterações. O treinamento foi parado quando 100.000 iterações ocorriam. De resultado apresentou como melhor resultado uma taxa de erro médio percentual 50% menor que o melhor resultado do *benchmark* que apresentou o melhor resultado.

2.5.1.4 *Artificial Higher Order Neural Networks for Economics and Business*

O livro(7) de 2009 apresenta uma série de aplicações de redes neurais a problemas de previsão econômica, como cotações de moedas, de ações, entre outros. Mostrando assim, que o problema aqui estudado além de ser atual desperta bastante interesse.

3 Metodologia

Esse capítulo tem como objetivo descrever os passos realizados para o desenvolvimento desse trabalho. De um modo geral, ocorreram basicamente três fases: tratamento dos dados; implementação e execução dos treinamentos e testes com as redes neurais; processamento e análise dos dados gerados pelo passo anterior.

3.1 Processamento dos dados

Os dados utilizados foram os preços de abertura, fechamento, máximo e mínimo. Esses valores foram retirados das cotações diárias dos ativos expostos na Tabela 1. A tabela informa o nome da ação, o seu código, o período das cotações que foi utilizado para esse trabalho e a quantidade de cotações desse período.

Tabela 1: Dados dos ativos utilizados nesse projeto.

Ativo	Código	Data inicial	Data final	Quantidade de cotações
Petrobras PN	PETR4	20/09/2007	19/09/2012	1237
Vale PNA	VALE5	20/09/2007	19/09/2012	1237
Gerdau PN	GGBR4	14/08/2007	14/08/2012	1238
OGX Petróleo ON	OGXP3	11/06/2008	19/09/2012	1062
Ambev PN	AMBV4	20/09/2007	19/09/2012	1237

3.1.1 Normalização e tratamento da tendência na prática

Como foi mostrado na seção 2.3, os dados não podem ser diretamente oferecidos a rede neural. É necessário um processamento prévio que normalize e trate a tendência presente nos dados. E para realizar os tratamentos foi criada a classe `ProcessarDados`.

Para o processamento dos dados é necessário fornecer a quantidade de valores passados e futuros que se deseja prever, além de quais são esses valores entre as seguintes opções: preço de abertura, preço de fechamento, preço de máximo e preço de mínimo. Esses dados informados são as entradas da rede neural e os valores de saída esperados para ela,

formando assim os pares de dados de entrada/saída necessários para o seu treinamento e avaliação.

Por tentativa e erro, foram utilizadas 15 cotações passadas para prever a cotação de 5 dias a frente. Os dados da cotação de entrada foram os preços de abertura, fechamento, máximo e mínimo da ação. E o dado a ser previsto escolhido foi o valor de fechamento do ativo.

Para cada conjunto de entrada/saída é descoberto o maior e o menor valor nos dados do conjunto de entrada, esses dados ficam armazenados para necessidade de desnormalizar os dados no futuro. A normalização é realizada pela equação (2.3) e os valores de a e b foram definidos, após a realização de pequenos testes, como 0,1384 e 0,8616 , respectivamente. Esses valores correspondem ao menor e maior valor possível dos dados normalizados. Quando esse procedimento tiver sido realizado para cada par entrada/saída da RNA, os dados estarão normalizados e com a tendência processada de um modo que não afetará o desempenho da RNA de maneira negativa.

3.1.2 Os conjuntos de dados

Na seção 2.2.2.2 foi apresentado o critério de parada por validação cruzada. De acordo com esse critério, são necessários dois conjuntos de dados para o treinamento da rede, o de treinamento e o de validação cruzada. O primeiro conjunto é o responsável pelo aprendizado da RNA indicando o ajuste dos pesos. O segundo conjunto é o que nos indica o momento ideal de parada do treinamento.

Mas para avaliar o desempenho da RNA é necessário um conjunto de dados que não tenha participado de modo algum do treinamento. Esse conjunto é o conjunto de teste. Então, agora que os dados já estão normalizados, eles são divididos em um conjunto de treinamento, um conjunto de validação cruzada e um conjunto de testes.

Como as redes recorrentes requerem que os dados estejam em sequência, a primeira metade dos dados foi utilizada para treinamento. Da metade dos dados restantes, a primeira metade foi utilizada para o conjunto de validação cruzada enquanto que a segunda metade foi utilizada para o conjunto de testes. Desse modo, as cotações mais antigas estão no conjunto de treinamento e as mais recentes no conjunto de testes. Após essa divisão, os dados de cada conjunto são separados em dados de entrada e saída.

3.1.3 Desnormalização

Por trabalhar com dados normalizados a resposta da RNA também é normalizada. Para descobrir o real valor previsto, foi implementada a função inversa da equação (2.3), dada por:

$$x(y) = \frac{(y - a)(x_{max} - x_{min})}{(b - a)} + x_{min}, \quad (3.1)$$

cujo os parâmetros são os mesmos da equação (2.3).

Essa funcionalidade é usada principalmente para avaliar o desempenho da previsão realizada pela RNA. A medição do desempenho relativo ao conjunto de testes deverá ser realizada com os valores desnormalizados, ao contrário do cálculo do erro dos outros dois conjuntos. Isso ocorre porque enquanto os conjuntos de validação cruzada e de treinamento são utilizados para avaliar o desempenho do treinamento, o conjunto de teste é utilizado para avaliar o desempenho da solução do problema.

3.1.4 Cálculo do erro

A comparação entre os dados é realizada através da comparação dos valores do erro cometido. As medidas de erros utilizadas foram o erro médio quadrado para fase de treinamento, e o erro médio absoluto para a fase de teste.

O Encog possui a implementação do erro médio quadrático (EMQ), mas não possui a do erro médio (EMA) absoluto. Um detalhe sobre o funcionamento do EMQ no Encog, é que esse essa métrica é calculada durante a fase *forward* do treinamento, então quando os pesos são atualizados na fase seguinte o EMQ retornado pelo é na verdade o erro do ajuste de pesos anterior.

Então ambos os cálculos de erro foram implementados. O objetivo foi evitar que detalhes desse nível viessem a afetar o resultado das simulações e também para tornar o programa mais homogêneo possível, no sentido de usar apenas as funcionalidades do Encog relativas a criação, treinamento e cálculo das entradas apresentadas a RNA. O Encog também possui algumas funcionalidades relacionadas ao tratamento de dados, mas que não atendiam a demanda desse trabalho.

3.2 Implementação e execução dos treinamentos e testes

Essa era a parte de maior risco desse projeto e o que poderia levar mais tempo para ser desenvolvido. Mas o uso do framework Encog tornou possível a implementação desse trabalho em menos de 6 meses.

