Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Victor Ribeiro de Azevedo

Identificação do perfil de clientes utilizando Redes Neurais Convolucionais

Rio de Janeiro

2020

Victor Ribeiro de Azevedo

Identificação do Perfil de Clientes Utilizando Redes Neurais Convolucionais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof.ª Dr.ª Nadia Nedjah

Orientadora: Prof.ª Dr.ª Luiza de Macedo Mourelle

Rio de Janeiro

2020

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

A994 Azevedo, Victor Ribeiro de.
Identificação do perfil de clientes utilizando redes neurais convolucionais / Victor Ribeiro de Azevedo. – 2020.
104f.

Orientadoras: Nadia Nedjah, Luiza de Macedo Mourelle.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia eletrônica - Teses. 2. Redes neurais (Computação) - Teses. 3. Aprendizado do computador - Teses. 4. Algorítmos - Teses. 5. Sistemas de reconhecimento de padrões - Teses I. Nedjah, Nadia. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 004.5

Bibliotecária: Júlia Vieira – CRB7/6022

Assinatura

Data

Victor Ribeiro de Azevedo

Identificação do Perfil de Clientes Utilizando Redes Neurais Convolucionais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 13 de fevereiro de 2020.

Banca Examinadora:

Prof.ª Dr.ª Nadia Nedjah (Orientadora)
Faculdade de Engenharia, UERJ

Prof.ª Dr.ª Luiza de Macedo Mourelle (Orientadora)
Faculdade de Engenharia, UERJ

Prof. Dr. José Gabriel Rodríguez Carneiro Gomes
Programa de Engenharia Elétrica, COPPE/UFRJ

Prof.ª Dr.ª Fernanda Duarte Vilela Reis de Oliveira
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET-RJ

Rio de Janeiro
2020

DEDICATÓRIA

Dedico este trabalho à minha família e à minha esposa Lysandra que me apoiaram em todos os momentos para que pudesse alcançar a conclusão deste projeto. Agradeço a minha mãe, Maria Antonietta, pelo apoio em cada momento desta trajetória. Ao meu pai, Glicério Azevedo, pelo exemplo e todo incentivo aos estudos desde o início de minha formação e ao grande exemplo para que eu me espelhasse. Dedico, em especial aos meus irmãos, Rodrigo Azevedo e Tatiana Campos que sempre foram exemplos e me apoiaram.

AGRADECIMENTOS

Sou grato a Deus pela perseverança para que pudesse alcançar a conclusão deste projeto. Agradeço a minha família e esposa por me darem todo o apoio para atingir este objetivo.

Sou grato às professoras Nadia Nedjah e Luiza de Macedo Mourelle pela orientação e todo aprendizado que me foi passado ao longo de todo o desenvolvido deste trabalho.

Agradeço aos meus colegas de mestrado Cesar Affonso, Thiago Guimarães e Ramon Alves pela troca de conhecimentos e pelo companheirismo ao longo de todo o mestrado.

Agradeço a SulAmérica Seguros S.A. em especial à Rodrigo Frahlich e Flávia Santos, e a Intelie Soluções em Informática S.A. em especial à Hubert Fonseca, por permitirem a minha eventual ausência no trabalho para que concluísse esta etapa de minha formação.

Agradeço a todos os professores do mestrado por todo conhecimento transmitido de forma plena. Agradeço ao PEL-UERJ por toda infraestrutura e toda coordenação para que eu pudesse desenvolver este projeto.

Encontraremos um caminho ou faremos o nosso (Aut viam inveniam aut faciam).

Aníbal Barca

RESUMO

AZEVEDO, Victor Ribeiro de. *Identificação do Perfil de Clientes Utilizando Redes Neurais Convolucionais*. 2020. 88f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

O grande volume de informações digitais na forma de imagens e vídeos nas redes sociais demonstra a necessidade do desenvolvimento de novas técnicas para extrair informações relevantes aos perfis dos usuários. Milhares de imagens estão sendo publicadas em redes sociais diariamente. Com isto, a classificação de imagens de usuários nas redes sociais, é um processo relevante para empresas no contexto de segmentação dos clientes. Neste trabalho são utilizadas técnicas de inteligência computacional, em particular a aprendizagem profunda, a fim de prever o interesse de usuários de redes sociais em determinadas categorias de produtos. Com a identificação do perfil de clientes é possível definir potenciais consumidores de produtos, permitindo a elaboração de propagandas direcionadas. Nesta dissertação, são aplicadas a aprendizagem profunda e algoritmos de otimização de redes neurais convolucionais para desenvolver modelos de classificação de imagens de redes sociais, levando a identificação do perfil do usuário. É realizada a comparação do desempenho dos cinco algoritmos mais recentes de otimização da taxa de aprendizagem adaptativa para redes neurais artificiais. São eles, os algoritmos de descida do gradiente: estocástico (*SGD*) e adaptativo (*AdaGrad*), assim como os algoritmos baseados na estimativa de momento adaptativo usando: a raiz do valor quadrático médio dos gradientes (*RMSprop*), as médias móveis (*Adam*) e a norma infinita (*Adamax*). Dos cinco algoritmos abordados, o algoritmo de estimativa de momento adaptativo denominado *Adam* demonstra superioridade atingindo 94% de acurácia de aproximadamente 99% de taxa de acerto em determinadas classificações de imagens, demonstrando ser o algoritmo mais eficiente para treinamento de redes neurais e, portanto, permitindo prever de uma maneira mais precisa o interesse e o perfil do usuário.

Palavras-chave: Redes Neurais Convolucionais; Aprendizagem Profunda; Algoritmos de Otimização; Identificação do perfil de clientes; Reconhecimento de padrões em imagens;

ABSTRACT

AZEVEDO, Victor Ribeiro de. *Identification of Client Profile Using Convolutional Neural Networks*. 2020. 88f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

The large volume of digital information in the form of images and videos on social networks demonstrates the need for the development of new techniques to extract relevant information from users. In this work computational intelligence and deep learning techniques are used to predict the interest of social network users in certain categories. Thousands of images are being posted to social networks daily, so classify images from social network users is a relevant process for businesses in the context of customer segmentation. Identifying the customer profile can helps to define potential product consumers and create targeted advertisements. The technology applied consists of using Deep Learning and convolutional neural network optimization algorithms to develop different social image classification models. The performance of the following adaptive learning rate algorithms of artificial neural networks is compared: stochastic gradient descent, adaptive gradient descent, estimation of adaptive momentum and it variation based on the infinite norm and the mean square root of the gradients. The main objective of this work is to use the optimization algorithms to train convolutional neural networks and to verify if these methods are more efficient than the traditional training algorithms, such as Backpropagation and Gradient Descent. Of the 5 algorithms addressed, the adaptive moment estimation algorithm textit Adam demonstrates superiority reaching 94% accuracy of approximately 99% accuracy rate in certain image classifications proving to be the best algorithm for predicting interest profile.

Keywords: Convolutional Neural Networks; Deep learning; Optimization algorithms; Customer profile identification; Image Pattern recognition;

LISTA DE FIGURAS

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

AC	Acurácia
Adam	Adaptive Moment Estimation
AdaMax	Adaptive Stochastic Gradient Descent
AdaGrad	Adaptive Gradient Algorithm
AMSGrad	Exponential Moving Avarage Variant
AdaDelta	Adaptive Learning Rate Method
BN	Batch Normalization
BoW	Bag of Words
BoVW	Bag of Visual Words
CNN	Convolutional Neural Network
Conv	Convolução
Dim	Dimensão
F1	F-Média
HSV	Hue Saturation Value
LDA	Latent Dirichlet Allocation
MP	Max Pooling
MKF	Multiple Kernel Fusion
MLP	Multi Layer Perceptron
MSE	Mean Squared Error

LISTA DE SIGLAS

NF	Número de filtros
NAG	Nesterov Accelerated Gradient
Nadam	Nesterov accelerated Adam
PS	Precisão
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
PLSA	Probabilistic Latent Semantic Analysis
RC	Sensibilidade
RL	Reinforcement Learning
RGB	Red Green Blue
RNA	Redes neurais Artificiais
ReLU	Rectified Linear Unit
RMSProp	Root Mean Square Propagation
SP	Especificidade
SGD	Stochastic Gradient Descent
SIF	Scale Invariant Feature
SVM	Support Vector Machine
SLIC	Simple Linear Iterative Clustering
TC	Camada Totalmente Conectada
VGG	Visual Geometry Group
VGGNet	Very Deep Convolutional Networks for Large Scale Image Recognition

SUMÁRIO

SUMÁRIO

INTRODUÇÃO

A identificação do perfil de clientes é uma abordagem relevante visto que há um grande volume de informações digitais na forma de imagens e vídeos nas redes sociais (ROSER; ORTIZ-OSPINA, 2019). Torna-se necessário o desenvolvimento de novas técnicas para extrair informações relevantes dos usuários. Milhares de imagens estão sendo publicadas em redes sociais diariamente, com isto, a classificação de imagens de usuários de redes sociais, se demonstra um processo relevante para empresas. Com a segmentação dos usuários é possível definir potenciais consumidores e clientes de seus produtos, ou ainda, criar propagandas direcionadas para públicos distintos.

