



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ  
CAMPUS LUIZ MENEGHEL - CENTRO DE CIÊNCIAS TECNOLÓGICAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL RIBEIRO DA SILVA

**DEEP LEARNING APLICADO À PREDIÇÃO DE  
TENDÊNCIAS NO MERCADO DE AÇÕES**

**BANDEIRANTES**

**2018**



RAFAEL RIBEIRO DA SILVA

**DEEP LEARNING APLICADO À PREDIÇÃO DE  
TENDÊNCIAS NO MERCADO DE AÇÕES**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Norte do Paraná para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Bruno Miguel N. de Souza

**BANDEIRANTES**

**2018**

RAFAEL RIBEIRO DA SILVA

**DEEP LEARNING APLICADO À PREDIÇÃO DE  
TENDÊNCIAS NO MERCADO DE AÇÕES**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Norte do Paraná para obtenção do título de Bacharel em Ciência da Computação.

**BANCA EXAMINADORA**

---

Prof. Me. Bruno Miguel N. de Souza  
Universidade Estadual do Norte do Paraná  
Orientador

---

Prof. Me. Glauco Carlos Silva  
Universidade Estadual do Norte do Paraná

---

Prof. Me. Wellington Aparecido Della Mura  
Universidade Estadual do Norte do Paraná

Bandeirantes, 19 de novembro de 2018

## AGRADECIMENTOS

Agradeço meus pais, que me deram todo o apoio necessário e não mediram esforços para me ajudar a chegar até aqui; meu orientador por toda a ajuda prestada no desenvolvimento deste trabalho; os membros da banca pelas orientações complementares; os professores em geral, que me ajudaram a adquirir os conhecimentos que me tornaram capaz de desenvolver este trabalho e meus colegas que me ajudaram sempre que necessário durante estes 4 anos.



SILVA, R. R. **Deep learning aplicado à predição de tendências no mercado de ações**. 56 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual do Norte do Paraná, Bandeirantes, 2018.

## RESUMO

Visando alcançar o objetivo de treinar redes neurais para torná-las capazes de identificar tendências no preço de uma ação, foram coletados dados sobre as ações de 7.145 empresas durante o período de 1 mês. Esses dados foram transformados em gráficos candlestick, que utilizando conceitos de análise técnica, foram classificados como sendo gráficos com tendências de alta, baixa e consolidação e formaram as bases utilizadas para treinar, validar e testar modelos das redes ResNet e Xception. Foi possível concluir que ambas obtiveram resultados satisfatórios, porém com vantagem para a ResNet, que conseguiu precisões maiores desde a fase de treinamento até os testes finais. Nos testes finais a ResNet chegou a 87% de acurácia e F1 Score, enquanto a Xception chegou a 86%.

**Palavras-chave:** Deep learning. Redes neurais. Mercado de ações.



SILVA, R. R. **Deep learning applied to stock market trend prediction**. 56 p. Final Project (Bachelor of Science in Computer Science) – State University Northern of Parana , Bandeirantes, 2018.

## **ABSTRACT**

Aiming to achieve the goal of training neural networks to enable them to identify trends in the price of a stock, data were collected on the stocks of 7,145 companies during the period of one month. These data were transformed into candlestick charts that, using technical analysis concepts, were classified as being charts with high, low and consolidation tendencies and formed the bases used to train, validate and test ResNet and Xception network models. It was possible to conclude that both obtained satisfactory results, but with advantage to ResNet, that obtained greater precisions from the training phase until the final tests. In the final tests ResNet reached 87% of accuracy and F1 Score, while Xception reached 86%.

**Keywords:** Deep learning. Neural networks. Stock market.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Gráfico Candlestick . . . . .	18
Figura 2 – Camadas de uma rede neural . . . . .	28
Figura 3 – Rede feed-forward . . . . .	29
Figura 4 – Rede neural recorrente . . . . .	29
Figura 5 – Porcentagem de erro durante treino (esquerda) e teste (direita) de redes neurais com 20 e 56 camadas analisando imagens da base CIFAR-10. Quanto mais camadas, maior a porcentagem de erro. He et al.[1] . . . . .	29
Figura 6 – Arquitetura de um bloco residual. He et al.[1] . . . . .	30
Figura 7 – Comparação dos resultados obtidos por redes que não utilizam blocos residuais (esquerda) e ResNets (direita) He et al.[1] . . . . .	30
Figura 8 – Módulo Inception Chollet[2] . . . . .	31
Figura 9 – Arquitetura Xception Chollet[2] . . . . .	32
Figura 10 – Candle de alta (esquerda) e candle de baixa (direita) . . . . .	37
Figura 11 – Engolfo de alta (esquerda) e Engolfo de baixa (direita). Chin et al.[3] . . . . .	37
Figura 12 – Informações obtidas sobre a Agilent Technologies Inc. no endpoint /ref-data/symbols . . . . .	41
Figura 13 – Arquivo de texto com os identificadores das empresas . . . . .	42
Figura 14 – Informações para compor um dos candles do gráfico da Agilent Technologies Inc. obtidas no endpoint /stock/A/time-series . . . . .	42
Figura 15 – Informações sobre alguns candles utilizados para formar o gráfico da Agilent Technologies Inc. salvos no banco de dados . . . . .	43
Figura 16 – Gráfico gerado pelo Bokeh com os dados da Agilent Technologies Inc. contidas no banco de dados . . . . .	44
Figura 17 – Comparação dos resultados finais de acurácia e F1 Score das redes ResNet e Xception . . . . .	54
Figura 18 – Comparação dos resultados finais de Loss e Mean Absolute Error das redes ResNet e Xception . . . . .	54



## LISTA DE TABELAS

Tabela 1	– Resultados das iterações nas camadas totalmente conectadas da ResNet com a base de treino . . . . .	44
Tabela 2	– Resultados das iterações nas camadas totalmente conectadas da ResNet com a base de validação . . . . .	45
Tabela 3	– Melhor resultado obtido em 10 iterações na ResNet na base de validação	45
Tabela 4	– Resultados das 30 iterações de processamento realizadas em toda a rede ResNet com a base de treino . . . . .	46
Tabela 5	– Resultados das 30 iterações de processamento realizadas em toda a rede ResNet com a base de validação . . . . .	47
Tabela 6	– Iteração que resultou na rede ResNet com o maior F1 Score durante o treinamento . . . . .	47
Tabela 7	– Resultados finais da ResNet, quando submetida às amostras da base de teste . . . . .	48
Tabela 8	– Resultados das 10 primeiras iterações nas camadas totalmente conectadas da Xception com a base de treino . . . . .	48
Tabela 9	– Resultados das 10 primeiras iterações nas camadas totalmente conectadas da Xception com a base de validação . . . . .	48
Tabela 10	– Resultados das outras 10 iterações nas camadas totalmente conectadas da Xception com a base de treino . . . . .	49
Tabela 11	– Resultados das outras 10 iterações nas camadas totalmente conectadas da Xception com a base de validação . . . . .	49
Tabela 12	– Melhor resultado obtido em 20 iterações na Xception . . . . .	49
Tabela 13	– Resultados das 30 iterações de processamento realizadas em toda a rede Xception com a base de treino . . . . .	50
Tabela 14	– Resultados das 30 iterações de processamento realizadas em toda a rede Xception com a base de validação . . . . .	51
Tabela 15	– Iteração que resultou na rede Xception com o maior F1 Score durante o treinamento . . . . .	51
Tabela 16	– Resultados finais da Xception, quando submetida às amostras da base de teste . . . . .	51



## LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
DL	Deep Learning
ML	Machine Learning
RN	Rede Neural



# SUMÁRIO

1	INTRODUÇÃO . . . . .	17
1.1	Contextualização . . . . .	17
1.2	Formulação e Escopo do Problema . . . . .	18
1.3	Objetivos . . . . .	19
1.3.1	Objetivo Geral . . . . .	19
1.3.2	Objetivos Específicos . . . . .	19
1.4	Justificativas . . . . .	19
1.5	Metodologia . . . . .	19
1.5.1	Classificação . . . . .	19
1.5.2	Ferramentas . . . . .	20
1.5.3	Condução da pesquisa . . . . .	21
1.6	Organização do Trabalho . . . . .	22
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	23
2.1	Inteligência Artificial, Machine Learning, Redes Neurais e Deep Learning . . . . .	23
2.1.1	Inteligência Artificial . . . . .	23
2.1.2	Machine Learning . . . . .	25
2.1.3	Redes Neurais . . . . .	27
2.1.4	Deep Learning . . . . .	31
2.1.5	Medidores de desempenho . . . . .	33
2.2	Mercado de Ações . . . . .	34
2.2.1	Bolsas de Valores . . . . .	35
2.2.2	Teoria de Dow . . . . .	35
2.2.3	Análise Técnica . . . . .	36
2.3	Trabalhos relacionados . . . . .	37
3	DESENVOLVIMENTO . . . . .	41
3.1	Coleta de dados, geração e classificação dos gráficos . . . . .	41
3.2	Resultados da ResNet . . . . .	44
3.3	Resultados da Xception . . . . .	46
4	CONCLUSÃO . . . . .	53
	REFERÊNCIAS . . . . .	55



# 1 INTRODUÇÃO

Nessa seção serão apresentados diversos aspectos iniciais com o intuito de apresentar o trabalho, como contextualização, apresentação do problema, justificativas, objetivos, metodologia e organização dos capítulos.

## 1.1 Contextualização

Na opinião de Fontanills, Gentile e Cawood[4], qualquer pessoa que possua uma quantidade considerável de dinheiro disponível deveria investir uma parte dessa quantia no mercado de ações. O investimento nesse mercado tem melhorado a renda e por consequência, a qualidade de vida de milhões de pessoas ao decorrer de sua existência. Porém, é preciso ressaltar que o fato de que inúmeras pessoas vem conseguindo tomar decisões corretas e assim conseguido lucro, não significa que todos conseguem lucrar o tempo todo.

Para ter sucesso com ações, é necessário considerar os riscos. Muitas pessoas ainda tratam o investimento no mercado de ações como uma tipo de aposta, que depende exclusivamente de sorte. Fontanills, Gentile e Cawood[4] nos lembram que, apesar de o desempenho de um investidor poder ser influenciado pelo acaso, não é assim que as coisas funcionam.

Todas as informações possíveis devem ser buscadas por novos investidores a fim de que os investimentos sejam feitos com maior probabilidade de renderem lucros e que gerem o menor impacto negativo caso escolhas erradas sejam feitas.

Com o surgimento e popularização da internet, adquirir essas informações ficou muito mais fácil. Conhecimento sobre diferentes aspectos do investimento no mercado de ações são abundantes, podendo ser recebidas de diversas formas, como e-mails, websites e por meio da utilização de aplicativos. Até mesmo as redes sociais podem ajudar nesse sentido, reunindo pessoas com esse interesse em comum para compartilharem conhecimento.

Esse aumento na quantidade de informações e dados sobre o comportamento do mercado de ações diariamente disponíveis para acesso possibilita o desenvolvimento de pesquisas de uma das maiores tendências da inteligência artificial (IA) nos dias de hoje, o deep learning (DL).

Com o DL pode-se utilizar grandes quantidades de dados para realizar o "treinamento" de um computador, o tornando capaz de reconhecer padrões, tendências, realizar previsões e diversas outras aplicações. Este trabalho visa se aproveitar da abundância de dados disponíveis sobre o comportamento do mercado de ações para realizar o treinamento

de modelos de predição.

O modelo desenvolvido será capaz de receber como entrada um gráfico candlestick com a variação de preço de uma determinada ação em um espaço de tempo e a partir da análise desse gráfico e de sua capacidade de realizar predições baseadas nos gráficos com que foi treinado, determinar se o preço daquela ação tende a aumentar, diminuir ou se consolidar. Permitindo assim, que investidores possam saber o momento certo de comprar e vender uma determinada ação.

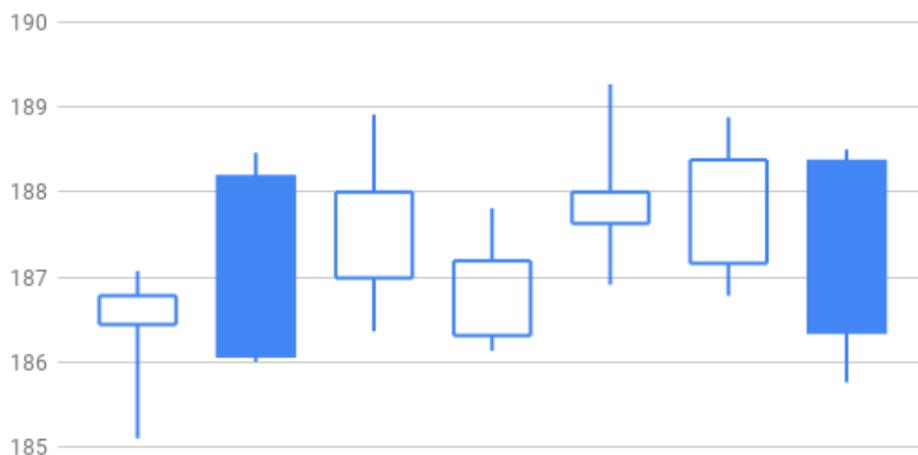


Figura 1 – Gráfico Candlestick

## 1.2 Formulação e Escopo do Problema

As bolsas de valores são importantes ambientes de negociação, onde investidores podem comprar e vender ações. Sendo assim, são responsáveis por grandes movimentações na economia.

