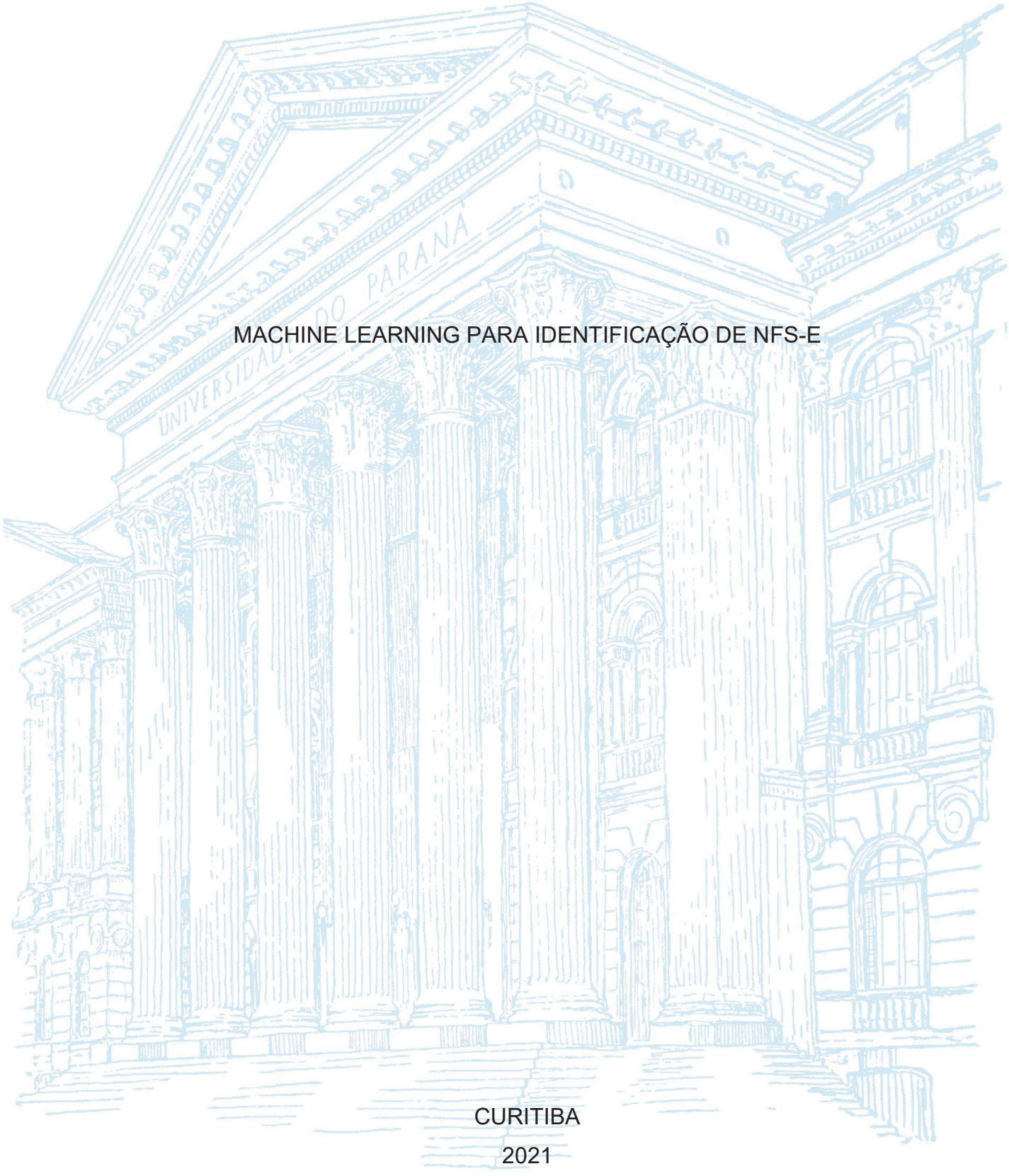
UNIVERSIDADE FEDERAL DO PARANÁ

WILLIAM VIRGILIO GASPARETTO



MACHINE LEARNING PARA IDENTIFICAÇÃO DE NFS-E

CURITIBA

2021

WILLIAM VIRGILIO GASPARETTO

MACHINE LEARNING PARA IDENTIFICAÇÃO DE NFS-E

Artigo apresentado como requisito parcial à conclusão de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientador: Prof. Alexander Robert Kutzke

Coordenador: Prof. Dr. Razer Anthom Nizer Rojas Montaño

CURITIBA

2021

MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO INTELIGÊNCIA ARTIFICIAL APLICADA - 40001016348E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **WILLIAM VIRGILIO GASPARETTO** intitulada: **Machine Learning para identificação de NFS-e**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua ___APROVAÇÃO____ no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 16 de Dezembro de 2021.

ALEXANDER ROBERT KUTZKE
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Machine Learning para identificação de NFS-e

William Virglio Gasparetto
SEPT Setor de Educação Profissional e Tecnológica
Curitiba, Brazil
william.gasparetto@gmail.com

Alexander Robert Kutzke
SEPT Setor de Educação Profissional e Tecnológica
Curitiba, Brasil
alexander@ufpr.br

***Resumo*—A digitalização de dados contidos em Documentos Auxiliares da Nota Fiscal de Serviços Eletrônica (DANFSE) é uma tarefa comum em empresas do Brasil. Com frequência, este trabalho é realizado de forma manual. Nesse sentido, a automação de trabalhos repetitivos é uma das principais áreas de aplicação da inteligência artificial e a digitação de documentos para alimentar sistemas corporativos é um processo que pode ser substituído. Esse processo de automação pode ser otimizado se a primeira etapa for uma filtragem do tipo de documento. Nesse contexto, o presente trabalho visa apresentar uma metodologia para a primeira etapa do processo de digitalização de DANFSEs, a filtragem dos documentos. O sistema proposto filtra documentos em duas classes: 'DANFSE' e 'outros documentos'. Para simplificação do treinamento do modelo, os documentos que compuseram a base de testes foram divididos em dois grupos. O primeiro conjunto foi composto de 500 exemplares de notas fiscais de serviço, enquanto o segundo apresentou 400 documentos de diferentes tipos selecionados aleatoriamente. Para a criação do modelo foi utilizado redes convolucionais. Após os treinamentos com diversas configurações de redes, o melhor resultado obtido apresentou acurácia de 96,93% de detecção de notas fiscais. Arquivos que continham tabelas, como por exemplo boletos, foram classificados erroneamente em 1,6% da amostra de validação. E, por fim, sete arquivos de notas fiscais que não faziam parte da base de treinamento, foram identificadas com sucesso.**

Palavras-Chave—IA, Aprendizagem de Máquina, CNN.

***Abstract*— The digitization of data contained in Auxiliary Documents of the Electronic Services Invoice (DANFSE) is a common task in companies in Brazil. This work is often done manually. In therefore, the automation of repetitive work is one of the main areas of application of artificial intelligence and the typing of documents to feed corporate systems is a process that can be replaced. This automation process can be optimized if the first step is document type filtering. Thus, the present work aims to present a methodology for the first stage of the digitalization process of DANFSEs, the filtering of documents. The proposed system filters documents into two classes: 'DANFSE' and 'other documents'. To simplify the training of the model, the documents that made up the test base were divided into two groups. The first set consisted of 500 copies of service invoices, while the second presented 400 documents of different types selected at random. Convolutional networks were used to create the model. After training with different network configurations, the best result obtained was 96,93% accurate in detecting invoices. Files that contained tables, were erroneously classified in 1.6% of the validation sample (e.g., billets). Finally, seven invoice files that were not part of the training base were successfully identified.**

Keywords—IA, Machine Learning, CNN.

I - Introdução

Todos os serviços prestados no Brasil precisam pagar impostos. O Imposto Sobre Serviços de Qualquer Natureza (ISSQN), instituido durante a reforma tributária de 1965, trouxe a separação de alguns impostos. Ainda nesse ano, a Emenda Constitucional nº 18, de 01 de dezembro de 1965, criou o imposto municipal. Foi à partir de 1967 que o imposto começou a ser cobrado em muitos municípios do país (SOARES, 2005).

Em 11 de novembro de 1983, é criada a Associação Brasileira das Secretarias de Finanças das Capitais (ABRASF) com foco em realizar discussões a respeito receitas e despesas da municipalidade. E é responsável por tentar padronizar a criação da Nota Fiscal de Serviço Eletrônica (NFS-e). Essa padronização englobaria também a parte sistêmica. Hoje cada município tem a sua padronização e com isso temos centenas de leiautes para a impressão, que são chamados de Documento Auxiliar da Nota Fiscal de Serviços Eletrônica (DANFSE) (ABRASF, 2021).

Diante deste histórico, as empresas que precisam extrair informações de serviços prestados para poder realizar o pagamento deste imposto deparam-se com centenas de leiautes diferentes. Prefeituras de grandes cidades acabam desenvolvendo seus próprios padrões, por terem maior volume de arrecadação que as prefeituras de cidades pequenas. As prefeituras com menores volumes de arrecadação optam por contratar empresas terceiras do mercado nacional, para gerenciarem toda a infraestrutura necessária. Cada empresa possui um padrão próprio, mas acabam atendendo muitas prefeituras e com frequência atendem, inclusive, várias cidades de um mesmo estado.

A solução tradicional para coletar as informações dos serviços prestados, seria de forma manual. Este formato, constitui-se em ter uma pessoa recebendo as DANFSEs e geralmente inserindo os dados em um sistema ERP (Enterprise Resource Planning). O que em grandes empresas se torna um problema, por conta do alto volume de documentos a serem digitados e o alto custo por unidade introduzida por operadores humanos.

Uma outra solução, seria a automação desta atividade. Onde a entrada seria composta de arquivos DANFSEs e a saída seriam as informações para o pagamento dos serviços

prestados. Nessa segunda abordagem identifica-se um complicador para o processo, por não existir página especifica para a nota fiscal ser lida. O que para o processo manual é intuitivo - selecionar qual é a página da nota fiscal – em um processo automatizado se torna um problema.

Existem teorias que contemplam o processo inteiro de extração de texto. Como por exemplo a *Document Analysis and Recognition* (DAR – Tradução livre: Análise e Reconhecimento de Documentos) que é um estudo focado na extração automática de informações apresentadas em papel e inicialmente direcionadas à compreensão humana. As aplicações que mais utilizam essa tecnologia, estão relacionadas ao processamento de documentos de escritório (como faturas, documentos bancários, cartas comerciais e cheques). Em uma visão geral o DAR se assemelha à maioria dos sistemas de reconhecimento de padrões, incluindo quatro fases principais: pré-processamento, segmentação de objetos, reconhecimento de objetos e pós-processamento. A primeira fase é verificar a qualidade do documento, a segunda segmenta a imagem em regiões, a terceira categoriza essas regiões e última realiza uma leitura via OCR (*Optical Character Recognition*) do texto (Mithe et al, 2015)

Considerando a metodologia DAR como base para o processo de extração, a fase detalhada nesse artigo, é a de pré-processamento.

A hipótese central do presente estudo é que, mesmo sem um leiuate padrão nacional, as notas seguem uma mesma lógica de criação e, com isso, é possível treinar um modelo com imagens de diversos leiautes. Tal modelo é capaz de classificar se o documento trata-se ou não uma NFS-e.

Para que isso seja possível, foi criado um banco de imagens de NFS-e e de outros documentos.

Essa base de dados de documentos, é rotulada manualmente identificando se a imagem é uma NFS-e ou se é uma imagem de outro documento. Esse conjunto de imagens é dividido entre a base de classificação e a base de testes para avaliar a acurácia do modelo.

O restante do artigo é organizado como segue: a Seção 2 traz todos os conceitos utilizados na construção deste conhecimento, abordando itens como um breve histórico da inteligência artificial, aprendizagem de máquina e seus tipos de aprendizagem, aprendizagem profunda, redes neurais e redes Convolucionais. A Seção 3 descreve a metodologia aplicada na solução do problema proposto. Na Seção 4 são discutidos os resultados obtidos, encerrando o texto com os comentários finais, na Seção 5.

II - FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas pesquisas e discussões acerca dos processos de automação de leitura de dados com a utilização de máquinas. Serão abordados conceitos, uma breve revisão da história e a contextualização dos processos e das aplicações da aprendizagem de máquina.

A primeira conceituação que precisamos fazer é a Inteligência Artificial, que é o estudo de agentes inteligentes. Essa inteligência pode ser desde escolher uma rota, até dar insumos para instruir um carro autônomo. Resumindo, qualquer sistema que tome decisões baseado em uma heurística, pode ser considerado inteligente (Malaquias, 2006).