3.2.1 RNA com Encog

O Encog possui as funcionalidades necessárias para implementação e treinamento das RNA utilizadas nesse trabalho: MLP, Elman e Jordan. Com apenas cinco linhas de código criamos uma rede MLP, seis linhas uma rede Elman e Jordan. Abaixo, em Algoritmo 3.1 vemos o método do programa em Java que retorna um RNA MLP criada usando o Encog:

```

1 public static RedeNeural getRedeNeuralPadrao ( ProcessarDados dados, int
   escondida ) {
2     BasicNetwork rede= new BasicNetwork ( ) ;
3     rede.addLayer ( new BasicLayer ( dados.getQtdEntradas ( ) ) ) ;
4     rede.addLayer ( new BasicLayer ( new ActivationSigmoid ( ) , true ,
       escondida ) ) ;
5     rede.addLayer ( new BasicLayer ( new ActivationSigmoid ( ) , false ,
       dados.getQtdSaidas ( ) ) ) ;
6     rede.getStructure ( ) .finalizeStructure ( ) ;
7     rede.reset ( ) ;
8     return new RedeNeural ( dados, rede ) ;
9 }

```

Algoritmo 3.1: Método que cria uma RNA MLP.

De maneira similar são criadas as redes de Jordan e Elman, como mostrados em Algoritmo 3.3 e Algoritmo 3.2 , respectivamente.

```

1 public static RedeNeural getRedeNeuralElman ( ProcessarDados dados, int
   escondida ) {
2     ElmanPattern ep = new ElmanPattern ( ) ;
3     ep.setActivationFunction ( new ActivationSigmoid ( ) ) ;
4     ep.setInputNeurons ( dados.getQtdEntradas ( ) ) ;
5     ep.addHiddenLayer ( escondida ) ;
6     ep.setOutputNeurons ( dados.getQtdSaidas ( ) ) ;
7     BasicNetwork rede = ( BasicNetwork ) ep.generate ( ) ;
8     rede.reset ( ) ;
9     return new RedeNeural ( dados, rede ) ;

```

```
10 }
}
```

Algoritmo 3.2: Método que cria uma RNA Elman.

```
1 public static RedeNeural getRedeNeuralJordan ( ProcessarDados dados, int
  escondida ) {
2     JordanPattern ep = new JordanPattern ( ) ;
3     ep.setActivationFunction ( new ActivationSigmoid ( ) ) ;
4     ep.setInputNeurons ( dados.getQtdEntradas ( ) ) ;
5     ep.addHiddenLayer ( escondida ) ;
6     ep.setOutputNeurons ( dados.getQtdSaidas ( ) ) ;
7     BasicNetwork rede = ( BasicNetwork ) ep.generate ( ) ;
8     rede.reset ( ) ;
9     return new RedeNeural ( dados, rede ) ;
10 }
```

Algoritmo 3.3: Método que cria uma RNA Jordan.

Desse modo podemos criar uma RNA seguindo os seguintes passos:

1. Criar o padrão de RNA: MLP, Elman, Jordan, ou outro tipo disponível no *framework*.
2. Definir a função de ativação: Nesse trabalho foi utilizada apenas a sigmoide logística.
3. Definir a quantidade de neurônios nas camadas: no caso desse experimento foram utilizados 60 neurônios na camada de entrada (15 cotações \times 4 preços), 105 neurônios na camada escondida (definido após pequenos testes) e 5 cinco neurônios na camada de saída (quantidade de cotações à prever) .
4. Finalizar a estrutura da rede e inicializar os pesos aleatoriamente.

Para abstrair esses passos foram criados 3 métodos (Algoritmos 3.1, 3.3 e 3.2) que retornam as redes neurais utilizadas nesse trabalho. Esses métodos recebem como entradas um objeto que tem todos os dados já processados, de onde é retirada a quantidade de entradas e saídas da rede, e a quantidade de neurônios da camada escondida.

3.2.1.1 Treinamento e critério de parada

Existem vários métodos de treinamento, mas o escolhido foi o RPROP por ser o recomendado para utilização com o Encog(13). Para a realização do treinamento com o Encog é necessário seguir os seguintes passos:

1. Criação do conjunto de treinamento : o Encog oferece uma classe para a criação de um objeto de treino. Essa classe recebe como entrada os dados de entrada e os valores que representam a resposta dada pela rede para a entrada relacionada. É mostrado em Algoritmo 3.4, a criação do conjunto de treinamento e do conjunto de validação cruzada. Os dados de entrada e saída para os conjuntos de treinamento e validação cruzada, são os dados normalizados e processados de acordo com o procedimento descrito nas seções 3.1.1 e 3.1.2.

```
1 MDLDataSet treinamento = new BasicMLDataSet ( entrada_t, saida_t ) ;  
2 MDLDataSet vCruzada = new BasicMLDataSet ( entrada_vc, saida_vc ) ;
```

Algoritmo 3.4: Criação dos conjuntos de treinamento e validação cruzada.

2. Criação do objeto de treino: aqui é escolhido o algoritmo de treinamento, a rede que deseja-se treinar e qual o seu conjunto de treinamento. Esses procedimento é mostrado em Algoritmo 3.5.

```
1 MLTrain treinar = new ResilientPropagation ( redeNeural, treinamento  
    ) ;
```

Algoritmo 3.5: Criação do objeto de treino.

3. Realizar as iterações do treinamento: é realizada através da chamada do método *iteration*, como pode ser visto em Algoritmo 3.6.

```
1 treinar.iteration ( ) ;
```

Algoritmo 3.6: Realizando uma iteração.

O item 3 é repetido enquanto o critério de parada não for satisfeito. Um detalhe importante é que ordem cronológica dos valores das ações é mantida, devido a ser um requisito das RNA recorrentes. O critério de parada aqui utilizado foi o de validação cruzada. A implementação desse critério ocorreu utilizando alguns critérios a mais do que os descritos na seção 2.2.2.2.

Como pode ser observado na Figura 7, nas primeiras iterações do treinamento ocorre uma certa instabilidade na taxa de erro, por esse motivo o treinamento ocorre por pelo

menos vinte iterações. Após essas iterações iniciais, passa-se a salvar o estado de treinamento que apresentou o menor erro no conjunto de validação cruzada. Caso não haja uma alteração nesse valor nas próximas 100 iterações, encerre o treinamento e use a rede de menor erro no conjunto de validação cruzada. Esse procedimento pode ser visualizado no fluxograma da Figura 9.