Este trabalho foi idealizado para ser mais uma contribuição para a comunidade de pesquisadores que buscam o aprimoramento das técnicas para identificação do perfil de clientes. O objetivo é treinar algoritmos de aprendizado de máquina utilizando técnicas de otimização de rede neural artificial e comparar a performance dos modelos para classificação de imagens. Com isto é possível verificar o melhor algoritmo de otimização em relação à performance e assertividade na classificação de imagens para a identificação do perfil de clientes. Vale ressaltar que este trabalho foi iniciado e motivado em (AZEVEDO, V. R.; NEDJAH, N.; MOURELLE, L. DE M., 2019) demonstrando a relevância do estudo e abordagem proposta.

Em (ASLAM, 2019) foi estimado que o *Instagram*, uma das redes sociais mais populares, possuía mais de 1 bilhão de usuários ativos mensais em 2018. Essa quantidade é mais do que o triplo dos usuários ativos mensais do *Twitter*. Não obstante, o *Instagram* está atrás dos usuários do *WhatsApp* e do *Facebook Messenger*. Conforme relatado em (ASLAM, 2019), todos os dias, os usuários do *Instagram* publicam em média mais de 100 milhões de fotos. É essencial salientar que 72% usuários do *Instagram* dizem que compraram um produto que viram no aplicativo (KEYES, 2017).

De acordo com (ROSER, 2019) aproximadamente 4 bilhões, isto é, 53% da população mundial têm acesso a internet, sendo que 3,1 bilhões (42%) são usuários ativos de redes sociais. Cerca de 66% dos brasileiros têm acesso à internet (ROSER, 2019). O Brasil está em terceiro lugar no mundo quanto ao tempo de uso da internet diariamente com 9 horas e 14 minutos.

Junto ao *Facebook*, a plataforma mais utilizada pelas empresas nacionais é o *Instagram* e o *Whatsapp* que possui aproximadamente 130 milhões de usuários ativos. Estudos realizados indicam que o *Instagram* foi citado como a rede preferida do público com 47% de aprovação e o *Facebook* fica em segundo lugar com 29,6%. Vale mencionar que o *Instagram* mostra-se como o principal meio de negócios entre as redes sociais (GREENFIELD, 2019).

Aproximadamente 76% dos usuários do *Facebook* realizam uma visita por dia nesta rede social. Já no *Instagram*, 51% dos usuários acessam diariamente a plataforma (BATARJAV; DANTU, 2011). Cerca de 95% das empresas nacionais possuem redes sociais, conforme o relatório da (GREENFIELD, 2019), sendo que 62% consideram a importância

dessas plataformas para o seu resultado. O relatório aponta ainda que 85,3% buscam as redes sociais para aumentar a sua visibilidade online, enquanto 64,8% procuram interagir com o seu público.

Há um crescimento acentuado nas redes sociais a cada dia. Até julho de 2018, o *Facebook* possuia cerca de 2,2 bilhões de usuários ativos mensais. Aproximadamente 336 milhões de usuários estão ativos mensalmente no *Twitter*. Ainda em 2018, os usuários ativos mensais do *LinkedIn* atingiam a marca de 294 milhões, para o *Instagram* era de cerca de 1,0 bilhão, enquanto o site de mídia social recém-lançado *Pinterest* possuía 200 mil usuários ativos mensais (ROSER, 2019).

Estes dados demonstram a relevância deste estudo e evidenciam que segmentar os usuários por interesse é uma estratégia relevante de vendas de produtos para empresas no contexto de redes sociais. Redes neurais convolucionais estão dentre as principais técnicas utilizadas para aplicações de visão computacional. As redes neurais foram inspiradas pela operação de um neurônio biológico. Espera-se que a rede neural artificial seja capaz de reconhecer e aprender novos padrões visuais assim como o cérebro é capaz de fazer.

O aprendizado profundo é um subconjunto de aprendizado de máquina em inteligência artificial. Neste contexto são desenvolvidos modelos de alta complexidade treinados com grandes quantidades de dados para reconhecer inúmeros tipos de padrões. As redes neurais convolucionais são modelos treinados com técnicas de aprendizagem profunda. Para o treinamento de uma rede neural como esta, são necessários não só muitas imagens mas um bom algoritmo de otimização, para que exista uma boa busca pelos melhores parâmetros do modelo. Como em redes neurais de menor complexidade, geralmente são utilizados algoritmos como o *Backpropagation* e a descida do gradiente para o treinamento de redes neurais convolucionais. No entanto, outros ramos da inteligência artificial fornecem algoritmos de otimização ainda mais eficientes, que podem ser combinados com estes modelos de reconhecimento de padrões para produzir melhores resultados

Como objetivo, este estudo visa treinar modelos de rede neural convolucional utilizando 5 algoritmos de otimização de redes neurais artificiais. Com isso, é esperado comprovar a superioridade do uso destes algoritmos em relação aos métodos tradicionais de treinamento, que utilizam o *Backpropagation*. Por outro lado, é desejado verificar qual é o algoritmo de otimização que gera os melhores resultados no treinamento rede neural.

Nesta dissertação, informações sobre sistemas de recomendação e perfil de usuários são apresentadas no Capítulo 1. No Capítulo 2 são descritos alguns dos trabalhos relacionados a esta pesquisa, onde outros métodos e resultados são apresentados para aprofundar o contexto de solução do problema em análise. Em seguida, no Capítulo 3, os métodos de otimização de rede neural artificial são descritos e cada algoritmo utilizado no trabalho é apresentado. Ainda na fundamentação teórica deste trabalho, o Capítulo 4 aborda os principais aspectos em torno da aprendizagem profunda, descrevendo os métodos propostos para o desenvolvimento da solução deste trabalho.

No Capítulo 5 são apresentados todos os procedimentos envolvidos nos experimentos e resultados deste trabalho. Nele são abordados os passos percorridos desde a pesquisa dos dados de treinamento até a obtenção dos resultados finais bem como cada configuração da rede neural convolucional.

Por fim, o Capítulo 6 apresenta as conclusões obtidas com o trabalho desenvolvido. Este capítulo também sugere possíveis modificações e melhorias para trabalhos futuros em aplicações semelhantes.

Capítulo 1

SISTEMAS DE RECOMENDAÇÃO

ESTE capítulo apresenta conceitos relacionados a sistemas de recomendação de produtos, filtragem de informação e perfis de clientes. É abordada a importância do desenvolvimento de ferramentas que possam oferecer apoio a decisão bem como a relevância deste estudo na área de inteligência computacional.

1.1 Sistemas de recomendação de produtos

Com o avanço da tecnologia, os sistemas de recomendação de produtos estão sendo aprimorados utilizando técnicas de inteligência computacional. As redes sociais permitem o acesso a informações dos usuários, o que possibilita o desenvolvimento de algoritmos para inferir o interesse dos usuários.

Os sistemas de recomendação são ferramentas automatizadas que utilizam algoritmos de filtragem para coletas de dados para realizar recomendações inteligentes com base no perfil de cada usuário. O número de usuários que utilizam a internet cresce constantemente e, com base na relevância que as redes sociais possuem atualmente, as recomendações devem ser mais precisas, conforme as preferências dos clientes. Com isto, os sistemas de recomendação impulsionam as vendas e garantem o engajamento do usuário com o conteúdo, levando o cliente a encontrar o que deseja.

Os avanços tecnológicos possuem um papel importante no processo de compra e as redes sociais influenciam de forma direta na tomada de decisão. Grande parte dos clientes utiliza as redes sociais para buscar informações sobre qual o melhor produto irá consumir.

As redes sociais mais usadas do mundo são o *Facebook*, *Youtube*, *Facebook Messenger*, *WeChat* e *Instagram*, sendo que nos três primeiros meses de 2012, o *Facebook* atingiu 1 bilhão de usuários. No fim de 2018, a rede social chegou a 2,26 bilhões de usuários onde o Brasil destaca-se com 127 milhões de usuários. A Figura 3 ilustra a distribuição de usuários em redes sociais.

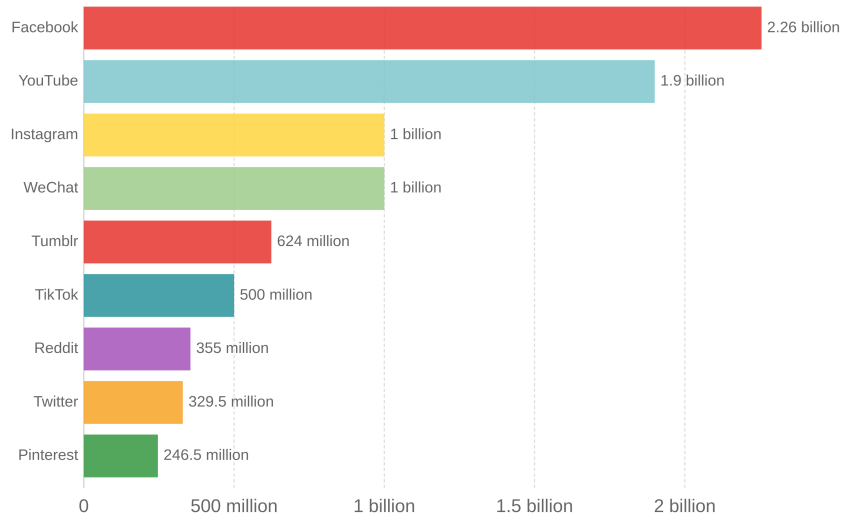


Figura 1: Usuários ativos ao longo de um mês em redes sociais no mundo [fonte (ROSER, 2019)]

Os sistemas de recomendação de produto foram elaborados especificamente com o intuito de gerar uma experiência de compra satisfatória, pois o ativo mais importante de um negócio é o cliente. Vale mencionar que estes sistemas possuem o objetivo de reduzir a informação apresentada ao usuário, fazendo a filtragem de produtos com base no perfil de interesses do usuário. Sistemas de recomendação visam a oferecer, a partir de grandes volumes de informações, aquilo que pode interessar especificamente ao usuário. As técnicas, em geral, dependem das contribuições dos indivíduos na avaliação da informação.

