

CENTRO UNIVERSITÁRIO FEI  
CLAUDIO DE OLIVEIRA VILÃO JUNIOR

**UM AGRUPAMENTO DE MODELOS CONEXIONISTAS POR MEIO DE SINAPSES  
ARTIFICIAIS E SUAS APLICAÇÕES NO MERCADO DE CRIPTOMOEDAS**

São Bernardo do Campo

2020

CLAUDIO DE OLIVEIRA VILÃO JUNIOR

**UM AGRUPAMENTO DE MODELOS CONEXIONISTAS POR MEIO DE SINAPSES  
ARTIFICIAIS E SUAS APLICAÇÕES NO MERCADO DE CRIPTOMOEDAS**

Tese de Doutorado apresentada ao Centro Universitário FEI como requisito para obtenção do título de Doutor em Engenharia Elétrica, orientado pelo Prof. Dr. Reinaldo Augusto da Costa Bianchi.

São Bernardo do Campo

2020

Vilão Júnior, Claudio de Oliveira.

UM AGRUPAMENTO DE MODELOS CONEXIONISTAS POR  
MEIO DE SINAPSES ARTIFICIAIS E SUAS APLICAÇÕES NO  
MERCADO DE CRIPTOMOEDAS / Claudio de Oliveira Vilão  
Júnior. São Bernardo do Campo, 2020.

161 p. : il.

Tese - Centro Universitário FEI.

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi.

1. Redes Neurais convolucionais. 2. Agrupamento. 3. Mercado  
de Criptomoedas. 4. Arbitragem. 5. Blockchain. I. Bianchi,  
Reinaldo Augusto da Costa , orient. II. Título.

**Aluno:** Cláudio de Oliveira Vilão Junior

**Matrícula:** 515305-1

**Título do Trabalho:** Um agrupamento de modelos conexonistas por meio de sinapses artificiais e suas aplicações no mercado de criptomoedas.

**Área de Concentração:** Inteligência Artificial Aplicada à Automação e Robótica

**Orientador:** Prof. Dr. Reinaldo Augusto da Costa Bianchi

**Data da realização da defesa:** 18/06/2020

**ORIGINAL ASSINADA**

Avaliação da Banca Examinadora

São Bernardo do Campo, / / .

**MEMBROS DA BANCA EXAMINADORA**

Prof. Dr. Reinaldo Augusto da Costa Bianchi	Ass.: _____
Prof. Dr. Carlos Eduardo Thomaz	Ass.: _____
Prof. Dr. Ricardo de Carvalho Destro	Ass.: _____
Prof. Dr. Luiz Antonio Celiberto Junior	Ass.: _____
Profª Drª Anna Helena Reali Costa	Ass.: _____

A Banca Examinadora acima-assinada atribuiu ao aluno o seguinte:

APROVADO

REPROVADO

**VERSÃO FINAL DA TESE**

**ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS  
RECOMENDAÇÕES DA BANCA EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

A meus pais.

## AGRADECIMENTOS

Agradeço pelo incentivo que me foi dado em momentos de dificuldade por todos os meus familiares, em especial meus pais Anna Maria Cortezi Oliveira Vilão e Claudio de Oliveira Vilão.

Preciso agradecer o Prof. Dr. Reinaldo Augusto da Costa Bianchi por me ajudar e suportar em todos os momentos que passei durante este período, mas principalmente por acreditar no meu trabalho e em mim. Não posso deixar de mencionar, e talvez nem precise, mas, gostaria de agradecer pelas oportunidades que tem me dado, além do apoio, da motivação, das críticas construtivas e das aulas. Muito obrigado Professor.

Aos professores do Centro Universitário da FEI pelo incentivos e pelas aulas ministradas: Prof. Dr. Carlos Eduardo Thomaz, Prof. Dr. Flávio Tonidandel, Prof. Dr. Paulo Sérgio Silva Rodrigues, Prof. Dr. Paulo E. Santos.

Agradeço ao colega Dr. Isaac Jesus da Silva por sua contribuição me ajudando nas pesquisas científicas que vem sendo desenvolvidas com sugestões e críticas construtivas. Agradeço aos colegas Dr. Danilo Hernani Perico e Dr. Thiago Pedro Donadon Homem por terem me ajudado com sugestões e críticas, mas também transmitindo motivação e foco.

Agradeço à minha esposa Mônica Farias da Silva pelo suporte e paciência.

Agradeço a Deus.

“Creativity is seeing what everyone else sees, but then thinking a new thought that has never been thought before and expressing it somehow.”

Neil Degrasse Tyson

## RESUMO

Este trabalho propõe um algoritmo, chamado de CMEAS, tem inspiração biológica focada na forma que o crescimento de axônios neuronais atinge seu destino sináptico em outras redes de neurônios. Esse crescimento segue caminhos específicos no cérebro de animais, definidos por determinadas proteínas.

O CMEAS foi desenvolvido para agrupar duas redes neurais convolucionais, treinadas *a priori* em dois tópicos que influenciam simultaneamente o mercado de criptomoedas, como o tópico de notícias e de cotações. O meio pelo qual as redes são agrupadas, ocorre usando-se conexões externas às redes originais, para se conectar aos neurônios internos de cada rede. Duas vertentes foram propostas para o treinamento do CMEAS, sendo um com aprendizado supervisionado e outro com aprendizado por reforço.

Os resultados comprovados pelos testes de Wilcoxon, demonstram que o CMEAS teve melhor fator de lucro e índice sharpe superior nos experimentos em relação aos algoritmos de agrupamento clássico por meio de votação e redes profundas usadas de forma individual, o algoritmo, também, foi superior em todas as métricas da estratégia compra e retêm (*buy and hold*), além disso, o algoritmo obteve resultados próximos, porém, melhores que os da CNN-LSTM considerada estado da arte, dadas as métricas utilizadas.

Palavras-chave: Agrupamento, Redes Neurais Convolucionais, Mercado Criptomoedas, Arbitragem, Blockchain

## ABSTRACT

This work proposes an algorithm, named CMEAS, has biological inspiration focused on the way that the growth of neuronal axons reaches their synaptic destination in other neural networks. This growth follows specific pathways in the brain of animals defined by certain proteins.

CMEAS was developed to group two convolutional neural networks, trained *a priori* on two topics that simultaneously influence the cryptocurrency market, such as news and prices. The means by which networks are grouped occurs using connections external to the original networks to connect to the internal neurons of each network. Two strands were proposed in order to train CMEAS, being one with supervised learning and the other with reinforcement learning.

The results confirmed by the Wilcoxon tests demonstrate that the CMEAS had a better profit factor and a higher sharpe index in the experiments in relation to the classic ensemble algorithms through voting and stand-alone deep networks, the algorithm was also superior in all the metrics of the buy and hold strategy, in addition, the algorithm obtained similar results, however, better than those of CNN-LSTM considered state of the art, given the metrics used.

Keywords: Ensemble, Convolutional Neural Networks, Cryptocurrency Market, Arbitrage, Blockchain

## LISTA DE ILUSTRAÇÕES

Figura 1	– Exemplo esquemático do neurônio artificial proposto por David E Rumelhart, Geoffrey E Hinton, Williams et al. (1988). . . . .	36
Figura 2	– Um exemplo de fronteira complexa. . . . .	38
Figura 3	– Um exemplo de vários classificadores mais simples ou mais fracos trabalhando em conjunto para determinação de fronteiras mais complexas. . . . .	40
Figura 4	– Como classificadores são agrupados usando o Bagging. . . . .	47
Figura 5	– Como classificadores são agrupados usando o Stacking. . . . .	48
Figura 6	– Um Exemplo de Rede Neural Convolutacional. . . . .	59
Figura 7	– Um Exemplo do uso de aprendizado por Reforço Profundo usando uma rede neural para aproximar os estados contínuos provindos do ambiente. . . . .	66
Figura 8	– Uma Q-network de fluxo único (em cima) e a Dueling Q-Network (em baixo). A Dueling Q-Network tem dois fluxos para estimar separadamente o valor do estado e as vantagens de cada ação; o módulo de saída verde para combiná-los. Ambas as redes emitem valores Q para cada ação. . . . .	68
Figura 9	– Arquitetura de uma <i>Arquitetura de Rede Neural Convolutacional com LSTM (Convolutional LSTM Neural Network)</i> (CNN-LSTM). . . . .	80
Figura 10	– Estrutura interna e externa de um Blockchain . . . . .	83
Figura 11	– Esquemático simplificado do acoplamento de redes neurais. . . . .	90
Figura 12	– Acoplamento de rede com o uso da Matriz $W^\alpha$ . . . . .	93
Figura 13	– Visão do acoplamento de redes. . . . .	93
Figura 14	– Redes simplificadas para treinamento do Acoplamento. . . . .	94
Figura 15	– Sentido da propagação do erro para Acoplamento de duas redes neurais simples. . . . .	95
Figura 16	– Dois sentidos da propagação do erro para Acoplamento de duas redes neurais simples. . . . .	95
Figura 17	– Esquemático do CMEAS-TD. . . . .	97
Figura 18	– Áreas de treinamento (azul) e teste (laranja) para períodos : de altas e baixas acentuadas (acima à esquerda), de altas proeminentes (acima à direita), de alta suave (abaixo à esquerda), de estabilidade (abaixo à direita) . . . . .	106
Figura 19	– Arquitetura da rede de notícias. . . . .	107
Figura 20	– Arquitetura da rede de cotações. . . . .	108

Figura 21 – Esquemático do acoplamento de redes neurais na sua forma profunda. . . .	109
Figura 22 – Ações tomadas pelo algoritmo CMEAS-DG no período de estabilidade com duas camadas. . . . .	109
Figura 23 – Sumário da estrutura da CNN-LSTM. . . . .	116
Figura 24 – Grafo orientado para criar rotas entre moedas Fiat e Criptomoedas. . . . .	126
Figura 25 – Operação de Arbitragem para um par de moedas. . . . .	127
Figura 26 – Ganhos de juros simples sobre capital investido por mês de 2019. . . . .	128
Figura 27 – Comparação de ganhos sobre capital investido no mês de dezembro de 2019 entra a arbitragem simples e a arbitragem CMEAS-DG. . . . .	130

## LISTA DE TABELAS

Tabela 1	– Tabela de principais eventos sobre redes neurais . . . . .	70
Tabela 2	– Tabela de principais eventos sobre redes neurais . . . . .	70
Tabela 3	– Treinamento com propagação do erro em período de estabilidade . . . . .	110
Tabela 4	– Treinamento com propagação do erro em período de alta leve . . . . .	110
Tabela 5	– Treinamento com propagação do erro em período de alta acentuada . . . . .	110
Tabela 6	– Treinamento com propagação do erro em período englobando estabilidade, alta e queda acentuada . . . . .	111
Tabela 7	– Treinamento com aprendizado por reforço em período de estabilidade . . . . .	112
Tabela 8	– Treinamento com aprendizado por reforço em período de alta leve . . . . .	112
Tabela 9	– Treinamento com aprendizado por reforço em período de alta acentuada . . . . .	112
Tabela 10	– Treinamento com aprendizado por reforço em período englobando estabi- lidade, alta leve, alta e queda acentuada . . . . .	113
Tabela 11	– Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de estabi- lidade . . . . .	114
Tabela 12	– Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de alta leve . . . . .	114
Tabela 13	– Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de alta acentuada . . . . .	115
Tabela 14	– Índices Sharpe, Fator de Lucro e ROI para Ensemble, CMEAS-DG e CMEAS -TD em período englobando estabilidade, alta e queda acentuada . . . . .	115
Tabela 15	– Comparação com o LSTM . . . . .	116
Tabela 16	– Comparação com o <i>buy and hold</i> . . . . .	117
Tabela 17	– Resultado do Teste de Wilcoxon considerando o índice Sharpe para redes de notícias e Preço em relação ao CMEAS-DG . . . . .	119
Tabela 18	– Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS- DG em relação ao CMEAS-TD . . . . .	119
Tabela 19	– Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS- DG em relação ao CMEAS-TD . . . . .	120
Tabela 20	– Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS- DG e CMEAS-TD em relação ao CNN-LSTM . . . . .	121

## LISTA DE ALGORITMOS

Algoritmo 1 – Resumo do Algoritmo Retropropagação do Erro para redes neurais multicamadas. . . . .	37
Algoritmo 2 – Algoritmo Stacking. . . . .	47
Algoritmo 3 – Algoritmo TD(0). . . . .	63
Algoritmo 4 – CMEAS-DG Algoritmo de propagação de erros entre multiplas redes multi-camadas. . . . .	96
Algoritmo 5 – Algoritmo CMEAS-TD . . . . .	101
Algoritmo 6 – Algoritmo Arbitragem . . . . .	128

## LISTA DE ABREVIATURAS

ADDEMUP	<i>Algoritmo de adição de classificadores (Add Them Up)</i>
AM	<i>Aprendizagem de Máquina (Machine Learning)</i>
API	<i>Aplicação de Interface de processo (Application Process Interface)</i>
AT	<i>Negociação Algorítmica ou Automatizada (Algorithm Trading)</i>
BCH	<i>Bitcoin Cash</i>
BG	<i>Problema dos Generais Bizantinos (Byzantine Generals)</i>
BGLM	<i>Modelo Binomial linear generalizado (Binomial generalised linear model)</i>
BOVESPA	<i>Bolsa de Valores do Estado de São Paulo, agora parte da B3, empresa de infraestrutura de mercado financeiro com atuação em ambiente de bolsa, futuros e de balcão.</i>
BRL	<i>Real Brasileiro (Brazilian Real)</i>
BTC	<i>Bitcoin</i>
CAPES	<i>Coordenação de Aperfeiçoamento de Pessoal de Nível Superior</i>
CDI	<i>Certificado de Depósito Interbancário</i>
CDQN	<i>Redes Q profundas Contínuas (Continuous Deep Q-Networks)</i>
CE	<i>Corretora de Criptomoedas (Cryptocurrency Exchange)</i>
CELS	<i>Aprendizado por Agrupamento de Cooperação de sinapses (Cooperative Synapse Ensemble Learning)</i>
CM	<i>Comitê de Máquinas, (Machine Comitee)</i>
CMEAS	<i>Agrupamento de modelos através de sinapses artificiais(Connectionist Model Ensemble via Artificial Synapses)</i>
CMEAS-DG	<i>Agrupamento de modelos através de sinapses artificiais(Connectionist Model Ensemble via Artificial Synapses) (CMEAS) com descida de gradiente</i>
CMEAS-TD	<i>CMEAS com Aprendizado por Reforço por Diferença Temporal (Reinforcement Learning- Temporal Difference) (TD)</i>
CNN	<i>Redes Neurais Convolucionais, (Convolutional Neural Networks)</i>
CNN-LSTM	<i>Arquitetura de Rede Neural Convolucional com Long-short Term Memory (LSTM) (Convolutional LSTM Neural Network)</i>
CoSYNE	<i>Evolução Neural por cooperação de sinapses (Cooperative Synapse Neuro Evolution)</i>
CQD	<i>Como Se Queria Demonstrar</i>

CV	<i>Validação cruzada (Cross-Validation)</i>
DAC	<i>Auto Associadores Profundos (Deep Auto Encoders)</i>
DASH	<i>Dash</i>
DBM	<i>Máquinas de Boltzman Profundas (Deep Boltzman Machines)</i>
DBN	<i>Redes de Crença Profunda (Deep Belief Networks)</i>
DCNN	<i>Redes Convolucionais Profundas (Deep Convolutional Neural Networks)</i>
DDPG	<i>Gradiente de Política Determinística (Deep Deterministic Policy Gradient)</i>
DDQN	<i>Redes Q profundas (Deep Q network) (DQN) dupla (Q-Network)</i>
DDR	<i>Reforço Direto Profundo (Deep Direct Reinforcement)</i>
DL	<i>Aprendizado Profundo (Deep Learning)</i>
DLVQ	<i>Aprendizagem Dinâmica por Quantização Vetorial (Dynamic Learning Vector Quantization)</i>
DNN	<i>Redes Neurais Profundas (Deep Neural Networks)</i>
DOWJONES	<i>Dow Jones Industrial Average é um índice criado em 1896 pelo editor do The Wall Street Journal e fundador do Dow Jones e Company, Charles Dow e, até hoje, publicado por essa instituição</i>
DPoS	<i>Prova por meio de propriedade e Delegação (Delegated proof of stake)</i>
DQL	<i>Aprendizado de Redes Q profundas (Deep Q Learning)</i>
DQLPER	<i>Aprendizado Q profundo com repetição de Experiência Prioritária (Deep Q-Learning with Prioritized Experience Replay)</i>
DQN	<i>Redes Q profundas (Deep Q network)</i>
DT	<i>Árvores de Decisão (Decision Trees)</i>
DW-CAV	<i>Consulta e voto dinamicamente ponderads (Dinamically Weighted consult and vote)</i>
E	<i>Ensemble</i>
ECDSA	<i>Curva elíptica de assinatura digital (elliptic curve digital signature algorithm)</i>
EM	<i>maximização de expectativa (Expectation Maximization)</i>
EMH	<i>Hipótese da Eficiência do Mercado (Market Efficiency Hypothesis)</i>
ENN	<i>Agrupamento de redes neurais (Neural Network Ensemble)</i>
EP	<i>Poda de Agrupamentos ou Poda de Conjuntos (Ensemble Pruning)</i>
ERNN	<i>Redes Neurais Recorrentes de Elman, (Elman Recurrent Neural Networks)</i>
ETH	<i>Ethereum</i>
EUR	<i>Euro União Européia (European Union Euro)</i>

FEI	Fundação Educacional Inaciana
Fiat	<i>Moeda fiduciária, é a moeda legal de qualquer país em que é impressa e emitida pelo seu governo e respectivo Banco Central.</i>
GAL	<i>Algoritmos de Crescimento e Aprendizado (Grow and Learn Algorithms)</i>
GAP	<i>Gap é uma palavra inglesa que significa lacuna, vão ou brecha. De acordo com a economia, gap é a diferença entre o valor real e o valor previsto de alguma coisa. Também pode significar espaço entre negociações</i>
GBT	<i>Árvores de Impulso de Gradiente (Gradient Boosting Trees)</i>
GDQN	<i>Gorila DQN (Gorila Deep Q-Network )</i>
GPI	<i>Iteração de Política Generalizada (Generalized Policy Iteration)</i>
HFT	<i>Negociação de Alta Frequência (High Frequency Trading)</i>
IA	<i>Inteligência Artificial</i>
IoT	<i>Internet das coisas (Internet of Things)</i>
ISLE	<i>Aprendizado por Agrupamento por Importância de Amostras (Importance Sample Learning Ensemble)</i>
KNN	<i>Algoritmo dos K vizinhos mais próximos (K Nearest Neighbors)</i>
LR	<i>Regressão Logística (Logistic Regression)</i>
LRCN	<i>Rede Convolutacional Recorrente de Longo Prazo (Long Term Recurrent Convolutional Network)</i>
LSTM	<i>Long-short Term Memory</i>
LTC	<i>Litecoin</i>
MAPD	<i>Desvio Médio Absoluto Percentual (Mean Absolute Percentage Deviation)</i>
MAPE	<i>Erro Médio Absoluto Percentual, também conhecido como Desvio Médio Absoluto Percentual (Mean Absolute Percentage Deviation) (MAPD) (Mean Absolute Percentage Error)</i>
MDP	<i>Processo de decisão de Markov (Markov Decision Process)</i>
MLP	<i>MultiLayer Perceptron</i>
MSE	<i>Erro Quadrático médio (Mean Squared Error)</i>
NASDAQ	<i>A Associação Nacional de Corretores de Títulos de Cotações Automáticas é um mercado de ações automatizado norte-americano e, é também, o segundo maior mercado de ações em capitalização de mercado do mundo, depois da Bolsa de valores de Nova York (New York Stock Exchange) (NYSE) NASDAQ Stock Market - National Association of Securities Dealers Automated Quotations</i>

NB	<i>Classificador de Bayes Ingênuo (Naive Bayes)</i>
NE	<i>Neuro Evolução (Neural Evolution)</i>
NEAT	<i>Neuro Evolução por Aumento de Topologia (Neural Evolution of Augmenting Topologies)</i>
NIFTY	<i>Índice da Bolsa de Valores da Índia (National Stock Exchange) (NSE)</i>
NLP	<i>Processamento de Linguagem Natural (Natural Language Processing)</i>
NSE	<i>Bolsa de Valores da Índia (National Stock Exchange)</i>
NYSE	<i>Bolsa de valores de Nova York (New York Stock Exchange)</i>
PBFT	<i>Prova por Tolerância a Falhas Bizantinas Práticas (Practical byzantine fault tolerance)</i>
PoS	<i>Prova por meio da garantia de propriedade (Prove of Stake)</i>
PoW	<i>Prova por meio de trabalho (Prove of Work)</i>
PSO	<i>Algoritmo de otimização de partículas de enxame (Particle Swarm Optimization)</i>
QED	<i>Como Se Queria Demonstrar (quod erat demonstrandum)</i>
QN	<i>Rede Q (Q-Network)</i>
RAF	<i>Florestas Aleatórias (Random Forests)</i>
RBF	<i>Radial Basis Function</i>
RBM	<i>Máquinas de Boltzman Restritas (Restricted Boltzman Machines)</i>
ReLU	<i>Unidade Linear Retificadora (Rectified Linear Units)</i>
RNA	<i>Redes Neurais Artificiais</i>
RNN	<i>Redes Neurais Recorrentes, (Recurrent Neural Networks)</i>
ROI	<i>Retorno sobre o investimento (Return Over Investment)</i>
RRL	<i>Aprendizado por reforço recorrente, (Recurrent Reinforcement Learning)</i>
RW	<i>Passeio Aleatório (Random Walk)</i>
SDS	<i>Estratégias Auto Destruidoras (Self Defeating Strategies)</i>
SELIC	<i>Sistema Especial de Liquidação e Custódia, também chamada de taxa básica de juros</i>
Sharpe	<i>Coefficiente Sharpe - lucro ajustado pelo risco relativo (Sharpe Ratio)</i>
SNN	<i>Redes Neurais Pulsadas, (Spiking Neural Networks)</i>
ST	<i>Negociação Sistemática (Systematic Trading)</i>
STDNN	<i>Rede Neural Siamesa com Deslocamento Temporal (Siamese Time Delay Neural Network)</i>
SVM	<i>Máquinas de Vetores de Suporte (Support Vector Machines)</i>

SZSE	<i>Bolsa de valores de Shenzhen-China (Shenzhen Stock Exchange)</i>
TD	<i>Aprendizado por Reforço por Diferença Temporal (Reinforcement Learning-Temporal Difference)</i>
UHFT	<i>Negociação de Altíssima Frequência (Ultra High Frequency Trading)</i>
USD	<i>Dolar Americano (United States Dolar)</i>
XRP	<i>Ripple</i>

## LISTA DE SÍMBOLOS

$a$	Ação do Aprendizado por reforço
$\alpha$	Taxa de Aprendizado do Aprendizado por Reforço
$D$	Conjunto de treinamento
$E$	Conjunto de experiências do aprendizado por reforço DQN
$e$	Experiência do Aprendizado por reforço DQN
$\gamma$	Fator de desconto do Aprendizado por Reforço
$H$	Espaço de hipóteses
$h$	Ponto único que corresponde a modelo específico pertencente a $H$
$m$	Tamanho do conjunto de treinamento
$p$	Quantidade de redes neurais
$\pi$	Política
$Q$	Função Qualidade da combinação estado-Ação do Aprendizado por Reforço
$r$	Recompensa do Aprendizado por Reforço
$s$	Estado do Aprendizado por Reforço
$T$	Conjunto de classificadores
$V$	Função Valor dado um Estado do Aprendizado por Reforço
$\varrho$	Ponderação dada ao especialista.
$X$	Valores pertencentes a classe $Y$
$x$	Estado interno
$Y$	classe
$y$	Saída Resultado

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	21
1.1	MOTIVAÇÃO	21
1.2	OBJETIVO	22
1.3	JUSTIFICATIVA	23
<b>1.3.1</b>	<b>Estudo de Caso - Do Mercado Financeiro seguindo para a arbitragem no Mercado de Criptomoedas</b>	24
1.4	Organização do Trabalho	27
<b>2</b>	<b>Redes Neurais</b>	29
2.1	Uma Breve Introdução à Redes Neurais Biológicas e Artificiais	29
2.2	Um Enfoque biológico para Conexão de Redes Neurais	29
2.3	Redes Neurais Artificiais	33
<b>2.3.1</b>	<b>Algoritmo de Retropropagação</b>	34
2.4	Combinando e Aumentando Redes Neurais	37
<b>2.4.1</b>	<b>Escolha do Modelo de Classificação</b>	39
<b>2.4.2</b>	<b>Agrupando Classificadores</b>	44
<b>2.4.3</b>	<b>Agrupador de Classificadores do Tipo <i>Bagging</i></b>	45
<b>2.4.4</b>	<b>Agrupador de Classificadores do Tipo <i>Stacking</i></b>	46
<b>2.4.5</b>	<b>Técnicas de Agrupamento para Redes Neurais</b>	52
2.5	Conclusão	56
<b>3</b>	<b>Aprendizado Profundo e Aprendizado por Reforço Profundo</b>	57
3.1	Aprendizado Profundo	57
3.2	Aprendizado Por Reforço	62
<b>3.2.1</b>	<b>O método das Diferenças Temporais – TD-Learning</b>	62
<b>3.2.2</b>	<b>Q-Learning</b>	64
3.3	Aprendizado por Reforço Profundo	65
<b>3.3.1</b>	<b>Deep Q-Network</b>	67
3.4	Dueling Network Architecture	67
3.5	Conclusão	68
<b>4</b>	<b>Estado da Arte</b>	69
4.1	Pesquisa e Divulgação Sobre Agrupamento de Redes Neurais	69
4.2	Pesquisas de Mercado Financeiro e Operação Automática	71

4.3	Métricas de Desempenho . . . . .	74
<b>4.3.1</b>	<b>Fator de lucro</b> . . . . .	75
<b>4.3.2</b>	<b>ROI</b> . . . . .	75
<b>4.3.3</b>	<b>Índice Sharpe</b> . . . . .	76
4.4	Aprendizado de Máquina para o Mercado Financeiro . . . . .	76
<b>4.4.1</b>	<b><i>Long Short Term Memory</i></b> . . . . .	79
4.5	Criptomoedas . . . . .	80
4.6	Blockchain e Mineração . . . . .	81
<b>4.6.1</b>	<b>Bloco</b> . . . . .	82
<b>4.6.2</b>	<b>Algoritmos de Consenso</b> . . . . .	84
4.7	Aprendizado de Máquina No Mercado de Criptomoedas . . . . .	85
4.8	Conclusão . . . . .	89
<b>5</b>	<b>Proposta</b> . . . . .	90
5.1	Algoritmos CMEAS vistos da Perspectiva do Algoritmo da Propagação do Erro	94
5.2	Algoritmos CMEAS vistos da Perspectiva do Algoritmo TD . . . . .	97
5.3	Descrição do CMEAS no mercado de Criptomoedas . . . . .	98
<b>6</b>	<b>Experimentos</b> . . . . .	103
6.1	Convertendo Redes convolucionais do Mercado Tradicional para o Mercado de Criptomoedas . . . . .	103
6.2	Considerações de Treinamento . . . . .	104
6.3	Redes neurais convolucionais notícias e preços. . . . .	107
6.4	Com aprendizado por reforço TD . . . . .	112
6.5	Comparando com ensemble . . . . .	114
6.6	Comparação com o LSTM . . . . .	115
6.7	Comparação com o Buy and hold . . . . .	116
6.8	Testes Estatísticos . . . . .	118
<b>6.8.1</b>	<b>Teste de Wilcoxon</b> . . . . .	118
<b>6.8.1.1</b>	<b><i>CMEAS-DG com relação às redes de Notícias e Cotações</i></b> . . . . .	119
<b>6.8.1.2</b>	<b><i>CMEAS-DG com relação ao e CMEAS-TD</i></b> . . . . .	119
<b>6.8.1.3</b>	<b><i>CMEAS-DG e CMEAS-TD com relação ao Ensemble</i></b> . . . . .	120
<b>6.8.1.4</b>	<b><i>CMEAS-DG e CMEAS-TD com relação ao CNN-LSTM</i></b> . . . . .	120
6.9	Discussão . . . . .	121

<b>7</b>	<b>Estudo de Caso</b> . . . . .	122
7.1	Arbitragem e as Criptomoedas . . . . .	122
<b>7.1.1</b>	<b>Tipos de Arbitragem</b> . . . . .	123
7.2	Grafos Orientados . . . . .	124
7.3	Robô automatizado de Arbitragem . . . . .	125
7.4	Previsão de comprar em um bot voltado para arbitragem . . . . .	129
7.5	Discussões e Conclusões . . . . .	129
<b>8</b>	<b>Conclusão</b> . . . . .	131
8.1	Enquadramento da CMEAS . . . . .	131
8.2	Contribuições . . . . .	132
<b>8.2.1</b>	<b>Publicações Resultantes desse Doutorado</b> . . . . .	132
8.3	Trabalhos futuros . . . . .	134
	<b>REFERÊNCIAS</b> . . . . .	136

## 1 Introdução

A origem da teoria das redes neurais artificiais vêm da intenção de mimetizar as unidades de processamento do sistema nervoso de um animal, os neurônios biológicos, assim como todos os elementos que os envolvem, suas conexões, como trabalham e, em última instância, como geram a inteligência e a consciência em si (KOVÁCS, 2002).

Redes neurais artificiais, profundas ou não, têm sido usadas em uma diversidade de aplicações como reconhecimento de caracteres (WU, C. et al., 2014), imagens (CIRESAN; MEIER; SCHMIDHUBER, 2012), (CIRESAN et al., 2012), (XU et al., 2014), análise de crédito (ADDO; GUEGAN; HASSANI, 2018), previsão de séries temporais (CUI; CHEN, W.; CHEN, Yixin, 2016), robótica e controle (DEMUTH et al., 2014).

### 1.1 MOTIVAÇÃO

Segundo Akabane (2012), o domínio da *Inteligência Artificial* (IA) é classificado em tarefas ordinárias, formais e especializadas. Computadores precisam de uma complexa representação de conhecimento para executar tarefas que são simples e intuitivas para seres humanos. Os mais recentes estudos nessa área apontam para arquiteturas profundas, onde são construídas hierarquias de características através de aprendizado e essa construção, pode, ou não, exigir muitas camadas. A esse tipo de abordagem dá-se o nome de aprendizado profundo (GOOD-FELLOW; BENGIO; COURVILLE, 2016).

Apesar do sucesso das redes neurais para áreas individualmente treinadas, essas redes não apresentam o resultado esperado em alguns domínios que possuem informações conflitantes e subjetivas, nesses domínios, em alguns, inclusive, a decisão é complicada até mesmo para seres humanos. No caso do mercado financeiro, a não trivialidade se estende até atingir o que pessoas precisam considerar para tomar uma decisão de compra ou de venda de ações, ou como escolher a linha de crédito de uma instituição financeira, ou como fornecer uma carteira de seguros usando como fonte o histórico de determinada pessoa.

Como será apresentado no capítulo 2, essas decisões são tomadas usando-se ferramentas como redes neurais isoladas e suas saídas conectadas com técnicas baseadas em lógica proposicional, raciocínio baseado em casos, ou mesmo outros classificadores.

Em estatística e aprendizagem de máquina, esses métodos de conjunção ou de agrupamento, do inglês *Ensemble*, de classificadores usam vários algoritmos de aprendizagem para obter um melhor desempenho preditivo do que poderia ser obtido a partir de qualquer um dos algoritmos de aprendizagem individualmente (POLIKAR, Robin, 2006). Durante o desenvolvimento desse trabalho a palavra agrupamento será usada como referência à palavra em inglês, *Ensemble*, assim como o modelo de agrupamento de redes neurais, se referirá ao termo ENN

A principal motivação para combinar redes neurais é a de melhorar a sua capacidade de generalização, ou evitar a falha de redes individualmente para um tipo de classificação e apenas uma área de atuação (POLIKAR, Robin, 2006), porém, a idéia do presente trabalho é permitir que o agrupamento tenha um significado mais amplo, permitindo que as redes interajam entre si. Além disso, não se pretende iniciar um novo treinamento para cada uma das redes mas, sim, que sejam aproveitadas todas as suas características, parâmetros e topologias, já que o treinamento de redes neurais artificiais individualmente é dispendioso, necessita do ajuste de uma diversidade de parâmetros, incluindo sua topologia, sem mencionar, a necessidade de se possuir dados de treinamento suficientes para cada área a qual são aplicadas.

Métodos de agrupamento necessitam de um método de decisão ou maneira pela qual para decidir entre os especialistas, aqui entenda-se redes neurais, qual tem maior probabilidade de estar certo. Decidir a forma pela qual serão agrupadas seja escolhida e/ou treinada. Usar redes neurais já treinadas para tomar decisões de alto nível, aproveitando uma característica intrínseca dessas redes, suas próprias conexões internas entre neurônios, pode ser uma solução viável para reduzir a necessidade de determinação desses itens.

## 1.2 OBJETIVO

O objetivo principal do presente trabalho é propor um algoritmo de agrupamento de duas redes neurais profundas treinadas *a priori* e especializadas em suas áreas de classificação. O agrupamento será chamado de CMEAS, acrônimo para *Agrupamento de modelos através de sinapses artificiais* (*Connectionist Model Ensemble via Artificial Synapses*) (CMEAS) ou, em português, Agrupamento de modelos através de sinapses artificiais. A forma pela qual funcionará, será por meio de conexões externas às redes de modo que neurônios internos de cada rede sejam conectados.

A proposta de conexão de redes que será apresentada, se baseia no fato da comunicação entre redes de forma cooperativa para atingir um determinado objetivo. Para exemplificar a

proposta, considere duas redes neurais artificiais  $c$  e  $v$ . Considere agora que, a rede  $c$  possua um treinamento ideal para a tarefa  $L$ , ou seja, para qualquer entrada a rede forneça a saída idealmente correta e que a rede  $v$  seja treinada idealmente na tarefa  $M$ , portanto, se  $L$  e  $M$  possuírem um índice de correlação zero, as tarefas, por consequência, não possuiriam correlação nenhuma de abstração.

Assim, ao executar uma terceira tarefa  $N$  que necessita de informações implícitas nas tarefas  $L$  e  $M$  independentemente, há de se esperar que essas informações também estarão implícitas nas redes  $c$  e  $v$ . Por essa suposição, conclui-se que  $c$  e  $v$  *a priori* treinadas e independentes entre si, agora possuem uma correlação abstrata, dada a tarefa  $N$ .

A partir do momento que é dada às redes uma nova tarefa, somente será possível agregar um contexto em comum se a tarefa necessitar de informações presentes nas duas redes. De forma análoga, se a tarefa  $N$  for mais correlacionada com uma determinada rede, essa poderá sobrepujar suas concorrentes em sua execução. Para determinar essa correlação, a CMEAS pretende conectar os neurônios internos de duas ou mais redes, por meio de uma conexão axônica inter-redes ponderada de modo que, os pesos fornecidos por essa técnica alterem a influência de uma rede na outra. Se essa hipótese for verdadeira, então as conexões criadas por um algoritmo que converge para o ótimo, dados tempo e recursos suficientes, implicaria em conexões que representam a melhor correlação entre os neurônios das duas redes para a tarefa  $N$ . Neurônios precedidos por essas conexões externas também poderiam indicar correlação com o contexto da tarefa. Tratam-se de redes especializadas, porém, não isoladas entre si.

### 1.3 JUSTIFICATIVA

A CMEAS está de acordo com as teorias de redes neurais artificiais, pois propõe uma forma de conectar redes neurais diferentes, previamente treinadas, usando conexões externas provenientes de neurônios internos, conforme inspiração biológica. Do ponto de vista do aprendizado de máquina, este trabalho será distinto das teorias clássicas de agrupamento, por se tratar de uma nova forma de uso de redes neurais agrupadas com fins de solução de conflito ou de aumento de crença.

Ao agrupar redes previamente treinadas independentemente, poupa-se o tempo de treinar uma ou mais redes incluindo todos os parâmetros e recursos necessários. Por exemplo, considere duas redes neurais, uma rede treinada para reconhecimento de sentimentos em notí-

cias e uma outra rede treinada para reconhecimento de tendências de mercado, ambas treinadas independentemente com topologias otimizadas e acurácia no estado da arte.

Uma abordagem seria ignorar o treinamento dessas duas redes e treinar uma única rede nova, com nova topologia, com novos parâmetros e com um novo conjunto de dados englobando as duas áreas, isso, logicamente demandaria tempo e conhecimento específico de cada área, já que se trata de aprendizado supervisionado.

Há a possibilidade de se usar uma *Agrupamento de redes neurais (Neural Network Ensemble)* (ENN) onde as saídas de cada uma dessas redes competiriam entre si, e para isso seria necessário um sistema decisório para definir qual das redes tem maior probabilidade de estar certa, em outras palavras necessitaria de um gestor que definiria em cada caso qual informação é mais relevante, como foi proposto por Mohammadi e Das (2016).

A formalização da CMEAS se dará utilizando o modelo de *Agrupamento de redes neurais (Neural Network Ensemble)* (ENN), mais especificamente o *Stacking* (BREIMAN, Leo, 1996b; SAKKIS et al., 2001; WOLPERT, D. H., 1992) e o treinamento das conexões inter-redes se dará com a formalização da retropropagação do erro, *Backpropagation* e em uma outra proposta com o erro fornecido pela técnica de aprendizado por reforço conhecida como *Aprendizado por Reforço por Diferença Temporal (Reinforcement Learning- Temporal Difference)* (TD). A inspiração biológica se estende definindo quais axônios poderão ter seus pesos alterados e por onde poderão crescer até atingir outras redes, algo análogo ao que acontece com algumas proteínas que definem o caminho de crescimento de neurônios pelo cérebro.

### **1.3.1 Estudo de Caso - Do Mercado Financeiro seguindo para a arbitragem no Mercado de Criptomoedas**

Como estudo de caso, a CMEAS será aplicada no mercado criptomoedas. Esse domínio é uma área interessante de pesquisa, pois trata-se de um exemplo de aplicação dinâmico. Isso implica que um sistema inteligente deve permitir ao agente identificar o cenário e, com base no seu conhecimento, decidir qual ação ele deve realizar de modo a atingir um objetivo pré determinado.

Os mercados financeiros desempenham um papel importante na organização econômica e social da sociedade moderna (TINO; SCHITTENKOPF; DORFFNER, 2001). Com a modernização das transações financeiras e dos sistemas de informação, a grande quantidade de

informações disponíveis para um negociante do mercado de ações, do inglês *trader*<sup>1</sup>, pode tornar proibitiva a análise de um ativo financeiro em sua totalidade. Segundo Cavalcante et al. (2016), do ponto de vista do mercado financeiro, duas abordagens são comumente utilizadas para analisar e prever o mercado:

- a) **Análise Fundamentalista:** Os fundamentalistas abordam a predição do valor de um papel, como sendo o agrupamento de fatores econômicos que influenciam o ativo (CAVALCANTE et al., 2016);
- b) **Análise Técnica :** Os analistas técnicos acreditam que todos os fatores que podem influenciar um papel já estão embutidos no valor daquele ativo (CAVALCANTE et al., 2016). Acreditam que a história tende a se repetir e, portanto, modelam o comportamento histórico de um ativo financeiro como uma série temporal (MURPHY, 1999)

Na análise técnica, o mercado acionário é tratado como sendo uma série temporal de indicadores e valores que representam todas as informações referentes a um determinado papel. Porém, é sabido que aspectos fundamentalistas, como fatores emocionais (ACKERT; CHURCH; DEAVES, 2003), políticas macro e micro econômicas (PAGANO; VOLPIN, 2006), aspectos naturais de determinadas áreas e como os investidores avaliam a gestão dos administradores das empresas (FAMA, 1965) também influenciam esse mercado. Os trabalhos relevantes que usaram textos para predição, incluem: Notícias financeiras (GROTH; MUNTERMANN, 2011; SCHUMAKER et al., 2012), relatórios financeiros (WANG, B.; HUANG, H.; WANG, X., 2012), informações em microblogs (RUIZ et al., 2012), e em redes sociais como o Twitter (BOLLEN; MAO; ZENG, 2011). Essas são consideradas fontes relevantes de informações para prever o futuro comportamento do mercado (LEMOS, 2015). Essas abordagens investigaram como a mineração de características importantes em dados textuais, assim como, identificar os sentimentos dos autores das notícias, podem melhorar, significativamente, a previsão de valores financeiros futuros.

Segundo Lemos (2015) parecem haver aspectos técnicos e fundamentalistas na predição do valor futuro de um ativo. A relevância disso fica mais acentuada quando se verifica que a maior parte da atividade gerada nas negociações se deve à resposta a fatores emocionais, os quais estão totalmente fora dos parâmetros de um modelo (LEMOS, 2015).

---

<sup>1</sup>Segundo (DOWNES; GOODMAN, J. E., 1993), um *trader* é um negociante, em geral, se trata de qualquer pessoa que compre ou venda bens e serviços para obter lucro.

As técnicas de aprendizagem de máquina foram aplicadas com relativo sucesso na modelagem e previsão de séries temporais financeiras (CAVALCANTE et al., 2016; LEE, M.-C., 2009). Entre essas técnicas, as redes neurais artificiais têm sido amplamente utilizadas na previsão de séries temporais, uma vez que são métodos auto adaptativos conduzidos por dados capazes de capturar comportamentos não lineares sem quaisquer conhecimentos estatísticos *a priori* sobre os dados (LU, C.-J.; WU, J.-Y., 2011) (ATSALAKIS; VALAVANIS, 2009).

O mercado de criptomoedas pode ser considerado, para fins de predição e análise, muito similar ao mercado financeiro tradicional, já que ambos podem ser consideradas séries temporais fortemente influenciáveis por decisões humanas (MCNALLY; ROCHE; CATON, 2018).

Em muitos aspectos, as criptomoedas podem ser consideradas como sendo uma sub-classe de moedas digitais (CHUEN, 2015), categoria que inclui pontos fornecidos por plataformas de fidelidade, tokens para redes sociais, milhas aéreas emitidas por companhias aéreas, tokens virtuais para jogos de computador.

Nesse contexto, o mercado de criptomoedas não é apenas uma forma de transferir dinheiro, mas também uma nova classe de investimento, além disso, as moedas digitais geralmente são emitidas centralmente, vinculadas à moeda fiduciária ou relacionadas às organizações emissoras, nesse contexto, a criptomoeda tem características muito diferentes.

Embora a classe das criptomoedas tenha sido classificada por algumas autoridades tributárias como tendo similaridade com commodities cite KuoChuen17, não é um consenso sobre que tipo de investimento essas moedas se encaixam. Atualmente, as criptomoedas são frequentemente usadas em transações on-line, e seu uso aumentou a cada ano desde sua introdução (BÖHME et al., 2015; GRINBERG, 2012).

Mais de mil altcoins e crypto-tokens foram criados desde o nascimento do Bitcoin em 2009 (REID; HARRIGAN, 2011; NAKAMOTO, 2008). Dos quais 90 % dessas moedas são negociadas livremente em bolsas específicas e especializadas.

A volatilidade dos preços e o número de flutuações das transações dessas moedas são induzidos por falta de regulamentação e notícias, eles também são suscetíveis à crença humana de incerteza de permanência dessas moedas no futuro (REID; HARRIGAN, 2011; BÖHME et al., 2015).

## 1.4 Organização do Trabalho

O desenvolvimento dessa proposta foi estruturado por meio de capítulos concatenados como segue:

No capítulo 2, é feita uma pequena introdução às redes neurais biológicas que é motivada para indicar a inspiração biológica da proposta. Ainda nesse capítulo, uma revisão da história de redes neurais artificiais, alguns exemplos de redes incluindo redes profundas e formas de treinamento.

Conceitos de agrupamento de classificadores são abordados no mesmo capítulo, apresentando, em forma de revisão, alguns dos mais conceituados algoritmos de agrupamento, para que se possa por em perspectiva onde a proposta se situa.

A pesquisa prossegue com a utilização de algoritmos de Aprendizado profundo e por reforço no treinamento de redes neurais, de forma que será explicada de forma breve com fins de utilização no algoritmo CMEAS.

Com os conceitos de agrupamento de classificadores e como redes neurais são agrupadas e treinadas, será possível apresentar o estado da arte, onde artigos publicados em jornais relevantes e recentes da área são apresentados de forma sucinta no capítulo 4, pondo em perspectiva o modo como alguns agrupamentos de redes neurais solucionaram os mais variados problemas do mercado financeiro e de criptomoedas, nesse capítulo há, também, uma revisão de conceitos básicos de Blockchain para contextualizar o estudo de caso.

Prosseguindo para a proposta e sua descrição abarcada pelo capítulo 5, onde também serão apresentadas algumas contribuições teóricas. A partir do capítulo 6, os testes e soluções encontradas são detalhadas no domínio do mercado de criptoativos Ethereum.

Após o capítulo de experimentos ter sido apresentado, ainda é feito um estudo de caso com uma aplicação em um robô de operações de ordens automáticas desenvolvido para usar uma estratégia de arbitragem com diversas moedas. Para descrever os processos pelos quais o robô de arbitragem opera, o capítulo 7 foi descrito explicando o Blockchain, passando pelo processo de mineração, consenso e estrutura do bloco, logo em seguida é feita uma breve explanação sobre o que é a estratégia de arbitragem e como essa estratégia foi implementada com o uso de grafo orientados, por fim o algoritmo CMEAS-DG foi incorporado ao processo de arbitragem visando a solução do problema de trava *lock* de operações devido à insuficiência de saldo.

Finalmente, nas considerações finais do capítulo 8, é feita uma breve síntese dos resultados positivos e limitações das técnicas apresentadas apontado para perspectivas futuras. Nesse capítulo está incluído, também, uma seção de discussão sobre algumas contribuições da proposta.

## 2 Redes Neurais

Este capítulo apresenta uma revisão dos conceitos relevantes para o desenvolvimento deste trabalho. Inicialmente é feita uma breve introdução às redes neurais biológicas que servirão de inspiração para a solução proposta. Em seguida será apresentada uma breve revisão histórica das redes neurais artificiais, como surgiram e, principalmente, o treinamento retropropagação que será utilizado na proposta a ser descrita no capítulo 5.

Serão apresentadas diversas formas de agrupamento demonstrando que são altamente dependentes do modelo de classificadores especializados escolhidos, passando pela escolha do modelo na seção 2.4.1 e estrutura para combinar esses modelos. As redes neurais artificiais são modelos amplamente utilizados e suas combinações são expostas de maneira a contextualizar e enquadrar a proposta.

