LUCAS JOÃO PEDRO SCHEREMETA

DESENVOLVIMENTO E AVALIAÇÃO DE REDE NEURAL APLICADA EM OPERAÇÕES EM BOLSA DE VALORES

LUCAS JOÃO PEDRO SCHEREMETA

DESENVOLVIMENTO E AVALIAÇÃO DE REDE NEURAL APLICADA EM OPERAÇÕES EM BOLSA DE VALORES

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Galdenoro Botura Junior

S326d Scheremeta, Lucas João Pedro

DESENVOLVIMENTO E AVALIAÇÃO DE REDE NEURAL

APLICADA EM OPERAÇÕES EM BOLSA DE VALORES / Lucas

João Pedro Scheremeta. -- , 2021

55 f.

Trabalho de conclusão de curso (-) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências Farmacêuticas, Araraquara, Orientador: Galdenoro Botura Junior

1. Redes neurais. 2. Bolsa de valores. 3. Mercado de ações Previsão. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências Farmacêuticas, Araraquara. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Ao meu filho, Elias Marino Scheremeta, que veio alegrar a minha vida e motivar-me a concluir o curso. Com apenas dois anos de idade, já mudou a vida de muita gente para melhor e ainda passou parte de seu tempo no meu colo enquanto eu escrevia este trabalho. O pequeno Elias é um grande orgulho para mim e a razão pela qual tenho tentado guiar melhor cada ação do meu viver.

AGRADECIMENTOS

Primeiramente, a Deus, por me dar a sabedoria e a capacidade necessárias para concluir essa graduação, durante a qual houve tantos altos e baixos, e por estar sempre ao meu lado, zelando por mim, mantendo minha plenitude e dando-me forças a todo momento.

Aos meus amados pais, Silvio Roberto Scheremeta e Roseli Maria João Pedro, por toda educação que me propuseram desde novo e por disporem seu apoio emocional e financeiro durante todos esses anos: o contínuo incentivo que me deram, inclusive as cobranças inerentes a ele, e a compreensão diante das adversidades foram fundamentais. Agradecimento extensível ao meu irmão, Rafael João Pedro Scheremeta, com o qual vivi memoráveis momentos desde criança, principalmente pela pouca diferença de idade, e vivenciei grande parte da faculdade. Mesmo sendo de outra área (Engenharia Ambiental), pudemos partilhar momentos de estudo, dificuldades e lazer.

À minha esposa, Vijahia Marino de Lima, e ao meu filho, Elias Marino Scheremeta: graças ao apoio diário que me deram nos últimos anos, fui encorajado a concluir o curso e este trabalho.

À Universidade Estadual Paulista "Júlio de Mesquita Filho" — Instituto de Ciência e Tecnologia — Campus de Sorocaba. Um agradecimento a todos os docentes, em especial ao meu orientador, Prof. Dr. Galdenoro Botura Junior, por todas as oportunidades que abriu para mim: desde a oportunidade de ser monitor de Circuitos Digitais II, até oportunidades de prestar serviços; demais, por fim, pelo apoio teórico e técnico para me ajudar a concluir este trabalho.

A todos os colegas de curso, especialmente aos da república e aos demais colegas com os quais dividi moradia durante esses anos.

Por fim, ao movimento escoteiro, em especial ao Grupo Escoteiro Goyotin — 34º — Tatuí/SP, que sempre me acolheu e me apoiou durante a graduação.

SCHEREMETA, L. J. P. Desenvolvimento e avaliação de rede neural aplicada em operações em bolsa de valores. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) — Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2021.

RESUMO

Desde a década de 1980, com o avanço tecnológico do mercado financeiro, vem-se buscando (por meio de teorias matemáticas, análise técnica e outros mecanismos) prever o comportamento do mercado de ações. Graças ao avanço tecnológico, que culminou em computadores com melhores processadores, internet das coisas, linguagens de programação e maior velocidade de transmissão de dados, foi possível automatizar estratégias para prever o mercado de ações. As pesquisas têm apresentado grandes avanços, e as aplicações de redes neurais cada vez mais fazem parte do cotidiano das pessoas, cujas áreas e aplicações são diversas. Dada a demanda da sociedade por ferramentas que permitam o acompanhamento do mercado de ações, o objetivo deste trabalho é desenvolver uma ferramenta que indique operações no mercado de ações, mais especificamente na empresa de infraestrutura do mercado financeiro Brasil Bolsa Balcão, que é a única regulamentada no país. Para isso, o foco foi desenvolver uma eficiente rede neural artificial que realizasse operações de compra e venda de ações, na modalidade de swing-trade. O período analisado contemplou operações com duração de dias a algumas semanas dos ativos da Vale, da Petrobras, do Banco do Brasil e da Brasil Bolsa Balcão. Como bases de dados de treinamento da rede neural foram levantados o preço de fechamento diário (do período de 2015 a 2019), e para a aplicação da rede neural foram usados dados ao longo de dez dias (de 2020 a 2021). Devido às especificidades dos ativos, foi necessário criar uma rede neural para cada ativo, obtendo-se um resultado 79,48% de taxa de acerto em média e 1,6 de relação de lucro/perda, resultado satisfatório e positivo ao final do período, considerando que o período foi de incertezas e grande volatilidade no mercado financeiro a nível mundial. Em síntese, a ferramenta desenvolvida utiliza rede neural artificial e técnicas de aprendizagem por reforço como forma de melhorar a tomada de decisão, com alta confiabilidade.

Palavras-chave: Rede neural artificial. Aprendizagem por reforço. Mercado de ações. Bolsa de Valores.

SCHEREMETA, L. J. P. Desenvolvimento e avaliação de rede neural aplicada em operações em bolsa de valores. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) — Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2021.

ABSTRACT

Since the 1980s, with the technological advancement of the financial market, they have sought, through mathematical theories, technical analysis, and other mechanisms, to predict the behavior of the stock market. As well as the computational advance, with better processors, internet of things, programming languages and higher data transmission speeds, it allowed the automation of these strategies over time. Since then, research has shown great advances and the applications of neural networks are increasingly part of our day. As an example: search engines on websites, applications, targeted advertising, banks, laboratory analysis, diagnostics, polluters' forecast, climate, among others. The objective of this work is to develop a tool that indicates operations in the stock market, more specifically, in the financial market infrastructure company Brasil Bolsa Balcão (B3), currently the only regulated company in Brazil, with a focus on developing an efficient neural network that carry out transactions, where it is necessary to buy a share and then sell it, in the swing-trade modality. The analyzed period includes operations lasting from days to a few weeks. Thus, the developed tool uses artificial neural network and reinforcement learning techniques to improve decision making. The assets analyzed were B3SA3, BBAS3, PETR4 and VALE3, where the daily closing price for the period from 2015 to 2019 was collected as a training database and the tool was applied in the year 2020. Due to its specificities, it was necessary, then, to create a neural network for each asset, obtaining a result of 79.48% of success rate on average and 1.6 of profit / loss ratio, satisfactory and positive at the end of the period, even experiencing a period of uncertainty and great volatility in the global financial market.

Keywords: Artificial neural network. Reinforcement Learning. Stock market. Stock Exchange.

LISTA DE FIGURAS

Figura 1 - Gráfico de candlestick Ibovespa	18
Figura 2 - Candlestick	18
Figura 3 - Neurônio biológico	19
Figura 4 - Neurônio artificial	20
Figura 5 - Funções de Ativação	21
Figura 6 - Classificação de Redes Neurais Artificiais	22
Figura 7 – Exemplo de rede neural de múltiplas camadas e densa	23
Figura 8- Interação do agente com o ambiente	24
Figura 9 - Navegador Anaconda	26
Figura 10 - Ambiente de desenvolvimento Spyder	27
Figura 11 - Estrutura da política	28
Figura 12 - Comparativo Q-Learning, série temporal e índice KOSPI	30
Figura 13 - Quadro comparativo dos resultados entre a rede neural MLP com a hij	pótese de
Random Walk	34
Figura 17 - Estrutura da rede neural	38
Figura 18 - Funções de ativação	38
Figura 19 - Exemplo de pré-processamento dos vetores de entrada	39
Figura 20 - Resultado financeiro das operações em B3SA3	43
Figura 21 - Resultado financeiro das operações em BBAS3	44
Figura 22 - Resultado financeiro das operações em PETR4	44
Figura 23 - Resultado financeiro das operações em VALES	45

LISTA DE TABELAS

Tabela 1 - Volume financeiro movimentado em ações na B3 por mês em 2020	17
Tabela 2 - Divisão do período em Treino, Validação e Teste e seus respectivos tamanhos .	29
Tabela 3 - Comparativo dos resultados para a série Nasdaq	31
Tabela 4 - Comparativo dos resultados para a série Dow Jones	31
Tabela 5 - Comparativo dos resultados para a série Ibovespa	32
Tabela 6 - Resultados do período de treinamento	41
Tabela 7 - Resultado do período de avaliação	42

LISTA DE ABREVIATURAS E SIGLAS

AR - Aprendizagem por Reforço

B3 - Brasil Bolsa Balcão

BM&F - Bolsa de Mercadorias e Futuros

BOVESPA - Bolsa de Mercadorias de São Paulo

CETIP - Central de Custódia e Liquidação Financeira de Títulos

IA - Inteligência artificial

IBOV - Índice Bovespa

MAPE - Mean Absolut Percentage Error

MCP - Neurônio de McCulloch e Pitts

MDP - Processo Decisório de Markov

MLP - Multilayer Perceptron

MMC - Média Móvel Centrada

MSE - Mean Square Error

RBF - Radial Basis Function

RELU - Rectified Linear Unit

RNA - Rede Neural Artificial

SELIC - Sistema Especial de Liquidação e Custódia

LISTA DE SÍMBOLOS

α	Ação
b_k	Bias
d	Dia
γ	Valor de desconto
η	Taxa de aprendizado
P	Função de transição
Q	Algoritmo Q-Learning
R	Função de retorno
r	Recompensa
S	Conjunto finito de estados
S	Estado
θ	Função de ativação
u_k	Saída do somador
V	Valor do estado
v	Soma do bias e saída do somador
W_i	Peso sináptico
X_i	Entrada de sinal
x	Variável
x_i	Sinal pré-sináptico
у	Saída do neurônio
y_k	Sinal pós-sináptico

Conjunto finito de ações

Α

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	15
2 CONCEITUAÇÃO	16
2.1 Mercado de Ações no Brasil	16
2.2 Análise Técnica	18
2.3 Redes neurais	19
2.4 Aprendizagem por reforço	23
2.4.1 Processos Decisórios de Markov	24
2.4.2 <i>Q-learning</i>	25
2.5 Softwares	26
2.5.1 Anaconda	26
2.5.2 Spyder	27
3 LEVANTAMENTO BIBLIOGRÁFICO	28
3.1 Estrutura de <i>Q-learning</i> multiagente para otimizar sistemas de nego	ciação de ações
	28
3.2 Comparação entre modelos Holt-Winters e redes neurais para previs	são de séries
temporais financeiras	30
3.3 Previsão de séries temporais no mercado financeiro de ações o	com o uso de
rede neural artificial	33
4 MATERIAL E MÉTODOS	35
4.1 Desenvolvimento	36
4.2 Pré-Processamento	38
4.3 Treinamento e aplicação	39
4.4 Análise de dados	40
5 RESULTADOS E DISCUSSÃO	
5.1 Treinamento	41
5.2 Avaliação	
6 CONCLUSÃO	
REFERÊNCIAS	
APÊNDICE A — CÓDIGO DE TREINAMENTO	
APÊNDICE B — CÓDIGO DE EXECUÇÃO	

1 INTRODUÇÃO

A inteligência artificial (IA) está relacionada à capacidade de soluções tecnológicas realizarem atividades de um modo considerado inteligente. IAs também podem "aprender por si mesmas", graças a sistemas de aprendizado que analisam grandes volumes de dados, possibilitando que elas ampliem seus conhecimentos. Para isso, utilizam ferramentas e técnicas de redes neurais artificiais, lógica fuzzy, internet das coisas, algoritmos, sistemas de aprendizado, percepção de ambiente e habilidade de análise para sustentar a tomada de decisão (TOTVS, 2019).