A inteligência artificial pode ser vista como sendo um grande guarda-chuva, conforme representado na Figura 1, que abriga várias técnicas que ficam a seguir.

- *Machine Learning* (ML), que aprendem de forma supervisionada ou não de uma programação específica. Ou seja, são algoritmos que necessitam de dados para extrair características e aprendizados, que podem ser utilizados na tomada de decisões futuras;
- Dentro desse grupo, existe um subgrupo específico de técnicas de ML, que são intitulados de *Deep Learning* (DL), geralmente utilizam redes neurais profundas;
- Dentro desse subgrupo existem técnicas em áreas tais como, visão computacional e processamento de linguagem natural. Um dos estudos dentro da DL é a Rede Neural Convolucional (CNN), a qual é geralmente composta por várias camadas (*Layers*).

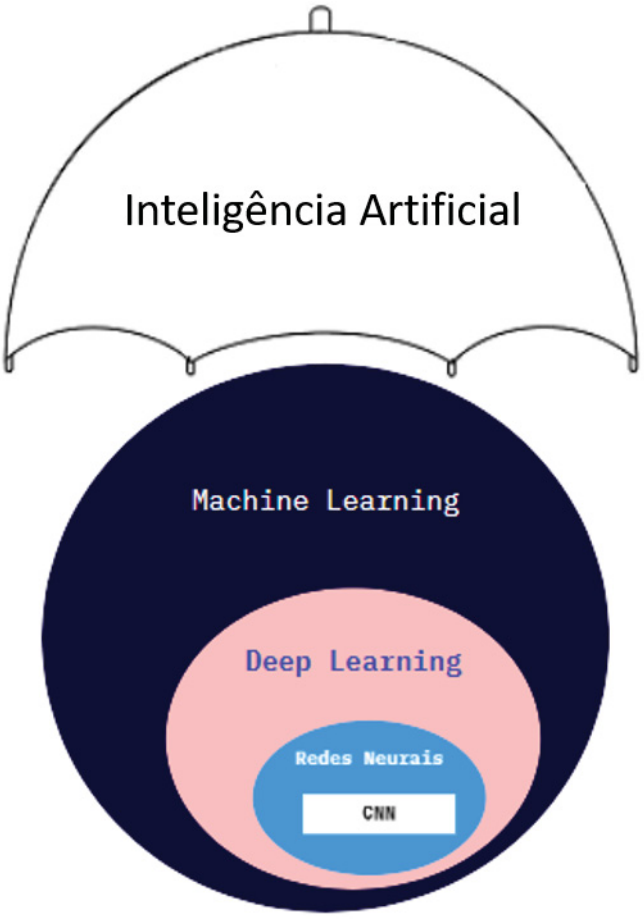


Figura 1. Ilustração do guarda-chuva da Inteligência Artificial (Autor, 2021)

Esses conceitos são explicados a seguir, para criar o entendimento de como foi possível automatizar a tarefa repetitiva e manual de identificar notas fiscais, diferenciando-as dentro de um grupo de outros documentos.

A. Histórico da criação das redes Neurais Arttificiais

As técnicas computacionais que utilizam modelos matemáticos baseados na estrutura neural de organismos inteligentes são as denominadas Redes Neurais Artificiais, e são criadas para adquirirem conhecimentos com a experiência. Podem ser constituídas de centenas ou milhares de unidades de processamento. Comparativamente, o cérebro de um mamífero tem bilhões de neurônios.

A Figura 2 apresenta-se um breve histórico sobre Redes Neurais Artificiais, tendo como base importantes publicações na temática.

De acordo Russell e Norvig (2013), o primeiro estudo sobre Redes Neurais Artificiais foi realizado por Warren McCulloch e Walter Pitts em 1943. A pesquisa propunha um modelo de neurônios artificiais, onde cada neurônio tinha dois estados: "ligado" e "desligado". Eles mostraram que uma função comutável poderia ser calculada por uma rede de neurônios conectados e que todos os conectivos lógicos poderiam ser implementados por estruturas de redes simples. Em 1947 Alan Turing já realizava palestras sobre o tema e em 1950 escreveu o artigo "*Computing Machinery and Intelligence*". Nesse artigo, foi descrito o teste de Turing, aprendizagem de máquina, algoritmos genéticos e aprendizagem por reforço.

O teste de Turing representado na Figura 3, chamado pelo seu criador - Alan Turing- de "Jogo da Imitação" foi desenhado para oferecer uma definição satisfatória de inteligência. Alan Turing definiu o comportamento inteligente como sendo a habilidade de um sistema realizar tarefas cognitivas e alcançar um nível de desempenho equiparado aos seres humanos de tal forma que conseguisse enganar uma pessoa que o interrogasse (Russell e Norvig, 2013).

O teste consistia em uma pessoa realizar um interrogatório, sem poder visualizar se estava conversando com um computador ou não. O computador passaria no teste de respondesse de tal forma que o interrogador não pudesse distinguir se as respostas eram dadas por uma pessoa ou um computador. Para executar o teste de Turing o sistema deveria:

- Ter capacidade de processamento em linguagem natural;
- Ser capaz de guardar toda informação que recebesse antes e durante o interrogatório, dessa forma fazendo a representação do conhecimento;
- Ter a capacidade de usar a informação guardada para responder questões novas que surgissem e inferir novas conclusões, ou seja, fazer a automatização do raciocínio;
- Ser capaz de detectar padrões e se adaptar a novas circunstancias (*machine learning*).

Ainda segundo Russell e Norvig (2013), a IA sempre foi o campo da ciência da computação que tentou construir máquinas que funcionassem de forma autônoma em ambientes complexos e mutáveis.

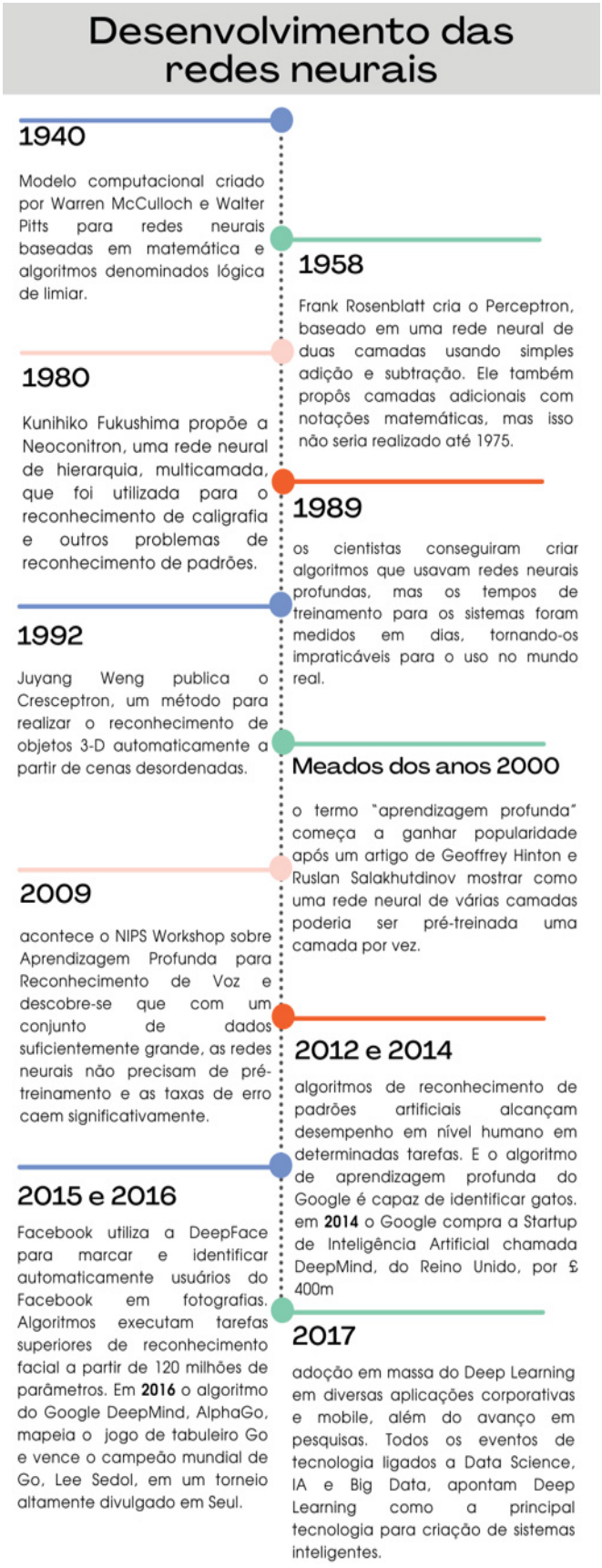


Figura 2. Linha de tempo do desenvolvimento das redes neurais (Autor, 2021).

Já em 1959 Nathaniel Rochester e seus colegas, na empresa IBM, criaram alguns dos primeiros programas de IA que poderiam demonstrar os teoremas complexos de matemática. A partir de 1952, Arthur Samuel escreveu alguns programas para jogos de damas, que aprendiam a jogar em um nível amador elevado. Em 1962 Frank Rosenblatt apresentou o teorema da convergência do perceptron, onde determinava que o algoritmo de aprendizagem poderia ajustar os pesos de conexão de um perceptron para corresponderem a quais dados de entrada. O perceptron é a representação mais simples de uma RNA (Redes Neurais Artificiais) utilizada para classificar padrões com apenas duas classes, é um único neurônio com pesos sinápticos e bias ajustáveis. Após o invento da IA, em torno de 1980, ao menos quatro grupos diferentes reinventaram o algoritmo de aprendizado por retro programação, que tinham sido criados em 1969 (Russell e Norvig, 2013).

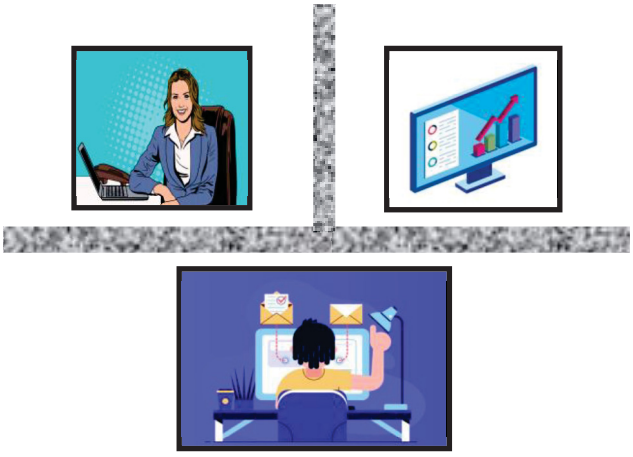


Figura 3. Teste de Turing (Autor, 2021)

Conforme McAllester (1998) resumiu:

> "No período inicial da IA, parecia plausível que novas formas de computação simbólica, como frames e redes semânticas, tornariam obsoleta grande parte da teoria clássica. Isso levou a uma forma de isolacionismo na qual a IA ficou bem separada do restante da ciência da computação. Atualmente, esse isolacionismo está sendo abandonado. Existe o reconhecimento de que o aprendizado da máquina não deve ser isolado da teoria da informação, de que o raciocínio incerto não deve ser isolado da modelagem estocástica, de que a busca não deve ser isolada da otimização clássica e do controle, e de que o raciocínio automatizado não deve ser isolado dos métodos formais e da análise estática. " McAllester (1998, p. 50).

Lee (2019) separa a revolução da IA em quatro ondas: IA de internet, IA de negócios, IA de percepção e IA autônoma.

Quando se fala da revolução da IA, está-se referindo à reestruturação impulsionada pela tecnologia e que causa impactos na vida das pessoas e das instituições.

As revoluções da internet e dos negócios que estão acontecendo hoje, estão moldando as empresas digitais e financeiras de maneiras nunca imaginadas. Mas por incrível que possa aparecer, essa revolução começou em 1998, nos estados Unidos, com a aplicação do *deep learning* em dados dos usuários da internet.

Neste ponto, faz-se necessário revisar e diferenciar os conceitos de *machine learning* e *deep learnig*. *Machine learning* refere-se à capacidade das máquinas aprenderem à partir de dados com que são alimentadas. Isso elimina a necessidade de programá-las explicitamente para realizar uma determinafda tarefa. As máquinas dotadas dessa tecnologia conseguem reconhecer padrões, processar grande volume de informações e fazer predições, aproveitando ao máximo os dados disponibilizados. *Deep learning* é uma técnica de se fazer *machine learning* e veremos mais sobre o assunto nas seções subsequentes.

Retomando ao tema da revolução da IA na internet e nos negócios, sua aplicação significa interpretar e gerar *insigths* com base em um grande número de informações. Um exemplo da IA de internet seriam os motores de recomendação. É assim que empresas como Amazon e Google são capazes de atingir um nível de entendimento tal, de seus clientes, a ponto de saberem qual produto indicar e quando, gerando assim mais vendas.