O capítulo segue com os adventos da aprendizagem profunda e aprendizado por reforço, de modo que são descritas apenas as técnicas relevantes, tais como redes de convolução e aprendizado por reforço. Essas técnicas atingiram o estado da arte em diversos segmentos e são descritas como forma de embasamento para a proposta.

### 2.1 Uma Breve Introdução às Redes Neurais Biológicas e Artificiais

Nas próximas seções, o enfoque se dará visando como redes neurais biológicas se agrupam ou se conectam, seguindo para redes neurais artificiais com uma breve revisão histórica até o aprendizado profundo que tem atingido o estado da arte em várias áreas e, finalmente, a última seção 3.1 discorrerá sobre os agrupamentos de redes neurais existentes.

### 2.2 Um Enfoque biológico para Conexão de Redes Neurais

Essa seção revisa a bibliografia das últimas tendências do que foi descoberto da morfologia de neurônios biológicos, além disso, é motivada para indicar a inspiração biológica da proposta que será apresentada no capítulo 5, podendo conter alguns termos usados apenas na área de biologia.

Os neurônios são células especiais responsáveis pela transmissão e processamento de impulsos nervosos em um animal (PARKER; NEWSOME, 1998). São divididos em corpo celular ou soma, axônio e dendritos: O corpo celular onde se encontram as organelas e contém

o núcleo celular; O axônio que é responsável pela propagação do impulso nervoso; e, por fim, as sinapses, contidas nos dendritos, são responsáveis por transmitir esse impulso para outros neurônios. Como não é possível que o impulso nervoso passe através do espaço entre neurônios, é necessário que as sinapses contenham substâncias neuro transmissoras, que podem acentuar ou reprimir impulsos nervosos (PARKER; NEWSOME, 1998).

Essas unidades nervosas transmitem o sinal, que pode parecer com uma corrente elétrica sendo carregada através de um fio, mas na realidade o movimento do sinal ao longo do axônio é mais semelhante à queima de um pavio. O axônio tem a parte exterior envolvida pela célula de Schwann (BHATHEJA; FIELD, 2006), que protege os neurônios que transmitem muitos sinais, e/ou de muita importância, além disso, o sinal se propaga mais rapidamente nos neurônios cobertos pela bainha de mielina, onde se encontram as células de Schwann (BHATHEJA; FIELD, 2006).

Durante a passagem do sinal a membrana que cobre o neurônio se altera, permitindo íons externos escoarem através dela, causando uma mudança súbita nas propriedades elétricas nesse ponto. Essas mudanças são propagadas ao longo do axônio provocando a despolarização do mesmo (SALA; CIOS, 1999) .

Após o pulso o neurônio volta ao estado de relaxamento, a membrana entra em um estado conhecido como período de refração absoluta, durante o qual é incapaz de produzir um outro potencial de ação independente da intensidade da despolarização. O próximo período é o de refração relativa, no qual o neurônio volta a ser capaz de produzir um novo potencial (MAASS, 2002). Milhares de neurônios estão em contato com outros através das sinapses, que podem ou não estar presentes nos dendritos (MAASS, 2002). Muitos desses neurônios estão produzindo ou transmitindo sinais excitadores ou inibidores. A característica temporal parece ter uma função vital no processamento e transmissão dos sinais (KÖNIG; ENGEL; SINGER, 1996) (SALA; CIOS, 1999)

Hill et al. (2012) identificou princípios chave que determinam a conectividade no nível das sinapses, reconstruindo virtualmente um micro circuito cortical em 3 dimensões de 298 células neocorticais e comparando com uma amostra de diversos mamíferos da mesma espécie, idade e região cerebral (HILL et al., 2012). Este achado sugere que os mecanismos químicos atrativos e repulsivos, geralmente não resultam em conectividade específica entre pares de neurônios. Além disso, essa descoberta sugere que o alinhamento completamente aleatório dos axônios e dendritos fornecem uma base suficiente para que uma conectividade funcional específica surja nos micro circuitos neurais locais (HILL et al., 2012).

Axônios em desenvolvimento que ainda não estão conectados possuem estruturas altamente dinâmicas e móveis em suas pontas. Essas estruturas são chamadas cones de crescimento, (do inglês *Growth Cones*). Eles orientam os axônios para o seu alvo sináptico, transduzindo sinais positivos e negativos em sinais que regulam o cito esqueleto do neurônio e, assim, determinam o curso e a taxa de crescimento axonal (KANDEL et al., 2000; MUELLER, 1999; FRANZE et al., 2009). Assim, os cones de crescimento são muito importantes durante a neurogênese embrionária e adulta, além disso, eles também são essenciais para a regeneração de conexões neuronais, bem como para o aumento da conectividade neuronal (KANDEL et al., 2000).

Cones de crescimento têm forma redonda ou cônica com dois tipos de protrusões que palpam sua vizinhança imediata reagindo a pistas de orientação atrativas ou repulsivas por meio do crescimento, torneamento ou retraimento de suas estruturas. A membrana dos cones de crescimento contém receptores especiais e moléculas de adesão celular que são sensíveis aos gradientes químicos, chamada quimiotaxia e propriedades mecânicas, chamada durotaxia (KOSER et al., 2016).

Os neurônios são criados através de um processo chamado neurogênese que ocorre principalmente durante o desenvolvimento pré-natal do sistema nervoso quando o cérebro em crescimento é povoado (HERRUP; YANG, Y., 2007). A zona ventricular no tubo neural embrionário contém células progenitoras (CASTRO, F. d., 2003), que se dividem em ciclos mitóticos, se diversificam e dão origem a neuroblastos e glioblastos. Eventualmente, neuroblastos e glioblastos serão diferenciados em neurônios e células gliais, respectivamente (NOCTOR et al., 2001; FIX, 2008; KANDEL et al., 2000).

As principais funções das células da glia, ou células gliais, são: Manter os neurônios em seus lugares; Fornecer nutrientes e oxigênio para os neurônios; Isolar um neurônio do outro; Destruir patógenos; Remover neurônios mortos; Formar mielina e participar na transmissão de sinais no sistema nervoso (KANDEL et al., 2000).

No início, são produzidas células gliais orientadas radialmente, mais tarde os neurônios e posteriormente todas as outras células gliais (NOCTOR et al., 2001; FIX, 2008). A glia radial fornece vias orientadoras para a migração de neurônios para seus locais finais na matéria cinza e branca do sistema nervoso (RAKIC, 1972, 2003). Uma vez que os neurônios deixam a zona ventricular, eles tornam-se permanentemente pós mitóticos, ou seja, eles não se dividem mais. Por outro lado, as células gliais não perdem a capacidade de multiplicar (HERRUP; YANG, Y., 2007; KANDEL et al., 2000).

Os neurônios, após migrar para seus locais finais, formam camadas bem definidas, cuja, posição está correlacionada com a data de nascimento do neurônio. As camadas internas do córtex cerebral são estabelecidas primeiro, sendo, a camada externa por último (KANDEL et al., 2000). Ao chegarem em seus locais de destino, os neurônios diferenciam-se adquirindo morfologias e conexões distintas e ampliam seus axônios, que seguem, precisamente, certos caminhos para seus alvos. Dentro de cada camada, os neurônios e, assim que, a migração neuronal é finalizada e a glia radial não é mais necessária como guia, ela desaparece (CHANAS-SACRE et al., 2000; MISSON; TAKAHASHI, T.; CAVINESS, 1991).

A neuro gênese e a migração neuronal não se limitam ao desenvolvimento embrionário. Demonstrou-se que, mesmo no cérebro adulto, novos neurônios são gerados (HERRUP; YANG, Y., 2007; ALVAREZ-BUYLLA; GARCIA-VERDUGO, 2002; ROY et al., 2000), o mesmo acontece em primatas adultos (GOULD et al., 1999), mesmo que o número de novos neurônios diminua enquanto o organismo envelhece (GHASHGHAEI; LAI; ANTON, 2007). Os autores afirmam que essa descoberta refutou a longa teoria de que o sistema nervoso é fixo e incapaz de regeneração, mas salientam que ela não refuta o conceito básico de que um neurônio maduro e diferenciado não se divide (GHASHGHAEI; LAI; ANTON, 2007). Além disso, eles relatam que uma lesão neuronal limitada e localizada induz a neuro gênese e, portanto, a substituição de neurônios no córtex cerebral adulto (GHASHGHAEI; LAI; ANTON, 2007).

Como já mencionado anteriormente, tanto a migração neuronal, como o crescimento axonal, parecem seguir caminhos específicos (KANDEL et al., 2000) e são guiadas pelas mesmas pistas moleculares (DANTAS et al., 2016). Para isso, neurônios e cones de crescimento precisam detectar uma variedade de sinais externos (DANTAS et al., 2016; TESSIER-LAVIGNE; GOODMAN, C. S. et al., 1996; DICKSON, 2002; MUELLER, 1999; HATTEN, 2002).

É importante ressaltar, no entanto, que a complexa migração e padrões de crescimento de neurônios não pode ser completamente explicada apenas com gradientes bioquímicos simples, especialmente quando se considera o comprimento de algumas vias ou a disseminação de alguns axônios (KOSER et al., 2016). No artigo Koser et al. (2016), os autores afirmam que, além das pistas bioquímicas, os neurônios são suscetíveis ao seu ambiente mecânico. Por exemplo, as estruturas dos cones de crescimento são muito sensíveis à força (CHEN, L. et al., 2016). Alguns neurônios crescem ao longo das células gliais, que são significativamente mais suaves do que seus neurônios vizinhos, isso sugere um envolvimento da mecânica na orientação neuronal e axonal. No entanto, os mecanismos subjacentes de sensibilidade mecânica neuronal ainda não são completamente entendidos.

A natureza dessas conexões parece ser essencialmente aleatória (HILL et al., 2012) e seguem proteínas que guiam seus caminhos até seu alvo sináptico (KANDEL et al., 2000; MUELLER, 1999).

A ideia biológica que redes neuronais, que já desempenhavam certas funções individualmente, pudessem estender suas conexões, através de conexões axoniais e sinápticas, até outras redes, até então isoladas, que também desempenhavam outras funções (HERRUP; YANG, Y., 2007; KANDEL et al., 2000), foi a inspiração para a proposta artificial deste trabalho.

### 2.3 Redes Neurais Artificiais

Inspirados na determinação experimental das propriedades eletrofísicas do neurônio, Warren McCulloch e Walter Pitts (MCCULLOCH; PITTS, 1943) criaram a primeira geração de neurônios artificiais. Esse modelo consistia, basicamente, de neurônios de disparo que enviavam um sinal alto para o próximo neurônio se a somatória ponderada das entradas recebidas nos dendritos ultrapassasse um determinado limiar de disparo.

Esses neurônios foram aplicados em redes neurais, chamadas Perceptrons (KUSSUL; BAIDYK, 2001), ideia desenvolvida por Rosenblatt no qual uma rede multi camadas de neurônios do tipo discriminadores lineares era interligada.

Em 1949, Hebb (HEBB, 1949) propôs um princípio pelo qual o aprendizado em sistemas nervosos complexos poderia ser reduzido a um processo puramente local, em que a intensidade das conexões sinápticas é alterada apenas em função dos erros detectados localmente.

Na segunda geração de redes neurais, uma função de ativação foi incluída para o processamento dos sinais, assim, esses neurônios são muito úteis para o processamento de sinais analógicos. Exemplos típicos são as redes neurais recorrentes e do tipo feed-forward. Ajustar o ganho sináptico pode alterar o fluxo de informação através da rede e usar uma função de ativação contínua permite usar um algoritmo de aprendizado como o algoritmo de retropropagação para treinar a rede de forma supervisionada.

É possível citar outros tipos de redes neurais artificiais como as redes *MultiLayer Perceptron* (MLP) (KUSSUL; BAIDYK, 2001), Jordan (JACOBS, R.; JORDAN, M. et al., 1993), Elman (WILLIAMS; ZIPSER, 1989), *Radial Basis Function* (RBF) (CHEN, S.; COWAN; GRANT, 1991), *Aprendizagem Dinâmica por Quantização Vetorial* (*Dynamic Learning Vector Quantization*) (DLVQ) (MARTINETZ; BERKOVICH; SCHULTEN, 1993), Hopfield (HOPFIELD; TANK, 1985). Existem ainda *Redes Neurais Pulsadas*, (*Spiking Neural Networks*)

(SNN) (LOVELACE; RICKARD; CIOS, 2010; KEMPTER; GERSTNER; VAN HEMMEN, 1999; VREEKEN, 2003; SALA; CIOS, 1999).

### 2.3.1 Algoritmo de Retropropagação

Na mesma época que Rosenblatt trabalhava no Perceptron, Widdrow desenvolveu o Adaline, um modelo linear de neurônio, conceitualmente simples, mas que possuía um sistema de treinamento chamado regra delta, que foi mais tarde generalizado para redes mais complexas.

Para explicar a regra delta, que é o princípio do backpropagation, será necessário detalhar o Adaline. Assim, esse neurônio é um elemento linear, cuja a saída  $y$  é simplesmente uma combinação linear das componentes do vetor de entrada  $x$  e  $w^t$  é a transposta do operador linear:

$$y = w^t x \quad (1)$$

Suponha agora, que sejam realizadas  $L$  observações de uma determinada função  $y = f(x_1, x_2, \dots, x_n) = f(x)$ :

$$\psi = \{x_l^e, y_l^e\}_{l=1}^L \quad (2)$$

Desta função será extraído um conjunto de treinamento  $(x, y)$  que permite determinar a aproximação linear, assume-se que será do tipo :

$$y(w; x) = \sum_{i=1}^n w_i x_i \quad (3)$$

Então para se obter o menor erro quadrático para a aproximação linear:

$$E(w^*) = \sum_{l=1}^L (y(w^*; x_l^e) - y_l^e)^2 \quad (4)$$

Onde  $E(w^*)$  é a função erro que se deseja minimizar. Se a função que se pretende aproximar é linear, então o mínimo erro quadrático será nulo.

Substituindo a equação 3 em 4, temos que o erro quadrático para um discriminador arbitrário de parâmetros  $w$  é:

$$E(w) = \sum_{l=1}^L \left( \sum_{i=1}^n w_i x_{i,l}^d - y_l^e \right)^2 \quad (5)$$

Assim, partindo de um ponto arbitrário  $w(0)$ , pode-se caminhar pela superfície  $E(w)$  em direção ao ponto de mínimo, bastando para isso dirigir-se sempre no sentido oposto ao do gradiente naquele ponto.

$$w(k+1) = w(k) - \eta' \frac{d}{dw} E(w(k)) \quad (6)$$

Onde  $\eta'$  determina o tamanho do passo que se dará naquela direção e influencia diretamente no tempo de convergência para o mínimo.

O modelo de neurônio proposto por Widrow foi estendido por David E Rumelhart, Geoffrey E Hinton, Williams et al. (1988) com a adição de uma função de ativação  $g(v)$ , cuja equação é:

$$y = g\left(\sum_{i=1}^n w_i x_i\right) \quad (7)$$

Então para o modelo de neurônio de Rumelhart, prossegue-se substituindo 7 em 4 para encontrar a função Erro para esse neurônio:

$$E(w) = \sum_{l=1}^L (g(w^l x_l^e) - y_l^e)^2 \quad (8)$$

E o gradiente dessa equação é:

$$\frac{dE(w)}{dw} = 2 \sum_{l=1}^L (y_l - y_l^e) \frac{dg(v)}{dv} x_l^e \quad (9)$$

Se for convencionado :

$$\delta_l = (y_l - y_l^e) \frac{dg(v)}{dv} \quad (10)$$

O resultado é:

$$\frac{dE(w)}{dw} = 2 \sum_{l=1}^L \delta_l x_l^e \quad (11)$$

O parâmetro  $w$  é atualizado usando a equação 6 fazendo:

$$w(k+1) = w(k) + \Delta w(k) \quad (12)$$

Onde,

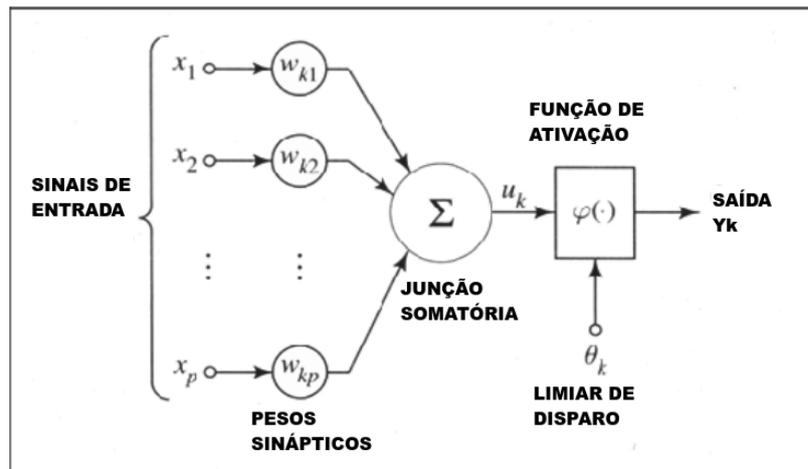
$$\delta w(k) = -\eta' \frac{dE(w)}{dw} = -\frac{\eta}{2} \frac{dE(w)}{dw} \quad (13)$$

Assim, resume-se a regra delta generalizada fazendo  $2 * \eta' = \eta$ , onde  $\eta$  também é chamada de taxa de aprendizado, e substituindo as equações 11 e 13 em 12:

$$w(k+1) = w(k) + \eta \sum_{l=1}^L \delta_l x_l^e \quad (14)$$

Visando resolver uma das maiores limitações no treinamento de redes multicamadas complexas da época, Rumelhart, Hinton e Williams, desenvolveram o algoritmo de *Backpropagation*, ou da retro propagação do erro. O algoritmo foi originalmente proposto para redes de neurônios com funções de ativação semilineares, que poderiam ser quaisquer funções contínuas como reta, Siebert, tangente hiperbólica, sigmóide (RUMELHART, David E; HINTON, G. E.; WILLIAMS et al., 1988) e, mais recentemente, a função linear retificada - ReLU (LECUN; BENGIO; HINTON, G., 2015).

Figura 1 – Exemplo esquemático de um neurônio artificial.



Fonte: “Adaptado de” David E Rumelhart, Geoffrey E Hinton, Williams et al. (1988) e Haykin e Network (1999)

Assim com a implantação da função de ativação é possível atualizar os pesos após a apresentação de cada exemplo do conjunto de treinamento. A equação 14 escrita para cada componente do vetor, se torna:

$$w(j+1) = w(j) + \eta \delta_l x_{i,l}^e \quad (15)$$

Visto que, para a função de ativação da sigmoide:

$$\delta_l = (y_l^e - y_l)(1 - y_l)y_l \quad (16)$$

Combinando as duas equações 15 e 16 verifica-se que a atualização do vetor  $w$  só depende do valor de referência  $y_l^e$ , das entradas  $x_l^e$  e da saída  $y_l$  todas as variáveis que circundam o neurônio.

Então o algoritmo de retro propagação do erro em redes de várias camadas, segundo (RUMELHART, David E; HINTON, G. E.; WILLIAMS et al., 1988) é descrito no algoritmo 1.

Algoritmo 1 – Resumo do Algoritmo Retropropagação do Erro para redes neurais multicamadas.

1 **Entrada:** Dado um conjunto de treinamento de  $L$  exemplos  $\psi = \{(x_l^e, y_l^e)\}_{l=1}^L$

2 Uma rede de  $M$  camadas com  $J$  elementos por camada é treinada sobre  $\psi$ ;

3 **Saída:**

4 Primeiro Passo: Escolhe-se um conjunto de pesos iniciais  $W(0)$  e uma taxa de aprendizado  $\eta$ ;

5 Segundo Passo: Escolhe-se um critério para definir quando a rede está treinada, que pode ser o número de iterações ou o erro ser menor que um determinado valor.

6 Terceiro passo: Calculam-se os erros de saída, da camada de saída e das camadas subsequentes:

7 **para**  $l \leftarrow 1$  **até**  $L$  **faça**

8  $\delta_{y(jM),l} = (y_{jM,l}^e - y_{jM,l})$

9  $\delta_{jM,l} = \delta_{y(jM),l} \frac{dg(v_{jM})}{dv_{jM}}$

10 **para**  $m \leftarrow (M - 1)$  **até**  $0$  **faça**

11  $\delta_{jm,l} = \left( \sum_{j(m+1)=1}^{J(m+1)} \delta_{j(m+1),l} w_{m+1,j(m+1),jm} \right) \frac{dg(h_{jm})}{dh_{jm}}$

12  $\Delta w_{m,jm,j(m-1)} = 2\eta \sum_{l=1}^L \delta_{jm,l} u_{j(m-1),l}^e$

13 onde a saída do neurônio  $u_{j(0),l}^e$  é igual a entrada do neurônio na próxima camada  $x_{j(0),l}^e$  para o mesmo exemplo.

14 **fim**

15 Quinto passo: Recalcula-se  $W(1)$  como sendo  $W(0) + \Delta W$

16 **fim**

O algoritmo 1 foi apresentado, pois, é a base para a proposta do treinamento da matriz de pesos responsável pelo agrupamento das redes.

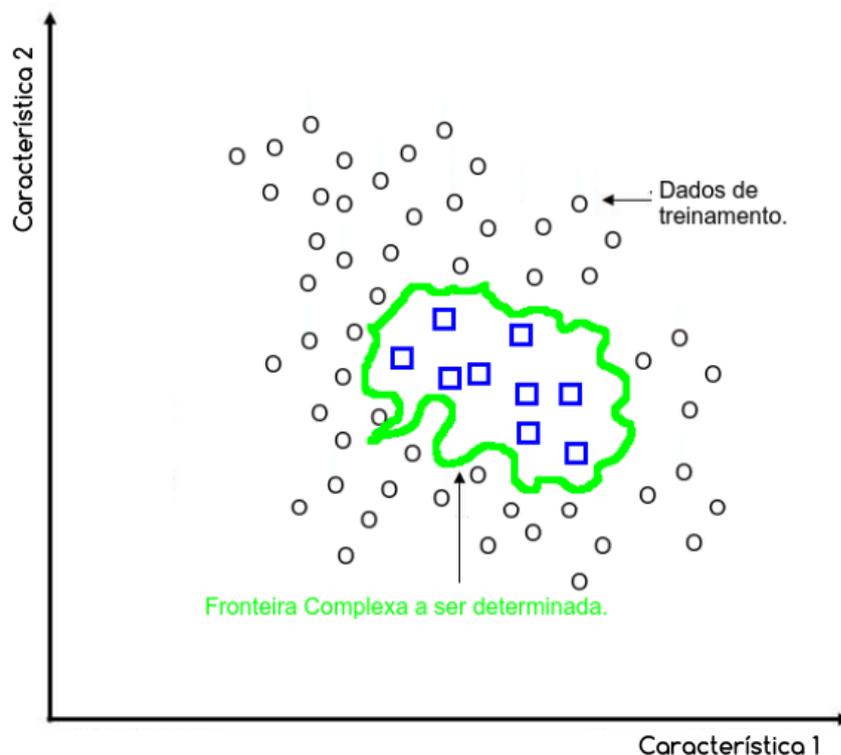
## 2.4 Combinando e Aumentando Redes Neurais

Em estatística e aprendizagem de máquina, métodos de conjunção ou de agrupamento *Ensemble*, de classificadores usam vários algoritmos de aprendizagem para obter um melhor desempenho preditivo do que poderia ser obtido a partir de qualquer um dos algoritmos de aprendizagem individualmente (POLIKAR, Robin, 2006).

A figura 2 mostra um conjunto de dados e uma fronteira complexa hipotética a ser determinada, já na figura 3 é demonstrado graficamente como vários classificadores mais simples podem ser combinados para se obter uma classificação mais acurada, aproximando a fronteira de decisão (MUHLBAIER; TOPALIS; POLIKAR, R., 2009).

Um sistema de agrupamento é obtido combinando diversos modelos de classificação chamados classificadores. Esses sistemas também são conhecidos como sistemas de classificação múltipla (POLIKAR, Robin, 2006).

Figura 2 – Um exemplo de fronteira complexa.



Fonte: “Adaptado de” Robin Polikar (2006)

Seguindo o exemplo dado por Robin Polikar (2006), considere o problema bidimensional de duas classes com um limite de decisão complexo representado na Figura 2. Um classificador que é capaz de aprender limites lineares não seria capaz de aprender este limite não linear complexo, porém, a combinação apropriada de um conjunto de tais classificadores pode aprender essa fronteira complexa.

Em seu artigo Robin Polikar (2006) supôs o uso de classificadores capazes de gerar contornos elípticos mas, é possível usar quaisquer tipos de classificadores mais fracos ou mais

simples, de modo que tal classificador não pode aprender o limite mostrado na figura 2 de forma isolada.

Na figura 2 Agora, considere uma coleção de limites de decisão circular gerados por um conjunto de tais classificadores como mostrado na 3

Na figura 3 foram utilizados classificadores capazes de gerar contornos circulares, que da mesma forma que os elípticos não são capazes, isoladamente, de aprender a fronteira exemplificada. Ainda na figura 3 cada classificador de contorno circular rotula os dados como classe 1 (Retângulo) ou classe 2 (círculo).

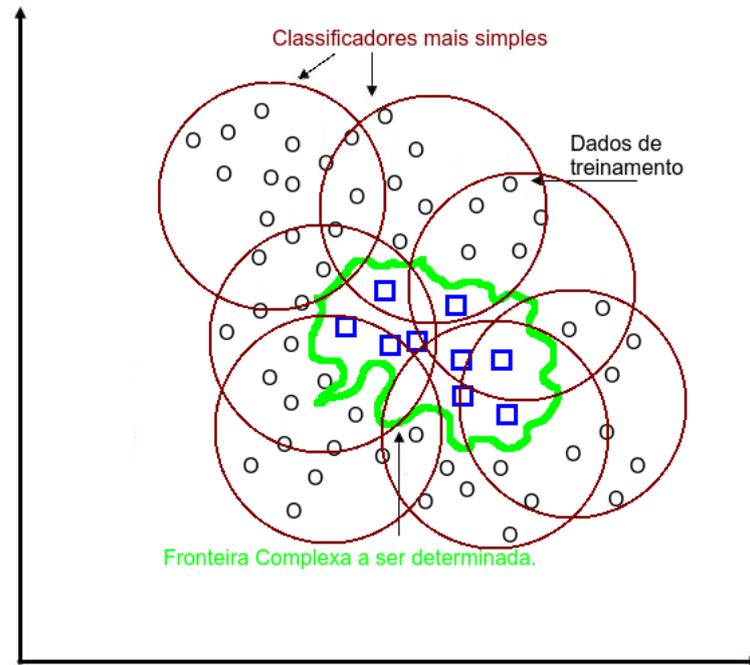
Ainda, segundo Robin Polikar (2006), uma decisão baseada na votação da maioria de um número suficiente de classificadores pode facilmente aprender esse limite não circular complexo. Encontrar a interseção das áreas de classificação do mesmo número de classificadores produziria o mesmo efeito. O sistema de classificação segue uma abordagem de dividir e conquistar, dividindo o espaço de dados em partições menores e mais fáceis de aprender, onde cada classificador aprende apenas uma das partições mais simples. Robin Polikar (2006) chega à conclusão de que o limite de decisão complexo subjacente poder ser aproximado por uma combinação apropriada de diferentes classificadores.

#### 2.4.1 Escolha do Modelo de Classificação

Apesar das redes neurais serem o modelo de classificação escolhido deste trabalho, essa seção apresenta a teoria do agrupamento de classificadores, de modo que, o termo modelo será usado no lugar de modelo de classificação por fins de simplicidade de notação.

Considerando uma coleção de alguns modelos possíveis como sendo um espaço de hipótese  $H$  e cada ponto único  $h \in H$  neste espaço corresponde a um modelo específico, uma variedade de algoritmos de classificação foram desenvolvidos (BISHOP, 2006), incluindo *Máquinas de Vetores de Suporte (Support Vector Machines)* (SVM), (VAPNIK; CHERVONENKIS, 1995; CRISTIANINI; SHAWE-TAYLOR, 2000), *Regressão Logística (Logistic Regression)* (LR) (HOSMER JR; LEMESHOW; STURDIVANT, 2013), *Classificador de Bayes Ingênuo (Naive Bayes)* (NB) (FRIEDMAN, N.; GEIGER; GOLDSZMIDT, 1997; RUSSELL; NORVIG; INTELLIGENCE, 1995), *Árvores de Decisão (Decision Trees)* (DT) (BREIMAN, L. et al., 1984; QUINLAN, J. R., 1986; QUINLAN, J. Ross, 1993), *Algoritmo dos K vizinhos mais próximos (K Nearest Neighbors)* (KNN) (COVER; HART, 2006), e *Redes Neurais Artificiais (RNA)* (RUMELHART, David E.; MCCLELLAND; PDP RESEARCH GROUP, 1986; MC-

Figura 3 – Um exemplo de vários classificadores mais simples ou mais fracos trabalhando em conjunto para determinação de fronteiras mais complexas.



Fonte: “Adaptado de” Robin Polikar (2006)

CULLOCH; PITTS, 1943; KUSSUL; BAIDYK, 2001). Eles diferem no espaço de hipóteses, critérios de qualidade do modelo e estratégias de busca (LI, Yaliang et al., 2014).

Assim, determinar qual o classificador é mais apropriado para um determinado problema de classificação é, talvez, a principal razão pela qual são usados sistemas de agrupamento na prática (MUHLBAIER; TOPALIS; POLIKAR, R., 2009; LI, Yaliang et al., 2014).

Segundo Muhlbaier, Topalis e R. Polikar (2009), essa determinação pode ser interpretada de duas formas diferentes:

- a) Que tipo de classificador deve ser escolhido entre muitos modelos existentes, como foi mencionado anteriormente.
- b) Dado um algoritmo de classificação particular, que implementação deste algoritmo deve ser escolhida. Por exemplo, diferentes inicializações de MLPs podem dar origem a diferentes limites de decisão, mesmo que todos os outros parâmetros sejam mantidos constantes.

Assim, existem dois componentes críticos na aprendizagem em conjunto: a determinação dos modelos de base e encontrar suas combinações.

O procedimento mais utilizado é o de escolher os classificadores com o menor erro nos dados de treinamento. Entretanto, esse procedimento não está imune a falhas (LI, Yaliang et al., 2014). O desempenho em um conjunto de dados de treinamento, mesmo quando calculado usando uma abordagem de *Validação cruzada (Cross-Validation) (CV)*, pode ser ilusória em termos de desempenho de classificação nos dados anteriormente não vistos (POLIKAR, Robin, 2006).

Geralmente, o objetivo da classificação é encontrar o modelo que consegue um bom desempenho ao prever dados futuros ainda não vistos. Para melhorar a capacidade de generalização de um modelo de classificação, este não deve se adequar demais, ficando restrito aos dados de treinamento, esse problema é mencionado em artigos em inglês com o nome *Overfitting*, em uma tradução livre seria algo como sobre-encaixe ou sobre-ajuste de dados. Em vez disso, o classificador deve ser abrangente o suficiente para cobrir casos não vistos. As abordagens de agrupamento podem ser consideradas como uma família de algoritmos de classificação, que são desenvolvidos para melhorar as habilidades de generalização dos classificadores (LI, Yaliang et al., 2014).

É difícil obter um modelo de classificação único com boa capacidade de generalização, normalmente chamado de classificador forte (LI, Yaliang et al., 2014), mas a aprendizagem em conjunto pode transformar uma combinação de classificadores fracos em um classificador forte. Formalmente, o classificador agrupado visa determinar os classificadores  $T$  dado um conjunto de treinamento  $D : \{h_1(x), \dots, h_T(x)\}$ , cada um dos quais mapeia os valores  $x$  na classe  $y$ . O próximo passo é combinar esses classificadores, em um agrupamento de classificadores de  $H(x)$  com a esperança de que ele atinja um melhor desempenho. (LI, Yaliang et al., 2014; MUHLBAIER; TOPALIS; POLIKAR, R., 2009)

Segundo, (BREIMAN, Leo, 1996a) existem dois fatores importantes que contribuem para o sucesso da aprendizagem em conjunto.

- a) Intuitivamente, se for conhecido a-priori que  $h_1(x)$  tem o melhor desempenho de previsão em dados futuros, então, sem qualquer dúvida, deve-se descartar os outros classificadores e escolher  $h_1(x)$  para futuras previsões. No entanto, as classes verdadeiras as quais pertencem os dados futuros são desconhecidos e, portanto, não é possível saber antecipadamente qual classificador é o melhor. Um exemplo simples pode ser usado para ilustrar melhor como o modelo de conjunto atinge um melhor desempenho é dado em Seni e Elder (2010), nele, cinco classificadores

completamente independentes com precisão de previsão de 70% individualmente são agrupados para atingir melhor desempenho.

- b) Outra vantagem do aprendizado em conjunto é a sua capacidade de superar a limitação do espaço de hipóteses feita por modelos únicos, ou seja, de um único classificador. Conforme anteriormente discutido, um classificador de modelo único geralmente busca o melhor modelo dentro de um espaço de hipótese. É muito provável que o modelo verdadeiro não resida no espaço da hipótese, e então, é impossível obter o modelo verdadeiro por essa abordagem.

O fator descrito pelo item 'b' define bem o motivo pelo qual está sendo proposto, neste trabalho, um novo modelo de agrupamento.

É importante enfatizar que não há garantia de que a combinação de vários classificadores sempre será melhor do que o melhor classificador individual. Nem uma melhoria no desempenho médio do conjunto pode ser garantida, exceto para certos casos especiais (FUMERA; ROLI, 2005). Assim, a combinação de classificadores pode não necessariamente superar o desempenho do melhor classificador no conjunto, mas certamente reduz o risco geral de fazer uma escolha particularmente fraca.

A proposta feita por Robin Polikar (2006) aponta que, para esse processo ser efetivo, os especialistas individuais devem exibir algum nível de diversidade entre eles. Em um trabalho posterior Muhlbaier, Topalis e R. Polikar (2009) propuseram um algoritmo de aprendizagem incremental chamado *Learn++*, que gerava um conjunto de classificadores para cada conjunto de dados, combinando-os pelo que ele chamou de voto da maioria ponderada. Essa técnica sofria de "outvoting" quando solicitado a aprender uma nova classe  $\omega$ , ou seja, não havia votos suficientes para que fosse criada uma fronteira de decisão para essa nova classe. Os votos coletivos dos classificadores anteriores, por uma decisão inevitavelmente incorreta, superam os votos da decisão correta dos novos classificadores em novas instâncias, isso acontecia até que existam novos classificadores suficientes para contrariar a saída incorreta.

Para solucionar esse problema Muhlbaier, Topalis e R. Polikar (2009) propôs um mecanismo de votação para combinar classificadores. Os classificadores individuais consultam-se uns com os outros para determinar quais são os mais qualificados para classificar uma determinada instância e decidir quanto peso, se houver, a decisão de cada classificador deve ser ter na decisão final, nomeou esse mecanismo de Consulta e voto dinamicamente ponderados, (do inglês *Dynamically Weighted Consult And Vote - DW-CAV*) (MUHLBAIER; TOPALIS; POLIKAR, R., 2009).

Quando a quantidade de dados de treinamento é muito grande, isso dificulta o treinamento de um único classificador, porém, nesse caso, os dados podem ser divididos estrategicamente em subconjuntos menores.

Cada partição pode então ser usada para treinar um classificador separado, que pode, então, ser combinado usando uma regra de combinação apropriada. Se, por outro lado, houver poucos dados, o *bootstrapping* pode ser usado para treinar diferentes classificadores usando diferentes amostras *bootstrap* dos dados, onde cada amostra *bootstrap* é uma amostra aleatória dos dados com substituição e tratados como se fossem independentes da distribuição subjacente (EFRON, 1992).

Na última década, houveram diversos estudos no sentido de combinar modelos competitivos em um *Comitê de Máquinas*, (*Machine Comitee*) (CM), e o sucesso de técnicas de agrupamento tem sido aplicada em múltiplas disciplinas e aplicações. Devido a estas vantagens, muitas abordagens de agrupamento (LIU, B. et al., 2016; PAISITKRIANGKRAI; SHEN; HENGEL, 2016; HASSAN; ABRAHAM, A., 2013; ZHOU, 2012; SENI; ELDER, 2010; SEWELL, 2008; POLIKAR, Robin, 2006; KUNCHEVA, 2004; DIETTERICH et al., 2000; BAUER; KOHAVI, 1999) foram desenvolvidas para combinar o poder de previsão complementar de múltiplos modelos.

Os classificadores base devem ser precisos e independentes para se obter um bom agrupamento (ZHOU, 2012). Geralmente, a literatura não exige que os modelos base sejam altamente precisos, desde que seja fornecida uma boa quantidade de classificadores base, de forma que, os classificadores fracos podem ser impulsionados a ser um classificador forte por combinação.

No entanto, Yaliang Li et al. (2014) comentam que, no caso extremo, em que a escolha de cada classificador seja errada, a combinação desses classificadores dará resultados ainda piores.

Portanto, pelo menos os classificadores base devem ser melhores do que a determinação de classes aleatoriamente (VIOLA; JONES, 2001). A independência entre os classificadores é outra propriedade importante que se faz necessária na coleção de classificadores de base (SENI; ELDER, 2010). Se os classificadores base estiverem altamente correlacionados e fizerem previsões muito semelhantes, sua combinação não melhorará mais a classificação final. Em contraste, quando os classificadores de base são independentes e fazem previsões diversas, os erros independentes têm melhores chances de serem cancelados (SENI; ELDER, 2010).

Dentro do contexto de classificação, a diversidade nos classificadores é tipicamente alcançada usando diferentes parâmetros de treinamento para cada classificador. Isso permite a classificadores individuais gerar diferentes limites de decisão. Se a diversidade adequada é alcançada, isso produzirá um erro diferente em cada classificador. A combinação estratégica desses classificadores pode reduzir o erro total.

Segundo (SENI; ELDER, 2010), os seguintes passos são usados para gerar um bom conjunto de classificadores base:

- a) Obtenha diferentes amostras de *bootstrap* do conjunto de treinamento e treine um classificador em cada amostra *bootstrap*;
- b) Extraia diferentes subconjuntos de exemplos ou subconjuntos de recursos e treine um classificador em cada subconjunto;
- c) Aplique diferentes algoritmos de aprendizagem no conjunto de treinamento;
- d) Incorpore aleatoriedade no processo de um determinado algoritmo de aprendizagem ou usar parametrização diferente para obter diferentes resultados de previsão.

Esses itens enumeram de maneira bem sucinta os tipos de modelos de agrupamento presentes na literatura.

## 2.4.2 Agrupando Classificadores

Uma vez que os classificadores base são obtidos, uma questão a ser considerada é como combiná-los. As estratégias de combinação usadas pelos métodos de aprendizagem por agrupamento são divididos em duas categorias: não ponderada e ponderada.<sup>1</sup>

A votação da maioria é uma estratégia típica de combinação não ponderada, na qual são contados o número de votos para cada classe prevista entre os classificadores base e é escolhido aquele que possua a maioria de votos para a classe prevista. Esta abordagem trata cada classificador de base como sendo igualmente preciso e, portanto, não os diferencia na combinação (ZHOU, 2012).

Por outro lado, a combinação ponderada geralmente atribui um peso a cada classificador, com a esperança de que pesos maiores sejam dados a classificadores mais precisos, de modo que a previsão final possa ir em direção aos classificadores mais precisos. Os pesos podem ser inferidos a partir do desempenho dos classificadores de base ou do classificador combinado no conjunto de treinamento (SENI; ELDER, 2010).

---

<sup>1</sup>Os modelos de agrupamento terão seus nomes em inglês sendo usados a partir desse ponto.

### 2.4.3 Agrupador de Classificadores do Tipo *Bagging*

Também chamada de agregação *Bootstrap* (BREIMAN, Leo, 1996a), o *Bagging* é um método de agrupamento que adota a técnica de amostragem chamada *Bootstrap* (EFRON; TIBSHIRANI, 1994) na construção de modelos base. Ele gera novos conjuntos de dados por amostragem do conjunto de dados original com substituição e treina os classificadores base nos conjuntos de dados amostrados.

Para obter um classificador agrupado, ele combina todos os classificadores base por maioria de votos. Embora seja uma abordagem simples, demonstrou ser um método poderoso empiricamente e teoricamente (FRIEDMAN, J. H.; HALL, 2007). Quando *Bagging* foi introduzido, (BREIMAN, Leo, 1996a) apresentaram uma explicação sobre por que *Bagging* é eficaz com classificadores fracos instáveis. Em Jerome H Friedman e Hall (2007), os autores estudaram *Bagging* através de uma decomposição de preditores estatísticos.

Para entender o *Bagging* e suas limitações, se faz necessário, primeiramente, explicar o *bootstrap*, então, formalmente define-se um dado um conjunto de dados de treinamento, como sendo,  $D = \{x_i, y_i\}_{i=1}^m (x_i \in \mathfrak{R}_n, y_i \in \gamma)$  onde  $m$ , é o tamanho do conjunto de treinamento. São amostrados dados de  $D$  com substituição para formar um novo conjunto de dados  $D'$ . O tamanho de  $D'$  permanece o mesmo que o de  $D$ . Alguns dos exemplos aparecem mais de uma vez em  $D'$  enquanto alguns exemplos em  $D$  não aparecem em  $D'$ .

Para um exemplo particular  $x_i$ , a probabilidade de aparecer  $K$  vezes em  $D'$  segue uma distribuição de Poisson com  $\lambda = 1$  (BREIMAN, Leo, 1996a). Ao definir  $k = 0$  e  $\lambda = 1$ , podemos obter que  $x_i$  não aparece em  $D'$  com uma probabilidade de  $\frac{1}{e}$ , então  $x_i$  aparece em  $D'$  com uma probabilidade de  $1 - \frac{1}{e} \approx 0.632$ . É esperado, portanto que  $D'$  tenha 63,2% de dados únicos de  $D$  enquanto o resto são compostos de dados duplicados.

Após a amostragem de conjuntos de dados  $T$  usando a amostragem de *bootstrap*, um classificador é treinado em cada um dos conjuntos de dados amostrados  $D'$  e, então, combinado, sendo a sua saída dada por maioria de votos. Para cada exemplo  $x_i$ , ocorre uma predição final, cujo o resultado é a classe com o maior número de predições feitas pelos classificadores base.

Segundo (LI, Yaliang et al., 2014) é necessário cautela ao selecionar algoritmos de aprendizagem para serem usados como classificadores base ao usar o *Bagging*. Como  $D'$  tem 63,2% de sobreposição com o conjunto de dados original  $D$ , se o algoritmo de aprendizagem for insensível à mudança nos dados de treinamento, todos os classificadores de base produzirão previsões semelhantes, ou seja, a combinação desses classificadores base, pode não melhorar o

desempenho do conjunto. Além disso, Yaliang Li et al. (2014) afirmam que para garantir uma grande diversidade de classificadores base, é melhor usar algoritmos de aprendizado instáveis, como Redes Neurais, Árvores de Decisão ao invés de algoritmos de aprendizado estáveis, como K-Vizinhos mais próximos.

Combinadores algébricos são combinadores que não são passíveis de treinamento, onde as saídas dos classificadores base são combinados através de uma expressão algébrica, os exemplos mais encontrados na literatura são: mínimo, máximo, soma, produtória, média, mediana, média ponderada.

Existem várias outras regras de combinação, que são indiscutivelmente mais sofisticadas do que as supracitadas, por exemplo:

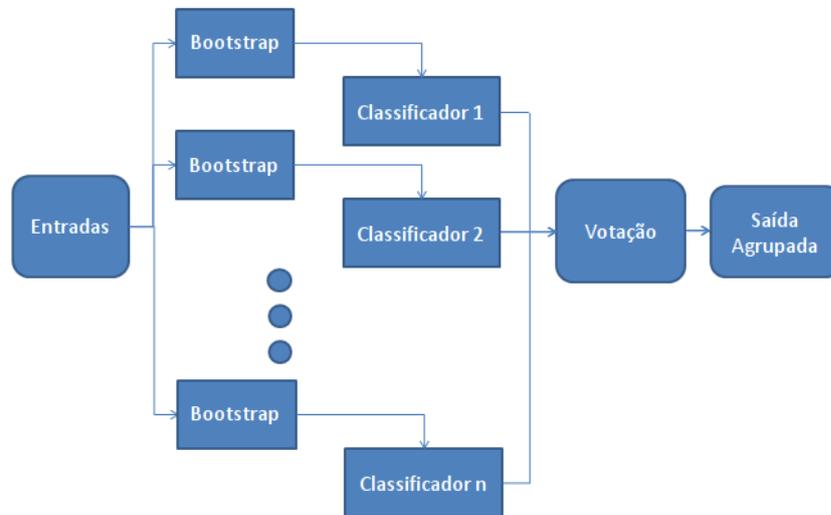
- a) A contagem Borda (EMERSON, 2013) que leva em consideração os rankings das classes,
- b) O espaço de conhecimento comportamental (HUANG, Y. S.; SUEN, 1993) que usa uma tabela de pesquisa que lista as classes corretas mais comuns para todas as combinações de classes possíveis dadas pelos classificadores;
- c) Modelos de decisão (KUNCHEVA; BEZDEK; DUIN, 2001) que calculam uma medida de similaridade entre o atual perfil de decisão da instância desconhecida e os perfis de decisão médios das instâncias de cada classe;
- d) Modelos que usam a regra de Dempster-Schafer (DEMPSTER, 1967) que calcula as medidas das crenças baseadas na plausibilidade para cada classe.

No entanto, muitos estudos empíricos mostraram que regras mais simples, como a regra da soma ou a votação da maioria ponderada, muitas vezes funcionam melhor que regras complexas (KITTLER et al., 1998). Em Kuncheva (2004), os autores demonstram uma visão detalhada destas e outras regras de combinação.

#### **2.4.4 Agrupador de Classificadores do Tipo *Stacking***

O agrupamento de classificadores do tipo *Stacking* (BREIMAN, Leo, 1996b; SAKKIS et al., 2001; WOLPERT, D. H., 1992) aprende um classificador de alto nível baseado nos classificadores base. Pode ser considerado como uma abordagem em que os classificadores de base são chamados classificadores de primeiro nível e um classificador de segundo nível é aprendido a combinar os classificadores de primeiro nível.

Figura 4 – Como classificadores são agrupados usando o Bagging.