O uso de IA está cada vez mais presente para solucionar problemas cotidianos e associados à Quarta Revolução Industrial, também conhecida como Indústria 4.0. Uma das aplicações do seu uso envolve a utilização de IA e internet das coisas para melhorar a qualidade de vida de idosos em habitações verticais, por exemplo (YOSHIDA; BOTURA JUNIOR, 2018). Outro exemplo, agora aplicado a indústrias, é a inserção de impressoras 3D que utilizam algoritmos inteligentes e sistemas de aprendizado, e que notavelmente já conquistaram grande aceitação e utilização na área de prototipagem rápida (PEREIRA et al., 2020).

O desenvolvimento de pesquisas com Redes Neurais Artificiais (RNAs) tem sido motivado, desde seu começo, pelo reconhecimento de que o cérebro humano processa informações de um modo completamente diferente de um computador digital convencional. O cérebro é um computador altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos como neurônios, de forma a realizar determinados processamentos mais rapidamente do que um computador digital convencional (HAYKIN, 2007). Mesmo com o desenvolvimento de seus conceitos fundamentais na década de 1940, as redes neurais precisaram esperar tanto o avanço tecnológico de alguns equipamentos (melhores computadores, interfaces e processadores) quanto o desenvolvimento de técnicas para a sua aplicação. Sendo assim, os primeiros sistemas funcionais aplicados na sociedade datam do final dos anos 1980 (BRAGA et al., 2007).

A partir de então, as pesquisas em redes neurais têm tido grandes avanços e estão cada vez mais presentes no cotidiano das pessoas, em diversas áreas e com diversas ferramentas: mecanismos de busca de sites, aplicativos, propagandas direcionadas, análise de bancos, análises laboratoriais, exames médicos, diagnósticos, previsão de poluidores, clima, entre outros, inclusive em operações no mercado de ações, mais especificamente na empresa Brasil Bolsa Balcão (B3) que atualmente é a única regulamentada no Brasil.

Em outubro de 2020, um estudo sobre o perfil e o comportamento de mais de 2 milhões de pessoas que iniciaram sua jornada de investimentos na bolsa de valores, entre abril de 2019 e abril de 2020, demonstrou que o número de contas na B3 ultrapassou a marca de 3,2 milhões. De acordo com os dados da pesquisa, o perfil médio da nova safra de investidores é jovem (média de 32 anos), sem filhos (60%), com renda mensal de até R\$ 5 mil (56%) e com trabalho em tempo integral (62%). Apesar de a maioria ainda ser formada por homens (74%), chama atenção o crescimento de mulheres investindo na bolsa (26%), saltando de 179.392 investidoras em 2018 para 809.533 em 2020 (B3, 2020).

O cenário financeiro brasileiro em dezembro de 2020, com taxas de juros determinadas pelo Sistema Especial de Liquidação e Custódia (SELIC) abaixo da inflação, resultou na inviabilidade de investimentos em Poupança, Tesouro Direto e outros atrelados à taxa SELIC. Dessa forma, a busca por maiores riscos de capital, cujos retornos são melhores, tornou-se um grande fator para que pessoas físicas adentrassem no mercado de ações.

Segundo a tendência verificada por pesquisas anteriores sobre o primeiro investimento (aporte inicial) de pessoas físicas (B3, 2020) é de que ele tem caído: nos últimos dois anos (2018–2020), caiu 58% (de R\$ 1.916,00, em outubro de 2018, para R\$ 660,00, em outubro de 2020). Entre os investidores mais jovens, os valores são ainda menores: em outubro de 2020, investidores de 16 a 25 anos de idade usaram R\$ 225,00 como valor médio de investimento inicial (B3, 2020).

Sendo assim, a busca de gerenciamento de riscos num mercado de renda variável é necessária; mesmo sabendo-se, por meio de análises, que ativos de uma empresa (ação) têm a probabilidade de crescer em determinado período, é necessário entender que a ação pode variar de preço negativamente, durante tal período, antes de atingir um objetivo determinado. Além de poder não dar resultados positivos, gerando prejuízos; disso surge a necessidade de o investidor ter um plano para parar ou adaptar-se ao mercado, a fim de evitar frustrações e prejuízos relativos ao capital investido. É comum também que a maioria que se está inserindo no mercado fique receosa ou tome decisões erradas devido à falta de conhecimento, ou veja suas aplicações diminuindo num primeiro instante.

Existem perdas no mercado de ações; nenhuma técnica, robô ou rede neural desenvolvida até então é cem por cento confiável, certeira. No entanto, a automatização pode ser uma alternativa segura e confiável: redes neurais, se usadas corretamente, podem avaliar melhor o risco e o retorno, além de determinarem indicativos de compra e venda de ativos e permitirem a conquista de um acurado gerenciamento de risco. Levando essas necessidades em consideração, propõe-se neste estudo a aplicação de uma rede neural.

1.1 Objetivos

Com o uso de redes neurais artificiais, este trabalho objetiva desenvolver uma ferramenta que realiza operações denominadas "operações de compra", nas quais é necessário comprar uma ação, para posteriormente vendê-la, na modalidade de *swing-trade*.

2 CONCEITUAÇÃO

2.1 Mercado de Ações no Brasil

Bolsa de valores é um clube de corretores de valores. É uma associação sem fins lucrativos que reúne corretores e intermediários no processo de comprar e vender ações. Todos os negócios realizados são públicos; as Bolsas preocupam-se em ser transparentes. Qualquer cidadão pode, por meio da internet, saber qual foi o último negócio realizado com as ações da Petrobras, por exemplo. Os preços e as quantidades negociadas são divulgados instantaneamente para o mundo todo (HALFELD, 2006).

Em 1965, devido ao aumento dos negócios, a figura do Corretor Oficial foi substituída pela Sociedade Corretora. A Bolsa deixava de estar subordinada à Secretaria da Fazenda, passando a ter autonomia administrativa. Em 1967, recebeu o nome de Bolsa de Valores de São Paulo (Bovespa). Juntamente com as Sociedades Corretoras, vieram também os operadores de pregão que, em meio à concorrência e ao aumento dos negócios, desenvolveram os famosos gritos e gestuais que seriam a marca registrada das atividades relacionadas à bolsa. Nessa época, em 1968, a Bovespa lançou o Índice Bovespa, o mais importante indicador de rentabilidade do mercado acionário nacional (ACERVOB3).

Em 1991, ocorreu a fusão com a Bolsa de Mercadorias de São Paulo, passando a ser chamada de Bolsa de Mercadorias & Futuros (BM&F). Na sequência, foi incorporada também a Bolsa Brasileira de Futuros (BBF). Em 8 de maio de 2008, os acionistas da BM&F S.A. e da Bovespa Holding S.A. aprovaram por unanimidade, em assembleia, a fusão das duas empresas. Nascia, então, a Bolsa de Valores, Mercadorias e Futuros (BM&FBOVESPA S.A.), uma das maiores bolsas do mundo em valor de mercado. Em março de 2017, a BM&FBOVESPA SA uniu suas atividades com as desenvolvidas pelo Mercados Organizados (CETIP SA) e passou a operar sob o nome de B3. Atualmente, além de ser a única bolsa de valores, mercadorias e futuros em operação no Brasil, também é a maior depositária de títulos de renda fixa da América Latina e a maior câmara de ativos privados do país (ACERVOB3).

Nos últimos cinco anos, o número de pessoas físicas inseridas no mercado de ações, no Brasil, mais que quintuplicou: foi de aproximadamente 557 mil, em 2015, a 2,483 milhões, em 2020, o que revela grande interesse da população nesse tipo de investimentos; entretanto, esse número ainda está bem abaixo proporcionalmente em comparação a outros países, como os Estados Unidos e alguns países europeus, onde as pessoas já estão inseridas e acostumadas com bolsas de valores há algumas décadas (MEDEIROS, 2020).

Houve um grande aumento do volume financeiro movimentado em ações na B3 (2020), no período de 2020, motivado tanto pela crise causada pelo coronavírus, causador da COVID-19, bem como pelas taxas de juros e inflação baixas (tabela 1).

Tabela 1 - Volume financeiro movimentado em ações na B3 por mês em 2020

mês/2020	R\$/mil
Janeiro	62.772
Fevereiro	70.432
Março	57.823
Abril	110.786
Maio	381.526
Junho	370.820
Julho	528.010
Agosto	1.028.742
Setembro	907.747
Outubro	1.335.334
Novembro	2.039.827
Dezembro	2.077.540

Fonte: B3, 2020.

Ações são títulos negociáveis de renda variável, que representam a menor parcela do capital social da empresa que os emitiu. Por isso, quando alguém compra ações de uma empresa, é como se possuísse pedaços, partes dessa empresa. As empresas precisam de dinheiro para financiar suas compras, ampliar instalações, ampliar negócios etc. A fim de não emprestarem dinheiro dos bancos, cujos juros são altos e desvantajosos, as empresas emitem ações para captar dinheiro sem precisar pagar juros. Para compensar, pagam aos acionistas a participação nos lucros (dividendos). Essa é uma forma de conseguirem dinheiro com baixo custo (CORREIA, 2008).

As ações podem ser negociadas por lotes de 100 unidades, ou pelo mercado fracionário que as vende individualmente. Comprar ou vender são os dois tipos de operações básicas existentes no mercado; embora existam outras possibilidades (por exemplo, vender a descoberto), elas não são o foco deste trabalho. Portanto, foram utilizadas neste trabalho exclusivamente operações de compras, que servem para o *trader* possuir efetivamente as ações e poder vendê-las posteriormente, visando à obtenção de lucros (comprá-las em "baixa" e vendê-las em "alta").

2.2 Análise Técnica

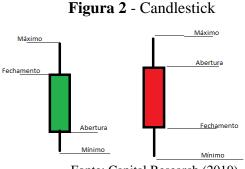
A análise técnica vem desde os primórdios das bolsas, quando já foram criadas teorias que pudessem medir, ou até mesmo prever com boa probabilidade de acerto, a projeção dos ativos. Inicialmente, era feita por meio de *Tape Reading*, dado que as informações de negociações eram emitidas em uma fita. Posteriormente, graças ao desenvolvimento tecnológico, gráficos com grande quantidade de informações passaram a ser plotados, e foram desenvolvidos indicadores para facilitar a leitura e o entendimento desses gráficos. O gráfico de *candlestick* (velas) é um dos exemplos (figura 1).



Figura 1 - Gráfico de candlestick Ibovespa

Fonte: Autoria própria.