É essencial mencionar a grande relevância da análise dos sistemas de recomendação baseado nas informações disponíveis dos usuários em redes sociais. Com isto torna-se importante o desenvolvimento de ferramentas que possam executar essa tarefa. Tais sistemas são essencialmente colaborativos, pois permitem o compartilhamento de informações dos usuários sobre produtos consultados ou adquiridos, permitindo também o registro de suas impressões sobre esses itens.

De acordo com (MOLENAAR, 2015) aproximadamente 35% da arrecadação da *Amazon* e cerca de 33% do conteúdo assistido na *Netflix* tem por base os algoritmos de recomendação. Nesse contexto, destacam-se as principais redes sociais como o *Facebook*, *Twitter* e *Instagram*. Os sistemas de recomendação são utilizados em diversos domínios para a recomendação de vários produtos e inclusive pessoas. Redes sociais como o *LinkedIn* e o *Facebook* utilizam estes tipos de sistemas para sugerir novos contatos.

Um sistema de recomendação geralmente baseia-se no perfil dos usuários e em uma lista de produtos adquiridos pelo usuário previamente. Esta modalidade de sistema já pode ser encontrada em grandes empresas de *e-commerce*, com a finalidade de aumentar suas vendas e atender melhor os seus clientes.

Os sistemas de recomendação apresentam objetivos técnicos comuns com três características: relevância, inovação e aumento da diversidade. A relevância consiste na recomendação ao usuário de itens que tem importância de acordo com suas preferências. O aspecto inovador entrega produtos que o usuário não viu anteriormente fora do que estava sendo procurado, pois uma recomendação repetida pode fazer com que o usuário abandone o sistema. O aumento da diversidade fornece uma lista de recomendações com itens variados, assegurando que o usuário não fique entediado com recomendações repetidas.

Neste contexto vale ressaltar que os sistemas de recomendação empregam técnicas de filtragem de informação e inteligência computacional para a recomendação de produtos. Conforme observado em (CASTILLO, 2013), os sistemas de recomendação representam um conjunto de algoritmos que utilizam técnicas de aprendizagem de máquina. Realizam a recuperação de informações para gerar as recomendações com base em algum tipo de filtragem, destacando-se a filtragem colaborativa, com base no conteúdo e a filtragem demográfica.

1.1.1 Filtragem colaborativa

Dentre as técnicas de sistemas de recomendação, utiliza-se a filtragem colaborativa que teve inicio na década de 1990 e mantém até hoje as características principais (HERLOCKER et al., 2004). A filtragem colaborativa aplicada aos sistemas de recomendação é uma técnica amplamente utilizada no ambiente onde cada usuário colabora com suas avaliações para o desempenho do sistema. Essa técnica apresenta um modelo que permite a análise de produtos a serem recomendados sem atentar para o conteúdo dos itens, mas sim as avaliações recebidas (ROLIM et al., 2017).

1.1.2 Filtragem demográfica

Conforme relatado em (AGGARWAL et al., 2016), as informações demográficas dos usuários são utilizadas para encontrar outros usuários que tenham semelhanças e fazer recomendações. Segundo (ROLIM et al., 2017), a filtragem demográfica utiliza elementos como idade, gênero, localização, entre outros. Considera-se que esta técnica é a mais fácil e a menos eficaz, pois não considera dados específicos do usuário ao contrário da filtragem colaborativa e filtragem baseada e conteúdo. Ao invés disso, os usuários são separados em grupos através de um critério de seleção demográfica.

1.1.3 Filtragem baseada em conteúdo

A filtragem baseada em conteúdo considera o conteúdo que foi acessado ou avaliado, anteriormente, pelo usuário. Conforme observado em (ROLIM et al., 2017), na filtragem com base no conteúdo recomendam-se itens semelhantes ao item de interesse do usuário.

Em (ROLIM et al., 2017) os autores relatam que embora esse sistema seja bastante utilizado, apresenta algumas limitações como a análise limitada ao conteúdo do item recomendado e super especialização, pois só recomenda itens acessados ou avaliados positivamente.

Com base nas avaliações que o usuário expressa sobre itens vistos em determinada rede social e na descrição dos mesmos, é possível desenvolver algoritmos que identifiquem o perfil que retrate os tipos de informação nos quais o usuário demonstra interesse. O modo de representação do perfil do usuário depende das técnicas de aprendizagem de máquina e de recuperação de informação utilizadas.

Diversos algoritmos de aprendizagem de máquina e de recuperação da informação podem ser utilizados para aprender um perfil de usuário. Assim, o perfil pode ser utilizado para gerar um classificador capaz de prever a relevância de um produto a partir dos seus atributos ou ainda podem ser deduzidas semelhanças entre itens com base no conhecimento do domínio do sistema, transformando a filtragem em uma aplicação.

Neste trabalho a filtragem baseada em conteúdo é utilizada para analisar a relevância de determinadas categorias de imagens postadas por uma amostra de usuários de redes sociais.

1.1.4 Usuários de redes sociais

O *Instagram* é um dos aplicativos mais populares do mundo. O crescimento dos usuários desta rede social afeta também a quantidade de dados do *Instagram*. De acordo com (KUNCORO; ISWANTO, 2015), pesquisadores têm realizado estudos considerando os sistemas de recomendação para detectar fotos com base na localização, agrupar o conteúdo da foto, estratégias de publicidade com base nos tipos de usuários, dentre outros com base em texto, imagem e vídeo.

Existe grande importância em detectar os perfis de usuário das redes sociais para que os sistemas de recomendação possam atuar de forma mais eficaz apresentando conteúdos relevantes para cada usuário.

As empresas aumentam as oportunidades de disseminar informações e envolver o público em suas ações utilizando redes sociais. As imagens apresentadas nas redes sociais estão entre os elementos mais significativos que envolvem os usuários das redes sociais.

Aproximadamente 45% da população é ativa nas redes sociais, sendo que houve um aumento de 13% no acesso a dispositivos móveis. Jovens com menos de 19 anos acessam mais a internet através de dispositivos móveis. Pessoas com mais de 40 anos usam mais *tablet* ou *SmartTV* (ROSER, 2019). Na América Latina, o Brasil é o país com mais usuários num total de 93,2 milhões.

O *Facebook* atinge a marca de 1 bilhão de usuários ativos todos os dias e seu aplicativo de bate-papo possui 900 milhões de usuários. O *Whatsapp* já tem 1 bilhão de usuários, com mais de 100 milhões de usuários no Brasil. O *Instagram* ultrapassa os 500 milhões de usuários, com 35 milhões brasileiros. Todos os dias 300 milhões de pessoas acessam essa rede social, sendo que 95 milhões de fotos são postadas todos os dias. Segundo o relatório (GREENFIELD, 2019), 92,1% do total de usuários de redes sociais utilizam o *Facebook* e 92,5% o *Instagram*, enquanto 72,3% usam o *Youtube*. Entre os entrevistados, 91,3% acessam suas redes sociais diariamente e 38,3% gasta, em média, mais de 4 horas por dia navegando nas redes sociais. Em contrapartida, 32,4% pensam em sair de alguma rede social. Foram identificadas faixas etárias variadas e uma participação equilibrada de homens e mulheres. Quanto à escolaridade, 90% dos participantes têm ao menos ensino superior incompleto. Verificou-se um aumento do uso do *Instagram* e do *Youtube* e todos os participantes da pesquisa possuem perfil em pelo menos uma rede social.

Conforme observado em (ASLAM, 2019), o *Instagram* é a rede social que mais cresce em todo mundo. O Brasil é o segundo país do mundo com mais usuários nesta rede social, ficando atrás apenas dos Estados Unidos. Dados do *Instagram* mostram que 85% dos usuários seguem um perfil comercial, os quais são quase 20 milhões (GREENFIELD, 2019).

Nota-se a relevância na compreensão mais profunda nos tipos de fotos e usuários ativos nas redes sociais. Com isto é possível analisar como essas diferenças entre usuários possuem relação com o conteúdo das fotos postadas.