De acordo com Lee (2019), a IA de negócios começou por volta de 2004, também utilizando o *Deep Learning* aplicado aos dados de empresas de diferentes tamanhos. Além dos *insigths*, é possível fazer avaliação de crédito, identificação de fraudes e até mesmo proceder a diagnósticos de exames médidos. Mais recentemente a criação WeChat que permite que os usuários conversam, joguem, comprem, leiam notícias, paguem por refeições e publicam seus pensamentos e fotos, além de poder reservar uma consulta médica, diretamente de sua conta bancária. A digitalização faz surgir novas oportunidades para diferentes setores.

Segundo Kaufman (2018), a IA da percepção está aprendendo a "ver" o mundo, aprendendo a reconhecer rostos e o mundo físico. Até o momento, essa tecnologia já está sendo utilizada em alguns lugares para ler dados gerados online, quando as pessoas utilizam computadores e outros equipamentos. A ideia por trás da IA Perceptiva é dar olhos e ouvidos à máquina, o que envolve a criação de objetos que se conectam em rede gerando dados. Como exemplo, na loja Amazon Go existem câmeras que conseguem ler quando você escolhe um produto e o debitam em sua conta, sem precisar passar pelo caixa. Além disso, também capta e gera dados das reações diante de um produto, com objetivo de melhora-lo e gerar mais vendas (Tototrelli, 2018)

Entende-se que a IA autônoma será a última onda, e que terá maior impacto através de carros e drones autônomos e robôs inteligentes tomando o controle das fábricas. Se na terceira onda demos olhos e ouvidos à máquina, na quarta

espera-se que elas adquiram braços e pernas, tornando robôs autônomos. Embora o algoritmo já exista, ainda é preciso aperfeiçoar os componentes que darão precisão aos movimentos, além de ter que realizar testes para que o sistema aprenda os comportamentos adequados. Um exemplo são os carros autônomos que ainda estão aprendendo a se movimentar sem provocar acidentes. E assim como na terceirta onda, as questões éticas terão que ser discutidas, leis precisam ser criadas para regulamentar a aplicação das tecnologias (Russel e Norvig, 2013).

No próximo tópico serão aprofundados os detalhes sobre a aprendizagem de máquina ou ML (*Machine Learning).*

B. Machine Learning

A aprendizagem de máquina tem como sua principal meta a construção de programas que melhorem seu próprio desempenho por meio da aprendizagem. Joshi (2017), considera o aprendizado de máquina um subcampo da inteligência artificial cujo propósito é entender a estrutura de dados e ajustá-los em modelos que podem ser entendidos e utilizados pelas pessoas. Os algoritmos utilizados nesse subcampo, permitem que os computadores treinem com entradas de dados e usem análises estatísticas para gerar valores que se enquadrem em um intervalo específico.

Por esse motivo, que o aprendizado de máquina pode facilitar os computadores na construção de modelos a partir de dados de amostra, no intuito de automatizar os processos de tomada de decisão, baseando-se nas entradas de dados.

Um programa aprende a partir da experiência **E** em relação a uma classe de tarefas **T**, com medida de desempenho **P**. (Mitchell, 1997).

Os sistemas de aprendizado de máquina são criados a partir de uma estrutura diferente da programação tradicional de software, em que se cria um conjunto de regras para gerar uma resposta a partir do processamento dos dados inseridos.

Os paradigmas da construção dos sistemas de aprendizado de máquina são (Stange, 2011, Domingos, 2017 e Neves, 2018):

- Simbólico: o sistema constrói representações simbólicas de um conceito através da análise de seus exemplos e contraexemplos. Geralmente assumem a forma de Expressão Logica, Árvore de Decisão, Regras de Produção e Rede Semântica;
- Baseado em instancias: segundo esse paradigma, os dados novos são classificados utilizando dados similares cuja classe é conhecida. É análogo ao que os humanos fazem para resolver problemas;
- Estatístico: constrói-se um modelo estatístico do problema, geralmente utilizando-se a regra de Bayes, que é uma fórmula matemática utilizada para calcular a probabilidade de ocorrencia de um evento dado que outro evento já ocorreu. É chamado de probabilidade condicional;
- Evolucionista: esse tipo de sistema para resolução de problemas utiliza modelos computacionais baseados na teoria da evolução natural. São chamados de algoritmos evolucionários, que possuem uma busca não pela solução ótima, mas sim por uma boa solução baseada em ações estocásticas com menor tempo. Dito de outra forma, são algoritmos que tentam resolver problemas que se fossem testar todas as combinações possíveis, a solução levaria um tempo absurdamente longo;
- Conexionista: diz respeito a redes neurais artificiais, que são formadas por um grande número de unidades de processamento conectadas entre si e serão detalhadas na *Seção D*.

Ao final do processamento dos dados introduzidos, o algoritmo de *Machine Learning* cria suas próprias regras (ou perguntas). Essa tecnologia permite que os modelos sejam treinados em várias combinações de dados antes de serem implementados, resultando em um aplicativo que melhora de forma gradual e automática à medida em que é submetido a repetidas experiencias de treinamento (Domingos, 2017).

Joshi (2017) explica que os dois métodos de aprendizagem comportamental amplamente adotados são os supervisionados e não supervisionados.

No primeiro são recebidos exemplos já rotulados com as saídas desejadas e destina-se a encontrar padrões que possam ser aplicados em um processo analítico. Essa ideia é que o computador aprenda com essas imagens rotuladas e depois consiga classificar se as novas entradas são do mesmo tipo. Um exemplo seria um algoritmo de aprendizado supervisionado fosse treinado com imagens de tubarões e depois identificar tubarões em imagens não rotuladas. Outro exemplo seria usar base de informações históricas do mercado de ações e depois identificar futuras flutuações do mercado.

Na aprendizagem não supervisionada, ou auto organizada, os dados não são rotulados, então o algoritmo precisa encontrar pontos comuns entre seus dados de entrada. É utilizada quando o problema requer uma quantidade muito grande de dados, em que é preciso entender o que há por trás desses dados, fazendo sua classificação com base nos padrões ou clusters encontrados. O objetivo desse segundo tipo pode ser tão direto quanto descobrir padrões ocultos dentro de um conjunto de dados o que permite que a máquina descubra automaticamente as representações necessárias para classificar os dados brutos (Simões, 2018), conforme representado na Figura 4.

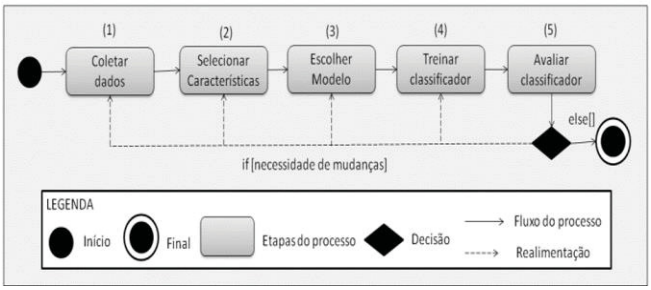


Figura 4. Ciclo de reconhecimento de padrões (Stange, 2011).

Um exemplo de aplicação desse ciclo de reconhecimento de padrões seria o uso para detecção de anomalias, incluindo

compras fraudulentas com cartão de crédito e sistemas de recomendação de próximas compras.

Ainda não conhecemos totalmente o processo de funcionamento de nosso cérebro para, por exemplo, reconhecer objetos tridimensionais em uma cena desordenada, a partir de um novo ponto de vista, em condições diferentes de iluminação. E mesmo que soubéssemos, provavelmente seria muito difícil escrever um programa de aprendizado de máquina.

Outra situação é fazer o reconhecimento, por exemplo, de transações fraudulentas em cartão de crédito. As regras de cálculo da probabilidade de fraude podem não ser simples nem confiáveis. Assim é preciso combinar um número muito grande de regras fracas. Além disso, o programa precisa ficar sempre mudando, porque a fraude é um alvo em movimento.

A solução, utilizando *machine learning*, é coletar muitos exemplos que especificam a saída correta de uma entrada específica. O algoritmo recebe esses exemplos e produz um programa que realiza a tarefa. O programa resultante pode parecer muito diferente daquele eventualmente escrito à mão. Além de poder conter grande número de dados, pode funcionar para novos casos pois, se esses mudam, o programa também vai ser novamente treinado.

O desenvolvimento da *machine learning* que pode ajudar a resolver melhor tarefas como:

- Realizar previsões de vendas, gêneros de filmes que uma pessoa quer assistir, taxas de câmbio ou preço de ações no mercado futuro;
- Identificar anomalias e assim detectar defeitos em máquinas industriais que emitem padrões incomuns de leitura de sensores, transações anômalas em cartão de crédito, o que pode indicar fraude, etc;
- Identificação de palavras escritas ou faladas, identificação de expressões faciais, reconhecimento de padrões, de objetos em cenas reais.

Em seguida será explicado uma subcamada da ML, que é denominada de aprendizagem profunda.

C. *Deep Learning*

Deep Learning, também conhecido pela sigla DL,é parte de uma família de métodos baseados na representação de dados. É um conjunto de algoritmos que tentam modelar abstrações de alto nível de dados, utilizando para isso um grafo profundo com várias camadas de processamento. Essas camadas são compostas por várias transformações que podem ser lineares e não lineares.

É feito para funcionar de forma semelhente ao cérebro humano no tocante ao processamento e troca de informações através de uma rede neural. Nesse método, as informações são processadas em camadas: na primeira estão os dados inseridos para análise e na última estão os resultados a serem exibidos. Entre ambas existem camadas escondifas cujo número varia dependendo da complexidade das operações (Haykin, 2001).

Os algoritmos de *deep learning* conseguem distinguir detalhes das informações de entrada e repassa-los à rede neural, onde passarão por análises através de cálculos matemáticos. Cada um desses cálculos recebe um peso, que assume um grau de relação entre o resultado final e o detalhe analisado. O resultado do cálculo matemático é passado para uma função de ativação de alguns neuroneos que darão continuidade ao processo. Aqui o processo tem certo grau de não linearidade e ocorrem inúmeras análises até alcançar a última camada de resultados (Haykin, 2001).

Esse sistema deve ser alimentado com um grande volume de dados que, somado à quantidade de cálculos, exige alta capacidade de processamento que um computador comum geralmente não é capaz de atender (Nielsen, 2015).

Devido à sua capacidade de reconhecer padrões, de otimizar um resultado específico e de tomada uma decisão, por ser tão abrangente e conseguir resolver problemas do cotidiano, grandes empresas como Google e Facebook investem milhões em pesquisa de projetos ambiciosos. Por isso não param de aparecer em manchetes e mostrar como pode ser a nova era, onde máquinas irão fortalecer ou deslocar os seres humanos (LEE, 2019).

Aprendizagem profunda é uma versão eficiente de redes neurais, segundo Hinton e Salakhutdinov que, para Johnson e Zhang pode executar aprendizagem não supervisionada, supervisionada e semi supervisionada (apud SANTOS, 2019)

O grande diferencial do DL é tornar possível que um computador consiga resolver problemas intuitivos ao ser humano. Como por exemplo reconhecimento de faces, ou neste caso, a diferença entre NFS-e e outros documentos. Que aprenda por experiências obtidas e consiga entender os conceitos do mundo através de uma hierarquia de conceitos mais simples, sem intervenção de um programador. E transformando essa ideia em um desenho, visualizando os conceitos são construídos em relação aos outros, criasse várias camadas. Desse desenho surgiu a expressão deep learning (SIMÕES, 2018).

As redes baseadas em *deep learning* vem sendo aplicadas com sucesso nos últimos anos para tarefas de classificação, regressão, redução de dimensionalidade, modelagem de texturas, recuperação de informações e processamento de linguagem natural (Simões, 2018).

O conceito de rede neurais foi citado em vários momentos e será detalhado no próximo tópico.

D. *Redes neurais*

Para iniciar a conceituação das redes neurais, faz-se importante entender do que elas são feitas. Para isso, iniciamos nosso estudo conceituando os neurônios artificiais, que estão no nucleo dos algoritmos de deep learning e são um subconjunto de aprendizado de máquina.

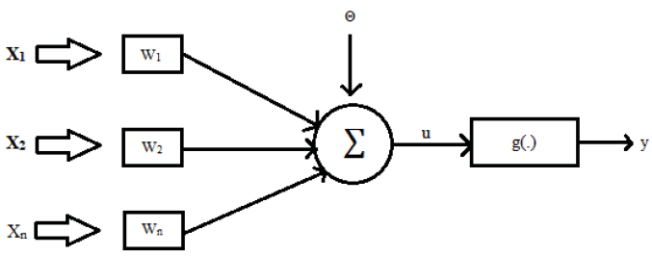


Figura 5. Modelo representativo de um neurônio matemático (Adaptado de DLB, 2020).