O *candle*, ou vela, é um sinal gráfico que compreende um período determinado. No exemplo acima, é o período de um dia. É composto pelo valor de abertura e de fechamento, e pelo valor máximo e mínimo, que são preenchidos (figura 2). O preenchimento verde indica que o preço da ação foi positivo e o vermelho indica que foi negativo.



Fonte: Capital Research (2019).

Cada informação contida nesse gráfico é valiosa para determinar se é um bom momento para comprar ações (compra) ou vendê-las (venda), ou não fazer nenhuma das operações (nada).

A junção do conteúdo de um *candle* com as informações gráficas (por exemplo, tendência, topos e fundos, suporte e resistência) cria uma série temporal na qual é possível aplicar as redes neurais artificiais, obtendo ações que maximizem o lucro, objetivo deste trabalho.

2.3 Redes neurais

O primeiro modelo artificial de um neurônio biológico foi fruto do trabalho pioneiro de McCulloch e Pitts (1943). O primeiro, psiquiatra e neuroanatomista, dedicou 20 anos à tentativa de representar um evento no sistema nervoso; o segundo, matemático recém-graduado, juntouse a McCulloch em 1942 (BRAGA et al., 2007). No trabalho publicado em 1943, "A Logical Calculus of the Ideas Immament in Nervous Activity", é apresentada uma discussão sofisticada de redes lógicas de nodos e novas ideias sobre máquinas de estados finitos, elementos de decisão de limiar lineares e representações lógicas de várias formas de comportamento e de memória (BRAGA et al, 2007; MCCULLOCH; PITTS, 1943).

Há semelhança estrutural entre um neurônio biológico (figura 3) e um neurônio artificial de McCulloch e Pitts (MCP) (figura 4). No caso do neurônio biológico, tem-se o corpo celular como responsável por processar as informações recebidas de outros neurônios pelos dendritos; após o processamento, a informação é passada pelos axônios. No caso do neurônio artificial, por sua vez, tem-se as sinapses compostas pelas entradas de informação Xi e Wki (peso sináptico); o processamento propriamente dito, que soma os sinais de entrada (representado pelo somatório); e por fim, a função de ativação (θ), que determina a resposta a ser enviada para a saída y (HAYKIN, 2007).

Dendritos (terminal de recepção)

Terminal do Axônio
(terminal de transmissão)

Sentido de propagação

Corpo

Axônio

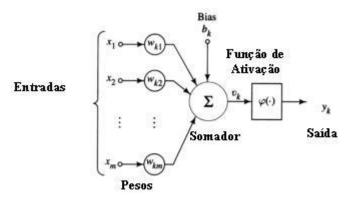
Nodo de Ranvier

Bainha de Mielina

Figura 3 - Neurônio biológico

Fonte: Canedo (2016).

Figura 4 - Neurônio artificial



Fonte: HAYKIN, (2007).

Na equação 1, o funcionamento do neurônio MCP, onde: u_k é a saída do somador e b_k é o bias, que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação (HAYKIN, 2007).

$$u_k = \sum_{i=1}^n x_i w_i \ge \theta \tag{1}$$

$$y_k = \theta (u_k + b_k) \tag{2}$$

Hebb (1949) melhorou o MCP, ao criar a primeira regra de aprendizagem descrita na equação 3:

$$\Delta w_{ki}(n) = \eta y_k(n) x_i(n) \tag{3}$$

Onde: η é uma constante positiva que determina a taxa de aprendizado, x_i representa o sinal pré-sináptico, y_k representa o sinal pós-sináptico do neurônio k na iteração do algoritmo de aprendizado e w_{ki} representa o peso sináptico que faz a conexão entre o sinal de entrada x_i e o neurônio k (HAYKIN, 2007).

Há anos, o conceito de neurônios e redes neurais de múltiplas camadas vem evoluindo.

A função de ativação, representada por θ (v), define a saída y de um neurônio, onde: v é a soma do *bias* b mais a saída u. Existem vários tipos de função de ativação (HAYKIN, 2007), e para exemplificá-los são trazidos três tipos básicos descritos nas seguintes equações (4, 5 e 6) e ilustrados (figura 5).

Função de Limiar:

$$\theta(v) = \begin{cases} 1, & v \ge 0 \\ 0, & v < 0 \end{cases} \tag{4}$$

Função Linear por Partes:

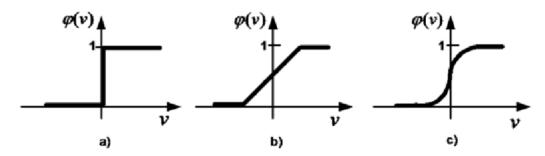
$$\theta(v) = \begin{cases} 1, & v \ge 0.5 \\ v, & -0.5 < v < 0.5 \\ 0, & v \le -0.5 \end{cases}$$
 (5)

Função Sigmoide:

$$\theta(v) = \frac{1}{1 + e^{-xv}} \tag{6}$$

Tem-se como parâmetro de inclinação a variável.

Figura 5 - Funções de Ativação



As funções apresentadas são: a) Limiar, b) Linear por Partes e c) Sigmoide. Fonte: Ribeiro, Goldschmidt e Choren (2020).

A definição da arquitetura de uma RNA é um parâmetro importante na sua concepção, uma vez que ela restringe o tipo de problema que pode ser tratado pela rede. Redes com uma camada única de nodos MCP, por exemplo, só conseguem resolver problemas linearmente separáveis. Redes recorrentes, por sua vez, são mais apropriadas para resolver problemas que envolvem processamento temporal. Fazem parte da definição da arquitetura os seguintes parâmetros: número de camadas da rede, número de nodos em cada camada, tipo de conexão entre os nodos e topologia da rede (BRAGA et al., 2007).

Alguns exemplos são dados a seguir:

- Redes de camada única: só existe um nó entre qualquer entrada e saída da rede (figura 6 "a" e "e");
- Redes de múltiplas camadas: existe mais de um neurônio entre alguma entrada e saída da rede (figura 6 "b", "c" e "d");
- Nodos acíclicos: a saída de um neurônio da *i*-ésima camada da rede não pode ser usada como entrada de camadas ou índices menores que *i* (figura 6 "a", "b" e "c");
- Nodos cíclicos: a saída de um neurônio na *i*-ésima camada pode ser utilizado como entrada de camadas cujos índices são menores ou iguais a *i* (figura 6 "d" e "e");
 - Rede com conexões fracas ou parcialmente (figura 6 "b", "c" e "d");
 - Redes completamente conectadas (figura 6 "a" e "e").

Figura 6 - Classificação de Redes Neurais Artificiais

Fonte: Braga et al. (2007).

Seguindo os exemplos anteriores, foi utilizado para esse trabalho uma rede neural de múltiplas camadas (figura 6 "b", "c" e "d") e a rede densa, ou seja, completamente conectada (figura 6 "a" e "e") chegando ao exemplo mostrado a seguir (figura 7)

Figura 7 – Exemplo de rede neural de múltiplas camadas e densa

Fonte: Autoria própria.

2.4 Aprendizagem por reforço

A aprendizagem por reforço é um problema de aprendizagem baseado em "comportamento": é realizada por meio da interação entre o sistema de aprendizagem e o seu ambiente, dentro do qual o sistema procura alcançar um objetivo específico, apesar da presença de incertezas (BARTO et al., 1983; SUTTON; BARTO, 1998). O fato de que essa interação é realizada sem a necessidade de um professor torna a aprendizagem por reforço particularmente atrativa para situações dinâmicas em que é custoso ou difícil, se não impossível, reunir um conjunto satisfatório de exemplos de entrada-saída (HAYKIN, 2007).

Existem duas abordagens para estudar a aprendizagem por reforço; neste trabalho, apenas a abordagem moderna foi usada. Fundamenta-se em uma técnica matemática conhecida como programação dinâmica, que serve para decidir o curso da ação considerando estágios futuros possíveis, sem realmente experimentá-los, com ênfase no planejamento. Segundo a terminologia adotada por Bertsekas e Tsitsiklis (1996), a abordagem moderna de aprendizagem por reforço é referida como programação neurodinâmica. Logo, a fundamentação teórica é dada pela programação dinâmica, ao passo que a capacidade de aprendizagem é dada pelas redes neurais (HAYKIN, 2007).

Programação neurodinâmica pode ser definida sucintamente como aquela que "permite que um sistema aprenda a tomar decisões observando o seu próprio comportamento e a melhorar as suas ações através de reforço, utilizando um mecanismo incorporado" (BERTSEKAS; TSITSIKLIS, 1996, p. 8). Consiste em um sistema de aprendizagem ou em um agente que interage com seu ambiente (figura 8).

Ambiente Ação

Reforço Estado

Agente

Figura 8- Interação do agente com o ambiente

Fonte: Autoria própria.

O ambiente evolui probabilisticamente e ocupa um conjunto finito de estados discretos; entretanto, o estado não contém estatísticas passadas, mesmo que elas pudessem ser úteis para o agente. Para cada estado do ambiente, há um conjunto finito de ações possíveis que podem ser realizadas pelo agente. Toda vez que o agente realiza uma ação, incorre em um certo custo, o reforço. A observação dos estados, a realização de ações e a incorrência de reforço ocorrem em tempo discreto (HAYKIN, 2007).

A técnica de programação dinâmica fundamenta-se em uma ideia muito simples, conhecida como o princípio da otimização de Bellman (1957). Expresso de uma forma sucinta, o princípio diz: "Uma política ótima tem a propriedade que, quaisquer que sejam o estado inicial e a decisão inicial, as decisões restantes devem construir uma política ótima em relação ao estado resultante da primeira decisão" (BELLMAN; DREYFUS, 1962, p. 15).

Uma política é definida como um mapeamento de estados para ações, ou seja, é uma regra usada pelo sistema de aprendizagem para decidir o que fazer, dado o conhecimento atual do ambiente.

2.4.1 Processos Decisórios de Markov

A técnica de aprendizagem por reforço (AR) que satisfaz a propriedade de Markov é chamada de Processo Decisório de Markov (MDP), e é definida por um conjunto (*S*; *A*; *P*; *R*), onde temos:

- S: conjunto finito de estados do sistema;
- *A*: conjunto finito de ações;

- \bullet P: $S \times A$: função de transição que mapeia os pares estado-ação em uma distribuição de probabilidades sobre o conjunto de estados;
- $R: S \times A$: função de retorno ou recompensa pela escolha de uma determinada ação $a \in A$ no estado $s \in S$.

Assim, o estado s e a ação a atuais informam sobre o próximo estado s' de acordo com a probabilidade P(s'|s,a) e com a recompensa associada r(s,a). Intuitivamente, um processo é Markoviano exclusivamente se o estado de transições depender apenas do estado corrente do sistema e for independente de todos os estados anteriores (NARAYANAN; JENNINGS, 2005).

2.4.2 Q-learning

O algoritmo Q-learning é um MDP, o que implica a necessidade de constante atualização, por meio de recompensas esperadas, Q(s,a). A cada interação com o ambiente, os valores de Q são atualizados (equação 7).