Em (CAMPANA; DELMASTRO, 2017) é relatado que as mídias sociais representam uma fonte importante de informações para caracterizar conteúdos e interesses dos usuários. O sistema de recomendação pode explorar essas informações para personalizar ainda mais as sugestões e melhorar o processo de recomendação.

Com base nessas informações, vale mencionar a importância no conhecimento dos dados para descrever e prever o comportamento futuro dos usuários. Na busca por esses objetivos, podem ser utilizadas diferentes estratégias como a aplicação de algoritmos específicos que venham a extrair padrões desses dados. Neste contexto destacam-se os algoritmos associação, clusterização e classificação.

A associação é uma tarefa que tem a finalidade de encontrar relacionamentos ou padrões que ocorrem com frequência em um conjunto de dados. Esta estratégia é muito utilizada por empresas para definir o perfil do consumidor e como serão distribuídos os produtos nas vitrines.

A clusterização ou análise de agrupamentos consiste na busca de semelhanças entre os dados tal que permita definir um conjunto limitado de classes ou categorias que os reúna e os descreva. A principal diferença desta abordagem para a classificação é que na clusterização não há o conhecimento prévio sobre o número de classes.

A classificação refere-se ao processo de encontrar um conjunto de funções que distinguem classes, utilizando um modelo para prever classes de objetos para possibilitar mapear o perfil de um cliente.

1.2 Aprendizagem profunda

A aprendizagem de máquinas é um ramo da engenharia da computação onde são desenvolvidas técnicas focadas em permitir que o computador realize tarefas humanas. A aprendizagem profunda permite a solução de problemas de alta complexidade utilizando algoritmos de aprendizagem de máquina.

Redes neurais artificiais são um subconjunto de técnicas de aprendizagem de máquina, onde cada elemento básico é um neurônio cuja inspiração é o neurônio biológico. Cada neurônio da rede neural está conectado com os demais. A rede tem como objetivo aprender a realizar uma determinada tarefa.

A aprendizagem profunda (*Deep Learning*) é uma subcategoria do aprendizado de máquinas onde modelos de redes neurais de alta complexidade buscam reconhecer padrões. A principal diferença entre as redes neurais comuns e as de aprendizagem profunda é a quantidade de camadas presentes em cada modelo. Assim como nas redes neurais artificiais, nas redes de aprendizagem profunda são realizadas operações matemáticas de soma e multiplicação em suas camadas para ajuste de seus parâmetros e assim aprender a realizar algum processo após o seu treinamento.

No contexto de visão computacional, o reconhecimento de padrões em imagens e vídeos pode ser considerado uma das aplicações da aprendizagem profunda mais utilizadas onde modelos de alta complexidade são desenvolvidos para detectar objetos em imagens ou vídeos.

1.3 Considerações finais

Com o conteúdo deste capítulo é possível verificar a importância de sistemas de recomendação e ter noção da dimensão dos números atrelados a redes sociais e a identificação do perfil de clientes. É possível constatar a real grandeza do problema, sendo importante o desenvolvimento de ferramentas que possam oferecer qualquer tipo de auxílio no apoio a decisão. Torna-se evidente que a crescente expansão das redes sociais requer o desenvolvimento de algoritmos que identifiquem o perfil dos clientes auxiliando no desenvolvimento

de sistemas de recomendação. Este capítulo demonstra que existe uma grande relevância em detectar os perfis de interesse de usuários das redes sociais para que sistemas de recomendação possam atuar de forma eficaz.

Capítulo 2

TRABALHOS RELACIONADOS

2.1 Introdução

Este capítulo apresenta trabalhos relacionados ao problema de classificação de imagens de usuários de redes sociais utilizando algoritmos de inteligência computacional a fim de inferir seus interesses em determinados produtos.

São elucidadas neste capítulo a metodologia utilizada em cada artigo assim como suas semelhanças e diferenças no escopo deste trabalho. Destacam-se também os desafios enfrentados pelos autores e os resultados obtidos pelos mesmos. Acredita-se que os resultados alcançados pelos trabalhos apresentados contribuem para base teórica deste projeto. Os trabalhos são dispostos de acordo com as técnicas de aprendizagem profunda, redes neurais convolucionais e outros métodos de inteligência computacional para classificação de imagens de redes sociais.

2.2 Aprendizagem profunda

O aprendizado profundo é uma técnica de inteligência artificial que são desenvolvidos modelos de alta complexidade amplamente utilizado na área de visão computacional. Um dos primeiros projetos utilizando Redes Neurais Convolucionais (*Convolutional Neural Network* - CNN) foi a LeNet proposta em (LECUN et al., 1998) abordando o reconhecimento de caracteres manuscritos. Neste projeto foi demonstrado o grande potencial das CNN como extratoras de características de imagens.

2.2.1 Inferência do interesse de usuários com CNN e BoW

Em (CINAR; ZOGHBI; MOENS, 2015) é realizada a inferência do interesse de usuários da rede social *Pinterest* baseado na classificação de textos e imagens. O trabalho visa a recomendação de produtos de determinadas marcas baseado na classificação de imagens postadas pelos usuários como uma estratégia e análise de *marketing*. Os autores relatam que um dos desafios do projeto é a interpretação do interesse do usuário quando a imagem ou comentário postado na rede social contém o produto a ser classificado, porém o usuário refere-se a outro assunto. Diante deste fato, foi realizada a comparação dos resultados utilizando um espaço multi-modal, combinando textos e imagens, e uni-modal, com textos ou imagens, a fim de alcançar uma melhor acurácia no resultado.

Os autores utilizam CNNs para classificação das imagens, a técnica de *Bag of Words* (BoW) para representação textual e a técnica *Bag of Visual Words* (BoVW) para

representação da imagem como um mapa de características. O BoW é frequentemente utilizado em métodos de classificação de documentos e textos, onde a frequência de ocorrência de cada palavra é vista como uma característica utilizada para treinar um classificador. Analogamente, no BoVW, cada imagem é representada como um histograma de ocorrência de palavras visuais (pequenas porções da imagem, capazes de descrever de forma satisfatória a imagem como um todo). Estes histogramas são utilizados para treinar um classificador. Este trabalho utilizou o modelo de vetor de palavras denominado *word2vec* obtido no *Google News dataset* e a CNN pré treinada na competição *ImageNet classification challenge* (RUSSAKOVSKY et al., 2015). Foi realizado o ajuste dos pesos na última camada da rede neural convolucional e a redução de 1000 classes de saída para 32 tipos de categorias pré-definidas. Os autores relatam a utilização da técnica de aprendizado supervisionado de máquina de vetores de suporte (*Support Vector Machine* - SVM) na classificação.

Para obtenção das imagens e dos textos como fonte de dados dos usuários foi realizado um programa que percorreu 650 usuários obtendo 443.000 conjuntos de textos e imagens denominados *pins* no *Pinterest*. Em seguida os dados foram divididos em treinamento, validação e teste para realização de onze experimentos utilizando diferentes combinações dos algoritmos supracitados.

A representação do interesse dos usuários foi feita com uma distribuição de frequência das categorias pré-definidas. Para medição do desempenho do modelo foi utilizada a acurácia e a taxa de acerto de classificação. Os autores demonstram que a utilização do processamento de imagem em um espaço multi-modal com imagens e textos aumenta a acurácia dos resultados da classificação quando comparado a espaços uni-modais.

2.2.2 Aprendizagem profunda para detecção de objetos

Em (PATHAK; RAUTARAY, 2018) é feita uma abordagem e comparação de diferentes métodos de aprendizagem profunda para detecção de objetos em imagens. Os autores citam e comparam relevantes bibliotecas disponíveis para aprendizagem profunda bem como suas linguagens utilizando fontes de dados de competições mundiais.

Este projeto demonstra os passos para detecção e classificação de um objeto em uma imagem com CNNs. Os passos citados pelo autor consistem na verificação do objeto na imagem, localização do objeto na imagem, classificação em determinadas categorias e descrição do relacionamento do objeto com o contexto.

As fontes de dados de competições mundiais citadas são: *Microsoft COCO*, *ImageNet Dataset*, *CIFAR-10 and CIFAR-100*, *CUB-200-2011*, *Caltech-256* e *ILSVRC*. As principais bibliotecas para classificação de imagens citadas são: *Caffe, Microsoft Cognitive, TensorFlow, Theano, Torch, Keras, IBM Watson Vision Recognition Service, Amazon Rekognition Google Cloud Vision API* (PATHAK; RAUTARAY, 2018).

Os autores mencionam também a comparação de métodos de aprendizagem profunda utilizados para detecção de objetos em determinadas regiões de uma imagem tais como o algoritmo de otimização por enxame de partículas (*Particle Swarm Optimization* - PSO). O PSO visa a otimização de uma função objetivo e é utilizado no algoritmo de descida do gradiente na busca do erro mínimo em uma rede neural artificial. Os autores demonstram que a utilização de algoritmos de inteligência de enxame como o PSO para o ajuste de parâmetros de redes neurais pode ser uma ótima ferramenta para o aumento de taxa de acerto na classificação.