Fonte: “Adaptado de” Robin Polikar (2006)

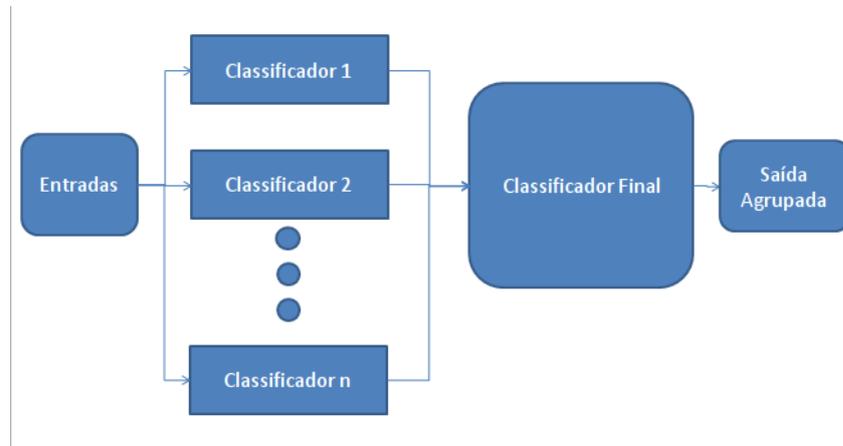
O método *Stacking*, ou, na tradução livre, empilhamento, é uma técnica de aprendizado conjunto usado para combinar vários modelos de classificação por meio de outro classificador. Os modelos de classificação individuais são treinados com base no conjunto completo de treinamento; então, o classificador final é ajustado com base nas saídas dos modelos de classificação individuais no conjunto. Esse classificador final pode ser treinado nas classes previstas ou probabilidades do conjunto (AGGARWAL, 2014).

Dois motivos decorrem do algoritmo 2 ser apresentado: o primeiro é devido ao seu uso para prova matemática do CMEAS a ser deduzida no capítulo 5 da proposta e a segunda porque foi usada no capítulo 6 de experimentos como comparação com os resultados do CMEAS

#### Algoritmo 2 – Algoritmo Stacking.

- 1 **Entrada:** Dados de treinamento  $D = \{x_i, y_i\}_{i=1}^m (x_i \in \mathbb{R}^n, y_i \in \gamma)$
- 2 **Saída:** Um classificador agrupado  $H$
- 3 Primeiro Passo: Aprenda classificadores do primeiro nível
- 4 **para**  $t \leftarrow 1$  **até**  $T$  **faça**
- 5   | Aprenda um classificador base  $h_t$  baseado em  $D$
- 6 **fim**
- 7 Segundo Passo: Construir nova base de dados a partir de  $D$
- 8 **para**  $i \leftarrow 1$  **até**  $m$  **faça**
- 9   | Crie nova base de dados contendo :
- 10   |  $\{x'_i, y_i\}$ , onde  $x'_i = \{h_1(x_i), h_2(x_i), \dots, h_T(x_i)\}$
- 11 **fim**
- 12 Terceiro Passo: Aprenda um classificador do segundo nível
- 13 Aprenda um novo classificador  $h'$  baseado na base de dados recém criada.
- 14 **retorna**  $H(x) = h'(h_1(x), h_2(x), \dots, h_T(x))$

Figura 5 – Como classificadores são agrupados usando o Stacking.



Fonte: “Adaptado de” Robin Polikar (2006)

Se for usado o mesmo conjunto de dados  $D$  para treinar classificadores de primeiro nível e para preparar os dados de treinamento para classificadores de segundo nível, isso pode levar a um *Overfitting*. Para resolver este problema, é incorporado a idéia de validação cruzada (KOHAVI et al., 1995) no *Stacking*. A validação cruzada é a técnica mais utilizada para avaliar o desempenho da classificação. Para avaliar a capacidade de previsão de um algoritmo de aprendizagem, é realizado o seguinte procedimento de validação cruzada: os dados de treinamento são subdivididos em  $K$  subconjuntos, o algoritmo de aprendizado é executado  $K$  vezes (DŽEROSKI; ŽENKO, 2004).

Cada classificador de subconjuntos  $K - 1$  aprendido é usado para prever o subconjunto restante e obter a precisão da aprendizagem. A precisão da predição final é obtida pela média entre as  $K$  execuções do algoritmo. A idéia de validação cruzada pode ser combinada com o *stacking* para evitar usar o mesmo conjunto de treinamento para a construção de classificadores de primeiro e segundo nível. Em vez de usar todos os exemplos de treinamento para obter classificadores de primeiro nível, agora os dados  $D$  são particionados em subconjuntos  $K$  para conseguir  $K$  classificadores, cada um dos quais é treinado apenas em  $K - 1$  subconjuntos. Cada classificador é aplicado ao subconjunto restante e a saída de todos os classificadores de primeiro nível constituem o espaço de características de entrada para o classificador de segundo nível. Observe que o *Overfitting* é evitado porque os dados para treinar os classificadores de primeiro nível e os dados para receber rótulos de classe previstos de classificadores do primeiro nível são diferentes.

Depois de treinar o classificador de segundo nível com base nos rótulos previstos dos classificadores de primeiro nível, é possível reestruturar os classificadores de primeiro nível em

todo o conjunto de treinamento  $D$ , para que todos os exemplos de treinamento sejam usados. A aplicação do classificador de segundo nível na saída atualizada dos classificadores de primeiro nível, dará a saída definitiva do agrupamento.

Os resultados empíricos (CLARKE, 2003) mostram que o *Stacking* possui um desempenho robusto que muitas vezes supera o modelo bayesiano médio. Desde a sua introdução (WOLPERT, D. H., 1992), o *Stacking* foi aplicado com sucesso a uma grande variedade de problemas, como regressão (BREIMAN, Leo, 1996b), estimativa de densidade (SMYTH; WOLPERT, D., 1999) e filtragem de spam (SAKKIS et al., 2001).

Além de usar rótulos de classe previstos de classificadores de primeiro nível, Ting e Witten (1999) considera o uso de probabilidades de classe como características. A vantagem de usar probabilidades condicionais como características é que o conjunto de treinamento do classificador de segundo nível incluirá não apenas previsões, mas também confiança de previsão de classificadores de primeiro nível. Os mesmos autores sugerem ainda o uso de regressão linear multi-resposta, uma variante de regressão linear de mínimos quadrados, como o algoritmo de aprendizado de segundo nível (TING; WITTEN, 1999).

A mistura de especialistas de Jordan e Jacobs (JACOBS, R. A. et al., 1991) gera vários classificadores especialistas cujas saídas são combinadas através de uma regra linear. Os pesos desta combinação são determinados por uma rede neural de gating, normalmente treinada usando o algoritmo de *maximização de expectativa* (*Expectation Maximization*) (EM) (DEMPSTER; LAIRD; RUBIN, 1977). Tanto os próprios especialistas quanto a rede de gating requerem instâncias da entrada para treinamento. Vários modelos de mistura de especialistas também podem ser combinados para obter uma mistura hierárquica de especialistas (JORDAN, M. I.; JACOBS, R. A., 1994). A mistura de especialistas é particularmente útil quando diferentes especialistas são treinados em diferentes partes do espaço de hipóteses, ou quando conjuntos heterogêneos estão disponíveis para ser usados para um problema de fusão de dados.

A equação genérica que representa esse tipo de agrupamento é mostrada na equação 17:

$$\bar{y}(x; \varrho) = \sum_{j=1}^p \varrho_j y_j(x) \quad (17)$$

Onde ,  $\bar{y}(x; \varrho)$  é a saída agrupada dado o conjunto de entradas  $x$  e um conjunto de pesos  $\varrho$ .  $y_j(x)$  é saída de cada especialista individualmente, assim como,  $\varrho_j$  o peso dado ao especialista.

Há também o *Feature-Weighted Linear Stacking* (SILL et al., 2009) que, também, combina classificadores usando combinação linear, e outros, que usam Lógica Fuzzy do tipo 1 (PULIDO, M.; MELIN; MENDOZA, 2017) e tipo 2 (PULIDO, M.; MELIN; MENDOZA, 2017; PULIDO, M.; MELIN; CASTILLO, 2013; PULIDO, M. E.; MELIN, 2012), partículas de enxame (swarm particles) (PULIDO, M.; MELIN; CASTILLO, 2014) e redes neurais que será melhor detalhado na seção 3.1

Além dessas técnicas outras foram propostas para gerar o modelo de agrupamento. Em (FRIEDMAN, J. H.; POPESCU et al., 2003) uma técnica chamada, *Aprendizado por Agrupamento por Importância de Amostras (Importance Sample Learning Ensemble)* (ISLE), foi fornecida para prover uma visão unificada dos métodos de agrupamento. Nessa estrutura um modelo de agrupamento é linear num espaço de alta dimensionalidade, no qual um ponto representa um modelo base e os coeficientes do modelo linear são os pesos dos modelos base.

A estrutura ISLE fornece uma forma de amostrar vários pontos no espaço de hipóteses e então aprender os coeficientes usando regressão linear. Baseado no ISLE Jerome H Friedman e Popescu (2008) continuou seus estudos propondo um conjunto de regras simples, onde cada regra consiste de um conjunto de afirmações baseado em valores de atributos, assim a vantagem desse método é que ela fornece um modelo mais interpretável comparado com outras abordagens de agrupamento.

A principal teoria desenvolvida para explicar o sucesso da aprendizagem em conjunto é baseada na decomposição do erro de distorção-tendência (DOMINGOS, 2000). Especificamente, o erro de previsão de um algoritmo de aprendizagem pode ser expresso como a soma de três erros não negativos: ruído intrínseco, viés e variância. O ruído intrínseco é um erro inevitável da tarefa, o desvio é a diferença entre a saída esperada e o valor verdadeiro, e a variância mede as flutuações nas previsões usando diferentes conjuntos de treinamento do mesmo tamanho.

Existe a análise da decomposição variância-variância em outros métodos de conjunto, como redes neurais (TUMER; GHOSH, 1996) e SVM (VALENTINI; DIETTERICH, 2004). A eficácia das abordagens de conjunto também pode ser explicada na perspectiva das margens de aprendizado, ou seja, as abordagens de conjunto podem ampliar a margem e, assim, melhorar as habilidades de generalização (MASON; BARTLETT; BAXTER, 2000).

Em (MELVILLE; MOONEY, 2005), um conjunto de modelos diversos são construídos usando exemplos de treinamento artificialmente construídos. Pode ser necessário remover

os modelos que são imprecisos ou redundantes, a nomenclatura para essa remoção dada pela literatura é poda de conjunto.

Em (KUNCHEVA; WHITAKER, 2003; TANG; SUGANTHAN; YAO, 2006), várias formas de diversidade entre os membros do conjunto de classificadores são estudadas e suas relações com a precisão da classificação são exploradas. Uma edição especial da revista *Information Fusion* foi dedicada à questão da diversidade na aprendizagem de conjunto (KUNCHEVA, 2005).

Essas abordagens são geralmente referidas como *Poda de Agrupamentos ou Poda de Conjuntos (Ensemble Pruning)* (EP), que pode ser grosseiramente categorizada nos seguintes grupos (ZHOU, 2012):

- a) Poda baseada em pedidos, que ordena os modelos de acordo com um determinado critério e, em seguida, seleciona os melhores modelos;
- b) Poda baseada em *cluster* ou grupos, que agrupa modelos semelhantes em cada cluster para manter um conjunto de modelos diversos e;
- c) Poda baseada em otimização, que tenta encontrar um subconjunto de classificadores otimizando uma determinada função objetiva em relação à capacidade de generalização do conjunto construído em classificadores selecionados.

A aprendizagem de conjuntos também foi estudada no contexto onde os dados chegam continuamente. Algumas abordagens de conjunto (GAO, J.; FAN; HAN, 2007; GAO, J. et al., 2007; KOLTER; MALOOF, 2005; SCHOLZ; KLINKENBERG, 2005; WANG, H. et al., 2003) foram desenvolvidas para lidar com derivações de conceito, ou seja, o fato de que as distribuições de dados mudam ao longo do tempo.

Muitas tarefas de classificação encontram o problema de desequilíbrio de classes, *class imbalance* em que os exemplos de uma classe dominam o conjunto de treinamento. Os classificadores aprendidos com este conjunto de treinamento terão bom desempenho na maioria dos casos, mas mal nas outras classes (WEISS; PROVOST, 2001).

Uma técnica efetiva para lidar com o desequilíbrio das classes é criar um novo treinamento amostrando mais exemplos minoritários ou amostrando menos exemplos majoritários (BATISTA; PRATI; MONARD, 2004; CHAWLA, N. V. et al., 2002).

Assim, essas técnicas de amostragem podem ser combinadas com abordagens de conjunto para melhorar o desempenho do classificador agrupado em exemplos de minorias (HIDO; KASHIMA; TAKAHASHI, Y., 2009; CHAWLA, N. et al., 2003; GAO, J. et al., 2007).

Os métodos de agrupamento também foram adaptados para lidar com a aprendizagem sensível à alguma medida de custo (DOMINGOS, 1999; FAN et al., 1999; VIOLA; JONES, 2002), em que a classificação errada em diferentes classes incorre em custos diferentes para cada classificador, sendo, o objetivo principal, minimizar o custo total combinado. Além disso, a aprendizagem em conjunto mostrou ser útil não só na classificação, mas também em muitas outras tarefas e aplicações de aprendizagem, como, por exemplo, a eficácia das técnicas de conjunto em filtragem colaborativa (KOREN; BELL; VOLINSKY, 2009).

Em tarefas de aprendizagem não supervisionadas, como o *clustering*, muitos métodos foram desenvolvidos para integrar várias soluções de cluster em uma única solução de cluster, encontrando consenso entre os modelos base (GHOSH; ACHARYA, 2011; STREHL; GHOSH, 2002).

O aprendizado multi-visão (CHRISTOUDIAS; URTASUN; DARRELL, 2012; BLUM; MITCHELL, 1998; SINDHWANI; NIYOGI; BELKIN, 2005; SINDHWANI; ROSENBERG, 2008), assume diferentes visualizações, compartilha funções de destino semelhantes e descobre os classificadores de múltiplas visualizações dos mesmos objetos, minimizando o erro de classificação e a inconsistência entre os classificadores.

#### **2.4.5 Técnicas de Agrupamento para Redes Neurais**

Essa seção foi descrita com a finalidade de apresentar outras técnicas de crescimento, combinação e neuro evolucionárias de redes neurais, além de apresentar redes siamesas e cápsulas, de maneira a situar e enquadrar o CMEAS.

Para o mapeamento de uma função, Alpaydin (1993) propõe a construção de vários modelos possíveis em vez de um, treiná-los de forma independente na mesma tarefa e votar sobre a probabilidade de suas respostas. Essas redes convergem para diferentes soluções devido ao uso de diferentes modelos, diferentes tamanhos de conjuntos de parâmetros ou qualquer outro fator relacionado ao treinamento. Nele é utilizado *Algoritmos de Crescimento e Aprendizado (Grow and Learn Algorithms) (GAL)* (ALPAYDM, 1990), sendo investigados diversos esquemas de votação. Em Ruta e Gabrys (2007) é proposto um modelo composto de previsão de séries temporais que combina redes neurais, aplicadas individualmente à subconjuntos de características deslocados no tempo. Em Smith e Jin (2014) foi proposto um algoritmo evolutivo multi-objetivo híbrido que treina e otimiza a estrutura das redes neurais recorrentes para a previsão de séries temporais. O algoritmo EpNet (YAO; LIU, Y., 1996) foi utilizado em

(LANDASSURI-MORENO; BULLINARIA, 2009) para previsão de séries temporais na bolsa de valores do Canadá. Kordík et al. (2010) forneceu um algoritmo, denominado GAME, para efetuar conexões que ultrapassam camadas internas da rede, através de agrupamento, indução e auto-organização.

O crescimento de rede apresentado em (ALPAYDM, 1990) é reportado em diversos artigos como sendo um *Agrupamento de redes neurais (Neural Network Ensemble)* (ENN), ou seja, um agrupamento ou montagem de redes neurais, pois altera a estrutura da mesma, adicionando uma ligação e uma unidade neuronal. Isso é feito para que essa rede tenha capacidade de aumentar a dimensionalidade da fronteira de decisão e possa classificar as classes as quais está sendo submetida (YAO, 1999). Para evitar distorções na interpretação da palavra em inglês, técnicas desse tipo serão referenciadas como sendo de crescimento de rede ou neuro evolução. Mais uma definição é necessária nesse ponto, já que existe uma diferenciação entre treinamento usando técnicas evolucionárias e técnicas de crescimento ou alteração de topologias. Uma questão importante na neuro evolução ou crescimento é como conseguir alguma vantagem com a evolução das topologias e dos pesos da rede neural. Em Stanley e Miikkulainen (2002) foi apresentado um método chamado *Neuro Evolução por Aumento de Topologia (Neural Evolution of Augmenting Topologies)* (NEAT). Nele foi empregado um método de onde diferentes topologias eram testadas usando algoritmos genéticos. Esse estudo é classificado na literatura como sendo da área de *Neuro Evolução (Neural Evolution)* (NE).

Em David B Fogel, Lawrence J Fogel e Porto (1990) foi proposto o uso de programação evolucionária para treinar redes neurais, enquanto em Alpaydin (1993), Hansen e Salamon (1990), Perrone e Cooper (1993), Zaamout e John Z Zhang (2012) e Jiménez (1998) foram usados agrupamentos, e, além disso Alpaydm (1990) altera a estrutura da mesma conforme necessário, também chamado de aprendizado incremental.

Em Andrews, Diederich e Tickle (1995) há uma revisão fornecendo uma taxonomia de definições para posicionar ou classificar as diversas técnicas existentes até sua publicação, fornecendo inclusive alguns critérios para avaliar a eficácia das mesmas.

Para o funcionamento de algoritmos de Neuro Evolução, o ambiente não precisa satisfazer nenhuma restrição em particular, podendo ser contínuo e parcialmente observável. Se o ambiente contiver uma regularidade suficiente para que uma determinada tarefa seja resolvida, assim como, se a modelagem dos fenótipos forem suficientemente corretas, então o NE pode encontrar uma solução (GOMEZ, F. J., 2003).

Em Faustino Gomez, Schmidhuber e Miikkulainen (2008) um método de neuro evolução denominado *Evolução Neural por cooperação de sinapses (Cooperative Synapse Neuro Evolution)* (CoSYNE), utilizou a coevolução cooperativa dos pesos sinápticos individuais para resolver o problema do pêndulo invertido.

Em Miikkulainen et al. (2017), os métodos de neuro evolução existentes foram expandidos para alterar a topologia, testar componentes e hiperparâmetros em uma rede neural profunda, sendo, escolhida como aplicação, a inserção de legendas em imagens como parte de um sistema automatizado de um site de revista.

Em (ROSEN, 1996) as redes individuais são treinadas por backpropagation não apenas para reproduzir uma saída desejada, mas também para ter seus erros linearmente relacionados com as outras redes. As saídas das redes individuais são então combinadas linearmente para produzir a saída da rede de conjunto. Os resultados empíricos mostram que quando as redes individuais são forçadas a discorrer entre si, o conjunto resultante apresenta erros quadrados médios mais baixos do que as redes de conjunto com redes individuais treinadas de forma independente.

Um trabalho similar foi apresentado por (LIU, Y.; YAO, 1999) com a proposta de um sistema de agrupamento cooperativo chamado *Aprendizado por Agrupamento de Cooperação de sinapses (Cooperative Synapse Ensemble Learning)* (CELS), nele as redes são treinadas simultaneamente e são passíveis de uma penalização criada pela interação entre redes podendo-se criar redes negativamente correlacionadas. O CELS foi testado no problema de previsão da série temporal Mackey-Glass e no problema de avaliação de cartas de crédito australiano.

Em (OPITZ; SHAVLIK, 1996) é apresentada uma técnica chamada *Algoritmo de adição de classificadores (Add Them Up)* (ADDEMUP), que usa algoritmos genéticos para procurar diretamente um conjunto preciso e diversificado de redes treinadas. O ADDEMUP funciona criando uma população inicial e, em seguida, usa operadores genéticos para criar continuamente novas redes, mantendo o conjunto de redes que são tão precisas quanto possível enquanto diversificam-se umas das outras.

Em Yuehui Chen, Bo Yang e Ajith Abraham (2007) usaram uma abordagem de crescimento de redes em árvores flexíveis com o uso de *Algoritmo de otimização de partículas de enxame (Particle Swarm Optimization)* (PSO) substituindo nós folha, representados por neurônios, por outros nós ou subárvores, para tal, foram analisados dados de 7 anos do índice Nasdaq-100 e 4 anos do *Índice da NSE (NIFTY)*.

Redes neurais siamesas, são, como o próprio nome sugere, idênticas, ou seja possuem os valores dos parâmetros entre si compartilhados, sendo a saída dessas redes conectadas no estilo agrupamento.

As redes neurais siamesas podem ser usadas para aplicações de reconhecimento ou verificação, onde o número de categorias é muito grande e não conhecido e onde o número de amostras de treinamento para uma única categoria é muito pequeno.

Em Bromley et al. (1994) foi proposta o que os autores chamaram de *Rede Neural Siamesa com Deslocamento Temporal (Siamese Time Delay Neural Network)* (STDNN), nele é descrito um algoritmo para verificação de assinaturas escritas em um tablet. O algoritmo baseia-se em uma rede neural artificial siamesa. Esta rede consiste em duas sub-redes idênticas conectadas em suas saídas. Durante o treinamento, as duas sub-redes extraem características de duas assinaturas, enquanto um neurônio de junção mede a distância entre os dois vetores de características. A verificação consiste em comparar um vetor de características extraído, com um vetor de características armazenado para o signatário. As assinaturas mais próximas dessa representação armazenada que um período escolhido são aceitas, todas as outras assinaturas são rejeitadas. Na verdade as redes são idênticas e são treinadas com o mesmo conjunto de pesos  $w$ , para encontrar a similaridade entre dados e não entre redes, ou seja, propuseram um método para treinar uma métrica de similaridade a partir dos dados.

A técnica proposta por Chopra, Hadsell e LeCun (2005) de redes neurais siamesas, usou a técnica de Bromley et al. (1994) para uma tarefa de verificação de faces. O processo de aprendizagem minimiza uma função de perda discriminativa que faz com que a métrica de utilidade seja pequena para um par de faces da mesma pessoa e grande para um par de faces de pessoas diferentes. O mapeamento é feito por uma *Redes Convolucionais Profundas (Deep Convolutional Neural Networks)* (DCNN), cuja, arquitetura é projetada para ser robusta à distorções geométricas. O sistema é testado no banco de dados facial Purdue / AR.

Em um artigo publicado por Sabour, Frosst e Geoffrey E Hinton (2018a) é apresentado o conceito de cápsulas. No artigo, as cápsulas são definidas como sendo um grupo de neurônios, cujo vetor de atividade representa os parâmetros de instanciação de um tipo específico de entidade, como um objeto ou parte dele. Eles utilizaram o comprimento de um vetor de atividade para representar a probabilidade de que a entidade exista e sua orientação para representar os parâmetros de instanciação. As cápsulas ativas de um nível fazem previsões, através de matrizes de transformação, para os parâmetros de instanciação de cápsulas de nível superior. Quando várias previsões concordam, uma cápsula de nível superior torna-se ativa. Os autores Sabour,

Frosst e Geoffrey E Hinton (2018a), mostram que um sistema de cápsulas multicamadas, com uma matriz de pesos entre camadas, com treinamento discriminado atinge o desempenho de última geração no MNIST e é consideravelmente melhor do que uma rede convolucional ao reconhecer dígitos altamente sobrepostos.

No artigo a matriz de pesos entre capsulas é chamado de *Agreement Coeficients*. Seguindo essa mesma linha, um artigo a ser publicado por Sabour, Frosst e Geoffrey E Hinton (2018b), criaram um roteamento de informações passando pelas matrizes entre cápsulas ativando partes específicas das cápsulas superiores, que segundo os autores são altamente relacionadas Sabour, Frosst e Geoffrey E Hinton (2018b). Essa alta relação é determinada com o uso do algoritmo *Expected Maximization*. Apesar de haver uma matriz de pesos interconectando as cápsulas estão sendo usadas para uma mesma área, no caso processamento de imagens.

## 2.5 Conclusão

Esse capítulo apresentou a inspiração biológica da proposta deste trabalho e introduziu, ainda que de forma breve, alguns conceitos de redes neurais artificiais, passando pelo o algoritmo de treinamento da retropropagação do erro, que apesar de seu forte embasamento matemático e de ser consolidado matematicamente à um certo tempo, foi necessário porque duas das equações apresentadas na dedução do backpropagation serão usadas na proposta. Nesse capítulo, também foram apresentadas diversas técnicas de agrupamento, passando por sua fundamentação teórica e, também, como o aumento de redes neurais e sua combinação são utilizadas para melhorar a classificação em apenas uma área ou apenas uma função, tornando-as especialistas. A fundamentação teórica do agrupamento do tipo *Stacking* será particularmente importante para a definição da proposta, mais especificamente nas relações da equação 17.

### 3 Aprendizado Profundo e Aprendizado por Reforço Profundo

Neste capítulo serão introduzidos, de forma breve, conceitos de aprendizado profundo e aprendizado por reforço profundo. O capítulo se desenrola apresentando o aprendizado profundo, particularmente para redes neurais convolucionais, e continua passando pelo aprendizado por reforço até a sua forma profunda com generalização de estados por meio de redes neurais. Alguns dos principais algoritmos de aprendizado profundo e por reforço também serão revisados focando na proposta deste trabalho, a ser apresentada no capítulo 5.

#### 3.1 Aprendizado Profundo

Abstrações de alto nível podem estar codificadas em funções complexas. Essas funções não são facilmente decifradas, e para aprendê-las pode ser necessário usar arquiteturas profundas. Arquiteturas profundas são compostas de vários níveis de operações não-lineares, como em redes neurais com várias camadas ocultas ou em fórmulas proposicionais reutilizando muitas sub-fórmulas (LECUN; BENGIO; HINTON, G., 2015).

Ao usar arquiteturas profundas se faz necessário pesquisar o espaço de parâmetros dessa arquiteturas. Essa é uma tarefa difícil, que apenas recentemente foram solucionados com a proposta de algoritmos, tais como as *Máquinas de Boltzman Profundas (Deep Boltzman Machines)* (DBM) (HINTON, G. E., 2009; GOODFELLOW; BENGIO; COURVILLE, 2016; HINTON, G. E.; OSINDERO; TEH, 2006; HINTON, G. E.; SALAKHUTDINOV, 2006; DENG, L.; YU, D. et al., 2014).

O *Aprendizado Profundo (Deep Learning)* (DL) é um conjunto de técnicas de aprendizado de máquina, sendo um modelo composto de múltiplas camadas para aprender a representação de dados em múltiplos níveis de abstrações, onde características de alto-nível são definidas a partir de características de baixo-nível (LECUN; BENGIO; HINTON, G., 2015; DENG, L.; YU, D. et al., 2014).

Em 2014 um programa criado por cientistas suíços superou um ser humano no reconhecimento de imagens de sinais de trânsito (CIRESAN; MEIER; SCHMIDHUBER, 2012). O programa identificou com precisão 99.46% das imagens em um conjunto de 50.000; a pontuação máxima em um grupo de 32 participantes humanos foi de 99.22% e a média para os seres humanos era de 98.84% (CIRESAN; MEIER; SCHMIDHUBER, 2012).

Em outubro de 2012, Richard F. Rashid, um coordenador de programas científicos da Microsoft apresentou em uma conferência em Tianjin, China uma tecnologia de tradução simultânea do Inglês para Mandarim acompanhado por uma simulação de sua própria voz (MICROSOFT, 2012).

As tecnologias de aprendizado profundo, são amplamente utilizados em diferentes sistemas, incluindo o assistente pessoal inteligente Apple Siri com base nas tecnologias da Nuance Communications e no reconhecimento de endereços no Google Street View.

Esse paradigma de aprendizagem implementa a idéia de aprendizagem em duas etapas. Na primeira fase, a informação sobre a estrutura interna dos dados de entrada é extraída a partir de uma grande variedade de dados não formatados, isso é feito através de um treinamento camada-por-camada não supervisionado. Em seguida, usando esta informação em uma rede neural multi-camada, ele passa por treinamento supervisionado pelos métodos conhecidos.

No aprendizado profundo foram desenvolvidas técnicas e arquiteturas para aprendizado supervisionado, não supervisionado e aprendizado por reforço. Algumas das técnicas de aprendizado profundo não supervisionado são: *As Máquinas de Boltzman Profundas (Deep Boltzman Machines)* (DBM) (DENG, L.; YU, D. et al., 2014; GOODFELLOW; BENGIO; COURVILLE, 2016); *Máquinas de Boltzman Restritas (Restricted Boltzman Machines)* (RBM) (DENG, L.; YU, D. et al., 2014; GOODFELLOW; BENGIO; COURVILLE, 2016); *Auto Associadores Profundos (Deep Auto Encoders)* (DAC) (DENG, L.; YU, D. et al., 2014; GOODFELLOW; BENGIO; COURVILLE, 2016) que foram usados no treinamento baseado em propagação de alvos.

Segundo LeCun, Bengio e Geoffrey Hinton (2015) o desempenho das *Redes Neurais Profundas (Deep Neural Networks)* (DNN) tornou-se possível com o uso de novas técnicas tais como: ReLUs (LECUN; BENGIO; HINTON, G., 2015), Dropout (SRIVASTAVA et al., 2014), e também realizando-se o treinamento em placas de processamento gráfico (GPU - do inglês Graphics Process Unit).

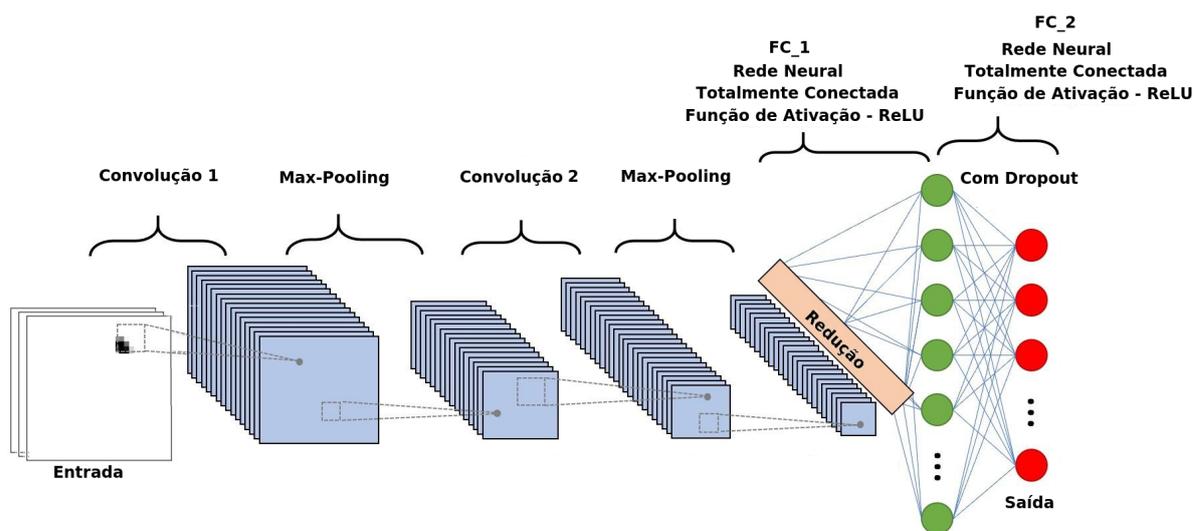
Desenvolvidas por LeCun, Bengio et al. (1995), as *Redes Neurais Convolucionais, (Convolutional Neural Networks)* (CNN), utilizam uma operação linear chamada convolução que condensa características contidas nos exemplos nas entradas da rede. Em Goodfellow, Bengio e Courville (2016), o uso de redes convolucionais, no seu formato profundo, foram abordadas para uso na classificação de imagens, reconhecimento de voz e séries temporais.

No método de convolução aplicam-se vários filtros chamados de *kernels*. Esse tipo de rede usa também operações como o *Pooling*, *Dropout*, *Softmax*. O *Pooling* é uma operação que

extrai um valor máximo de um *kernel* de um *pooling*. Da mesma forma que a convolução, o *kernel* é passado em todo o quadro, logo após a saída da convolução. Assim, uma típica rede convolucional possui quatro estágios:

- a) No primeiro estágio realizam-se várias convoluções em paralelo, isso faz com que os neurônios fiquem em contato com regiões locais dos dados de entrada, no caso específico de processamento de imagens um certo kernel de pixels, no caso de *Processamento de Linguagem Natural (Natural Language Processing) (NLP)*, matrizes de palavras de um texto.
- b) No segundo estágio o *Pooling*, que como já foi mencionado, reduz o uso de recursos computacionais, mas também serve para reduzir progressivamente o tamanho espacial da representação, reduzir o número de parâmetros e, portanto, também ajuda a controlar o *overfitting*.
- c) No terceiro estágio uma camada ReLU que aumenta as propriedades não-lineares da função de decisão e da rede global sem afetar os resultados da camada de convolução. Esse tipo de função de ativação é usada, pois Krizhevsky, Sutskever e Geoffrey E Hinton (2012) argumenta que essa função permite um treinamento muitas vezes mais rápido que outras funções de ativação.
- d) No quarto estágio são neurônios completamente conectados,

Figura 6 – Um Exemplo de Rede Neural Convolucional.



Fonte: “Adaptado de” LeCun, Bengio e Geoffrey Hinton (2015)

O uso da operação *max-pooling*, que extrai somente o maior valor do *pooling*, faz com que a memória demandada durante o treinamento seja menor, o tempo de *feed-forward* e *feed-backward* também torna-se menor.

Já o Dropout (SRIVASTAVA et al., 2014) é uma técnica para reduzir o *overfitting* nas redes neurais. As redes neurais profundas contêm múltiplas camadas ocultas não-lineares de neurônios completamente conectados. Isso permite que a rede aprenda diversos padrões. No entanto, quando se utiliza um conjunto de dados de treinamento limitado, acaba resultando em *overfitting* (SRIVASTAVA et al., 2014).

A ideia do *Dropout* é descartar aleatoriamente e momentaneamente uma porção de neurônios da rede neural quando se realiza a retropropagação, então esses neurônios voltam para a rede, e o passo se repete novamente descartando-se aleatoriamente uma nova porção de neurônios para realizar novamente a retropropagação (SRIVASTAVA et al., 2014).

A função *Softmax* é usada em vários métodos de classificação, como a regressão logística multinomial, análise discriminante linear de classes, classificadores Bayes ingênuos e redes neurais artificiais (BISHOP, 2006). Essa função é uma generalização da função logística que converte um vetor  $z$  de dimensão  $k$  de valores reais arbitrários, para um vetor  $\sigma(z)$  de  $j$  valores reais que variam de 0 a 1, conforme 18:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (18)$$

Diversas definições de camadas são utilizadas na literatura, alguns autores consideram, como sendo, uma camada, cada um dos estágios supracitados, outros consideram a junção de convoluções, *Pooling* e ReLU, como sendo uma única camada.

Apesar da possibilidade redes convolucionais possuírem invariância de localização e a composicionalidade local no caso de imagens, essas características não são muito úteis para resolver o NLP, isso porque é muito importante onde a palavra aparece na frase. Pixels próximos uns dos outros são provavelmente semanticamente correlacionados, ou seja, parte de um mesmo objeto ou região, mas o mesmo não acontece com a linguagem natural, em muitas línguas, inclusive, partes de frases podem ser separadas por várias outras palavras (KIM, Y., 2014). O aspecto da composição também não é óbvio. Claramente, as palavras se complementam de certa forma, como um adjetivo que modifica um substantivo, mas como exatamente isso funciona, o que as representações de nível superior realmente significam não é tão óbvio quanto no caso da visão de computador (WANG, P. et al., 2015).

As redes neurais recorrentes fazem sentido mais intuitivo (MIKOLOV et al., 2010). Elas se assemelham à forma como processamos o idioma ou pelo menos como pensamos que processamos o idioma (MIKOLOV et al., 2010). Para modelagem de problemas de NLP, as DCNN apresentaram desempenhos satisfatórios (MICROSOFT, 2012).

Em Yoon Kim (2014) é avaliada uma arquitetura DCNN em vários conjuntos de dados de classificação, principalmente composta por tarefas de análise de sentenças e categorização de tópicos. A arquitetura DCNN consegue um desempenho muito bom em alguns conjuntos de dados, e um novo estado-da-arte em outros. A camada de entrada é uma frase composta de palavras concatenadas no formato word2vec. Isso é seguido por uma camada convolucional com vários filtros, então uma camada de maxpooling e, finalmente, uma função Softmax (BISHOP, 2006).

O artigo também efetua experimentos usando dois canais diferentes na forma de incorporações de palavras estáticas e dinâmicas, onde um canal é ajustado durante o treinamento e o outro não. Uma arquitetura semelhante, mas um pouco mais complexa, foi previamente proposta em (KALCHBRENNER; GREFFENSTETTE; BLUNSOM, 2014). Em Peng Wang et al. (2015) é incluída uma camada adicional, que executa um agrupamento semântico, nesta arquitetura de rede.

Uma grande vantagem que as DCNN apresentam é a sua velocidade de execução. As convoluções, que são a parte central da DCNN, são implementadas em um nível de hardware em GPUs, além disso, também são eficientes em termos de representação de n-gramas (KIM, Y., 2014).

Como foi mencionado anteriormente redes neurais recorrentes profundas fazem mais sentido para o NLP (GRAVES; MOHAMED; HINTON, G., 2013), mas também foram implementadas com sucesso na predição de séries temporais, como por exemplo, em redes recorrentes (TINO; SCHITTENKOPF; DORFFNER, 2001), e também em sua versão profunda (PRASAD, S. C.; PRASAD, P., 2014).

Mesmo redes neurais profundas são treinadas em apenas uma área de atuação se tornando especialistas nessa área, generalizando as saídas com ótimos resultados, porém, ainda existem problemas que são muito complexos para redes neurais, principalmente aqueles que englobam várias áreas.

### 3.2 Aprendizado Por Reforço

A aprendizagem de reforço é uma abordagem computacional para entender e automatizar a aprendizagem direcionada para objetivos e a tomada de decisões (WATKINS, 1989). Distingue-se de outras abordagens computacionais por sua ênfase na aprendizagem pela interação direta com o ambiente.

Em cada passo de tempo discreto  $t$ , o agente detecta o estado atual  $s_t$ , escolhe uma determinada ação  $a$ , e a executa.

O ambiente responde dando ao agente uma recompensa  $r_{t+1} = r(s_t, a_t)$  e produzindo o estado seguinte  $s_{t+1} = \delta(s_t, a_t)$ . Aqui, as funções  $\delta$  e  $r$  são a parte do ambiente e não são necessariamente conhecidas pelo agente.

No *Processo de decisão de Markov (Markov Decision Process)* (MDP), as funções  $\delta(s_t, a_t)$  e  $r(s_t, a_t)$  dependem apenas do estado atual e da ação executada e não em estados ou ações anteriores. A tarefa do agente é escolher uma política,  $\pi : S \rightarrow A$ , onde  $S$  é o conjunto de estados e  $A$  é o conjunto de ações, para selecionar sua próxima ação com base nos estados observados atuais  $s$ ; isto é,  $\pi(s) = a$ . Uma política ótima é uma política que maximiza a possível recompensa de um estado, chamado valor,  $V^\pi(s)$ , para todos os estados (WATKINS, 1989).

$$V^\pi(s_1) = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} \quad (19)$$

Onde  $(0 < \gamma < 1)$  é uma constante que determina o valor relativo das recompensas atrasadas versus imediatas.  $V^\pi(s)$  significa a possível recompensa cumulativa obtida seguindo uma política arbitrária  $\pi$  de um estado inicial arbitrário. Então, uma política ótima,  $\pi^*$ , é definida da seguinte forma:

$$\pi^* = \arg \max_{\pi} V^\pi(s), (\forall s) \quad (20)$$

#### 3.2.1 O método das Diferenças Temporais – TD-Learning

O *Aprendizado por Reforço por Diferença Temporal (Reinforcement Learning- Temporal Difference)* (TD) (SUTTON; BARTO, 1998) foi um dos primeiros algoritmos de aprendizado por reforço desenvolvido a ter uma base matemática consistente. Foi proposto como uma

versão adaptativa do algoritmo de Iteração. Um processo iterativo, denominado *Iteração de Política Generalizada (Generalized Policy Iteration) (GPI)*, é necessário para obter  $\pi^*$

Como os algoritmos de Monte Carlo, os algoritmos TD podem aprender diretamente da experiência sem um modelo da dinâmica do meio ambiente. Como a programação dinâmica, os algoritmos TD são algoritmos de bootstrapping, ou seja, eles baseiam sua atualização em parte em uma estimativa existente. Afinal, os algoritmos TD combinam a amostragem de algoritmos de Monte Carlo com o bootstrapping de algoritmos de programação dinâmica.

Os algoritmos TD têm uma vantagem sobre os algoritmos de programação dinâmica, na medida em que não requerem um modelo do meio ambiente, da sua recompensa e nem das distribuições de probabilidade do próximo estado (WATKINS, 1989).

No passo de tempo  $t + 1$  ele imediatamente forma um alvo e faz uma atualização útil usando a recompensa observada  $r_{t+1}$  e a estimativa  $V_{(s_{t+1})}$

O algoritmo de diferenças temporais mais simples é conhecido como TD(0) e sua equação :

$$V(s_t) \leftarrow V(s_t) + \alpha.[r_{t+1} + \gamma.V(s_{t+1}) - V(s_t)] \quad (21)$$

A forma procedural segue o algoritmo 3

#### Algoritmo 3 – Algoritmo TD(0).

```

1 Inicialize  $V(s)$  arbitrariamente,  $\pi$  sendo a política a ser avaliada
2 repita
3   para cada episódio faça
4     Inicialize  $s$ 
5     repita
6       para cada passo do episódio faça
7          $a \leftarrow$  ação dada por  $\pi$  para  $s$ 
8         Execute a ação  $a$ 
9         recompensa  $r$  e próximo estado  $s'$ 
10         $V(s_t) \leftarrow V(s_t) + \alpha.[r_{t+1} + \gamma.V(s_{t+1}) - V(s_t)]$   $s \leftarrow s'$ 
11       fim
12     até fim dos passos;
13   fim
14 até  $s$  ser estado terminal;

```

### 3.2.2 Q-Learning

Diferentemente do método das diferenças temporais, que considera apenas o valor do estado, o algoritmo Q-learning (WATKINS, 1989) determina uma função Q (par de estado-ação). A Equação 22, mostra a equação do Q-learning.

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha [r(s_t, a_t) + \gamma \max_{a'_t} Q_t(s'_t, a'_t) - Q_t(s_t, a_t)] \quad (22)$$

Onde  $s_t$  é o estado atual,  $a_t$  é a ação que será realizada estando em  $s$ ,  $r(s_t, a_t)$  é o reforço ao se realizar uma ação  $a$  estando no estado  $s$ ;  $s'_t$  é o estado futuro observado;  $a'_t$  é a ação a ser realizada no estado  $s'$ ,  $\gamma$  é o fator de desconto que deve estar entre 0 e 1 e, finalmente  $\alpha$  que é taxa de aprendizado  $0 \leq \alpha < 1$ .

O Q-learning precisa de muitos episódios para se aproximar ou encontrar a política ótima e isso é causado pela superestimação da função valor Q (HASSELT, 2010). Para sobrepujar essa limitação Hasselt (2010) propôs um algoritmo chamado Double Q-Learning que possui duas funções valor:  $Q_A$  e  $Q_B$ , onde cada função é atualizada com o valor da outra função.

(VAN HASSELT; GUEZ; SILVER, 2016) A Tese de doutorado de Lin (1993) combinou o aprendizado por reforço com redes neurais, assim foi possível aplicar este método em ambientes mais complexos. O objetivo do trabalho foi usar uma rede neural de multicamadas com back-propagation para generalização dos estados similares abstraíndo-os e aplicando o algoritmo TD, isso ajuda a reduzir o tempo necessário para o agente aprender a tarefa

Lin (1993) também propôs a técnica chamada experience replay que armazena as experiências passadas. No caso de um espaço de estados contínuos, uma rede neural é frequentemente usada para aproximar o valor  $Q(s, a)$ . Esta rede é frequentemente referida como uma *Rede Q (Q-Network)* (QN) (MNIH et al., 2013). Uma *Q-network* pode ser treinada minimizando o erro de predição de  $Q$ .

Os algoritmos de aprendizado por reforço necessitam de um número de iterações muito grande para sua convergência, como consequência, à medida que aumenta o espaço de estados e o número de ações, aumenta-se a necessidade de um número maior de iterações para sua aprendizagem, aumentando significativamente o tempo de convergência do algoritmo, tornando em alguns casos o processo de aprendizado extremamente lento.

Dessa forma para minimizar esse problema, algumas propostas foram apresentadas, tais como: melhorar o aproveitamento das experiências por meio de generalizações temporais, es-

paciais ou das ações; o uso de aceleração por abstração temporal ou espacial e a aceleração baseada em casos (**Donandon2020cases**).