Na Figura 5 está representada a constituição de um neurônio matemático, que é um modelo simplificado do neurônio biológico. É baseado na análise da geração e propagação de impulsos elétricos que passam pela membrana celular.

Resumidamente, esses são os componentes de um neurônio matemático.

- **{$X_1, X_2, ...X_n$}** são os dados que alimentam o modelo preditivo. São os sinais externos que sofrem normalização para maximizar a eficiência dos algoritmos de aprendizagem;
- **{$W_1, W_2, …, W_n$}** são os valores aprendidos durante o treinamento e são usados para ponderar os sinais de cada entrada da rede;
- **{∑}** é um combinador linear que agrega todos os sinais de entrada que foram ponderados respectivamente pelos pesos sinápticos com o propósito de gerar um potencial de ativação;
- {Θ} especifica qual será o limiar de ativação. Esse é o patamar indicativo de que o resultado obtido pelo combinador linear gere um valor de disparo de ativação;
- {u} é o chamado Potencial de ativação, obtido pela diferença de valores do combinador linear e o limiar de ativação. Se $u \geq 0$ então o neurônio produz um potencial excitatório. Se $u \leq 0$ o potencial será inibitório;
- {g} á função de ativação, responsável por limitar a saída de um neuronio em um intervalo de valores;
- {y} é o valor final, o sinal de saída que pode ser utilizado como sinal de entrada por neuronios interligados sequencialmente.

É a partir dessa concepção de neurônios matemáticos que se obteve a resolução de problemas de IA nos quais não se conseguia avanço com outras técnicas. Novas arquiteturas, combinações de modelos e aplicação de diferentes técnicas de matemática e estatística possibilitaram a criação de arquiteturas avançadas de *deep learning* como as redes convolucionais (Nielsen, 2015).

Dependendo do propósito para que são utilizadas, as redes neurais podem ser classificadas em diferentes tipos, sendo que as mais comuns são: perceptron, convolucionais e recorrentes.

Segundo Nielsen (2015), o neurônio artificial conhecido por perceptron funciona com entradas binárias e resulta em uma saída. Essa é a rede neural mais antiga, criada por Frank Rosemblatt em 1958, inspirtado em trabalhos anteriores de Warren McCulloch e Walter Pitts.

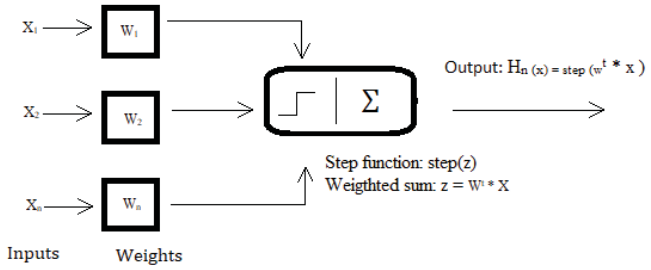


Figura 6. Exemplo de um perceptron. (Adaptado de Nielsen, 2015).

Na Figura 6, Nielsen (2015), mostra um perceptron com três entradas, sendo que esses podem ter mais ou menos entradas. O objetivo é ter uma regra simples para calcular a saída. São introduzidos pesos (w_1, w_2, ...), números reais que expressam os valores das respectivas entradas para a saída. A saída do neurônio será binária, determinada pela soma ponderada Sum ∑jwjxj, se for maior ou menor que algum valor limite. Com os pesos, o limiar é um número real que é um parâmetro do neurônio. Abaixo é exibido em termos algébricos (1):

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

Esse é um modelo matemático simples de um dispositivo que toma decisões após a comprovação de evidências, que são as variáveis binárias x_1, x_2 e x_3. Atribui-se pesos diferentes para cada uma dessas variáveis e o perceptron implementa o modelo de tomada de decisão requerido. Variar os pesos e o *threshold* faz com que se obtenha diferentes modelos de tomada de decisão.

Um perceptron é considerado a menor rede neural *feedforward,* conforme Figura 7, que se propaga apenas em uma direção, representada na Figura 6 (Nascimento&Oliveira, 2016).

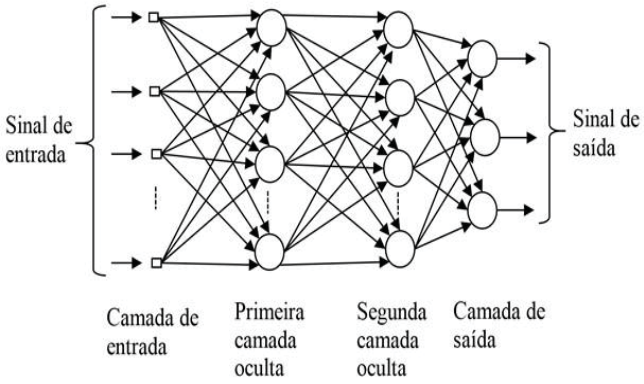


Figura 7. Exemplo de uma rede *feedforward*. (Nascimento&Oliveira, 2016).

É de se esperar que uma rede complexa de perceptrons possa tomar decisões com alto grau de sutilezas. Além de ser utilizado na aprendizagem supervisionada, pode ainda ser usado para classificar os dados de entrada fornecidos. Outras características importantes são a representação de condicionais lógicos (and, or, xor), as funções de ativação e tem uso nos problemas com dados não lineares separáveis (Nascimento&Oliveira, 2016).

Um outro tipo de neurônio artificial citado por Nielsen (2015) é o sigmoide. Ele é parecido com o perceptron, mas pequenas alterações em seus pesos e polarização causa apenas uma pequena alteração na sua produção. Essa diferença que permite que a rede de neurônios sigmoides aprenda. Colocando em prática, o que seria diferente é que as entradas do neurônio variam entre 0 (não ativação) a 1(ativação). A sua saída é chamada função sigmoide, descrita abaixo:

$$\sigma\,(z) \equiv \frac{1}{1 + e^{-z}}$$

Até agora dois tipos de neurônios, em seguida será detalhado a arquitetura das redes neurais através de uma rede simples. A camada mais à esquerda é a camada de entrada (input layer), onde os padrões são apresentados à rede. Os neurônios dentro dessa camada são chamados de neurônios de entrada. A camada mais à direita é chamada de neurônios de saída (output layer), onde o resultado final é concluído e apresentado. As camadas que ficam entre essas duas, são chamadas de camadas ocultas (hidden layers). Nessas camadas é feita a maior parte do processamento, através de conexões ponderadas, podem ser consideradas como extratoras de características. O termo "oculto" é para ficar claro que não fazem parte das camadas de entrada ou saída, representadas na Figura 8 (Nascimento&Oliveira, 2016).

Denomina-se algoritmo de aprendizado o conjunto de regras definidas especificamente para solucionar um problema.

A especificidades de uma rede neural se deve à sua topologia, se refere às características dos nós e também pelas regras de treinamento. Existem diferentes algoritmos de aprendizagem criados para determinados modelos de redes neurais. A principal diferença entre eles é a forma como os pesos são modificados (Nielsen, 2015).

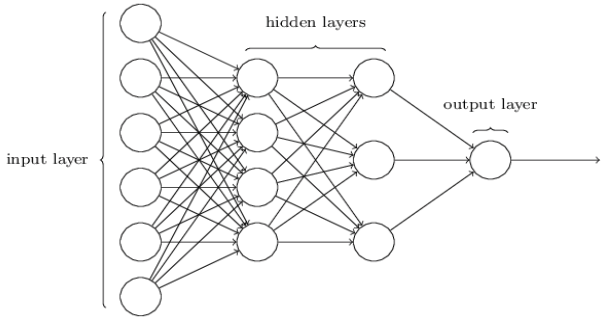


Figura 8. Exemplo de uma rede neural (Nielsen 2015).

Ainda segundo Nielsen (2015), essas redes de múltiplas camadas são as vezes chamadas de multilayer perceptrons ou MLPs, mesmo sendo constituídas por neurônios sigmóides e não por perceptrons. Redes neurais onde uma camada de saída é utilizada como camada de entrada para a próxima, são denominadas *feedforward*. Isso significa que não existe loops nesse tipo de rede, as informações sempre vão para frente e nunca retornam.

Nielsen (2015) expõe que as redes em que seja possível voltar no fluxo, são denominadas "*recurrent neural networks*". Nesse tipo de modelo os neurônios disparam por um tempo limitado, antes de se tornarem quiescentes. Com isso obtemos uma cascata de neurônios disparando.

No processo de aprendizado de máquina, é chamado de ciclo uma apresentação de todos os N pares (entrada e saída) do conjunto de treinamento (Haykin, 2001). Pode-se executar a correção dos pesos em um ciclo de 2 maneiras:

1. Modo padrão em que correção dos pesos acontece cada vez que se faz a apresentação de um exemplo do conjunto de aprendizagem à rede. Assim, a correção dos pesos é baseada somente no erro do exemplo a que se refere aquela iteração. Isso faz com que a cada ciclo ocorram N correções;
2. Modo Batch em que é feita uma única correção por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede, calcula-se o erro médio e a partir daí são feitas as correções dos pesos.

O objetivo é minimizar a função de erro para o mínimo erro local. O cuidado na escolha da taxa de aprendizado tem impacto no tempo necessário para se chegara a esse mínimo. Os erros de generalização da rede podem ser, segundo Papagelis e Kim (2018), os três tipos estão representados na Figura 9:

- de *overfitting*, quanto a rede não generaliza o suficiente durante o processo de aprendizagem dos dados de entrada;
- *de underfitting*: é quando a rede generaliza demais e não detacta os padrões de forma correta.

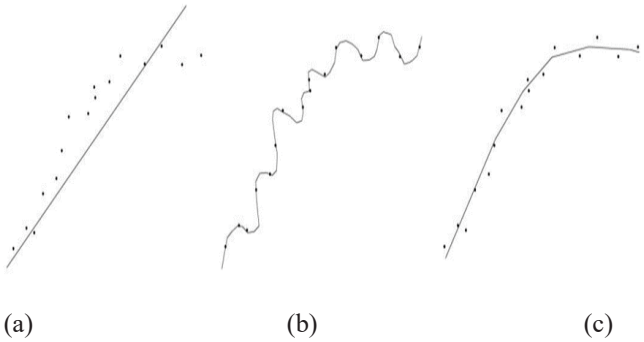


Figura 9. Representação de (a) underfitting, (b) overfitting e (c) boa generalização (Papagelis e Kim, 2018):

O *overfitting* ocorre quando durante o treino o modelo apresenta um desempenho excelente, mas o resultado é ruim

quando se utiliza os dados de teste. Diz-se que o modelo está super adaptando. É como se a rede estivesse apenas memorizando o conjunto de treinamento, sem entender os dígitos suficientemente de forma a generalizar o conjunto de testes. Para reduzir o *overfitting* pode-se adotar estratégias como aumentar a quantidade de dados de treinamento ou reduzir o tamanho da rede. Quando isso não é possível, pode-se adotar técnicas de regularização, sendo a técnica de decaimento de peso (*weight* decay) uma das mais utilizadas. É adicionado um termo extra à função de custo, chamado termo de regularização. Intuitivamente, o efeito da regularização é fazer com que a rede prefira aprender pesos pequenos, sendo que pesos maiores serão permitidos apenas se forem capazes de melhorar a função de custo (Papagelis e Kim, 2018).

Quando o desempenho do modelo já é ruim durante o treinamento, ocorre o *underfitting*. O modelo não consegue encontrar relações entre as variáveis e o teste não precisa ser feito, pois, o modelo deve ser descartado.

É desejável que o modelo apresente uma boa generalização, representada por uma reta bem ajustada. Isso significa um baixo erro na previsão.

Uma rede neural *feedforward* é formada por neurônios ou nós inteligados. Tem a tendência de armazenar conhecimento experimental e disponibilizá-lo. Na rede, o algoritmo atua em conjunto com o *feedforward*. O erro é calculado na ida e os pesos na volta (Nielsen, 2015). Um ponto de atenção é que a rede pode apontar resultados considerados ótimos mas que não são o melhor que poderia ser obtido (mínimo global).