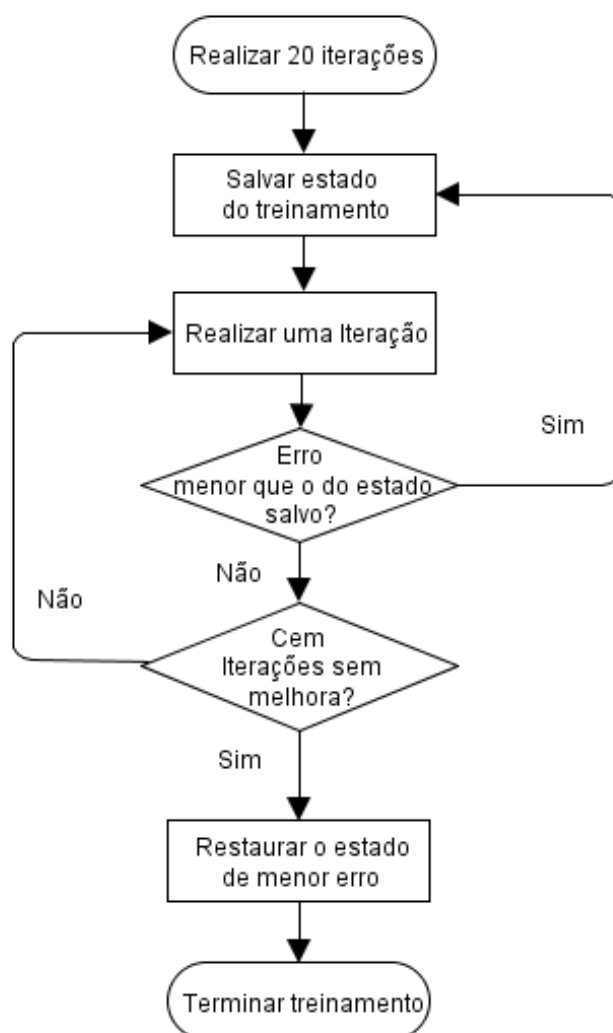


Figura 9: Fluxograma do treinamento de uma RNA comparando o erro do conjunto de validação cruzada.

3.3 Experimento

Para avaliar o desempenho da RNA se utiliza um conjunto de dados que não fez parte do treinamento, isto é, um conjunto de testes. No final do treinamento é apresentado o conjunto de testes para a RNA e os valores do erro cometido na previsão são mensurados através do erro médio absoluto dado pela equação

$$EMA = \frac{1}{N} \sum_{p=1}^N |P_i - O_i|, \quad (3.2)$$

onde N é quantidade de valores do conjunto de verificação, P_i e O_i são o i -ésimo valor previsto e ocorrido, respectivamente.

Tabela 2: EMA da previsão um dia a frente das RNA para a ação PETR4.

Amostra	Elman	Jordan	MLP
1	0,0200156145880232	0,0222512175075539	0,0184878859836125
2	0,0215462450205464	0,0179705870920909	0,0188831958547379
3	0,018377336524931	0,0196765993524593	0,0174833491665251
4	0,0185435194394912	0,0172888918455397	0,018067245193589
5	0,0195947718156088	0,0178741512842203	0,0181717524310211
6	0,0190513406864927	0,0181985646539665	0,0172647197783084
7	0,0217751639437895	0,0194328827692134	0,0174779186659981
8	0,0194047012566486	0,0180462440739479	0,0200244871660977
9	0,0245455837567232	0,019121573332327	0,017386462633845
10	0,0185773351051217	0,0191726601882284	0,019070096849835
11	0,0193232550643213	0,0190764946100427	0,0181728806899414
12	0,020077288813303	0,0183572842931497	0,0176035825722283
13	0,0196788333756289	0,0200125095859378	0,0177212735782587
14	0,0184638876814041	0,0183459803893175	0,0174429061252143
15	0,0202315304741268	0,0191420735816795	0,0181233332995587
16	0,0186423379511996	0,0230092084055104	0,0180925904647444
17	0,0194428168485703	0,0198235542395728	0,0191539935829784
18	0,0225118443302944	0,0194680520142597	0,019476191055109
19	0,0228752439111251	0,0192442167627624	0,0180739134520077
20	0,0181108235310211	0,0233025641493559	0,0176919308100559
21	0,0213163264922717	0,0184040829047207	0,0223582888235628
22	0,0176074016860918	0,0187848634791294	0,017862210984701
23	0,0189662543318906	0,0201030872934845	0,0185414476837276
24	0,0177314434048097	0,0190100989000251	0,0185122074576027
25	0,0209027323667757	0,0216617056097482	0,0219383888560385
26	0,0194013036144774	0,0181880196922336	0,0171965933990307
27	0,0184785730153731	0,019050935233933	0,017154599300922
28	0,0186622034311647	0,0185986339812822	0,0180629835694323
29	0,0180964260001621	0,0190894694192119	0,0184441934689028
30	0,0202556921069892	0,0174476699903777	0,0188142210607812

Com um objetivo de realizar uma comparação estatisticamente confiável, o procedimento de criar, treinar e avaliar uma RNA é repetido trinta vezes. Isso é feito para todos os ativos presentes na Tabela 1 para cada arquitetura de RNA. Para isso foi implementado um programa que realiza e gerencia esses passos gerando um relatório no final.

Para cada execução Arquitetura de RNA/Ativo é gerado um arquivo de relatório. Nesse arquivo constam para cada uma das trinta execuções o número da execução, a iteração que a rede foi treinada, o erro dos conjuntos de treinamento, validação cruzada e teste, além, obviamente, da arquitetura de RNA utilizada e o nome da ação. Na Tabela 2 temos um exemplo resumido dos dados gerados pelas simulações. Nessa tabela é exibido apenas o erro médio absoluto do conjunto de treinamento.

3.4 Resumo: parâmetros utilizados

Os parâmetros utilizados para realização dos testes e que foram expostos nesse capítulo, podem ser consultados na Tabela 3.

Tabela 3: Valores dos parâmetros utilizados na simulação.

Parâmetro	Valor (es)
Quantidade de cotações passadas	15
Cotações futuras a prever	1 e 5
Entradas	Preços de abertura, fechamento, máximo e mínimo
Saída	Preço de fechamento
Valor mínimo do dado normalizado (a)	0,1384
Valor máximo do dado normalizado (b)	0,8616
Critério de parada do treinamento	Validação cruzada
Conjunto de dados utilizados	Treinamento (50%) , validação (25%) e teste (25%)
Função de ativação	Sigmoide logística
Número de amostras por arquitetura	30
Quantidade de neurônios na camada escondida	105

4 *Análise dos resultados*

Para realizar a comparação de desempenho das arquiteturas de RNA para a previsão da cotação das ações foi utilizado o erro médio absoluto. Para tal comparação foram realizados testes estatísticos. O nível de significância estatística estabelecido em todos os testes foi de 5%(0,05) .

Antes de realizar um teste comparativo, foi realizado um teste de normalidade da amostra. O teste aplicado para verificar a normalidade dos dados foi o teste de Shapiro-Wilk(20). Caso as amostras sejam normalmente distribuídas um teste paramétrico, como o teste t de Student é aplicado. Caso contrário, o teste comparativo, não paramétrico de Wilcoxon(20, 21) é aplicado.

4.1 Teste de normalidade

A decisão de aplicar ou não testes estatísticos paramétricos é baseada nos resultados do teste de normalidade. A amostra, onde esse teste foi aplicado, é composta pelos valores do erro médio absoluto cometido na previsão dos preços dos ativos listados na Tabela 1 para a previsão do primeiro e do quinto dia futuro.