Ações representam parte do capital social de uma empresa, ou seja, o proprietário da ação lucra caso esta se valorize e perde dinheiro em caso de depreciação. A variação no preço das ações acontece de acordo com as ofertas de compra e venda dos investidores. Caso haja grande interesse na compra de uma ação, ela se valoriza.

Qualquer tipo de investimento gera riscos, esses riscos precisam ser muito bem estudados para serem evitados ao máximo, mas também para que se saiba exatamente que atitudes tomar caso ocorram.

Ter à disposição uma ferramenta capaz de realizar análises somente a partir de gráficos, ou seja, sem exigir conhecimentos técnicos, pode ser extremamente útil, servindo como uma forma a mais de prevenir os riscos de se investir dinheiro.

## 1.3 Objetivos

Nessa seção, serão apresentados os objetivos gerais e específicos que serão buscados no desenvolvimento deste trabalho.

### 1.3.1 Objetivo Geral

O principal objetivo deste trabalho é treinar um modelo, tornando-o capaz de aprender a identificar tendências na variação do preço de uma ação a partir de imagens que representam esta variação, tornando possível a identificação de momentos cruciais para realizar vendas e compras de ações.

### 1.3.2 Objetivos Específicos

Alguns dos objetivos específicos deste trabalho são:

- Coletar e rotular imagens de gráficos de ações;
- Implementar técnicas de machine learning (ML) e DL;
- Treinar redes neurais (RN) para identificação de tendência em ações;
- Implementar técnicas de ML existentes para análise e predição de preços de ações.

## 1.4 Justificativas

Visando que identificar tendências em ações pode ser extremamente útil para um investidor, uma ferramenta que o auxilie na predição de altas, baixas e consolidações no preço de uma determinada ação possibilita a identificação do melhor momento para realizar compra e venda.

## 1.5 Metodologia

Nessa seção, a pesquisa será descrita em relação à sua natureza, objetivos, abordagem, etc. Além de descrever como a pesquisa será conduzida e falar sobre as ferramentas que serão utilizadas.

### 1.5.1 Classificação

Com relação à sua natureza, este trabalho desenvolve uma pesquisa aplicada, pois resulta em um modelo de predição capaz de receber a imagem de um gráfico candlestick e a partir dessa imagem, utilizar redes neurais artificiais profundas (deep learning) treinadas para identificar tendências de alta, baixa e consolidação no preço da ação representada na imagem de entrada.

No que diz respeito aos objetivos, esta se caracteriza como uma pesquisa explicativa, pois visa desenvolver na prática os conceitos abordados na revisão bibliográfica,

através da implementação das redes neurais artificiais profundas e analisar os fenômenos observados, ou seja, os resultados finais das previsões, com o objetivo de medir sua eficiência na capacidade de detectar tendências corretamente.

Sobre a abordagem utilizada, este trabalho pode ser definido como tendo uma abordagem quantitativa, uma vez que os resultados finais consistem na relação de erros e acertos por parte do modelo de previsões, baseando-se nos dados fornecidos para que o sistema fosse treinado. Será analisada numericamente a eficiência alcançada pelo modelo em desempenhar sua função.

Por fim, tratando da classificação em relação aos procedimentos, esta é uma pesquisa experimental, pois tem como objeto de estudo as redes neurais artificiais com diversas camadas intermediárias, técnica conhecida como deep learning, para realizar o processamento e análise das informações extraídas das imagens dos gráficos utilizadas no treinamento, com o objetivo de observar os efeitos, ou seja, sua eficiência em realizar previsões corretas sobre o preço de uma determinada ação.

### 1.5.2 Ferramentas

Nessa seção, serão apresentadas as ferramentas utilizadas para a coleta de dados e desenvolvimento do modelo de previsão.

Para o desenvolvimento foi usado o Python, uma linguagem de programação interpretada bastante popular, criada em 1991 por Guido van Rossum. Como Downey[5] lembra, o Python, assim como C, C++, Java e muitas outras, é uma linguagem de programação de alto nível, ou seja, uma linguagem cujos códigos escritos nela são mais próximos das linguagens humanas e mais distantes dos códigos de máquina. Sua sintaxe enfatiza a legibilidade. Python é uma linguagem cujos códigos podem ser escritos seguindo diferentes paradigmas, como funcional ou orientado a objetos, por exemplo.

Em relação a deep learning optou-se por utilizar o framework Keras, por este abstrair mecanismos de programação e possibilitar a customização das arquiteturas de deep learning. Segundo Zhang[6], “Keras é uma API de alto nível para aprendizado de máquina em Python, que permite executar redes neurais”. Ao fazer a utilização do Keras, o usuário tem à sua disposição uma coleção de métodos para serem utilizados.

Uma alternativa ao Keras é o Pytorch, porém, esta ferramenta apresenta um nível mais baixo de abstração, o que tornaria o desenvolvimento mais custoso.

Juntamente com o Keras é utilizado o Tensor Flow, descrito por Raschka[7] como "uma biblioteca de código aberto para computação numérica e aprendizado profundo". O TensorFlow foi desenvolvido pelo Google e passou a ser uma biblioteca de código aberto em 9 de novembro de 2015. Impulsionado por sua biblioteca para a linguagem Python, o TensorFlor se tornou bastante popular tanto por parte de empresas quanto no

meio acadêmico. O TensorFlow fornece, por exemplo, estruturas de dados e cálculos para matrizes multidimensionais.

O Theano é um exemplo de alternativa que pode ser utilizada em conjunto com o Keras, mas optou-se por utilizar o Tensor Flow porque se trata de uma ferramenta amplamente utilizada pela comunidade científica.

Para o armazenamento dos dados utilizados para a geração dos gráficos, foi utilizado o SQLite. Para a geração dos gráficos, foi utilizada a biblioteca Bokeh.

### 1.5.3 Condução da pesquisa

Para realizar a coleta das imagens, a API da IEX Group, que fornece diversos dados sobre a movimentação dos preços de inúmeras ações foi acessada por scripts desenvolvidos na linguagem Python.

A princípio, acessando a API, foram coletados os identificadores de todas as empresas que tem sua informações à disposição na API. Em seguida, utilizando estes identificadores, foram obtidos candles (valores de alta, baixa, abertura e fechamento) do período de 1 mês de cada empresa.

Os dados coletados da API foram salvos em um banco de dados SQLite, em uma tabela onde cada linha guarda informações sobre um candle específico. Cada linha tem as colunas "symbol", "date", "high", "low", "open" e "close". A coluna "symbol" guarda o identificador da empresa, possibilitando saber qual candle tem informações de qual empresa. A coluna "date" informa o dia, mês e ano dos valores registrados no candle e as colunas "high", "low", "open" e "close" apresentam os valores de alta, baixa, abertura e fechamento do candle.

Os dados salvos no banco de dados foram posteriormente consultados por outro script, responsável por capturá-los e fornece-los como dados de entrada para a biblioteca Bokeh, que fez o trabalho de transformar aquelas informações em gráficos e salvá-los como imagens.

Assim que todas as imagens foram geradas, elas foram classificadas como tendências de alta, baixa e consolidação. Depois, foram separadas em 3 bases, uma para ser utilizada para treino, uma para teste e outra para validação.

Após a coleta, geração, classificação e separação das imagens, as bases de treino, teste e validação foram aplicadas nas redes neurais Xception e ResNet.

Para cada uma das redes, os processos de treinamento e validação foram feitos em duas etapas. Primeiramente, utilizando redes pré treinadas com a base ImageNet, que ajudam a preparar a rede para resolver tarefas genéricas, as bases de treino e validação foram aplicadas somente às camadas finais totalmente conectadas da rede por 10 iterações,

afim de encontrar os melhores pesos para essas camadas. Encontrada a melhor combinação de pesos para as camadas totalmente conectadas, as bases de treino e validação foram aplicadas à toda a rede por 30 iterações, utilizando nas camadas totalmente conectadas, os pesos salvos na primeira etapa.

Após realizadas as 30 iterações e selecionada a iteração com o melhor resultado, os pesos da rede nesta iteração foram salvos para serem utilizados com a base de teste.

## 1.6 Organização do Trabalho

O segundo capítulo deste trabalho traz a fundamentação teórica utilizada para o desenvolvimento. Nesta seção, são abordados os aspectos teóricos de diversos assuntos relacionados ao que foi desenvolvido, servindo também como uma introdução a esses assuntos para o leitor que não tem conhecimento sobre alguns termos e definições que serão necessários conhecer para o entendimento do restante do texto.

Na fundamentação teórica, são apresentados diferentes assuntos, sendo os principais, a definição de termos muito utilizados como IA, ML, DL, RN, as redes Xception e ResNet, além da ideia do funcionamento do mercado de ações e das bolsas de valores e análise técnica.

Também são apresentados assuntos mais específicos, como as diferentes categorias dos sistemas inteligentes, os diferentes tipos de aprendizados, as principais arquiteturas das RNs, e as principais ideias da teoria de Dow, base de muitos estudos sobre análise técnica no mercado de ações utilizados hoje em dia.

No terceiro capítulo, são mostrados todos os passos em relação ao desenvolvimento do trabalho. Este capítulo descreve exatamente como tudo foi desenvolvido e testado para chegar nos resultados finais.

O quarto e último capítulo apresenta as conclusões às quais foram possíveis chegar com os resultados, considerando tudo o que foi descrito nos capítulos anteriores.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nessa seção, serão apresentados conceitos sobre as áreas de conhecimento tratadas neste trabalho, como a diferenciação entre sub-campos da IA, análise técnica e o funcionamento das bolsas de valores e do mercado de ações.

### 2.1 Inteligência Artificial, Machine Learning, Redes Neurais e Deep Learning

Nessa seção serão apresentadas as diferenças conceituais entre os termos IA, ML, RN e DL, afim de diferenciar essas áreas e compreender como elas se relacionam.

#### 2.1.1 Inteligência Artificial

As definições dos termos “inteligência artificial”, “machine learning” e “deep learning” podem ser bastante difíceis de distinguir, não somente para leigos, mas também para aqueles que já tem certa afinidade com tecnologia.

Antes de definir o que é IA, é preciso pensar sobre o conceito de inteligência. Para Alvi[8], a inteligência pode ser definida como uma característica utilizada quando se está tentando resolver algo. É o processo de analisar as possibilidades do que pode ser feito e encontrar uma solução. Ou seja, possuir as habilidades necessárias para realizar este processo, tais como pensar, planejar, resolver problemas e aprender, por exemplo, demonstra inteligência.

Existem diferentes formas de definir o que é uma IA. Swarup[9] cita que livros da área costumam definir a IA como “o estudo e projeto de agentes inteligentes”, onde esses “agentes inteligentes” são sistemas capazes de “perceber seu ambiente” a fim de “realizar ações que maximizam suas chances de sucesso”. Swarup[9] também lembra que John McCarthy, cientista da computação que, em 1956, inventou o termo “inteligência artificial”, definiu a área como “a ciência e engenharia de fazer máquinas inteligentes”.

Swarup[9] cita alguns períodos e momentos (ou acontecimentos) importantes na linha do tempo da IA. O primeiro período, entre 1943 e 1955 se caracteriza pelo primeiro trabalho reconhecido como IA. Desenvolvido em 1943 por Warren McCulloch e Walter Pitts. Neste trabalho, os autores propuseram um modelo de neurônios artificiais interligados, formando uma rede, sendo que cada um deles poderia estar em um estado de “ligado” ou “desligado”. Um neurônio passa para o estado “ligado” em resposta aos estímulos dos neurônios vizinhos. McCulloch e Pitts concluíram que essas redes, se bem definidas, são capazes de aprender.

No ano de 1956, John McCarthy, personalidade importante na área, mudou-se para Dartmouth College. No verão daquele ano, McCarthy se reuniu com outros pesquisadores para organizar um workshop com duração de 2 meses para estudos sobre diferentes assuntos como a teoria dos autômatos, RNs e também sobre inteligência. Ao fim dos dois meses, nenhuma nova descoberta foi feita, mas o evento foi caracterizado pela primeira utilização do termo “inteligência artificial” por McCarthy.

O período entre os anos de 1952 e 1969 foi marcado por grandes expectativas e grande entusiasmo sobre tudo que a IA seria capaz de realizar. Os primeiros anos foram de bastante sucesso, porém este sucesso era limitado pela capacidade das máquinas e ferramentas disponíveis na época. O entusiasmo desse período se deu pelo fato de que até poucos anos antes, os computadores eram vistos somente como máquinas capazes de realizar cálculos. Demonstrações de inteligência vindas de máquinas era algo novo.