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma eQ(s')] \tag{7}$$

Onde: γ é o valor de desconto utilizado para garantir o número finito para valores de Q e α é a constante de aprendizagem, que pode ser $0 \le \gamma < 1$ e $0 < \alpha \le 1$. Após a execução da ação α , o algoritmo sai do estado s, entra no estado s' e deve receber uma recompensa r, conforme o resultado obtido. Em s', é feita uma busca pela ação α' que apresente o maior valor de retorno esperado, representado por $eQ(s') = max_{\alpha'}Q(s', \alpha')$ (WATKINS; DAYAN 1992).

Para este trabalho foi utilizada uma versão simplificada da equação de Bellman:

$$V(s) = \max_{a} (R(s, a) + \gamma V(s'))$$
 (8)

Onde: s é o estado, a a ação, a é a recompensa, a é o fator de desconto e a0 é o valor do estado. Valor do estado a0 estado valor máximo de uma ação dentro das possíveis ações, sendo os valores o resultado do somatório da recompensa a0 de uma ação a0 em um estado a0 com a multiplicação do fator de desconto a0 estado seguinte a0 (GRANATYR, 2019).

2.5 Softwares

Para o desenvolvimento deste trabalho foram utilizados os Softwares Anaconda 3 e Spyder 4.1.5, já que o primeiro é ambiente de distribuição Python e o segundo de desenvolvimento. Ambos os softwares são gratuitos e de código aberto.

Foram utilizadas várias bibliotecas de desenvolvimento, como principal biblioteca para criar e desenvolver *machine learning* foi utilizada o TensorFlow, desenvolvido pelo Google, na sua versão 2.0.0.

2.5.1 Anaconda

O Anaconda Individual Edition é a plataforma de distribuição Python mais popular, com mais de 20 milhões de usuários no mundo todo, e é a preferida por aqueles que trabalham com ciência de dados Python. Além disso, é uma solução flexível e de código aberto que fornece utilitários para construir, distribuir, instalar, atualizar e gerenciar softwares em uma plataforma cruzada. O Anaconda facilita o gerenciamento de vários ambientes de dados, que podem ser mantidos e executados separadamente, sem que ocorra interferência de uns nos outros (ANACONDA, 2020). A interface é mostrada na figura 9.

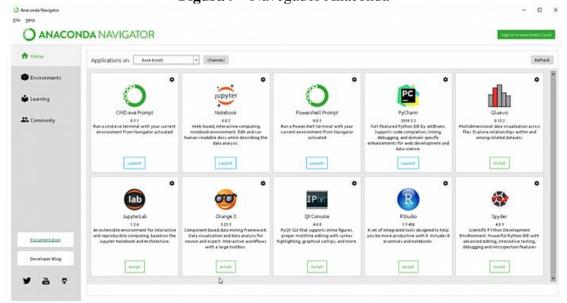
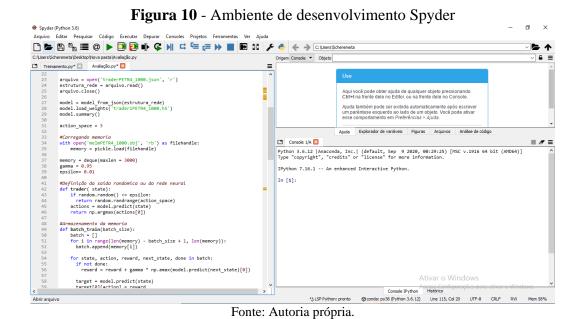


Figura 9 - Navegador Anaconda

Fonte: Anaconda (2020).

2.5.2 *Spyder*

Spyder é um ambiente científico gratuito e de código aberto escrito em Python, para Python e projetado por cientistas, engenheiros e analistas de dados para eles usarem. Apresenta uma combinação única de edição avançada, análise, depuração e funcionalidade de criação de perfil de uma ferramenta de desenvolvimento abrangente com a exploração de dados, execução interativa, inspeção profunda e belas capacidades de visualização de um pacote científico (SPYDER, 2020). A interface é mostrada na figura 10.



3 LEVANTAMENTO BIBLIOGRÁFICO

Para o desenvolvimento deste trabalho, foram utilizados como base artigos científicos que demonstram o interesse na área de desenvolvimento e eficácia de redes neurais e na aprendizagem por reforço para aplicação em mercado financeiro. Há uma ampla abordagem sobre o assunto, de forma que nem todos os artigos seguem a mesma linha de desenvolvimento.

3.1 Estrutura de *Q-learning* multiagente para otimizar sistemas de negociação de ações

O artigo de Lee e Jangmin (2002) foi selecionado como base, pois trata de uma aplicação de aprendizagem por reforço com multiagentes, cada qual com uma função distinta, os quais e trabalham em conjunto para analisar a política e obter resultados favoráveis.

Lee e Jangmin (2002) apresentam uma estrutura de aprendizagem de reforço para sistemas de negociação de ações. Os parâmetros do sistema de negociação são otimizados pelo algoritmo *Q-learning*, e as redes neurais são adotadas para aproximar o valor. Nesta estrutura (figura 11), vários agentes cooperativos são usados para integrar de forma eficiente a previsão de tendência global e a estratégia de comércio local para obter melhor desempenho comercial. Agentes comunicam-se e compartilham uns com os outros episódios de treinamento e políticas aprendidas, mantendo o esquema de *Q-learning* convencional (LEE; JANGMIN, 2002).

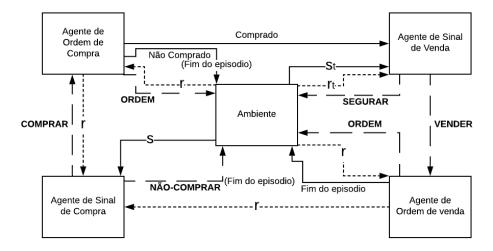


Figura 11 - Estrutura da política

Fonte: Lee e Jangmin (2002). (Adaptado)

O agente sinalizador de venda difere do agente sinalizador de compra, porque também engloba a política, bem como a previsão de venda. Os agentes de ordens de compra/venda não

realizam predições, dado que seu único objetivo é fornecer uma política para uma ótima execução de ordens para aumentar a taxa de sucesso das negociações (LEE; JANGMIN, 2002).

Os autores adotaram uma matriz binária, a estrutura do ponto de inflexão, resumindo as dependências de longo prazo com base nos pontos de inflexão superior e inferior das médias móveis de cinco dias. Além dessas estruturas temporais, o agente de sinal de compra compartilha os indicadores técnicos de curto prazo com os agentes de pedido. O estado do agente de sinal de compra consiste em 353 bits binários; o agente de sinal de venda, por sua vez, tem alguns bits a mais que representam a taxa de lucro atual durante um período de manutenção de uma ação. Limitaram arbitrariamente a faixa de lucro: portanto, se o lucro subir acima de + 30% ou cair abaixo de -20%, o agente sinalizador de venda é compelido a vender as ações à força (LEE; JANGMIN, 2002).

Os agentes de sinais podem realizar apenas dois tipos de ações. O agente de sinalização de compra pega *NÃO-COMPRAR* ou *COMPRAR*, e o agente de sinalização de venda pega *SEGURAR* ou *VENDER*. O cálculo da recompensa (r) é o seguinte: o agente de sinal de compra recebe recompensa zero enquanto recebe *NÃO COMPRA*. Se for *COMPRAR*, o cálculo de sua recompensa é adiado (rt) até que a venda do estoque seja feita pelo agente da ordem de venda. A taxa de variação diária dos preços das ações é fornecida enquanto o agente de sinal de venda assume *SEGURAR*. Mas quando leva *VENDA*, zero pode subir acima de +30% ou cair abaixo de -20%, o agente sinalizador de venda é compelido a vender as ações à força. Bits adicionais para o agente sinalizador de venda para representar o lucro atual, que é 100 × (preço de fechamento do dia atual - preço de compra) / recompensa do preço de compra, o sinal significa que ele saiu do mercado (LEE; JANGMIN, 2002).

A fins comparativos, testaram uma rede neural com duas camadas 40x20 aplicando séries temporais no mesmo período. Após o período de treino, testes e avaliação (tabela 2), foi gerado o comparativo, em que o lucro acumulado do sistema de aprendizagem por reforço é representado pela linha contínua, a rede neural com séries temporais é representada pela linha tracejada e o índice da bolsa coreana (KOSPI) por pontos (figura 11).

Tabela 2 - Divisão do período em Treino, Validação e Teste e seus respectivos tamanhos

Partição	Período	Tamanho
Conjunto de treinamento	Jan-1999 a Dez-2000	32019
Conjunto de validação	Jan-2001 a Maio-2001	6102
Conjunto de teste	Jun-2001 a Out-2001	6213

Fonte: Lee e Jangmin (2002, traduzido pelo autor).

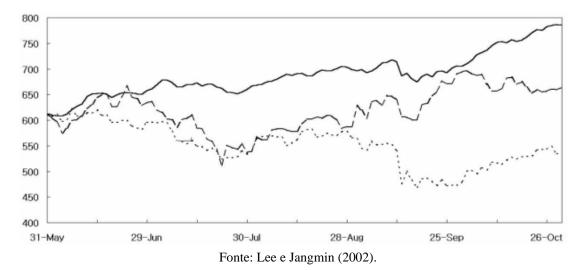


Figura 12 - Comparativo Q-Learning, série temporal e índice KOSPI

Os autores concluem que o sistema de aprendizagem por reforço é mais eficiente se comparado a um sistema tradicional de série temporal, porém deve-se considerar o tempo de amostragem e ter considerações realistas adicionais, como o número de carteiras, a distribuição do ativo para cada carteira e a adaptação da tendência do mercado de ações. Embora a aprendizagem por reforço seja promissora, a necessidade de incorporar as considerações supracitadas tornam o problema mais complexo.

3.2 Comparação entre modelos Holt-Winters e redes neurais para previsão de séries temporais financeiras

O trabalho de Kirsten (2009) foi utilizado como base, principalmente devido à análise comparativa de diversos métodos de previsão de séries temporais. Por meio das observações ao longo do estudo, conclui que cada método pode variar sua eficácia conforme a série temporal e o período analisado, contendo ou não crise, mudanças bruscas, entre outros.

Nesse trabalho foram utilizadas três séries temporais financeiras e um sistema caótico para servir de *benchmark* com outros trabalhos. O Sistema Mackey-Glass foi escolhido para servir de *benchmark* devido sua utilização para o mesmo fim em outros trabalhos. Como base de dados, foram utilizadas as seguintes séries financeiras: o índice Ibovespa, o índice Dow Jones e o índice Nasdaq. Todas durante o mesmo período (de 1 de janeiro de 1995 até 31 de dezembro de 2008). O desempenho dos modelos estudados foi analisado pelos índices de desempenho *Mean Squared Error* (MSE, Erro Quadrático Médio), *Mean Absolute Percentage Error* (MAPE, Erro Percentual Médio Absoluto) e coeficiente R^2 (KIRSTEN, 2009).

Os modelos de previsão lineares estudados foram o de média móvel centrada (MMC) com índice de sazonalidade e o Holt-Winters. Enquanto o primeiro modelo é mais simples, o segundo modelo é mais robusto e, consequentemente, foi mais utilizado em outros estudos. Quanto aos modelos não lineares, dois modelos de redes neurais foram escolhidos, a Rede Neural *MultiLayer Perceptron* (RN-MLP) e a Rede Neural *Radial Basis Function* (RN-RBF). (KIRSTEN, 2009).