As camadas de agregação (*pooling layer*) são responsáveis por reduzirem a dimensionalidade das matrizes e a complexidade na rede neural. Os autores comparam os seguintes tipos de camadas: *Max pooling, Average pooling, Deformation pooling, Spatial pyramid pooling, Scale dependent pooling*. Acredita-se que este trabalho contribui de forma a definir as melhores ferramentas e linguagens para classificação de imagens bem como fontes de dados a serem utilizadas.

2.2.3 Classificação de logotipos com CNN

Em (PIMKOTE; KANGKACHIT, 2018) o autor apresenta uma abordagem para classificar logotipos de marcas usando CNNs. A classificação é binária entre imagens com conteúdo alcoólico ou não e a arquitetura da rede utilizada foi baseada em outras redes existentes: *AlexNet*, *VGG-19* e *GoogLeNet*.

Neste trabalho são apresentadas técnicas de medição de performance na classificação e pré-processamento das imagens. Os autores relatam que a base de dados foi coletada por eles em diversos *sites* devido a não terem encontrado uma única base com as imagens adequadas. Com isto foi necessário um pré-processamento para ajustar tamanho, resolução, nitidez e brilho das imagens. Os dados foram divididos entre treino (70%) e teste (30%). Os autores relatam a utilização das métricas de precisão (*precision*) e taxa de acerto (*recall*).

2.2.4 Classificação de imagens em redes sociais

Em (SEGALIN et al., 2017) é feita a inferência da personalidade de usuários da rede *Flickr* a partir de imagens de suas galerias. Foi investigada a interação entre os traços que as pessoas deixam nas redes sociais (postagens, fotos, perfis, etc.) e traços de personalidade. Os experimentos realizados com 60.000 imagens adicionadas aos favoritos por 300 usuários permitiram extrair características utilizando técnicas alternativas como a transformada *wavelet* destas imagens e segmentar os usuários.

Em (CHATZILARI et al., 2012) é feita a seleção de um conjunto de imagens da rede social *Flickr* para detecção de objetos. É feita a segmentação das imagens com o agrupamento de regiões das imagens utilizando o algoritmo de propagação por afinidade (*affinity propagation*). Os autores utilizam de aprendizagem supervisionada para classificação de objetos. Os autores citam a utilização do algoritmo de análise semântica probabilística latente (*probabilistic Latent Semantic Analysis* -PLSA) e o algoritmo de Alocação de Dirichlet Latente (*Latent Dirichlet Allocation* - LDA) para distribuição de probabilidade das características das imagens.

2.2.5 Classificação hierárquica

Em (VIEIRA; MOREIRA, 2017) utiliza-se o aprendizado profundo com CNNs em conjunto com Classificação Hierárquica para agrupar imagens de lojas virtuais (*e-commerce*) em categorias pré-definidas. A base de dados utilizada foi *CIFAR10*. A rede neural utilizada era composta de 20 camadas incluindo as de entrada e saída, o treinamento foi realizado com 500 épocas e a função de ativação utilizada foi a *sigmoide*.

Na classificação hierárquica multirrótulo o classificador processa e avalia todas as classes da hierarquia em uma única vez. Os autores demonstram que a acurácia na classificação obtida para um dos conjuntos de dados foi melhor utilizando a classificação hierárquica (94.77%) do que utilizando a classificação sem hierarquia (85.13%).

Os autores citam a camada de agregação (*max pooling*) como fator importante para prevenir o sobreajuste (*overfitting*) da rede, ou seja, para que o algoritmo tenha boa capacidade de generalização. O trabalho apresenta o método de classificação hierárquica não convencional.

2.2.6 Reconhecimento de produtos com CNN

Em (LI et al., 2017) é feito o reconhecimento de imagens de vestuários da base de dados *e-Clothing* utilizando CNNs. São utilizadas aproximadamente 350.000 imagens de produtos definindo 24 categorias previamente. O autor demonstra diferentes tipos de filtros na camada convolucional bem como a função objetivo da rede a fim de melhorar a convergência da Rede Neural Convolucional.

2.3 Utilização de outros métodos

Em (BALLAN et al., 2012) são realizadas classificações de objetos de redes sociais em 101 categorias. As bases de dados utilizadas são MICC-Flickr101 e *Caltech101* contendo imagens da rede social *Flickr*. O diferencial deste projeto é a utilização dos algoritmos de Alocação de Dirichlet Latente (*Latent Dirichlet Allocation* - LDA) e o algoritmo de otimização Fusão de múltiplos Kernels (*Multiple Kernel Fusion* - MKF) na classificação das imagens. Além disto também é utilizado o SVM e o BoW. Os autores relatam que a utilização do LDA, SVM e MKF na classificação reduziu o custo computacional. Demonstra-se também que a acurácia do resultado também é aumentada com a utilização do MKF. Desta forma este trabalho contribui apresentando técnicas não convencionais para classificação das imagens.

2.3.1 Predição de gênero com SIF

Em (YOU et al., 2014) é realizada a classificação binária de usuários da rede social Pinterest entre sexo masculino e feminino baseado nas suas imagens postadas. O diferencial deste projeto é que utiliza do algoritmo de Característica de Escala Invariante (Scale Invariant Feature - SIF) para descobrir características locais em cada imagem.

Para obtenção das imagens para fonte de dados dos usuários foi realizado um programa que percorreu 1500 usuários do *Pinterest* para obtenção do perfil do *Twiter*. A partir destes perfis foi possível obter o sexo de uma amostra selecionada aleatoriamente com 160 usuários (80 masculinos e 80 femininos). Foram definidas a partir destes usuários 33 categorias.

Os resultados de novas amostras foram medidos com a utilização das métricas acurácia (*accuracy*), precisão (*precision*), taxa de acerto (*Recall*) e *score* (*F-measure*). Acredita-se que este projeto contribui com a utilização da técnica não convencional SIF e com as métricas de desempenho na classificação.

2.3.2 Classificação com SLIC, SVM e PCA

Em (DIAS; TABB; MEDEIROS, 2018) é utilizada CNN para reconhecimento de flores em imagens. O autor utiliza do procedimento de Análise de Componentes Principais (*Principal Component Analysis* - PCA) para redução de dimensionalidade na classificação e do

algoritmo de Clusterização Linear iterativo (*Simple Linear Iterative Clustering* - SLIC) para o agrupamento de *pixel* similares em *superpixels*.

A sequência realizada para classificação elucidada compreende o agrupamento de regiões similares com o SLIC, extração de características com a CNN, utilização do PCA na classificação e por fim o algoritmo binário SVM classifica a presença ou não das flores. Os autores abordam diferentes desempenhos de classificação para utilização do sistema de cores HSV (*Hue Saturation Value*) e RGB (*Red Green Blue*).Os autores utilizam para medição do desempenho dos resultados as métricas de acurácia, precisão, taxa de acerto e *score*. Este projeto destaca-se pela utilização de diferentes sistemas de cores na classificação e na utilização da técnica SLIC na extração de características. Em (BERGMAN; NACHLIELI, 2011) são aplicadas diversas técnicas de segmentação de imagens baseadas nas cores e texturas das imagens para classificação de objetos. Em (LIU et al., 2018) também é utilizada a CNN para classificação de imagens de redes sociais utilizando o PCA para redução da custo computacional .

2.3.3 Classificação com Haartraining

Em (CHUNPRUGSUK et al., 2017) é feita a classificação de imagem de redes sociais para detecção de armas e objetos suspeitos. Os procedimentos realizados nas imagens neste trabalho são a detecção do objeto, extração de características, reconhecimento do objeto e classificação. O algoritmo utilizado neste trabalho foi o *Haartraining* e destaca-se pela sua alta performance e precisão. Neste algoritmo é realizada a varredura com uma janela na imagem para investigar objetos a partir de uma imagem de entrada. Os autores mencionam a transformação das imagens para escala de cinza e destaque de bordas nas imagens para melhor classificação porém não apresentam dados quantitativos dos resultados. O trabalho apresenta um método não convencional para detecção de objetos em imagens.

2.3.4 Detecção de comidas com Superpixel

Em (ZHENG; WANG; JI, 2016) é utilizada a técnica de segmentação de imagens utilizando *Superpixels* para o reconhecimento de imagens de comidas. A segmentação *Superpixel* é o agrupamento de pixels com cores similares com a finalidade de reduzir o número de nós da rede. A base de dados utilizada foi a *UEC Food dataset* contendo 10.000 imagens contemplando 100 categorias de comidas. Em cada imagem segmentaram-se 30 *Superpixels* e aplicou-se o PCA para redução da dimensionalidade na classificação

Os autores demonstram que a performance da classificação é maior utilizando a segmentação *Superpixels*. O trabalho apresenta abordagens relevantes para tratamento de imagens e redução da complexidade da rede neural utilizada.

2.4 Considerações finais

Neste capítulo, foram apresentados trabalhos relacionados ao problema de classificação de imagens de usuários de redes sociais utilizando algoritmos de inteligência computacional para inferir seus interesses. Foi possível identificar as diferenças e semelhanças com outros projetos bem como verificar os desafios e soluções desenvolvidas em outros trabalhos. Foi possível verificar as melhores ferramentas, linguagens e fontes de dados para serem utilizados. Ao observar outros trabalhos é possível propor soluções que ainda não foram

realizadas. Foi possível também verificar a implementação de vários modelos e algoritmos que auxiliam na obtenção de melhor acurácia dos resultados de classificação. Cada trabalho contribuiu com uma base sólida para o desenvolvimento deste estudo.