Entre 1966 e 1973, pesquisadores e entusiastas da área provaram uma dose de realidade. Muito se falava, por parte dos pesquisadores, sobre o que seria possível realizar com a utilização das inteligências artificiais e em quanto tempo tudo aquilo seria possível. Herbert Simon, por exemplo, deu uma declaração no ano de 1957 onde dizia que a capacidade das inteligências artificiais de resolver diferentes problemas iria aumentar rapidamente até que em um futuro não tão distante, seria possível lidar com tantos tipos diferentes de problemas quanto a mente humana é capaz. Simon também fez previsões mais concretas, como dizer que em dentro de 10 anos, um computador seria campeão de xadrez e provar um teorema matemático significativo. As previsões de Simon se concretizaram, mas levaram por volta de 40 anos. O excesso de confiança de Simon foi motivado pelo fato de que, no princípio, lidando com problemas mais simples, o desenvolvimento das inteligências artificiais parecia bastante promissor, mas quando se tratava de problemas mais complexos, o desempenho era bem diferente.

No período entre 1969 e 1979, houve a popularização dos sistemas baseados em conhecimento. Em seus primeiros anos, as inteligências artificiais se resumiam a um mecanismo de busca de propósito geral, que funcionava tentando encadear passos para formar um raciocínio na tentativa de encontrar soluções para o problema. Esta abordagem é por vezes chamada de “método fraco”, pois apesar de ser, de certa forma geral, apresenta dificuldades com problemas grandes ou mais complexos. Uma alternativa para essa abordagem é utilizar conhecimento específico de um domínio, fazendo com que seja possível lidar com mais facilidade com problemas de áreas restritas. Guardar esse conhecimento foi possível graças à maior disponibilidade de máquinas com mais espaço de armazenamento na década de 1970, como lembrado por Swarup[9].

Na década de 1980, surgiram os sistemas especialistas comerciais. O primeiro bem-sucedido foi o R1, utilizado na Digital Equipment Corporation. Já em 1986, sua utilização era responsável por uma economia de cerca de U\$40 milhões por ano na empresa. Sistemas

especialistas passaram a ser utilizados (ou ao menos estudados) por quase todas as grandes corporações dos Estados Unidos. Com o aumento do interesse das empresas pela IA e pelos sistemas especialistas, essa indústria, de alguns milhões de dólares no início da década de 1980, cresceu para bilhões já em 1988.

Em 1986, alguns grupos de pesquisadores reinventaram um antigo algoritmo de aprendizagem de propagação retrógrada, introduzido em 1969 por Bryson e Ho. Esse algoritmo, depois de modificado foi aplicado a diversos problemas de aprendizagem em diferentes áreas, como ciência da computação e psicologia. Após a publicação, os resultados geraram grande entusiasmo, trazendo de volta o interesse no estudo e nas aplicações das RNs.

### 2.1.2 Machine Learning

O ML dá aos sistemas a habilidade de aprender e melhorar sua performance baseando-se em experiências e conhecimentos sobre a tarefa que desempenha, sem a necessidade de ser explicitamente programado.

Em uma definição sobre a ideia do funcionamento do ML, Shalev-Shwartz e Ben-David[10] dizem que, quando estamos trabalhando com desenvolvimento nesta área, o objetivo é programar computadores capazes de “aprender” a partir de uma determinada entrada recebida. Ou seja, transformando os quaisquer tipos de dados em conhecimento.

Sobre o funcionamento dos algoritmos de aprendizado, Shalev-Shwartz e Ben-David[10] mencionam que estes recebem como entrada os dados que serão utilizados para realizar o treinamento e a saída é alguma especialidade, geralmente programa de computador capaz de executar uma determinada tarefa.

Em uma definição menos técnica, Alpaydin[11] lembra da definição de um algoritmo, “Um algoritmo é uma sequência de instruções que devem ser realizadas para transformar a entrada em saída” e dá o exemplo de um algoritmo de ordenação, onde:

- Entrada: lista de números.
- Sequência de instruções: instruções em uma determinada linguagem de programação que comparam os números da lista da entrada e os trocam de posição se necessário.
- Saída: lista ordenada com os números contidos na lista da entrada.

Alpaydin[11] também lembra que, para realizar uma determinada tarefa, podem haver diferentes possibilidades, ou seja, para resolver um problema com a utilização de um computador, podem existir diversos algoritmos diferentes.

Porém, não existem algoritmos para realizar todas as tarefas de forma satisfatória para todos os casos. Um exemplo também citado por Alpaydin[11] é a detecção de spam recebido por e-mail:

- Entrada: conteúdo do e-mail (caracteres, imagens, links, anexos, etc)

- Sequência de instruções: ?
- Saída: mensagem sim/não indicando se o conteúdo do e-mail analisado é spam ou não

Por não conhecermos tudo o que pode ser considerado spam, é muito difícil criar um algoritmo capaz de identificar e-mails de spam com uma porcentagem alta de acerto. Porém, essa falta de conhecimento pode ser compensada com dados. Compilando a maior quantidade possível de e-mails, sabendo quais deles são spam, podemos “aprender” quais características fazem dele spam. O que queremos é que o computador extraia automaticamente o algoritmo para realizar essa tarefa.

No caso do exemplo do algoritmo de ordenação, não precisamos aprender a fazer a ordenação pois já contamos com inúmeros algoritmos para isso, mas quando precisamos resolver um problema para o qual não temos ou é muito difícil desenvolver um algoritmo, podemos utilizar dados para “aprender” o que deve ser feito.

Com a tecnologia disponível nos dias de hoje, somos capazes de armazenar e processar grandes quantidades de dados por um preço baixo, além de não haver a necessidade de utilizar somente os dados armazenados por nós, mas também todo tipo de dado que conseguimos acessar em outras máquinas pela utilização de redes.

Alpaydin[11] cita também o exemplo de uma rede de supermercados com lojas por todo o território de um determinado país. Cada transação realizada em cada unidade da rede tem seus detalhes registrados, contando com informações como data, produtos comprados, preço da compra e etc. Para uma grande rede, com muitos clientes, isso pode resultar em uma quantidade muito grande de dados gerados todos os dias. Tendo posse desses dados, a rede de supermercados pode, por exemplo, tentar prever quais clientes poderiam estar interessados em quais tipos de produtos.

Este é outro exemplo onde não é possível ter uma visão clara de todas as instruções necessárias para se criar um algoritmo. Os produtos pelos quais um determinado cliente se interessa podem mudar no decorrer do tempo, dependendo da unidade em que ele está visitando, entre diversos outros aspectos.

Depois de reunir os dados necessários para “ensinar” ao computador aquilo que desejamos que ele faça, ele não necessariamente vai conseguir “aprender” todo o necessário para realizar seu trabalho sem erros, mas é possível conseguir uma boa aproximação. Embora o computador não consiga em 100% dos casos “aprender” tudo baseado nos dados que lhe foram disponibilizados, ele ainda pode ser capaz de fazer previsões e identificar certos padrões. Estes padrões podem ajudar a entender determinado fenômeno da área de aplicação do problema para o qual o computador foi treinado.

Segundo Shalev-Shwartz e Ben-David[10], devido ao fato de a aprendizagem ser um campo bastante amplo, o ML, por consequência, também amplo, possui subcampos, cada um dele lidando com diferentes tipos de aprendizado, como a aprendizagem supervisionada

e não supervisionada.

Como no exemplo também mencionado por Shalev-Shwartz e Ben-David[10], considere a tarefa de detectar e-mails de spam. Para essa tarefa, são fornecidos e-mails para treinamento. Estes e-mails já vem com rótulos, através dos quais já é possível saber qual e-mail é spam e qual não é. Com base nas características dos e-mails rotulados recebidos como treinamento, o aluno deve rotular os novos e-mails que receber, ou seja, identificar se esses novos e-mails são spam ou não.

Este é um exemplo de aprendizagem supervisionada, onde os dados utilizados para ensinar o aluno contém informações significativas (o aluno foi ensinado com dados que já tinham sido classificados como spam e não spam).

Utilizando o mesmo exemplo de detecção de e-mails de spam, imagine que o aluno tenha recebido os e-mails de treinamento sem os rótulos informando se cada um dos e-mails é ou não spam. Este é o aprendizado não supervisionado, onde não há distinção entre os dados recebidos para treinamento. O aluno deve analisar esses e-mails, entender quais são spam e quais não são e quais características não necessárias analisar para chegar a essa conclusão.

### 2.1.3 Redes Neurais

As RNs artificiais são introduzidas por Rojas[12] como uma tentativa de modelar o processamento de informações baseando-se nos sistemas nervosos. Para podermos projetar RNs artificiais capazes de serem simuladas e analisadas, é preciso considerar as propriedades das RNs biológicas.

Sistemas nervosos são conjuntos de milhões de células interconectadas. Essas células, bastante complexas, lidam de diferentes maneiras com os sinais que recebem como entrada. As entradas sensoriais são codificadas e processadas para que então, a partir delas, uma resposta apropriada seja chamada.

Shalev-Shwartz e Ben-David[10] definem uma RN artificial como “modelo computacional inspirado na estrutura das RNs do cérebro”. O aprendizado com a utilização de RNs foi proposto em meados do século XX. É uma forma efetiva de aprendizado, apresentando resultados satisfatórios em diferentes tarefas de aprendizado.

Segundo Silva[13], uma RN artificial pode ser dividida em 3 partes: camada de entrada, camadas ocultas (ou intermediárias) e camada de saída. As RNs artificiais podem ser representadas por um grafo direcionado, onde os nós correspondem aos neurônios e as arestas correspondem aos elos entre eles.

A camada de entrada é responsável por receber informações do ambiente externo, ou seja, as informações necessárias para que a rede possa desempenhar sua função. Os neurônios das camadas intermediárias realizam o processo de extração de informações

e padrões referentes ao processo para qual a rede foi designada. São responsáveis pela maior parte do processamento interno da rede. No caso das camadas de saída, sua responsabilidade é produzir e apresentar as saídas finais da rede, ou seja, o resultado do processamento realizado pelos neurônios das camadas anteriores.

A entrada, para cada neurônio, é uma soma ponderada das saídas dos neurônios que estão conectados às suas entradas. A aprendizagem consiste em encontrar pesos para as conexões entre os neurônios que fazem com que a RN desempenhe determinada função.

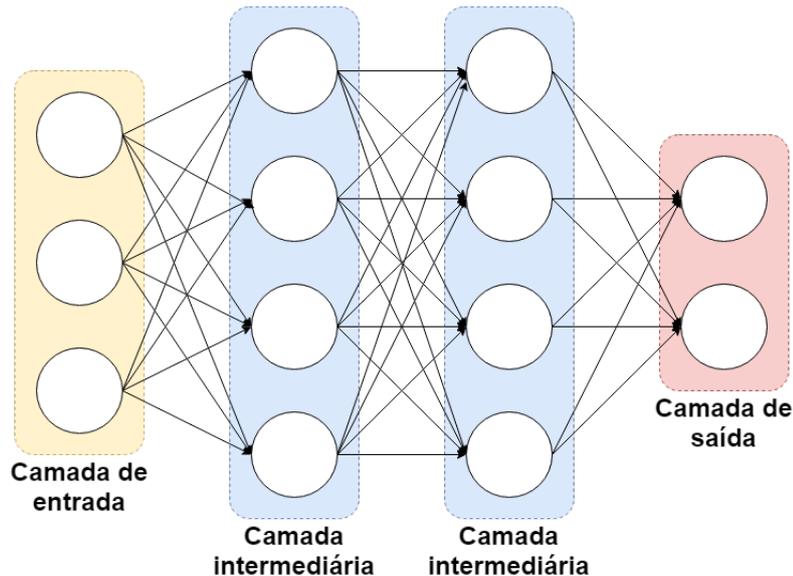


Figura 2 – Camadas de uma rede neural

Schmidhuber[14] trata as RNs como um conjunto de processadores conectados (neurônios) que produzem seqüências de ativações. Os neurônios de entrada são ativados pelo que ele descreve como “sensores que percebem o ambiente”. Neurônios das camadas seguintes são ativados pelas conexões ponderadas de neurônios que foram ativados anteriormente.

Segundo SAZLI[15], há duas categorias principais de arquiteturas de RNs: redes feed-forward e as redes recorrentes. Essas categorias se diferem na maneira como as conexões entre os neurônios podem ser feitas.

As redes feed-forward permitem que os sinais viajem apenas em um sentido: da entrada para a saída, ou seja, um sinal gerado como saída de um neurônio não afeta esse mesmo neurônio ou algum outro de uma camada anterior.

No caso das redes recorrentes (também chamadas de redes de feedback ou redes interativas), podem haver sinais viajando para neurônios de camadas anteriores ou até para o mesmo neurônio de onde saiu.