Para descobrir se a amostra pode ser considerada normalmente distribuída, foi aplicado o teste de Shapiro-Wilk ao nível de significância 0.05. Para realizar o esse teste é necessário, primeiramente, formular as hipóteses de teste, que são as descritas abaixo.

H_0 : A amostra provém de uma população normal.

H_1 : A amostra não provém de uma população normal.

Para o nível de significância adotado, o valor do coeficiente W , calculado a partir da amostra, deve ser menor que 0,927 para que a hipótese nula (H_0) seja rejeitada, esse valor também está associado com o tamanho da amostra que contém 30 observações. Nesse caso, a rejeição da hipótese nula implica na aceitação da hipótese alternativa (H_1) .

O teste de normalidade foi aplicado para cada amostra de EMA de cada par (Arquitetura de RNA , ativo). E, para melhor visualização, foram montadas duas tabelas com os valores do coeficiente W calculado para cada par. Sendo uma tabela para o valor de W da previsão do primeiro dia a frente, enquanto a outra tabela mostra os valores para a previsão do quinto dia a frente.

Comparando o valor limite de W com os valores calculados para as amostras de um dia a frente, exibidos na Tabela 4, vemos apenas que a previsão da Ambev PN para uma RNA MLP não rejeita a hipótese nula. Isto é, só ela pode ser considerada normalmente distribuída de acordo com o nível de significância de 0,05.

Tabela 4: Valores do coeficiente W do teste de Shapiro-Wilk da previsão para o 1º dia futuro.

Ativo	Elman	Jordan	MLP
Petrobras PN	0,904	0,855	0,793
Vale PNA	0,666	0,740	0,541
Gerdau PN	0,904	0,279	0,881
OGX Petróleo ON	0,921	0,771	0,917
Ambev PN	0,899	0,257	0,948

Foi realizado o mesmo cálculo para a amostra da previsão do quinto dia a frente. E o resultado desses cálculos podem ser vistos na Tabela 5. Repetindo a mesma comparação com o valor de w para esses dados, têm-se que apenas os dados das redes Elman e MLP de Ambev PN não rejeitaram a hipótese nula e podem ser considerados normalmente distribuídos.

Tabela 5: Valores do coeficiente W do teste de Shapiro-Wilk da previsão para o 5º dia futuro.

Ativo	W Elman	W Jordan	W MLP
Petrobras PN	0,537	0,702	0,816
Vale PNA	0,601	0,698	0,448
Gerdau PN	0,886	0,438	0,444
OGX Petróleo ON	0,402	0,447	0,922
Ambev PN	0,930	0,251	0,967

Com os resultados apresentados nas Tabelas 4 e 5, fica inviável utilizar um teste paramétrico, como o teste t de Student para a comparação de desempenho das arquiteturas de RNA. Por esse motivo será utilizado o teste não paramétrico de Wilcoxon.

4.2 Teste comparativo

Métodos para inferência estatística paramétricos assumem que os dados, objetivos de estudo, pertencem a alguma distribuição e que por causa disso podem ser descritos em poucos parâmetros, que caracterizam essa distribuição. Por exemplo, o comportamento probabilístico de um dado normalmente distribuído pode ser descrito apenas por sua média e desvio padrão, já que esses são os parâmetros que caracterizam uma distribuição normal(20).

Já os métodos não paramétricos, como o nome já diz, não fazem afirmações sobre a natureza da distribuição dos dados. E podem ser usados quando a distribuição dos dados é desconhecida, ou não é possível o uso de técnicas paramétricas(21). Um dos testes estatísticos não paramétrico comumente usado é o teste de Wilcoxon.

O teste de Wilcoxon verifica a magnitude da diferença entre dois dados através da comparação das medianas das amostras. Esse teste assume que a distribuição das diferenças é simétrica. Para amostras grandes, acima de 20 valores, o parâmetro T desse teste se aproxima da distribuição normal.

A avaliação da previsão das arquiteturas de RNA ocorreu através da comparação do erro médio absoluto da previsão. A arquitetura que obteve o menor EMA foi considerada a melhor. Para realizar essa comparação, os desempenhos das arquiteturas por ação são comparados dois a dois (Elman e Jordan, Elman e MLP, Jordan e MLP) .

O procedimento citado foi feito tanto para a previsão de um dia a frente como a previsão do quinto dia futuro e em ambos os casos foram realizadas duas comparações. A primeira verifica se o desempenho das arquiteturas são semelhantes. Caso não seja, é realizado um segundo teste para averiguar qual arquitetura apresenta o melhor desempenho. Como os dados não foram considerados normalmente distribuídos, o teste estatístico utilizado foi o de Wilcoxon.

4.2.1 Teste de igualdade

O primeiro passo da comparação foi verificar se as duas amostras comparadas são iguais ou diferentes. Para tal foram elaboradas as seguintes hipóteses:

H_0 : Não existe diferença entre as medianas dos erros das duas amostras.

H_1 : As medianas dos erros das duas amostras são diferentes.

O teste de Wilcoxon foi aplicado para avaliar essas hipóteses a cada par de arquiteturas de RNA. Isso foi feito para as previsões do primeiro e quinto dia futuro. Como a quantidade de amostras é grande pode-se utilizar a curva normal para a avaliação das hipóteses.

Os resultados da aplicação do teste de Wilcoxon podem ser vistos nas Tabelas 6 e 7, onde a primeira representa a avaliação da previsão de um dia a frente e a segunda a avaliação da previsão do quinto dia. Na tabela temos na primeira coluna as comparações efetuadas e na primeira linha o ativo que gerou os dados. Nas células que representam o cruzamento da comparação com o ativo é exibido o valor de Z calculado. Para rejeitar a hipótese nula com uma significância de 0,05 é necessário que $|Z| > 1,96$ (valor tabelado)

Observando a Tabelas 6, vemos que com exceção do ativo PETR4, as arquiteturas MLP e Jordan apresentaram desempenho estatisticamente semelhantes. As previsões de Elman e Jordan também foram estatisticamente semelhantes para os ativos PETR4 e OGXP3. E todas as arquiteturas de RNA tiveram estatisticamente o mesmo desempenho na previsão do ativo OGXP3. O valor nas demais células indica H_1 como a hipótese vencedora. Isto é, os dados são estatisticamente diferentes.

Tabela 6: Teste de igualdade para o EMA da previsão do primeiro dia a frente.

	PETR4	VALE5	GGBR4	OGXP3	AMBV4
EMA(Elman)≠EMA(Jordan)	$Z = 1,152$	$Z = 4,216$	$Z = -2,859$	$Z = -0,082$	$Z = -2,509$
EMA(Elman)≠EMA(MLP)	$Z = 3,517$	$Z = 3,024$	$Z = -1,995$	$Z = -1,563$	$Z = -2,96$
EMA(Jordan)≠EMA(MLP)	$Z = 2,674$	$Z = -0,884$	$Z = 0,432$	$Z = -0,782$	$Z = -0,579$

Já para os dados das Tabelas 7 os valores são estatisticamente diferentes apenas para as comparações de Elman com Jordan e Elman com MLP para VALE5 e AMBV4 e para GGBR4 na comparação entre Jordan e MLP. Para os demais resultados, a hipótese H_0 não pode ser rejeitada e os dados são considerados estatisticamente iguais.