Capítulo 3

OTIMIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

NESTE capítulo serão abordados os conceitos e definições de rede neural artificial, redes neurais convolucionais, aprendizagem profunda, descida do gradiente e os algoritmos de otimização da descida do gradiente para o treinamento de redes neurais artificiais. São mencionados também conceitos de aprendizado de máquina e arquiteturas de redes neurais artificiais.

Este capítulo contribui para a comparação de diferentes algoritmos de otimização da descida de gradiente para classificação de imagens utilizados neste trabalho. Com comparação dos algoritmos é possível melhorar a qualidade da solução final utilizando os métodos mais eficientes para aprendizagem de máquina, buscando assim resultados no estado da arte.

3.1 Redes Neurais Artificiais

Redes neurais Artificiais (RNA) são modelos computacionais não lineares inspirados no cérebro humano, baseados em conjuntos de neurônios interconectados que procuram reproduzir características tais como aprendizado, associação e generalização (CROSS; HARRISON; KENNEDY, 1995). As RNAs possuem conexões interneurais objetivando armazenar e processar o conhecimento por meio do ajuste de parâmetros denominados pesos sinápticos. Os pesos são fatores que representam a influência de cada entrada na característica que está sendo identificada pelo neurônio.

O primeiro modelo de neurônio artificial foi proposto em (MCCULLOCH; PITTS, 1988) como uma unidade de processamento fundamental para a operação de uma rede neural. Além disto, algumas são aproximadores universais e possuem capacidade de classificar dados. Possuem aplicações em áreas como processamento de sinais, aproximação de funções, previsão de séries temporais, otimização, classificação e reconhecimento de padrões. O treinamento de redes neurais consiste em encontrar os pesos que levam um neurônio a classificar corretamente os dados.

A aprendizagem em RNAs é subdivida em não-supervisionada, por reforço e supervisionada e geralmente é formulada com a minimização de uma função erro. Na aprendizagem não supervisionada o objetivo é a organização dos padrões existentes nos dados por meio de agrupamentos consistentes por meio da correlação dos dados de entrada. Na aprendizagem por reforço um agente interage com o ambiente e é estimulado ou inibido a realizar ações. Após um determinado número de interações, o agente aprende a otimizar

suas ações e a se adaptar às mudanças do ambiente. A aprendizagem supervisionada é feita pela comparação entre a saída da rede e a saída desejada na qual o objetivo é aprender uma regra geral que mapeie as entradas e saídas.

Em (MAGLOGIANNIS et al., 2007) é feita a comparação de diversos algoritmos supervisionados e não supervisionados tais como o *K-Means*, SVM, árvores de decisão e *Perceptron*. O *K-Means* é um método de clusterização para particionar n observações dentre k grupos onde cada observação pertence ao grupo mais próximo da média. A técnica supervisionada de máquina de vetores de suporte (*Support Vector Machine* - SVM) é um classificador linear binário não probabilístico. Em (SCHWENKER; TRENTIN, 2014) são relatados métodos não supervisionados tais como a clusterização de dados para o reconhecimento de padrões em treinamento supervisionado.

A Figura 2 ilustra os principais algoritmos de aprendizado de máquina supervisionados e não supervisionados.

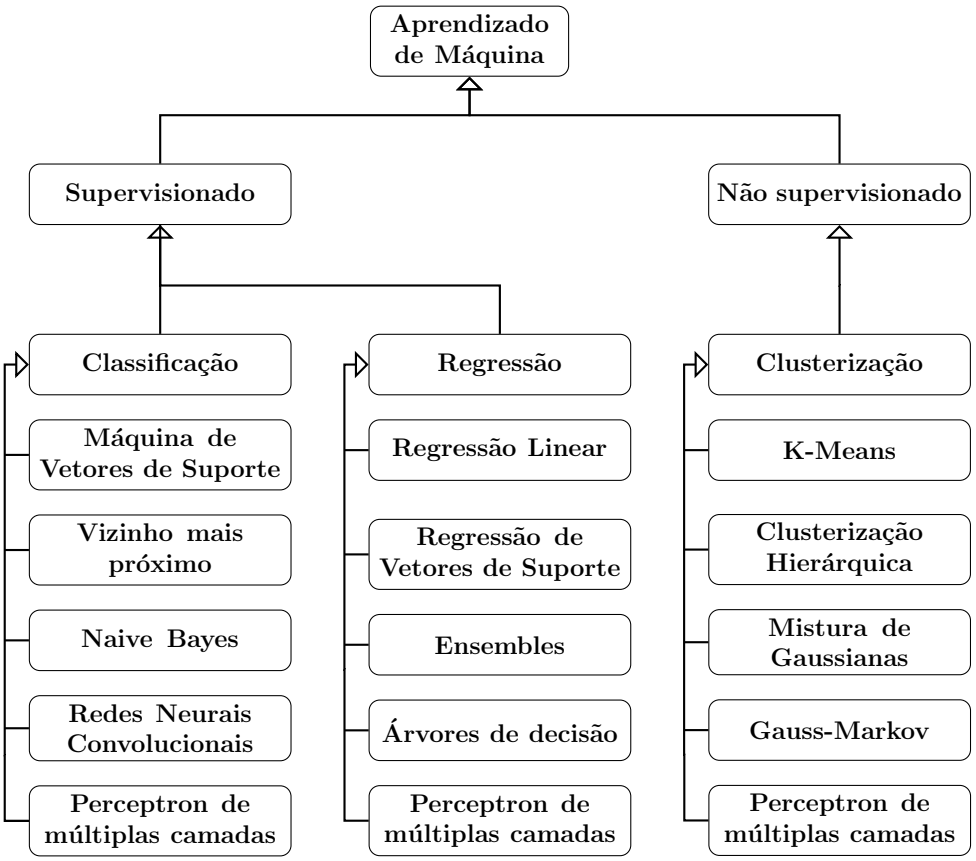


Figura 2: Métodos de aprendizado de máquina

3.1.1 Perceptron

O *Perceptron* é o tipo mais simples de rede neural artificial desenvolvida por (ROSENBLATT, 1958) no *Cornell Aeronautical Laboratory* e pode ser definido como um classificador linear binário com aprendizado supervisionado. Neste modelo, foi proposto um algoritmo inspirado pelo primeiro modelo de células cerebrais, cujas ondas nervosas são envolvidas no processo de transmissão de sinais químicos e elétricos.

As redes neurais possuem nós que são as unidades de processamento representadas por cada neurônio. Cada unidade possui ligações para outras unidades, onde recebem e enviam sinais. Essas unidades são os neurônios da rede neural. Os neurônios recebem e transmitem informações. A operação que um neurônio realiza pode ser definida conforme as etapas descritas de forma sequencial:

- Sinais são recebidos pela camada de entrada da rede neural;
- A camada de entrada pode possuir uma constante denominada *bias*, usada para aumentar os graus de liberdade e adaptação ao conhecimento a ela fornecido.
- Cada sinal é multiplicado por um peso;
- Os sinais são propagados de um elemento a outro por meio de conexões;
- É feita a soma ponderada dos sinais que produzem um nível de atividade. Trata-se de uma combinação linear das entradas pelos pesos associados;
- Uma função de ativação é geralmente utilizada com dois propósitos: limitar a saída do neurônio e introduzir não-linearidade no modelo;
- Se este nível de atividade exceder um certo limite a unidade produz uma determinada resposta na saída;

A Figura 3 ilustra o neurônio artificial com sinais de entrada e função de ativação. As funções de ativação introduzem um componente não linear nas redes neurais permitindo que elas possam aprender de forma mais eficaz. As funções mais utilizadas incluem a função linear, *sigmoide*, a tangente hiperbólica, tangente hiperbólica retificada, *softsign*, ReLU (unidade linear retificada) e *softmax*.