Uma das redes utilizadas no trabalho foi a ResNet. He et al.[1] citam que evidências recentes revelam que a profundidade da rede é de importância crucial e que muitas

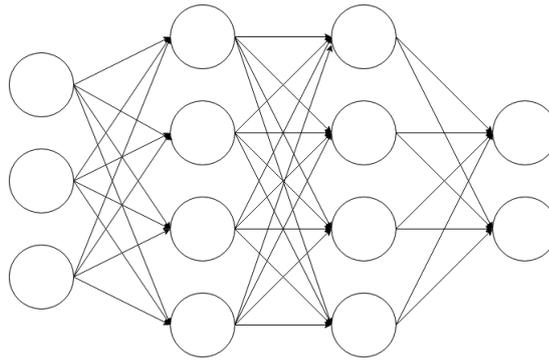


Figura 3 – Rede feed-forward

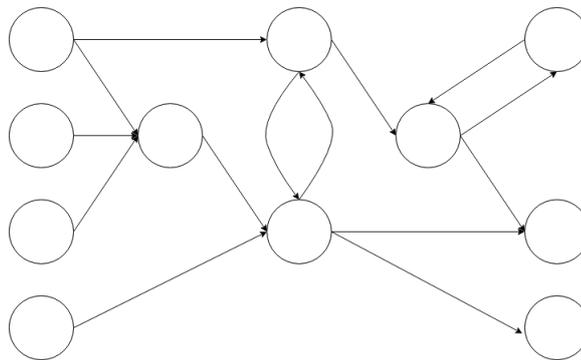


Figura 4 – Rede neural recorrente

tarefas de reconhecimento visual não triviais se beneficiaram de modelos muito profundos. Considerando esta importância, surge a seguinte questão: a capacidade de aprendizado de uma rede pode ser melhorado simplesmente adicionando mais camadas? Quanto mais profunda a rede, mais saturada fica a precisão, que acaba se degradando.

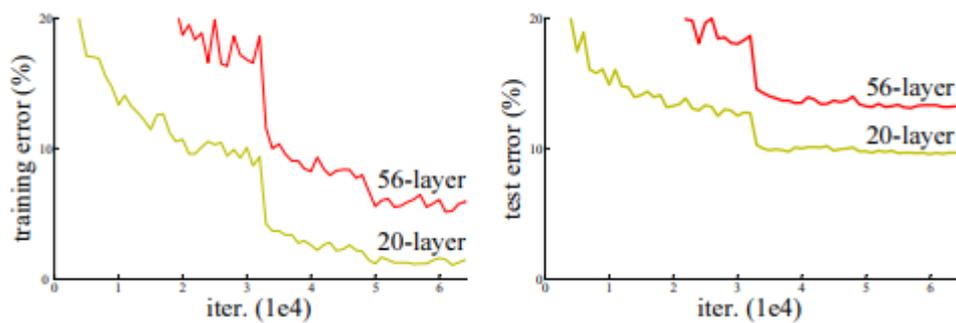


Figura 5 – Porcentagem de erro durante treino (esquerda) e teste (direita) de redes neurais com 20 e 56 camadas analisando imagens da base CIFAR-10. Quanto mais camadas, maior a porcentagem de erro. He et al.[1]

As ResNets (abreviação de Residual Networks) tentam lidar com este problema. Em geral, em uma rede neural profunda, várias camadas são empilhadas e treinadas para a tarefa em questão. A rede aprende várias características no final de suas camadas. Na aprendizagem residual, em vez de tentar aprender algumas características, utilizamos

os blocos residuais para tentarmos aprender alguns "resíduos", que podem ser entendidos como uma subtração do recurso aprendido da entrada dessa camada. Isso direciona a nova camada a aprender algo diferente do que as camadas anteriores já haviam aprendido.

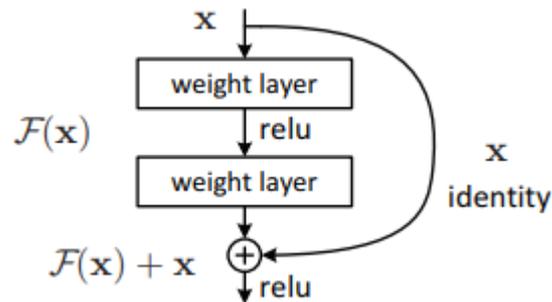


Figura 6 – Arquitetura de um bloco residual. He et al.[1]

A ResNet faz isso usando conexões de atalho (conectando diretamente a entrada da  $n$ -ésima camada a alguma camada  $n+x$ ). Treinar redes dessa forma é mais fácil do que treinar redes neurais profundas da forma convencional e resolve o problema de degradação.

O ImageNet é um banco de dados de imagens comumente utilizado como recurso para pesquisadores, educadores e estudantes para o desenvolvimento de softwares de reconhecimento de objetos. Ao aplicar as imagens disponíveis no ImageNet em 4 redes neurais (2 ResNets e 2 que não utilizam blocos residuais de 18 e 34 camadas cada, chamadas de "planas"), pôde-se constatar que as ResNets diminuíram a porcentagem de erro aumentando a quantidade de camadas, enquanto as redes "planas" sofreram o efeito contrário.

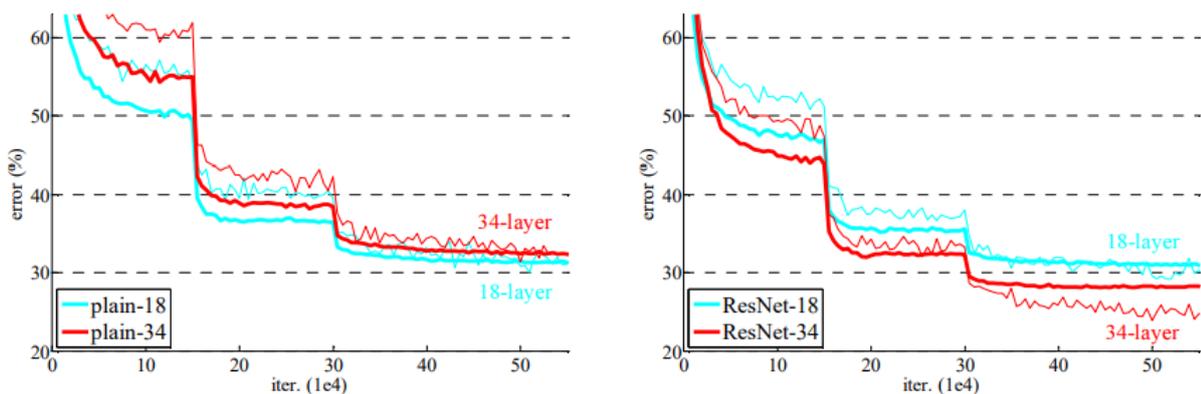


Figura 7 – Comparação dos resultados obtidos por redes que não utilizam blocos residuais (esquerda) e ResNets (direita) He et al.[1]

A outra rede utilizada foi a Xception. Como citado por Chollet[2], as redes com arquitetura Inception foram introduzidas em 2014. Desde então, as redes Inception tem sido uma das famílias de modelos com melhor desempenho no conjunto de dados ImageNet.

O building block fundamental dos modelos Inception é o módulo Inception, que possui várias versões diferentes. Um modelo Inception pode ser entendido como uma pilha de tais módulos.

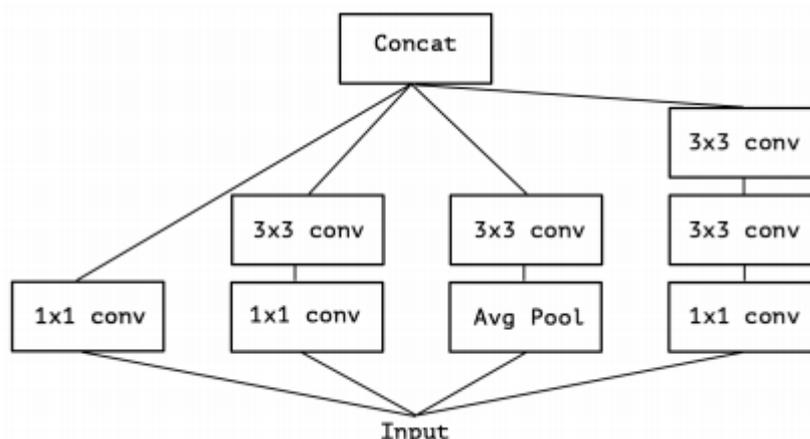


Figura 8 – Módulo Inception Chollet[2]

A inspiração para as redes Inception veio da ideia de que você precisa tomar uma decisão sobre o tipo de convolução que deseja fazer em cada camada (3x3, 5x5, etc). Com o modelo Inception, é possível usar todos eles. Isso é feito realizando cada convolução em paralelo e concatenando os resultados antes de ir para a próxima camada. O resultado de cada convolução será passado para a próxima camada.

O nome "Xception" é uma abreviação de "Extreme Inception". Sobre sua arquitetura, Chollet[2] descreve que a Xception possui 36 camadas convolucionais, que formam a base de extração de características da rede. Essas 36 camadas são separadas em 14 módulos que, com exceção do primeiro e do último, tem conexões residuais lineares. A arquitetura é separada em 3 fluxos. Primeiramente, os dados passam pelo fluxo de entrada, depois pelo fluxo médio que é repetido oito vezes e então passam pelo fluxo de saída.

#### 2.1.4 Deep Learning

De acordo com Gulli e Pal[16], o DL nada mais é do que um “subconjunto de metodologias do machine learning”, que faz utilização das RNs artificiais. A palavra “deep” refere-se à profundidade da rede, mais precisamente, das camadas intermediárias, ou seja, o DL consiste na utilização de RNs artificiais com uma quantidade considerável de camadas intermediárias para realizar o processamento das informações recebidas nas camadas de entrada. Gulli e Pal[16] lembram que não existe um número exato de camadas intermediárias necessárias na rede para definir o que é ou não DL, mas que há quem defenda que 10 camadas seriam suficientes para tal.

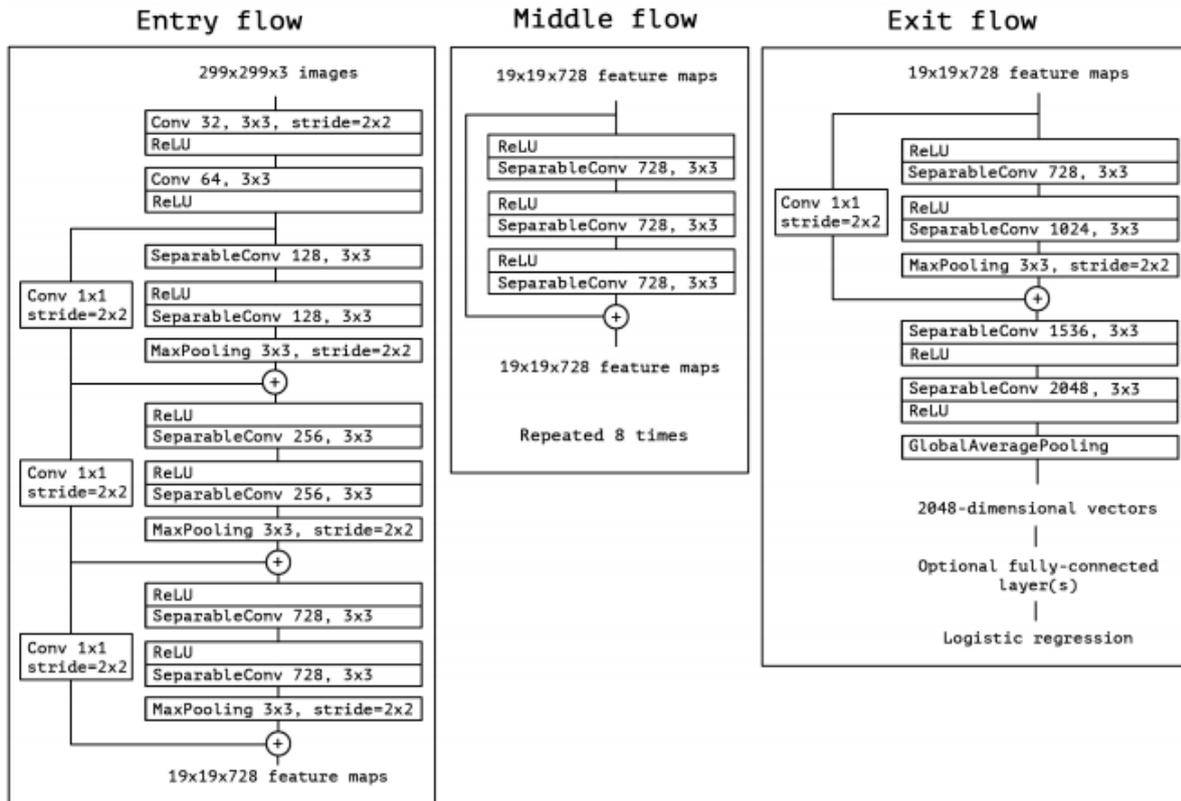


Figura 9 – Arquitetura Xception Chollet[2]

Deep learning vem sendo um assunto bastante comentado. Parte dessa popularidade pode ser creditada à imensa quantidade de dados gerados diariamente a qual podemos ter acesso.

LeCun, Bengio e Hinton[17] citam que o DL possibilitou que as soluções para problemas que a comunidade de pesquisadores de IA já vinha tentando resolver por muitos anos tivessem grandes avanços. O DL vem tendo bastante sucesso nas áreas de reconhecimento de imagens e fala e também vem se mostrando bastante eficiente na tarefa de descobrir padrões e estruturas presentes em grandes quantidades de dados, fazendo com que seja aplicável em inúmeros domínios. Um dos pontos mais surpreendentes é que o DL também vem produzindo resultados extremamente satisfatórios em várias tarefas que envolvem o entendimento de linguagem natural, como análise de sentimentos e tradução entre diferentes idiomas.