Para aumentar a acurácia algumas técnicas podem ser utilizadas:

- *Dropout* é um técnica que ajuda a evitar o *overfitting*, pois aumenta a quantidade de padões que a rede detecta de uma mesma entrada. Para isso, é feita a remoção aleatória de neuronios das camadas ocultas durante o treinamento de um mini lote de entrada. Esses neuronios são reiniciados ao final do processo, conforme Figura 10;

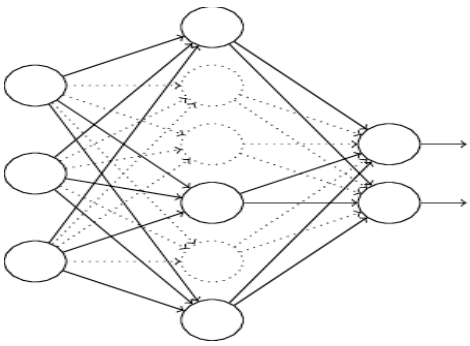


Figura 10. Exemplo de Dropout (DLB, 202O)

- Utilizando-se a técnica de *data augmentation* se consegue reduzir muito o *overfitting*, sem adicionar nenhuma característica nova a ser aprendida e tornando a rede mais robusta. Consiste em uma técnica de aumento artificial dos dados quando não se tem um conjunto de treinamento suficientemente grande e se torna difícil resolver um problema com banco de dados de treino pequenos. Esse aumento pode ser feito por saturação, zoom, variação de brilho, reflexão vertical e horizontal, entre outros métodos.

As redes neurais tem diversas aplicações, transportando para os computadores as premissas de funcionamento na natureza. Essa história, iniciada nos anos 40 do século XX, ainda está sendo escrita a assim como no passado, seu futuro depende do desenvolvimento de hardware específico para usá-las (Araújo et all, 2017).

Não se espera que, com o aprendizado profundo, computadores comecem a pensar como seres humanos. O que se faz é demonstrar que com um conjunto de dados suficientemente grande, processadores rápidos e algoritmos com alto grau de sofisticação, computadores sejam capazes de realizar tarefas que até então eram atribuidas especificamente a seres humanos, como reconhecer imagens e voz, produzir arte ou tomar decisões autônomas (Amaral, 2016).

Isso é póssível após pesquisadores, baseados no funcionamento do neuronio humano, tentarem simular este sistema em computador (Corea, 2017).

Em seguida, é feito o detalhamento de um tipo de rede neural em particular, as redes convolucionais.

E. Convolutional Neural Network

Uma forma bem lúdica de entender *a convolutional neural network* (CNN), descrita por Rothman (2018): imagine ter a tarefa de desenhar um sol com um lápis em um pedaço de papel. Terá que ser em um dia bem claro, em que seja possível olhar o sol claramente. Você coloca um par de óculos de sol com lentes muito densas. Nesse passo acaba de aplicar um filtro de cor, que também é uma das primeiras camadas de uma CNN, que são usadas principalmente para reconhecer imagens, padrões. Com base apenas nesse filtro, poderíamos desenhar um círculo e colocar um pouco de cinza no meio. Essa é a ideia de um filtro de borda. E se acrescentar algumas linhas de raios solares ao redor do círculo, qualquer pessoa poderia deduzir que o que foi desenhado é um sol. Com esses simples passos, foi criado um processo básico de uma CNN, que se vale dos princípios da álgebra linear, em particular a multiplicação de matrizes, para identificar padrões dentro de uma imagem.

A convolução é a aplicação de um filtro que executa um melhor ajuste ao conjunto de dados da imagem em função da redução no número de parâmetros envolvidos.

De uma forma mais técnica, uma rede convolutiva, segundo o autor Haykin (2001), é um perceptron de múltiplas camadas que foi projetado para reconhecer formas bidimensionais, não importando as distorções da imagem. Essa complexa tarefa é aprendida de forma supervisionada e existem algumas formas de restrições.

Em termos matemáticos, a convolução é um operador linear que tem duas funções como entrada e retorna uma terceira função que é decorrente do somatório da multiplicação da função *kernel* superposta da função alvo pelo deslocamento do *kernel* sobre ela. No domínio discreto

a formula da convolução pode ser escrita como (Haykin 2001):

$$(f * g)(k) = h(k) = \sum_{i=0}^{k} f(i) \cdot g(k - i)$$

Sendo:

- *f* e *g* = sequências numéricas de tamanhos iguais ou variados;
- a função retorna o k-ésimo elemento do somatório da multiplicação.

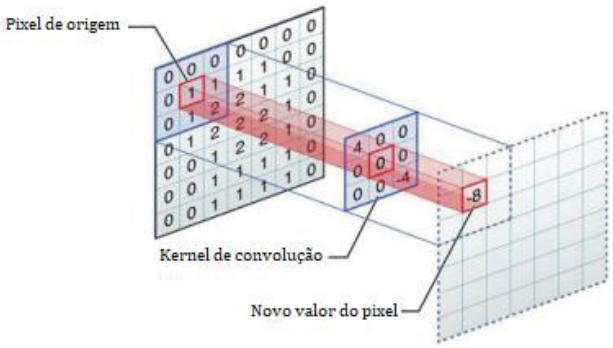


Figura 11. Exemplo de uma convolução (Autor, 2021)

Na Figura 11 é demostrado como funciona o processo de convolução. Utilizando a formula de

Retomando o tema principal deste artigo: Dada uma série de documentos, qual deles trata-se de uma Nota Fiscal?

A forma usual de se resolver tal problema é fornecer uma quantidade grande de dados para um algoritmo de aprendizagem de máquina e deixarmos que ele funcione, explorando os dados e criando um modelo capaz de alcançar o objetivo proposto (Rosa, 2019).

Na extração de características, cada neurônio ao receber os dados de entrada, extrai características locais. Após ser extraída, a sua posição exata na imagem deixa de ser relevante. O que importa agora, é que sua posição em relação a outras características seja preservada.

Outra característica da rede neural *convolucional* é o mapeamento, em que cada camada é composta por múltiplos mapas de características, cada mapa forma um plano dentro do neurônio, onde estão restritos a compartilhar os mesmos pesos sinápticos. Essa segunda forma tem efeitos benéficos quando induzida na operação de um mapa de características através do uso de convolução, seguido por uma função sigmoide (Araújo, 2017).

A última restrição seria a da sub amostragem, onde cada camada que é calculada a média local e realiza uma sub amostragem. Reduzindo assim a resolução do mapa de características. O autor enfatiza que os pesos de todas as camadas de uma CNN são aprendidos por treinamento supervisionado e ela também extrai suas próprias características automaticamente.

Uma outra abordagem aplicada pelo o autor Joshi (2017), detalha que os neurônios em CNNs são organizados em 3 dimensões: largura, altura e profundidade. Cada camada atual é conectada a um filtro NxN na imagem de entrada. Isso contrasta com uma camada totalmente interligada, onde cada neurônio está conectado a todos os neurônios da camada anterior. Como um único filtro não pode capturar todas as nuances da imagem, fazemos isso N vezes, para ter certeza de capturar todos os detalhes. Esses N filtros atuam como extratores de recursos. A filtragem é um conjunto de técnicas destinadas a corrigir e realçar uma imagem. A convolução é o processo de calcular a intensidade de determinado *pixel* em função da intensidade de seus vizinhos. Usa-se a ponderação entre os pesos diferentes dos *pixeis* vizinhos, obtendo-se assim a matriz de pesos, chamada *kernel.*

A Figura 12, ilustra alguns filtros que podem ser utilizados com a ajuda do kernel.

Figura 12. Um template de uma nota fiscal de serviço de Curitiba, utilizando vários kernels. Embaixo das imagens estão os nomes dos filtros utilizados na sua criação(Autor, 2021).

As saídas desses filtros, extraem recursos como arestas, cantos e outras características da imagem.

Isso é verdade para as camadas iniciais. À medida que avançamos pelas camadas da rede, veremos que as camadas posteriores extraem recursos de nível superior. As camadas mais normais em CNNs são: entrada, convolucional, unidade linear retificada e de *pooling*.

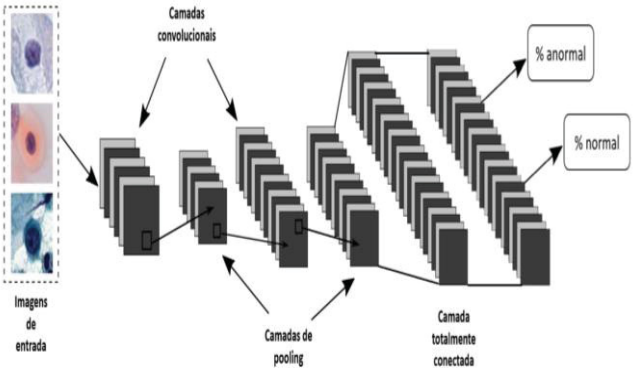


Figura 13. Exemplo da arquitetura de rede para um problema de classificação (ARAUJO et all, 2017).

Estão representados na Figura 13:

- A primeira camada utiliza os dados brutos;
- A segunda calcula as convoluções, basicamente calculando o produto escalar entre os pesos e um pequeno patch na saída da camada anterior;
- A próxima camada aplica uma função de ativação na saída da camada anterior. Essa função é algo como MAX (0, x). Essa camada de retificação é necessária para adicionar a não linearidade à rede, para que possa generalizar bem para qualquer tipo de função;
- A última camada mostra a saída da camada anterior, resultando em uma estrutura com dimensões menores;
- O *pooling* mantém apenas as partes importantes à medida que se progride na rede;
- O pool máximo é frequentemente usado na camada de *pooling*, onde selecionamos a última camada;
- A saída resultante tem o tamanho 1x1xL, onde L é o número de classes no conjunto de dados de treinamento;

A precisão e a robustez do modelo dependerão de muitos fatores, como os tipos das camadas, a profundidade da rede, o arranjo de vários tipos de camadas na rede, as funções escolhidas para cada camada, dados de treinamento e outros (JOSHI, 2017).

Os filtros vão sendo ajustados a cada processamento de entrada no período de treinamento da rede, de tal modo que vão aprendendo estruturas cada vez mais complexas. Isso significa que quanto mais filtros convoluvionais, mais *features* extraímos da entrada. No entanto, isso tem um custo de memória e processamento que precisa ser equacionado ao se definir a arquitetura (ROSA, 2019).

A aplicação de uma camada *pooling* após uma camada convolucional é uma maneira bastante usada para reduzir o tamanho dos dados de entrada. Aplica-se a técnica de *maxpooling*, representado na Figura 14, em que subpartes dos dados originais são reduzidas pelo maior valor encontrado nessas sub-regiões. Com isso, se reduz o tamanho da imagem por um fator de filtro m x n, o que torna o processamento de imagens muito grandes menos pesado (ROSA, 2019).

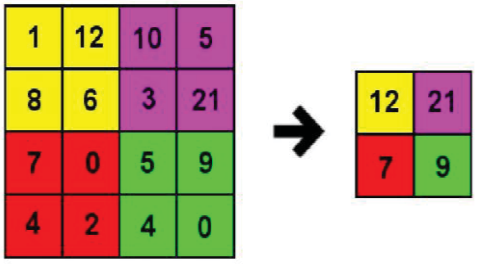


Figura 14. Max pooling com filtro 2x2 em imagem 4x4 (FERREIRA, 2018)

A rede neural também se beneficia pela aplicação da técnica de *pooling*. Na medida em que se reduz a quantidade de dado para a camada seguinte, melhora a regularização da rede. Reduzindo custos de memória e processamento.

No final da rede estão as camadas totalmente conectadas, em que os *features* extraídos nas camadas convolucionais anteriores são utilizados como saída de classificação da rede na Figura 15 pode ser vista uma arquitetura com as três camadas básicas (ARAÚJO et al., 2017)

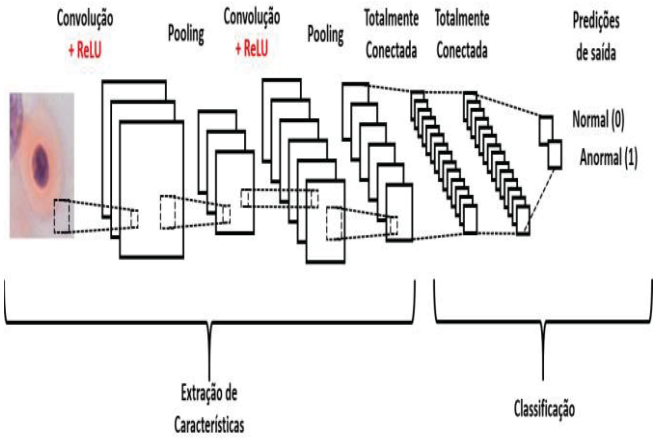


Figura 15 - Exemplo de arquitetura utilizando as camadas básicas de convolução, pooling e totalmente conectadas (ARAUJO et all, 2017).

Em seguida será descrito com medir os resultados de cada modelo.

F. Métricas

Depois do treinamento os diferentes modelos, serão avaliados para que possam ser comparados. Para realizar a

avaliação algumas métricas comuns utilizadas em problemas de classificação serão utilizadas.