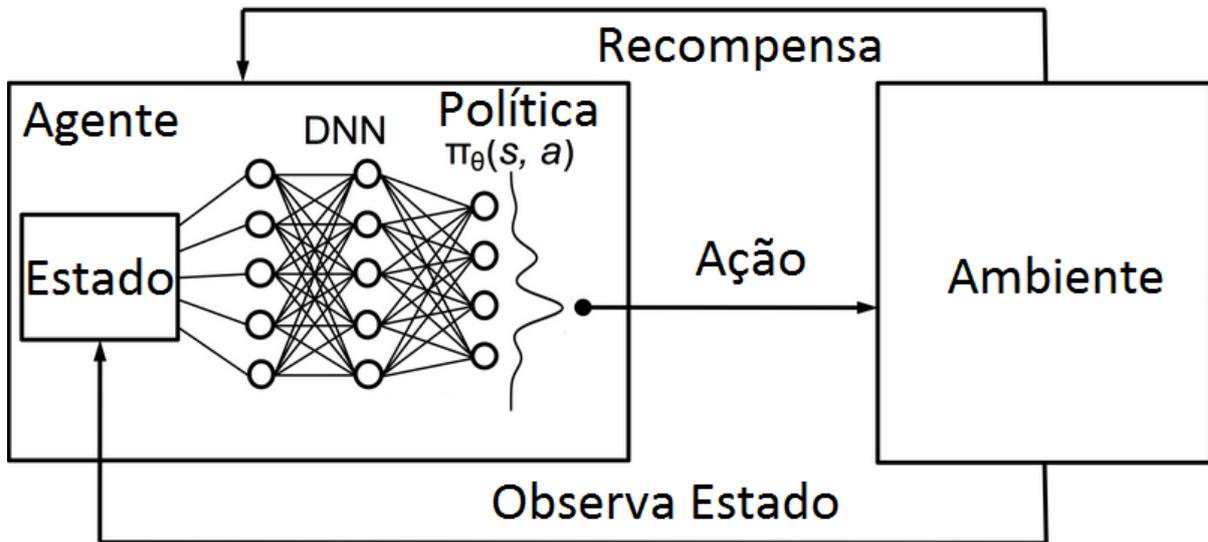
### 3.3 Aprendizado por Reforço Profundo

Lange e Riedmiller (2010) apresentaram um algoritmo de Aprendizado por Reforço Profundo usando *Auto Associadores Profundos (Deep Auto Encoders)* (DAC) que aprendia políticas de ações usando os pixels da imagem. Eles testaram o algoritmo em um mundo de grades, sendo que o agente deve aprender a chegar ao objetivo a partir de qualquer posição do mapa. Este modelo utiliza a rede neural de autoencoder para diminuir o espaço de estados da imagem de entrada, onde o algoritmo de aprendizado por reforço aprende a partir do espaço de estados generalizado pela rede neural de autoencoder.

Recentemente, o Q-learning baseado em modelos neurais profundos, também conhecido como Q-learning profundo, foi aplicado com êxito a algumas tarefas desafiadoras, como jogos e movimentos robóticos. Em (WANG, Y. et al., 2017), foi empregado o Q-learning profundo para construir um sistema que os autores nomearam Q-trading profundo. O Q-trading, segundo os autores, pode determinar automaticamente a posição a manter em cada hora de negociação. Nos resultados experimentais, os autores mostram que o sistema Q-trading profundo pode superar a estratégia de compra e retenção *Buy and Hold*, estratégia essa que confia na valorização do papel a longo prazo, bem como a estratégia aprendida pelo *Aprendizado por reforço recorrente, (Recurrent Reinforcement Learning)* (RRL) que em outros artigos eram mais eficazes do que o Q-learning, como é o caso de artigos como (DENG, Yue et al., 2017) e (PRASAD, S. C.; PRASAD, P., 2014). O relatório técnico apresentado por Yang Wang et al. (2017), não foi publicado em anais de renomadas revistas ou jornais. Na figura 7 é possível ver um exemplo de aprendizado por reforço profundo que usa uma rede neural para generalizar os estados de um ambiente contínuo.

A partir do DQN outros algoritmos foram apresentados, tais como: *Gorila DQN (Gorila Deep Q-Network)* (GDQN) (NAIR et al., 2015); *DQN dupla (Q-Network)* (DDQN) (VAN HASSELT; GUEZ; SILVER, 2016); *Aprendizado Q profundo com repetição de Experiência Prioritária (Deep Q-Learning with Prioritized Experience Replay)* (DQLPER) (SCHAUL et al., 2015); *Gradiente de Política Determinística (Deep Deterministic Policy Gradient)* (DDPG) (LILLICRAP et al., 2016); *Redes Q profundas Contínuas (Continuous Deep Q-Networks)* (CDQN) (GU, S. et al., 2016).

Figura 7 – Um Exemplo do uso de aprendizado por Reforço Profundo usando uma rede neural para aproximar os estados contínuos provindos do ambiente.



Fonte: “Adaptado de” Mnih et al. (2013)

A aprendizagem de reforço foi aplicada na análise financeira tradicional e em diversos tipos de investimento por uma grande quantidade de pesquisadores. Por exemplo, Moody et al. (1998) propuseram um algoritmo para otimizar portfólios de segurança. Xiu Gao e Chan (2000) usou o *Coefficiente Sharpe - lucro ajustado pelo risco relativo (Sharpe Ratio)* (Sharpe), como função de desempenho para treinar o sistema de negociação com base no Q-learning. Du, Jinjian Zhai e Lv (2016) compararam o desempenho do Q-learning na forma de *Aprendizado por reforço recorrente, (Recurrent Reinforcement Learning) (RRL)*, e relataram que o RRL obteve um melhor desempenho na negociação de papéis.

Em (DENG, Yue et al., 2017) foi apresentada uma rede neural recorrente para representação e negociação de sinais financeiros em tempo real, chamada de *Deep Direct Reinforcement - DDR*. Na estrutura proposta, a parte DL detecta automaticamente a condição dinâmica do mercado para o aprendizado de características. Então, o módulo RL interage com representações profundas e toma decisões comerciais para acumular as recompensas. Esse formato, segundo os autores apresenta características presentes nas arquiteturas profundas e recorrentes. O DDR foi testado com o índice SP500, assim como outros índices que o influenciam como os índices FTSE100 da Inglaterra, Hangseng da bolsa de valores de Hong Kong, Nikkei 225 da bolsa do Japão e da bolsa de Shanghai na China.

### 3.3.1 Deep Q-Network

A DQN desenvolvida por (Mnih et al., 2013) é composta por três camadas de redes neurais convolucionais seguida por duas camadas com redes completamente conectadas, sendo a saída da rede as ações possíveis.

Usa-se a descida de gradiente do *Erro Quadrático médio (Mean Squared Error)* (MSE) para treinar a rede e ajustar os valores dos pesos  $w$  pelo back-propagation, conforme 23. Esse algoritmo é aplicado na função  $L$  para cada iteração  $i$ , onde  $\theta_i^- = \theta_{i-1}$  e  $\gamma$  é o fator de desconto.

$$\nabla_{w_i} L(w_i) = (r_j + \gamma \max_{a_t} Q(s_{j+1}, a; w_{i-1}) - Q(s, a; w_i)) \nabla_{w_i} Q(s, a; w_i) \quad (23)$$

O termo  $r_j + \gamma \max_{a_t} Q(s_{j+1}, a; w_{i-1})$  é frequentemente chamado de  $y^{DQN}$ . Segundo os autores esse algoritmo sofre de divergências ou oscilações na política. Para solucionar esse problema realizaram a cópia dos valores de  $Q$  para um  $\hat{Q}$  a cada  $c$  passos e utilizaram esse novo  $\hat{Q}$  para atualizar o termo  $y^{DQN}$ . Com essa alteração o algoritmo se beneficia de maior estabilidade, já que essas oscilações se tornam menos comuns. (Mnih et al., 2015)

A sequência de ações e observações é o estado  $s_t$ , onde, no domínio apresentado, cada ação e observação é um passo, assim:  $s_t = x_1, a_1, x_2, a_2, x_3, a_3, \dots, x_t$ , sendo o tempo  $t$  finito e limitado. Isso é feito de modo a se ter um *Processo de decisão de Markov (Markov Decision Process)* (MDP) em que cada sequência é um estado distinto.  $E$  é um conjunto de experiências  $E = e_1, e_2, \dots, e_f$  extraído de forma aleatória da memória.

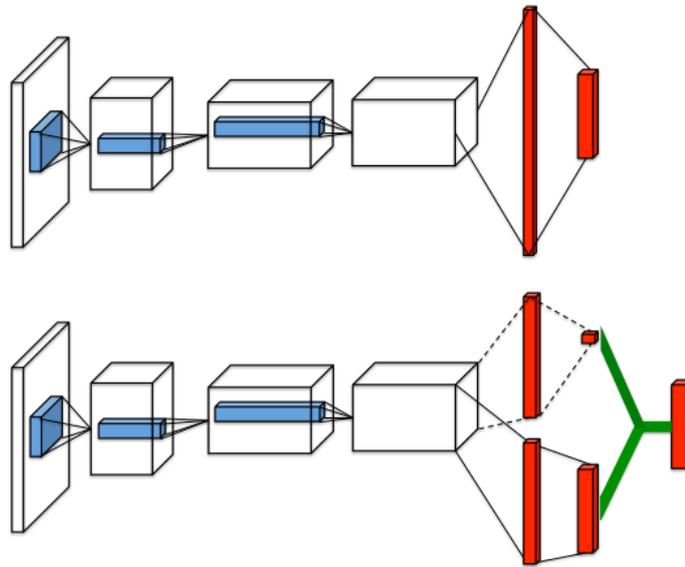
O objetivo do agente é interagir com o ambiente através da escolha de certas ações sempre visando maximizar a recompensa acumulada executando um determinado número de episódios enquanto fator de desconto  $\gamma$  reduz a recompensa futura a cada passo.

### 3.4 Dueling Network Architecture

Sendo muito próxima de uma das propostas do presente trabalho, os pesquisadores da Google Ziyu Wang et al. (2016) desenvolveram uma arquitetura de Aprendizado por Reforço Profundo, conforme apresentado na Figura 8. Os pesquisadores combinaram essa nova arquitetura com o algoritmo DDQN e também com o algoritmo Prioritized Experience Replay.

Essa arquitetura possui duas redes em paralelo de neurônios completamente conectadas representando a função valor  $V(s)$  e a função advantage  $A(s, a)$ , já a saída combina essas duas

Figura 8 – Uma Q-network de fluxo único (em cima) e a Dueling Q-Network (em baixo). A Dueling Q-Network tem dois fluxos para estimar separadamente o valor do estado e as vantagens de cada ação; o módulo de saída verde para combiná-los. Ambas as redes emitem valores Q para cada ação.



Fonte: Ziyu Wang et al. (2016)

funções produzindo a função valor estado-ação  $Q(s,a)$  ( $Q(s,a) = V(s) + A(s,a)$ ), mas compartilham das mesmas camadas de redes convolucionais. A função advantage específica como é bom para o agente executar uma ação em comparação com outras ações (WANG, Z. et al., 2016).

### 3.5 Conclusão

Esse capítulo expôs uma visão geral de aprendizado profundo, particularmente para redes neurais convolucionais e foi desenvolvido passando pelo aprendizado por reforço até a sua forma profunda com generalização de estados por meio de redes neurais. Alguns dos principais algoritmos de aprendizado profundo e por reforço também foram revisados focando na proposta deste trabalho, a ser apresentada no capítulo 5. Entretanto, quando se leva em conta diversas áreas, a função a ser aprendida pode-se tornar muito complexa, dessa forma, no próximo capítulo será discorrido como técnicas de agrupamento no estado da arte apresentadas em jornais e revistas, recentes, de grande representatividade estão sendo usadas por pesquisadores para fundir informações, área conhecida como *Information Fusion*, ou fusão de informação(ões).

## 4 Estado da Arte

Como continuação ao capítulo 2, onde foram apresentados conceitos básicos para o entendimento das estruturas e técnicas de redes neurais, agrupamentos e aprendizado profundo, este capítulo estruturará a evolução das pesquisas relevantes nas áreas de economia e inteligência artificial, de modo que seja possível enquadrar a proposta deste trabalho no estado da arte das linhas de pesquisa, mas, também, para fins de comparação e demonstração de suas contribuições. Para tal, foram revisados diversos artigos publicados em revistas e conferências de alta relevância nessas duas áreas entre os anos de 2015 e 2019 sobre agrupamento de redes neurais e aprendizado por reforço para aplicação no mercado financeiro tradicional.

### 4.1 Pesquisa e Divulgação Sobre Agrupamento de Redes Neurais

A *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)* oferece a toda comunidade a plataforma WEBQUALIS, um sistema de Classificação de periódicos nacionais e internacionais, nos quais haja publicações que representem a produção intelectual de todas as áreas do conhecimento.

Até onde pode-se verificar, não parece haver revista de relevância, webqualis acima de B2 para a grande área engenharias IV, que abranja simultaneamente as sub-áreas de mercado financeiro e inteligência artificial.

As seguintes revistas são webqualis A1 na área de avaliação economia, entretanto não possuem relevância em engenharias IV, por exemplo O JOURNAL OF FINANCIAL ECONOMETRICS e o JOURNAL OF APPLIED ECONOMETRICS, no entanto, tiveram alguns artigos utilizados para fins de contextualização.

Os resultados das principais pesquisas referentes a redes neurais e seus agrupamentos com algum viés de séries temporais e/ou mercado financeiro são apresentados regularmente nos eventos apresentados nas tabelas 1 e 2.

Além disso, os seguintes eventos possuem uma classificação não tão alta na grande área economia ou contábeis, porém, apresentaram artigos com alguma relação com a proposta.

Tabela 1 – Tabela de principais eventos sobre redes neurais

Título	Classificação	Observação
IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS	A1	
NEURAL NETWORKS	A1	
JOURNAL OF MACHINE LEARNING RESEARCH	A1	
IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS	A1	
EXPERT SYSTEMS WITH APPLICATIONS	A1	
INFORMATION FUSION	A1	
NEUROCOMPUTING	A1	Interdisciplinar
INTERNATIONAL JOURNAL OF NEURAL SYSTEMS	A2	
INVESTMENT MANAGEMENT AND FINANCIAL INNOVATIONS	C	Apesar de ter avaliação C para engenharias IV, possui avaliação B1 para Administração de empresas ver tabela 2

Tabela 2 – Tabela de principais eventos sobre redes neurais

Título	Classificação	Observação
NEUROCOMPUTING	ENGENHARIAS IV	A1
INVESTMENT MANAGEMENT AND FINANCIAL INNOVATIONS	Administração de Empresas	B1
QUANTITATIVE FINANCE DE EMPRESAS	Administração A1	
QUANTITATIVE FINANCE	ECONOMIA	B1

## 4.2 Pesquisas de Mercado Financeiro e Operação Automática

Sendo um domínio relativamente recente, as pesquisas na área específica de aprendizado de máquina para operações automáticas no mercado de Bitcoin ainda são escassas na literatura e seu uso em criptomoedas secundárias, como o Ethereum, são ainda mais raras.

Antes de adentrar no mercado de criptomoedas faz-se necessário determinar alguns conceitos do mercado de capitais tradicional. Em outras palavras para prever os valores de papéis, se faz necessário entender como funciona o mercado financeiro. Apesar do objetivo não ser a predição propriamente dita e sim executar ações que retornam o maior lucro com menos risco.

Segundo Downes e Jordan Elliot Goodman (1993), um *trader* ou negociante em geral, se trata de qualquer pessoa que compre ou venda bens e serviços para realizar um lucro. Realizar um lucro, aqui, também é um jargão, que representa a expressão do mercado de ações utilizada quando os investidores vendem ou executam suas aplicações para obter lucro. A negociação automática para ações de mercado é atraente para pesquisadores e profissionais do mercado. As abordagens existentes para negociação algorítmica podem ser categorizadas em métodos baseados no conhecimento e métodos baseados em AM

Métodos baseados no conhecimento projetam estratégias comerciais baseadas em pesquisas financeiras ou experiência comercial; Métodos baseados em ML, em contraste, aprendem estratégias de negociação a partir dos dados históricos de mercado. Uma vantagem óbvia dos métodos baseados em *Aprendizagem de Máquina (Machine Learning)* (AM) é que eles podem descobrir padrões lucrativos que ainda são desconhecidos para as pessoas.

Basicamente, existem três hipóteses que alertam sobre a possibilidade de não haver uma maneira lucrativa de prever as tendências do mercado financeiro de maneira gráfica. São elas:

- a) A teoria do *Passeio Aleatório (Random Walk)* (RW) (PEARSON, 1905)
- b) A *Hipótese da Eficiência do Mercado (Market Efficiency Hypothesis)* (EMH) (FAMA, 1965)
- c) *Estratégias Auto Destruidoras (Self Defeating Strategies)* (SDS) (MALKIEL, Burton G.; FAMA, 1970)

A hipótese de caminhada ou passeio aleatório estabelece uma visão da imprevisibilidade do mercado de ações. Um passeio aleatório é um objeto matemático, conhecido como processo estocástico ou aleatório, que descreve um caminho que consiste em uma sucessão de etapas aleatórias em algum espaço matemático (PEARSON, 1905). A hipótese diz que, analisando pelo

aspecto gráfico, o preço de mercado de um papel é essencialmente aleatório. A hipótese implica que qualquer tentativa de prever o mercado de ações inevitavelmente falhará. Para provar essa hipótese, BG Malkiel (2013), demonstrou ser possível enganar um especialista do mercado de ações na previsão de um mercado falso. Ele criou um experimento onde repetidamente jogava uma moeda. Se a moeda mostrava Cara, ele movia o preço de um papel fictício para cima, e se ela mostrava Coroa, então ele baixava o preço do papel. O autor, então, levou sua tabela aleatória de preços de ações para um suposto especialista em previsão de ações e pediu uma previsão. O especialista, sem fazer ideia de que estava sendo enganado, recomendou a compra imediata do papel.

Informalmente, a *Hipótese da Eficiência do Mercado (Market Efficiency Hypothesis)* (FAMA, 1965) diz que o mercado é eficiente em encontrar o preço correto para uma determinada ação. Três hipóteses foram propostas por Burton G. Malkiel e Fama (1970) baseadas na EMH, no entanto, ainda é uma questão de debate qual delas se aproxima mais da realidade. As hipóteses são: A hipótese fraca; A hipótese semi-fraca e; A hipótese forte.

A hipótese fraca diz que ninguém pode lucrar com o mercado de ações apenas observando tendências e padrões dentro do preço da ação em si. É importante ressaltar que isso não exclui lucrar com as previsões do preço de um papel com base em dados externos ao preço. Em (DUNNE, 2015) são apresentados exemplos previsão baseados em dados de preços internos de ações e dados externos ao preço, e fornecendo algumas evidências que apoiam da hipótese fraca.

A hipótese semi-forte diz que nenhum método de previsão é capaz de descobrir o valor futuro de um determinado papel, com exceção de pessoas que possuam informações privilegiadas, também chamadas de *Insiders*. Isso significa que, se apenas informações de domínio público forem usadas, não haverá sucesso na tentativa de predição.

Já na hipótese forte, nenhum método ou técnica seria capaz de prever o mercado, não importando a quais informações essas técnicas têm acesso.

Finalmente, na hipótese de estratégias autodestruidoras (MALKIEL, Burton G.; FAMA, 1970), existe a idéia de um modelo de sucesso que pode deixar gradativamente de obter lucro. A percepção é que, se houvesse um modelo verdadeiramente preditivo que qualquer um pudesse aplicar e lucrar, ao longo do tempo, todas as vantagens seriam negociadas e corroídas, já que segundo a hipótese de eficiência de mercado o valor seria reajustado. Se um modelo verdadeiramente bem sucedido fosse amplamente conhecido, então, não demoraria muito até deixar de ser útil.

A seguir são relacionados alguns termos citados e definidos por Treleaven, Galas e Lalchand (2013) e que são comumente usados na negociação automática, a saber:

- a) *A Negociação Algorítmica ou Automatizada (Algorithm Trading) (AT)* refere-se a qualquer forma de interação usando algoritmos para automatizar a totalidade ou parte do ciclo de negociação. A AT geralmente envolve aprendizagem (TAYLOR; KIM, M.; CHOI, 2014), planejamento dinâmico, raciocínio e tomada de decisão (SHERSTOV; STONE, 2004; BAHRAMMIRZAEI, 2010; LÄNGKVIST; KARLSSON; LOUTFI, 2014; SUBRAMANIAN et al., 2006).
- b) *Negociação Sistemática (Systematic Trading) (ST)*: Refere-se a qualquer estratégia de negociação que seja uma abordagem sistemática ou repetitiva baseada em regras de comportamento. Isso geralmente é alcançado através da utilização de um sistema especializado que replica ações previamente capturadas de *traders* reais .
- c) *A Negociação de Alta Frequência (High Frequency Trading) (HFT)* é um conceito onde a execução de estratégias de negociação computadorizadas é caracterizada por períodos de posse de posição extremamente curtos da ordem de alguns segundos a algumas dezenas de milissegundos (MCGOWAN, 2010).
- d) *A Negociação de Altíssima Frequência (Ultra High Frequency Trading) (UHFT)*: ou negociação de baixa latência refere-se à execução do HFT em períodos ainda menores, da ordem de dezenas de milissegundos. Isso é alcançado por meio do posicionamento servidores, acesso direto ao mercado ou feeds de dados individuais e outros para minimizar diversos tipos de latências de rede. Um dos principais tipos de UHFT é conhecido como *Scalping*, trata-se de uma estratégia operacional executada nos mercados financeiros com o intuito de lucrar com os movimentos curtos e rápidos da variação dos valores das ações. O comércio de baixa latência é um dos principais impulsionadores do desempenho e também a pesquisa de algoritmos em *GPUs* e em *FPGAs* (TRELEAVEN; GALAS; LALCHAND, 2013; MCGOWAN, 2010) .
- e) *Interação de algoritmos*: Na verdade, sabe-se que o próprio movimento de compra ou venda influencia o mercado (LEMOS, 2015).

Ainda segundo Treleaven, Galas e Lalchand (2013) e Cavalcante et al. (2016), o processo de negociação automatizada pode ser dividido em cinco estágios:

- a) Obtenção de dados e respectiva limpeza. Aqui a palavra limpeza é usada no sentido de retirar outliers e ruídos que não pertencem ao padrão que pretende-se aprender.
- b) Análise pré-negociação ou análise de dados: analisa propriedades de ativos para identificar oportunidades de negociação usando dados de mercado ou notícias financeiras. Compreende recursos analíticos de computação através de três componentes principais: modelo alfa, o modelo matemático projetado para prever o comportamento futuro dos instrumentos financeiros; Modelo de risco, que avalia os níveis de exposição ou de risco associados aos instrumentos financeiros; e modelo de custo de transação que calcula os custos associados à negociação dos instrumentos financeiros.
- c) Geração de sinal de negociação: Identifica e seleciona o portfólio de ativos a serem acumulados com base na análise pré-negociação e determina quando executar determinada ação. É preciso adequar todos os dados que serão enviados em formatos específicos, incluindo data, tipo de operação e quantidade.
- d) Execução: Envio de pedidos referentes ao sinal de negociação. Nesse ponto ocorre a tramitação entre o pedido feito pelo AT e a resposta de recebimento do servidor.
- e) Análise pós-negociação: Analisa os resultados da atividade de negociação, como a diferença entre o preço quando foi tomada uma decisão de compra ou venda e o preço de execução final.

Com os sistemas automáticos definidos, o próximo passo é definir como esses sistemas serão avaliados.

### 4.3 Métricas de Desempenho

Três métricas de desempenho são comumente usadas no mercado financeiro, o fator de lucro, o *Retorno sobre o investimento (Return Over Investment)* (ROI) e o índice Sharpe. Essas métricas serão utilizadas no capítulo 6

### 4.3.1 Fator de lucro

Quando são efetuadas diversas operações de *trading* é normal que ocorram operações que geram prejuízo. Visando identificar a relação de ganhos *versus* as perdas das estratégias, será usado o fator de lucro.

Para exemplificar, considere um fator de lucro de 2,50, isso significa que se for investido um dólar, após todas as negociações executadas, pode-se esperar recuperar o dólar e obter um lucro de 1,5. Da mesma forma, um fator de lucro ou próximo da unidade significa que, se for investido 1 dólar, recebe de volta exatamente o dólar que investiu, então quando ele se aproxima da unidade não é considerado um bom indicador de lucro.

$$FL = \frac{G_b}{P_b} \quad (24)$$

Onde, o fator de lucro  $FL$  é a razão entre  $G_b$  que são os ganhos brutos de todas as operações e  $P_b$  que são as perdas brutas de todas as operações.

Analisar um fator de lucro baixo de forma individual e isolada pode levar a conclusões equivocadas como rentabilidade abaixo do esperado com relação aos riscos que se corre para aumentar o capital, por isso, esse indicador está sendo analisado juntamente com o índice Sharpe.

### 4.3.2 ROI

O *Retorno sobre o investimento (Return Over Investment)* (ROI), é a taxa de rentabilidade mais comum. Existem várias maneiras de determinar o ROI, mas o método mais frequentemente usado é dividir o lucro líquido pelo total investido no ativo. Portanto, se o lucro líquido for de US \$10.000 e o total de ativos for de US\$ 30.000, o ROI será de 0,33 ou 33%.

$$ROI = \frac{L_l}{T_I} \quad (25)$$

onde o retorno sobre o investimento  $ROI$  é a relação entre o lucro líquido  $L_l$  e o total investido  $T_I$ .

Retorno do investimento não é necessariamente o mesmo que lucro. O ROI lida com o dinheiro investido no ativo e com o retorno que obtém sobre esse dinheiro com base no lucro

líquido, ou seja, descontando-se taxas de emolumentos, de mineração (ver capítulo 7 e seção 4.6) e de corretoras.

### 4.3.3 Índice Sharpe

O índice Sharpe, Razão Sharpe e/ou coeficiente Sharpe foi desenvolvido pelo ganhador do Nobel William F. Sharpe e é usado para ajudar os investidores a entender o retorno de um investimento em comparação com seu risco.

A razão é o retorno médio obtido em excesso da taxa livre de risco por unidade de volatilidade ou risco total. Volatilidade é uma medida das flutuações de preço de um ativo ou portfólio.

Subtrair a taxa livre de risco do retorno médio permite ao investidor isolar melhor os lucros associados às atividades de risco. A taxa de retorno livre de risco é o retorno de um investimento com risco zero, o que significa que é o retorno que os investidores esperariam por não correr riscos.

No Brasil, o rendimento de um título CDI ou a taxa referenciada pelo *Sistema Especial de Liquidação e Custódia*, também chamada de *taxa básica de juros* (SELIC), por exemplo, poderiam ser usadas como uma taxa livre de risco. Como a relação do Ethereum é frente ao dólar, foi usado o rendimento de um título do Tesouro dos EUA, os *Treasure Bills* que inclusive, são considerados os títulos mais seguros do mundo. Para o presente a taxa vigente era de 0.72% ao mês.

$$IS = \frac{R_p - R_f}{\sigma_p} \quad (26)$$

Onde,  $R_p$  é o retorno do investimento,  $R_f$  é a taxa livre de risco e  $\sigma_p$  é o desvio padrão do excesso de retorno esperado. Esse desvio padrão também é referenciado na literatura como volatilidade. O desvio padrão ajuda a mostrar quanto o retorno real se desvia do retorno esperado. Geralmente, quanto maior o valor da razão Sharpe, mais atraente é o retorno ajustado ao risco.

## 4.4 Aprendizado de Máquina para o Mercado Financeiro

Para efeito de contextualização de métodos baseados em AM, os seguintes artigos escritos por Pulido descrevem métodos de otimização para modelos de agrupamento de redes

neurais, usando: algoritmos genéticos e fuzzy para a série Mackey-Glass (PULIDO, M.; MELIN; CASTILLO, 2011) e índice DOWJONES (PULIDO, M. E.; MELIN, 2012); Otimização por PSO com Fuzzy do tipo I e do tipo II para unificar as saídas das redes no estilo Stacking para a relação Dolar e Pesos (PULIDO, M.; MELIN; CASTILLO, 2014) e índice da Bolsa de valores de Taiwan (PULIDO, M.; MELIN; MENDOZA, 2017)

Já em, Kara, Boyacioglu e Baykan (2011), os autores se basearam em duas técnicas de classificação a fim de compará-las: RNA e SVM. Para tal selecionaram dez indicadores técnicos, relacionados com a bolsa de valores de Istambul, como entradas dos modelos propostos. Os resultados mostraram que o desempenho médio das redes neurais foi significativamente melhor do que o modelo SVM. Em Luss e Aspremont (2015) usaram artigos de notícias para prever movimentos de preços intradiários de ativos financeiros usando o modelo SVM.

Os autores O'Connor e Madden (2006) avaliaram a eficiência de se usar indicadores externos, como taxas de câmbio e commodities para prever as oscilações do índice DOWJONES, eles afirmam que a rede neural foi melhor que o índice em si, porém sua maior contribuição foi a disponibilização de um conjunto de dados para treinamento dessa rede. Também para o índice DOWJONES, uma rede neural de 4 camadas foi usada para descobrir padrões não lineares em séries temporais, de modo que os autores a usaram para executar uma análise técnica de predição de valor para papéis desse índice. Para a bolsa de valores de São Paulo, um mercado pouco explorado, vale citar o trabalho de Martinez et al. (2009) que usaram redes neurais e como entradas indicadores dos últimos 5 dias para prever os valores das ações preferencias da Petrobras e Vale do Rio Doce.

Visando ignorar *outliers* e desvios anormais, a moda foi usada como operador de agrupamento de redes neurais por Kourentzes, Barrow e Crone (2014). Notícias e tendências de mercado foram combinados por meio de SVM, por Yuzheng Zhai, Hsu e Halgamuge (2007). Existem muitas outras abordagens para combinar redes neurais para previsão de séries temporais como mostra a pesquisa de (BAHRAMMIRZAEI, 2010), (LI, Yuhong; MA, 2010) e (ATSALAKIS; VALAVANIS, 2009). Em um trabalho mais recente Atsalakis, Protopapadakis e Valavanis (2016) propôs um sistema neuro-fuzzy para predição de valor de papéis em períodos turbulentos.

Há também aqueles que propuseram o uso de aprendizado por reforço, seja no formato profundo ou não, (DENG, Yue et al., 2017; DING et al., 2015; WANG, Y. et al., 2017). Em Långkvist, Karlsson e Loutfi (2014), há uma revisão sobre técnicas de aprendizado profundo supervisionados e não supervisionados para previsão de séries temporais.

Diversos pesquisadores desenvolveram estratégias voltadas para o mercado financeiro, com o uso de agrupamento de redes neurais usando algoritmos genéticos, como por exemplo os trabalhos Martha Pulido, Melin e Mendoza (2017), Martha Pulido, Melin e Castillo (2014), Martha Elena Pulido e Melin (2012), Martha Pulido, Melin e Castillo (2013) e Hassan e Ajith Abraham (2013). Em Hassan e Ajith Abraham (2013) foi apresentada uma investigação do uso de modelos de rede neural supervisionada no formato de agrupamento *stacking* para previsão de empréstimos de clientes sob diferentes algoritmos de treinamento. Em Felipe Giacomel, Pereira e Galante (2015), os autores argumentam que o tipo de previsão de valores dos papéis não precisam fornecer os valores futuros exatos, a tendência de séries temporais é suficiente e para isso propuseram um agrupamento de redes neurais que recebe como entrada os últimos valores de uma série temporal e retorna uma previsão que indica a tendência do próximo valor.

Segundo Radinsky, Davidovich e Markovitch (2012) e Ding et al. (2014) os títulos das notícias são mais importantes que seu conteúdo para a tarefa de predição do valor do papel. Em B. Yang, Gong e W. Yang (2017) foi proposto um agrupamento do tipo *Bagging* de redes neurais profundas para previsão do índice composto de Shanghai e o índice de componentes *Bolsa de valores de Shenzhen-China (Shenzhen Stock Exchange)* (SZSE). Uma fusão de redes neurais profundas e lógica fuzzy foi proposta por Y. Deng et al. (2017). Visando o HFT, Arévalo et al. (2017) propôs o uso de redes neurais profundas para operar na bolsa de Nova York. Em Heaton, Polson e Witte (2016) é apresentado como algumas ferramentas de aprendizado profundo, como autoencoders, podem ser usados na área de finanças. Visando a predição do índice SP500 Krauss, Do e Huck (2017) implementaram e analisaram a eficácia de redes profundas, *Árvores de Impulso de Gradiente (Gradient Boosting Trees)* (GBT), *Florestas Aleatórias (Random Forests)* (RAF), além do agrupamento desses métodos. Uma revisão sobre aprendizado profundo e previsão de séries temporais é apresentada por Santos e Kern (2016). Uma *Redes Convolucionais Profundas (Deep Convolutional Neural Networks)* (DCNN) com o uso de Wavelets foi proposta por Borovykh, Bohte e Oosterlee (2017) para previsão do índice SP500. Os trabalhos propostos por Felipe dos Santos Giacomel (2016), Carmo Roque e Mello (2009), Bortoli Filho (2016), Wesley da Silva e Claudimar da Veiga e Ubiratã Tortato (2011) e Cristiano Leite de Castro, Pádua Braga e Andrade (2005) usaram redes neurais e/ou agrupamentos de RNA com foco no índice *Bolsa de Valores do Estado de São Paulo, agora parte da B3, empresa de infraestrutura de mercado financeiro com atuação em ambiente de bolsa, futuros e de balcão.* (BOVESPA), ou seus papéis .

#### 4.4.1 *Long Short Term Memory*

*Long-short Term Memory* (LSTM) é uma variante das RNN e foi proposta por Hochreiter e Schmidhuber (1997) para resolver o problema do desvanecimento de gradiente (*Vanishing gradient problem*) por meio de uma célula de memória que mantém estados sobre uma série de dados temporais de longo termo.

As sequências de dados em que há mais de uma observação para cada etapa do tempo são consideradas séries temporais multivariadas. Segundo Zainudin, Shamsuddin e Hasan (2020), existem dois modelos principais que podemos utilizar com dados de séries temporais multivariadas, que são:

- a) Série de entradas múltiplas. Um problema pode ter duas ou mais séries temporais de entrada paralela e uma série temporal de saída que depende da série temporal de entrada. As séries temporais de entrada são paralelas porque cada série possui uma observação nas mesmas etapas de tempo.
- b) Múltiplas séries paralelas. Um problema alternativo de série temporal é o caso em que há várias séries temporais paralelas e um valor deve ser previsto para cada um.

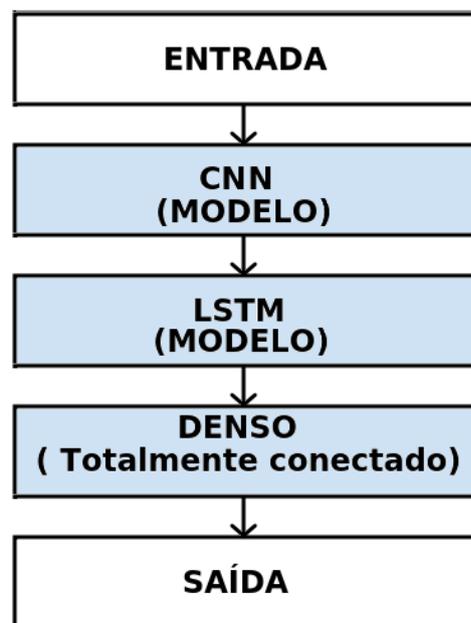
A *Arquitetura de Rede Neural Convolutacional com LSTM* (*Convolutional LSTM Neural Network*) (CNN-LSTM) (ZAINUDIN; SHAMSUDDIN; HASAN, 2020) envolve o uso de camadas da *Redes Neurais Convolucionais*, (*Convolutional Neural Networks*) (CNN) para extração de recursos em dados de entrada combinados com LSTMs para suportar a previsão de sequência. É possível citar (CHINA BHANJA; LASKAR, M. A.; LASKAR, R. H., 2020) e (BARTZ et al., 2017) usaram redes recorrentes LSTM convolucionais para reconhecimento de NLP.

A A CNN-LSTM foi desenvolvida para problemas de previsão de séries temporais e aplicação de gerar descrições textuais a partir de sequências de imagens ou vídeos. Segundo China Bhanja, Mohammad A Laskar e Rabul H Laskar (2020), a CNN-LSTM é uma classe de modelos que é espacial e temporalmente profunda e tem flexibilidade para ser aplicada a uma variedade de tarefas que envolvem entradas e saídas sequenciais. Essa arquitetura foi originalmente chamada de LRCN, embora a literatura se refira com o nome CNN-LSTM.

Em Donahue et al. (2014) essa arquitetura é usada para a tarefa de gerar descrições textuais de imagens. A chave é o uso de uma CNN pré-treinada em uma tarefa desafiadora de classificação de imagens, que é reutilizada como um extrator de recursos para o problema de

geração de legendas. No mesmo trabalho os autores dizem que usam CNN como um codificador de imagem, treinando-o primeiro para uma tarefa de classificação de imagem e usando a última camada oculta como uma entrada para o decodificador RNN que gera sentenças.

Figura 9 – Arquitetura de uma *Arquitetura de Rede Neural Convolucional com LSTM* (*Convolutional LSTM Neural Network*) (CNN-LSTM).



Fonte: “Adaptado de” LeCun, Bengio e Geoffrey Hinton (2015)

#### 4.5 Criptomoedas

No final de 2008, um novo sistema criptográfico de pagamentos e transferências descentralizado foi publicado anonimamente sob pseudônimo Satoshi Nakamoto, que formou a base da tecnologia blockchain. Simultaneamente, lançou a aplicação mais conhecida da tecnologia blockchain na forma de uma criptomoeda que foi chamada de Bitcoin (NAKAMOTO, 2008). Transações bancárias normalmente são centralizadas em uma instituição bancária que pode estar sujeita a ataques de hackers, sofrem de altos custos relativos e longos períodos associados a transações entre fronteiras e / ou interbancárias. Esses problemas foram solucionados pela aplicação do Blockchain no formato de criptomoedas. Nos anos após o início do Bitcoin, muitas outras criptomoedas, conhecidas como altcoins, como Ethereum, Litecoin foram desenvolvidas. Muitas vezes, essas altcoins foram desenvolvidas para um propósito diferente ou tentaram

melhorar as limitações do Bitcoin, como o suprimento limitado de Bitcoin, o alto consumo de energia da rede ou o mecanismo de consenso entre nós. (KRAAIJEVELD; SMEDT, 2020)

Inicialmente, as criptomoedas tinham uma reputação questionável, muitas vezes sendo rotuladas como obscuras ou moedas para criminosos (MIHM, 2013), mas isso mudou quando o interesse no mercado de criptomoedas explodiu ao longo de 2017 e início de 2018, levando a um mercado de extrema alta, alimentado pelo medo de perdas financeiras (BAUR; DIMPFL, 2018).

Como resultado, o número de criptomoedas listadas mais que triplicou para 1.865 e a capitalização total do mercado de criptomoedas aumentou de US\$ 17 bilhões em 1 de janeiro de 2017 para US\$813 bilhões quase um ano depois (COINMARKETCAP. . . , 2019). Um bom exemplo desse aumento é ilustrado em Corbet et al. (2020), que descrevem como o anúncio do desenvolvimento de uma criptomoeda baseada na empresa alimentou o preço das ações da empresa.

#### 4.6 Blockchain e Mineração

Apesar da indústria financeira ser a maior consumidora da tecnologia, o Blockchain pode ser usado em outros campos, como *smart contracts* (KOSBA et al., 2016), serviços públicos (FOROGLOU; TSILIDOU, 2015), *Internet das coisas (Internet of Things)* (IoT) (ZHANG, Y.; WEN, 2015), sistemas de reputação (AKINS; CHAPMAN; GORDON, 2014) e seguros (SHARPLES; DOMINGUE, 2016). Estes campos se valem da imutabilidade do Blockchain, uma vez que uma transação é registrada, ela não pode ser mais retirada. Antes do Blockchain as moedas digitais foram contextualizadas em uma configuração centralizada, na qual haveria a confiança de se evitar o gasto duplo, *Double Spending*.

O Blockchain, como o próprio nome diz é uma cadeia sequencial de blocos interconectados que armazena a lista de transações anteriores, de forma análoga aos registros de um livro de caixa. Cada bloco possui um hash do bloco anterior, chamado de bloco pai, em seu cabeçalho, de modo que, um bloco tem apenas um bloco pai. Contudo no Blockchain do Ethereum existem estruturas como Blocos Tio, filhos dos ancestrais do bloco (BUTERIN et al., 2014).

A tecnologia Blockchain permite que para todos que tem acesso aos nós e aos seus registros, que também são chamados de *Ledgers*, compartilhar e atualizar blocos de informação por meio de um algoritmo de criptografia elíptica, cujo o estado é coletivamente mantido pela comunidade (ZHENG et al., 2017).

Todo o Blockchain fica armazenado em um nó, que pode ser localmente salvo em um computador. Os mineradores tem acessos em diferentes níveis dentro do nó, que também fornece números de carteiras, chaves criptográficas e hashes de transação.

Segundo Zheng et al. (2017), algumas das propriedades do Blockchain incluem:

- a) **Descentralização** : Nos sistemas convencionais de transações, cada transação precisa ser validada através de alguma agência confiável central, por exemplo, o banco central, resultando inevitavelmente em custos e gargalos nos servidores centrais. Isso também elimina a necessidade de terceiros. Algoritmos de consenso em blockchain são usados para manter a consistência dos dados na rede distribuída.
- b) **Auditabilidade dos dados** : Qualquer transação deve se referir a uma ou mais transações não gastas anteriormente. Uma transferência pode ser originada em diversas carteiras diferentes simultaneamente e depois que a transação atual é registrada no blockchain, o estado de todas as transações, que deram origem a essa transação e que estavam definidas como não gastas anteriormente, tem seu status alterado de não gasto, para gasto. O registro do status das transações facilitam sua verificação e permitem sua rastreabilidade. Como são validadas por muitos mineradores, as transações inválidas não são passíveis de inserção em um bloco.
- c) **Persistência** : Devido a natureza criptográfica e distribuída é quase impossível excluir ou reverter transações uma vez incluídas na blockchain.
- d) **Anonimato**: Cada usuário pode interagir com o Blockchain com um endereço, que pode ser gerado pela rede e que não revela sua identidade.

#### 4.6.1 Bloco

A estrutura do Blockchain difere de acordo com a criptomoeda ou aplicação, mas, de modo geral, seus blocos consistem de um cabeçalho (*Header*) e um corpo (*Body*). A figura 10 mostra a estrutura interna dos blocos, assim como estrutura externa. O cabeçalho é composto por: (Ver figura 10)

- a) A versão do bloco indica o conjunto de regras para validação do bloco que o Blockchain deve seguir;
- b) As árvores de dispersão de Hash, *Merkle Tree Root Hash* são um tipo de estrutura de dados que contém uma árvore de informações resumidas sobre um pedaço

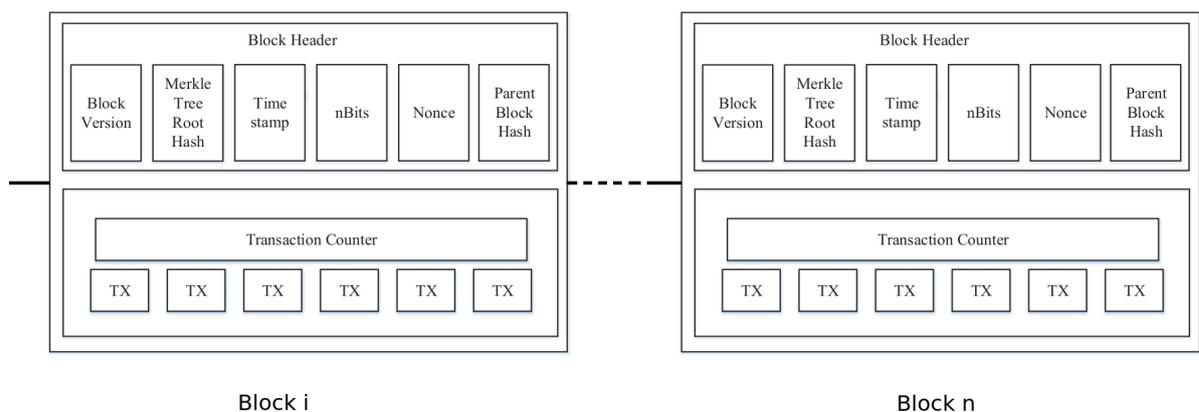
maior de dados, no Blockchain é uma forma de armazenar todas as transações do bloco.

- c) Hora Atual
- d) nBits : Limiar alvo de um hash de bloco válido.
- e) Nonce : Um campo de 4 bytes que é incrementado a cada cálculo de Hash.
- f) Hash do bloco pai: um valor de 256 bits que aponta para o bloco anterior

O corpo do bloco é composto pelas transações e respectivo contador, sendo o limite de tamanho e quantidade de transações limitado pelo tamanho do bloco. O Blockchain usa um mecanismo de assinaturas digitais baseadas em criptografia assimétrica, tipicamente com o algoritmo de *Curva elíptica de assinatura digital (elliptic curve digital signature algorithm)* (ECDSA), para validar a autenticidade dessas transações. De maneira geral, essas assinaturas provêm um par de chaves, sendo uma pública e outra privada. A chave privada deve ser mantida em sigilo e é usada para assinar as transações.(ZHENG et al., 2017)

No bloco do blockchain do ethereum é possível armazenar outros dados que não representam transações ou algo relacionado diretamente com o Blockchain, por exemplo, no caso de *Smart Contracts* é possível inserir dados de uma propriedade, um imóvel, que, ao ter sua titularidade alterada pode ser inserida no blockchain como tal. Deste modo as informações ficam registradas permanentemente nesse blockchain, além disso, por ser um protocolo de transação computadorizado ele pode executar os termos desse contrato, inclusive em casos de quebra do mesmo. (BUTERIN et al., 2014), (KOSBA et al., 2016)

Figura 10 – Estrutura interna e externa de um Blockchain



Fonte: Adaptado de Zheng et al. (2017)

#### 4.6.2 Algoritmos de Consenso

No blockchain, como chegar a um consenso entre nós, a princípio não confiáveis, é uma transformação do *Problema dos Generais Bizantinos (Byzantine Generals)* (BG), que foi levantada em (LAMPORT; SHOSTAK; PEASE, 1980), (LAMPORT; SHOSTAK; PEASE, 2019). No *Problema dos Generais Bizantinos (Byzantine Generals)* (BG), um grupo de generais que comanda uma porção do exército bizantino circunda a cidade. Alguns generais preferem atacar, enquanto outros generais preferem recuar, no entanto, o ataque falharia se apenas parte dos generais atacasse a cidade, de modo que, há uma divergência de consenso sobre qual atitude tomar. No caso do Blockchain a divergência de consenso ocorre entre os nós, se devem ou não validar uma transação, por exemplo, considere um nó que registrou um bloco que possui uma ou mais transações com gasto duplicado, durante o consenso os nós divergiram sobre qual a origem daquela transação. Se a maioria dos nós definirem que a transação é válida, então ela é registrada como tal. No caso de criptomoedas que possuem poucos mineradores ou possuem dificuldade de rede baixa, há a possibilidade de se apoderar de mais de 50% da mineração e alterar o bloco, isso faria com que, durante o consenso dos nós, blocos com transações contendo gastos duplos passassem despercebidas, já que os blocos serão validados pela maioria dos nós, ou seja, por quem tem o controle da mineração.