LeCun, Bengio e Hinton[17] também destacam que o sucesso da utilização do DL tende a aumentar ainda mais no futuro, visto que sua implementação não exige muito a ser feito à mão, podendo desfrutar do constante aumento de dados disponíveis e da evolução dos componentes de hardware.

Sobre os benefícios da utilização do DL, Najafabadi et al.[18] dizem que o campo é “uma avenida promissora de pesquisa na extração automatizada de representações com-

plexas de dados (características) em altos níveis de abstração” e cita que os algoritmos de DL trazem muitos benefícios quando é necessários lidar com grandes quantidades de dados.

A quantidade de áreas e diferentes tarefas onde se pode aplicar o DL é enorme. Além das predições, já mencionadas anteriormente, Hordri, Yuhaniz e Shamsuddin[19] citam algumas possibilidades como reconhecimento de fala, processamento de linguagem natural e sistemas de recomendação.

### 2.1.5 Medidores de desempenho

Como resultado do processamento do Keras, foi possível obter diferentes informações sobre o desempenho de cada rede em cada iteração. Essas informações serão mostradas em tabelas no decorrer do desenvolvimento.

A acurácia representa precisão da rede naquela determinada iteração, ou seja, a relação entre os resultados obtidos das predições feitas pela rede e os resultados esperados. A acurácia (A) se baseia na quantidade de predições corretas (C) e na quantidade total de predições realizadas (T) e é definida por:

$$A = \frac{C}{T} \quad (2.1)$$

O F1 Score é uma outra medida de precisão, que leva em conta os conceitos de precision (precisão) e recall (revocação). O precision (P) e o recall (R) são calculados baseados nas definições de negativos verdadeiros (TN), negativos falsos (FN), positivos verdadeiros (TP) e positivos falsos (FP). Sendo que:

$$P = \frac{TP}{TP + FP} \quad (2.2)$$

$$R = \frac{TP}{TP + FN} \quad (2.3)$$

Os negativos verdadeiros são os casos dos gráficos que, por exemplo, não são tendências de alta e corretamente não foram classificadas como tendências de alta. Os negativos falsos são, por exemplo, gráficos com tendência de alta que não foram classificados como tendência de alta (incorretamente). Os positivos verdadeiros são os casos em que, por exemplo, gráficos com tendência de alta foram corretamente classificados como tendência de alta. Por fim, os positivos falsos são os casos em que gráficos que, por exemplo, não apresentam tendência de alta foram classificados como tendência de alta (incorretamente). O F1 Score então é definido por:

$$F1 = 2 * \frac{P * R}{P + R} \quad (2.4)$$

A coluna "Loss" é a função de perda, que pondera a qualidade do acerto. Por exemplo, para uma amostra  $N$ , o classificador identifica que há 55% de chances de  $N$  ser  $X$  e 45% de chances de ser  $Y$ . Neste caso ele classifica como  $X$ , porém, com o loss é possível ter uma visão geral do quão eficiente está a classificação. Se, por exemplo, seja conseguido 90% de acurácia, mas um valor de loss alto, significa q seu classificador está classificando corretamente, mas com alta taxa de "dúvida" pois as probabilidades que geram a classificação não são tão certas. Porém, em casos em que, por exemplo, existam 70% de chances de  $N$  ser  $X$ , e 30% de chances de  $N$  ser  $Y$ , isto vai influenciar o loss a ter menor valor. O Loss ( $L$ ) é definido por:

$$L = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.5)$$

O "Mean Absolute Error" mede a magnitude dos erros em um conjunto de predições, não levando em consideração sua direção. É a média das diferenças entre a predição e o valor real sobre as amostras, em que todas as diferenças individuais têm peso igual. O mean absolute error (MAE) é definido por:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (2.6)$$

Onde  $n$  é o tamanho do conjunto de dados passado para a rede e  $|y_i - x_i|$  é, para cada predição, a diferença entre o resultado da predição e o valor real.

## 2.2 Mercado de Ações

Segundo Arestis, Demetriades e Luintel[20], os mercados de ações tem grande importância em todo o mundo e em decorrência disso, há inúmeras pesquisas relacionando desenvolvimento financeiro e crescimento econômico, pesquisas essas que se concentram nos impactos do desenvolvimento dos mercados de ações.

Arestis, Demetriades e Luintel[20] também citam que alguns estudos concluíram que os mercados de ações são capazes de promover crescimento a longo prazo, pois são responsáveis por encorajar a aquisição e disseminação de informações, além de ajudar na redução de custos em mobilizações de poupanças, pois facilitam e incentivam o investimento.

Investidores de sucesso não compram ou vendem ações simplesmente baseados em suas intuições, mas identificando oportunidades e sabendo tomar decisões nos momentos corretos. Outro ponto importante é o processo de aprendizado e o desenvolvimento de um plano de investimentos. Fontanills, Gentile e Cawood[4] também citam que grande parte dos entusiastas dos investimentos no mercado de ações começam da mesma forma, "Eles

leem alguns livros, abrem uma pequena conta e perdem um monte de dinheiro em pouco tempo”. Investidores de sucesso adquirem a maior quantidade de conhecimento possível e começam aos poucos.

Especializar-se em uma ou duas estratégias de negociação antes de iniciar os estudos nas inúmeras outras é fundamental, pois permite o desenvolvimento da habilidade de reconhecer condições de mercado favoráveis ou desfavoráveis e tendências, qualidade indispensável para grandes investidores.

Outro aspecto vital é definir quanto dinheiro você tem condição de perder. Antes de começar a investir, é necessário realizar uma avaliação de sua condição financeira para que seja definido quanto do dinheiro disponível pode ser investido e quanto desse dinheiro pode ser perdido, pois é imprescindível considerar os riscos.

### **2.2.1 Bolsas de Valores**

As bolsas de valores são organizações que dão às pessoas a capacidade de comprar e vender ações. As ações compõem o capital social de uma empresa, ou seja, ao comprar as ações de uma empresa, você passa a ser proprietário de uma parte dela.

Como Fontanills, Gentile e Cawood[4], as bolsas de valores são ambientes que permitem que as empresas listem suas ações e que investidores as comprem e vendam. Quando um negociante proprietário de uma ação e outro negociante, disposto a comprar esta ação, entram em um acordo de preço, a ação é negociada. As bolsas de valores fazem a intermediação dessas compras e vendas.

Com o uso de computadores, as bolsas de valores permitem a realização de leilões simultâneos, com inúmeras negociações a todo momento. A tecnologia é bastante importante nas bolsas de valores hoje em dia, pois, é por meio da utilização dela que compradores e vendedores conseguem realizar negociações quase instantaneamente, recebendo os valores ou ações de imediato.

A internet facilitou a vida das pessoas que se interessam pelo investimento no mercado de ações, possibilitando o recebimento de diferentes tipos de informação, via e-mail, websites e aplicativos específicos sobre o assunto. Todas essas informações, geradas a todo momento, podem ser utilizadas para compreender melhor as movimentações das bolsas de valores.

### **2.2.2 Teoria de Dow**

A Teoria de Dow foi desenvolvida com base nos trabalhos de Charles Henry Dow, um jornalista nascido nos Estados Unidos no ano de 1851, fundador da Dow Jones & Company e primeiro editor do *The Wall Street Journal*, importante jornal focado em economia.

A Teoria de Dow é um conjunto de princípios que podem ser utilizados para realizar análises do comportamento dos mercados. Como lembrado por Nicholson[21], a teoria foi baseada em editoriais escritos por Dow para o *The Wall Street Journal*, nos quais Dow expressava suas ideias sobre o comportamento do mercado de ações. Para Dow, o andamento do mercado de ações era uma forma de analisar as condições gerais de negócios na economia. Dow também acreditava que com esta análise, era possível identificar tendências de mercado.

Muito do que é estudado sobre análise técnica hoje tem os princípios apresentados na Teoria de Dow como base.

### 2.2.3 Análise Técnica

Segundo Nicholson[21] e Chin et al.[3], a análise técnica teve sua origem baseada na teoria de Dow. A análise técnica é um estudo que tem como objetivo compreender a movimentação do mercado fazendo análise numérica e de gráficos se baseando em preços passados e dados de volume de negociação.

Os sinais indicados por uma análise técnica podem auxiliar os negociantes em relação ao timing das operações, ou seja, ajudam a saber a hora certa de comprar ou vender uma ação.

Chin et al.[3] também ressaltam as evidências positivas que comprovam a eficácia da análise técnica, além de descobertas recentes de finanças comportamentais que ajudaram a validar a irracionalidade dos investidores na tomada de decisão, agregando mais valor à análise técnica.

Nos gráficos candlestick, cada candle é formado pelos preços de abertura, alta, baixa e fechamento e pode representar a movimentação de um dia, hora, etc. Há dois tipos de candles, os candles de alta e os candles de baixa, que são representados por cores diferentes. O candle de alta é aquele em que o preço de fechamento é maior que o preço de abertura, já o candle de baixa é o que o preço de fechamento é menor que o preço de abertura.

Cada gráfico tem sua periodicidade. Em um gráfico em que cada candle representa um hora por exemplo, o valor "Baixa" indica o menor valor pelo qual aquela ação foi negociada dentro daquele período de uma hora e o valor "Alta" representa o maior valor pelo qual aquela ação foi negociada. Da mesma forma, o valor "Abertura" indica o primeiro valor pelo qual a ação foi negociada e o valor "Fechamento" indica o valor da última negociação daquele período de tempo.

Ao analisar gráficos candlestick, procura-se identificar sinais de reversão, ou seja, padrões construídos a partir de uma sequência de candles, que indicam que há probabilidade de alterações de tendência.

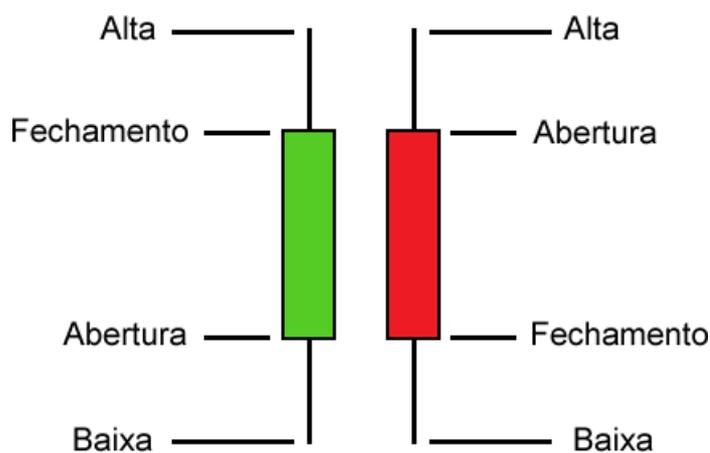


Figura 10 – Candle de alta (esquerda) e candle de baixa (direita)

Alguns exemplos de sinais de reversão são o Bullish Engulfing e o Bearish Engulfing. O Bullish Engulfing é um sinal que vem após uma tendência de baixa e indica uma reversão para tendência de alta. É identificado por um pequeno candle de baixa seguido por um candle de alta que "engole" completamente o corpo do candle anterior. O Bearish Engulfing é o contrário, pois é um sinal que aparece depois de uma tendência de alta e indica uma reversão para tendência de baixa. O Bearish Engulfing pode ser identificado por um pequeno candle de alta seguido de um candle de baixa que também "engole" completamente o corpo do candle anterior.

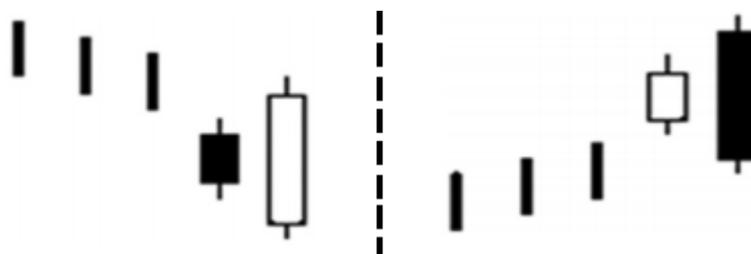


Figura 11 – Engolfo de alta (esquerda) e Engolfo de baixa (direita). Chin et al.[3]

Os sinais de reversão Bullish são aqueles que aparecem após uma tendência de baixa e indicam um bom momento para compra, uma vez que antecedem uma tendência de alta. Já os sinais Bearish, que aparecem após tendências de alta, indicam bons momentos para venda, pois antecedem tendências de baixa.

## 2.3 Trabalhos relacionados

Visando analisar algumas das diferentes formas de realizar previsões no mercado de ações, esta seção apresentará alguns trabalhos relacionados ao tema, apontando as semelhanças e diferenças em relação ao que foi desenvolvido neste trabalho e quais foram os resultados obtidos.