A matriz de confusão, representada na Figura 16, é uma ferramenta básica para analisar de forma rápida o desempenho de cada modelo. Os valores que compõem a matriz são obtidos fornecendo os segmentos do conjunto de teste ao método de classificação e comparando sua predição com a classe correta de cada segmento.

		Valor Verdadeiro (confirmando por análise)	
		Positivos	Negativos
Valor Previsto (predito pelo teste)	Positivos	**VP**	**FP**
		Verdadeiro Positivo	*Falso Positivo*
	Negativos	**FN**	**FP**
		Falso Negativo	*Verdadeiro Negativo*

Figura 16. Matriz de confusão de problema de duas classes (Autor, 2021)

Os valores são classificados em quatro possíveis opções:

- Verdadeiro Positivo (VP): segmentos que pertencem a classe positiva e foram classificados como positivos;
- Falso Positivo (FP): segmentos que pertencem a classe negativa e foram classificados como positivos;
- Falso Negativo (FN): segmentos que pertencem a classe positiva e foram classificados como negativos;
- Verdadeiro Negativo (VN): segmentos que pertencem a classe negativa e foram corretamente classificados como negativos.

Foi utilizado a métrica da precisão, que é a quantidade de segmentos que, dentre todos os segmentos classificados como positivos, são pertencem a classe positiva.

$$\text{Precisão} = \frac{\text{VP}}{(VP + FP)}$$

E a métrica da acurácia, que é relativa aos acertos na classificação. É a soma de verdadeiros positivos (VP) e verdadeiros negativos (VN) divididos pelo todo. Essa métrica mostra como o classificador se saiu de uma maneira geral (POWERS, 2011). A acurácia é dada pela equação:

$$Acurária = \frac{|VP| + |VN|}{|VP| + |VN| + |FP| + |FN|}$$

III - MATERIAIS E MÉTODOS

Para construir um detector por segmentação de imagens, o banco de dados se constituiu de imagens e suas respectivas etiquetas. As redes neurais usam coeficientes para aproximar a função que relaciona entradas e saídas e seu aprendizado consiste em encontrar os coeficientes ou pesos certos, fazendo os ajustes de maneira iterativa ao longo dos gradientes que oferecem menos erro. Durante seu aprendizado, as redes convolucionais interpretam as imagens e atribuem um rótulo a ela.

Os resultados estão apresentados conforme as conclusões que foram extraídas, seguindo os passos discriminados na Figura 17.

Para atingir os objetivos propostos e testar as hipóteses experimentais, realizou-se um estudo que foi dividido em seis fases.

A - Coleta dos arquivos

Os arquivos foram coletados de um dos clientes da empresa que tinha a necessidade de extrair informações de serviços prestados a fim de realizar o pagamento de imposto sobre circulação de mercadoria, separando as notas fiscais de outros documentos.

Esses arquivos eram enviados em formato *Portable Document Format* (PDF). Todas as páginas vinham no mesmo arquivo. Cada fornecedor deste cliente tinha o seu padrão ao gerar o documento. Alguns chegavam a colocar mais de uma nota fiscal de serviço eletrônico em um mesmo arquivo.

A qualidade dos arquivos tinha grande variação, tais como:

- Arquivos de imagem gerados em scanners profissionais e outros não;
- Imagens sem enquadramento, com uma pequena rotação;
- Arquivos de imagens com baixa qualidade, obtidas com celular antigo, reunidas posteriormente utilizando-se um programa de edição de fotos.

Conforme as características citadas acima, os arquivos acabavam ficando com tamanho bem diferentes.

Outra variação refere-se a documentos que podiam vir a cores ou preto e branco. Esse ponto traz uma variação em uma imagem da nota fiscal de serviço eletrônica referente a prefeitura. Esse brasão que geralmente fica na parte superior à esquerda do documento, quando vem no preto e branco tende a perder um pouco de sua qualidade nos traços.

A base de trabalho foi constituída por 245 arquivos formatados para a extensão PDF. A quantidade de páginas variava de 5 a 21 por arquivo.

O trabalho seguiu uma metodologia qualitativa, cujos passos principais são a coleta de dados, sua separação em um conjunto de treinamentos e um conjunto de testes, destinados a apontar as métricas da avaliação:

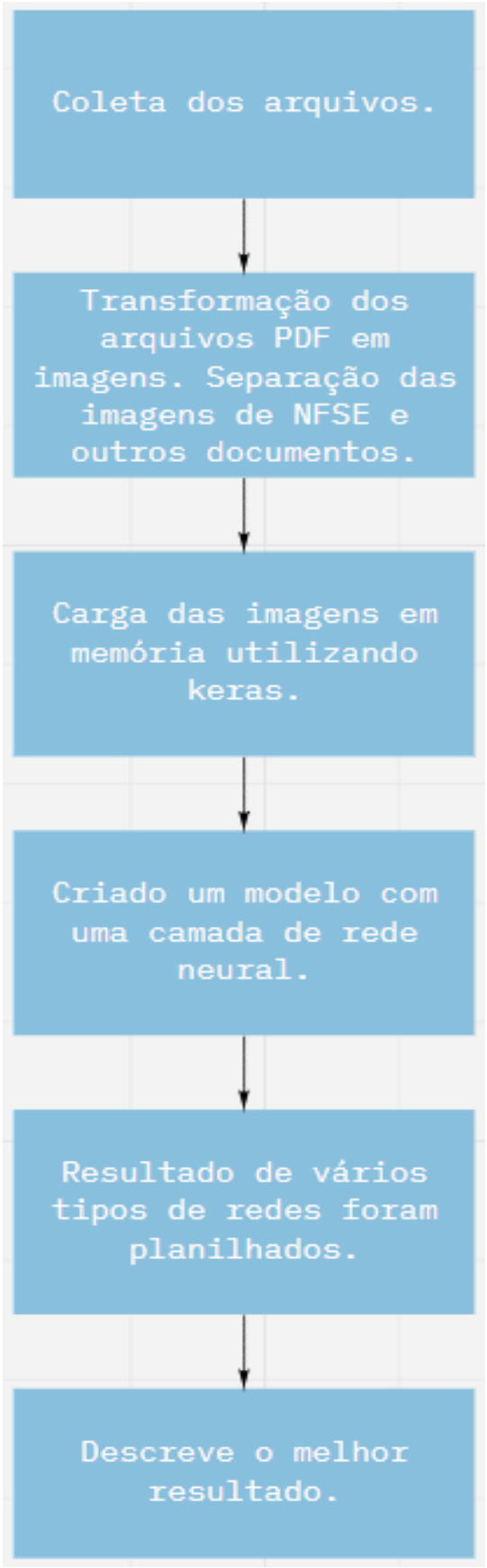


Figura 17. Passos utilizados na execução do projeto (Autor, 2021).

B - Transformação dos arquivos PDF em imagens de NFSE e outros documentos

Foi criado um pequeno programa para transformar as páginas dos arquivos que estavam no formato PDF para o formato JPEG (*Joint Photographic Experts Group*). Esse formato foi escolhido por suportar de 224 até 16.8 milhões de cores e por conseguir compactar bem o tamanho. A média de tamanho das imagens ficou em 1,15 MB, com uma dimensão de 3306 x 4678, com a resolução de 96 DPI e a intensidade de bits de 24.

No programa foi utilizado o Pdf2Image, que é um módulo *python* que envolve *pdftoppm* e *pdftocairo* para converter PDF em um objeto do tipo imagem PIL. O PIL é uma classe que vem do módulo *Image*. Esse modelo tem várias funções de fábrica, a que foi utilizado é criar novas imagens (ROSA, 2019).

Executados os passos para transformação dos arquivos, em seguida, procedeu-se à tarefa de unir todos os arquivos PDF no mesmo diretório. Em seguida foi criada uma estrutura de loop, que com a ajuda do *glob* capturava a lista de arquivos. Um segundo loop era utilizado para pegar cada página, que virava uma estrutura de *file então* era enviada para o método "*convert_from_path*" passando o *file* e o tamanho em pixel que seria transformado, o selecionado foi 400. E para salvar a página em uma forma de imagem era utilizado o método *save* do file, passando o nome completo do arquivo.

C - *Carga das imagens em Memória utilizando Keras*

Dataset são componentes fundamentais para um projeto de *machine learning*. São a base utilizada para que um algoritmo aprenda, evolua e exiba seus resultados. Essas bases de dados servem de amostras para treinamento. São importados e processados com as bibliotecas específicas da linguagem utilizada, neste trabalho, em *Python*.

Para carregar as imagens foi utilizado o método "*image_dataset_from_directory*" da biblioteca do *Keras*.

Esse método retornou um objeto do tipo *data. Dataset* que produz lotes de imagens dos subdiretórios class_a e class_b, junto com os rótulos 0 e 1. Foi utilizado esse método pois ele aceita os formatos de imagem: jpeg, png, bmp, gif.

Após criado o objeto, foi chamado a função *flow_from_directory,* setando os seguintes atributos:

Função	Atributo
batch_size	valor 64
directory	google drive
shuffle	*False*
target_size	150 para as duas dimensões
class_mode.	*binary*

Tab 1- Atributos da função *flow_from_directory (Autor, 2021)*

Essa forma de carregar foi utilizada para criar a base de treinamento, validação e teste, adaptando o fluxograma apresentado por Rosa (2019) na Figura 18.

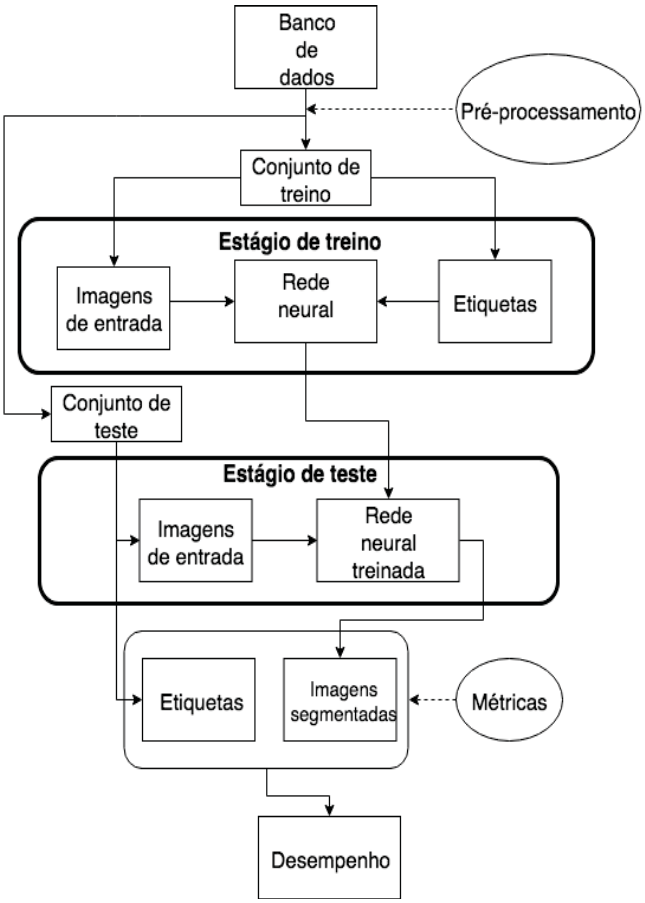


Figura 18. Fluxograma para segmentação e validação das imagens (ROSA, 2019)

Os dois primeiros passos são a coleta de dados relativos ao problema e sua separação em dois conjuntos: treinamento e teste. Além dessas é possível ainda criar - dentro do conjunto de treinamento - um conjunto de validação utilizado para verificar a eficiência da rede quanto à sua capacidade de generalização durante o treinamento. Depois disso, são colocados em ordem aleatória.

Os totais abaixo são a quantidade de imagens utilizadas em cada base:

- Treinamento tipo NFSE: 258
- Treinamento tipo Outros Arquivos: 294
- Validação NFSE: 67
- Validação Outros Arquivos: 88

Durante a fase de teste é determinada a performance da rede, com dados que não foram utilizados previamente. A performance da rede, medida na fase de teste é uma boa indicação de seu desempenho real.

A fase de teste é o momento para analisar o comportamento da rede diante de entradas especiais e também para se fazer a análise dos pesos: se os valores forem muito pequenos, podem indicar que as conexões associadas são insignificantes e devem ser eliminadas. Caso contrário, valores muito discrepantes podem indicar que houve over-training da rede.