Tabela 3 - Comparativo dos resultados para a série Nasdaq

Índice de desempenho	MMC com índice de saz.	Holt-Winters	RN-MLP	RN-RBF	
	Estimação				
MSE	-		1796,83	1895,60	
MAPE			1,22590%	1,23110%	
R^2			0,99712	0,99697	
	Validação				
MSE	458,50	453,59	448,45	457,40	
MAPE	0,68805%	0,68990%	0,67599%	0,68530%	
R^2	0,99993	0,99993	0,99993	0,99993	
	Teste				
MSE	2054,64	2081,37	2052,35	2060,65	
MAPE	1,60710%	1,62590%	1,59980%	1,60930%	
R^2	0,98513	0,98493	0,98514	0,98508	

Fonte: Kirsten (2009, grifo do autor).

Tabela 4 - Comparativo dos resultados para a série Dow Jones

Índice de desempenho	Holf-Winters		RN-MLP	RN-RBF			
Estimação							
MSE			9388,04	9543,65			
MAPE			0,77159%	0,78137%			
R^2			0,99781	0,99777			
	Validação						
MSE	7476,20	7334,74	9390,53	7468,81			
MAPE	0,50123%	0,50302%	0,60935%	0,50233%			
R^2	0,99994	0,99994	0,99992	0,99994			
	Teste						
MSE	44932,00	44779,36	46305,46	44901,99			
MAPE	1,40340%	1,4165%	1,4431%	1,40390%			
R^2	0,98362	0,98367	0,98312	0,98363			

Fonte: Kirsten (2009, grifo do autor).

Tabela 5 - Comparativo dos resultados para a série Ibovespa

Índice de desempenho			RN-MLP	RN-RBF
	Estimação			
MSE			83178,33	83908,26
MAPE			1,6818%	1,6941%
R^2			0,99850	0,99849
	Validação			
MSE	427324,18	423746,24	455112,73	425509,17
MAPE	1,1586%	1,1480%	1,2059%	1,1558%
R^2	0,99994	0,99994	0,99994	0,99994
	Teste			
MSE	2057891,19	2064644,51	2406968,44	2060447,57
MAPE	2,0992%	2,1094%	2,3217%	2,0981%
R^2	0,98119	0,98113	0,978	0,98117

Fonte: Kirsten (2009, grifo do autor).

Os métodos, tanto lineares quanto não lineares, alternaram-se como os melhores métodos entre uma e outra série analisada. Considerando somente as séries financeiras e as fases de validação e teste, o modelo RN-MLP obteve o melhor desempenho para a série Nasdaq; os modelos MMC com índice de sazonalidade e Holt-Winters dividiram a liderança da série Dow Jones; e, por fim, os modelos lineares suplantaram os modelos não lineares na série Ibovespa (KIRSTEN, 2009).

O modelo RN-MLP foi, comparativamente com os outros modelos estudados, melhor para a série Nasdaq, mediano para a série Dow Jones e ruim para a série Ibovespa. Uma possível resposta para esse comportamento é que as amostras da fase de teste fazem parte de uma crise, na qual os preços mudam bruscamente, e que apenas a série Nasdaq, durante a fase de estimação, apresenta-se fortemente afetada por algum tipo de crise. Dessa forma, a RN-MLP aprendeu com a crise durante a fase de estimação e apresentou um desempenho melhor na fase de teste, na qual outra crise estava em andamento (KIRSTEN, 2009).

Essa mesma linha também pode explicar o motivo pelo qual um método linear simples, como o MMC com índice de sazonalidade, obtém melhor desempenho comparado a um método não linear mais complexo, como foi o caso da série Ibovespa. Como a primeira crise praticamente não afetou esta série, nenhum dos modelos mais complexos conseguiu prever com mais acurácia o que estava acontecendo na fase de teste (KIRSTEN, 2009).

3.3 Previsão de séries temporais no mercado financeiro de ações com o uso de rede neural artificial

O trabalho de Carvalho (2018) foi utilizado como base, pois trata da aplicação de redes neurais com séries temporais e analisa dois períodos distintos do mercado brasileiro — a crise de 2008 e o período de recessão econômica a partir de 2015. Chega à conclusão de que o modelo desenvolvido foi eficiente para análises de curto prazo.

O trabalho visa a estudar a previsão de séries temporais com o uso dos dados obtidos da B3, tendo como base os valores das ações no fechamento do pregão. Para a previsão foi utilizada uma RNA com arquitetura MLP. Foi mostrado através desse estudo de previsão do mercado financeiro como a rede neural se comporta e como ela pode ser de grande valia para previsões com séries de dados temporais (CARVALHO, 2018).

Para o processamento dos experimentos ele selecionou dois lotes de dados da ação PETR4, sendo o primeiro lote continha 2.223 registros do período de 02/01/2000 a 30/12/2009, que foi tratado como cenário 1, e outro lote continha 1973 registros do período de 04/01/2010 a 28/12/2017, que foi tratado como cenário 2 (CARVALHO, 2018).

No cenário 1, os experimentos de 1 a 8 usam 5 neurônios e taxa de aprendizado igual a 0,8; já os experimentos de 9 a 16, por sua vez, usam 10 neurônios e taxa de aprendizado igual a 0,8. No cenário 2, os experimentos de 1 a 8 usam 5 neurônios e taxa de aprendizado igual a 0,7; os experimentos de 9 a 16 usam 10 neurônios e taxa de aprendizado igual a 0,7. Por meio da análise do comportamento dos resultados do processo preditivo na hipótese de Random Walk, percebeu-se que em ambos os cenários os resultados da predição são muito próximos dos valores observados, o que tornou o modelo recomendável para predição (CARVALHO, 2018). Seus resultados estão descritos na figura 13.

Analisando-se os resultados obtidos depreende-se que ambos os processos são demasiadamente bons para a previsão de séries temporais, acompanhando a não linearidade do comportamento da série e trazendo resultados muito próximos do observado. Não se pode em razão das diferenças comparativas observadas excluir a rede neural desse processo preditivo, pois o resultado individual também é importante (CARVALHO, 2018).

Salvaguardada a volatilidade dos preços das ações no mercado financeiro, que por vezes sofrem variações bruscas, esse método de previsão (com rede neural ou hipótese de Random Walk) pode servir para orientar investimentos; no entanto, recomenda-se cautela para investimentos de médio e longo prazo (CARVALHO, 2018).

Figura 13 - Quadro comparativo dos resultados entre a rede neural MLP com a hipótese de Random Walk

			Resultados			
		Cenário 1			Cenário 2	
Experimento	Interações	Erro RMSE - MLP	Erro RMSE - RW	Interações	Erro RMSE - MLP	Erro RMSE - RW
1	1230	0,057659278	0,025535941	660	0,114504915	0,019249829
2	1760	0,026703787	0,02424997	1360	0,133119741	0,019165655
3	1370	0,051365495	0,025742461	870	0,163654717	0,018089722
4	880	0,035786861	0,024461559	970	0,037707915	0,019378211
5	750	0,033425514	0,025251587	950	0,116923002	0,019327209
6	1150	0,057466288	0,024541882	650	0,039374456	0,018482957
7	910	0,069608328	0,024651951	200	0,091490599	0,018824853
8	730	0,043099599	0,024588257	1410	0,032347052	0,018327835
9	1020	0,091900937	0,02372751	1170	0,04095351	0,018655488
10	2110	0,027339725	0,024984203	820	0,075487526	0,018555133
11	1840	0,028145167	0,026149424	1420	0,100892976	0,019149326
12	1300	0,0423	0,025489412	260	0,106325683	0,018525667
13	890	0,078071939	0,024814822	490	0,027002221	0,019122994
14	1840	0,059189989	0,024289655	1020	0,05988059	0,019128395
15	1340	0,058555567	0,025196787	750	0,039719557	0,019082994
16	1180	0,03745168	0,025054243	540	0,032788732	0,019627004
Média Desvio Padrão Percentual		0,049879385 0,01910817 4,99%	0,024920604 0,000627941 2,49%		0,075760825 0,042947721 7,58%	0,018918329 0,000432609 1,89%

Fonte: Carvalho (2018).

4 MATERIAL E MÉTODOS

Conforme os conceitos visitados (seção 2), o funcionamento dos softwares descritos anteriormente (seção 2.5) e os artigos que embasaram este trabalho (seção 3), foi desenvolvida uma ferramenta que dispunha de um sistema de rede neural, na qual o aprendizado por reforço foi aplicado como forma de melhorar o treinamento da rede para a tomada de decisão. Para isso, foram utilizados diferentes aspectos na preparação da ferramenta.

O comportamento dos preços de uma ação foi analisando o periodo diário com os preços de fechamento, o que gerou uma série temporal (figura 14), cuja variação do preço de fechamento do dia é melhor observada no gráfico de linha (figura 15).



Figura 14 - Gráfico de Candlestick diário de PETR4



Fonte: Autoria própria.

A escolha da linguagem em Python foi determinada por ser uma linguagem de facil utilização com relação a computação quantitativa e analítica e suportar programação estruturada, codificação orientada a objetos. Sendo uma linguagem estremamente funcional, lider do setor há algum tempo e é utilizada em grandes setores e empresas. Python é uma linguagem flexivel e de código aberto, sendo assim o desenvolvimento de bibliotecas e ferramentas que se utilizam de sua liguangem é amplamente desenvolvida e atualizada.

Sendo assim, a escolha do Anaconda como plataforma de distribuição de Python foi uma escolha natural, tendo em vista a disponibilidade de ambientes de desenvolvimento, ferramentas e bibliotecas disponíveis, sendo tambem uma plataforma amplamente divulgada e utilizada. Dentre os ambientes de desenvolvimento, o Spyder foi escolhido pois permitia a total execução do projeto, premitindo executar, salvar e alterar a programação por partes, ambiente offiline, aplicabilidade das principais bibliotecas voltadas para *MachineLearning* e *DeepLearning* como Tensorflow e Keras, assim como permitindo uma boa análise e avaliação de resultados.

4.1 Desenvolvimento

Tem-se como objetivo obter lucro com operações de compra e venda de ações, primeiramente ao comprar uma ação da empresa, e posterioremente ao vendê-la com possível lucro. Então, o sistema desenvolvido pode executar uma das três operações: Compra, Venda ou Nada, decisão que é tomada pelos neurônios da camada de saída, que seleciona o maior valor para execução.

Para o desenvolvimento da ferramenta foi priorizado o uso de uma rede neural com o aprendizado por reforço como complemento, foram levados em consideração os seguintes requisitos:

- Analizar um periodo de 10 dias;
- Operar na Compra quando viavel;
- Se comprado: Vender objetivando Lucro;
- Fazer Nada, quando não for favoravel;
- Armazenar em memória;
- Aplicar a técnica de Aprendizado por Reforço.

Como visto nos trabalhos (seção 3), foi considerado a utilização de uma rede neural do tipo Multilayer Perceptron, pois além de ser um dos tipos mais comuns e simples de serem utilizadas, é necessario um volume de neurônios, que quando rearranjados nessa formatação

melhoram o processamento que é necessario para as operações definidas nesse trabalho. E utilizado camadas densas para interligar todos os neuronios de uma camada a outra, para sempre levar em consideração toda informação processada.