Tabela 7: Teste de igualdade para o EMA da previsão do quinto dia a frente.

	PETR4	VALE5	GGBR4	OGXP3	AMBV4
EMA(Elman)≠EMA(Jordan)	$Z = -1,687$	$SZ = 3,270$	$Z = -1,275$	$Z = 0,514$	$Z = 2,633$
EMA(Elman)≠EMA(MLP)	$Z = -1,296$	$Z = 2,674$	$Z = 1,666$	$Z = 0,596$	$Z = 2,941$
EMA(Jordan)≠EMA(MLP)	$Z = 0,213$	$Z = -0,679$	$Z = 2,406$	$Z = 0,935$	$Z = -1,152$

4.2.2 Comparação das amostras

Agora que são conhecidas quais amostras não são iguais entre si, é necessário outro teste de Wilcoxon para verificar qual delas apresenta o menor EMA na previsão. O procedimento seguido foi similar ao da seção 4.2.1, com algumas pequenas modificações que começam pelas novas hipóteses:

H_0 : A mediana do EMA da arquitetura X é menor que a mediana do EMA da arquitetura Y .

H_1 : A mediana do EMA da arquitetura X é maior que a mediana do EMA da arquitetura Y .

O teste acima foi aplicado para cada par de arquiteturas que teve hipótese de igualdade rejeitada no teste anterior. Como a comparação são entre as mesmas amostras o valor de Z é o mesmo do teste anterior.

Os resultados dos testes de Wilcoxon são mostrados nas Tabelas 8 e 9. O primeiro elemento de qualquer linha da primeira coluna dessas tabelas está relacionado a X , enquanto o segundo está relacionado a Y . Nas células da tabela temos a resposta para a afirmação da primeira coluna, onde essa resposta é cabível.

Observando a Tabela 8, que está relacionada à previsão de um dia a frente, nota-se que a arquitetura de Elman apresentou um resultado estatisticamente melhor apenas para a previsão do ativo GGBR4. A Tabela 10 contém os dados que serviram de base para essa análise. Nos demais casos, Elman perde para as arquiteturas Jordan e MLP nos ativos onde houve esse teste. Já para a ação PETR4, a rede MLP obteve o melhor resultado, enquanto que para os ativos VALE5 e AMBV4 ela empatou com a arquitetura de RNA Jordan.

Tabela 8: Valores teste de Wilcoxon para a previsão do primeiro dia a frente.

	PETR4	VALE5	GGBR4	OGXP3	AMBV4
EMA (Elman) < EMA (Jordan)	Iguais	Não	Sim	Iguais	Não
EMA (Elman) < EMA (MLP)	Não	Não	Sim	Iguais	Não
EMA (Jordan) < EMA (MLP)	Não	Iguais	Iguais	Iguais	Iguais

Para a Tabela 9, que está relacionada a previsão do quinto dia a frente, temos como ocorrido para a previsão de um dia um empate entre MLP e Jordan para os ativos VALE5 e AMBV4, enquanto que Elman estatisticamente perde para os dois nesses ativos. Isso

Tabela 9: Valores teste de Wilcoxon para a previsão do quinto dia a frente.

	PETR4	VALE5	GGBR4	OGXP3	AMBV4
EMA (Elman) < EMA (Jordan)	Iguais	Não	Iguais	Iguais	Não
EMA (Elman) < EMA (MLP)	Iguais	Não	Iguais	Iguais	Não
EMA (Jordan) < EMA (MLP)	Iguais	Iguais	Não	Iguais	Iguais

Tabela 10: EMA da previsão um dia a frente das RNA para a ação GGBR4.

Amostra	Elman	Jordan	MLP
1	0,0175537478718386	0,0198875593615543	0,0211976419021641
2	0,0180493215476363	0,0211933511138604	0,019913247286119
3	0,0218840575015476	0,0200849467421032	0,0208444032824404
4	0,0199607275738111	0,0207431239317142	0,0193856548890672
5	0,0217873297149565	0,0256844513197723	0,0197850113936091
6	0,0232625581204266	0,0208815612197999	0,019996727144761
7	0,0232928113380664	0,0241739265319825	0,0192421325178241
8	0,0176019098922113	0,0196881798635152	0,0202408643742556
9	0,0178484708571505	0,0196461735220954	0,020747889979549
10	0,0179887456361109	0,0207990916779613	0,0218381880302351
11	0,0184365114730295	0,0223774572747595	0,0200460787822538
12	0,0186343977398947	0,0202754050637543	0,0198173494798584
13	0,018602451683617	0,0196635494974225	0,0225095351528453
14	0,0202948036721072	0,0209192619504239	0,0191038460084007
15	0,0231474056756854	0,021143195832458	0,022128593813031
16	0,0190911826074107	0,0203014399711802	0,0198936908336739
17	0,0216840388872005	0,0775877448753155	0,0198920138302231
18	0,017543816900599	0,0202736892122609	0,0216863085563304
19	0,019509995820332	0,0207113911722513	0,0212869221859793
20	0,0181447023820065	0,0207401660056724	0,0203436876264674
21	0,0177351591545761	0,0210737093213687	0,0200458498197163
22	0,022141429460324	0,0207844286503933	0,0199652379563936
23	0,0185005660976384	0,0197752882660126	0,0203128692234499
24	0,0210143468753846	0,0199636643554095	0,0196423681656817
25	0,0196410379673308	0,0237990501599992	0,022653953187846
26	0,0250436809895523	0,0229456949079437	0,025074859785252
27	0,0206092335313508	0,0197824817910837	0,022941665912758
28	0,0196672128194495	0,020110234676167	0,0208183381621717
29	0,0182542943403761	0,0212880868858888	0,0211021800448516
30	0,019543140289112	0,0195597303373406	0,0191924871581202

faz com que reste apenas a análise do ativo GGBR4 onde a arquitetura MLP apresentou o melhor desempenho e ocorreu um empate entre Elman e Jordan.

4.3 Previsões

Essa seção tem como objetivo mostrar graficamente o resultado das previsões. Todas as previsões realizadas e exibidas nos gráficos dessa seção foram retiradas do conjunto de teste. Ou seja, os dados utilizados para plotar os gráficos não fizeram parte do treinamento da RNA e são mostrados a ela pela primeira vez.