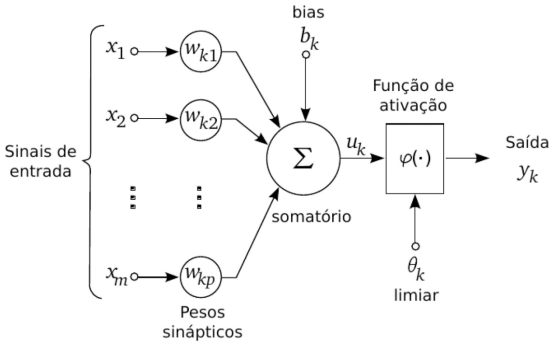


Figura 3: Ilustração de um neurônio artificial

As funções *sigmoide* e tangente hiperbólica proporcionam um aprendizado lento e custo computacional alto por conta de expoentes e multiplicações nas operações (NWANKPA et al., 2018). A vantagem da função *sigmoide* é que ela é uma função diferenciável continuamente porém pode proporcionar uma convergência lenta para o treinamento dependendo da complexidade da rede neural. A função sigmoide é apresentada na Equação 1:

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

A função tangente hiperbólica é uma modificação da função *sigmoide*. Enquanto os valores de saída da *sigmóide* variam entre 0 e 1 a tangente hiperbólica varia entre -1 e 1. A função tangente hiperbólica é apresentada na Equação 2:

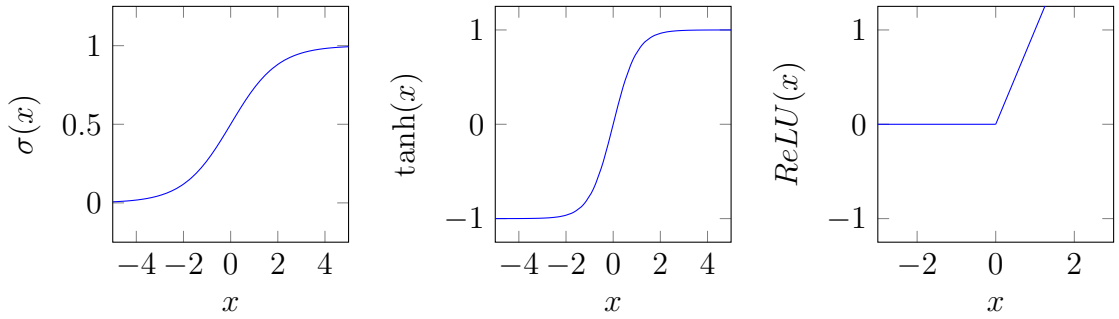
$$f(x) = tanh(x). \tag{2}$$

A função *softmax* normaliza a saída da rede para valores positivos cuja soma é igual a 1. Esta função permite que a saída de uma rede neural represente a probabilidade dos dados serem de uma das classes definidas. Sem ela as saídas dos neurônios são simplesmente valores numéricos onde o maior indica a classe vencedora.

A função de ativação ReLU proporciona um aprendizado rápido e um custo operacional mais baixo. A ReLU evita a rápida redução nos valores dos gradientes denominada dissipação dos gradientes e é uma das causas da progressiva lentidão no treinamento de redes neurais. A função de ativação ReLU é representada na Equação 3:

$$f(x) = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{senão.} \end{cases} \tag{3}$$

Por questões de performance, a prioridade na escolha de uma função de ativação deve ser primeiramente a ReLU. A Figura 4 ilustra as funções *sigmoide*, tangente hiperbólica e ReLU.



(a) Sigmoide (b) Tangente hiperbólica (c) ReLU

Figura 4: Funções de ativação: Sigmoide, Tangente hiperbólica e ReLU

3.1.2 Perceptron de multicamadas

O algoritmo de *Perceptron* de múltiplas camadas (*Multilayer Perceptron* - Multi Layer Perceptron (MLP)) consiste em uma rede neural de múltiplas camadas com conexões progressivas (*feedforward*) sem realimentação. Esta rede é composta por camadas de neurônios ligadas entre si por sinapses com pesos. A MLP geralmente é utilizada para solução de problemas complexos tais como o reconhecimento de padrões em imagens, vídeos, textos e na modelagem de sistemas não lineares (HAYKIN, 1994).

O aprendizado nesse tipo de rede é geralmente feito por meio do algoritmo de retro-propagação do erro (*backpropagation*) que corrige os pesos em todas as camadas (HECHT-NIELSEN, 1988), partindo da saída até a entrada a partir do erro calculado. O *backpropagation* é utilizado pelo algoritmo de otimização da descida do gradiente, apresentado na Seção 3.2.1 para ajustar os pesos dos neurônios calculando o gradiente da função erro.

O processo de retro-propagação do erro é constituído de duas etapas: uma etapa de propagação do sinal funcional (*feedforward*) e uma de retro-propagação do erro (*backpropagation*). Na etapa da propagação do sinal, os vetores de dados passam pelas unidades de entrada propagando pela rede a cada camada e um conjunto de saídas é produzido. Durante a etapa de propagação do sinal, os pesos das conexões são mantidos fixos. Na retro-propagação do erro os pesos são ajustados de acordo com uma regra de correção do erro (HECHT-NIELSEN, 1989). A resposta da rede em um instante de tempo é subtraída da saída desejada para produzir um sinal de erro. Este sinal de erro é propagado da saída para a entrada no processo de retro-propagação do erro. Os pesos são ajustados de forma que o erro entre a resposta da rede e a resposta desejada seja minimizado.

As RNAs possuem hiperparâmetros que são valores definidos inicialmente e permanecem constantes durante todo o processo de treinamento. Os hiperparâmetros são responsáveis por melhorar a precisão final da classificação e controlar o comportamento da rede neural. A taxa de aprendizado controla o quanto os pesos serão ajustados em cada etapa do processo de aprendizado. O número de épocas refere-se a quantas vezes o algoritmo itera pela etapa de treinamento. A função de ativação introduz um componente de não linearidade. O tamanho do lote refere-se a quantos exemplos de treinamento são utilizados. O número de camadas escondidas auxilia no tratamento de dados que contêm descontinuidades e o número de neurônios por camada define a arquitetura da rede.

Vale ressaltar que uma quantidade baixa de neurônios levam a baixa capacidade preditiva (*underfitting*) e muitos neurônios levam a alto processamento e perda da capacidade de generalização (*overfitting*). Os processos de *underfitting* e *overfitting* devem ser controlados com estratégias na definição da arquitetura e treinamento da rede a fim de que se tenha boa capacidade de generalização e classificação (LING, 1995).

Uma rede MLP é composta de uma camada de entrada, camadas intermediárias denominadas camadas escondidas e uma camada de saída. A Figura 5 ilustra uma rede MLP com uma camada de entrada, duas camadas escondidas e uma camada de saída.

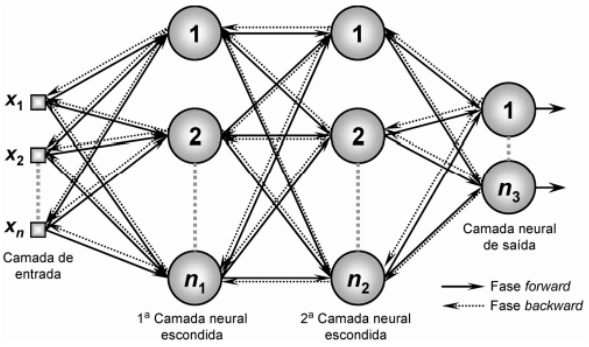


Figura 5: Ilustração de uma rede MLP com duas camadas intermediárias

3.1.3 Redes neurais convolucionais

Redes neurais convolucionais são modelos de aprendizagem profunda geralmente utilizados para aplicações de reconhecimento e classificação de padrões em imagens e vídeos. A CNN foi inspirada no mecanismo de percepção visual dos seres vivos. Em (HUBEL; WIESEL, 1968) foi descoberto que as células do córtex visual em animais são responsáveis pela detecção de luz em campos receptivos. Este tipo de rede é formado por uma camada

de entrada que recebe as imagens, uma série de camadas que realizam operações de tratamento de imagens com mapeamento de características e uma última etapa que contém uma rede neural de classificação. A última etapa recebe o mapeamento de características e fornece como saída o resultado da classificação.

As redes convolucionais são compostas por neurônios que possuem os parâmetros pesos e *bias*. Cada neurônio recebe entradas, aplica o produto escalar das entradas e pesos além de uma função de ativação não-linear. Neste tipo de rede neural é pressuposto que todas as entradas são imagens, o que permite definir propriedades na arquitetura. Redes neurais tradicionais não são escaláveis para imagens, uma vez que produzem um número muito alto de pesos a serem treinados. Uma CNN é composta por uma sequência de camadas. Além da camada de entrada, que recebe as imagens, existem três camadas principais: camada convolucional responsável por identificar regiões significantes com a extração de características, camada de *pooling* (agrupamento) e camada totalmente conectada. Em (PATHAK; RAUTARAY, 2018) as camadas de *pooling* são descritas como responsáveis por reduzir a dimensionalidade das matrizes e a complexidade na rede neural. Além disso, após uma camada de convolução é frequente uma camada de ativação que é responsável pelo aprendizado da rede de forma mais eficaz com a introdução das funções de ativação. A camada totalmente conectada (*fully connected layer*) é uma rede neural que recebe a saída da etapa de extração de características conforme a Figura 6 e realiza a classificação final. Essas camadas, quando colocadas em sequência, formam a arquitetura de uma CNN como ilustrado na Figura 6.