Para a rede neural deste trabalho, foi proposta uma MLP sequencial, com uma camada de entrada com dez neurônios (uma camada para cada dia), duas camadas ocultas (uma com 20 e outra com 40 neurônios, respectivamente) e uma camada de saída com três neurônios (figura16). Todas as camadas eram densas, ou seja, todos os neurônios de uma camada estavam interligados com cada neurônio da camada seguinte, visto a estrutura final na programação (figura 17).

O número de neuronios de entrada é baseado em que para a analise técnica os primeiros sinais de formação de direcionamento de mercado se da em um periodo minimo de 7 dias, como a bolsa de valores no Brasil só funciona de segunda a sexta, uma semana são apenas 5 dias de operação, portanto foram levados em consideração 10 dias para analise o equivanlente a duas semanas e que emgloba o periodo minimo de 7 dias.

ENTRADA CAMADAS OCULTAS SAIDA

1
1
2
2
2
2
3

Figura 16- Configuração da rede neural

Fonte: Autoria própria.

Figura 14 - Estrutura da rede neural

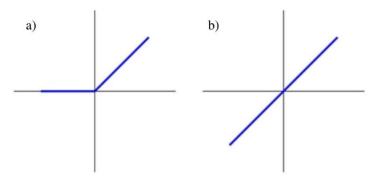
Model: "sequential"				
Layer (type)	Output Shape	Param #		
dense (Dense)	(None, 20)	220		
dense_1 (Dense)	(None, 40)	840		
dense_2 (Dense)	(None, 3)	123		
Total params: 1,183 Trainable params: 1,183 Non-trainable params: 0				

Fonte: Autoria própria.

Os valores do de entrada foi selecionado o preço de fechamento do periodo diário pois repesenta o consenso entre compradores e vendedores, de quanto uma ação valia ao final do dia de pregão, sendo assim o valor mais importante a ser analisado no dia.

Para as camadas de entrada e as camadas ocultas foram utilizadas funções de ativação do tipo ReLU (figura 18 "a"), e para a camada de saída foi utilizada a função de ativação do tipo Linear (figura 18 "b").

Figura 15 - Funções de ativação



Funções de ativação a) ReLU e b) Linear. Fonte: Autoria própria.

4.2 Pré-Processamento

O espaço de estados foi obtido a partir do cálculo do tratamento dos valores ao longo de dez dias (equação 9) e da aplicação de uma função sigmoide na diferença dos valores do dia d e do dia d+1(equação 10).

$$x = \frac{(Valor_{d+1} - Valor_d) * 100}{Valor_{d+1}}$$
(9)

$$f(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

Com isso, os vetores de entrada e saída ficaram conforme o exemplo a seguir (figura19).

Figura 16 - Exemplo de pré-processamento dos vetores de entrada

```
In [2]: entrada
Out[2]:
Date
2020-01-30
             28.940001
2020-01-31 28.450001
2020-02-03 28.180000
2020-02-04
            28.629999
2020-02-05
            28.389999
2020-02-06
            29.180000
2020-02-07
            28.930000
2020-02-10 29.129999
            29.480000
2020-02-11
            30.129999
2020-02-12
2020-02-13 29.719999
Name: Close, dtype: float64
In [3]: saida[0]
Out[3]:
array([[0.1515727 , 0.27725317, 0.82803635, 0.30040557, 0.93745819,
       0.29647201, 0.66520425, 0.76624831, 0.89635055, 0.20108258]])
```

Fonte: Autoria própria.

4.3 Treinamento e aplicação

Para o treinamento da rede neural, além de um modelo convencional de *backpropagation*, foi utilizado também um sistema de aprendizagem por reforço, utilizando-se da simplificação da equação 8 de Bellman (GRANATYR, 2019) e de uma memória de 3.000 eventos. As etapas e os processos de criação da rede neural, pré-processamento, aprendizado e treinamento são descritos no apêndice A.

As recompensas foram aplicadas da seguinte forma: o sistema recebia o valor de lucro quando havia lucro, e recebia zero quando não realizava nenhuma ação ou obtinha prejuízo.

Como base de dados foram utilizados os dados históricos das seguintes ações: 1) VALE3 (Vale), 2) PETR4 (Petrobras), 3) BBAS3 (Banco do Brasil) e 4) B3SA3 (B3), considerando o período de treinamento (de 01 de janeiro de 2015 a 01 de janeiro de 2020) e o período de avaliação (de 01 de janeiro de 2020 a 01 de janeiro de 2021). Os dados foram obtidos no site YahooFinances (2020), e o Spyder (software de desenvolvimento) extraía esses dados online.

A fim de auxiliar a tomada de decisão, foi utilizada a variável randômica epsilon durante

o treinamento, de maneira que sua frequência de aparição diminuísse à medida que a rede fosse treinada; porém foi garantido que essa frequência nunca zerasse, para evitar que a rede estacionasse em um valor mínimo local ou máximo local. Para a análise e a validação, essa variável epsilon ainda permaneceu com valores de aparição mínimos (uma a cada 100 ações), justamente para o comportamento do resultado não ser previsível, de forma que a rede aprendesse a lidar com situações adversas de todo tipo.

Para a análise do período de avaliação (de 2020 a 2021), foram utilizados o carregamento da rede neural treinada (com seus respectivos pesos) e a memória armazenada. Os demais aspectos semelhantes ao *loop* de treinamento, cujo rebalanceamento dos pesos não foi feito, foram utilizados levando ainda em consideração a recompensa e adaptando a memória, como pode ser visto no programa descrito no apêndice B.

4.4 Análise de dados

Os resultados foram analisados para cada ação de maneira independente, pois, mesmo com base em análise técnica de mercado, há estratégias que geram resultados divergentes de acordo com o ativo. Logo, foi necessário analisar caso a caso e identificar quais eram as melhores estratégias aplicáveis a um ativo, o que tornou necessária a criação de uma rede neural a partir da base de dados do ativo em questão.

Visto que cada ativo tem suas particularidades, além de os valores de fechamento influenciarem-no diariamente, e que algumas empresas são estatais ou trabalham com *commodities*, entre outros fatores, não há por que utilizar a mesma rede neural para outros ativos.

5 RESULTADOS E DISCUSSÃO

Os resultados são apresentados de acordo com a fase à qual se referem: 1) treinamento e 2) aplicação da rede neural aqui desenvolvida.

5.1 Treinamento

Os resultados obtidos durante o treinamento foram satisfatórios: tiveram uma média de 91,5% de taxa de acerto e uma média de relação lucro/perda de 8,32. No entanto, deve-se levar em consideração que o número de operações não foi limitado, a fim de obter uma melhor taxa de aprendizagem, isto quer dizer, havia vários pontos de compras ou vendas consecutivas que acumulavam ativos na carteira e permaneciam durante mais tempo na carteira, diferentemente do período de avaliação (no qual foi permitida apenas uma operação por vez).

Deve-se levar em consideração o contexto do período de treinamento (2015 e 2019): de modo geral, os índices eram muito positivos para o mercado financeiro brasileiro; o IBOV chegou a ter uma alta de 130%, manteve-se em alta, quebrou recordes e atingiu altos valores de mercado.

As redes neurais foram todas treinadas com cerca de 1.000 interações, um pouco variáveis entre elas, cuja variação é explicada pelo fato de que não há como haver uma convergência total do sistema devido à existência de componentes randômicos. Para a análise final, foram escolhidas as que tiveram os melhores resultados. Os resultados das redes neurais durante o período de treinamento podem ser observados a seguir (tabela 6).

Tabela 6 - Resultados do período de treinamento

Treinamento	Operações	Taxa de acerto	Relação lucro/perda
B3SA3	371	93,53%	9,53
BBAS3	355	92,68%	7,42
PETR4	331	92,46%	11,79
VALE3	311	87,36%	4,57

Fonte: Autoria própria.

Na tabela acima, o número de operações indica uma operação completa de compra e venda, a taxa de acerto indica a porcentagem de operações que obtiveram resultados positivos e a relação lucro/perda indica a média de quanto foi ganho proporcionamente em cada operação positiva.

Pode-se observar que, além de todas as redes treinadas terem uma alta taxa de acerto, cada qual aplicada à sua respectiva empresa, tiveram uma relação lucro/perda favorável durante o treinamento devido à abertura simultânea de mais de uma operação. Muitos sistemas de análise técnica de mercado possuem taxas elevadas de acerto, em torno de 70% a 90%, mas a relação lucro/perda varia entre 0 e 2 em sua grande maioria.

A taxa de acerto deste estudo variou de 87,36% a 93,53, e a relação lucro/perda, entre 4,57 a 11,79, o que mostra que este sistema de redes neurais chegou a superar a margem que se espera desses resultados durante a fase de treinamento.

5.2 Avaliação

Os resultados obtidos durante a avaliação também foram positivos, e não foi possível compará-los diretamente com os resultados de treinamento. Isso porque, ao contrário do período de treinamento, o período de avaliação permitia apenas uma operação por vez, como normalmente uma pessoa comum realizaria suas operações.

Deve-se levar em consideração o contexto do período de avaliação (2020): houve alerta constante, quebra de empresas e alta volatilidade de mercado, devido à pandemia de COVID-19. Não por acaso, os índices do mercado financeiro brasileiro, bem como do mercado estrangeiro, eram muitos negativos; no índice IBOV, por exemplo, houve uma queda de 48% entre janeiro e março de 2020.

Os resultados das redes neurais durante o período de avaliação podem ser observados a seguir (tabela 7).

Tabela 7 - Resultado do período de avaliação

Avaliação	Operações	Taxa de acerto	Relação lucro/perda
B3SA3	15	60%	1,05
BBAS3	12	50%	1,33
PETR4	3	100%	
VALE3	9	33%	4,62

Fonte: Autoria própria.

Na tabela acima, o número de operações indica uma operação completa de compra e venda, a taxa de acerto indica a porcentagem de operações que obtiveram resultados positivos e a relação lucro/perda indica uma média do ganho proporcional em cada operação positiva. Obteve-se uma média de taxa de acerto de 79,48% e uma relação de lucro/perda média de 1,6.

Assim sendo, ficou na média de taxas de acerto de outros métodos comuns de análise técnica.

O número de operações é significativamente menor comparado ao período de treinamento, a princípio por causa do espaço temporal — o treinamento foi de cinco anos e com várias operações simultâneas, ao passo que a avaliação foi de um ano e só se permitiu uma operação por vez. Além disso, como o sistema só opera em operações de compra objetivando lucro, ou seja, ele primeiramente compra uma ação e depois a vende, e o mercado financeiro de modo geral apresentava queda, ele evitou ou fez poucas operações durante os períodos de grande volatilidade, tendo aproximadamente 59,5% de volatilidade no ano de 2020, mais do que o dobro de 2019, que foi de 23,8% (GREGÓRIO, 2020). Isso sugere que as taxas de acerto foram impactadas pela grande volatilidade do mercado, sinalizando um período de indecisão, assim como a relação lucro/perda foi influenciada por essa incerteza.

Mesmo diante desse contexto, o sistema aqui proposto portou-se bem, já que obteve resultados positivos em um momento de crise e volatilidade, a respeito do qual, aliás, analistas financeiros não tinham certeza da direção nem das operações de mercado (G1, 2020).

Para melhor visualização dos resultados positivos durante a avaliação, são apresentados os gráficos de B3SA3 (figura 20), BBAS3 (figura 21), PETR4 (figura 22) e VALE3 (figura 23), que mostram o quanto foi ganho ou perdido em cada operação. No caso, um lote padrão de ações equivale a 100 ações.