D – Criação de um modelo com uma camada de rede neural

As CNN se utilizam de uma arquitetura particularmente bem adequada para classificar imagens, que mostra ser um modelo de *deep learning* confiável para uma previsão automatizada de ponta a ponta. Possui três camadas principais:

1. Camada convolucional – extrai recursos de alto nível dos dados de entrada e os repassa para a próxima camada. Na forma de mapas de recursos;
2. Camada *pooling* – utilizada para reduzir as dimensões dos dados. Recebe cada saída do mapa de recursos da camada anterior e prepara um mapa de características condensado;
3. Camada totalmente conectada – realiza a tarefa de classificação. As pontuações de probabilidade são calculadas para cada rótulo de classe pela função de ativação – *softmax*.

Foi utilizado o *Sequential* para agrupar a pilha linear de camadas em um *tf. Keras. O keras* é uma API de *deep learning* escrita em *Python*

Model. Essa função fornece recursos de treinamento e inferência para o modelo. Utilizando a função *add* para adicionar a instância de camada no topo da pilha de camadas.

E – *Resultado de vários tipos de redes forma planilhados*

A cada execução de uma configuração da rede, eram executados testes com a base de validação.

Para isso foi utilizado o método *predict* do objeto do *model*. Com isso era gerado a tabela verdade de cada experimento.

F – Descrever o melhor resultado

Baseado nas métricas de precisão e acuracia, os modelos foram comparados.

Assim que os medelos eram treinados e testados. Eram guardados as informações, para posteriormete serem comparados.

IV - RESULTADOS E DISCUSSÕES

Os documentos coletados apresentaram grande heterogeneidade, devido ao fato de que não existe um padrão para criação de notas fiscais no Brasil.

Na figura 19 está destacado em vermelho as localidades de origem das NF utilizadas para realização do testes.

Estados	Cidades	Empresas
14	91	33

Figura 19. Origem das NFS-e e demais documentos utilizados (Autor, 2021).

Cada estado ou até mesmo cidades diferentes adotam seus próprios leiautes, sendo que algumas características comuns possibilitaram o reconhecimento pela máquina, das imagens como sendo NF:

- A denominação Nota fiscal;
- Número de ordem, a série e subsérie e o número da via;
- Nome, endereço e inscrição da empresa no CNPJ;
- Data da operação;
- Descrição dos bens ou serviços;
- Os valores, unitário e total, das mercadorias, outros valores cobrados a qualquer título e o total da operação.

Os documentos recebidos no formato PDF foram exportados como arquivos de imagem no formato JPEG com tamanho em torno de tamanho das imagens ficou em 1,15 MB, com uma dimensão de 3306 x 4678, com a resolução de 96 DPI e a intensidade de bits de 24.

Procedeu-se ao processamento sequencial de extração de características e *pooling* dessas características. Ao final essa imagem é convertida em *array*, que são valores efetivamente utilizados como dados de entrada para a rede neural, como pode ser observado na Figura 20. A transformação da imagem do modelo matriz para o formato *array* é o passo final do processo de modificação da estrutura da imagem. Isso melhora a qualidade dos dados de entrada e para que a CNN aprenda, é preciso que o *dataset* não tenha valor nulo, todos os dados sejam numéricos e estejam normalizados.

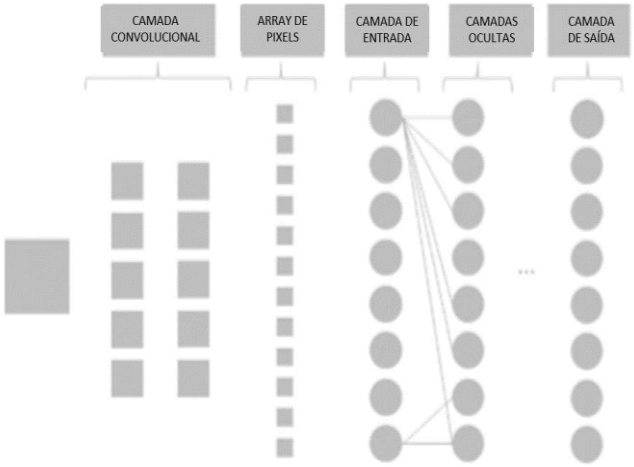


Figura 20. Estrutura final da rede neural (SILVA & SCHIMIGUEL, 2020)

A configuração de redes neurais é feita de forma empírica, por isso foram feitas algumas compilações e treinos, até chegar à melhor configuração. Durante o treinamento o modelo percorre o *dataset* e aprende os padrões.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 150, 150, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 75, 75, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 75, 75, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 37, 37, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 37, 37, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 18, 18, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 20736)             0
_________________________________________________________________
dense (Dense)                (None, 512)               10617344
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 10,641,441
Trainable params: 10,641,441
Non-trainable params: 0
```

Matriz de confusão - Imagens retiradas da empresa terceira

		Positivos	Negativos
Positivos		8	2
Negativos		8	2

Matriz de confusão - Imagens de outras fontes

		Positivos	Negativos
Positivos		51	7
Negativos		54	4

Figura 21. Arquitetura CNN implementada com 30 épocas e as matrizes de confusão geradas (Autor, 2021).

Durante o processo, o modelo da CNN vai aprender e também cometer erros. A quantidade de treinos é denominada

épocas. O melhor modelo deve gerar o mínimo de perdas e a maior precisão possível no final da última época.

A. Testes

Testou-se com diferentes quantidades de épocas e fez-se a tabela de matriz verdade tendo como base dois bancos de dados. Uma com imagens retiradas do grupo que veio das imagens da empresa parceira e um segundo grupo de imagens coletas fora deste universo. Contou-se com ajuda de particulares para adquirir essas notas de serviço.

A primeira base ficou com 20 imagens e foram selecionadas de forma aleatória. E as que foram coletadas externamente da empresa terceira, foram 116 imagens.

Foram utilizadas variadas configurações de arquitetura. Focando no melhor resultado.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 1)	513

Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0

Matriz de confusão - Imagens retiradas da empresa terceira

	Positivos	Negativos
Positivos	10	0
Negativos	7	3

Matriz de confusão - Imagens de outras fontes

	Positivos	Negativos
Positivos	55	3
Negativos	53	5

Figura 22. Arquitetura CNN implementada com 30 épocas e as matrizes de confusão geradas (Autor, 2021).

A Figura 21 mostra uma arquitetura com 30 épocas e que o resultado foi melhor em identificar imagens que não eram notas fiscais de serviço eletrônica.

Na Figura 22 mostra uma arquitetura com mesma quantidade de épocas e com um resultado melhorando, mostrando que era possível ter uma acurácia alta. E na Figura 23 a arquitetura se mostrou totalmente ineficiente, errando tudo.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d_4 (MaxPooling2	(None, 75, 75, 32)	0
conv2d_5 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_5 (MaxPooling2	(None, 37, 37, 64)	0
conv2d_6 (Conv2D)	(None, 37, 37, 128)	73856
max_pooling2d_6 (MaxPooling2	(None, 18, 18, 128)	0
conv2d_7 (Conv2D)	(None, 18, 18, 128)	147584
max_pooling2d_7 (MaxPooling2	(None, 9, 9, 128)	0
conv2d_8 (Conv2D)	(None, 9, 9, 256)	295168
max_pooling2d_8 (MaxPooling2	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 512)	2097664
dense_1 (Dense)	(None, 1)	513

Total params: 2,634,177
Trainable params: 2,634,177
Non-trainable params: 0

Matriz de confusão - Imagens retiradas da empresa terceira

	Positivos	Negativos
Positivos	0	10
Negativos	10	0

Matriz de confusão - Imagens de outras fontes

	Positivos	Negativos
Positivos	0	58
Negativos	58	0

Figura 23. Arquitetura CNN implementada com 40 épocas e as matrizes de confusão geradas (Autor, 2021).

B. Ruidos na aprendizagem de máquina

Algumas imagens podem apresentar deteriorações (ruídos) devido ao processo de escaneamento, manchas de fluidos e dobras do papel, entre outros. Esses ruídos podem provocar a classificação incorreta das imagens.

Minimizar os erros e maximizar a precisão nas classificações é o grande objetivo dos algoritmos de aprendizagem indutivos. No entanto, podem aparecer fatores que interferem na qualidade da base de dados, gerando ruídos que podem ser:

- Ruídos de atributo: geralmente causado por erros na coleta dos dados. Isso causa os *outliers*, amostra que possui dados discrepantes, se comparada à população;
- Ruídos de classe: podem advir de exemplos contraditórios ou instâncias classificadas de maneira incorreta.

Ao final do trabalho foi realizado um levantamento de imagens que foram classificadas erroneamente em diferentes tipos de arquitetura. A Figura 24 é um exemplo de erro, em várias arquiteturas essa imagem foi classifica como uma NFSE. Não foi explorado o problema, mas como tem um emblema no mesmo lugar que nas figuras que representam as prefeituras. E existem umas linhas após o cabeçalho, isso deve ter enviesado o treinamento.

Figura 24. Documento de descrição (Autor, 2021).

A Figura 25 foi também classificada erroneamente. Mesmo sendo bem diferente das notas ficais de serviço eletrônica. É uma imagem com muitas cores, mas os traços e retas, podem ter gerado esse problema.

E esse erro ocorreu mesmo em arquiteturas que foram treinadas com mais de 60 épocas e com mais camadas ocultas. Não foi mais explorado o porque do erro, por falta de tempo de experimentos.

Essa imagem é de uma conta de energia da ciade de Curitiba.

Figura 25. Capa de conta de luz (Autor, 2021).

Na amostra de teste foi inserido uma imagem de pagamento de serviço, que não fazia parte das amostras coletadas no cliente terceiro. A figura 26 está borrada pois é de uma pessoa física. E o modelo não a classificou como uma NFSE.

Figura 26. Nota Fiscal de Serviço Eletronica (Autor, 2021).

Figura 27. Descrição da conta de serviço (Autor, 2021).

A Figura 27 confundiu o modelo, pois se assemelha muito a o padrão geral de uma NFSE.

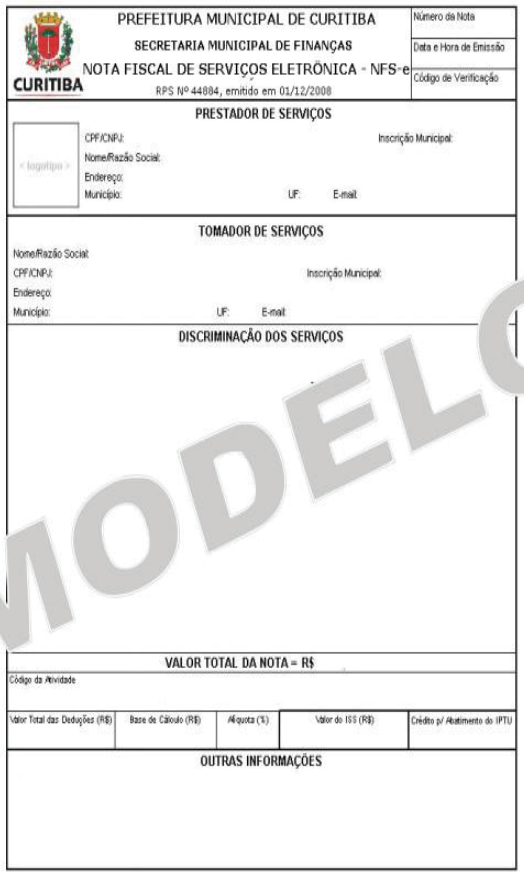


Figura 28. Modelo NFSe de Curitiba (Autor, 2021).

A Figura 28 é do modelo de Curitiba, que é parecido com o padrão das outras capitais do Brasil.

Esse modelo disponibilizado no site da prefeitura de Curitiba, pode mostrar as semelhaças nos contornos.

O melhor resultado foi atingido com a arquitetura demostrada na Figura 29.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling 2D)  (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling 2D)  (None, 17, 17, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 36992)             0
_________________________________________________________________
dense (Dense)                (None, 512)               18940416
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 19,034,177
Trainable params: 19,034,177
Non-trainable params: 0
```

Matriz de confusão - Imagens retiradas da empresa terceira

	Positivos	Negativos
Positivos	8	2
Negativo	8	2

Matriz de confusão - Imagens de outras fontes

	Positivos	Negativos
Positivos	54	4
Negativos	54	4

Figura 29. Arquitetura CNN implementada com 40 épocas e as matrizes de confusão geradas (Autor, 2021).

Foram utilizadas 40 épocas para realizar o treinamento. Chegando a 96,93% de acurácia na última época de treino.

Os gráficos X e Y mostram o resultado da acurácia e pela perda, demostrados na figura 24.