No trabalho de Egeli, Ozturan e Badur[22], o objetivo foi realizar a predição da ações da Istanbul Stock Exchange (ISE), passando como valores de entrada do sistema o valor do índice do dia anterior, a taxa de câmbio Lira Turca/Dólar americano do dia anterior, a taxa de juros overnight do dia anterior e 5 variáveis para representar os dias úteis da semana. Os dados utilizados para treinamento e testes foram recolhidos dados de 1 de julho de 2001 a 28 de fevereiro de 2003 do Banco Central da República da Turquia, sendo que 90% desses dados foram utilizados para treino e 10% para testes. Os dados serviram de entradas para redes com arquiteturas Perceptron Multicamadas e Feed Forward generalizadas com que diferentes números de camadas ocultas (1, 2 e 4) com um valor médio de erro quadrático de 0,003. Após o fim do processamento por parte das redes, seus resultados foram comparados com os resultados das médias móveis para períodos de 5 e 10 dias. Após o fim dos testes, os autores concluíram que os modelos de previsão baseados em redes neurais foram mais precisos que os baseados nas médias móveis, sendo que entre os modelos de redes neurais utilizados, a Feed Forward generalizada foi mais adequada para a predição.

Para o trabalho de Zheng e Jin[23], 82 ações negociadas na New York Stock Exchange (NYSE) foram selecionadas aleatoriamente utilizando a API da Alpha Vantage. Foram capturados dados diários da variação dos preços, como preço mais alto, mais baixo, de abertura e fechamento, como os utilizados neste trabalho, além do volume diário de negociações de cada ação. O objetivo do trabalho é prever se o preço de uma determinada ação vai aumentar ou diminuir nos próximos  $n$  dias. Para isso, foram implementados os métodos regressão logística, rede Bayesiana, rede neural simples, e SVM com kernel rbf, que foram executados nos preços de uma ação específica. Os resultados foram que os modelos estavam próximos de adivinhar aleatoriamente, com taxas de erro em torno de 40% a 50%, concluindo que apenas preços passados não são suficientes. Foram então utilizados 13 indicadores técnicos juntamente com os preços passados. Utilizando tanto dados passados quanto indicadores técnicos, foi possível alcançar até 70% de precisão.

O trabalho de Tiong, Ngo e Lee[24] se assemelha a este em alguns pontos, principalmente pelo fato de também levar em consideração os gráficos candlestick, tendo também uma seção introduzindo a interpretação de um candle e as redes neurais artificiais. Os dados utilizados foram os preços das ações da Dell Inc e da taxa de câmbio Euro/Dólar americano do ano de 2011, sendo que 70% dos dados foram utilizados para treino e 30% para teste. Esses dados foram conseguidos nos serviços Yahoo Finance e MarketWatch, entre eles, constam os valores como preço mais alto, mais baixo, de abertura e fechamento, além do volume negociações. As características de cada candle foram formalizados em valores numéricos e nominais, expressos em unidade relativa. Com a utilização do WEKA, os dados foram aplicados a uma rede neural Perceptron multicamada e a uma otimização mínima sequencial como Máquina de Vetores de Suporte (SVM). Para os resultados dos experimentos, foram utilizados o Mean Absolute Error e Root Mean Squared Error para

avaliar o desempenho dos modelos implementados. Os dados por hora para representar os gráficos melhoram a rede neural e o SVM no aprendizado dos padrões de candles. Para a validação, 100 novos conjuntos de dados para a Dell Inc e EUR/USD foram testados. Foi possível classificar corretamente a posição dos padrões de candles com até 64% de acerto usando dados diários e por horas. Nos diversos experimentos com 100 amostras, tanto das ações da Dell quanto da taxa de câmbio, a rede neural e o SVM alternaram no sentido de atingir maior precisão, chegando a ter, no máximo, 2% de diferença entre eles.

No caso da pesquisa de Kar[25], foram implementadas diferentes funções de ativação e o algoritmo foi testado no banco de dados do índice de ações Nifty, onde a predição foi feita com base nos valores dos últimos  $n$  dias. As funções de ativação consideradas foram a sigmóide unipolar e bipolar, a tangente hiperbólica e a função de base radial. Foram capturados os dados referentes aos candles, como abertura, fechamento, maior e menor preço. Esses dados foram separados em 60% para treino e 40% para teste, sendo que dos 60% para treino, 20% foi utilizado para validação. Em 30 iterações, houve diminuição considerável de erro entre as 3 primeiras, depois o erro continuou diminuindo, mas em menor velocidade. Na média, a porcentagem de acurácia ficou em 88%, mas em seu melhor caso, chegou a alcançar a taxa de 96%. Entre as funções de ativação citadas, a sigmóide bipolar teve a maior taxa de erro tanto em treino quanto em teste. Durante o treino, a sigmóide unipolar e a tangente hiperbólica obtiveram os melhores resultados, mas em testes, a função de base radial teve os melhores números.

A utilização somente dos dados dos candles não apresentou vantagens em todos os casos como foi possível constatar nesta seção, pois somente uma das pesquisas citadas conseguiu resultados melhores, sendo 1% melhor na média, enquanto as outras chegaram a apresentar resultados até 20% mais baixos.



## 3 DESENVOLVIMENTO

Nesta seção, serão descritos os passos realizados para desenvolver as etapas do trabalho, iniciando pela coleta das informações sobre ações de diversas empresas e geração dos gráficos a partir das informações coletadas.

### 3.1 Coleta de dados, geração e classificação dos gráficos

Para formar as bases de treinamento, validação e teste, foi preciso coletar dados sobre os candles e gerar os gráficos a partir desses dados. Para realizar a coleta dos dados que foram utilizados para a composição dos gráficos, foi utilizada a API da IEX Group, que foi acessada por um script na linguagem de programação Python. Após o recolhimento dos dados, estes foram salvos em um banco de dados SQLite e utilizados para a construção dos gráficos com a biblioteca Bokeh.

A API utilizada fornece diversos endpoints, utilizados para fazer a requisição de diferentes conjuntos de dados. O primeiro passo foi saber quais empresas tinham os dados sobre suas ações disponíveis. Para isso a API fornece um endpoint específico (`/ref-data/symbols`), que retorna em formato JSON, informações sobre todas as empresas das quais é possível obter os preços das ações. O identificador de cada empresa, disponível no campo "symbol" do arquivo JSON, foi salvo em um arquivo de texto chamado "symbols.txt".

```
{
    "symbol": "A",
    "name": "Agilent Technologies Inc.",
    "date": "2018-09-21",
    "isEnabled": true,
    "type": "cs",
    "iexId": "2"
}
```

Figura 12 – Informações obtidas sobre a Agilent Technologies Inc. no endpoint `/ref-data/symbols`

Com os identificadores das empresas em mãos, começou a extração dos dados de cada empresa. Para isso, em outro script, o arquivo "symbols.txt" foi lido e o código de cada empresa foi colocado em uma lista. Percorrendo a lista, para cada empresa, foi feita uma requisição para outro endpoint da API (`/stock/IDENTIFICADOR/time-series`), que

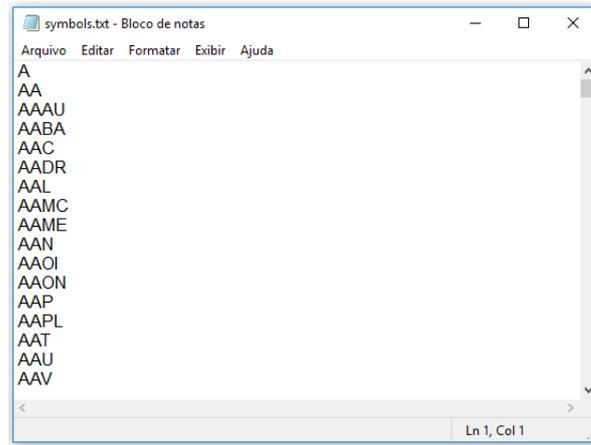


Figura 13 – Arquivo de texto com os identificadores das empresas

retorna os dados disponíveis sobre as ações daquela empresa no último mês em formato JSON. O retorno vem em forma de uma lista onde cada item corresponde a um candle de um dia, com informações como data, valores de abertura e de fechamento e valores de preço mais alto e mais baixo. As informações de cada candle foram salvas em um banco de dados SQLite.

```
{
  "date": "2018-08-21",
  "open": 64.74,
  "high": 65.4,
  "low": 64.62,
  "close": 64.93,
  "volume": 2982791,
  "unadjustedVolume": 2982791,
  "change": 0.46,
  "changePercent": 0.714,
  "vwap": 65.0419,
  "label": "Aug 21",
  "changeOverTime": 0
}
```

Figura 14 – Informações para compor um dos candles do gráfico da Agilent Technologies Inc. obtidas no endpoint `/stock/A/time-series`

Para controlar a geração dos gráficos, que era demorada e deixava o computador lento depois de um tempo, todos os diferentes símbolos contidos no banco de dados foram colocados (por um outro script) em um outro arquivo de texto chamado "va-

symbol	date	high	low	open	close
Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
A	2018-08-10	66.87	65.93	66.82	66.26
A	2018-08-13	66.99	65.67	66.44	65.94
A	2018-08-14	67.26	66.05	66.05	66.75
A	2018-08-15	66.39	64.22	65.94	64.73
A	2018-08-16	65.67	64.81	65.04	65.5
A	2018-08-17	65.48	64.28	65.48	64.66
A	2018-08-20	65.11	64.46	64.57	64.47

Figura 15 – Informações sobre alguns candles utilizados para formar o gráfico da Agilent Technologies Inc. salvos no banco de dados

lid\_symbols.txt". Esse arquivo é semelhante ao "symbols.txt", mas com menos símbolos, visto que, na API consultada, não haviam dados disponíveis para todos os símbolos. Todas as imagens geradas são nomeadas com o símbolo da empresa que representam, seguido da extensão ".png", logo, quando a geração dos gráficos deixava o computador lento, o script era parado, todas as linhas do arquivo "valid\_symbols.txt" cujos gráficos já haviam sido gerados eram apagadas e o script começava a gerar os gráficos baseado nos símbolos restantes do arquivo "valid\_symbols.txt".

Para a geração dos gráficos, foi utilizada a biblioteca Bokeh, que pode ser utilizada para a construção de diversos outros tipos de gráficos além do candlestick. Para isso, em outro script, o arquivo "valid\_symbols.txt" era lido e então, para cada linha, o banco de dados era consultado para conseguir as informações dos candles de cada empresa. Recolhendo os candles, os dados eram passados a biblioteca Bokeh, que fez o trabalho de gerar os gráficos e salvar as imagens geradas.

No total, foram gerados 7.145 gráficos, que classificados, resultaram em 2.955 gráficos com tendência de alta, 2.258 com tendência de baixa e 1.932 com tendência de consolidação.

Os gráficos foram então separados em três bases diferentes, sendo uma para treino das redes, uma para teste e uma para validação. Na base de treino, foram colocados 1.906 gráficos com tendência de alta, 1.423 com tendência de baixa e 1.238 com tendência de consolidação. A base de teste foi formada com 516 gráficos com tendência de alta, 416 com tendência de baixa e 386 com tendência de consolidação. Já a base de validação ficou com 533 gráficos com tendência de alta, 419 com tendência de baixa e 308 com tendência de consolidação.



Figura 16 – Gráfico gerado pelo Bokeh com os dados da Agilent Technologies Inc. contidas no banco de dados

### 3.2 Resultados da ResNet

Na ResNet, a princípio, as bases de treino e validação foram aplicadas somente às camadas totalmente conectadas durante 10 iterações, em um modelo pré treinado na base de imagens ImageNet. Os resultados podem ser observados nas tabelas 1 e 2.

Tabela 1 – Resultados das iterações nas camadas totalmente conectadas da ResNet com a base de treino

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
1	0,5428886439	nan	0,9296979863	0,3717118473
2	0,7179713341	0,7043754376	0,6482043794	0,2512894247
3	0,7477398014	0,7417342419	0,5802652331	0,2209467203
4	0,7625137817	0,7579624469	0,5455309509	0,2075770239
5	0,7819184124	0,7780137132	0,502276908	0,1915622576
6	0,8101433295	0,8052983401	0,4469598382	0,1751153519
7	0,8264608599	0,8231145627	0,411313832	0,1642781691
8	0,8518191841	0,8482069135	0,368404991	0,1510344915
9	0,8646085998	0,8631094418	0,331165828	0,140133676
10	0,8524807055	0,8505337109	0,3563320729	0,1440420423

Pôde-se observar que, apesar de a rede ter conseguido aumentar sua precisão de 54% para 85% na base de treino, além de também ter conseguido chegar à marca de 85% de F1 Score e atingir valores muito baixos de Loss e Mean Absolute Error, os valores resultantes do processamento feito com a base de validação não foram bons, ficando durante todas as iterações com 42% de acurácia e F1 Score, enquanto o valor de Loss aumentou em todas as iterações. Ou seja, nessas 10 iterações, a rede atingiu os 42% e a partir daí, foi incapaz de continuar aprendendo.

Tabela 2 – Resultados das iterações nas camadas totalmente conectadas da ResNet com a base de validação

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
1	0,4206730769	0,4206730769	1,2078634913	0,4123760118
2	0,4206730769	0,4206730769	1,4106301895	0,4041997072
3	0,4206730769	0,4206730769	1,6910496033	0,3954186867
4	0,4206730769	0,4206730769	2,5943999627	0,3898793933
5	0,4206730769	0,4206730769	2,7889084999	0,389554697
6	0,4206730769	0,4206730769	3,2254668321	0,3879540784
7	0,4206730769	0,4206730769	3,6440381759	0,3869468318
8	0,4206730769	0,4206730769	3,7819044407	0,3867671001
9	0,4206730769	0,4206730769	3,8590545226	0,3866835481
10	0,4206730769	0,4206730769	4,5807613409	0,3863458076

A primeira iteração dos resultados com a base de validação foi selecionada como o melhor resultado, uma vez que, como pode ser visto na tabela 2, apesar de a acurácia e o F1 Score não terem se desenvolvido no decorrer das iterações, esta possuía um valor de Loss muito menor do que as iterações finais. Os pesos das conexões entre os neurônios das camadas totalmente conectadas foram salvos.