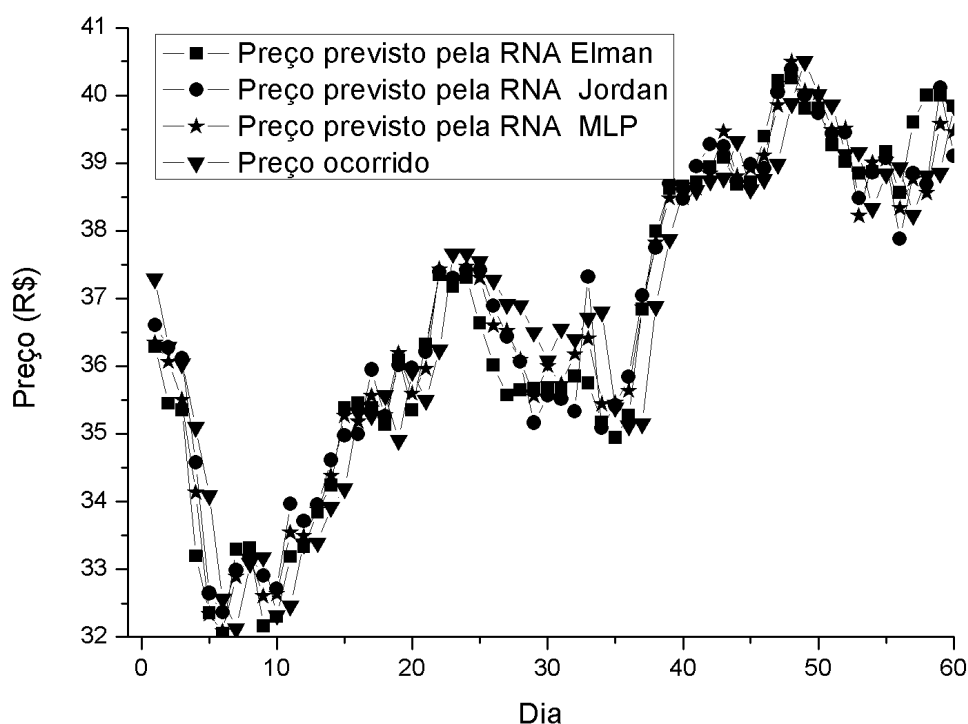


Figura 10: Valores previstos para a cotação do ativo VALE5 para um dia a frente.

Na Figura 10, foram plotados no gráfico os valores previstos para sessenta dias pelas RNA Elman, Jordan e MLP para o ativo VALE5. A previsão tenta prever um dia a frente. No gráfico é possível visualizar que os valores estão bastante próximos.

A Figura 11 utiliza os mesmos dados da Figura 10, só que deixa apenas os dados da rede MLP e do valor ocorrido. O número de previsões foi de cem em vez das sessenta. Nesse gráfico é possível observar que os valores previstos estão realmente bem próximos dos valores ocorridos.

Tabela 11: Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente.

	Média	Desvio Padrão	Mínimo	Mediana	Máximo
Elman	0,02323	$9,81 \times 10^{-3}$	0,01549	0,01937	0,0484
Jordan	0,01642	$1,07 \times 10^{-3}$	0,01536	0,01536	0,02094
MLP	0,01758	$4,54 \times 10^{-3}$	0,01484	0,01625	0,03788

Para melhor visualização dos erros, alguns dados estatísticos obtidos da avaliação das 30 amostras do EMA do ativo VALE5 podem ser visualizados na Tabela 11. Os dados são referentes à previsão de um dia a frente e são úteis pois eles revelam o resumo da análise de trinta amostras, enquanto que o gráfico mostra os dados de uma execução.

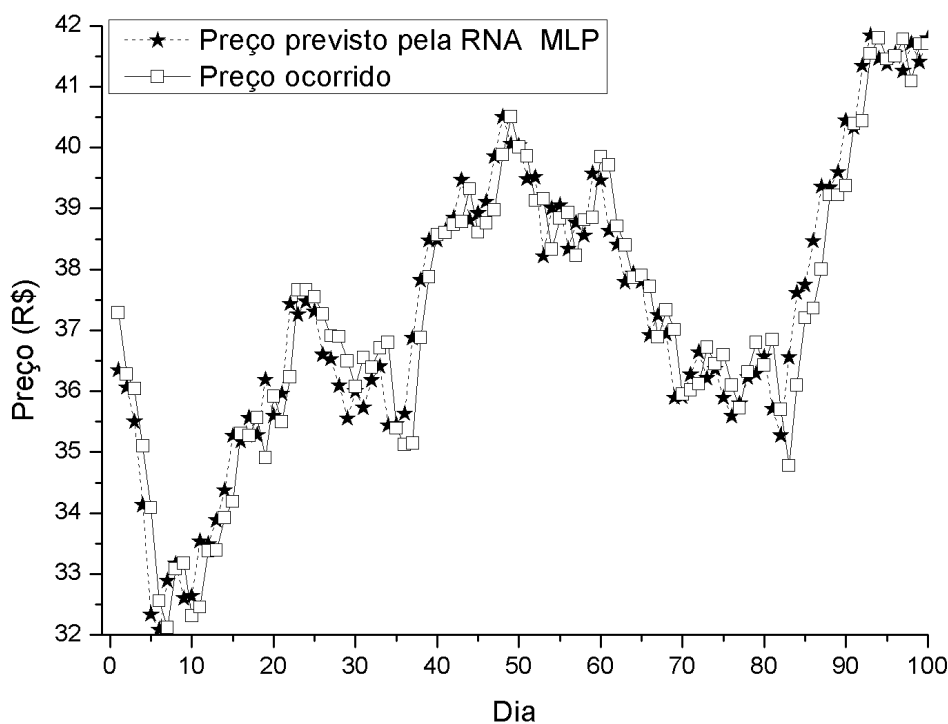


Figura 11: Comparação da previsão para o primeiro dia a frente com o valor ocorrido para VALE5.

Já na Figura 12 podemos ver uma maior diferença entre os valores ocorridos e os valores previstos. As previsões realizadas pelas RNA Elman, Jordan e MLP apresentam valores bem próximos entre si, mas que diferem do valor ocorrido.