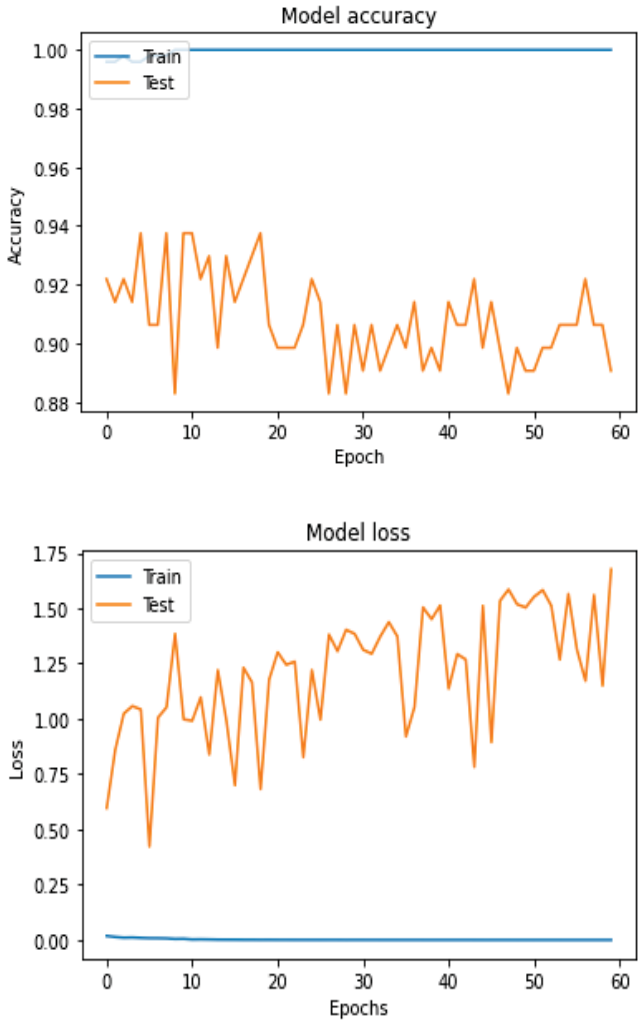


Figura 30. Gráficos de acurácia e de perda (Autor, 2021).

V - CONSIDERAÇÕES FINAIS

O objetivo proposto no estudo foi criar um sistema capaz de desempenhar autonomamente a tarefa de recebimento de arquivos contendo DANFSE, tendo como saida informações para o pagamento dos serviços prestados. Um complicador para o processo é o fato de que os arquivos recebidos geralmente são constituídos de muitas páginas e não existe uma pagina especifica para a nota fiscal a ser lida.

Portanto, o modelo de machine learning deve ser capaz de selecionar qual página apresenta a nota fiscal.

O modelo criado conseguiu com uma precisão de 90% e ao final do treino alcançou uma acurácia de 96,93%.

Com esse resultado se tornou viável a utilização deste modelo para realizar a primeira etapa de filtro de classificação de documento.

São vários os pontos que deixam a solução bem interessante:

- A execução da função de verificação é de milissegundos;
- O modelo pode ser treinado em uma máquina remota;
- Para realização dos testes contidos nesse artigo, foi utilizado a plataforma Colab, gratuita, que foi desenvolvida pelo Google.

A possibilidade de utilização de uma plataforma gratuita se mostrou ainda mais interessante, uma vez que a solução utilizada anteriormente na empresa, envolvia enviar os documentos para uma empresa que disponibilizava um serviço na nuvem e cobrava a execução por documento.

Alguns apontamentos para futuras pesquisas:

Nem todas as cidades do Brasil foram utilizadas para criação deste modelo.

Com isso para melhorar a precisão do modelo, será necessário recria-lo quando novas cidades chegarem. Não será necessário alterar a rede ou qualquer outra configuração. Apenas adicionar as novas imagens nos exemplos de treinamento e de validação.

Quanto às imagens que foram classificadas incorretamente, a solução é desenvolver um algoritmo que limpe as imagens utilizando as abordagens:

- Iteração – o problema pode ser abordado como uma regressão em que os parâmetros da função que recebe o pixel, ou conjunto de *pixeis*, é ajustado numa dada posição e deve retornar o valor correto ou limpo dele. Com a iteração desse processo para todos os pixels, espera-se que al final a imagem esteja limpa;
- A influência dos vizinhos no valor predito de um pixel- essa influência pode ocorrer devido as propriedades percebidas nas imagens como a continuidade entre traços e padrões de ruído. Assim, se os valores dos vizinhos de um pixel variam muito, é provável que este seja um pixel com valor incorreto. O padrão caótico é característico de sujeiras nas imagens;
- Filtro para separar cinza, branco e preto – através de uma técnica chamada *Tresholding* que faz com que se o pixel for clusterizado como branco ou preto, seu valor atual é mudado para 0 (pret) ou 225 (branco). O valor para cinza permanece inalterado.

REFERÊNCIAS BIBLIOGRAFICAS

[1] ABRASF Associação Brasileira das Secretarias de Finanças das Capitais Commons Brasil. Disponível em: http://www.abrasf.org.br/. Acesso em: 21 jan. 2021.

[2] AMARAL, F. **Introdução a ciência de dados**: mineração de dados e Big Data. Rio de Janeiro: Alta Books, 2016.

[3] ARAÚJO, Flávio H. D.; CARNEIRO, Allan C.; SILVA, Romuere R. V.; MEDEIROS, Fátima N. S.; USHIZIMA, Daniela M. **Redes Neurais Convolucionais com Tensorflow: Teoria e Prática**. III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, v. 1, n. 1, p. 382-406, jun, 2017. Disponível em: http://www.eripi.com.br/2017/images/anais/minicursos/7.pdf. Acesso em: 15 out. 2021.

[4] COREA, F. **Artificial Intelligence and Exponential Technologies**: Business Models Evolution and New Investment Opportunities. Rome: Springer, 2017.

[5] DPL, Deep Learning Book. Disponível em: http://deeplearningbook.com.br/. Acesso em: 14 nov 2021.

[6] DOMINGOS, P. **O Algoritmo Mestre**. Tradução de: SILVA, A. J. C. C. 1. ed. São Paulo: Novatec Editora Ltda, 2017. Título original: The Master Algorithm.

[7] FERREIRA, Alessandro dos Santos. Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja. Dissertação de mestrado, Universidade Federal de Mato Grosso do Sul, 2017. Disponível em: http://www.gpec.ucdb.br/pistori/orientacoes/dissertacoes/alessandro2017.pdf. Acesso em: 15 nov 2021.

[8] GRUS, J. **Data Science do Zero**. Tradução de: NASCIMENTO, W. 1. Ed. Rio de Janeiro, Alta Books, 2016. Título original: Data Science From Scratch: First Principles with Python.

[9] HAYKIN, S. **Redes neurais**: princípios e prática. Tradução de: ENGEL, P. M. 2. ed. Porto Alegre: Bookman, 2001.

[10] https://www.researchgate.net/publication/312027170_Sensitivity_Analysis_of_Cutting_Force_on_Milling_Process_using_Factorial_Experimental_Planning_and_Artificial_Neural_Netwo rks. Acesso em: 12 out. 2021.

[11] JOSHI, P. Artificial **Intelligence with Python. Birmingham**: Packt Publishing, 2017.

[12] KAUFMAN, D. -**A inteligencia artificial irá suplantar a inteligencia humana?** [recurso eletronico] Barueri, Sãp Paulo: Estação das Letras e Cores, 2018. 94 p; ePUB.

[13] LEE, K. **Inteligência artificial**: como os robôs estão mudando o mundo, a forma como amamos, nos relacionamos, trabalhamos e vivemos. Tradução de: BARBÃO, M. 1. ed. Rio de Janeiro: Globo Livros, 2019. Título original: AI Superpowers: China, Silicon Valley and the New World Order.

[14] LUGER, G. F. **Inteligência Artificial**. Tradução de: VIEIRA, D. 6. ed. São Paulo: Paerson Education do Brasil, 2013. Título original: Artificial intelligence.

[15] MALAQUIAS, N. G. L. **Uso dos algoritmos para a otimização de rotas de distribuição**. (Mestrado). Universidade Federal de Uberlândia. 2006. Acesso em 23 de novembro 2021.

[16] MCALLESTER, D. A. (1998). **What is the most pressing issue facing AI and the AAAI today?** Candidate statement, election for Councilor of the American Association for Artificial Intelligence.

[17] MITCHEL, T. M. Machine learning. 1ª edição. mcGraw-Hill, 1997. ISBN: 0070428077.

[18] MITHE, R; INDALKAR, S; DIVEKAR, N. **Optical character recognition.** International journal of recent technology and engineering (IJRTE), v. 2, n. 1, p. 72-75, 2013.

[19] NASCIMENTO, E. O.; OLIVEIRA, L. N. Sensitivity Analysis of Cutting Force on Milling Process using Factorial Experimental Planning and Neural Networks. **IEEE Latin America Transactions**, vol. 14, n. 12, p. 4811-4820, 2016. Disponível em: https://www.researchgate.net/publication/317078568_Determinacao_da_Resistencia_Mecanica_em_Juncoes_Adesivadas_Estruturalmente_usando_Redes_Neurais_Artificiais_e_Planejamento_Fatorial_de_Experimentos. Acesso em 20 nov 2021.

[20] NEVES, S. A. **Técnicas de aprendizado de máquina aplicadas a classificação da qualidade de pavimentos asfálticos utilizando smartphones**. 49 f. Trabalho de Graduação (Engenharia da Computação) – Curso de Engenharia da Computação do Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2018.

[21] NIELSEN, M. **Neural Networks and Deep Learning: Introduction to the core principles.** San Francisco: Independent, 2015. E-book. Disponível em: http://neuralnetworksanddeeplearning.com/. Acesso em: 17 jan. 2021.

[22] PAPAGELIS, A. J.; KIM, D. Soo. **Backpropagation**. Disponível em: https://www.cse.unsw.edu.au/~cs9417ml/MLP2/BackPropagation.html. Acesso em: 13 out. 2021.

[23] ROSA. M. C. da, **Redes neurais convolutivas aplicadas à detecção de ervas daninhas**. UFRGS, 2019. Disponível em: https://lume.ufrgs.br/bitstream/handle/10183/204423/001110031.pdf?sequence=1&isAllowed=y. Acesso em 21 out 2021.

[24] ROTHMAN, D. **Artificial Intelligence By Example**. Birmingham: Packt Publishing, 2018.

[25] RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. Tradução de: SEMILLE, R. C. 3. ed. Rio de Janeiro: Elsevier, 2013. Título original: Artificial intelligence.

[26] SANTOS, K. B. C. **Categorização de textos por aprendizagem de máquina.** 2019. 86 f. Tese (Mestrado em Modelagem Computacional de Conhecimento) – Instituto de Computação, Universidade Federal de Alagoas, Maceió, 2013. Disponível em: http://www.repositorio.ufal.br/bitstream/riufal/6174/1/Categoriza%c3%a7%c3%a3o%20de%20textos%20por%20aprendizagem%20de%20m%c3%a1quina.pdf. Acesso em: 15 set. 2020.

[27] SILVA, M. J. e SCHIMIGUEL, J. IDENTIFICAÇÃO DE DOÊNÇAS EM PLANTAS POR MEIO DE PROCESSAMENTO DE IMAGENS: Redes Neurais Convolucionais como Auxílio à Agricultura. Revista Ubiquidade, ISSN 2236-9031 – v.3, n.1 – jan. a jun. de 2020, p. 91

[28] SIMÕES, C. C. **Reconhecimento de logotipos usando deep learning e recuperação de imagem baseada em conteúdo**. 59 f. Trabalho de Graduação (Bacharelado em Computação) – Departamento Acadêmico de Computação, Universidade Federal do Paraná, Medianeira, 2018.

[29] SOARES, Lirian Sousa. Imposto sobre Serviço de Qualquer Natureza. Base de cálculo. Empresa prestadora de serviços, com preponderância de mão-de-obra. **Revista Jus Navigandi**, ISSN 1518-4862, Teresina, ano 10, n. 646, 15 abr. 2005. Disponível em: https://jus.com.br/artigos/6582. Acesso em: 21 jan. 2021.

[30] STANGE, R. L. **Adaptabilidade em aprendizagem de máquina: conceitos e estudo de caso**. Tese. 2011. Disponível em https://www.teses.usp.br/teses/disponiveis/3/3141/tde-02072012-175054/publico/Dissertacao_RLStange_2011_Revisada.pdf. Acesso em 23 nov 2021.

[31] TORTORELLI, H. F. **Estratégias de Negócios Internacionais: O caso da empresa Amazon.com.** (Tese) dispónível em: https://ubibliorum.ubi.pt/bitstream/10400.6/10005/1/6226_13259.pdf. Acesso 23 nov 2021.

[32] POWERS, D. M. W. **Evaluation: From Predcision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation**. J. Mach. Learn. Technol, v. 2, n. 1, p. 37- 63, 2011.