Tabela 3 – Melhor resultado obtido em 10 iterações na ResNet na base de validação

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
1	0,4206730769	0,4206730769	1,2078634913	0,4123760118

Iniciou-se então um novo processamento, desta vez com 30 iterações, cujos resultados constam nas tabelas 4 e 5, porém, agora envolvendo toda a rede e utilizando nas camadas totalmente conectadas, os pesos salvos do melhor resultado das 10 primeiras iterações, mostrado na tabela 3.

Novamente, para cada iteração, além dos resultados para a base de treino, o Keras também gerou os resultados da rede quando submetida às amostras da base de validação. Assim como o que foi feito após o processamento com as 10 primeiras iterações, os resultados obtidos com a base de validação são aqueles entre os quais se procura a iteração com o melhor resultado. Foi selecionada a iteração 16, que tem suas informações mostradas na tabela 6.

A partir daqui o treinamento da rede acaba e a iteração com a melhor pontuação F1 Score (tabela 6) é salva para ser aplicada à base de teste.

Depois de selecionada, a iteração com o melhor F1 Score recebeu as amostras da base de teste para que fosse possível medir seu desempenho. Os resultados finais do teste realizado com a ResNet podem ser conferidos na tabela 7.

Tabela 4 – Resultados das 30 iterações de processamento realizadas em toda a rede ResNet com a base de treino

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
1	0,7764057331	0,7653862277	0,5323321436	0,2121476498
2	0,8487320838	0,8479624482	0,3805304793	0,1443649893
3	0,8650496142	0,8651852757	0,3443359451	0,1269095593
4	0,8829106947	0,882438221	0,3103629803	0,111579425
5	0,8983461964	0,8983100623	0,263833398	0,0958765786
6	0,9492833518	0,949104731	0,1461845448	0,052325515
7	0,9828004411	0,9826639359	0,059757017	0,0230741071
8	0,9955898567	0,995477853	0,0202646977	0,0077645169
9	0,9980154355	0,9982324426	0,011210486	0,0042307064
10	0,9986795775	0,9986795775	0,0095679082	0,0033953654
11	0,9995581087	0,9994458824	0,0067563593	0,0020258875
12	0,9790518192	0,9792338257	0,0691733994	0,0178332422
13	0,9243660419	0,9237402372	0,1953438018	0,0712746243
14	0,9746416759	0,9751526906	0,0681747773	0,0256255206
15	0,9911797133	0,9911657121	0,0297244156	0,0108285678
16	0,9991179713	0,9991179713	0,0098644549	0,0037540644
17	0,9993384785	0,9993384785	0,0068011523	0,0021369045
18	0,9995589857	0,9995589857	0,0056998951	0,0014643743
19	0,9995589857	0,9995589857	0,0054989512	0,0013000194
20	0,9980154355	0,9980154355	0,0085840991	0,0022867736
21	0,9565600883	0,9567070904	0,1172565518	0,0392607358
22	0,9841549296	0,983686843	0,0486676993	0,017407419
23	0,9951391958	0,9951391958	0,0223119706	0,0058405562
24	0,9984564498	0,9984564498	0,0080015825	0,002321599
25	0,9997794928	0,9997794928	0,0055583553	0,0013883665
26	0,9995589857	0,9995589857	0,0054666308	0,0012837384
27	0,9997794928	0,9997794928	0,0048177345	0,0009330191
28	0,9997799296	0,9997799296	0,0041713565	0,0005498988
29	0,992487848	0,992487848	0,0205418024	0,0069340242
30	0,9799735915	0,9800574263	0,0544125965	0,0175766906

### 3.3 Resultados da Xception

A forma de conseguir os resultados com a Xception foi a mesma utilizada com a ResNet. De início, as bases de treino e validação foram aplicadas somente às camadas totalmente conectadas durante 10 iterações, em um modelo pré treinado. Os resultados são mostrados nas tabelas 8 e 9.

Após analisar os resultados das primeiras 10 iterações na base de validação, foi notado que os resultados da pontuação F1 Score não se estabilizou e continuou subindo nas últimas iterações, por isso, foram feitas mais 10 iterações partindo da 10<sup>a</sup>, cujos resultados constam nas tabelas 10 e 11.

Tabela 5 – Resultados das 30 iterações de processamento realizadas em toda a rede Res-Net com a base de validação

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
1	0,4286858974	0,4299906553	1,5176745378	0,3911644878
2	0,7580128205	0,7275868226	0,5787747732	0,2587555154
3	0,5913461538	0,5834594698	0,8354400106	0,3230459751
4	0,8084935897	0,7894200683	0,4381606583	0,1781606332
5	0,7195512821	0,7191914626	0,668182911	0,2125982669
6	0,7756410256	0,7747248656	0,7055451411	0,1673154468
7	0,8413461538	0,8413461538	0,6205696473	0,1107815416
8	0,8605769231	0,8608821737	0,5850128226	0,0999921677
9	0,8533653846	0,8537087914	0,6288809608	0,0988451526
10	0,8501602564	0,8504655071	0,6416355757	0,0989612617
11	0,8565705128	0,8565705128	0,7121962496	0,0967909973
12	0,3878205128	0,380618939	1,2374811371	0,3944025819
13	0,5544871795	0,5547739649	1,8212827979	0,3003681868
14	0,8076923077	0,8059838934	0,5313174656	0,1486253826
15	0,8581730769	0,8581730769	0,5643460939	0,0979664096
<b>16</b>	<b>0,8637820513</b>	<b>0,8636294298</b>	<b>0,6406282177</b>	<b>0,0960193658</b>
17	0,8565705128	0,8568884837	0,6732389963	0,0968721732
18	0,8549679487	0,8549679487	0,7104577155	0,0982324504
19	0,8629807692	0,8633114589	0,7904925316	0,0954930612
20	0,8213141026	0,8202330119	0,7227925983	0,1266970145
21	0,6842948718	0,6835740056	1,0305985625	0,2153131286
22	0,8461538462	0,8468533785	0,6681961471	0,1057714779
23	0,8573717949	0,8572573264	0,7066020016	0,097742636
24	0,8605769231	0,8612637367	0,6699738064	0,0954943387
25	0,859775641	0,8593432017	0,6943621764	0,0951732441
26	0,8573717949	0,8572827623	0,7300603017	0,0951820803
27	0,8605769231	0,8609203299	0,7730045248	0,0955161697
28	0,859775641	0,859775641	0,8248918991	0,0962969779
29	0,6858974359	0,6838878745	1,1014404083	0,2197554807
30	0,7996794872	0,7996794872	0,8744228429	0,1386049445

Tabela 6 – Iteração que resultou na rede ResNet com o maior F1 Score durante o treinamento

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
16	0,8637820513	0,8636294298	0,6406282177	0,0960193658

Assim como foi feito com a ResNet, uma das iterações foi selecionada como a que obteve o melhor resultados na pontuação F1 Score. Então, foram salvos os pesos atribuídos às conexões das camadas na 18ª iteração com a base de validação, como mostrado na tabela 12.

Seguindo os mesmos passos, foram processadas então 30 iterações com toda a rede e utilizando nas camadas totalmente conectadas os pesos salvos do melhor resultado das

Tabela 7 – Resultados finais da ResNet, quando submetida às amostras da base de teste

<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
0,8740515933	0,8741961147	0,6041708158	0,0903599588

Tabela 8 – Resultados das 10 primeiras iterações nas camadas totalmente conectadas da Xception com a base de treino

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
1	0,4350606395	nan	1,0606697152	0,4283710333
2	0,5609702315	0,4124899411	0,9294090027	0,3806300121
3	0,6456449834	0,5996796909	0,7932040188	0,3156494201
4	0,6674751928	0,6351055078	0,7526464273	0,2921623981
5	0,6904079383	0,6700548528	0,6989689066	0,2715748618
6	0,7091510475	0,6921375489	0,6461118654	0,2565745589
7	0,7369349505	0,7250765291	0,6081322953	0,2402725785
8	0,7614112459	0,7435519537	0,5704447367	0,2308240942
9	0,7594266814	0,7458507431	0,5630053462	0,2288830421
10	0,7620727674	0,7509130476	0,566208147	0,2267139541

Tabela 9 – Resultados das 10 primeiras iterações nas camadas totalmente conectadas da Xception com a base de validação

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
1	0,4463141026	nan	1,064650126	0,4344303302
2	0,3605769231	nan	1,2224027316	0,4470455784
3	0,3477564103	0,3215002551	1,1794707286	0,4297249653
4	0,3565705128	0,241996734	1,2568990512	0,4443389399
5	0,4006410256	0,2694083701	1,1750325392	0,431306651
6	0,4439102564	0,324043972	1,0967936011	0,4164754924
7	0,40625	0,326562835	1,1707530495	0,4203907939
8	0,3830128205	0,3380261534	1,2845221758	0,4245526707
9	0,4102564103	0,352646705	1,1992662442	0,4157248499
10	0,4270833333	0,3596651848	1,1672933331	0,4110765067

20 primeiras iterações. Os resultados das 30 iterações com as bases de treino e validação podem ser observados nas tabelas 13 e 14, respectivamente.

Mais uma vez, foi entre os resultados conseguidos com a base de validação que buscamos a iteração com o melhor F1 Score, para salvarmos os pesos da rede e utilizarmos na base de teste. O melhor resultado obtido foi na 22<sup>a</sup> iteração, cujos resultados são mostrados pela tabela 15.

Por fim, aplicando na rede mostrada na tabela 15 as amostras da base de teste, foi possível obter os resultados finais de desempenho da Xception, que constam na tabela 16.

Tabela 10 – Resultados das outras 10 iterações nas camadas totalmente conectadas da Xception com a base de treino

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
11	0,7649393604	0,757808994	0,5406039596	0,2216550374
12	0,770893054	0,7610171225	0,5394110797	0,2184802348
13	0,7647188534	0,7564761429	0,5651438873	0,2226352231
14	0,7461962513	0,7348664666	0,5914461188	0,2299281492
15	0,7472987871	0,7349096144	0,5936240546	0,2310354227
16	0,7711135612	0,767186035	0,5286295815	0,2105102112
17	0,8011025357	0,7941701306	0,4817700196	0,1963865508
18	0,8291069458	0,8235317688	0,4301161537	0,1819085669
19	0,8359426681	0,8287899945	0,4095542863	0,1752165474
20	0,8255788312	0,8205823358	0,4236598169	0,1761408624

Tabela 11 – Resultados das outras 10 iterações nas camadas totalmente conectadas da Xception com a base de validação

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
11	0,3854166667	0,3534090897	1,309704212	0,4193416245
12	0,3854166667	0,3600496825	1,323459448	0,4145340981
13	0,3549679487	0,3502443131	1,6627255036	0,4229852045
14	0,3485576923	0,3352843003	1,6279752713	0,426581868
15	0,3533653846	0,3431180872	1,4643018735	0,4224796952
16	0,3741987179	0,3521558528	1,4808239906	0,4187491613
17	0,3982371795	0,3701435114	1,4207897889	0,4068335608
18	0,3822115385	0,3723716247	1,5148025659	0,4092978033
19	0,3669871795	0,3613960907	1,663878392	0,4169151202
20	0,3709935897	0,3659117394	1,6425352097	0,4138802366

Tabela 12 – Melhor resultado obtido em 20 iterações na Xception

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
18	0,3822115385	0,3723716247	1,5148025659	0,4092978033

Tabela 13 – Resultados das 30 iterações de processamento realizadas em toda a rede Xception com a base de treino

<b>Iteração</b>	<b>Acurácia</b>	<b>F1 Score</b>	<b>Loss</b>	<b>Mean Absolute Error</b>
1	0,8145462536	0,8106198786	0,4690708941	0,1818023259
2	0,8330037355	0,8319462984	0,4114373815	0,1561362667
3	0,8811250275	0,8804530699	0,324228782	0,1228680217
4	0,8925510877	0,8923441108	0,283735473	0,1056295565
5	0,9195781147	0,919499671	0,2010062793	0,0761140881
6	0,9661612832	0,9658635809	0,0937666789	0,0352129314
7	0,9859371567	0,986121448	0,0400132365	0,0164963647
8	0,9967040211	0,9967040211	0,0130638633	0,0055858336
9	0,9991210723	0,9991210723	0,0052946533	0,0029855544
10	0,9975829488	0,9975829488	0,0083263021	0,0035816799
11	0,9887936717	0,9888929054	0,027541007	0,0105467996
12	0,9582509339	0,957967408	0,1195669942	0,0417472297
13	0,9646231597	0,9640419329	0,0968702643	0,0349961967
14	0,9841793013	0,9843706803	0,042056492	0,0166402349
15	0,9960448253	0,9959314152	0,0112458979	0,0050573714
16	0,9989013404	0,9987879302	0,0058792475	0,0026235999
17	0,9993408042	0,9993408042	0,0028920263	0,0015725051
18	1	1	0,001229874	0,0007635837
19	1	1	0,0009605463	0,0006025218
20	1	1	0,0005563663	0,0003569749
21	1	1	0,0004954778	0,0003086754
22	0,9993408042	0,9993408042	0,0021640339	0,0007855091
23	1	1	0,0006183817	0,0003865805
24	0,9997802681	0,9997802681	0,001079279	0,0004110099
25	0,9997802681	0,9997802681	0,0008655987	0,0004876335
26	0,9989013404	0,9989013404	0,0022747648	0,0008451836
27	0,9993408042	0,9993408042	0,0020287239	0,0010786627
28	0,9852779609	0,9852566961	0,0407456252	0,0122256698
29	0,9793451989	0,9793310224	0,0587204634	0,0206520172
30	0,9905515271	0,9905515271	0,0270701718	0,0103853621