O consenso em ambiente distribuído, como no caso da rede Blockchain é um desafio, já que não há um nó central que garanta que os ledgers nos nós distribuídos sejam todos referentes ao mesmo registro. Alguns protocolos são necessários para garantir que os ledgers em diferentes nós sejam consistentes. Esses protocolos são chamados de algoritmos de consenso e os mais comuns são a *Prova por meio de trabalho (Prove of Work)* (PoW) (NAKAMOTO, 2008) e a *Prova por meio da garantia de propriedade (Prove of Stake)* (PoS)<sup>1</sup> O *Ethereum* (ETH) é uma criptomoeda que usa o PoW(WOOD, 2014) como algoritmo de consenso, mas tem previsão de migração para o PoS(INTRODUCING... , 2015)

Existem outros algoritmos de consenso, como a *Prova por Tolerância a Falhas Bizantinas Práticas (Practical byzantine fault tolerance)* (PBFT) (CASTRO, M.; LISKOV et al., 1999; CASTRO, M.; LISKOV, 2002; MAZIERES, 2015; CACHIN et al., 2016; ANTSHARES... , 2015), *Prova por meio de propriedade e Delegação (Delegated proof of stake)* (DPoS)(BITSHARES... , 2015), Ripple (SCHWARTZ, D.; YOUNGS; BRITTO et al., 2014) e o Tendermint (KWON,

<sup>1</sup>A tradução literal remete a estaca, como em alguém que está afirmando, declarando ou marcando que possui aquela propriedade (Fincando estacas).

2014). O trabalho no consenso *Prova por meio de trabalho (Prove of Work)* (PoW) significa cálculos matemáticos feitos por um computador, a premissa é que dificilmente alguém que tenha trabalho para fazer parte do consenso vá tentar hackear a rede. No PoW, cada nó da rede está calculando um valor de hash do cabeçalho do bloco. (ZHENG et al., 2017)

O cabeçalho do bloco contém um nonce e os mineradores mudam o valor de nonce frequentemente para obter diferentes valores de hash, de acordo com a dificuldade da rede. Quando um nó atinge o valor alvo, ele transmite o bloco para outros nós e todos os outros nós devem confirmar mutuamente a correção do valor do hash, sendo, cada confirmação chamada de confirmação de rede. Normalmente aguarda-se no mínimo 3 confirmações de rede para assegurar que a transação realmente foi validada. Quando o bloco é validado, outros mineradores anexam esse novo bloco aos seus próprios Blockchains.

Os nós que calculam os valores de hash, por consequência validam o nó e suas transações, são chamados de mineradores. Esses são recompensados pela rede a cada validação, porém, o minerador que consegue fechar o bloco recebe um valor maior da rede na própria criptomoeda.

#### 4.7 Aprendizado de Máquina No Mercado de Criptomoedas

Como já foi mencionado, as pesquisas na área de aprendizado de máquina no mercado de criptomoedas simultaneamente, até onde se pôde verificar, ainda é restrito, porém, é possível citar casos como Shah e Kang Zhang (2014) que implementou um modelo de fonte latente desenvolvido por (CHEN, G. H.; NIKOLOV; SHAH, 2013) para prever o preço do Bitcoin, notando retorno de 89 % em 50 dias com um coeficiente de Sharpe, de 4.1. Também houve trabalho usando dados de texto de plataformas de mídia social e outras fontes para prever os preços do Bitcoin. Georgoula et al. (2015) fez uma análise de sentimentos usando *Máquinas de Vetores de Suporte (Support Vector Machines)* acopladas à frequência das visualizações da Wikipedia e à taxa de hash da rede. Matta, Lunesu e Marchesi (2015a) investigou a relação entre o preço do Bitcoin, os tweets e as visualizações do Bitcoin no Google Trends. A Matta, Lunesu e Marchesi (2015b) implementou uma metodologia semelhante, exceto que, em vez de prever o preço do Bitcoin, eles previram o volume de negociação usando as visualizações do Google Trends. No entanto, uma limitação de tais estudos é o tamanho da amostra, geralmente pequeno, e a propensão à desinformação de se espalhar por vários canais de mídia sociais, como o Twitter ou em quadros de mensagens como o Reddit, que aumentam ou deflacionam artificialmente os

preços (GU, B. et al., 2006). Nas trocas de Bitcoin, a liquidez é consideravelmente limitada e, como resultado, o mercado sofre um risco maior de manipulação.

Greaves e Au (2015) analisaram o Blockchain do Bitcoin para prever o seu preço usando *Máquinas de Vetores de Suporte (Support Vector Machines)* e *Redes Neurais Artificiais* relatando precisão de direção de preço de 55 % com uma rede neural regular. Eles concluíram que havia previsibilidade limitada apenas nos dados do Blockchain. Madan, Saluja e Zhao (2015) também usou dados do Blockchain, implementando SVM, RAF e *Modelo Binomial linear generalizado (Binomial generalised linear model)* (BGLM), observando uma precisão de previsão superior a 97 %, no entanto, sem validar cruzadamente seus modelos, limitando a generalização de seus resultados.

As wavelets também foram utilizadas para prever os preços do Bitcoin, em (DELFINVIDAL; ROMERO-MELÉNDEZ, 2016) e (KRISTOUFEK, 2015) observando correlações positivas entre visualizações de mecanismos de pesquisa, taxa de hash da rede e dificuldade de mineração com o preço do Bitcoin.

Giles, Lawrence e Tsoi (2001) observa que o relacionamento temporal da série é explicitamente modelado pelos estados internos, contribuindo significativamente para a eficácia do modelo.

Rather, Agarwal e Sastry (2015) adotou com êxito essa abordagem na previsão do retorno das ações combinando uma RNN com um algoritmo genético para otimização da rede.

A tese de Júnior (2019) usa comitês de máquinas que são a união de mais de uma técnica de aprendizado de máquina na geração de uma solução para um determinado problema usando um sistema multiagente. A segunda versão dessa técnica, chamada de iEnsemble2 (UBER JUNIOR et al., 2019) utiliza um agente com aprendizado por reforço acelerado por heurística. Nesse ponto, a teoria do agente desempenha um papel fundamental, pois permite a tomada de decisão autônoma do agente, com base em suas experiências, além de fornecer mecanismos para dimensionar e distribuir o processamento. A aprendizagem por reforço é baseada na existência de um crítico externo ao meio ambiente, que avalia a ação definida, mas sem indicar explicitamente a ação correta a ser tomada, dessa forma, permitindo o treinamento dos agentes de forma gradual e auxiliando na aprendizagem. Esse processo de aprendizado pode ser acelerado usando heurísticas no domínio do problema. Dessa forma, este artigo propõe um modelo de comitê de máquina baseado em sistema multiagentes e aprendizado por reforço multipessoal acelerado por heurísticas, descrevendo os experimentos realizados e os resultados obtidos.

Outra forma de *Redes Neurais Recorrentes*, (*Recurrent Neural Networks*) (RNN) é a rede *Long-short Term Memory* (LSTM). Eles diferem das *Redes Neurais Recorrentes de Elman*, (*Elman Recurrent Neural Networks*) (ERNN), pois, além de, ter uma memória intrínseca, eles podem escolher quais dados serão propagados e quais serão esquecidos, com base no peso e na importância desse recurso.

Neste estudo, o modelo série temporal ARIMA e o algoritmo *Long-short Term Memory* (LSTM) foram comparados para estimar o preço futuro do Bitcoin. Sendo o modelo ARIMA, amplamente usado na previsão de séries temporais e o modelo LSTM, que é o algoritmo de aprendizado profundo mais usado na previsão de séries temporais. Foram estimados 30 dias futuros com os modelos obtidos.

O *Erro Médio Absoluto Percentual*, também conhecido como *MAPD* (*Mean Absolute Percentage Error*) (MAPE) dos resultados obtidos foram: aproximadamente 11,86% para o ARIMA e 1,40% para o LSTM. Ainda de acordo com os autores e com outros resultados de testes de precisão, observaram que o modelo LSTM é mais bem-sucedido quando os resultados são comparados.

Balcilar et al. (2017) destacam a importância de modelar a não linearidade e contabilizar o comportamento da cauda ao analisar as relações causais entre os retornos do Bitcoin e o volume de negociação. Entretanto, afirmam também, que o volume pode não ajudar a prever a volatilidade dos retornos do Bitcoin em qualquer ponto da distribuição condicional. Para tal foi utilizado o teste não-paramétrico de causalidade em quantiles para analisar a relação causal entre o volume de negociação e os retornos e volatilidade do Bitcoin, ao longo de todas as suas respectivas distribuições condicionais.

Em Baur e Dimpfl (2018) foram analisados os efeitos assimétricos da volatilidade das 20 maiores criptomoedas. Os autores relatam uma assimetria muito diferente em comparação aos mercados de ações. No mercado tradicional essa assimetria está relacionada com atividade comercial de traders desinformados, esses resultados são consistentes com o medo de assumir prejuízos de investidores desinformados e a existência de esquemas de (pump and dump).

Em Beneki et al. (2019) propuseram-se a testar a hipótese de existirem spillovers de volatilidade entre Bitcoin e Ethereum por uma metodologia multivariada chamada BEKK-GARCH. Os resultados revelaram trocas significativas na correlação variável no tempo e uma resposta positiva atrasada da volatilidade do Bitcoin em um choque de volatilidade positivo nos retornos do Ethereum. As implicações gerais dos resultados são que o Bitcoin e o Ethe-

reum, embora tenham apresentado algumas capacidades diversificadoras nos primeiros anos do estudo, essas capacidades foram recentemente reduzidas significativamente.

Stenqvist e Lönnö (2017) estudaram se a análise de sentimentos nos dados do Twitter, relacionada ao Bitcoin, pode servir como base preditiva para indicar alterações no preço do Bitcoin. O modelo mostrou que o tempo agregado mais preciso para fazer previsões foi de 1 hora, indicando uma alteração no preço do Bitcoin em 4 horas no futuro, entretanto a principal conclusão é que, embora o modelo de previsão apresentado tenha produzido uma precisão de 83%, o número de previsões era tão pequeno que se aventurar nas conclusões do modelo de previsão seria infundado. Além disso os autores não adicionaram léxicos específicos do domínio, que ajudariam a identificar termos financeiros e de criptomoedas e, isso, provavelmente, afetou a precisão da previsão, já que os sentimentos podem ter sido menos representativos.

Várias outras pesquisas analisaram o sentimento no Twitter e tentam encontrar sua relação com o Bitcoin (INAMDAR et al., 2019; KARALEVICIUS; DEGRANDE; DE WEERDT, 2018; ROUHANI; ABEDIN, 2019; LI, T. R. et al., 2019) (PANG; SUNDARARAJ; REN, 2019; DULĂU, T.-M.; DULĂU, M., 2019; BIBI; HUSSAIN; FAISAL, 2019; STEINERT; HERFF, 2018; COLIANNI; ROSALES; SIGNOROTTI, 2015). Em Pant et al. (2018) os tweets do Bitcoin foram coletados de diferentes fontes de notícias foram classificados em sentimentos positivos ou negativos. A porcentagem obtida de tweets positivos e negativos é alimentada pelo modelo RNN, juntamente com o preço histórico para prever o novo preço para o próximo período de tempo. A precisão para classificação de sentimentos de tweets em duas classes positiva e negativa foi de 81,39% e a precisão geral de previsão de preço usando RNN é de 77,62%. Já em (ABRAHAM, J. et al., 2018) usou *Twits* e *google trends* para prever as duas maiores moedas o Bitcoin e o Ethereum também afirmam que o volume de *Twits*, ao invés do *Twit* em si, é um preditor melhor da direção do preço. Kraaijeveld e Smedt (2020) usaram um modelo para testar a presença de robôs de sentimento vinculados ao Twitter para as nove maiores criptomoedas em capitalização de mercado. Em Valencia, Gómez-Espinosa e Valdés-Aguirre (2019) usaram dados de mídia social disponíveis para prever o movimento dos preços das criptomoedas Bitcoin, Ethereum, Ripple e Litecoin, comparando a utilização de *Redes Neurais Artificiais* (RNA), *Máquinas de Vetores de Suporte* (*Support Vector Machines*) (SVM) e *Florestas Aleatórias* (*Random Forests*) (RAF).

#### 4.8 Conclusão

Nesse capítulo, foi apresentado o estado da arte e o domínio ao qual se pretende aplicar a proposta contextualizando o que os principais pesquisadores têm feito nessa área de forma pioneira e inovadora. Por ser uma área de cunho recente, o mercado de criptomoedas ainda possui muitas oportunidades em diversas linhas de pesquisa.

Com o estado da arte apresentado, no próximo capítulo discorrerá sobre a proposta e duas vertentes que foram desenvolvidas visando o agrupamento de redes neurais.

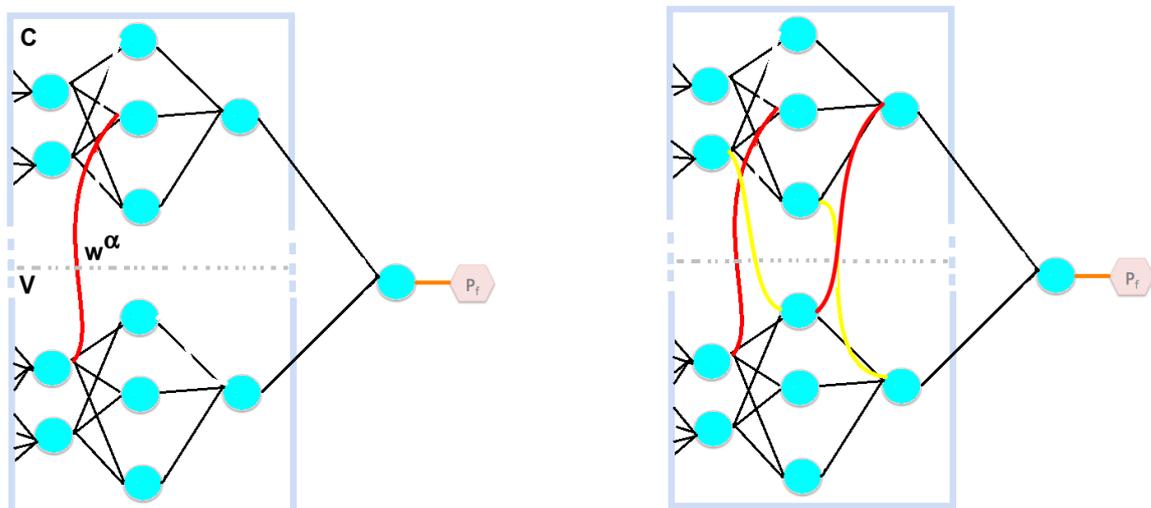
## 5 Proposta

Nesse capítulo será apresentado a proposta para conectar redes neurais artificiais treinadas a priori em suas respectivas áreas para que possam executar uma determinada tarefa. Essa proposta gerou duas vertentes, ambas usando a descida de gradiente, porém uma é implementada por meio de treinamento supervisionado e outra com o uso de aprendizado por reforço.

Os parâmetros dessas redes serão ajustados internamente, conforme seus respectivos autores representaram em seus artigos. A forma pela qual essas redes serão conectadas serão definidas por dois algoritmos de aprendizado de máquina, conforme será apresentado nas seções 5.1 e 5.2.

Para exemplificar a proposta, considere que os neurônios  $n_c$  e  $n_v$  pertencentes às duas redes neurais artificiais  $c$  e  $v$  respectivamente, possuem uma conexão externa com peso  $w_\alpha$  que conecta a saída de  $n_c$  com a entrada de  $n_v$ , o mesmo ocorre na direção inversa, assim,  $\{ \forall n \in (c,v) \exists w_\alpha, n_c \rightarrow n_v \text{ e } n_v \rightarrow n_c \}$ , ver figura 11.

Figura 11 – Esquemático simplificado do acoplamento de redes neurais.



Fonte: Autor

De modo a descrever a proposta, algumas premissas, notações e nomenclaturas serão definidas. Para relacionar quais neurônios serão interconectados as seguintes premissas e notações serão usadas:

- A matriz de pesos  $W^\alpha$  será usada para especificar todos os pesos  $w^\alpha$  pertencentes ao conjunto de pesos entre as redes, também poderá ser chamada de conjunto de pesos inter-redes;

- b) Sempre haverá, mesmo que nula, pelo menos uma conexão entre  $n_c$  e  $n_v$ ;
- c) O peso  $w_{ic,jc,iv,jv}^\alpha \in W^\alpha$  denomina o parâmetro que conecta um determinado neurônio de número  $i$  pertencente a camada  $j$  de uma rede neural  $c$  com um outro neurônio de número  $i$  pertencente a camada  $j$  de uma outra rede neural  $v$ , os índices também se aplicam aos pesos internos das redes e seus parâmetros internos;
- d) As conexões inter-redes devem ter seus pesos alterados para que haja influência de uma rede na outra, sem que seus pesos internos sejam alterados. Não há perda da especialidade de nenhuma das redes individualmente.

De maneira geral, a conexão acontece na saída de um determinado neurônio da rede  $v$  e se conecta a entrada de um determinado neurônio da camada subsequente da rede  $c$  conforme esquemático da figura 11.

Para formalizar a saída do acoplamento de redes, a dedução segue usando a equação 17 apresentada na seção 2.4.4, referente a um *stacking* de classificadores.

Inicia-se a dedução com a fórmula do *stacking*, onde  $y_k$  representa cada uma das  $k$  saídas de cada classificador especialista e  $\bar{y}$  é a saída do agrupamento após cada saída  $y_k$  ser ponderada pelo fator  $\varrho_k$ .

Sendo cada classificador especialista uma rede neural, a ideia é provar que o CMEAS é, na pior das hipóteses, um *stacking* de redes neurais. Assim, prossegue-se aplicando a equação 17 ao caso de acoplamento de redes e incluindo os pesos  $\varrho_j$  às saídas de cada rede  $y_j(x)$ , sendo  $p$  a quantidade de redes:

$$\bar{y} = \sum_{k=1}^p \varrho_k y_k(x) \quad (27)$$

Cada uma das saídas  $y_k$  dos  $k$  classificadores serão ponderadas pelo fator  $\varrho_k$  específico de cada rede. Como a dedução que segue é referente a redes neurais e a saída de cada uma dessas redes neurais é dada por 28 (RUMELHART, David E; HINTON, G. E.; WILLIAMS et al., 1988):

$$y_t = \sum_{j=1}^J f(w_{i,j} x_j) \quad (28)$$

Então, substituindo 28 em 27 obtém-se a equação 29:

$$\bar{y} = \sum_{k=1}^p \varrho_j \left[ \sum_{j=1}^J f(w_{i,j} x_j) \right] \quad (29)$$

Onde  $f(w_{i,j}x_j)$  é a função de ativação aplicada ao peso  $w_{i,j}$  e à respectiva entrada do neurônio  $x_j$ .

Nesse ponto, a equação 29 apresenta um *Stacking* de  $p$  redes neurais, com as saídas sendo acareadas por uma razão de  $\varrho$ , porém essa equação ainda possui um termo referente a ponderação após as saídas das redes. Esse termo conforme foi discutido nos capítulos anteriores é de difícil determinação e é motivo de discussões na literatura, já que a decisão do tipo de ponderamento pode influenciar diretamente na saída do agrupamento.

Para eliminar esse fator de ponderação diretamente na saída das redes serão feitas conexões entre neurônios, ao invés de ponderamentos na saída de cada rede, será considerado um gradiente de pesos  $w^\alpha$  referentes às conexões inter-redes, assim prossegue-se:

$$\sum_{j=1}^{Jc.Jv} w_{ic,jc \Leftrightarrow iv,jv}^\alpha x_{ic,jc \Leftrightarrow iv,jv} \quad (30)$$

Onde  $J$  é o número de camadas para cada uma das redes  $c$  e  $v$ . De forma que se for aplicado o termo apresentado pela equação 30 na equação 29:

$$\bar{y} = \sum_{k=1}^p \varrho_j \left[ \sum_{j=1}^J f(w_{i,j}x_j) + \sum_{j=1}^{Jc.Jv} w_{ic,jc \Leftrightarrow iv,jv}^\alpha x_{ic,jc \Leftrightarrow iv,jv} \right] \quad (31)$$

Nessa equação, de número 31,  $x_{ic,jc,iv,jv}$  representam os estados internos de cada uma das redes, e são ponderadas pelo fator  $w_{ic,jc,iv,jv}^\alpha$ .

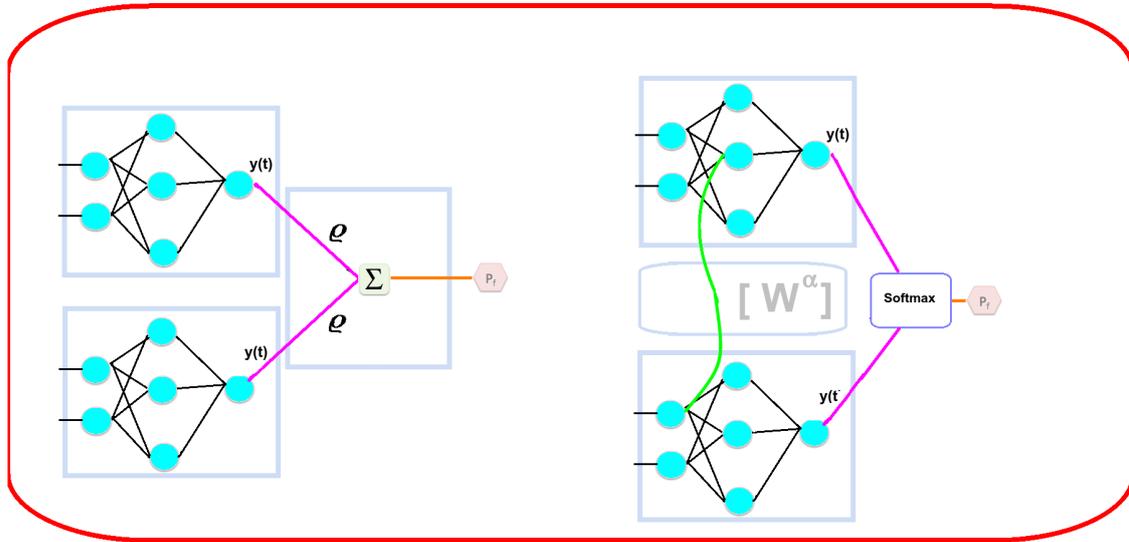
O fator de ponderação  $\varrho_j$  define o quão importante é a saída de cada especialista, porém, como os especialistas agora estão conectados e acoplados, seu valor pode ser considerado igual a 1, já que, não há mais dois especialistas. Chega-se então na equação 32.

$$\bar{y} = \sum_{j=1}^J f(w_{i,j}x_j) + \sum_{j=1}^{Jc.Jv} w_{ic,jc \Leftrightarrow iv,jv}^\alpha x_{ic,jc \Leftrightarrow iv,jv} \quad (32)$$

A prova matemática deduzida até a equação 32 transformou a ponderação de saída de um sistema de votação de um agrupamento clássico em um gradiente de pesos de uma rede neural, de modo que, todas as as interações entre redes são resumidas pela matriz  $W^\alpha$ . O mesmo conceito é apresentado nas figuras 12 e 13 de forma gráfica.

No caso de redes com mais de uma classe, espera-se que as saídas do acoplamento tenham valores muito próximos para a mesma classe, já que os pesos das redes estão sendo alterados pelos pesos da matriz  $w^\alpha$ , entretanto, ainda podem haver diferenças entre as saídas. Para lidar com essa situação será aplicada a função *Softmax*, visando identificar a saída de

Figura 12 – Acoplamento de rede com o uso da Matriz  $W^\alpha$

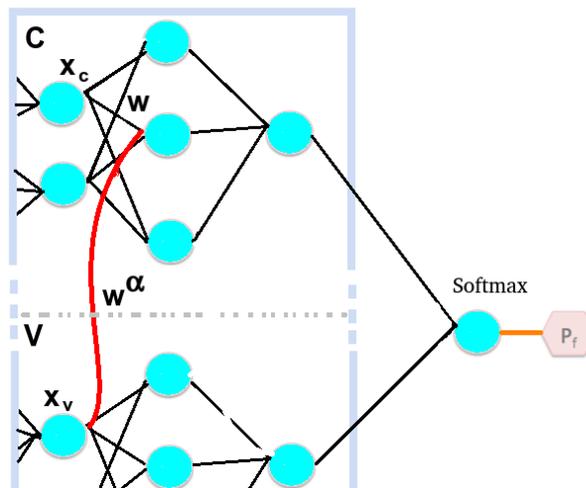


Fonte: Autor

cada classe com maior probabilidade de estar correta. Assim, a equação final do CMEAS é apresetada na equação 33.

$$\bar{y} = Softmax \left[ \sum_{j=1}^J f(w_{i,j}x_j) + \sum_{j=1}^{Jc.Jv} w_{ic,jc \Leftrightarrow iv,jv}^\alpha x_{ic,jc \Leftrightarrow iv,jv} \right] \quad (33)$$

Figura 13 – Visão do acoplamento de redes.



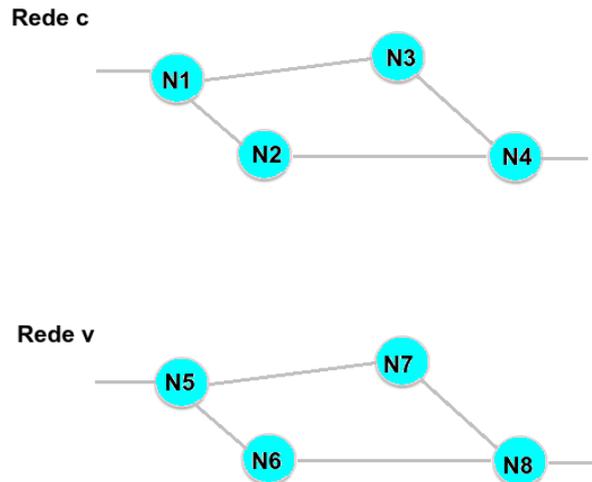
Fonte: Autor

Com essas formalizações será possível prosseguir para as seções que descrevem a forma pela qual essas redes serão treinadas e conectadas.

## 5.1 Algoritmos CMEAS vistos da Perspectiva do Algoritmo da Propagação do Erro

O treinamento da arquitetura proposta se dará nas seguintes condições: Suponha que as redes  $c$  e  $v$  estejam conectadas internamente conforme figura 14.

Figura 14 – Redes simplificadas para treinamento do Acoplamento



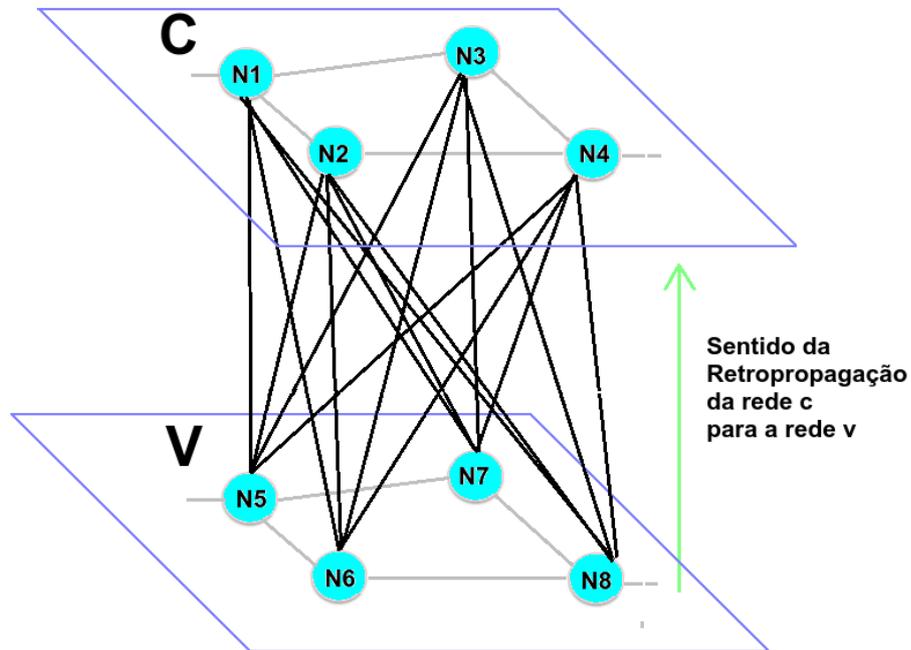
Fonte: Autor

Agora considere um acoplamento entre essas redes conforme figura 15. Se a intenção é treinar uma rede de modo que possa influenciar a outra rede, o treinamento deve ser feito nesse sentido, ou seja, transposta ao treinamento original das duas redes.

Assim, a matriz de pesos  $W^\alpha$  deverá ser transposta para efetuar esse cálculo. A propagação do erro, conforme mostrada na 15, apresenta somente o treinamento da rede  $c$  para a rede  $v$ , no entanto, essa configuração representa, apenas, o efeito de uma rede na outra. Para que o treinamento abranja a influência das duas redes, a propagação do erro deverá ser feita no outro sentido da mesma forma. Assim, as redes se conectam tal qual figura 16

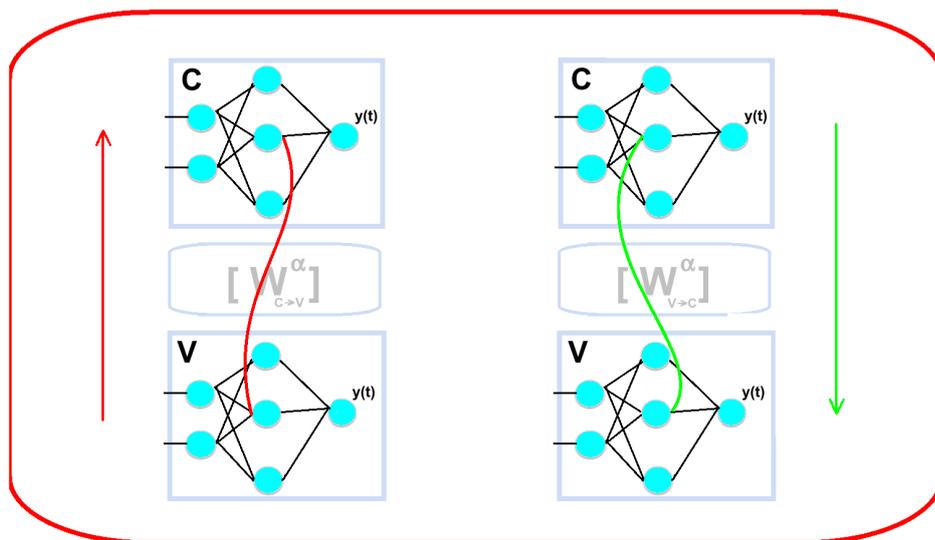
Como foi mostrada nas equações 15 e 16, verifica-se que a atualização do vetor  $w$  só depende do valor de referência, das entradas e da saída, ou seja, todas as variáveis que circundam o neurônio. Dada essa condição, é claro que, para se ter um vetor de entrada  $x_t$  para esses neurônios, são necessário vetores de entrada disponíveis nas duas redes, no entanto, não necessariamente, seria mandatório um conjunto de treinamento para essas duas redes. Em teoria, os dados podem ser extraídos diretamente de seus ambientes e isso pode ser feito de

Figura 15 – Sentido da propagação do erro para Acoplamento de duas redes neurais simples



Fonte: Autor

Figura 16 – Dois Sentidos da propagação do erro para Acoplamento de duas redes neurais simples



Fonte: Autor

maneira online, basta que o treinamento seja efetuado durante a execução da tarefa a qual esta sendo proposta.

Algoritmo 4 – CMEAS-DG Algoritmo de propagação de erros entre multiplas redes multi-camadas.

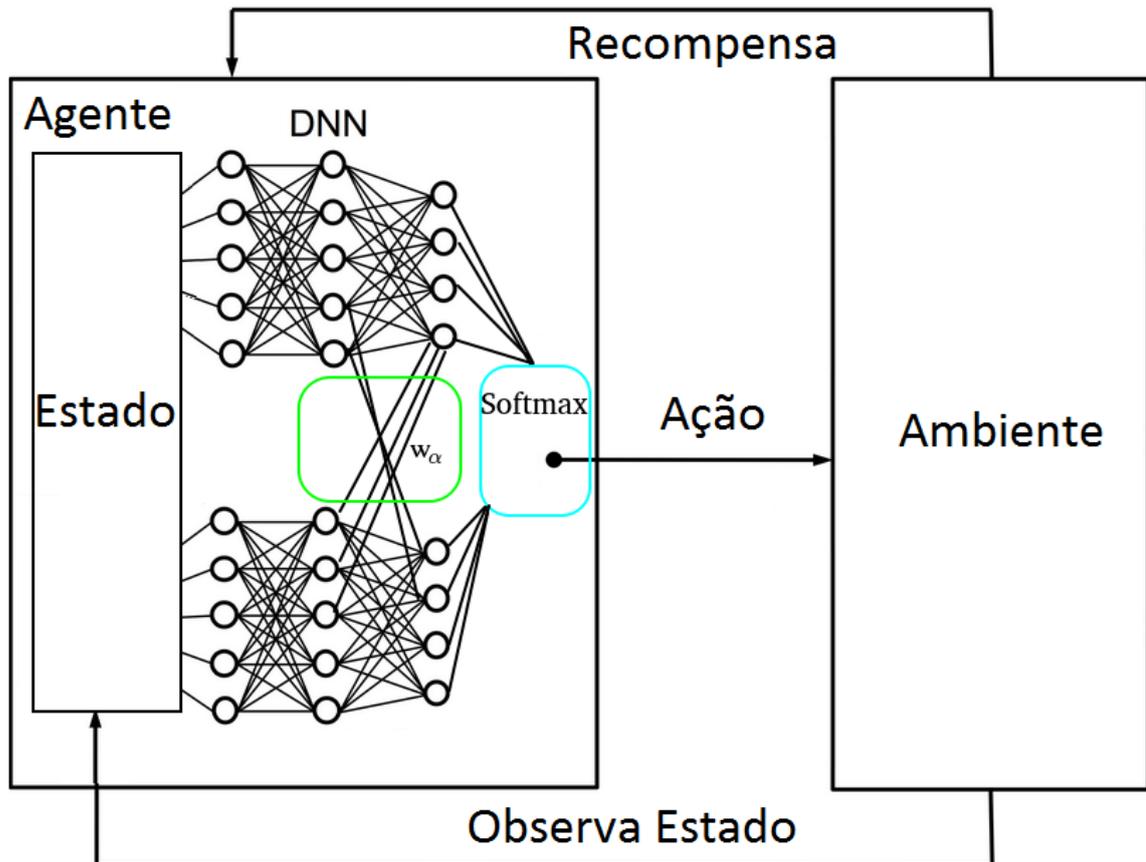
- 1 **Entrada:** Uma rede convolucional  $c$  treinada em uma determinada tarefa  $h_a$
- 2 **Entrada:** Uma rede convolucional  $v$  treinada em uma determinada tarefa  $h_b$  diferente de  $h_a$
- 3 **Entrada:** Um conjunto de treinamento de  $L$  exemplos  $\psi = \{(x_l^e, y_l^e)\}_{l=1}^L$  para  $h_a$
- 4 **Entrada:** Um conjunto de treinamento de  $M$  exemplos  $\psi = \{(x_l^e, y_l^e)\}_{l=1}^L$  para  $h_b$
- 5 **Saída:**
- 6 Define-se quantas camadas serão conectadas entre redes, encontrando as dimensões da matriz  $W^\alpha$
- 7 Escolhe-se um conjunto de pesos iniciais  $W^\alpha(0)$  e uma taxa de aprendizado  $\eta$ ;
- 8 Define-se o erro como sendo menor que um limiar como critério para fim de treinamento.
- 9 Calculam-se os erros de saída, da camada de saída e das camadas subsequentes das duas redes:
- 10 **para**  $l \leftarrow 1$  **até**  $L$  **faça**
- 11  $\delta_{y(jM),l} = (y_{jM,l}^e - y_{jM,l})$
- 12  $\delta_{jM,l} = \delta_{y(jM),l} \frac{dg(v_{jM})}{dv_{jM}}$
- 13 **para**  $m \leftarrow (M - 1)$  **até**  $0$  **faça**
- 14  $\delta_{jm,l} = \left( \sum_{j^{(m+1)}=1}^{J^{(m+1)}} \delta_{j^{(m+1)},l} w_{m+1,j^{(m+1)},jm}^\alpha \right) \frac{dg(h_{jm})}{dh_{jm}}$
- 15  $\Delta w_{m,i_c,j_c,i_v,j_v,j^{(m-1)}}^\alpha = 2\eta \sum_{l=1}^L \delta_{im,jm,l} u_{i^{(m-1)},j^{(m-1)},l}^e \cdot w_{j^{(m+1)},l}$
- 16 onde a saída do neurônio  $u_{i^{(0)},j^{(0)},l}^e$  da rede  $c$  é igual a entrada do neurônio  $x_{j^{(0)},l}^e$  ponderada pelo peso  $w^\alpha(0)_{i_c,j_c,i_v,j_v}$  na próxima camada da rede  $v$  para o mesmo exemplo.
- 17 **fim**
- 18 Quinto passo: Recalcula-se  $W^\alpha(1)$  como sendo  $W^\alpha(0) + \Delta W^\alpha$
- 19 **fim**

No algoritmo 4 não ocorre alteração nos pesos internos  $w$  conforme ocorreu na equação 1 clássica, porém, os pesos  $w_{i,j}$  fazem parte da

## 5.2 Algoritmos CMEAS vistos da Perspectiva do Algoritmo TD

Essa seção visa apresentar o acoplamento de redes com o uso do algoritmo TD. As conexões inter-redes ocorrem da mesma forma que no CMEAS-DG. Na figura 17 é possível ver o esquemático das conexões no caso do CMEAS-TD.

Figura 17 – Esquemático do CMEAS-TD



Fonte: Autor

Para formalizar a proposta é necessário definir as estruturas do aprendizado por reforço, como: os estados; a recompensa; as ações; modelo de transição; e estados iniciais e finais. O estado é definido como um vetor com um número fixo de componentes de valor real, relacionados com a tarefa a ser executada.

$$\vec{\phi}_s = (\phi_s(1), \phi_s(2), \dots, \phi_s(n)) \quad (34)$$

De agora em diante esse vetor será chamado de vetor de estado e será usado como parte de entrada da tarefa. Considera-se as ações  $a$  e recompensa  $r$  previamente definidas para a tarefa a ser executada.

O espaço de estados considerado para o uso da CMEAS-TD é contínuo. Isso significa que a maioria dos estados encontrados em exemplos de treinamento nunca terá sido experimentado exatamente antes. Portanto, é necessário generalizar esses estados anteriormente experimentados para aqueles que nunca foram vistos. Em métodos de descida de gradiente, a função valor aproximada no tempo  $t$ ,  $V_t$ , é representada como uma forma de função parametrizada vinculada com o vetor de parâmetros (WANG, Y. et al., 2017) (LEE, J. W., 2001).

No caso, este vetor é atualizado usando cada exemplo por:

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \Delta \vec{\theta}_t \quad (35)$$

Onde,

$$\Delta \vec{\theta}_t = \alpha \cdot [r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)] \nabla_{\vec{\theta}_t} V_t(s_t) \quad (36)$$

O que representa a descida de gradiente truncada, já apresentada na equação 23 para a função  $V$ . O agente, no entanto, recebe apenas erros de valor estado-ação calculados a partir de estados atualmente estimados e recompensas imediatas em vez de erros exatos (LEE, M.-C., 2009).

### 5.3 Descrição do CMEAS no mercado de Criptomoedas

Existe uma dificuldade em se obter um modelo que possa ilustrar claramente as mudanças nos preços das ações de mercado em termos de ações do agente ou políticas de investidores porque eles têm muitos tipos diferentes de políticas e ações (PULIDO, M.; MELIN; MENDOZA, 2017).

Além disso, o aprendizado por reforço exige um modelo de transição, que para esse caso é difícil de modelar já que nem sempre uma ação tomada em um determinado estado leva ao mesmo estado subsequente.

Será definido apenas o estado e a recompensa no processo de mudança de preço das ações para o aprendizado, assumindo que a política e as ações do agente podem ser refletidas implicitamente nos dados da ação de mercado.

A próxima formalização a ser considerada para modelar as mudanças nos preços das ações é o estado de uma criptomoeda em um determinado momento. Pela definição da política no capítulo 4 na equação, um estado é o parâmetro de entrada exclusivo de uma função de política. A ação é selecionada dependendo apenas de um determinado estado.

Jae Won Lee (2001) fez a suposição que as políticas gerais de investidores raramente se aproximam. Assim, quaisquer tipos de dados referidos, todos os dias, pelos investidores podem ser considerados na definição do estado de uma ação de mercado.

Os dados primitivos disponíveis diariamente para cada ação de mercado podem ser representados pelas seguintes séries temporais:

- a)  $y_0$  : O preço de abertura no dia, de forma que, apesar das negociações ocorrerem durante as 24 horas do dia, foi definida como hora de abertura as 10h GMT/UTC +0 horas (COINMARKETCAP..., 2019).
- b)  $y_c$  : O preço de fechamento no dia, da mesma forma que a abertura, o fechamento possui um GAP de 5 minutos anteriores com relação a de abertura (COINMARKETCAP..., 2019).
- c)  $y_H$  : O maior preço negociado no dia
- d)  $y_L$  : O menor preço negociado no dia
- e)  $v$  : O volume de transações capitalizadas e negociadas no dia
- f)  $m$  : A capitalização do mercado

Há também, uma variedade conceituada de indicadores e osciladores na área de análise técnica que podem ser utilizados. Esses indicadores identificam pontos de resistência ou de suporte nos dados de mercado do ativo e de acordo com algumas regras fornecem um sinal de compra ou de venda (APPEL, 2005). Já os osciladores usam regressões para tentar prever quando será o próximo sinal de compra ou de venda. Um oscilador é um indicador de análise técnica que varia ao longo do tempo dentro de uma faixa. Osciladores são usados para descobrir condições de *overbought* ou *oversold* a curto prazo <sup>1</sup> (APPEL, 2005). Osciladores comuns são MACD, ROC, RSI, CCI (APPEL, 2005). É possível, como foi feito em (LEE, J. W., 2001) criar um oscilador usando o aprendizado por reforço.

Finalmente, o estado de um papel é definido como um vetor com um número fixo de componentes de valor real, cada um dos quais corresponde a um dos dados diários brutos ou aos dados derivados enumerados acima.

<sup>1</sup>Supercomprado ou supervendido são condições de um valor mobiliário que sofreu aumento ou redução de preços inesperado e está sujeito a correção no sentido oposto

$$\vec{\phi}_s = (\phi_s(1), \phi_s(2), \dots, \phi_s(n)) \quad (37)$$

De agora em diante esse vetor será chamado de vetor de estado e será usado como parte de entrada da tarefa de predição.

A saída  $z$  da maioria dos algoritmos de aprendizado indutivo para tarefa de predição geral com base na abordagem de séries temporais tem a forma apresentada na equação 38.

$$z(t+h) = y(t+h) \quad (38)$$

Isto quer dizer, a equação 38 pretende prever o valor da série temporal após uma certa quantidade de passos de tempo  $h$ , ou seja, um horizonte de previsão fixo.

Apesar da previsão de valores não ser diretamente o foco da proposta há de se esperar que uma ordem, seja de compra, venda ou apenas um *hold*, preveja, em algum nível de abstração, qual será a tendência do ativo.

Além disso, a tarefa de previsão do valor do ativo, representando a saída como na equação 38 tem uma desvantagem, o valor  $y$  normalmente varia muito por um longo período de tempo, além disso,  $y$  para diferentes papéis podem ser diferentes da ordem de várias décadas, dificultando a criação de um modelo válido. Assim, o retorno, podendo ser lucro ou prejuízo, é definido como a variação relativa no valor, já que o valor no período de tempo anterior é frequentemente usado (HELLSTRÖM, 1998)

$$R_h(t) = 100 \frac{y(t) - y(t-h)}{y(t-h)} \quad (39)$$

A partir da equação 39, escolhendo-se um dia  $h = 1$  como horizonte e recompensa imediata  $r_t$  no tempo  $t$  determina-se a equação 40.

$$r_t = 10 \frac{y_c(t) - y_c(t-1)}{y_c(t-1)} \quad (40)$$

onde  $y_c(t)$  é o preço de fechamento de um papel em  $t$ . Com  $t$  substituído por  $t+h$ , a equação 39 corresponde ao retorno truncado do passo  $h$  com  $\gamma = 1$  na estrutura de aprendizado de reforço que é usado para calcular os limites de estados definidos na equação 19.

Pela definição da recompensa e do termo alvo na equação 21, o valor estimado atual de um estado, ou seja, a tendência de preço atual de um papel, pode ser calculado em cada período de tempo.

Segundo (GAO, X.; CHAN, 2000), as estimativas podem resultar em valores diferentes de acordo com o fator de desconto  $\gamma$ . Se  $\gamma = 0$ , somente a recompensa imediata, a taxa de variação do preço no dia seguinte, é refletida nos valores. Portanto, segundo Jae Won Lee (2001)  $\gamma$  se aproximando de 0 é adequado para predição de curto prazo. Por outro lado, se  $\gamma = 1$ , todas as recompensas após o estado são tratadas da mesma forma, independentemente da distância temporal desse estado e, portanto, com  $\gamma$  se aproximando de 1 parece ser adequado para a predição a longo prazo (LEE, J. W., 2001).

Na previsão de preços dos papéis, os preços intermediários entre o preço no passo atual e o mesmo após o horizonte temporal de interesse também são significativos. Nesse sentido, o fator de desconto possibilitaria a representação desses termos.