Aumentando o número de previsões da Figura 12 para cem e retirando os dados das RNA Elman e Jordan, temos a Figura 13. Nessa figura vemos que já existe uma separação maior entre o valor previsto e ocorrido do que o da Figura 11, isto era esperado, já que para

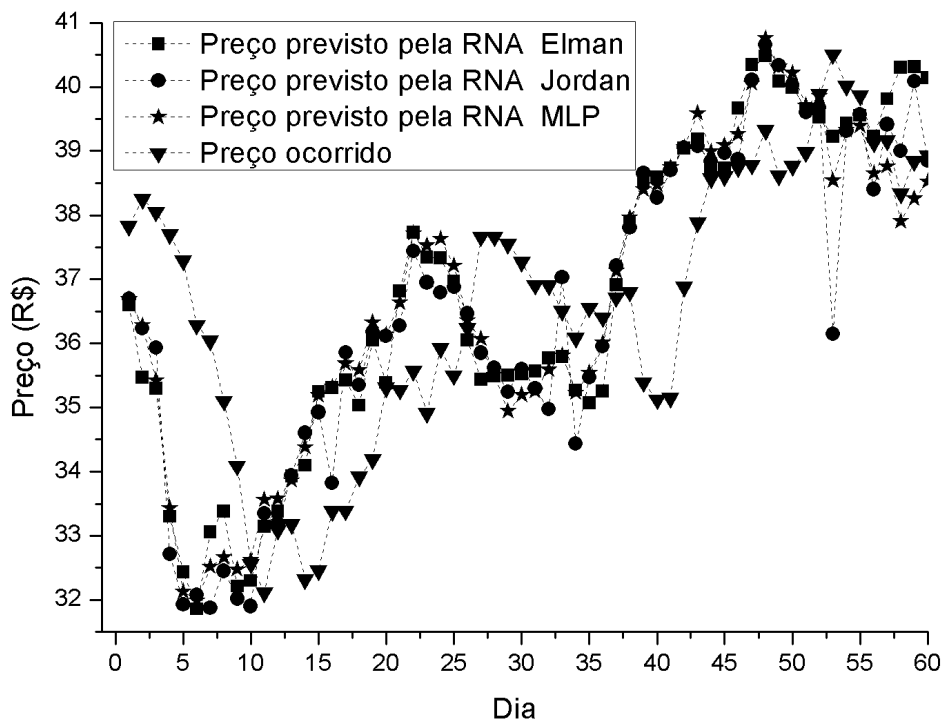


Figura 12: Valores previstos para a cotação do ativo VALE5 para o quinto dia a frente.

os valores mais a frente a possibilidade de eventos que influenciam também é aumentada o que torna a previsão mais difícil, aumentando sua taxa de erro.

Tabela 12: Dados estatísticos das 30 amostras do EMA cometido na previsão do quinto dia a frente.

	Média	Desvio Padrão	Mínimo	Mediana	Máximo
Elman	0,0399	$6,664 \times 10^{-3}$	0,03556	0,03732	0,05853
Jordan	0,0364	$0,9619 \times 10^{-3}$	0,03553	0,03613	0,03978
MLP	0,0377	$4,960 \times 10^{-3}$	0,03509	0,03625	0,05585

Na Tabela 12 temos os mesmos dados estatísticos da Tabela 11 sendo que são referentes a previsão do quinto dia a frente. Comparando essas duas tabelas vemos que as medidas referentes ao EMA como o mínimo, a mediana e o EMA máximo são maiores para a Tabela 12 do que Tabela 11. E isso pode ser visto nos gráficos relacionados a previsão.

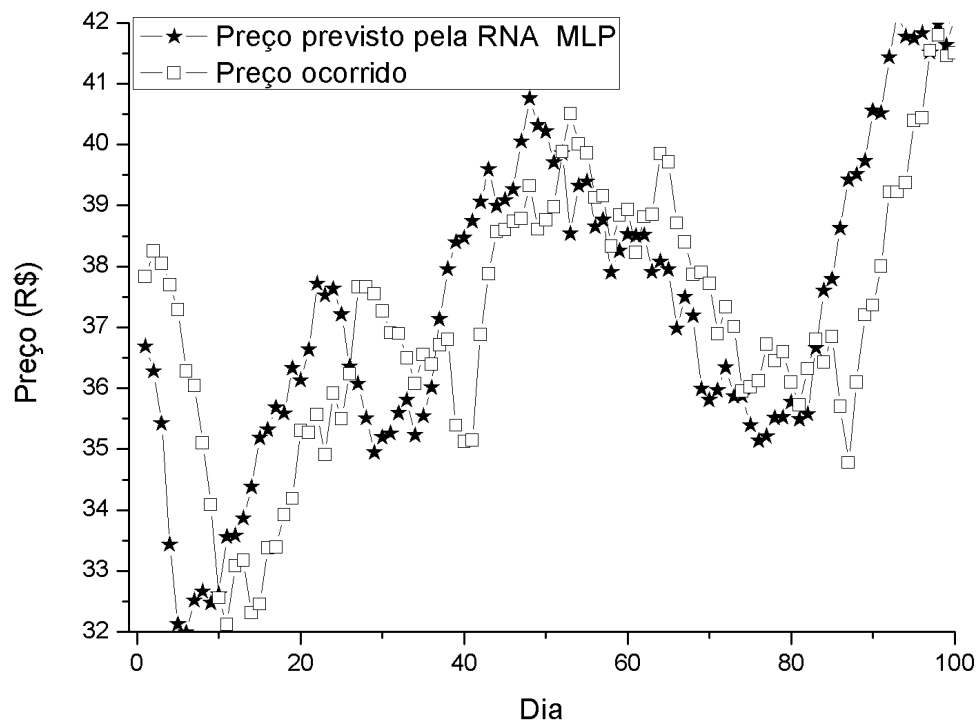


Figura 13: Comparação da previsão para o quinto dia a frente com o valor ocorrido para VALE5.

5 Considerações Finais

Esse trabalho realizou um estudo comparativo entre diferentes arquiteturas de redes neurais para previsão de ações na bolsa de valores. Para avaliar o desempenho dessas redes foram realizados testes estatísticos, nesse caso, o teste de Wilcoxon por causa da não normalidade verificada nos dados. Um resumo do resultado da execução desse teste pode ser visualizado na Tabela 13, onde para cada ativo é indicada a rede que teve o menor EMA, para a previsão do primeiro e do quinto dia a frente.

Tabela 13: Resumo do teste comparativo.

[Fonte: elaboração própria]

Ativo	Melhor previsão para o 1º dia	Melhor previsão para o 5º dia
Petrobras PN	MLP	Empate
Vale PNA	Jordan e MLP	Jordan e MLP
Gerdau PN	Elman	MLP
OGX Petróleo ON	Empate	Empate
Ambev PN	Jordan e MLP	Jordan e MLP

Analisando os resultados obtidos pelos testes, observa-se que na maior parte dos casos as RNA MLP obtiveram um desempenho estatisticamente semelhante aos da RNA de Jordan. Quanto as RNA de Elman, no geral, apresentaram um desempenho estatisticamente inferior as outras duas arquiteturas de RNA. Uma exceção foi para a previsão do ativo GGBR4 para um dia a frente. Um resumo dos dados desse experimento pode ser visto na Tabela 14. Nota-se que nesse caso a RNA Elman obteve a menor média, mediana, valores máximos e mínimos de EMA.

Tabela 14: Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente para o ativo GGBR4.

[Fonte: elaboração própria]

	Média	Desvio Padrão	Mínimo	Mediana	Máximo
Elman	0,01988	$2,07 \times 10^{-3}$	0,01754	0,01953	0,02504
Jordan	0,02286	$10,44 \times 10^{-3}$	0,01956	0,02074	0,07759
MLP	0,02072	$1,34 \times 10^{-3}$	0,0191	0,02028	0,02507

As redes de Jordan não apresentaram o desempenho esperado para um rede recorrente.