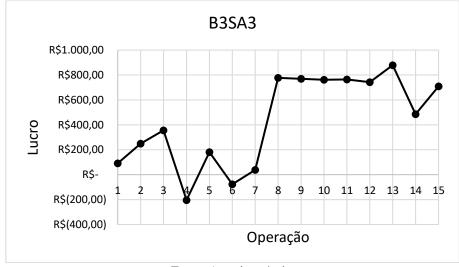


Figura 17 - Resultado financeiro das operações em B3SA3

Fonte: Autoria própria.

Em B3SA3, obteve-se um lucro total de R\$ 709,00, sendo em média um investimento de R\$ 5.070,00 para cada operação, e em média R\$ 47,00 de lucro, ou seja, o lucro por operação foi em torno de 0,93%.

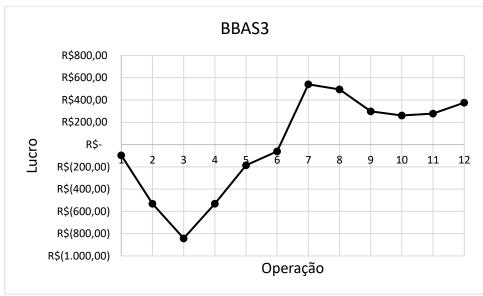


Figura 18 - Resultado financeiro das operações em BBAS3

Fonte: Autoria própria.

Em BBAS3, obteve-se um lucro total de R\$ 377,00, sendo em média um investimento de R\$ 3.380,00 para cada operação, e em média R\$ 31,00 de lucro, ou seja, o lucro por operação foi em torno de 0,92%.

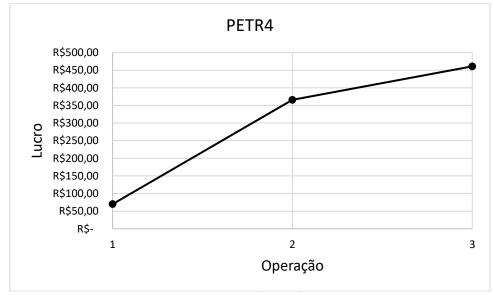


Figura 19 - Resultado financeiro das operações em PETR4

Fonte: Autoria própria.

Em PETR4, obteve-se um lucro total de R\$ 461,00, sendo em média um investimento de R\$ 1.600,00 para cada operação, e em média R\$ 150,00 de lucro, ou seja, o lucro por operação foi em torno de 9,32%.

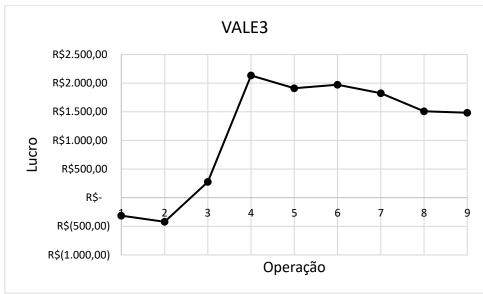


Figura 20 - Resultado financeiro das operações em VALE3

Fonte: Autoria própria.

Em VALE3, obteve-se um lucro de R\$ 1.480,00, sendo em média um investimento de R\$ 5.460,00, e em média R\$ 164,00 de lucro, ou seja, o lucro por operação foi em torno de 3,0%.

Obteve-se lucro na perspectiva geral, mas, devido ao momento de incertezas, quase todos tiveram uma queda nas primeiras operações antes de se recuperarem. A exceção foi PETR4, que teve poucas operações e conseguiu um resultado surpreendente (cem por cento de acerto) e alto lucro.

Os resultados da avaliação de um sistema inteligente de rede neural baseado apenas em preço de fechamento, no mercado de ações, nunca serão iguais aos do período de treinamento, pois o mercado financeiro age como um organismo complexo, respondendo a outras variáveis, como matemática, políticas econômicas e sanitárias, notícias e estado emocional das pessoas. Além disso, foi também observado que cada ativo nesse sistema exige uma rede neural própria, pois cada um tem um comportamento de mercado que varia não apenas com o preço, mas também com outras variáveis, o que exige um treinamento específico para a rede neural.

Em síntese, realizado o comparativo entre os ganhos e as perdas, bem com a análise de confiança, chegou-se a um resultado positivo, cuja média de taxa de acerto foi de 79,48% e cuja média de relação lucro/perda foi de 1,6. Isso mostra que o sistema proposto é eficiente e está dentro das médias de confiança e acerto de sistemas convencionais de análise técnica, que giram em torno de 70% a 90% e têm uma relação de lucro/perda entre 0 e 2.

6 CONCLUSÃO

O objetivo principal de uma operação no mercado de ações é obter lucro com ela, e por meio dela maximizar seu capital de maneira eficiente. Ao longo deste trabalho, foi demonstrado que as redes neurais desenvolvidas são capazes de auxiliar a maximização do capital. Apesar do fato de o ano de 2020 ter sido de imprevisibilidades devido à pandemia causada pelo coronavírus, o que ocasionou intempéries no mercado brasileiro e estrangeiro, o sistema validou positivamente os dados, pois conseguiu obter lucro em um mercado de alta volatilidade que estava em queda de 48% no primeiro trimestre daquele ano.

Além disso, este trabalho demonstrou que é recomendado montar uma carteira de ativos que não dependa somente de um único ativo, operando simultaneamente mais de um ativo, a fim de maximizar os resultados. No entanto, não pode haver um exagero na quantidade de ativos, pois isso pulveriza o capital e não traz resultados positivos; é preciso, portanto, escolher um número ideal de ativos, de empresas boas e diversificadas nos setores de atuação.

Como principal proposta para trabalhos futuros, pode-se levar em consideração a necessidade de estudos que tratem do desenvolvimento de uma rede que aceite diversas variáveis de entrada (preços de abertura, fechamento, máxima, mínima e volume diário), bem como análise uma carteira de ativos. Como proposta secundária, há que se ressaltar o fato de os softwares usados e a linguagem Python permitirem a criação de servidores ou de aplicativos que facilitam o uso da rede e são esteticamente agradáveis, o que também deve ser objeto de estudos futuros.

REFERÊNCIAS

ACERVOB3. **História da Bolsa**. 2020. Disponível em: http://acervob3.com.br/History. Acesso em: 20 jul. 2020.

ANACONDA. **Your data science toolkit**. 2020. Disponível em: https://www.anaconda.com/products/individual. Acesso em: 20 de jul. 2020.

B3. **Resumo das operações**. 2020. Disponível em: http://www.b3.com.br/pt_br/market-data-e indices/servicos-de-dados/market-data/consultas/mercado-de-derivativos/resumo-das-operacoes/estatisticas/. Acesso em: 20 de jan. 2020.

BELLMAN, R. E. **Dynamic Programming**. Princeton University Press: Princeton, 1957.

BELLMAN, R. E.; DREYFUS, S. E. **Applied Dynamic Programming**. Princeton University Press: Princeton, 1962.

BERTSEKAS, D. P.; TSITSIKLIS, J. N. **Neuro-dynamic Programming**. Belmont: Athena Scientific, 1996.

BRAGA, A. P. *et al.* **Redes Neurais Artificiais**: Teorias e Aplicações. Rio de Janeiro: LTC, 2007.

CANEDO, G; **Inteligência Artificial – Parte I**. Guia do Código. Disponível em: https://guildadocodigo.atelie.software/intelig%C3%AAncia-artificial-parte-i-ebd62adbc10. Acesso em: 10 mai. 2021

CAPITAL RESEARCH. Padrões de candlestick: o que é e como interpretar? Out. 2019. Disponível em: https://www.capitalresearch.com.br/blog/investimentos/padroes-de-candlestick/. Acesso em: 20 jul. 2020.

CARVALHO, V. P. **Previsão de séries temporais no mercado financeiro de ações com o uso de rede neural artificial**. 2018. Dissertação (Mestrado em Engenharia Elétrica e Computação) — Universidade Presbiteriana Mackenzie, São Paulo, 2018. Disponível em: http://tede.mackenzie.br/jspui/handle/tede/3710#preview-link0. Acesso em: 8 ago. 2020.

CORREIA, J. S. **Operando na Bolsa de Valores Uutilizando Análise Técnica**. São Paulo: Navatec Editora, 2008.

G1. Bovespa fecha em queda nesta terça e tem pior mês em mais de 20 anos. G1, 31 mar. 2020. Disponível em: https://g1.globo.com/economia/noticia/2020/03/31/bovespa.ghtml. Acesso em: 25 abr. 2021.

GRANATYR, J. Equação de Bellman. **Inteligência Artificial Expert Academy**. 2019. Disponível em: https://iaexpert.academy/2019/07/16/equacao-de-bellman/. Acesso em: 8 ago. 2020.

GREGORIO, R. Ibovespa tem maior volatilidade desde 2008, e ações oscilam até 117%; veja ranking. **Valor Investe**, São Paulo, 2020. Disponível em: https://valorinveste.globo.com/mercados/renda-variavel/bolsas-e-indices/noticia/2020/11/09/ibovespa-tem-maior-volatilidade-desde-2008-e-acoes-oscilam-ate-

- 117percent-veja-ranking.ghtml, novembro de 2020. Acesso em: 8 de abr. 2021.
- HALFELD, M. **Investimentos**: como administrar melhor seu dinheiro. São Paulo: Fundamento Educacional, 2006.
- HAYKIN, S. Redes Neurais: Princípios e Prática. 2. ed. Porto Alegre: Bookman, 2007.
- HEBB, D. O. **The organization of behavior**: A neuropsychological theory. New York: Wiley, 1949.
- KIRSTEN, H. A. Comparação entre os modelos Holt-Winters e redes neurais para previsão de séries temporais financeiras. 2009. Dissertação (Mestrado em Engenharia de Produção e Sistemas) Pontifícia Universidade Católica do Paraná, Curitiba, 2009. Disponível em:
- https://bdtd.ibict.br/vufind/Record/P_PR_1a9e303c3b4ffb2e8ea35b559c28640c. Acesso em: 20 jul. 2020.
- LEE, J. W.; JANGMIN, O. A Multi-agent Q-learning Framework for Optimizing Stock Trading Systems. *In*: HAMEURLAIN, A.; CICCHETTI, R.; TRAUNMÜLLER, R. (Eds.). **Database and Expert System Applications:** 13th International Conference; Proceedings / DEXA 2002. Berlin: Springer, 2002. p. 153–162. Disponível em: https://link.springer.com/chapter/10.1007/3-540-46146-9_16. Acesso em: 20 jul. 2020.
- MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943.
- MEDEIROS, I. Número de Pessoas Físicas na Bolsa de Valores Dobra em Relação a 2019. **Correio Braziliense**, Brasília, 8 jun. 2020. Disponível em: https://www.correiobraziliense.com.br/app/noticia/economia/2020/06/08/internas_economia,8 61914/numero-de-pessoas-fisicas-na-bolsa-de-valores-dobra-em-relacao-a-2019.shtml. Acesso em: 20 jul. 2020.
- NARAYANAN, V.; JENNINGS, N. R. An adaptive bilateral negotiation model for ecommerce settings. *In*: IEEE INTERNATIONAL CONFERENCE ON E-COMMERCE TECHNOLOGY, 7., 2005. **Proceedings** [...]. 2005. Agents, Interactions & Complexity: 2005. Disponível em: https://eprints.soton.ac.uk/260816/. Acesso em: 20 jul. 2020.
- PEREIRA, J. F. et al. **Prototipagem Rápida no Processo da Revolução 4.0**: Atuação do designer e projetista no desenvolvimento de projetos. 2020. Disponível em: https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariadeControleeAutomacao/galdeno ro1906/prototipagem-rapida-no-processo-da-revolucao-4.pdf. Acesso em: 15 abr. 2020.
- RIBEIRO, C. V.; GOLDSCHMIDT, R. R.; CHOREN, R. Métodos para previsão de séries temporais e suas tendências de desenvolvimento. **Monografias em Sistemas e Computação**, n. 3, p. 1–26, set. 2009. Disponível em: https://www.researchgate.net/figure/Figura-05-Tipos-de-funcao-de-ativacao-a-funcao-de-limiar-b-funcao-linear-c_fig7_255635041. Acesso em: 20 jul. 2020.
- SANTOS, V. S. **O que é neurônio?** Brasil Escola. 2020. Disponível em: https://brasilescola.uol.com.br/o-que-e/biologia/o-que-e-neuronio.htm. Acesso em: 20 de jul. 2020.