#### Algoritmo 5 – Algoritmo CMEAS-TD

```

1 Entrada: N redes treinadas a priori
2 Inicialize  $V(s)$  arbitrariamente
3 Inicialize  $W^\alpha$  para todas as redes com a mesma dimensão de  $M(w^\alpha)$ 
4 Escolha a taxa de aprendizado  $\alpha$ , a taxa de desconto  $\gamma$ .
5 repita
6   para cada episódio faça
7     Inicialize  $s$ 
8     repita
9       para cada passo do episódio faça
10         $a \leftarrow softmax(y_t)$ 
11        Execute a ação  $a$ 
12        Recompensa  $r$  e próximo estado  $s'$ 
13        repita
14          para cada Par de redes faça
15            Insira o vetor de dados nas duas redes
16             $y_t = \sum_{j=1}^J w_{i,j}x_j + \Phi(t)$  para as duas redes
17            Para todas as conexões em  $W^\alpha$ 
18             $\Delta W^\alpha \leftarrow \alpha \cdot [r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)] \nabla_{w^\alpha} V_t(s_t)$ , Onde
19             $W^\alpha$  é transposta a matriz  $W$  de cada uma das redes
20             $W^\alpha(1) = W^\alpha(0) + \Delta W^\alpha$ 
21          fim
22        até que todas as redes sejam visitadas;
23         $V(s_t)' \leftarrow V(s_t)$ 
24         $V(s_t) \leftarrow V(s_t) + \alpha \cdot [r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)]$ 
25         $s \leftarrow s'$ 
26      fim
27    até fim dos passos;
28    Aplique a função Softmax às saídas das redes
29  fim
30 Saída: Pesos  $W^\alpha$  de acoplamento entre redes

```

Até agora a dedução do algoritmo considerou que as redes possuíam saídas únicas para cada classe, de modo que o acoplamento entre as redes aumentasse a crença das respostas das duas redes. Essa condição é suficiente para correlacionar as duas redes dado um contexto, porém não é suficiente para tarefas que necessitam de mais classes de saída, ou para tarefas que possuam/necessitem de inter-dependência entre classes de redes diferentes.

Para tal representação, considere duas redes acopladas por uma CMEAS com diversas saídas cada uma, para agrupar essas saídas seria necessário um agrupamento do tipo *Bagging*, que ponderasse as saídas através de um peso  $\rho$ . Uma outra forma de solucionar o problema, seria o uso de agrupamento do tipo *Stacking* com uma rede neural como classificador de segundo nível. Essas abordagens requerem que a rede de segundo nível seja treinada e, além disso, que seja extraído um vetor de características das camadas de saída das redes de primeiro nível para servir de entrada para esse classificador de segundo nível (MOHAMMADI; DAS, 2016). Essas duas formas requerem treinamento de uma rede neural de segundo nível ou da determinação dos ponderamentos  $\rho$ .

Visando endereçar o problema, considere uma CMEAS de duas redes, que possuam  $k$  e  $l$  saídas, respectivamente, representando as classes  $C_k$  e  $V_l$  de cada uma delas. Considere também, que existe uma fronteira de decisão entre essas classes que produz uma terceira classe  $O$  solucionadora de uma terceira tarefa. Para determinar a função que representa a influência das classes  $C_k$  e  $V_l$  na tarefa, seria necessário conhecimento específico do domínio ou algoritmos de aprendizado de máquina. A fim de, evitar esse tipo de análise, foi incluída uma função softmax nas saídas da CMEAS.

Essa suposição simplifica o processo de treinamento, já que a CMEAS já fornece uma função erro que pode ser retropropagada. A CMEAS poderia simplificar a rede de *Stacking* em termos de topologia, já que as saídas das redes estão ponderadas. Não são necessárias muitas suposições para inferir que, quanto menor for a correlação entre as classes  $C_k$  e  $V_l$ , mais complexa será a fronteira de decisão.

Nesse capítulo foi apresentada a formalização da proposta e duas variações do acoplamento com o uso de aprendizado supervisionado por meio da descida de gradiente 5.1 e com o uso de aprendizado por reforço por meio do 5.2. Com os algoritmos apresentados, no próximo capítulo serão dispostos os experimentos no domínio do mercado de criptomoedas.

## 6 Experimentos

Nesse capítulo serão apresentados os experimentos que foram realizados, comparando a proposta e suas duas vertentes com outros algoritmos. O acoplamento com o uso de aprendizado supervisionado por meio da descida de gradiente 5.1 e com o uso de aprendizado por reforço conforme demonstrado em 5.2.

Para todos os experimentos o domínio é o mercado de criptomoedas da moeda Ethereum em relação ao dólar. O Ethereum é a segunda criptomoeda em capitalização de mercado, atrás apenas do próprio Bitcoin e em termos de tarefa de aprendizado de máquina, prever a cotação do Dólar em relação ao Ethereum pode ser considerado análogo a outras tarefas de previsão de séries temporais financeiras (MCNALLY; ROCHE; CATON, 2018).

Esse capítulo foi estruturado em três partes:

A primeira parte se refere aos procedimentos de treinamento através das seções 6.1 e 6.2, onde são explicadas as alterações nos algoritmos desenvolvidos para o mercado de ações e considerações de treinamento, respectivamente.

Na segunda parte, são apresentados os resultados dos algoritmos, sendo que foram feitos testes: Comparando o CMEAS-DG e CMEAS-TD individualmente com uma rede de notícias e com uma de cotações, variando da quantidade de camadas a serem conectadas na seção 6.3, Comparando com um agrupamento clássico (*Ensemble*) na seção 6.5, uma comparação com uma rede neural recorrente, no caso uma CNN-LSTM na seção 6.6 e, finalmente, na seção 6.7 foi feita uma comparação com uma estratégia conservadora como a compre e retêm *Buy and Hold*.

Já na terceira parte, será apresentada na seção 6.8 os testes de comparação estatística entre os algoritmos para que possa-se validar os resultados com respectivas conclusões e discussões. Visando a transparência dos dados e para facilitar a reprodutibilidade dos experimentos, todos os códigos e conjuntos de dados de todas as séries temporais estão disponíveis em <https://github.com/FinFEI/CMEAS>.

### 6.1 Convertendo Redes convolucionais do Mercado Tradicional para o Mercado de Criptomoedas

Devido a escassez de trabalhos voltados exclusivamente para o mercado de criptomoedas no início deste trabalho e devido a grande quantidade de pesquisa no campo de mercados

financeiros tradicionais, especialmente no campo de previsão de séries temporais, optou-se por usar técnicas já bastante validadas para previsão desse mercado. A ideia é se aproveitar das redes, tal qual, foram concebidas, aproveitando-se de sua topologia e hiperparâmetros. Porém, devido ao uso de informações do mercado de ações, foi necessário refazer os treinamentos de ambas as redes com um novo conjunto de dados provenientes do mercado do Ethereum.

As redes foram escolhidas segundo os seguintes dois tópicos:

As redes usadas foram:

- a) Uma rede neural profunda convolucional treinada para abstrair apenas os títulos das notícias, em forma textual, de fonte confiável, seguindo os moldes da análise fundamentalista. A rede foi usada como foi concebida, aproveitando sua topologia. No trabalho original, proposto por (DING et al., 2015), a rede foi treinada para classificar se o título das notícias é sentimentalmente falando, bom ou ruim, além disso, as notícias usadas em seu trabalho foram fornecidas pela agência Reuters, em outras palavras, o conjunto de dados veio do mercado de ações dos EUA.
- b) Uma rede neural profunda convolucional treinada com dados de preços conforme mencionado em 5.3 e com a topologia de (DENG, Yue et al., 2017), o que seria, de forma análoga à análise técnica.

O conjunto de dados foi estruturado coletando-se notícias e preços no período compreendido de 2016 a 2018 e adquiridos em COINMARKETCAP... (2019). Todos os dados coletados eram de domínio público e, quando aplicável, tiveram informações pessoais excluídas.

Considerando que se trata de treinamento supervisionado, os dados contendo os títulos das notícias foram sincronizadas no tempo com dados de cotação dos dados históricos, a fim de inferir o vínculo implícito entre turnos de baixa / alta e as notícias. Esse conjunto de dados sincronizados servem de entrada para a rede.

## 6.2 Considerações de Treinamento

Os experimentos foram realizados com o objetivo de entender a dinâmica do modelo, apresentando principalmente análises, projetadas para obter-se uma margem de lucro no mercado de criptomoedas Ethereum.

Como foi anteriormente mencionado, na seção 4.2 e corroborado por Radinsky, Davidovich e Markovitch (2012), Ding et al. (2014) e, também por Ding et al. (2015), os títulos de

notícias são mais importantes que o corpo para influenciar o mercado. No mercado tradicional é possível ter várias notícias em um único dia, porém, no caso das criptomoedas as notícias são mais escassas sendo de uma a duas por dia.

Para situações em que havia mais de uma notícia, foi escolhida apenas a primeira do dia já que, como foi apresentado por Stenqvist e Lönnö (2017), uma notícia pode causar uma alteração na cotação em até 4 horas no futuro, então contenta-se em tentar antever essa alteração e são deixadas de lado eventuais notícias posteriores no mesmo dia. O período diário foi escolhido devido ao casamento de notícias e cotações, além de, reduzir a quantidade de operações.

A rede de preços foi treinada usando, apenas, preços do mercado de criptomoedas, incluindo abertura por dia, fechamento por dia, máximo diário, mínimo diário e capitalização de mercado semelhante ao trabalho de (WANG, Y. et al., 2017) e as ações permitidas foram comprar, vender e manter posição. Para treinar sua rede neural profunda, os mesmos autores também usaram o conjunto de dados provenientes do mercado de ações dos EUA.

As séries temporais foram divididas em treinamento e teste em quatro registros de tempo diferentes, de acordo com a figura 18.

Cada divisão da sequência foi escolhida com o objetivo de determinar se as técnicas podem abstrair tendências diferentes. Para isso, os dados usados para treinar as redes neurais profundas foram divididos em um período de estabilidade, de subida proeminente, de subida suave e um período que abrange subidas e descidas acentuadas, veja a figura 18.

Com o objetivo de comparar as duas técnicas, as classes tiveram que ser alteradas de classificação em boas ou más notícias, para classificação em comprar, vender ou manter posição, além disso, o trabalho rastreia uma lista de cotações, usando a biblioteca `urllib` Python (URLLIB..., 2017), para obter os detalhes de várias empresas públicas, o que não é o caso aqui, pois, existe apenas uma criptomoeda que será usada, ou seja, o Ethereum.

Seguindo os mesmos passos de Yoon Kim (2014), o pré-processamento dos dados de treinamento para a rede de notícias seguiram a seguinte sequência:

- a) Coleta de Dados
  - Obtenha toda a lista de cotações
  - Rastrear notícias do site COINMARKETCAP... (2019) usando *BeautifulSoup* (RICHARDSON, 2007)
  - Cotações usando a biblioteca *requests* do *python*
- b) Usar o *GloVe* (PENNINGTON; SOCHER; MANNING, 2014) para treinar um vetor denso de palavras do título *headline* da notícia. O *GloVe* é um algoritmo de

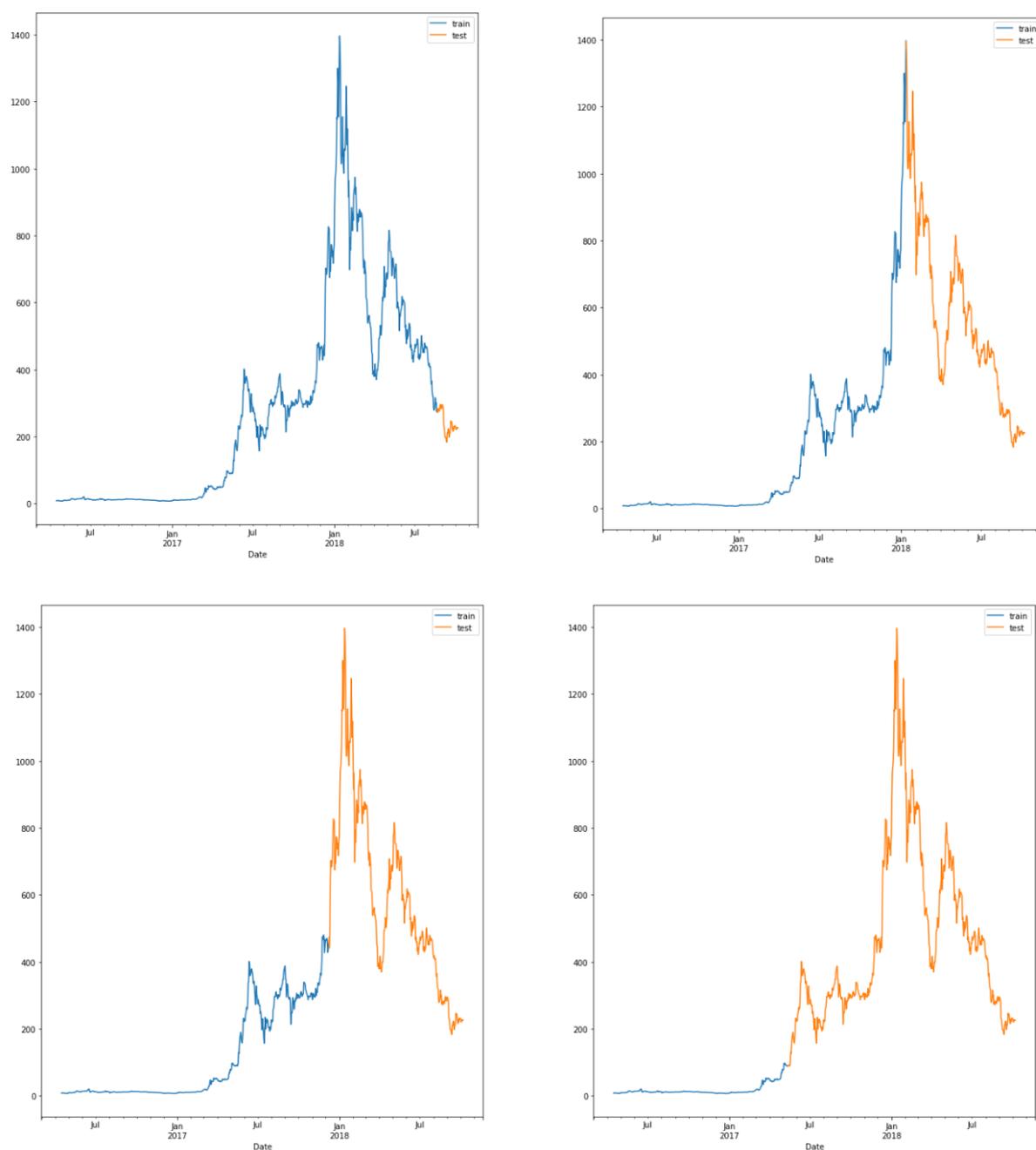


Figura 18 – Áreas de treinamento (azul) e teste (laranja) para períodos : de altas e baixas acentuadas (acima à esquerda), de altas proeminentes (acima à direita), de alta suave (abaixo à esquerda), de estabilidade (abaixo à direita)

aprendizado não supervisionado para obter representações vetoriais de palavras (PENNINGTON; SOCHER; MANNING, 2014).

- Construir a matriz de co-ocorrência palavra-palavra fatorando o log ponderado dessa matriz

c) Engenharia de Características

- Unificar o formato das palavras: unificar tempo, singular e plural, remover pontuações e remover palavras antes ou após o processamento de um texto em linguagem natural que não agregam contexto, chamadas palavras vazias (*stopwords*).
- Extrair recurso usando hash de recurso com base no vetor de palavras treinadas.
- Sequência de palavras de preenchimento aleatória, essencialmente uma matriz, para manter a mesma dimensão

### 6.3 Redes neurais convolucionais notícias e preços.

Para verificar a viabilidade da abordagem proposta, o CMEAS foi diretamente comparado às outras duas técnicas individualmente. A rede de notícias e preços foi concebida conforme (DING et al., 2015) e (DENG, Yue et al., 2017), respectivamente, sendo os sumários, para cada rede, representados nas figuras 19 e 20.

Figura 19 – Arquitetura da rede de notícias.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 18, 1, 128)	256
max_pooling2d_1 (MaxPooling2D)	(None, 1, 1, 128)	0
dropout_1 (Dropout)	(None, 1, 1, 128)	0
conv2d_1 (Conv2D)	(None, 18, 1, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 1, 1, 64)	0
dropout_1 (Dropout)	(None, 1, 1, 64)	0
flatten_1 (Flatten)	(None, 64)	0
dense_1 (Dense)	(None, 64)	4160
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	4160
dropout_3 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 2)	130
Total params: 4,546		
Trainable params: 4,546		
Non-trainable params: 0		

Fonte: “Adaptado de” Ding et al. (2015)

Figura 20 – Arquitetura da rede de cotações.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 20, 1, 150)	256
max_pooling2d_1 (MaxPooling2D)	(None, 1, 1, 150)	0
dense_1 (Dense)	(None, 100)	200
dense_2 (Dense)	(None, 50)	5050
dropout_1 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 50)	2550
dropout_2 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 50)	2550
dropout_3 (Dropout)	(None, 50)	0
dense_5 (Dense)	(None, 40)	2040
dense_6 (Dense)	(None, 1)	41
Total params: 12,431		
Trainable params: 12,431		
Non-trainable params: 0		

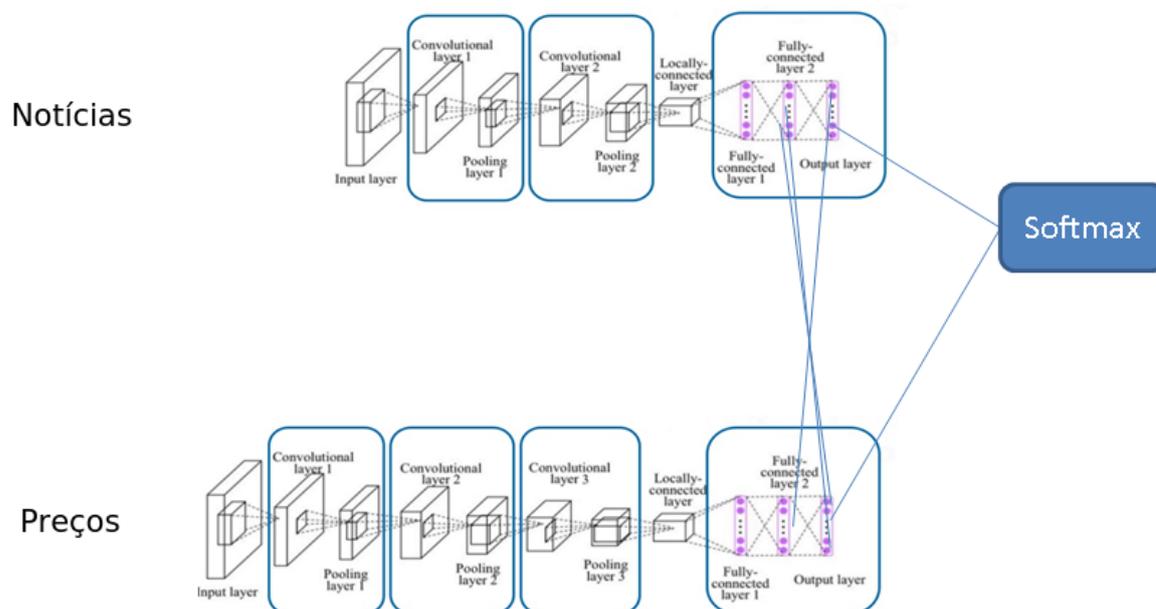
Fonte: “Adaptado de” Yue Deng et al. (2017)

Visando identificar a quantidade de camadas para fazer o acoplamento, definiu-se que as conexões são permitidas apenas entre as  $1 \leq N \leq 3$  últimas camadas. Foram utilizadas três métricas, a razão Sharpe, o lucro no período e o retorno sobre o investimento inicial, conforme descrição na seção 4.3. A proporção Sharpe examina o desempenho de um investimento enquanto ajusta seu risco, enquanto o ROI calcula em quanto tempo o investimento retornará.

As experiências foram conduzidas em um processador Intel Core i7 7700k de 3,6 Ghz com 16 GB de RAM DDR4 e duas NVIDIA GTX1070ti sli em modo ponte. Os dados foram coletados usando várias bibliotecas Python, incluindo Keras (CHOLLET et al., 2015) com Tensorflow (MARTÍN ABADI et al., 2015) como back-end, urllib (URLLIB..., 2017), no Linux Ubuntu 16.04.

A figura 21 mostra o esquemático das conexões entre as redes e sua saída pela função Softmax e, na figura 22, é possível ver os movimentos de compra (seta azul), venda (seta vermelha) do CMEAS-DG, com duas camadas interconectadas, no período de estabilidade do Ethereum, sendo que, ações de retenção (*Hold*) não são mostradas por uma questão de simplicidade e clareza. Considerando uma contribuição de 10000.00 dólares, os resultados, após 100 tentativas, são mostrados nas tabelas 3 à 6.

Figura 21 – Esquemático do acoplamento de redes neurais na sua forma profunda.



Fonte: Autor

Figura 22 – Ações tomadas pelo algoritmo CMEAS-DG no período de estabilidade com duas camadas.



Fonte: Autor

A rede convolucional de preços usa a função de ativação *Unidade Linear Retificadora* (*Rectified Linear Units*) (ReLU), como otimizador foi usado o Adamax (KINGMA; BA, 2014) com uma taxa de aprendizado de 0.0001

Considerando que quanto maior for o índice Sharpe, mais vantajoso um investimento é sobre outro, é possível verificar nas tabelas 3 à 6 que comparada com as redes individualmente treinadas o CMEAS-DG foi um melhor investimento, quando se leva em conta o lucro ajustado com o risco da aplicação. Nas outras métricas o CMEAS-DG também foi melhor.

Tabela 3 – Treinamento com propagação do erro em período de estabilidade

	Camadas	Sharpe		Fator de Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-DG	1	2.89	0.01	1.21	0.03	1.34	0.07
CMEAS-DG	2	3.29	0.02	1.36	0.02	1.59	0.05
CMEAS-DG	3	2.92	0.01	1.25	0.03	1.56	0.08
Preços		2.18	0.04	1.15	0.08	1.21	0.04
Notícias		1.98	0.06	1.06	0.09	1.12	0.03

Tabela 4 – Treinamento com propagação do erro em período de alta leve

	Camadas	Sharpe		Fator de Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-DG	1	4.38	0.03	1.27	0.03	1.28	0.07
CMEAS-DG	2	6.29	0.04	1.39	0.02	1.48	0.05
CMEAS-DG	3	5.08	0.04	1.25	0.03	1.46	0.08
Preços		3.39	0.04	1.22	0.07	1.19	0.04
Notícias		2.66	0.06	1.11	0.06	1.08	0.03

Tabela 5 – Treinamento com propagação do erro em período de alta acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-DG	1	5.44	0.03	1.23	0.05	1.21	0.02
CMEAS-DG	2	7.11	0.04	1.42	0.04	1.31	0.03
CMEAS-DG	3	5.77	0.07	1.34	0.06	1.48	0.04
Preços		5.07	0.08	1.15	0.03	1.25	0.07
Notícias		4.34	0.12	1.02	0.10	1.16	0.06

Tabela 6 – Treinamento com propagação do erro em período englobando estabilidade, alta e queda acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-DG	1	4.58	0.05	1.47	0.09	1.29	0.072
CMEAS-DG	2	6.83	0.04	1.56	0.12	1.36	0.075
CMEAS-DG	3	6.19	0.07	1.68	0.08	1.42	0.078
Preços		3.09	0.10	1.33	0.13	1.18	0.02
Notícias		1.89	0.09	1.26	0.08	1.15	0.02

Ainda nas tabelas 3 à 6, verifica-se, pelo lado da quantidade de camadas interconectadas, que a interconexão com apenas uma camada teve um desempenho geral pior do que a de duas e três camadas nas três métricas.

#### 6.4 Com aprendizado por reforço TD

O experimento CMEAS com aprendizado por reforço seguiu os mesmos preceitos do treinamento com a propagação do erro. A rede neural convolucional treinada apenas usando preços do mercado de cripto moedas, incluindo abertura, fechamento, máximo diário, mínimo diário e capitalização de mercado semelhante ao trabalho de (WANG, Y. et al., 2017) e as ações permitidas foram comprar, vender e manter posição.

Tabela 7 – Treinamento com aprendizado por reforço em período de estabilidade

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-TD	1	5.33	0.01	1.19	0.03	1.38	0.06
CMEAS-TD	2	5.37	0.02	1.41	0.02	1.55	0.04
CMEAS-TD	3	3.22	0.01	1.46	0.03	1.44	0.09

Tabela 8 – Treinamento com aprendizado por reforço em período de alta leve

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-TD	1	7.02	0.03	1.26	0.03	1.31	0.07
CMEAS-TD	2	8.13	0.04	1.44	0.02	1.35	0.05
CMEAS-TD	3	5.59	0.04	1.39	0.03	1.38	0.08

Tabela 9 – Treinamento com aprendizado por reforço em período de alta acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-TD	1	3.58	0.03	1.12	0.06	1.10	0.02
CMEAS-TD	2	5.21	0.04	1.16	0.06	1.11	0.02
CMEAS-TD	3	4.92	0.05	1.20	0.05	1.14	0.02

A exemplo do que ocorreu com o CMEAS-DG, o CMEAS-TD com duas camadas teve o índice Sharpe melhor nos períodos de estabilidade, alta leve e alta acentuada individualmente, porém, quando são considerados todos os períodos o CMEAS-TD com três camadas foi superior nas métricas apresentadas.

Tabela 10 – Treinamento com aprendizado por reforço em período englobando estabilidade, alta leve, alta e queda acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
CMEAS-TD	1	3.69	0.02	1.39	0.08	1.29	0.03
CMEAS-TD	2	4.44	0.04	1.62	0.05	1.34	0.05
CMEAS-TD	3	5.35	0.02	1.69	0.02	1.42	0.04

## 6.5 Comparando com ensemble

Nesse experimento a comparação foi feita usando-se apenas as duas últimas camadas já que, apesar, do treinamento com 3 camadas apresentar fator de lucro melhor, as redes com duas camadas tiveram, na média, melhores índices Sharpe. Além disso, utilizar duas camadas apresentou um tempo de treinamento menor.

Para o experimento de comparação com o Ensemble a construção de vários modelos possíveis em vez de um, treiná-los de forma independente na mesma tarefa e votar sobre a probabilidade de suas respostas. O Ensemble utilizado foi um sistema de votação conforme utilizado por B. Yang, Gong e W. Yang (2017). Esse sistema foi utilizado para previsão do índice de Shanghai e SZSE, porém, é importante ressaltar que no experimento, ao invés, da previsão do índice, a saída do *Bagging* foi alterada para compra, vende e retêm.

Tabela 11 – Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de estabilidade

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
Ensemble	2	4.37	0.01	1.19	0.03	1.43	0.07
CMEAS-DG	2	3.29	0.02	1.36	0.02	1.59	0.05
CMEAS-TD	2	3.96	0.01	1.33	0.03	1.56	0.08

Tabela 12 – Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de alta leve

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
Ensemble	2	5.32	0.03	1.21	0.08	1.34	0.07
CMEAS-DG	2	6.78	0.04	1.34	0.05	1.59	0.05
CMEAS-TD	2	8.13	0.04	1.44	0.02	1.35	0.05

Verificando nas tabelas 11 à 14, é possível identificar que no período de estabilidade o Ensemble teve um índice de Sharpe melhor, porém, pior rendimento. Nas outras métricas e períodos o CMEAS-DG com 2 camadas foi melhor, com exceção da alta leve, período pelo qual o CMEAS-TD apresentou melhores métricas.

Tabela 13 – Índices para Ensemble, CMEAS-DG e CMEAS -TD em período de alta acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
Ensemble	2	3.68	0.03	1.23	0.05	1.21	0.02
CMEAS-DG	2	7.11	0.04	1.42	0.04	1.31	0.03
CMEAS-TD	2	5.21	0.04	1.16.	0.06	1.11	0.02

Tabela 14 – Índices Sharpe, Fator de Lucro e ROI para Ensemble, CMEAS-DG e CMEAS -TD em período englobando estabilidade, alta e queda acentuada

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
Ensemble	2	5.12	0.04	1.38	0.04	1.29	0.06
CMEAS-DG	2	6.83	0.04	1.56	0.12	1.36	0.075
CMEAS-TD	2	4.44	0.04	1.62	0.05	1.34	0.05

## 6.6 Comparação com o LSTM

A rede neural recorrente do tipo *Arquitetura de Rede Neural Convolutacional com LSTM (Convolutional LSTM Neural Network)* (CNN-LSTM) foi utilizada como meio de comparação já que é normalmente utilizada para previsão de séries temporais. A rede aqui utilizada é uma CNN-LSTM.

Diferentemente do que foi feito com as CNN, os dados sincronizados foram inseridos sequencialmente, visando uma previsão de tendência. Devido a esse fato, a comparação foi feita englobando o período incluindo estabilidade, alta proeminente, alta acentuada e queda abrupta. A rede utilizada foi a mesma descrita por KERAS... (2019).

A previsão do valor da cotação é usada como identificador de compra ou venda, de modo que, se o valor previsto pela rede menos o valor anterior, segundo equação 39 resultasse em lucro ou prejuízo previsto relativo a um limiar, determinado empiricamente, emitia-se uma ordem segundo equações 6.6.

- Se a previsão de lucro  $> 50$ , emite-se Ordem de Compra.
- Se a previsão de lucro/prejuízo  $-10 > \text{limiar} \leq 50$ , mantêm-se posição.
- Se a previsão de lucro/prejuízo  $\text{limiar} \leq -10$ , emite-se Ordem de Venda.

Figura 23 – Sumário da estrutura da CNN-LSTM.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 500, 32)	160000
conv1d_1 (Conv1D)	(None, 500, 32)	3104
max_pooling1d_1 (MaxPooling1D)	(None, 250, 32)	0
lstm_1 (LSTM)	(None, 100)	53200
dense_1 (Dense)	(None, 1)	101
Total params: 216,405		
Trainable params: 216,405		
Non-trainable params: 0		

Fonte: “Adaptado de” KERAS... (2019)

Tabela 15 – Comparação com o LSTM

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
LSTM	-	5.58	0.05	1.56	0.12	1.24	0.07
CMEAS-DG	2	6.83	0.04	1.56	0.12	1.36	0.075
CMEAS-TD	2	4.44	0.04	1.62	0.05	1.34	0.05

## 6.7 Comparação com o Buy and hold

Na análise fundamentalista, a estratégia *Buy and Hold* é o ato de comprar um ativo e mantê-lo em carteira por um longo prazo, tipicamente abrange período de 5 anos à décadas, de forma a se beneficiar dos rendimentos e valorizações que o papel por ventura venha a apresentar no futuro. Possui como vantagens a redução de custos de transação como corretagens, emolumentos e impostos, e proteção contra oscilações espúrias e pontuais provindas de boatos e outros que não afetam diretamente a saúde financeira da empresa, sendo considerada uma estratégia conservadora.

A estratégia *Buy and Hold* se baseia no fato de, apenas empresas consideradas de alta qualidade pela análise fundamentalista serem passíveis de compra, já que empresas com desempenho econômico duvidoso tendem a ser mais propensas a apresentar problemas financeiros e a ter seu ativos desvalorizados no longo prazo.

No caso do mercado de criptomoedas, ainda não se têm certeza do seu futuro e, nem mesmo como, até onde pôde se verificar, de alguma forma, medir seu desempenho econômico,

da mesma forma que é feito com papéis de uma empresa. Além disso, devido a sua natureza recente, a moeda Ethereum ainda não tem histórico de notícia e valores comparáveis com a escala de tempo do longo prazo.

Entretanto, a comparação com o *Buy and Hold* foi feita, pois, apesar de apresentar essas desvantagens, ela é, ainda, uma métrica bastante aceita para comparação de rendimentos.

Por se tratar de um período relativamente curto para os padrões de longo prazo, o mesmo conjunto de treinamento em período englobando estabilidade, alta e queda acentuada, foi utilizado. A idéia é pegar o maior período de teste possível, fazendo uma compra no início de cada período e uma venda no fim do mesmo período, sendo os resultados apresentado na tabela 16.

Tabela 16 – Comparação com o *buy and hold*

	Camadas	Sharpe		Lucro		ROI	
		Média	$\sigma$	Média	$\sigma$	Média	$\sigma$
Buy and Hold	2	1.75	0.05	1.17	0.09	1.09	0.072
CMEAS-DG	2	6.83	0.04	1.56	0.12	1.36	0.075
CMEAS-TD	2	4.44	0.04	1.62	0.05	1.34	0.05

## 6.8 Testes Estatísticos

Os resultados apresentados nas seções 6.3, 6.4, 6.5, 6.6 e 6.7 não são suficientes para considerar que um algoritmo pode ter um desempenho melhor que o outro. Normalmente, o teste pareado T-student (STUDENT, 1908) é usado para inferir estatisticamente que um par de métodos é diferente. No entanto, como as amostras não podem ser consideradas normalmente distribuídas por observação direta, o teste de Wilcoxon (DERRICK; WHITE, 2017) foi usado como uma alternativa.

### 6.8.1 Teste de Wilcoxon

O teste não paramétrico de Wilcoxon é um teste de hipótese usado para comparar amostras correspondentes para determinar se as classificações médias da população diferem (DERRICK; WHITE, 2017). No trabalho original de Derrick e White (2017) foi definida a hipótese nula  $H_0$  como a diferença dos pares que tem uma distribuição simétrica em torno de zero, enquanto  $H_1$  não. Aqui  $H_0$  é a hipótese de que não há diferença entre algoritmos.

O parâmetro usado nos testes estatísticos foi o lucro ajustado pelo risco, devido a sua natureza de ser um indicador que caracteriza a eficácia e estabilidade da estratégia, o índice Sharpe foi escolhido para comparar os algoritmos. Os outros dois índices, apesar de apresentarem resultados relevantes no sentido do lucro e retorno, não levam em conta o risco do investimento.

### 6.8.1.1 CMEAS-DG com relação às redes de Notícias e Cotações

O primeiro teste foi feito para validar os resultados do CMEAS-DG contra as outras duas redes neurais profundas. O objetivo do teste de Wilcoxon é comparar os desempenhos de um par de algoritmos, a fim de verificar se há diferenças significativas entre seus resultados. Os resultados do teste de Wilcoxon são apresentados na tabela 17 após dez execuções do algoritmo e com as duas últimas camadas conectadas.

Tabela 17 – Resultado do Teste de Wilcoxon considerando o índice Sharpe para redes de notícias e Preço em relação ao CMEAS-DG

Wilcoxon	Estabilidade	Alta Proeminente	Alta Leve	Alta e Queda Acentuada
CMEAS-DG x Notícias	0.99938	0.96440	0.97438	0.98851
CMEAS-DG x Cotações	0.99974	0.96431	0.97609	0.98933

Considerando o índice Sharpe e um valor referencial de  $p = 0,05$ , é possível verificar na tabela 17 e rejeitar toda a hipótese  $H_0$ , uma vez que todos os valores  $p < 0,05$ , portanto, o teste de Wilcoxon mostra que o CMEAS-DG é melhor que cada uma das redes individualmente em todas as condições de treinamento.

### 6.8.1.2 CMEAS-DG com relação ao e CMEAS-TD

O teste de Wilcoxon foi utilizado para validar os resultados da CMEAS-DG contra a rede proposta com TD. Para os testes foram consideradas todas as possibilidades de conexão de camadas, sendo a única separação os períodos de treinamento. Ver tabela 18.

Tabela 18 – Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS-DG em relação ao CMEAS-TD

Wilcoxon	Estabilidade	Alta Proeminente	Alta Leve	Alta e Queda Acentuada
CMEAS-DG x CMEAS-TD	0.65071	0.74319	0.68675	0.77287

O teste de Wilcoxon não demonstrou diferenças significativas entre a CMEAS treinada com propagação de erro e com método das diferenças temporais. Assim não é possível rejeitar  $H_0$ , uma vez que todos os valores  $p > 0,05$ .

### 6.8.1.3 CMEAS-DG e CMEAS-TD com relação ao Ensemble

O teste de Wilcoxon foi utilizado para validar os resultados do CMEAS-DG contra o ensemble. Para os testes foram consideradas todas as possibilidades de conexão de camadas, sendo a única separação os períodos de treinamento. Ver tabela 18.

Tabela 19 – Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS-DG em relação ao CMEAS-TD

Wilcoxon	Estabilidade	Alta Proeminente	Alta Leve	Alta e Queda Acentuada
CMEAS-DG x Ensemble	0.97728	0.9127	0.97558	0.99252
CMEAS-TD x Ensemble	0.9351	0.99131	0.98425	0.99844

O teste de Wilcoxon demonstrou diferenças significativas entre o CMEAS-DG com relação ao ensemble. Houveram duas exceções, no caso de período de alta leve na comparação com o CMEAS-DG e no caso do período de estabilidade na comparação com o CMEAS-TD. Ver tabela 19.

### 6.8.1.4 CMEAS-DG e CMEAS-TD com relação ao CNN-LSTM

O teste de Wilcoxon foi utilizado para validar os resultados do CMEAS-DG e do CMEAS-TD contra o CNN-LSTM. Para os testes foram consideradas todas as possibilidades de conexão de camadas, sendo a única separação os períodos de treinamento. Ver tabela 18.

O teste de Wilcoxon demonstrou diferenças significativas entre o CMEAS-DG e CMEAS-TD com relação ao CNN-LSTM. Houveram duas exceções, no caso de período de alta acentuada na comparação com o CMEAS-DG e no caso do período de estabilidade na comparação com o CMEAS-TD, esse último, porém, muito próximo do valor de  $0,05865 > 0,05$ . Ver tabela 20.

Tabela 20 – Resultado do Teste de Wilcoxon considerando o índice Sharpe para o CMEAS-DG e CMEAS-TD em relação ao CNN-LSTM

Wilcoxon	Estabilidade	Alta Proeminente	Alta Leve	Alta e Queda Acentuada
CMEAS-DG x CNN-LSTM	0.95528	0.9615	0.93941	0.97168
CMEAS-TD x CNN-LSTM	0.94135	0.9757	0.96222	0.96556

## 6.9 Discussão

Os experimentos visaram demonstrar uma visão do desempenho dos algoritmos com relação às métricas estabelecidas e depois validar esses resultados através de teste estatístico.

Comparando os resultados de duas e três camadas, nas métricas ROI e fator de lucro, os resultados são semelhantes e não é possível distinguí-los. Usar três camadas exigiu mais conexões inter-redes, tanto para o CMEAS-DG quanto para o CMEAS-TD e, portanto, um tempo de treinamento maior. A interconexão com apenas uma camada teve um desempenho geral pior do que a de duas e três camadas nas três métricas.

Com relação aos períodos de treinamento, o desempenho do período de estabilidade foi, no geral, pior, apresentando índices Sharpes menores em relação aos outros períodos. A comparação estatística também não demonstrou haver diferenças significativas entre as implementações do CMEAS-DG e do CMEAS-TD.

Nas comparações com o Ensemble e com o CNN-LSTM, houveram alguns períodos individualmente de treinamento nos quais o CMEAS-DG obteve métricas piores, por exemplo, é possível identificar que no período de estabilidade o Ensemble teve um índice de Sharpe melhor, porém, obteve um pior rendimento. Nas outras métricas e períodos o CMEAS-DG com 2 camadas foi melhor, com exceção da alta leve, período pelo qual o CMEAS-TD apresentou melhores métricas. Foi feita, também, uma comparação com a estratégia conservadora *Buy and Hold*, na qual os resultados foram expressivos considerando as métricas, porém, pelos motivos apresentados na seção 6.7, pode haver divergências com relação a essa conclusão, já que uma das vantagens da estratégia é se valer da saúde financeira e já que há a necessidade de mais dados, da ordem de mais de 5 anos.

No próximo capítulo, será exposto um estudo de caso do uso do CMEAS em um robô automatizado desenvolvido para executar ordens em diversas corretoras de criptomoedas visando a estratégia conhecida como arbitragem.

## 7 Estudo de Caso

Nesse capítulo será apresentado um robô de arbitragem automatizado de criptomoedas.

Para tal, a seção 4.6 começa adentrando em alguns conceitos de *Blockchain*, mineração de criptomoedas e mercado através da *Corretora de Criptomoedas (Cryptocurrency Exchange)* (CE), para que fique claro como as operações dentro do *Blockchain* funcionam e como se relacionam com as carteiras (*wallets*) e *exchanges*.

O capítulo prossegue apresentando a estratégia de arbitragem, que nada mais é, do que um método reativo de se auferir lucro. Com a estratégia definida é possível seguir para como a arbitragem será controlada do ponto de vista do robô, para isso é feita uma brevíssima apresentação de grafos orientados. Esse passo é necessário, pois, o robô usa grafos orientados em um banco de dados NoSQL para calcular rotas, a fim de, restaurar saldos das carteiras após uma operação de arbitragem, o robô também efetua podas automáticas em algumas arestas do grafo e ordena as melhores oportunidades de arbitragem, de modo que o processo de arbitragem não seja interrompido. Com os aspectos da parte operacional e passiva do robô retratada e definida por meio de um grafo orientado, o desenvolvimento prossegue com o uso de uma *CMEAS com descida de gradiente (CMEAS-DG)* com duas camadas conectadas para identificar ativamente qual oportunidade de arbitragem é mais vantajosa, sendo, a conclusão apresentada ao final.

### 7.1 Arbitragem e as Criptomoedas

Na Bolsa de valores tradicional ordens de compra e venda são emitidas por pessoas através de corretoras cadastradas, algo análogo acontece no caso das criptomoedas, onde existe uma instituição, chamada *Corretora de Criptomoedas (Cryptocurrency Exchange)* (CE), que é responsável pelas negociações entre pessoas que visam a conversão das criptomoedas em moedas fiduciárias e/ou em outras criptomoedas, através do *Blockchain* e de transações bancárias tradicionais. A CE cria carteiras para armazenar o saldo de criptomoedas e moedas fiduciárias, porém, é conveniente ressaltar que transações de transferência entre a mesma criptomoeda pode ser enviada diretamente pelo *Blockchain* daquela moeda.

No mercado de criptomoedas, a disparidade de cotações ocorre porque os mercados de cripto-ativos não são centralizados, ou seja, o valor do ativo não é propagado entre as corretoras, ao invés disso cada *trader* pode negociar usando qualquer valor que entenda que a moeda valha

em sua própria cotação em comparação com a moeda Fiat local. Como apresentada na seção 4.2, essa disparidade é uma medida da ineficiência do mercado de criptomoedas.

Uma das mais antigas e comuns práticas do mercado de capitais é a Arbitragem. Em ciências econômicas, o processo conhecido como Arbitragem consiste em lucrar com as diferenças de preços entre dois ativos financeiros iguais, porém, negociados em mercados ou sobre códigos diferentes (SHLEIFER; VISHNY, 1997). A oportunidade de arbitragem acontece quando ocorre uma disparidade irracional temporária com os preços de um determinado ativo.

As taxas para execução de uma ordem de compra ou venda são pagas diretamente à corretora, de modo que, ordens parcialmente executadas são taxadas distintamente. As transferências dependem de taxas de mineração a serem oferecidos aos mineradores para execução das transferências das criptomoedas entre seu próprio *Blockchain*.

### 7.1.1 Tipos de Arbitragem

Existem diferenças entre os valores praticados em cada *Corretora de Criptomoedas (Cryptocurrency Exchange)* (CE) para uma mesma determinada moeda. No caso do mercado financeiro tradicional essa diferença é quase inexistente, já que os mercados são centralizados e mesmo que haja um transiente de valores, este seria rapidamente compensado por algoritmos de *Negociação de Altíssima Frequência (Ultra High Frequency Trading)* (UHFT). A arbitragem tende a causar a convergência dos preços dos ativos de diferentes mercados. A velocidade de convergência de preços é uma medida da eficiência do mercado: quanto mais rapidamente a diferença de preços é eliminada, mais eficiente é o mercado (SHLEIFER; VISHNY, 1997). Esse cenário difere do mercado das criptomoedas, já que podem ser cotadas em duas corretoras diferentes com preços substancialmente diferentes entre si. Neste caso, o arbitrador, que executa a arbitragem, que visualizar essa oportunidade, pode comprar o ativo no qual o preço está mais baixo em uma corretora e vendê-lo em outra aproveitando o lapso de tempo existente, onde o preço está mais elevado, auferindo, com isso, o lucro das diferenças de preços enquanto o preço ainda não se ajustou.

De maneira geral, existem 4 tipos de arbitragem:

- a) Arbitragem Cambial: operação com o câmbio de moeda em dois mercados financeiros diferenciados;

- b) Arbitragem de bolsa a bolsa: compra e venda de um ativo que esteja fluando em diferentes bolsas;
- c) À vista contra a prazo: busca aproveitar as oscilações dos mercados relacionados aos ativos a prazo, como por exemplo no mercado futuro.
- d) Arbitragem triangular: similar a cambial, o trader converte uma moeda para outra em um banco, converte essa segunda moeda para outra em um segundo banco e, assim sucessivamente até retornar para a original em um banco. Esse tipo exige o uso de diferentes instituições financeiras e sofre de taxações consecutivas para cada transação. (KAKUSHADZE; YU, W., 2019)

Cabe uma ressalva ao discernir as arbitragens do tipo Cambial e de Bolsa a Bolsa, embora mercados financeiros diferenciados e bolsas não se encaixem diretamente no mercado de criptomoedas, é possível comparar bolsas com corretoras. Por essa definição, a arbitragem que está sendo proposta, se enquadra na arbitragem do tipo de Bolsa a Bolsa, porém, como será demonstrado nas seções a seguir, o robô também faz a conversão entre moedas Fiat por meio de diversos *Blockchains*, sendo, portanto, podendo ser enquadrada, também no tipo Cambial.

## 7.2 Grafos Orientados

O primeiro trabalho na história da teoria de grafos remete ao rigor matemático de Leonhard Euler ao desenvolver uma técnica para provar que o problema das sete pontes de *Königsberg* não possuía solução (EULER, 1741).

Um grafo é um par de conjuntos do tipo  $G = (V, A)$ , de modo que  $A \subseteq (V)^2$ , assim, os elementos de  $A$  são subconjuntos de  $2V$ . Os elementos de  $V$  são denominados Vértices ou nós, do grafo  $G$ , enquanto  $A$  são as arestas que conectam esses nós (DIESTEL, 2017). O Grafo ou Dígrafo dirigido possui uma função de peso  $w(u, v) : A \leftarrow \mathbb{R}$  onde,  $u$  e  $v$  são vértices, que é equivalente ao custo entre esses dois nós do grafo por meio de uma determinada aresta.

O peso de um determinado caminho  $p = \langle v_0, v_1, \dots, v_k \rangle$  é a soma dos pesos das arestas que conectam os nós pelos quais se deseja caminhar. Também fica definido, se houver, que o menor caminho entre  $u$  e  $v$  é  $\delta(u, v) = \min(w(p) : u \implies v)$

Grafos normalmente utilizados para solucionar o problema de menor caminho, que consiste em determinar o menor caminho entre um vértice inicial  $s \in V$  e todos os demais vértices de  $V$ . Essa propriedade é útil para encontrar o menor caminho entre duas cidades ou encontrar o menor custo entre transações de criptomoedas, como será apresentado.