Tabela 14 – Resultados das 30 iterações de processamento realizadas em toda a rede Xception com a base de validação

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
1	0,7852564103	0,7869658084	0,5068720203	0,1778537333
2	0,8213141026	0,8233715815	0,4140550773	0,1639352015
3	0,8461538462	0,8447322142	0,4131896911	0,1286894576
4	0,7828525641	0,7809518071	0,489314307	0,1624336835
5	0,8477564103	0,8481872036	0,4282655408	0,1232073931
6	0,8309294872	0,8319633985	0,5602642177	0,1187314453
7	0,8389423077	0,8379083948	0,6626692965	0,1117679999
8	0,8541666667	0,8545285356	0,6066541572	0,1006657884
9	0,8573717949	0,8577078161	0,6346143854	0,1007226227
10	0,8445512821	0,8435690648	0,7158744268	0,1064882215
11	0,8165064103	0,8178763435	0,8223722982	0,1274326195
12	0,8036858974	0,8043527702	0,6085068976	0,1400284955
13	0,8357371795	0,8357888743	0,6059627511	0,1178472689
14	0,8485576923	0,8485576923	0,6731363765	0,1062094965
15	0,8477564103	0,8477564103	0,7788351449	0,1033423671
16	0,8517628205	0,8511424729	0,7481807613	0,1005919158
17	0,8549679487	0,8542442093	0,7638751466	0,1001842949
18	0,8525641026	0,8520988417	0,8054106129	0,1008242802
19	0,8533653846	0,85140095	0,8891350259	0,101172017
20	0,8509615385	0,8500310167	0,9537649898	0,1023756021
21	0,8517628205	0,8520729947	0,9870515176	0,1005150652
<b>22</b>	<b>0,8581730769</b>	<b>0,8588709671</b>	<b>0,959987961</b>	<b>0,0969898031</b>
23	0,858974359	0,8588192723	0,9520434951	0,0970060142
24	0,8565705128	0,8568548384	0,9706321173	0,0969527516
25	0,8581730769	0,8581730769	0,9592943913	0,0961926772
26	0,8549679487	0,8549679487	0,9505002506	0,0985297213
27	0,8549679487	0,8549679487	0,9505394934	0,0995296685
28	0,8012820513	0,8007650948	0,9283732068	0,1360108923
29	0,8245192308	0,8235887098	0,6766927682	0,1244035934
30	0,8581730769	0,858043838	0,7204481578	0,1010809519

Tabela 15 – Iteração que resultou na rede Xception com o maior F1 Score durante o treinamento

Iteração	Acurácia	F1 Score	Loss	Mean Absolute Error
22	0,8581730769	0,8588709671	0,959987961	0,0969898031

Tabela 16 – Resultados finais da Xception, quando submetida às amostras da base de teste

Acurácia	F1 Score	Loss	Mean Absolute Error
0,8634294386	0,8628909881	0,9330935813	0,0938368157



## 4 CONCLUSÃO

Em relação à fase de treino, nas 10 primeiras iterações, a ResNet parecia estar evoluindo seu desempenho, chegando à marca de 86%, tanto de acurácia quanto de F1 Score, além de conseguir marcas baixas de Loss e Mean Absolute Error, porém, quando recebia como entrada as amostras da base de validação, todos os valores de acurácia e F1 Score foram menores que todas as iterações do treino, se estabilizando todo o tempo em 42% e apresentando valores mais altos de Loss e Mean Absolute Error.

Na segunda fase do treinamento, com 30 iterações em toda a rede, a ResNet apresentou ótimos resultados, chegando a 99% de acurácia e F1 Score em diversas iterações, além de alcançar baixíssimos valores de Loss e Mean Absolute Error. Com a base de validação, os resultados caíram mas continuaram apresentando um desempenho muito bom, chegando a 86% de acurácia e F1 Score e bons valores de Loss e Mean Absolute Error.

Na primeira fase do treinamento da Xception, quando esta recebeu as amostras da base de treino, foi possível conseguir uma precisão máxima de 76% de acurácia e 75% de F1 Score, marcas mais baixas do que as conseguidas com a ResNet. Com a base de validação, nas 10 primeiras iterações, o maior F1 Score alcançado foi de 35%, porém, como o F1 Score estava aumentando continuamente, foram feitas mais 10 iterações na primeira fase, que com a base de treino, chegaram a alcançar 83% de acurácia e 82% de F1 Score. Com a base de validação, as últimas 10 iterações da primeira fase de treino da Xception chegaram ao valor máximo de 37% de F1 Score.

Na segunda fase da Xception, com a base de treino, foi possível, em várias iterações, chegar à valores perfeitos de acurácia e F1 Score, além de conseguir valores muito baixos de Loss e Mean Absolute Error. Com a base de validação, esses valores diminuíram, mas ainda mostraram porcentagens satisfatórias de precisão, chegando a 85% de acurácia e F1 Score.

Finalizando as fases de treinamento, foi possível notar que a ResNet obteve resultados melhores, apresentando 4% e 1% a mais de precisão na primeira e segunda fase, respectivamente.

Os resultados da ResNet com a base de teste superaram os resultados da validação em pouco mais de 1%, chegando a 87% de acurácia e F1 Score. Os valores de Loss e Mean Absolute Error também melhoraram, mas com diferenças menores. Com relação à rede Xception, também houve melhora de acurácia, F1 Score, Loss e Mean Absolute Error, porém os resultados continuaram abaixo daqueles alcançados pela ResNet.

Com os gráficos das figuras 17 e 18, fica claro que a ResNet mostrou resultados significativamente superiores que a Xception em acurácia, F1 Score e Loss. Em relação ao

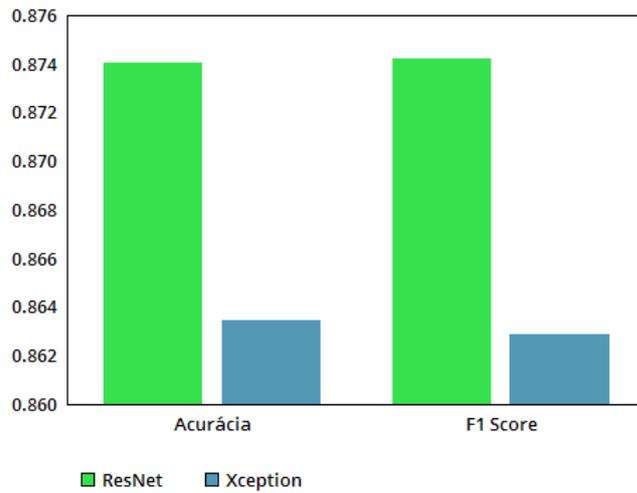


Figura 17 – Comparação dos resultados finais de acurácia e F1 Score das redes ResNet e Xception

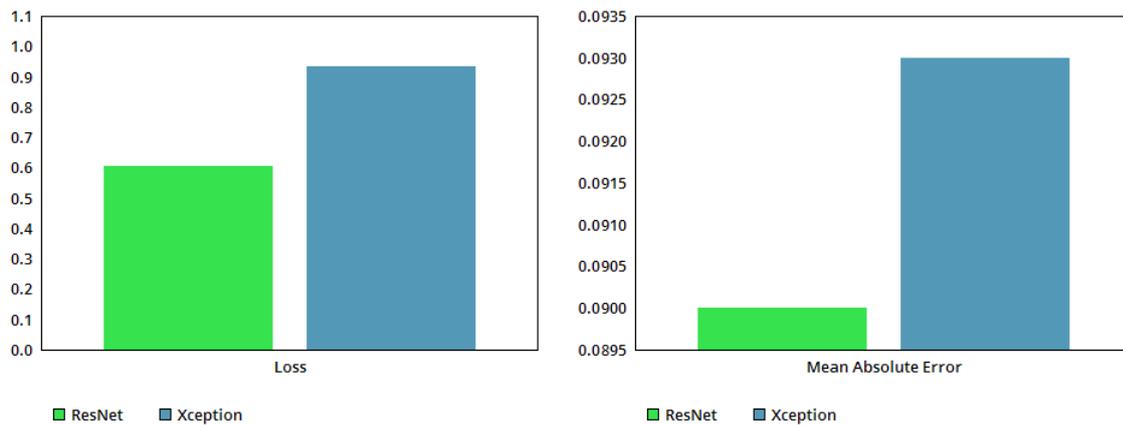


Figura 18 – Comparação dos resultados finais de Loss e Mean Absolute Error das redes ResNet e Xception

Mean Absolute Error, a diferença foi menor do que nos outros quesitos, porém a ResNet ainda assim conseguiu um resultado melhor.

Mesmo com as diferenças de resultados, visíveis desde a fase de treinamento e validação até os testes finais, pode-se concluir que ambas as redes atingiram resultados satisfatórios realizando as predições, uma vez que em questão de acertos, a ResNet apresentou uma precisão somente 1% maior que a Xception. Ambas se mostraram capazes de identificar tendências de alta, baixa e consolidação, podendo então servir como ferramentas auxiliares para investidores que desejam se aventurar no mercado de ações.

## REFERÊNCIAS

- [1] HE, K. et al. Deep residual learning for image recognition. 2015.
- [2] CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. 2017.
- [3] CHIN, C. L. et al. Candlestick charting and trading volume: Evidence from bursa malaysia. *International Review of Management and Marketing*, p. 153–165, 2016.
- [4] FONTANILLS, G. A.; GENTILE, T.; CAWOOD, R. M. *The Stock Market Course*. 1. ed. New York: John Wiley & Sons, Inc., 2001.
- [5] DOWNEY, A. *Think Python: How to Think Like a Computer Scientist*. 2. ed. Needham, Massachusetts: Green Tea Press.
- [6] ZHANG, J. Intro to keras. 2017.
- [7] RASCHKA, S. *Introduction to Artificial Neural Networks and Deep Learning: A Practical Guide with Applications in Python*. 1. ed. 1321 Blanshard Street, Suite 301 Victoria, British Columbia, Canada V8W 0B6: Leanpub, 2017.
- [8] ALVI, Z. M. Artificial intelligence.
- [9] SWARUP, P. Artificial intelligence. *International Journal of Computing and Coporate Research*, v. 2, n. 4, 2012.
- [10] SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms*. 1. ed. 32 Avenue of the Americas, New York, NY 10013-2473, USA: Cambridge University Press, 2014.
- [11] ALPAYDIN, E. *Introduction to Machine Learning*. 2. ed. Cambridge, Massachusetts: The MIT Press, 2010.
- [12] ROJAS, R. *Neural Networks: A Systematic Introduction*. 1. ed. Berlin: Springer-Verlag, 1996.
- [13] SILVA, I. N. *Artificial Neural Networks*. 1. ed. Basel, Switzerland: Springer International Publishing Switzerland, 2017.
- [14] SCHMIDHUBER, J. Deep learning in neural networks: An overview. 2014.
- [15] SAZLI, M. H. A brief review of feed-forward neural networks. 2006.
- [16] GULLI, A.; PAL, S. *Deep Learning with Keras*. 1. ed. Livery Place 35 Livery Street Birmingham B3 2PB, UK: Packt Publishing, 2017.
- [17] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, 2015.
- [18] NAJAFABADI, M. M. et al. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2015.

- [19] HORDRI, N. F.; YUHANIZ, S. S.; SHAMSUDDIN, S. M. Deep learning and its applications: A review. 2016.
- [20] ARESTIS, P.; DEMETRIADES, P. O.; LUINTEL, K. B. Financial development and economic growth: The role of stock markets. *Journal of Money, Credit and Banking*, v. 33, n. 1, p. 16–41, 2001.
- [21] NICHOLSON, C. Dow theory.
- [22] EGELI, B.; OZTURAN, M.; BADUR, B. Stock market prediction using artificial neural networks. 2015.
- [23] ZHENG, A.; JIN, J. Using ai to make predictions on stock market. 2017.
- [24] TIONG, L. C.; NGO, D. C.; LEE, Y. Stock price prediction model using candlestick pattern feature. *International Journal of Interactive Digital Media*, p. 58–64, 2013.
- [25] KAR, A. Stock prediction using artificial neural networks.
-