SPYDER. **Welcome to Spyder's Documentation**. 2020. Disponível em: https://docs.spyderide.org/current/index.html. Acesso em: 20 jul. 2020.

SUTTON, R. S.; BARTO, A. G. Reinforcement Learning I: Introduction. 1999.

TOTVS. **O que é inteligência artificial**? Como funciona, exemplos e aplicações. 2019. Disponível em: https://www.totvs.com/blog/inovacoes/o-que-e-inteligencia-artificial/. Acesso em: 15 abr. 2021.

WATKINS, C. J. C. H; DAYAN, P. Technical Note Q-Learning. **Machine Learning**, v. 8, p. 279–292, 1992.

YAHOOFINANCES. **Quote**. 2020. Disponível em: https://finance.yahoo.com/quote/. Acesso em: 20 jul. 2020.

YOSHIDA, D. M.; BOTURA JUNIOR, G. Domótica, inteligência artificial e internet das coisas na habitação destinada aos idosos. **Assentamentos Humanos**, v. 20, p. 49–60, 2018. Disponível em:

https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariadeControleeAutomacao/galdeno ro1906/final-domo769ticaj-intelige770ncia-artificial-e-internet-das-coisas-na-habitac807a771o-destinada-aos-idosos.pdf. Acesso em: 15 abr. 2021.

APÊNDICE A — CÓDIGO DE TREINAMENTO

```
#Programa de treinamento TCC
#Bibliotecas
import math
import random
import numpy as np
import pandas as pd
import tensorflow as tf
from urllib.parse import urlencode
import matplotlib.pyplot as plt
import pandas_datareader as data_reader
from pandas.testing import assert_frame_equal
import json
import pickle
from tqdm import tqdm_notebook, tqdm
from collections import deque
# Criação da Classe de Treinamento
class IA Operador():
# Definições iniciais, nome, tamanho espaço de estados, neuronios de saida
 def __init__(self, state_size, action_space = 3, model_name = "IAOperador"):
    self.state_size = state_size
    self.action_space = action_space
    self.memory = deque(maxlen = 3000)
    self.model_name = model_name
# Definição da probabilidade de variavel randomica pelo epsilon
    self.gamma = 0.95
    self.epsilon = 1.0
    self.epsilon_final = 0.01
    self.epsilon_decay = 0.995
 #Criação do modelo de rede neural
    self.model = self.model_builder()
  def model builder(self):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(units = 20, activation = "relu",
                                    input_dim = self.state_size))
    model.add(tf.keras.layers.Dense(units = 40, activation = "relu"))
    model.add(tf.keras.layers.Dense(units = self.action_space,
                                    activation = "linear"))
    model.compile(loss = "mse", optimizer =
                  tf.keras.optimizers.Adam(lr = 0.0001))
    return model
#Definição da saida randomica ou da rede neural
  def trade(self, state):
    if random.random() <= self.epsilon:</pre>
      return random.randrange(self.action_space)
    actions = self.model.predict(state)
    return np.argmax(actions[0])
#Armazenamento de eventos na memoria
```

```
def batch_train(self, batch_size):
    batch = []
    for i in range(len(self.memory) - batch_size + 1, len(self.memory)):
      batch.append(self.memory[i])
    for state, action, reward, next_state, done in batch:
      if not done:
        reward = (reward + self.gamma *
                  np.amax(self.model.predict(next_state)[0]))
      target = self.model.predict(state)
      target[0][action] = reward
      self.model.fit(state, target, epochs=1, verbose=0)
    if self.epsilon > self.epsilon_final:
      self.epsilon *= self.epsilon_decay
#Pré-processamento
#Definição da função sigmoide
def sigmoid(x):
 return 1 / (1 + math.exp(-x))
#Formatação dos valores do preço
def stocks_price_format(n):
 if n < 0:
    return "- $ {0:2f}".format(abs(n))
  else:
    return "$ {0:2f}".format(abs(n))
#Carregamento da base de dados
def dataset_loader(stock_name):
 dataset = data_reader.DataReader(stock_name, data_source = "yahoo",
                                   start="2015-01-01", end="2020-01-01")
  close = dataset['Close']
 return close
#Criação do Espaço de estados
def state_creator(data, timestep, window_size):
 starting_id = timestep - window_size + 1
 if starting_id >= 0:
    windowed_data = data[starting_id:timestep + 1]
  else:
    windowed_data = - starting_id * [data[0]] + list(data[0:timestep + 1])
  state = []
  for i in range(window_size - 1):
    state.append(sigmoid(((windowed_data[i+1]-windowed_data[i])*100)/
                         windowed_data[i+1]))
 return np.array([state]), windowed_data
#Definições das variaveis iniciais, ativo, dataset, janela de analise
stock_name = "PETR4.SA"
data = dataset_loader(stock_name)
window size = 10
episodes = 1000
batch_size = 32
```

```
data_samples = len(data) - 1
#chamada da Classe para criação
trader = IA_Operador(window_size)
trader.model.summary()
# Loop de treinamento
for episode in range(1, episodes + 1):
  print("Episode: {}/{}".format(episode, episodes))
  state = state_creator(data, 0, window_size + 1)
  total profit = 0
  trader.inventory = []
  for t in tqdm(range(data_samples)):
    action = trader.trade(state)
    next_state = state_creator(data, t + 1, window_size + 1)
    reward = 0.01
   if action == 1 : # Comprando uma ação
     trader.inventory.append(data[t])
      print("AI Trader compra: ", stocks_price_format(data[t]))
    elif action == 2 and len(trader.inventory) > 0: # Vendendo uma ação
      buy_price = trader.inventory.pop(0)
      reward = data[t] - buy_price
      total_profit += data[t] - buy_price
      print("AI Trader venda: ", stocks_price_format(data[t]), " Profit: " +
            stocks_price_format(data[t] - buy_price))
    if t == data_samples - 1:
      done = True
    else:
      done = False
    trader.memory.append((state, action, reward, next_state, done))
    melm = trader.memory
    state = next_state
    if done:
      print("#############")
      print("Total: {}".format(total_profit))
     print("############")
    if len(trader.memory) > batch_size:
     trader.batch_train(batch_size)
#Salvando modelo e variaveis
  if episode % 10 == 0:
  trader.model.save("ai_traderPETR4_{}.h5".format(episode))
   with open('melmPETR4_{}.obj'.format(episode), 'wb') as filehandle:
     pickle.dump(trader.memory, filehandle)
   tradermodel_json = trader.model.to_json()
   with open('traderPETR4_{}, json'.format(episode), 'w') as json_file:
    json_file.write(tradermodel_json)
   trader.model.save_weights('trader1PETR4_{}.h5'.format(episode))
```

APÊNDICE B — CÓDIGO DE EXECUÇÃO

```
#Programa de execução TCC
#Bibliotecas
import math
import random
import numpy as np
import pandas as pd
import tensorflow as tf
from urllib.parse import urlencode
import matplotlib.pyplot as plt
import pandas_datareader as data_reader
from pandas.testing import assert_frame_equal
import ison
from tensorflow.keras.models import load_model
from tqdm import tqdm_notebook, tqdm
from collections import deque
import pickle
from tensorflow.keras.models import model_from_json
#Carregando modelo treinado
arquivo = open('traderPETR4_1000.json', 'r')
estrutura_rede = arquivo.read()
arquivo.close()
model = model_from_json(estrutura_rede)
model.load_weights('trader1PETR4_1000.h5')
model.summary()
action_space = 3
#Carregando memoria
with open('melmPETR4_1000.obj', 'rb') as filehandle:
    memory = pickle.load(filehandle)
memory = deque(maxlen = 3000)
gamma = 0.95
epsilon= 0.01
#Definição da saida randomica ou da rede neural
def trader( state):
   if random.random() <= epsilon:
     return random.randrange(action_space)
    actions = model.predict(state)
    return np.argmax(actions[0])
#Armazenamento da memoria
def batch_train(batch_size):
    batch = []
    for i in range(len(memory) - batch_size + 1, len(memory)):
      batch.append(memory[i])
    for state, action, reward, next_state, done in batch:
      if not done:
        reward = reward + gamma * np.amax(model.predict(next_state)[0])
      target = model.predict(state)
      target[0][action] = reward
#Pré-processamento
```

```
#Definição da função sigmoide
def sigmoid(x):
 return 1 / (1 + math.exp(-x))
#Formatação dos preços
def stocks_price_format(n):
 if n < 0:
   return "- $ {0:2f}".format(abs(n))
  else:
   return "$ {0:2f}".format(abs(n))
#Carregamento da base de dados
def dataset_loader(stock_name):
    dataset = data_reader.DataReader(stock_name, data_source = "yahoo",
                                     start="2020-01-01", end="2021-01-01")
    start_date = str(dataset.index[0]).split()[0]
    end_date = str(dataset.index[-1]).split()[0]
    close = dataset['Close']
   return close
#Criação do Espaço de estados
def state_creator(data, timestep, window_size):
  starting_id = timestep - window_size + 1
 if starting_id >= 0:
   windowed_data = data[starting_id:timestep + 1]
  else:
   windowed_data = - starting_id * [data[0]] + list(data[0:timestep + 1])
  state = []
 for i in range(window_size - 1):
    state.append(sigmoid(((windowed_data[i+1]-windowed_data[i])*100)/
                         windowed_data[i+1]))
 return np.array([state]), windowed_data
#Definições das variaveis iniciais, ativo, dataset, janela de analise
stock_name = "PETR4.SA"
data = dataset_loader(stock_name)
window_size = 10
episodes = 1
state = state_creator(data, 0, window_size + 1)
total_profit = 0
data_samples = len(data) - 1
batch_size = 32
#Loop de execução
for t in tqdm(range(data_samples)):
    next_state = state_creator(data, t + 1, window_size + 1)
    reward = 0
    action = trader(state)
   if action == 1 and len(trader) == 0: # Comprando uma ação
      trader.append(data[t])
      print("AI Trader compra:{} ".format(t), stocks_price_format(data[t]))
    elif action == 2 and len(trader) > 0: # Vendendo uma ação
      buy_price = trader.pop(0)
```