Para evitar ambiguidades notacionais será assumido tácitamente que  $V \cap A = \phi$  (DI-ESTEL, 2017). Normalmente na literatura encontra-se o acrônimo para  $E$  *Edges*, ou arestas, porém, para não haver ambiguidades com o  $E$  de conjunto de experiências do aprendizado por reforço, o acrônimo para  $E$ , arestas, foi substituído pelo acrônimo  $A$ .

O problema de menor caminho em um grafo consiste em determinar o menor caminho entre um vértice inicial  $s \in V$  e todos os demais vértices de  $V$ . Diversos algoritmos foram propostos para encontrar o menor caminho em grafos orientados, sendo os mais famosos o de Dijkstra (DIJKSTRA et al., 1959) e Bellman-Ford (BELLMAN et al., 1978) para arestas com pesos negativos.

### 7.3 Robô automatizado de Arbitragem

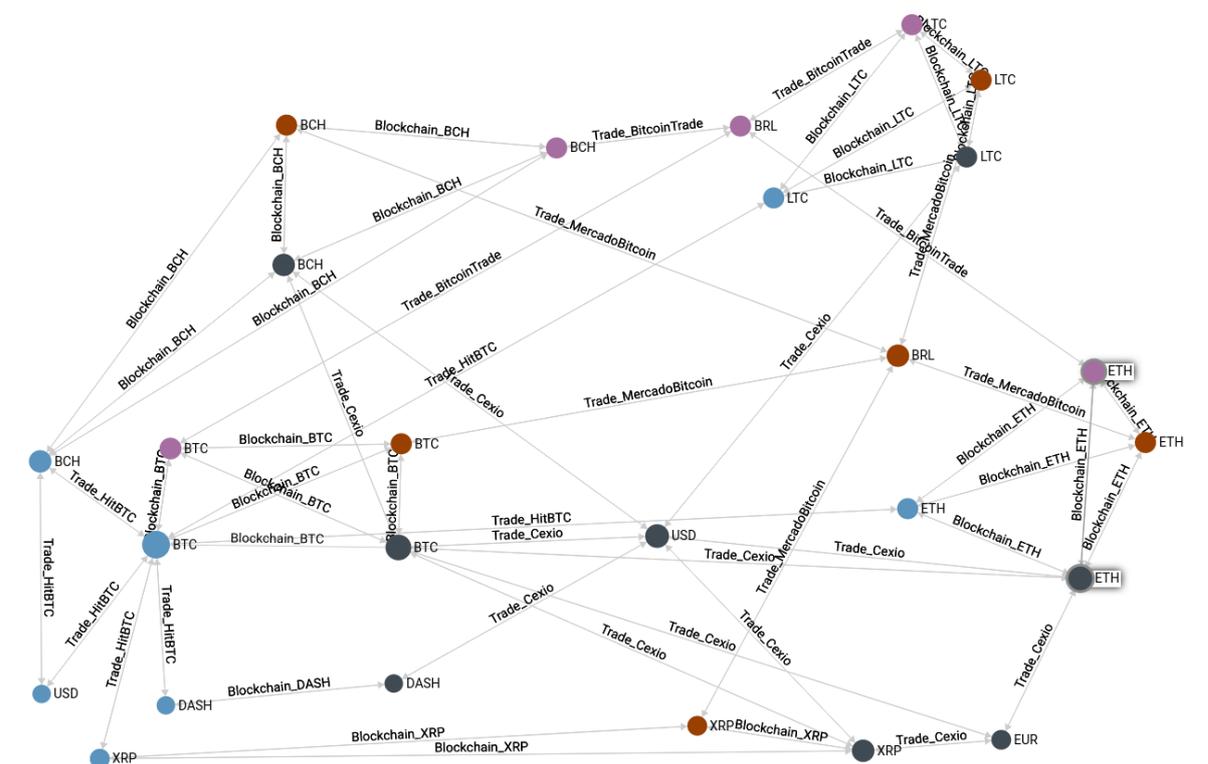
O problema de arbitragem foi estruturado em um banco de dados NoSQL onde os nós são armazenados em *collections* de nós e arestas em *collections* de arestas da seguinte forma:

- a) Cada criptomoeda de cada *exchange* foi alocada em um nó e inserida em uma *collection* de nós.
- b) Moedas Fiat de cada *exchange* também possuem seus próprios nós, alocados em *collections* diferentes.
- c) Cada nó armazena um identificador único, dados da carteira, saldo da carteira e se já foi visitado.
- d) Cada *collection* de arestas armazena arestas que conectam nós.
- e) Cada aresta representa uma transação que pode ser feita pelo *Blockchain* específico de cada moeda ou uma transação dentro de cada corretora.
- f) Cada transação representa um custo para sair de um determinado nó (moeda) para outro nó, ou seja, custos de mineração no caso do *Blockchain* ou taxas fixas e variáveis da corretora.
- g) A aresta também armazena a cotação para convergência de uma moeda em outra, por exemplo, para caminhar do nó BTC de uma corretora para o nó Real da mesma corretora multiplica-se o valor da cotação no sentido da aresta, de modo que se o caminho for inverso faz-se a operação inversa.
- h) Foram escolhidas as cinco maiores criptomoedas em capitalização já que teriam maior liquidez. Sendo as escolhidas: *Bitcoin* (BTC), *Litecoin* (LTC), *Bitcoin Cash* (BCH), *Ethereum* (ETH), *Ripple* (XRP) e *Dash* (DASH).

- i) Foram escolhidas três moedas fiat já que queria-se medir se a diferença entre cotações entre Dolar e Real ou Euro seriam refletidas nas criptomoedas. Sendo as escolhidas, portanto: *Real Brasileiro (Brazilian Real)* (BRL), *Dolar Americano (United States Dolar)* (USD), *Euro União Européia (European Union Euro)* (EUR).

Na figura 24 é possível ver todos os nós interligados por arestas.

Figura 24 – Grafo orientado para criar rotas entre moedas Fiat e Criptomoedas.



Fonte: Autor

É possível verificar na figura 24 que, apesar de, todas as criptomoedas terem arestas entre o mesmo tipo por causa do *Blockchain*, as corretoras não fazem a conversão entre todas as criptomoedas, por exemplo, todos os nós ETH estão conectados pelo *Blockchain* do ETH, mas não são todas as corretoras que convertem ETH em BTC ou em BRL.

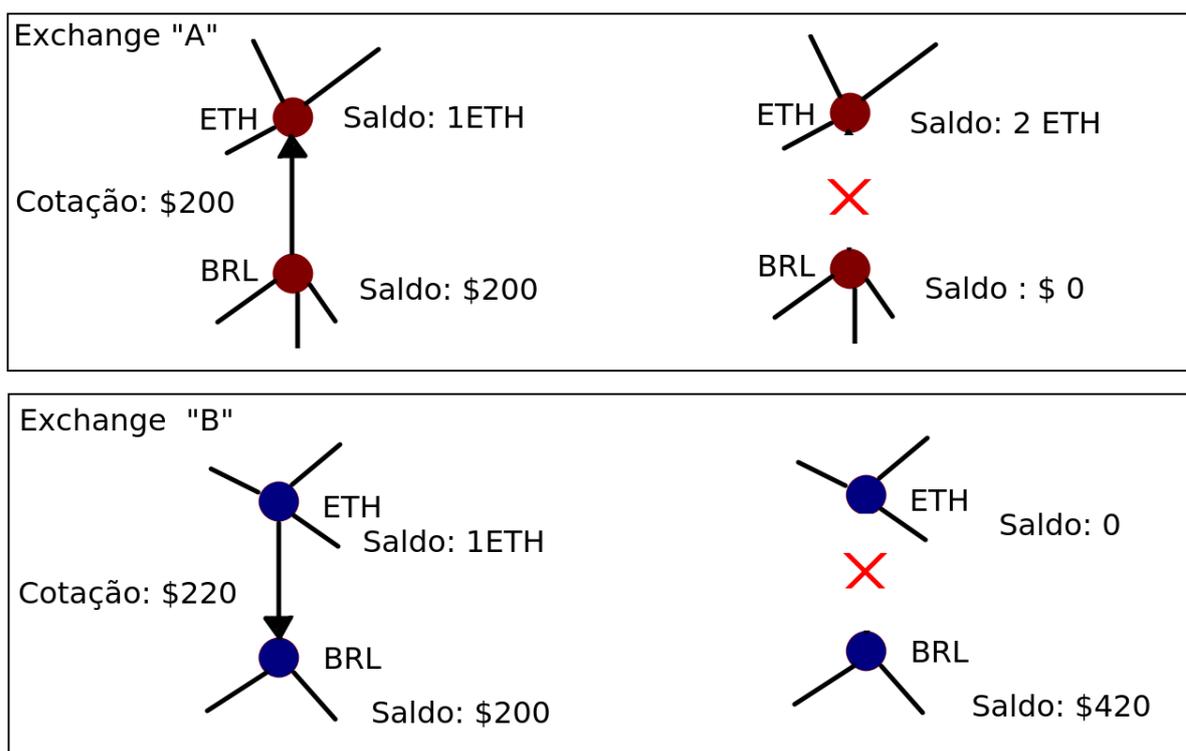
Com a estrutura definida, considere agora, que em um instante  $t$ , um *Bitcoin* (BTC) possua uma ordem de venda em aberto à R\$35.000,00 em uma corretora "A" e esteja sendo cotado a R\$34.800,00 em uma corretora "B", um arbitrador pode efetuar simultaneamente a compra em "B" e a venda em "A" lucrando com a diferença de R\$200,00, ou seja, cerca de 0,28% de lucro em apenas uma operação. Para tal é necessário que haja um saldo em carteira de \$35.000,00 BRL na corretora "B" e um saldo de 1 BTC na corretora "A" para que as ordens sejam

simultaneamente executadas. Isso faz com o saldo em carteira de um lado da arbitragem fique zerado e, se ocorrer uma nova oportunidade de arbitragem, essa será perdida por insuficiência de saldo.

Para solucionar o problema da insuficiência de saldo, foi proposta uma solução que é análoga ao encontrar um caminho entre dois vértices (ou nós) em um grafo, de modo que a soma dos pesos, no caso custos, de suas arestas constituintes seja minimizada.

A transação de arbitragem é o caminho  $p$  de um nó  $v$  por uma aresta  $a1$  até outro nó  $u$ , enquanto simultaneamente ocorre o caminho de outro nó  $d$  qualquer por outra aresta  $a2$  para outro nó  $f$ . Uma operação de transferência de valores é apresentada na figura 25. Após serem efetuadas as transações é necessário podar as arestas, para que o processo de restauração de saldo não percorra o caminho inverso da arbitragem, eliminando os lucros.

Figura 25 – Operação de Arbitragem para um par de moedas.



Fonte: Autor

Todos os saldos são inicializados com o mesmo valor sendo sua conversão feita com as cotações do início da operação do robô. Entre todas as corretoras encontra-se o maior e o menor valor de cotação para uma determinada criptomoeda.

O robô, agindo passivamente, retornou em média 8,1% ao mês de lucro no período que abrange os primeiros 6 meses no formato de juros simples, já que a restauração de saldo se dá,

### Algoritmo 6 – Algoritmo Arbitragem

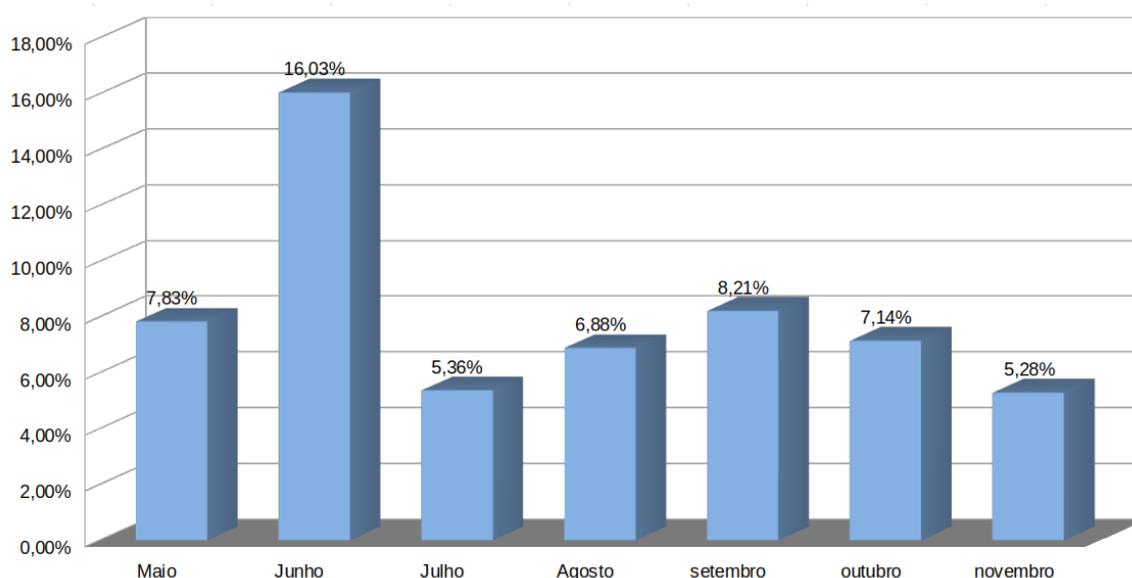
```

1 Entrada: V vertices e A Arestas
2 Inicialize saldos de todas as carteiras
3 Inicialize cotações de todos os pares de moedas
4 Inicialize taxas de corretoras e de mineração de todos os pares de moedas
5 repita
6   para cada criptomoeda faça
7     Retorne a corretora que possui a maior cotação como sendo  $h$ 
8     Retorne a corretora que possui a menor cotação como sendo  $l$ 
9     Execute ordens simultâneas de compra e venda em  $l$  e  $h$  simultaneamente
10    Efetue a poda do grafo das arestas envolvidas na arbitragem Faça operação
        de transferência pelo Blockchain da criptomoeda para restaurar seu saldo
11    Aguarde o período equivalente a 3 confirmações de rede. Usando o
        algoritmo Dijkstra, encontre o melhor caminho  $p^*$  de taxas e cotações para
        restaurar o saldo da moeda Fiat
12    Execute as operações de compra e venda selecionadas pelo melhor caminho
         $p^*$ , aguardando o período equivalente a 3 confirmações de rede para cada
        transação.
13  fim
14 até;

```

apenas, nos vértices que tiveram sua ordem executada. Se for considerado juros compostos, os ganhos atingem 6,627% ao mês. A figura 26 mostra os retornos sobre o capital investido em percentuais por mês no ano de 2019.

Figura 26 – Ganhos de juros simples sobre capital investido por mês de 2019.



Fonte: Autor

#### 7.4 Previsão de comprar em um bot voltado para arbitragem

Até a seção 7.3, o robô apenas age reativamente, ou seja, aguarda uma situação de arbitragem acontecer para, só depois, tomar alguma ação. Essa premissa funciona bem na maioria dos casos, porém, existem casos em que as ordens de compra ou de venda ficaram paradas, devido à oscilação do mercado, necessitando de auxílio humano para decidir se deveria aceitar o prejuízo da referida volatilidade.

Duas soluções são bem diretas nesse sentido. A primeira envolve o equivalente a uma ordem de *stop* no mercado financeiro tradicional. Nesse caso o quando o valor da cotação atinge um valor acima ou abaixo do desejado a operação é cancelada e absorvem-se os prejuízos. A ordem de stop no mercado de criptomoedas precisa de atraso para acontecer, já que devido a sua volatilidade, repiques de cotação são frequentes e não necessariamente representam tendências de mercado.

Visando mitigar essa situação, uma CMEAS-DG, tal qual foi usada em 21, foi treinada com 2 camadas e tendo a saída de manter posição removida. Isso é feito para prever situações de compra ou de venda antes que a oportunidade de arbitragem tenha realmente aparecido, assim, atuando ativamente e evitando que uma futura queda possa travar quaisquer operações.

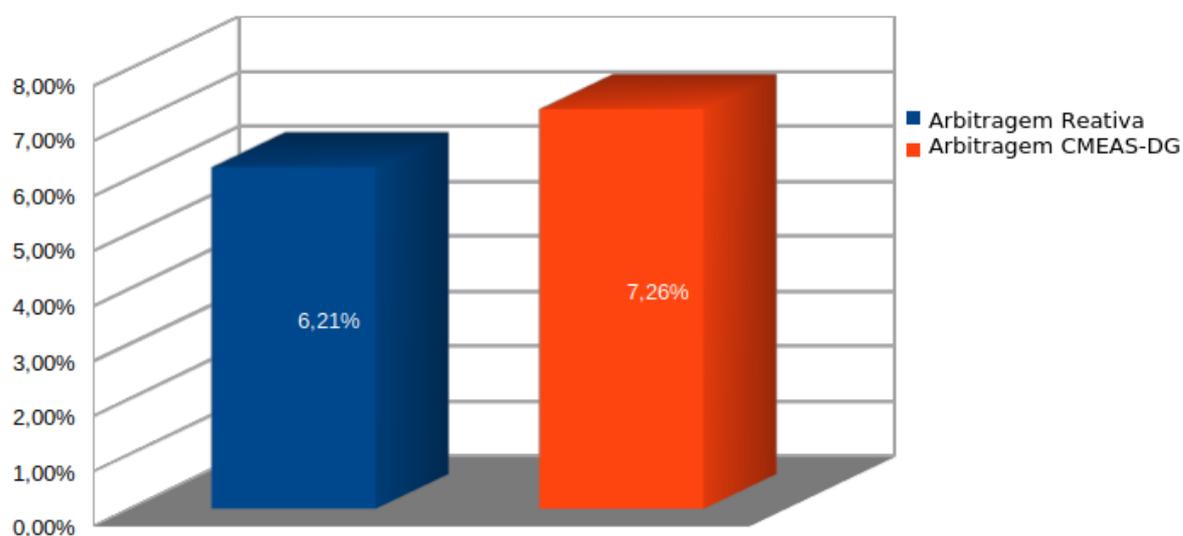
Durante um mês, todas as transações que seriam travadas por mudanças repentinas de valores e que foram evitadas pela CMEAS-DG, foram armazenadas em um arquivo de log. Ao final pôde-se comparar as duas metodologias reativa e ativa, decrementando do resultado da Arbitragem CMEAS-DG atingiu-se o valor da arbitragem simples.

A figura 27 mostra um aumento de 1,05 pontos percentuais, ou seja, cerca de 16,88% de ganho sobre a abordagem reativa durante sua operação de um mês.

#### 7.5 Discussões e Conclusões

Para efetuar esse tipo de operação é necessário que os saldos de cada moedas tenham alta disponibilidade. Apesar de ser um problema de simples solução, essas moedas estão sujeitas a variação do mercado, já que estão paradas aguardando uma nova operação. Nesse capítulo foi apresentado um estudo de caso de aplicação da CMEAS sobre um robô automatizado, desenvolvido para identificar e fazer operações de arbitragem, restaurando saldos com uma transformação do problema de menor caminho em grafos. Os riscos dessa abordagem derivam da incerteza da permanência dessas moedas no futuro, de legislações ainda pendentes,

Figura 27 – Comparação de ganhos sobre capital investido no mês de dezembro de 2019 entra a arbitragem simples e a arbitragem CMEAS-DG.



Fonte: Autor

instabilidades das *Aplicação de Interface de processo (Application Process Interface) (API)* das *Corretora de Criptomoedas (Cryptocurrency Exchange) (CE)* e o tempo de transações pelo *Blockchain*. Os resultados foram animadores, já que foi possível vislumbrar ganhos da faixa de 6%, que são muito superiores a ganhos no mercado de capitais tradicional.

## 8 Conclusão

O agrupamento de informações subjetivas conflitantes de um mercado volátil, como o mercado de criptomoedas, é uma tarefa não trivial. Para endereçar esse problema, essa tese apresentou uma técnica de acoplamento de duas redes neurais convolucionais profundas e um estudo de caso, se utilizando de uma estratégia de arbitragem, em um mercado de criptomoedas extremamente recente. Esse método pôde ser dividido em duas vertentes, sendo uma com aprendizado supervisionado e, outra, com aprendizado por reforço. Foram usadas redes profundas com pesos pré-definidos, ou seja, já treinadas *a priori* em suas respectivas áreas, sem alteração de seus hiperparâmetros, usando conexões externas às redes para conectar neurônios internos.

Esse método tem como vantagens não necessitar de conhecimento particular das áreas das redes às quais estão sendo usadas e também foi mostrado, através dos experimentos, que é possível se aproveitar de suas topologias.

Os experimentos e resultados mostram que em ambos os métodos houve ganho de rendimento com relação a redes individualmente treinadas, a um *ensemble*, uma rede estado da arte para previsão de séries temporais como a CNN-LSTM e uma estratégia conservadora de se obter lucro em mercados financeiros tradicionais. Resultados corroborados pelo teste de Wilcoxon que mostrou que o CMEAS teve um desempenho melhor que cada uma dessas estratégias.

Em adição foi possível verificar que, apesar de, o treinamento com 3 camadas apresentar um fator de lucro melhor em algumas comparações, acoplando-se as duas últimas camadas obteve-se, em média, melhores índices Sharpe. Além disso, utilizar duas camadas apresentou um tempo de treinamento menor.

### 8.1 Enquadramento da CMEAS

O algoritmo proposto para agrupar duas ou mais redes usando suas próprias conexões axônicas, se enquadra, segundo as definições do capítulo 2, no contexto de agrupamento, por se tratar de uma conjunção de redes neurais, mesmo que não seja nos mesmos moldes do agrupamento clássico. Esse algoritmo também tem um aspecto de evolução já que as conexões entre neurônios entre duas redes diferentes precisam ser criadas, além disso, é possível ver um aspecto de crescimento, não por um neurônio por vez, mas, sim, por uma segunda ou, mais sub-redes inteiras também treinadas a-priori.

## 8.2 Contribuições

- a) Neste trabalho foi proposto um algoritmo de agrupamento de redes treinada com descida de gradiente, que apresenta um melhor desempenho para o mercado de criptomoedas dadas determinadas métricas, apenas treinando as conexões inter-redes, sem necessitar aplicar todos os conceitos envolvidos no desenvolvimento das redes base.
- b) Também foi proposto um algoritmo de agrupamento de redes treinada com uso de aprendizado por reforço, que apresenta um desempenho razoável para o mercado de criptomoedas dadas determinadas métricas, apenas treinando as conexões inter-redes, sem necessitar aplicar todos os conceitos envolvidos no desenvolvimento das redes base.
- c) Finalmente, como contribuição também foi proposto um algoritmo de arbitragem passiva e um de arbitragem ativa com o uso do CMEAS.
- d) Para realização dos experimentos foi criado um banco de dados de treinamento de valores de cotação de abertura, fechamento, maior valor, menor valor, volume, capitalização e notícias sincronizadas por dia para o mercado de criptomoedas do Ethereum.
- e) Finalmente, também foi criado um banco de dados de treinamento de valores de cotação de abertura, fechamento, maior valor, menor valor e notícias sincronizadas por dia para o mercado tradicional dos Estados Unidos, fonte Reuters.

### 8.2.1 Publicações Resultantes desse Doutorado

A partir do trabalho realizado nesta tese foram publicados e submetidos diversos artigos, que foram de suma importância para o desenvolvimento dessa tese, especialmente com relação ao aprofundamento na área de aprendizado profundo, mais especificamente, no que tange ao *end-to-end learning*.

O artigo intitulado: "A Biologically Inspired Artificial Neural Network Grouping Technique Applied In The Ethereum Cryptocurrency Market" foi submetido para o periódico com fator de impacto 10.716 Journal Information Fusion 2020.

O artigo intitulado: "A Biologically Inspired Artificial Neural Network Grouping Technique Applied In The Ethereum Cryptocurrency Market" foi submetido e aceito no 6th Intl. Conference on Soft Computing e Machine Intelligence ISCFM 2019.

O artigo: "Towards Robotic Cognition Using Deep Neural Network Applied in a Goalkeeper Robot" foi publicado no Latin American Robotics Symposium (LARS) em 2017, este artigo apresenta um sistema de cognição com DNN, mais especificamente no que tange ao *end-to-end learning*.

O artigo intitulado: "Toward Robotic Cognition by Means of Decision Tree of Deep Neural Networks applied in a Humanoid Robot" foi submetido para a revista Journal of Robotics and Autonomous Systems, este artigo apresenta o sistema de cognição com árvore de decisão de DNNs.

"Toward Robotic Cognition by Means of Decision Tree of Deep Neural Networks applied in a Humanoid Robot" SILVA, ISAAC J.; VILAO, C. O.; REALI, C. ANNA; BIANCHI, R. A. C.. Journal of Control, Automation and Electrical Systems. 2020.

"Humanoid Robot Framework for Research on Cognitive Robotics" PERICO, D. H.; HOMEM, THIAGO P. D.; ALMEIDA, A. C.; SILVA, ISAAC J.; VILAO, C. O.; NICASSIO, V.; BIANCHI, R. A. C.. Journal of Control, Automation and Electrical Systems. 2018.

Evaluating The Performance Of Two Visual Descriptors Techniques For A Humanoid Robot. Communications in Computer and Information Science. VILÃO, CLAUDIO O.; SILVA, ISAAC J.; BIANCHI, REINALDO A.C.. Communications in Computer and Information Science. Springer. 2016.

Humanoid Robot Gait on Sloping Floors Using Reinforcement Learning. Communications in Computer and Information Science. SILVA, ISAAC J.; PERICO, DANILOH.; HOMEM, THIAGO P. D.; VILÃO, CLAUDIO O.; TONIDANDEL, FLAVIO; BIANCHI, REINALDO A.C.. Communications in Computer and Information Science. Springer. 2016.

Towards Robotic Cognition Using Deep Neural Network Applied in a Goalkeeper Robot. SILVA, ISAAC J.; VILÃO, CLAUDIO O.; BIANCHI, REINALDO A. C.. Latin American Robotics Symposium (LARS), Curitiba - Paraná - Brazil. 2017.

A Robot Simulator Based on the Cross Architecture for the Development of Cognitive Robotics. PERICO, DANILOH.; HOMEM, THIAGO P.D.; ALMEIDA, AISLAN C.; SILVA, ISAAC J.; VILÃO, CLAUDIO O.; FERREIRA, VINICIUS N.; BIANCHI, REINALDO A.C.. Latin American Robotics Symposium (LARS), Recife-PE - Brazil. 2016.

### 8.3 Trabalhos futuros

Um aspecto citado por Lima Filho (2013), decorre do fato de ordens de compra e de venda serem processadas em diferentes partes do cérebro e, mais, ressalta que as ordens de compra e venda acionaram diferentes circuitos neuronais quando comparados às ordens de manutenção *hold*, mesmo frente a um mercado previsível (LIMA FILHO, 2013). Isso mostra como diferentes áreas afetam setores diferenciados. Uma proposta seria usar redes treinadas para compra e venda separadamente e depois agrupá-las por meio da CMEAS.

Outra oportunidade para trabalhos futuros reside nos pesos da matriz  $W_\alpha$  após o treinamento, pois alguns deles permaneceram em zero, o que pode ser uma indicação de que esse grupo de neurônios de uma rede neural não tem relação com o outro. Portanto, os neurônios apontados por pesos não nulos podem estar relacionados a essa tarefa e podem ser marcados por proteínas artificiais e essas proteínas podem ser usadas para acoplar outras redes em outras tarefas co-relacionadas.

Uma dificuldade do uso de redes neurais, se baseia no fato de ser difícil explicar como elas estão resolvendo as tarefas, ou seja, não é possível identificar nos pesos internos da rede, onde determinada característica está sendo processada. Mas talvez seja possível identificar onde grupos de características em redes diferentes se relacionam. A representação do resultado interno é muitas vezes tão complexa que é impossível analisar, exceto os casos mais simples que normalmente não são de nenhum interesse.

Em Jaderberg et al. (2017), foi proposto um desacoplamento das camadas da rede neural através do uso do que os autores chamaram de gradientes sintéticos. A ideia é que o treinamento de redes neurais normalmente requer que os dados sejam propagados para frente, seguido de um sinal de erro sendo propagado para trás na rede, para fins de atualizações dos pesos.

Todas as camadas estão, portanto, bloqueadas, no sentido de que devem aguardar o restante da rede antes propagar os dados ou os erros. Os autores denominam as camadas chamando-as de módulos e para se efetuar um treinamento assíncrono eles propuseram uma interface de módulos desacoplados, do inglês *Decoupled Neural Interfaces - DNI*. O que o gradiente sintético faz, informalmente em resumo, é prever, criando diversas sub-redes, o gradiente verdadeiro minimizando a distância  $d(\hat{\delta}_T, \delta_T)$  tendendo ao gradiente alvo  $\delta_T$ .

Para redes neurais, o sinal de erro de retorno  $\delta$  pode assumir formas diferentes, os gradientes podem ser usados como sinal de erro para trabalhar com backpropagation, mensagens de destino como o sinal de erro para trabalhar com a propagação de alvos, do inglês *Difference Tar-*

*get Propagation* (LEE, D. et al., 2014), ou mesmo uma função valor acumulada com desconto de recompensa futura como no caso da estrutura da aprendizagem por reforço (JADERBERG et al., 2017), dessa forma, seria possível utilizar gradientes sintéticos para treinar os pesos da matriz  $w^\alpha$  isoladamente identificando a função delta responsável pela propagação do erro.

Mais um trabalho futuro seria adaptar o algoritmo CMEAS-TD para uso em uma DQN substituindo a função  $V_s$  pela função  $Q_{(s,a)}$  no treinamento do CMEAS.

## REFERÊNCIAS

- ABRAHAM, Jethin et al. Cryptocurrency price prediction using tweet volumes and sentiment analysis. **SMU Data Science Review**, v. 1, n. 3, p. 1, 2018.
- ACKERT, Lucy F; CHURCH, Bryan K; DEAVES, Richard. Emotion and financial markets. **Economic Review-Federal Reserve Bank of Atlanta**, Federal Reserve Bank of Atlanta, v. 88, n. 2, p. 33, 2003.
- ADDO, Peter Martey; GUEGAN, Dominique; HASSANI, Bertrand. Credit risk analysis using machine and deep learning models. **Risks**, Multidisciplinary Digital Publishing Institute, v. 6, n. 2, p. 38, 2018.
- AGGARWAL, Charu C. **Data classification: algorithms and applications**. [S.l.]: CRC Press, 2014.
- AKABANE, G.K. **Gestão Estratégica das Tecnologias Cognitivas**. [S.l.]: Saraiva Educação S.A., 2012. ISBN 9788536527925. Disponível em: <<https://books.google.com.br/books?id=Z4uwDwAAQBAJ>>.
- AKINS, Benjamin W; CHAPMAN, Jennifer L; GORDON, Jason M. A whole new world: Income tax considerations of the Bitcoin economy. **Pitt. Tax Rev.**, HeinOnline, v. 12, p. 25, 2014.
- ALPAYDIN, Ethem. Multiple networks for function learning. In: IEEE. NEURAL Networks, 1993., IEEE International Conference on. [S.l.: s.n.], 1993. p. 9–14.
- ALPAYDM, Ethem. Grow-and-learn: An incremental method for category learning. In: SPRINGER. INTERNATIONAL Neural Network Conference. [S.l.: s.n.], 1990. p. 761–764.
- ALVAREZ-BUYLLA, Arturo; GARCIA-VERDUGO, Jose Manuel. Neurogenesis in adult subventricular zone. **Journal of Neuroscience**, Soc Neuroscience, v. 22, n. 3, p. 629–634, 2002.
- ANDREWS, Robert; DIEDERICH, Joachim; TICKLE, Alan B. Survey and critique of techniques for extracting rules from trained artificial neural networks. **Knowledge-based systems**, Elsevier, v. 8, n. 6, p. 373–389, 1995.
- ANTSHARES digital assets for everyone. [S.l.: s.n.], 2015. Disponível em: <https://www.antshares.org>. Acesso em: 01 Set. 2016.
- APPEL, G. **Technical Analysis: Power Tools for Active Investors**. [S.l.]: Financial Times/Prentice Hall, 2005. (Financial Times Prentice Hall books). ISBN 9780131479029. Disponível em: <<https://books.google.com.br/books?id=RFYIAAAACAAJ>>.
- ARÉVALO, Andrés et al. Algorithmic Trading Using Deep Neural Networks on High Frequency Data. In: SPRINGER. WORKSHOP on Engineering Applications. [S.l.: s.n.], 2017. p. 144–155.

ATSALAKIS, George S; PROTOPAPADAKIS, Eftychios E; VALAVANIS, Kimon P. Stock trend forecasting in turbulent market periods using neuro-fuzzy systems. **Operational Research**, Springer, v. 16, n. 2, p. 245–269, 2016.

ATSALAKIS, George S; VALAVANIS, Kimon P. Surveying stock market forecasting techniques–Part II: Soft computing methods. **Expert Systems with Applications**, Elsevier, v. 36, n. 3, p. 5932–5941, 2009.

BAHRAMMIRZAEI, Arash. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. **Neural Computing and Applications**, Springer, v. 19, n. 8, p. 1165–1195, 2010.

BALCILAR, Mehmet et al. Can volume predict Bitcoin returns and volatility? A quantiles-based approach. **Economic Modelling**, Elsevier, v. 64, p. 74–81, 2017.

BARTZ, Christian et al. Language identification using deep convolutional recurrent neural networks. In: SPRINGER. INTERNATIONAL Conference on Neural Information Processing. [S.l.: s.n.], 2017. p. 880–889.

BATISTA, Gustavo EAPA; PRATI, Ronaldo C; MONARD, Maria Carolina. A study of the behavior of several methods for balancing machine learning training data. **ACM Sigkdd Explorations Newsletter**, ACM, v. 6, n. 1, p. 20–29, 2004.

BAUER, Eric; KOHAVI, Ron. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. **Machine learning**, Springer, v. 36, n. 1, p. 105–139, 1999.

BAUR, Dirk G; DIMPFL, Thomas. Asymmetric volatility in cryptocurrencies. **Economics Letters**, Elsevier, v. 173, p. 148–151, 2018.

BELLMAN, RE et al. Eugene L. Lawler, Combinatorial optimization: networks and matroids. **Bulletin of the American Mathematical Society**, American Mathematical Society, v. 84, n. 3, p. 461–463, 1978.

BENEKI, Christina et al. Investigating volatility transmission and hedging properties between Bitcoin and Ethereum. **Research in International Business and Finance**, Elsevier, v. 48, p. 219–227, 2019.

BHATHEJA, Kanav; FIELD, Jeffrey. Schwann cells: origins and role in axonal maintenance and regeneration. **The international journal of biochemistry & cell biology**, Elsevier, v. 38, n. 12, p. 1995–1999, 2006.

BIBI, Shaista; HUSSAIN, Shahid; FAISAL, Muhammad Imran. Public Perception Based Recommendation System for Cryptocurrency. In: IEEE. 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST). [S.l.: s.n.], 2019. p. 661–665.

BISHOP, Christopher M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738.

BITSHARES - your share in the decentralized exchange. [S.l.: s.n.], 2015. Disponível em: <https://bitshares.org/>. Acesso em: 20 Jun. 2015.

BLUM, Avrim; MITCHELL, Tom. Combining labeled and unlabeled data with co-training. In: ACM. PROCEEDINGS of the eleventh annual conference on Computational learning theory. [S.l.: s.n.], 1998. p. 92–100.

BÖHME, Rainer et al. Bitcoin: Economics, technology, and governance. **Journal of Economic Perspectives**, v. 29, n. 2, p. 213–38, 2015.

BOLLEN, Johan; MAO, Huina; ZENG, Xiaojun. Twitter mood predicts the stock market. **Journal of computational science**, Elsevier, v. 2, n. 1, p. 1–8, 2011.

BOROVYKH, Anastasia; BOHTE, Sander; OOSTERLEE, Cornelis W. Conditional Time Series Forecasting with Convolutional Neural Networks. **arXiv preprint arXiv:1703.04691**, 2017.

BORTOLI FILHO, Mario Antonio. **Indicadores Baseados em Diferenças de Primeira Ordem para Orientar Robôs no Mercado FOREX**. 2016. Tese (Doutorado).

BREIMAN, L. et al. **Classification and Regression Trees**. [S.l.]: Taylor & Francis, 1984. (The Wadsworth and Brooks-Cole statistics-probability series). ISBN 9780412048418. Disponível em: <<https://books.google.com.br/books?id=JwQx-WOmSyQC>>.

BREIMAN, Leo. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

\_\_\_\_\_. Stacked regressions. **Machine learning**, Springer, v. 24, n. 1, p. 49–64, 1996.

BROMLEY, Jane et al. Signature verification using a "siamese" time delay neural network. In: ADVANCES in Neural Information Processing Systems. [S.l.: s.n.], 1994. p. 737–744.

BUTERIN, Vitalik et al. A next-generation smart contract and decentralized application platform. **white paper**, v. 3, n. 37, 2014.

CACHIN, Christian et al. Architecture of the hyperledger blockchain fabric. In: WORKSHOP on distributed cryptocurrencies and consensus ledgers. [S.l.: s.n.], 2016. v. 310, p. 4.

CARMO ROQUE, Reginaldo do; MELLO, Flávio Luís de. Estudo sobre a empregabilidade da previsão do índice BOVESPA usando Redes Neurais Artificiais, 2009.

CASTRO, Cristiano Leite de; PÁDUA BRAGA, Antônio de; ANDRADE, Alessandro Vivas. Aplicaç ao de um Modelo Ensemble de Redes Neurais Artificiais para Previsao de Séries Temporais nao Estacionárias, 2005.

CASTRO, Fernando de. Chemotropic molecules: guides for axonal pathfinding and cell migration during CNS development. **Physiology**, Am Physiological Soc, v. 18, n. 3, p. 130–136, 2003.

CASTRO, Miguel; LISKOV, Barbara. Practical Byzantine fault tolerance and proactive recovery. **ACM Transactions on Computer Systems (TOCS)**, ACM New York, NY, USA, v. 20, n. 4, p. 398–461, 2002.

CASTRO, Miguel; LISKOV, Barbara et al. Practical Byzantine fault tolerance. In: 1999. OSDI. [S.l.: s.n.], 1999. v. 99, p. 173–186.

CAVALCANTE, Rodolfo C et al. Computational intelligence and financial markets: A survey and future directions. **Expert Systems with Applications**, Elsevier, v. 55, p. 194–211, 2016.

CHANAS-SACRE, Grazyna et al. Radial glia phenotype: origin, regulation, and transdifferentiation. **Journal of neuroscience research**, Wiley Online Library, v. 61, n. 4, p. 357–363, 2000.

CHAWLA, Nitesh V et al. SMOTE: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

CHAWLA, Nitesh et al. SMOTEBoost: Improving prediction of the minority class in boosting. **Knowledge Discovery in Databases: PKDD 2003**, Springer, p. 107–119, 2003.

CHEN, George H.; NIKOLOV, Stanislav; SHAH, Devavrat. **A Latent Source Model for Nonparametric Time Series Classification**. [S.l.: s.n.], 2013. arXiv: 1302.3639 [stat.ML].

CHEN, La et al. Statistical study of biomechanics of living brain cells during growth and maturation on artificial substrates. **Biomaterials**, Elsevier, v. 106, p. 240–249, 2016.

CHEN, Sheng; COWAN, Colin FN; GRANT, Peter M. Orthogonal least squares learning algorithm for radial basis function networks. **Neural Networks, IEEE Transactions on, IEEE**, v. 2, n. 2, p. 302–309, 1991.

CHEN, Yuehui; YANG, Bo; ABRAHAM, Ajith. Flexible neural trees ensemble for stock index modeling. **Neurocomputing**, Elsevier, v. 70, n. 4, p. 697–703, 2007.

CHINA BHANJA, Chuya; LASKAR, Mohammad A; LASKAR, Rabul H. Cascade convolutional neural network-long short-term memory recurrent neural networks for automatic tonal and nontonal preclassification-based Indian language identification. **Expert Systems**, Wiley Online Library, e12544, 2020.

CHOLLET, François et al. **Keras**. [S.l.: s.n.], 2015. <https://keras.io>.

CHOPRA, Sumit; HADSELL, Raia; LECUN, Yann. Learning a similarity metric discriminatively, with application to face verification. In: IEEE. **COMPUTER Vision and Pattern Recognition, 2005. CVPR 2005**. IEEE Computer Society Conference on. [S.l.: s.n.], 2005. v. 1, p. 539–546.

CHRISTOUDIAS, C; URTASUN, Raquel; DARRELL, Trevor. Multi-view learning in the presence of view disagreement. **arXiv preprint arXiv:1206.3242**, 2012.

CHUEN, David Lee Kuo. **Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data**. [S.l.]: Academic Press, 2015.

CIRESAN, Dan; MEIER, Ueli; SCHMIDHUBER, Jürgen. Multi-column deep neural networks for image classification. In: IEEE. **COMPUTER Vision and Pattern Recognition (CVPR)**, 2012 IEEE Conference on. [S.l.: s.n.], 2012. p. 3642–3649.

CIREŞAN, Dan et al. Multi-column deep neural network for traffic sign classification. **Neural networks**, Elsevier, v. 32, p. 333–338, 2012.

CLARKE, Bertrand. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. **Journal of Machine Learning Research**, v. 4, Oct, p. 683–712, 2003.

COINMARKETCAP. [S.l.: s.n.], 2019. Disponível em: <https://coinmarketcap.com/currencies/ethereum/>. Acesso em: 03 Set. 2019.

COLIANNI, Stuart; ROSALES, Stephanie; SIGNOROTTI, Michael. Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. **CS229 Project**, p. 1–5, 2015.

CORBET, Shaen et al. KODAKCoin: a blockchain revolution or exploiting a potential cryptocurrency bubble? **Applied Economics Letters**, Taylor & Francis, v. 27, n. 7, p. 518–524, 2020.

COVER, T.; HART, P. Nearest Neighbor Pattern Classification. **IEEE Trans. Inf. Theor.**, IEEE Press, Piscataway, NJ, USA, v. 13, n. 1, p. 21–27, set. 2006. ISSN 0018-9448. DOI: 10.1109/TIT.1967.1053964. Disponível em: <<http://dx.doi.org/10.1109/TIT.1967.1053964>>.

CRISTIANINI, Nello; SHAWE-TAYLOR, John. **An introduction to support vector machines and other kernel-based learning methods**. [S.l.]: Cambridge university press, 2000.

CUI, Zhicheng; CHEN, Wenlin; CHEN, Yixin. Multi-scale convolutional neural networks for time series classification. **arXiv preprint arXiv:1603.06995**, 2016.

DANTAS, Tiago J. et al. Emerging roles for motor proteins in progenitor cell behavior and neuronal migration during brain development. **Cytoskeleton**, v. 73, n. 10, p. 566–576, 2016. ISSN 1949-3592. DOI: 10.1002/cm.21293. Disponível em: <<http://dx.doi.org/10.1002/cm.21293>>.

DELFIN-VIDAL, Rafael; ROMERO-MELÉNDEZ, Guillermo. The fractal nature of bitcoin: Evidence from wavelet power spectra. In: **TRENDS in mathematical economics**. [S.l.]: Springer, 2016. p. 73–98.

DEMPSTER, A. P. Upper and Lower Probabilities Induced by a Multivalued Mapping. **Ann. Math. Statist.**, The Institute of Mathematical Statistics, v. 38, n. 2, p. 325–339, abr. 1967. DOI: 10.1214/aoms/1177698950. Disponível em: <<https://doi.org/10.1214/aoms/1177698950>>.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. **JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B**, v. 39, n. 1, p. 1–38, 1977.

DEMUTH, Howard B et al. **Neural network design**. [S.l.]: Martin Hagan, 2014.

DENG, Li; YU, Dong et al. Deep learning: methods and applications. **Foundations and Trends® in Signal Processing**, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014.

DENG, Y. et al. A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification. **IEEE Transactions on Fuzzy Systems**, v. 25, n. 4, p. 1006–1012, ago. 2017. ISSN 1063-6706. DOI: 10.1109/TFUZZ.2016.2574915.

DENG, Yue et al. Deep direct reinforcement learning for financial signal representation and trading. **IEEE transactions on neural networks and learning systems**, IEEE, v. 28, n. 3, p. 653–664, 2017.

DERRICK, Ben; WHITE, Paul. Comparing two samples from an individual Likert question. **International Journal of Mathematics and Statistics**, Ceser, v. 18, n. 3, 2017.

DICKSON, Barry J. Molecular mechanisms of axon guidance. **Science**, American Association for the Advancement of Science, v. 298, n. 5600, p. 1959–1964, 2002.

DIESTEL, Reinhard. The basics. In: **GRAPH Theory**. [S.l.]: Springer, 2017. p. 1–34.

DIETTERICH, Thomas G et al. Ensemble methods in machine learning. **Multiple classifier systems**, Springer, v. 1857, p. 1–15, 2000.

DIJKSTRA, Edsger W et al. A note on two problems in connexion with graphs. **Numerische mathematik**, v. 1, n. 1, p. 269–271, 1959.

DING, Xiao et al. Deep Learning for Event-Driven Stock Prediction. In: **IJCAI**. [S.l.: s.n.], 2015. p. 2327–2333.

\_\_\_\_\_. Using Structured Events to Predict Stock Price Movement: An Empirical Investigation. In: **EMNLP**. [S.l.: s.n.], 2014. p. 1415–1425.

DOMINGOS, Pedro. A unified bias-variance decomposition. In: **PROCEEDINGS of 17th International Conference on Machine Learning**. [S.l.: s.n.], 2000. p. 231–238.

\_\_\_\_\_. Metacost: A general method for making classifiers cost-sensitive. In: **ACM. PROCEEDINGS of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 1999. p. 155–164.

DONAHUE, Jeff et al. **Long-term Recurrent Convolutional Networks for Visual Recognition and Description**. [S.l.: s.n.], 2014. arXiv: 1411.4389 [cs.CV].

DOWNES, John; GOODMAN, Jordan Elliot. **Dicionário de Termos Financeiros e de Investimento**. 3ª edição. [S.l.]: Editora Nobel, 1993. p. 572.

DU, Xin; ZHAI, Jinjian; LV, Koupin. Algorithm Trading using Q-Learning and Recurrent Reinforcement Learning. **positions**, v. 1, p. 1, 2016.

DULĂU, Tudor-Mircea; DULĂU, Mircea. Cryptocurrency–Sentiment Analysis in Social Media. **Acta Marisiensis. Seria Technologica**, Sciendo, v. 16, n. 2, p. 1–6, 2019.

DUNNE, Mark. **Stock Market Prediction**. [S.l.], 2015.