As melhores previsões realizadas pelas redes Jordan obtiveram desempenho estatisticamente igual ao das redes MLP. Já quando comparado com a rede Elman, as redes Jordan só tiveram um desempenho inferior para a previsão do primeiro dia a frente para a ação da Gerdau.

Um outro caso interessante foi o da previsão para um dia e cinco dias a frente da OGXP3. Em ambos os casos todas as arquiteturas apresentaram desempenho estatisticamente idênticos. Na Tabela 15 é mostrado um resumo estatístico dos dados do experimento da previsão de um dia a frente.

Tabela 15: Dados estatísticos das 30 amostras do EMA cometido na previsão de um dia a frente para o ativo OGXP3.

[Fonte: elaboração própria]

	Média	Desvio Padrão	Mínimo	Mediana	Máximo
Elman	0,0336	$3,53 \times 10^{-3}$	0,02873	0,03321	0,04467
Jordan	0,03651	$24,15 \times 10^{-3}$	0,02766	0,02964	0,01245
MLP	0,03153	$3,87 \times 10^{-3}$	0,0273	0,02986	0,0412

Pode-se concluir para os experimentos aqui realizados que com as bases utilizadas nesse trabalho que as redes MLP, mesmo com sua simplicidade, obteve em média um desempenho igual ou superior as redes recorrentes de Elman e Jordan.

5.1 Sugestões para trabalhos futuros

Um dos trabalhos futuros relacionados à arquitetura das redes aqui utilizadas seria a utilização de uma técnica de otimização associada a essas redes com o objetivo de definir a arquitetura ótima para cada caso. O processo de tentativa e erro aqui utilizado pode ter prejudicado o desempenho das redes recorrentes, uma vez que, apenas algumas possíveis arquiteturas foram testadas. Outras alternativas para o problema seriam:

1. Tentar utilizar outras entradas além dos preços das ações para as RNA. Essa entrada poder ser indicadores da economia, indiciadores fundamentalistas ou indicadores técnicos, como o índice de força relativa(15) ou média móveis(15).
2. Tentar prever a tendência em vez do valor da cotação. A previsão da tendência seria um indicativo para o investidor se posicionar.
3. Uma outra experiência interessante é o uso de outras funções objetivos durante o treinamento para a solução desse problema. Os gráficos passam a impressão que

quando se aumenta o intervalo da previsão, a rede passa a dizer que o valor para esse dia futuro será próximo do valor do dia atual. Para minimizar o erro quadrático a RNA indica que a melhor estimativa é estimar o valor futuro como sendo próximo do valor do dia atual.

4. Finalmente, uma abordagem mais robusta é utilizar todas essas sugestões juntas através de um comitê de redes neurais(12). Onde existem várias redes neurais especializadas em alguma função, como prever tendência, por exemplo.

Referências

- 1 RASSIER, L. H. *Aprenda a investir na bolsa de valores com ênfase em análise técnica*. 3. ed. [S.l.]: Xp Educação, 2009.
- 2 BM&FBOVESPA. *Sobre a Bolsa*. Setembro 2012. <http://www.bmfbovespa.com.br/pt-br/intros/intro-sobre-a-bolsa.aspx?idioma=pt-br>.
- 3 BM&FBOVESPA. *BM&FBOVESPA divulga balanço de operações de agosto*. Setembro 2012. <http://bmfbovespa.comunique-se.com.br/Show.aspx?IdMateria=zCkrAEdARwr6/ukZ6IISTA>.
- 4 LI, F.; LIU, C. Application study of bp neural network on stock market prediction. In: *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*. [S.l.: s.n.], 2009. v. 3, p. 174 –178.
- 5 MARTINEZ, L. et al. From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. [S.l.: s.n.], 2009. p. 2006 –2013. ISSN 1098-7576.
- 6 MALLIARIS, M. Modeling the behavior of the s&p 500 index: a neural network approach. In: *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*. [S.l.: s.n.], 1994. p. 86 –90.
- 7 ZHANG, M. *Artificial Higher Order Neural Networks For Economics And Business*. [S.l.]: Igi Global, 2009. (Premier Reference Source). ISBN 9781599048970.
- 8 OLIVEIRA, E. M. de J. *Previsão da Série Temporal do Índice da Bolsa de Valores de São Paulo Usando Redes Neurais e Estatística*. Dissertação (Dissertação de Mestrado) — Centro de Informática - Universidade Federal de Pernambuco, Recife-PE, 2001.
- 9 ROSENBLATT, F. Perceptron simulation experiments. *Proceedings of the IRE*, v. 48, n. 3, p. 301 –309, march 1960. ISSN 0096-8390.
- 10 WIDROW, B.; HOFF, M. E. Neurocomputing: foundations of research. In: ANDERSON, J. A.; ROSENFELD, E. (Ed.). Cambridge, MA, USA: MIT Press, 1988. cap. Adaptive switching circuits, p. 123–134. ISBN 0-262-01097-6.
- 11 VALENÇA, M. *Fundamentos das Redes Neurais*. 2. ed. Olinda - PE: Livro Rapido, 2010. 310 p.
- 12 HAYKIN, S. *Redes Neurais - 2ed*. [S.l.]: BOOKMAN COMPANHIA ED, 2001. ISBN 9788573077186.

- 13 HEATON, J. *Programming Neural Networks With Encog 2*. St. Louis, MO USA: Heaton Reseach, Inc, 2010.
- 14 MURPHY, J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. [S.l.]: New York Inst. of Finance, 1999. (New York Institute of Finance Series). ISBN 9780735200661.
- 15 HILGERT, S. P. *Análise gráfica: domine as técnicas mais utilizadas no mercado*. 3. ed. [S.l.]: Xp Educação, 2009.
- 16 MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, Springer New York, v. 5, n. 4, p. 115–133, dez. 1943. ISSN 0007-4985.
- 17 HEATON, J. *Introduction to Neural Networks with Java*. 2. ed. St. Louis, MO USA: Heaton Reseach, Inc, 2008.
- 18 JORDAN, M. Generic constraints on underspecified target trajectories. In: *Neural Networks, 1989. IJCNN., International Joint Conference on*. [S.l.: s.n.], 1989. p. 217 –225 vol.1.
- 19 RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: *Neural Networks, 1993., IEEE International Conference on*. [S.l.: s.n.], 1993. p. 586 –591 vol.1.
- 20 BOX, G. E. P.; HUNTER, J. S.; HUNTER, W. G. *Statistics for experimenters*. 2. ed. [S.l.]: Wiley-Interscience, 2005.
- 21 GIBBONS, J. D.; CHAKRABORTI, S. *Nonparametric statistical Inference*. 4. ed. New York: Marcel Dekker, 2003.