DŽEROSKI, Saso; ŽENKO, Bernard. Is combining classifiers with stacking better than selecting the best one? **Machine learning**, Springer, v. 54, n. 3, p. 255–273, 2004.

EFRON, Bradley. Bootstrap Methods: Another Look at the Jackknife. In: **Breakthroughs in Statistics: Methodology and Distribution**. Edição: Samuel Kotz e Norman L. Johnson. New York, NY: Springer New York, 1992. p. 569–593. ISBN 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9\_41. Disponível em: <[https://doi.org/10.1007/978-1-4612-4380-9\\_41](https://doi.org/10.1007/978-1-4612-4380-9_41)>.

EFRON, Bradley; TIBSHIRANI, Robert J. **An introduction to the bootstrap**. [S.l.]: CRC press, 1994.

EMERSON, Peter. The original Borda count and partial voting. **Social Choice and Welfare**, v. 40, n. 2, p. 353–358, fev. 2013. ISSN 1432-217X. DOI: 10.1007/s00355-011-0603-9. Disponível em: <<https://doi.org/10.1007/s00355-011-0603-9>>.

EULER, Leonhard. Solutio problematis ad geometriam situs pertinentis. **Commentarii academiae scientiarum Petropolitanae**, p. 128–140, 1741.

FAMA, Eugene F. The Behavior of Stock-Market Prices. **The Journal of Business**, University of Chicago Press, v. 38, n. 1, p. 34–105, 1965. ISSN 00219398, 15375374. Disponível em: <<http://www.jstor.org/stable/2350752>>.

FAN, Wei et al. AdaCost: misclassification cost-sensitive boosting. In: ICML. [S.l.: s.n.], 1999. v. 99, p. 97–105.

FIX, James D. **Neuroanatomy**. [S.l.]: Lippincott Williams & Wilkins, 2008.

FOGEL, David B; FOGEL, Lawrence J; PORTO, VW. Evolving neural networks. **Biological cybernetics**, Springer, v. 63, n. 6, p. 487–493, 1990.

FOROGLU, George; TSILIDOU, Anna-Lali. Further applications of the blockchain. In: 12TH student conference on managerial science and technology. [S.l.: s.n.], 2015.

FRANZE, Kristian et al. Neurite branch retraction is caused by a threshold-dependent mechanical impact. **Biophysical journal**, Elsevier, v. 97, n. 7, p. 1883–1890, 2009.

FRIEDMAN, Jerome H; HALL, Peter. On bagging and nonlinear estimation. **Journal of statistical planning and inference**, Elsevier, v. 137, n. 3, p. 669–683, 2007.

FRIEDMAN, Jerome H; POPESCU, Bogdan E. Predictive learning via rule ensembles. **The Annals of Applied Statistics**, JSTOR, p. 916–954, 2008.

FRIEDMAN, Jerome H; POPESCU, Bogdan E et al. Importance sampled learning ensembles. **Journal of Machine Learning Research**, v. 94305, 2003.

FRIEDMAN, Nir; GEIGER, Dan; GOLDSZMIDT, Moises. Bayesian Network Classifiers. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, n. 2-3, p. 131–163, nov. 1997. ISSN 0885-6125. DOI: 10.1023/A:1007465528199. Disponível em: <<https://doi.org/10.1023/A:1007465528199>>.

FUMERA, Giorgio; ROLI, Fabio. A theoretical and experimental analysis of linear combiners for multiple classifier systems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 27, n. 6, p. 942–956, 2005.

GAO, Jing; FAN, Wei; HAN, Jiawei. On appropriate assumptions to mine data streams: Analysis and practice. In: IEEE. DATA Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. [S.l.: s.n.], 2007. p. 143–152.

GAO, Jing et al. A general framework for mining concept-drifting data streams with skewed distributions. In: SIAM. PROCEEDINGS of the 2007 SIAM International Conference on Data Mining. [S.l.: s.n.], 2007. p. 3–14.

GAO, Xiu; CHAN, Laiwan. An algorithm for trading and portfolio management using Q-learning and sharpe ratio maximization. In: PROCEEDINGS of the international conference on neural information processing. [S.l.: s.n.], 2000. p. 832–837.

GEORGOULA, Ifigeneia et al. Using time-series and sentiment analysis to detect the determinants of bitcoin prices. **Available at SSRN 2607167**, 2015.

GHASHGHAEI, H Troy; LAI, Cary; ANTON, ES. Neuronal migration in the adult brain: are we there yet? **Nature reviews. Neuroscience**, Nature Publishing Group, v. 8, n. 2, p. 141, 2007.

GHOSH, Joydeep; ACHARYA, Ayan. Cluster ensembles. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 1, n. 4, p. 305–315, 2011.

GIACOMEL, Felipe dos Santos. Um método algorítmico para operações na bolsa de valores baseado em ensembles de redes neurais para modelar e prever os movimentos dos mercados de ações, 2016.

GIACOMEL, Felipe; PEREIRA, Adriano CM; GALANTE, Renata. Improving financial time series prediction through output classification by a neural network ensemble. In: SPRINGER. INTERNATIONAL Conference on Database and Expert Systems Applications. [S.l.: s.n.], 2015. p. 331–338.

GILES, C Lee; LAWRENCE, Steve; TSOI, Ah Chung. Noisy time series prediction using recurrent neural networks and grammatical inference. **Machine learning**, Springer, v. 44, n. 1-2, p. 161–183, 2001.

GOMEZ, Faustino John. **Robust non-linear control through neuroevolution**. 2003. Tese (Doutorado).

GOMEZ, Faustino; SCHMIDHUBER, Jürgen; MIIKKULAINEN, Risto. Accelerated neural evolution through cooperatively coevolved synapses. **Journal of Machine Learning Research**, v. 9, May, p. 937–965, 2008.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. [S.l.]: MIT press, 2016.

GOULD, Elizabeth et al. Neurogenesis in the neocortex of adult primates. **Science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 548–552, 1999.

GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. Speech recognition with deep recurrent neural networks. In: IEEE. ACOUSTICS, speech and signal processing (icassp), 2013 ieee international conference on. [S.l.: s.n.], 2013. p. 6645–6649.

GREAVES, Alex; AU, Benjamin. Using the bitcoin transaction graph to predict the price of bitcoin. **No Data**, 2015.

GRINBERG, Reuben. Bitcoin: An innovative alternative digital currency. **Hastings Science Technologies LJ**, HeinOnline, v. 4, p. 159, 2012.

GROTH, Sven S; MUNTERMANN, Jan. An intraday market risk management approach based on textual analysis. **Decision Support Systems**, Elsevier, v. 50, n. 4, p. 680–691, 2011.

GU, Bin et al. Identifying information in stock message boards and its implications for stock market efficiency. In: WORKSHOP on Information Systems and Economics, Los Angeles, CA. [S.l.: s.n.], 2006.

GU, Shixiang et al. Continuous deep q-learning with model-based acceleration. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2016. p. 2829–2838.

HANSEN, Lars Kai; SALAMON, Peter. Neural network ensembles. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 12, n. 10, p. 993–1001, 1990.

HASSAN, Amira Kamil Ibrahim; ABRAHAM, Ajith. Modeling consumer loan default prediction using ensemble neural networks. In: IEEE. COMPUTING, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on. [S.l.: s.n.], 2013. p. 719–724.

HASSELT, Hado V. Double Q-learning. In: LAFFERTY, J. D. et al. (Ed.). **Advances in Neural Information Processing Systems 23**. [S.l.]: Curran Associates, Inc., 2010. p. 2613–2621. Disponível em: <<http://papers.nips.cc/paper/3964-double-q-learning.pdf>>.

HATTEN, Mary E. New directions in neuronal migration. **Science**, American Association for the Advancement of Science, v. 297, n. 5587, p. 1660–1663, 2002.

HAYKIN, Simon; NETWORK, Neural. A comprehensive foundation. **Neural Networks**, v. 2, n. 2004, 1999.

HEATON, JB; POLSON, NG; WITTE, JH. Deep learning in finance. **arXiv preprint arXiv:1602.06561**, 2016.

HEBB, Donald O. **The organization of behavior**. [S.l.]: New York: Wiley, 1949.

HELLSTRÖM, Thomas. **A random walk through the stock market**. 1998. Tese (Doutorado).

HERRUP, Karl; YANG, Yan. Cell cycle regulation in the postmitotic neuron: oxymoron or new biology? **Nature reviews. Neuroscience**, Nature Publishing Group, v. 8, n. 5, p. 368, 2007.

HIDO, Shohei; KASHIMA, Hisashi; TAKAHASHI, Yutaka. Roughly balanced bagging for imbalanced data. **Statistical Analysis and Data Mining: The ASA Data Science Journal**, Wiley Online Library, v. 2, n. 5-6, p. 412–426, 2009.

HILL, Sean L. et al. Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits. **Proceedings of the National Academy of Sciences**, v. 109, n. 42, e2885–e2894, 2012. DOI: 10.1073/pnas.1202128109. eprint: <http://www.pnas.org/content/109/42/E2885.full.pdf>. Disponível em: <<http://www.pnas.org/content/109/42/E2885.abstract>>.

HINTON, Geoffrey E. Deep belief networks. **Scholarpedia**, v. 4, n. 5, p. 5947, 2009.

HINTON, Geoffrey E; OSINDERO, Simon; TEH, Yee-Whye. A fast learning algorithm for deep belief nets. **Neural computation**, MIT Press, v. 18, n. 7, p. 1527–1554, 2006.

HINTON, Geoffrey E; SALAKHUTDINOV, Ruslan R. Reducing the dimensionality of data with neural networks. **science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOPFIELD, John J; TANK, David W. Neural computation of decisions in optimization problems. **Biological cybernetics**, Springer, v. 52, n. 3, p. 141–152, 1985.

HOSMER JR, David W; LEMESHOW, Stanley; STURDIVANT, Rodney X. **Applied logistic regression**. [S.l.]: John Wiley & Sons, 2013. v. 398.

HUANG, Yea S; SUEN, Ching Y. The behavior-knowledge space method for combination of multiple classifiers. In: INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE). IEEE computer society conference on computer vision and pattern recognition. [S.l.: s.n.], 1993. p. 347–347.

INAMDAR, Abid et al. Predicting Cryptocurrency Value using Sentiment Analysis. In: IEEE. 2019 International Conference on Intelligent Computing and Control Systems (ICCS). [S.l.: s.n.], 2019. p. 932–934.

INTRODUCING casper the friendly ghost. [S.l.: s.n.], 2015. Disponível em: <https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/>. Acesso em: 01 Ago. 2015.

JACOBS, Robert A et al. Adaptive mixtures of local experts. **Neural computation**, MIT Press, v. 3, n. 1, p. 79–87, 1991.

JACOBS, Robert; JORDAN, Michael et al. Learning piecewise control strategies in a modular neural network architecture. **Systems, Man and Cybernetics, IEEE Transactions on**, IEEE, v. 23, n. 2, p. 337–345, 1993.

JADERBERG, Max et al. Decoupled neural interfaces using synthetic gradients. **arXiv preprint arXiv:1608.05343**, 2017.

JIMÉNEZ, Daniel. Dynamically weighted ensemble neural networks for classification. In: IEEE. NEURAL Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on. [S.l.: s.n.], 1998. v. 1, p. 753–756.

JORDAN, Michael I; JACOBS, Robert A. Hierarchical mixtures of experts and the EM algorithm. **Neural computation**, MIT Press, v. 6, n. 2, p. 181–214, 1994.

JÚNIOR, Arnaldo Uber. **iEnsemble: Proposta de Framework para Comitê de Máquinas baseado em Sistemas MultiAgentes**. 2019. Tese (Doutorado) – Universidade Federal de Santa Catarina.

KAKUSHADZE, Zura; YU, Willie. Altcoin-Bitcoin Arbitrage. **arXiv preprint arXiv:1903.06033**, 2019.

KALCHBRENNER, Nal; GREFFENSTETTE, Edward; BLUNSON, Phil. A Convolutional Neural Network for Modelling Sentences. **CoRR**, abs/1404.2188, 2014. Disponível em: <<http://arxiv.org/abs/1404.2188>>.

KANDEL, Eric R et al. **Principles of neural science**. [S.l.]: McGraw-hill New York, 2000. v. 4.

KARA, Yakup; BOYACIOGLU, Melek Acar; BAYKAN, Ömer Kaan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. **Expert systems with Applications**, Elsevier, v. 38, n. 5, p. 5311–5319, 2011.

KARALEVICIUS, Vytautas; DEGRANDE, Niels; DE WEERDT, Jochen. Using sentiment analysis to predict interday Bitcoin price movements. **The Journal of Risk Finance**, Emerald Publishing Limited, 2018.

KEMPTER, Richard; GERSTNER, Wulfram; VAN HEMMEN, J Leo. Spike-based compared to rate-based Hebbian learning. In: *ADVANCES in neural information processing systems*. [S.l.: s.n.], 1999. p. 125–131.

KERAS CNN-LSTM. [S.l.: s.n.], 2019. Disponível em: [https://github.com/keras-team/keras/blob/master/examples/imdb\\_lstm.py](https://github.com/keras-team/keras/blob/master/examples/imdb_lstm.py). Acesso em: 03 fev. 2019.

KIM, Yoon. Convolutional Neural Networks for Sentence Classification. **CoRR**, abs/1408.5882, 2014. Disponível em: <<http://arxiv.org/abs/1408.5882>>.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KITTLER, Josef et al. On combining classifiers. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 20, n. 3, p. 226–239, 1998.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA, 2. IJCAI. [S.l.: s.n.], 1995. v. 14, p. 1137–1145.

KOLTER, Jeremy Z; MALOOF, Marcus A. Using additive expert ensembles to cope with concept drift. In: ACM. PROCEEDINGS of the 22nd international conference on Machine learning. [S.l.: s.n.], 2005. p. 449–456.

KÖNIG, Peter; ENGEL, Andreas K; SINGER, Wolf. Integrator or coincidence detector? The role of the cortical neuron revisited. **Trends in neurosciences**, Elsevier, v. 19, n. 4, p. 130–137, 1996.

KORDÍK, Pavel et al. Meta-learning approach to neural network optimization. **Neural Networks**, v. 23, n. 4, p. 568–582, 2010. The 18th International Conference on Artificial Neural Networks, ICANN 2008. ISSN 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2010.02.003>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608010000456>>.

KOREN, Yehuda; BELL, Robert; VOLINSKY, Chris. Matrix factorization techniques for recommender systems. **Computer**, IEEE, v. 42, n. 8, 2009.

KOSBA, Ahmed et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE. 2016 IEEE symposium on security and privacy (SP). [S.l.: s.n.], 2016. p. 839–858.

KOSER, David E et al. Mechanosensing is critical for axon growth in the developing brain. **Nature neuroscience**, Europe PMC Funders, v. 19, n. 12, p. 1592, 2016.

KOURENTZES, Nikolaos; BARROW, Devon K; CRONE, Sven F. Neural network ensemble operators for time series forecasting. **Expert Systems with Applications**, Elsevier, v. 41, n. 9, p. 4235–4244, 2014.

KOVÁCS, Zsolt L. **Redes neurais artificiais**. [S.l.]: Editora Livraria da Física, 2002.

KRAAIJEVELD, Olivier; SMEDT, JD. The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. **J. Int. Financ. Mark. Inst. Money**, 2020.

KRAUSS, Christopher; DO, Xuan Anh; HUCK, Nicolas. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. **European Journal of Operational Research**, Elsevier, v. 259, n. 2, p. 689–702, 2017.

KRISTOUFEK, Ladislav. What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. **PloS one**, Public Library of Science, v. 10, n. 4, 2015.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: **ADVANCES in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

KUNCHEVA, Ludmila I. **Combining pattern classifiers: methods and algorithms**. [S.l.]: John Wiley & Sons, 2004.

\_\_\_\_\_. **Diversity in multiple classifier systems**. [S.l.]: Elsevier, 2005.

KUNCHEVA, Ludmila I; BEZDEK, James C; DUIN, Robert PW. Decision templates for multiple classifier fusion: an experimental comparison. **Pattern recognition**, Elsevier, v. 34, n. 2, p. 299–314, 2001.

KUNCHEVA, Ludmila I; WHITAKER, Christopher J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. **Machine learning**, Springer, v. 51, n. 2, p. 181–207, 2003.

KUSSUL, Ernst; BAIDYK, Tatiana. Rosenblatt perceptrons for handwritten digit recognition. In: **IEEE. NEURAL Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on**. [S.l.: s.n.], 2001. v. 2, p. 1516–1520.

KWON, Jae. Tendermint: Consensus without mining. **Draft v. 0.6, fall**, v. 1, n. 11, 2014.

LAMPORT, Leslie; SHOSTAK, Robert; PEASE, Marshall. **The Byzantine generals Problem**. [S.l.], 1980.

\_\_\_\_\_. The Byzantine generals problem. In: **CONCURRENCY: the Works of Leslie Lamport**. [S.l.: s.n.], 2019. p. 203–226.

LANDASSURI-MORENO, Victor M; BULLINARIA, John A. Neural network ensembles for time series forecasting. In: **ACM. PROCEEDINGS of the 11th Annual conference on Genetic and evolutionary computation**. [S.l.: s.n.], 2009. p. 1235–1242.

LANGE, Sascha; RIEDMILLER, Martin. Deep auto-encoder neural networks in reinforcement learning. In: **IEEE. NEURAL Networks (IJCNN), The 2010 International Joint Conference on**. [S.l.: s.n.], 2010. p. 1–8.

- LÄNGKVIST, Martin; KARLSSON, Lars; LOUTFI, Amy. A review of unsupervised feature learning and deep learning for time-series modeling. **Pattern Recognition Letters**, Elsevier, v. 42, p. 11–24, 2014.
- LECUN, Yann; BENGIO, Yoshua et al. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. 10, p. 1995, 1995.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, Nature Research, v. 521, n. 7553, p. 436–444, 2015.
- LEE, Dong-Hyun et al. Target Propagation. **CoRR**, abs/1412.7525, 2014. arXiv: 1412.7525. Disponível em: <<http://arxiv.org/abs/1412.7525>>.
- LEE, Jae Won. Stock price prediction using reinforcement learning. In: IEEE. INDUSTRIAL Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on. [S.l.: s.n.], 2001. v. 1, p. 690–695.
- LEE, Ming-Chi. Using support vector machine with a hybrid feature selection method to the stock trend prediction. **Expert Systems with Applications**, Elsevier, v. 36, n. 8, p. 10896–10904, 2009.
- LEMOS, F. **Análise Técnica Dos Mercados Financeiros**. [S.l.]: SARAIVA EDITORA, 2015. ISBN 9788502637825.
- LI, Tianyu Ray et al. Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. **Frontiers in Physics**, Frontiers, v. 7, p. 98, 2019.
- LI, Yaliang et al. Ensemble Learning. In: [s.l.]: Chapman e Hall/CRC, 2014. p. 483–509.
- LI, Yuhong; MA, Weihua. Applications of artificial neural networks in financial economics: a survey. In: IEEE. COMPUTATIONAL Intelligence and Design (ISCID), 2010 International Symposium on. [S.l.: s.n.], 2010. v. 1, p. 211–214.
- LILLICRAP, Timothy Paul et al. **Continuous control with deep reinforcement learning**. [S.l.]: Google Patents, jun. 2016. US Patent App. 15/217,758.
- LIMA FILHO, Roberto Ivo da Rocha. **Aspectos neuroeconômicos da tomada de decisão na BM&FBovespa**. 2013. Tese (Doutorado) – Universidade de São Paulo.
- LIN, Long-Ji. **Reinforcement learning for robots using neural networks**. [S.l.], 1993.
- LIU, Bin et al. iRSpot-EL: identify recombination spots with an ensemble learning approach. **Bioinformatics**, Oxford University Press, v. 33, n. 1, p. 35–41, 2016.
- LIU, Yong; YAO, Xin. Simultaneous training of negatively correlated neural networks in an ensemble. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 29, n. 6, p. 716–725, 1999.

- LOVELACE, Jeffrey J; RICKARD, John T; CIOS, Krzysztof J. A spiking neural network alternative for the analog to digital converter. In: IEEE. NEURAL Networks (IJCNN), The 2010 International Joint Conference On. [S.l.: s.n.], 2010. p. 1–8.
- LU, Chi-Jie; WU, Jui-Yu. An efficient CMAC neural network for stock index forecasting. **Expert Systems with Applications**, Elsevier, v. 38, n. 12, p. 15194–15201, 2011.
- LUSS, Ronny; ASPREMONT, Alexandre. Predicting abnormal returns from news using text classification. **Quantitative Finance**, Taylor Francis, v. 15, n. 6, p. 999–1012, 2015.
- MAASS, Wolfgang. Computing with spikes. **Special Issue on Foundations of Information Processing of TELEMATIK**, v. 8, n. 1, p. 32–36, 2002.
- MADAN, Isaac; SALUJA, Shaurya; ZHAO, Aojia. Automated bitcoin trading via machine learning algorithms. URL: [http://cs229.stanford.edu/proj2014/Isaac% 20Madan](http://cs229.stanford.edu/proj2014/Isaac%20Madan), v. 20, 2015.
- MALKIEL, BG. **A Random Walk Down Wall Street (p. 463)**. [S.l.]: New York, NY: WW Norton & Company. Retrieved from <http://http://site.iugaza.edu.ps/wdaya/files>, 2013.
- MALKIEL, Burton G.; FAMA, Eugene F. EFFICIENT CAPITAL MARKETS: A REVIEW OF THEORY AND EMPIRICAL WORK\*. **The Journal of Finance**, Blackwell Publishing Ltd, v. 25, n. 2, p. 383–417, 1970. ISSN 1540-6261. DOI: 10.1111/j.1540-6261.1970.tb00518.x. Disponible em: <<http://dx.doi.org/10.1111/j.1540-6261.1970.tb00518.x>>.
- MARTINETZ, Thomas M; BERKOVICH, Stanislav G; SCHULTEN, Klaus J. Neural-gas network for vector quantization and its application to time-series prediction. **Neural Networks, IEEE Transactions on**, IEEE, v. 4, n. 4, p. 558–569, 1993.
- MARTINEZ, L. C. et al. From an artificial neural network to a stock market day-trading system: A case study on the BM-F BOVESPA. In: 2009 International Joint Conference on Neural Networks. [S.l.: s.n.], jun. 2009. p. 2006–2013. DOI: 10.1109/IJCNN.2009.5179050.
- MARTÍN ABADI et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. [S.l.: s.n.], 2015. Software available from [tensorflow.org](http://tensorflow.org). Disponible em: <<https://www.tensorflow.org/>>.
- MASON, Llew; BARTLETT, Peter L; BAXTER, Jonathan. Improved generalization through explicit optimization of margins. **Machine Learning**, Springer, v. 38, n. 3, p. 243–255, 2000.
- MATTA, Martina; LUNESU, Ilaria; MARCHESI, Michele. Bitcoin Spread Prediction Using Social and Web Search Media. In: UMAP Workshops. [S.l.: s.n.], 2015. p. 1–10.
- \_\_\_\_\_. The predictor impact of Web search media on Bitcoin trading volumes. In: IEEE. 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). [S.l.: s.n.], 2015. v. 1, p. 620–626.

MAZIERES, David. The stellar consensus protocol: A federated model for internet-level consensus. **Stellar Development Foundation**, Citeseer, v. 32, 2015.

MCCULLOCH, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MCGOWAN, Michael J. The rise of computerized high frequency trading: use and controversy. **Duke L. & Tech. Rev.**, HeinOnline, p. i, 2010.

MCNALLY, S.; ROCHE, J.; CATON, S. Predicting the Price of Bitcoin Using Machine Learning. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). [S.l.: s.n.], mar. 2018. p. 339–343. DOI: 10.1109/PDP2018.2018.00060.

MELVILLE, Prem; MOONEY, Raymond J. Creating diversity in ensembles using artificial data. **Information Fusion**, Elsevier, v. 6, n. 1, p. 99–111, 2005.

MICROSOFT. **Speech Recognition Breakthrough for the Spoken, Translated Word**. Microsoft. 2012. Disponível em: <<https://www.youtube.com/watch?v=Nu-nlQqFCKg>>. Acesso em:

MIHM, Stephen. Are Bitcoins the Criminal's Best Friend. **Bloomberg View**, v. 18, 2013.

MIKKULAINEN, Risto et al. Evolving Deep Neural Networks. **arXiv preprint arXiv:1703.00548**, 2017.

MIKOLOV, Tomas et al. Recurrent neural network based language model. In: INTERSPEECH. [S.l.: s.n.], 2010. v. 2, p. 3.

MISSION, Jean-Paul; TAKAHASHI, Takao; CAVINESS, Verne S. Ontogeny of radial and other astroglial cells in murine cerebral cortex. **Glia**, Wiley Online Library, v. 4, n. 2, p. 138–148, 1991.

MNIH, Volodymyr et al. Human-level control through deep reinforcement learning. **Nature**, Nature Research, v. 518, n. 7540, p. 529–533, 2015.

MNIH, Volodymyr et al. Playing atari with deep reinforcement learning. **arXiv preprint arXiv:1312.5602**, 2013.

MOHAMMADI, Milad; DAS, Subhasis. SNN: Stacked Neural Networks. **CoRR**, abs/1605.08512, 2016. arXiv: 1605.08512. Disponível em: <<http://arxiv.org/abs/1605.08512>>.

MOODY, John et al. Performance functions and reinforcement learning for trading systems and portfolios. **Journal of Forecasting**, v. 17, n. 56, p. 441–470, 1998.

MUELLER, Bernhard K. Growth cone guidance: first steps towards a deeper understanding. **Annual review of neuroscience**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 22, n. 1, p. 351–388, 1999.

MUHLBAIER, M. D.; TOPALIS, A.; POLIKAR, R. Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes. **IEEE Transactions on Neural Networks**, v. 20, n. 1, p. 152–168, jan. 2009. ISSN 1045-9227. DOI: 10.1109/TNN.2008.2008326.

MURPHY, J.J. **Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications**. [S.l.]: New York Institute of Finance, 1999. (New York Institute of Finance Series). ISBN 9780735200661. Disponível em: <[https://books.google.com.br/books?id=5zhXEqr%5C\\_IcC](https://books.google.com.br/books?id=5zhXEqr%5C_IcC)>.

NAIR, Arun et al. Massively parallel methods for deep reinforcement learning. **arXiv preprint arXiv:1507.04296**, 2015.

NAKAMOTO, Satoshi. **Bitcoin: A peer-to-peer electronic cash system**. [S.l.: s.n.], 2008. Available at: <https://bitcoin.org/bitcoin.pdf> Last Access: December, 7th 2018.

NOCTOR, Stephen C et al. Neurons derived from radial glial cells establish radial units in neocortex. **Nature**, Nature Publishing Group, v. 409, n. 6821, p. 714, 2001.

O'CONNOR, Niall; MADDEN, Michael G. A neural network approach to predicting stock exchange movements using external factors. **Knowledge-Based Systems**, Elsevier, v. 19, n. 5, p. 371–378, 2006.

OPITZ, David W; SHAVLIK, Jude W. Generating accurate and diverse members of a neural-network ensemble. In: **ADVANCES in neural information processing systems**. [S.l.: s.n.], 1996. p. 535–541.

PAGANO, Marco; VOLPIN, Paolo. Alfred marshall lecture shareholder protection, stock market development, and politics. **Journal of the European Economic Association**, Wiley Online Library, v. 4, n. 2-3, p. 315–341, 2006.

PAISITKRIANGKRAI, S.; SHEN, C.; HENGEL, A. v. d. Pedestrian Detection with Spatially Pooled Features and Structured Ensemble Learning. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 6, p. 1243–1257, jun. 2016. ISSN 0162-8828. DOI: 10.1109/TPAMI.2015.2474388.

PANG, Yan; SUNDARARAJ, Ganeshkumar; REN, Jiewen. Cryptocurrency Price Prediction using Time Series and Social Sentiment Data. In: **PROCEEDINGS of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies**. [S.l.: s.n.], 2019. p. 35–41.

PANT, Dibakar Raj et al. Recurrent neural network based bitcoin price prediction by twitter sentiment analysis. In: **IEEE. 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)**. [S.l.: s.n.], 2018. p. 128–132.

PARKER, Andrew J; NEWSOME, William T. Sense and the single neuron: probing the physiology of perception. **Annual review of neuroscience**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 21, n. 1, p. 227–277, 1998.

PEARSON, Karl. The problem of the random walk. **Nature**, Nature Publishing Group, v. 72, n. 1867, p. 342–342, 1905.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. GloVe: Global Vectors for Word Representation. In: **EMPIRICAL Methods in Natural Language Processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543. Disponível em: <<http://www.aclweb.org/anthology/D14-1162>>.

PERRONE, Michael P.; COOPER, Leona N. When Networks Disagree: Ensemble Methods for Hybrid Neural Networks. In: p. 126–142.

POLIKAR, Robin. Ensemble based systems in decision making. **Circuits and Systems Magazine, IEEE**, v. 6, p. 21–45, 2006.

PRASAD, Sharat C.; PRASAD, Piyush. Deep Recurrent Neural Networks for Time Series Prediction. **CoRR**, abs/1407.5949, 2014. Disponível em: <<http://arxiv.org/abs/1407.5949>>.

PULIDO, Martha Elena; MELIN, Patricia. Optimization of type-2 fuzzy integration in ensemble neural networks for predicting the dow jones time series. In: **IEEE. FUZZY Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American**. [S.l.: s.n.], 2012. p. 1–6.

PULIDO, Martha; MELIN, Patricia; CASTILLO, Oscar. Genetic optimization of ensemble neural networks for complex time series prediction. In: **IEEE. NEURAL Networks (IJCNN), The 2011 International Joint Conference on**. [S.l.: s.n.], 2011. p. 202–206.

\_\_\_\_\_. Optimization of ensemble neural networks with fuzzy integration using the particle swarm algorithm for the US Dollar/MX time series prediction. In: **IEEE. NORBERT Wiener in the 21st Century (21CW), 2014 IEEE Conference on**. [S.l.: s.n.], 2014. p. 1–7.

\_\_\_\_\_. Optimization of type-2 fuzzy integration in ensemble neural networks for predicting the US Dollar/MX pesos time series. In: **IEEE. IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint**. [S.l.: s.n.], 2013. p. 1508–1512.

PULIDO, Martha; MELIN, Patricia; MENDOZA, Olivia. Particle Swarm Optimization of Ensemble Neural Networks with Type-1 and Type-2 Fuzzy Integration for the Taiwan Stock Exchange. In: **NATURE-INSPIRED Design of Hybrid Intelligent Systems**. [S.l.]: Springer, 2017. p. 409–421.

QUINLAN, J. R. Induction of Decision Trees. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 81–106, mar. 1986. ISSN 0885-6125. DOI: 10.1023/A:1022643204877. Disponível em: <<http://dx.doi.org/10.1023/A:1022643204877>>.

QUINLAN, J. Ross. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

RADINSKY, Kira; DAVIDOVICH, Sagie; MARKOVITCH, Shaul. Learning causality for news events prediction. In: **ACM. PROCEEDINGS of the 21st international conference on World Wide Web**. [S.l.: s.n.], 2012. p. 909–918.

RAKIC, Pasko. Elusive radial glial cells: historical and evolutionary perspective. **Glia**, Wiley Online Library, v. 43, n. 1, p. 19–32, 2003.

\_\_\_\_\_. Mode of cell migration to the superficial layers of fetal monkey neocortex. **Journal of Comparative Neurology**, Wiley Online Library, v. 145, n. 1, p. 61–83, 1972.

RATHER, Akhter Mohiuddin; AGARWAL, Arun; SASTRY, VN. Recurrent neural network and a hybrid model for prediction of stock returns. **Expert Systems with Applications**, Elsevier, v. 42, n. 6, p. 3234–3241, 2015.

REID, Fergal; HARRIGAN, Martin. An analysis of anonymity in the bitcoin system. In: IEEE. PRIVACY, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. [S.l.: s.n.], 2011. p. 1318–1326.

RICHARDSON, Leonard. Beautiful soup documentation. **April**, 2007.

ROSEN, Bruce E. Ensemble learning using decorrelated neural networks. **Connection science**, Taylor & Francis, v. 8, n. 3-4, p. 373–384, 1996.

ROUHANI, Saeed; ABEDIN, Ehsan. Crypto-currencies narrated on tweets: a sentiment analysis approach. **International Journal of Ethics and Systems**, Emerald Publishing Limited, 2019.

ROY, Neeta Singh et al. In vitro neurogenesis by progenitor cells isolated from the adult human hippocampus. **Nature medicine**, Nature Publishing Group, v. 6, n. 3, p. 271, 2000.

RUIZ, Eduardo J et al. Correlating financial time series with micro-blogging activity. In: ACM. PROCEEDINGS of the fifth ACM international conference on Web search and data mining. [S.l.: s.n.], 2012. p. 513–522.

RUMELHART, David E.; MCCLELLAND, James L.; PDP RESEARCH GROUP, CORPORATE (Ed.). **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations**. Cambridge, MA, USA: MIT Press, 1986. ISBN 0-262-68053-X.

RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J et al. Learning representations by back-propagating errors. **Cognitive modeling**, v. 5, n. 3, p. 1, 1988.

RUSSELL, Stuart; NORVIG, Peter; INTELLIGENCE, Artificial. A modern approach. **Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs**, v. 25, p. 27, 1995.

RUTA, Dymitr; GABRYS, Bogdan. Neural network ensembles for time series prediction. In: IEEE. NEURAL Networks, 2007. IJCNN 2007. International Joint Conference on. [S.l.: s.n.], 2007. p. 1204–1209.

SABOUR, Sara; FROSST, Nicholas; HINTON, Geoffrey E. Dynamic Routing Between Capsules. **arXiv preprint arXiv:1710.09829**, 2018.

SABOUR, Sara; FROSST, Nicholas; HINTON, Geoffrey E. Matrix capsules with EM routing. **International Conference on Learning Representations**, 2018. Disponível em: <<https://openreview.net/forum?id=HJWLFgWRb>>.

SAKKIS, Georgios et al. Stacking classifiers for anti-spam filtering of e-mail. **arXiv preprint cs/0106040**, 2001.

SALA, Dorel M; CIOS, Krzysztof J. Solving graph algorithms with networks of spiking neurons. **Neural Networks, IEEE Transactions on**, IEEE, v. 10, n. 4, p. 953–957, 1999.

SANTOS, Tiago; KERN, Roman. A Literature Survey of Early Time Series Classification and Deep Learning. In: SAMI@ iKNOW. [S.l.: s.n.], 2016.

SCHAUL, Tom et al. Prioritized experience replay. **arXiv preprint arXiv:1511.05952**, 2015.

SCHOLZ, Martin; KLINKENBERG, Ralf. An ensemble classifier for drifting concepts. In: PORTO, PORTUGAL. PROCEEDINGS of the Second International Workshop on Knowledge Discovery in Data Streams. [S.l.: s.n.], 2005. p. 53–64.

SCHUMAKER, Robert P et al. Evaluating sentiment in financial news articles. **Decision Support Systems**, Elsevier, v. 53, n. 3, p. 458–464, 2012.

SCHWARTZ, David; YOUNGS, Noah; BRITTO, Arthur et al. The ripple protocol consensus algorithm. **Ripple Labs Inc White Paper**, v. 5, n. 8, 2014.

SENI, Giovanni; ELDER, John F. Ensemble methods in data mining: improving accuracy through combining predictions. **Synthesis Lectures on Data Mining and Knowledge Discovery**, Morgan & Claypool Publishers, v. 2, n. 1, p. 1–126, 2010.

SEWELL, Martin. Ensemble learning. **RN**, v. 11, n. 02, 2008.

SHAH, Devavrat; ZHANG, Kang. **Bayesian regression and Bitcoin**. [S.l.: s.n.], 2014. arXiv: 1410.1231 [cs.AI].

SHARPLES, Mike; DOMINGUE, John. The blockchain and kudos: A distributed system for educational record, reputation and reward. In: SPRINGER. EUROPEAN conference on technology enhanced learning. [S.l.: s.n.], 2016. p. 490–496.

SHERSTOV, Alexander A; STONE, Peter. Three automated stock-trading agents: A comparative study. In: SPRINGER. INTERNATIONAL Workshop on Agent-Mediated Electronic Commerce. [S.l.: s.n.], 2004. p. 173–187.

SHLEIFER, Andrei; VISHNY, Robert W. The Limits of Arbitrage. **The Journal of Finance**, v. 52, n. 1, p. 35–55, 1997. DOI: 10.1111/j.1540-6261.1997.tb03807.x. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1997.tb03807.x>. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1997.tb03807.x>>.

SILL, Joseph et al. Feature-weighted linear stacking. **arXiv preprint arXiv:0911.0460**, 2009.

SINDHWANI, Vikas; NIYOGI, Partha; BELKIN, Mikhail. A co-regularization approach to semi-supervised learning with multiple views. In: PROCEEDINGS of ICML workshop on learning with multiple views. [S.l.: s.n.], 2005. p. 74–79.

SINDHWANI, Vikas; ROSENBERG, David S. An RKHS for multi-view learning and manifold co-regularization. In: ACM. PROCEEDINGS of the 25th international conference on Machine learning. [S.l.: s.n.], 2008. p. 976–983.

SMITH, Christopher; JIN, Yaochu. Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. **Neurocomputing**, v. 143, Supplement C, p. 302–311, 2014. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2014.05.062>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231214007279>>.

SMYTH, Padhraic; WOLPERT, David. Linearly combining density estimators via stacking. **Machine Learning**, Springer, v. 36, n. 1-2, p. 59–83, 1999.

SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. **Journal of machine learning research**, v. 15, n. 1, p. 1929–1958, 2014.

STANLEY, Kenneth O; MIIKKULAINEN, Risto. Evolving neural networks through augmenting topologies. **Evolutionary computation**, MIT Press, v. 10, n. 2, p. 99–127, 2002.

STEINERT, Lars; HERFF, Christian. Predicting altcoin returns using social media. **PLoS one**, Public Library of Science, v. 13, n. 12, 2018.

STENQVIST, Evita; LÖNNÖ, Jacob. **Predicting Bitcoin price fluctuation with Twitter sentiment analysis**. [S.l.: s.n.], 2017.

STREHL, Alexander; GHOSH, Joydeep. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. **Journal of machine learning research**, v. 3, Dec, p. 583–617, 2002.

STUDENT. The probable error of a mean. **Biometrika**, JSTOR, p. 1–25, 1908.

SUBRAMANIAN, Harish et al. Designing safe, profitable automated stock trading agents using evolutionary algorithms. In: ACM. PROCEEDINGS of the 8th annual conference on Genetic and evolutionary computation. [S.l.: s.n.], 2006. p. 1777–1784.

SUTTON, Richard S; BARTO, Andrew G. **Reinforcement learning: An introduction**. [S.l.]: MIT press Cambridge, 1998. v. 1.

TANG, E Ke; SUGANTHAN, Ponnuthurai N; YAO, Xin. An analysis of diversity measures. **Machine Learning**, Springer, v. 65, n. 1, p. 247–271, 2006.

TAYLOR, Brett; KIM, Min; CHOI, Anthony. Automated stock trading algorithm using neural networks. In: SPRINGER. PROCEEDINGS of the 2nd International Conference on Intelligent Technologies and Engineering Systems (ICITES2013). [S.l.: s.n.], 2014. p. 849–857.

TESSIER-LAVIGNE, Marc; GOODMAN, Corey S et al. The molecular biology of axon guidance. **Science**, JSTOR, v. 274, n. 5290, p. 1123–1133, 1996.

TING, Kai Ming; WITTEN, Ian H. Issues in stacked generalization. **J. Artif. Intell. Res.(JAIR)**, v. 10, p. 271–289, 1999.

TINO, Peter; SCHITTENKOPF, Christian; DORFFNER, Georg. Financial volatility trading using recurrent neural networks. **IEEE Transactions on Neural Networks**, IEEE, v. 12, n. 4, p. 865–874, 2001.

TRELEAVEN, Philip; GALAS, Michal; LALCHAND, Vidhi. Algorithmic Trading Review. **Commun. ACM**, ACM, New York, NY, USA, v. 56, n. 11, p. 76–85, nov. 2013. ISSN 0001-0782. DOI: 10.1145/2500117. Disponível em: <<http://doi.acm.org/10.1145/2500117>>.

TUMER, Kagan; GHOSH, Joydeep. Analysis of decision boundaries in linearly combined neural classifiers. **Pattern Recognition**, Elsevier, v. 29, n. 2, p. 341–348, 1996.

UBER JUNIOR, Arnaldo et al. iEnsemble2: Committee Machine Model-Based on Heuristically-Accelerated Multiagent Reinforcement Learning. In: \_\_\_\_\_. **Complex, Intelligent, and Software Intensive Systems**. Cham: Springer International Publishing, 2019. p. 363–374. ISBN 978-3-319-93659-8.

URLLIB Python. [S.l.: s.n.], 2017. Disponível em: <https://docs.python.org/3/library/urllib.html>. Acesso em: 01 Set. 2017.

VALENCIA, Franco; GÓMEZ-ESPINOSA, Alfonso; VALDÉS-AGUIRRE, Benjamín. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 21, n. 6, p. 589, 2019.

VALENTINI, Giorgio; DIETTERICH, Thomas G. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. **Journal of Machine Learning Research**, v. 5, Jul, p. 725–775, 2004.

VAN HASSELT, Hado; GUEZ, Arthur; SILVER, David. Deep Reinforcement Learning with Double Q-Learning. In: AAAI. [S.l.: s.n.], 2016. p. 2094–2100.

VAPNIK, Vladimir N.; CHERVONENKIS, Alexey Ya. Support-vector networks. **Machine Learning** **20 (3): 273.**, p. 273, 1995. DOI: 10.1007/BF00994018.

VIOLA, Paul; JONES, Michael. Fast and robust classification using asymmetric adaboost and a detector cascade. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 2002. p. 1311–1318.

\_\_\_\_\_. Rapid Object Detection using a Boosted Cascade of Simple Features. **Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 511–518, 2001.

VREEKEN, Jilles. **Spiking neural networks, an introduction**. [S.l.]: Utrecht University: Information e Computing Sciences, 2003.

WANG, Baohua; HUANG, Hejiao; WANG, Xiaolong. A novel text mining approach to financial time series forecasting. **Neurocomputing**, Elsevier, v. 83, p. 136–145, 2012.

WANG, Haixun et al. Mining concept-drifting data streams using ensemble classifiers. In: ACM. PROCEEDINGS of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. [S.l.: s.n.], 2003. p. 226–235.

WANG, Peng et al. Semantic clustering and convolutional neural network for short text categorization. In: PROCEEDINGS of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). [S.l.: s.n.], 2015. v. 2, p. 352–357.

WANG, Yang et al. Deep Q-trading, 2017.

WANG, Ziyu et al. Dueling network architectures for deep reinforcement learning. **arXiv preprint arXiv:1511.06581**, 2016.

WATKINS, Christopher John Cornish Hellaby. **Learning from Delayed Rewards**. 1989. Tese (Doutorado) – Cambridge University.

WEISS, Gary M; PROVOST, Foster. The effect of class distribution on classifier learning: an empirical study. **Rutgers Univ**, 2001.

WESLEY DA SILVA E CLAUDIMAR DA VEIGA E UBIRATÃ TORTATO, Edgar Filho e. Previsão dos retornos do IBOVESPA utilizando redes neurais artificiais feedforward evolutivas. **Revista Produção Online**, v. 11, n. 4, p. 1114–1140, 2011. ISSN 16761901. DOI: 10.14488/1676-1901.v11i4.784. Disponível em: <<https://producaoonline.org.br/rpo/article/view/784>>.

WILLIAMS, Ronald J; ZIPSER, David. A learning algorithm for continually running fully recurrent neural networks. **Neural computation**, MIT Press, v. 1, n. 2, p. 270–280, 1989.

WOLPERT, David H. Stacked generalization. **Neural networks**, Elsevier, v. 5, n. 2, p. 241–259, 1992.

WOOD, Gavin. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum project yellow paper**, v. 151, p. 1–32, 2014.

WU, Chunpeng et al. Handwritten character recognition by alternately trained relaxation convolutional neural network. In: IEEE. 2014 14th International Conference on Frontiers in Handwriting Recognition. [S.l.: s.n.], 2014. p. 291–296.

XU, Li et al. Deep convolutional neural network for image deconvolution. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 2014. p. 1790–1798.

YANG, B.; GONG, Z. J.; YANG, W. Stock market index prediction using deep neural network ensemble. In: 2017 36th Chinese Control Conference (CCC). [S.l.: s.n.], jul. 2017. p. 3882–3887. DOI: 10.23919/ChiCC.2017.8027964.

YAO, Xin. Evolving artificial neural networks. **Proceedings of the IEEE**, v. 87, n. 9, p. 1423–1447, jul. 1999. ISSN 0018-9219. DOI: 10.1109/5.784219.

YAO, Xin; LIU, Yong. EPNet for chaotic time-series prediction. In: SPRINGER. ASIA-PACIFIC Conference on Simulated Evolution and Learning. [S.l.: s.n.], 1996. p. 146–156.

ZAAMOUT, Khobaib; ZHANG, John Z. Improving classification through ensemble neural networks. In: IEEE. NATURAL Computation (ICNC), 2012 Eighth International Conference on. [S.l.: s.n.], 2012. p. 256–260.

ZAINUDIN, Zanariah; SHAMSUDDIN, Siti Mariyam; HASAN, Shafaatunnur. Convolutional Neural Network Long Short-Term Memory (CNN + LSTM) for Histopathology Cancer Image Classification. In: \_\_\_\_\_. **Machine Intelligence and Signal Processing**. Singapore: Springer Singapore, 2020. p. 235–245.

ZHAI, Yuzheng; HSU, Arthur; HALGAMUGE, Saman K. Combining news and technical indicators in daily stock price trends prediction. In: SPRINGER. INTERNATIONAL symposium on neural networks. [S.l.: s.n.], 2007. p. 1087–1096.

ZHANG, Yu; WEN, Jiangtao. An IoT electric business model based on the protocol of bitcoin. In: IEEE. 2015 18th international conference on intelligence in next generation networks. [S.l.: s.n.], 2015. p. 184–191.

ZHENG, Zibin et al. An overview of blockchain technology: Architecture, consensus, and future trends. In: IEEE. 2017 IEEE international congress on big data (BigData congress). [S.l.: s.n.], 2017. p. 557–564.

ZHOU, Zhi-Hua. **Ensemble methods: foundations and algorithms**. [S.l.]: CRC press, 2012.