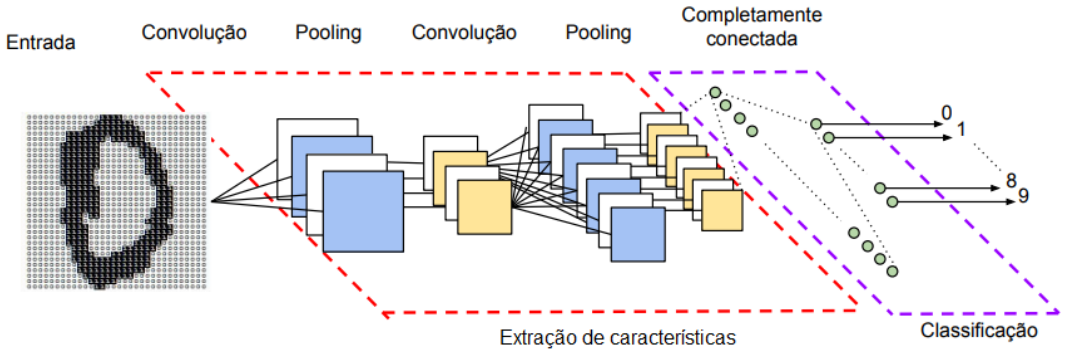


Figura 6: Arquitetura de uma rede neural convolucional

A camada de convolução é composta por um conjunto de filtros que são matrizes de valores reais que podem ser interpretados como pesos. Os filtros contribuem para o aprendizado de acordo com um treinamento. Além disto, os filtros auxiliam na convolução com os dados de entradas para obter um mapa de características. Estes mapas indicam regiões em que as características específicas em relação ao filtro, são encontradas na entrada. Os valores dos filtros se alteram ao longo do treinamento (assim como os pesos de uma rede neural tradicional) fazendo com que a rede aprenda a identificar regiões significantes de extração de características dos dados como ilustrado na Figura 7.

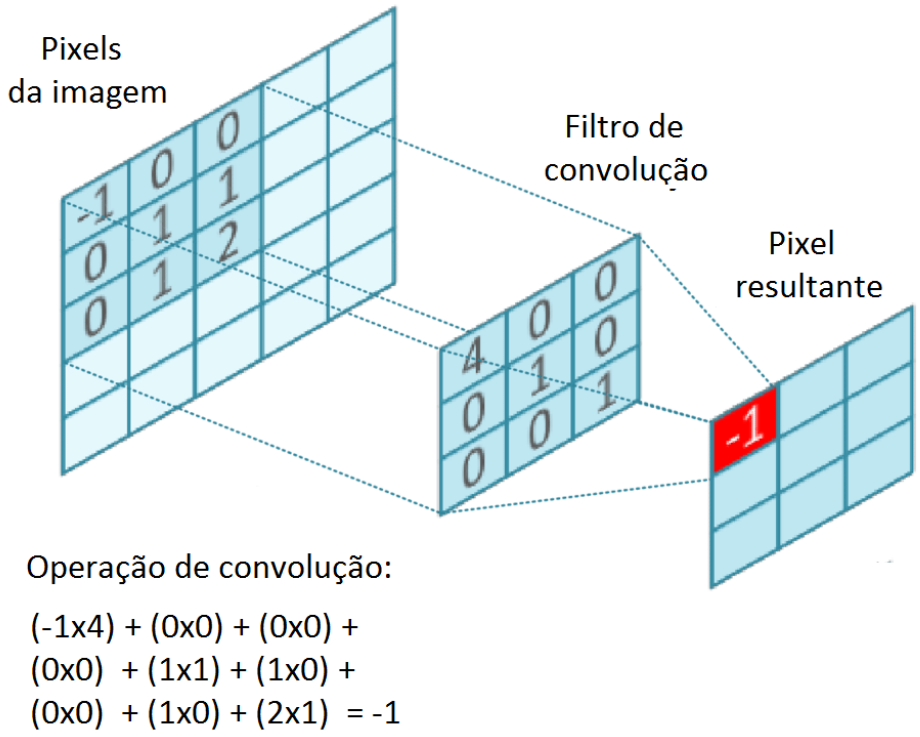


Figura 7: Ilustração do processo de convolução

Os principais tipos de camadas de *pooling* são: *max pooling, average pooling, deformation pooling, spatial pyramid pooling, scale dependent pooling.* A técnica de *pooling* é utilizada após a camada de convolução com o objetivo de reduzir o tamanho espacial das matrizes resultantes da convolução. A Figura 8 ilustra um exemplo do procedimento de *pooling* onde a matriz de entrada é de dimensão 4x4 e a matriz de saída é de dimensão 2x2 utilizando um filtro de dimensão 2x2.

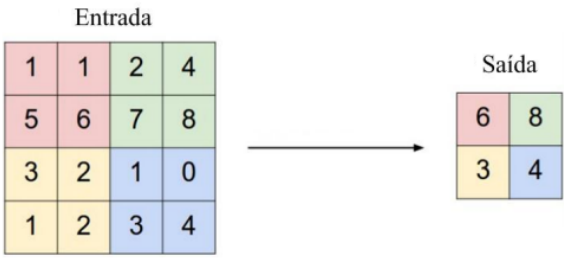


Figura 8: Ilustração do processo de *pooling* máximo com filtro de dimensão 2x2

O processo de *data augumentation* é uma técnica para aumentar diversidade de dados disponíveis para modelos de treinamento aplicando rotações e espelhamento das imagens (LENG; YU; QIN, 2017) enquanto a normalização em lotes (*batch normalization*) é uma técnica para padronizar as entradas para uma camada em uma rede neural.

Nas matrizes de resultados da convolução são aplicadas funções de ativação. A função mais utilizada é a ReLU devido aos melhores resultados obtidos conforme observado em (BIRCANOGLU; ARICA, 2018). Nessa referência a função de maximização é aplicada em cada elemento do resultado da convolução. A função ReLU não ativa todos os neurônios

ao mesmo tempo, com isto apenas alguns neurônios são ativados, tornando a rede esparsa e eficiente para a computação. As entradas negativas são convertidas em zero e o neurônio não é ativado.

3.2 Técnicas de aprendizado de máquina

O aprendizado de máquina utiliza métodos e modelos computacionais preditivos criados a partir de dados observacionais. Neste contexto, a máquina desenvolve a capacidade de aprender continuamente com os dados, conseguindo realizar previsões e reconhecimento de padrões assim como os humanos (LOURIDAS; EBERT, 2016). O aprendizado de máquina se aplica a problemas de reconhecimento de padrões, previsão de séries temporais, visão computacional, análises textuais e sistemas de recomendação.

A partir dos esforços de corporações como *Google, Microsoft, Facebook, Amazon* o aprendizado de máquina se tornou um dos tópicos mais importantes da ciência computacional na última década (EDGAR; MANZ, 2017).

Os sistemas especialistas são aqueles capazes de solucionar um problema de uma área específica assim como um especialista humano desta área. Sistemas especialistas são capazes de realizar inferência baseada em dados auxiliando decisões humanas.

Destacam-se no ramo de aprendizado de máquina os modelos de classificação de dados tais como redes neurais artificiais, regressão linear, os modelos de clusterização que buscam agrupar dados similares, e algoritmos de aprendizado por reforço. O aprendizado por reforço (*Reinforcement Learning*) é uma técnica cujo objetivo é treinar um programa denominado agente para atingir um objetivo por meio de recompensas. Redes neurais são um subconjunto de técnicas de aprendizagem de máquina que simulam os neurônios humanos para aprender determinada tarefa por meio de um processo de treinamento (UHRIG, 1995).

As redes neurais profundas são uma categoria de algoritmos de aprendizado de máquina que possuem vários níveis de operações com muitas camadas ocultas na arquitetura das redes neurais (BENGIO, 2013). As redes neurais profundas exigem uma grande quantidade de dados para treinamento. A principal diferença entre as redes neurais tradicionais e as profundas é a quantidade de camadas na rede. Os modelos de aprendizado profundo são amplamente aplicados para reconhecimento de imagem, superando outros algoritmos de aprendizado de máquina (RAWAT; WANG, 2017).

3.2.1 Descida do gradiente

A descida de gradiente é um algoritmo de otimização iterativa de primeira ordem para encontrar o mínimo de uma função, geralmente denotada como função custo, de aprendizagem de máquina. A minimização da função custo é feita para obter os valores de otimização para os parâmetros de aprendizado. O algoritmo busca minimizar uma função objetivo atualizando os parâmetros na direção oposta do gradiente desta função. A taxa de aprendizado determina o tamanho do passos utilizados para atingir um mínimo local. O algoritmo segue na direção da inclinação da superfície criada pela função objetivo em declive até encontrar um vale.

O algoritmo de descida de gradiente possui três variantes explicadas posteriormente que diferem na quantidade de dados que são utilizados para calcular o gradiente da função objetivo. Dependendo da quantidade de dados é modificada a precisão da atualização dos parâmetros do algoritmo e o tempo necessário para executar uma atualização.

Para contextualizar a rede neural artificial de forma algébrica, considera-se uma rede neural com C camadas intermediárias e N neurônios cada um com suas entradas expressas por $x_i^{(c)}$ e suas funções de ativação denotadas como $f^{(c)}$. Um nerônio $x_j^{(c)}$ recebe um sinal de outro denominado $x_i^{(c-1)}$ de uma camada anterior com um peso denotado como $w_{ij}^{(c)}$. Cada neurônio $x_j^{(c)}$ possui a constante *bias* definida como $b_j^{(c)}$. O sinal de saída de cada neurônio pode ser expresso conforme a Equação 4:

$$y_j^{(c)} = f^{(c)} \left(\sum_{i=1}^{N_{c-1}} x_i^{(c-1)} w_{ij}^{(c)} + b_j^{(c)} \right) \qquad j = 1, 2, \ldots, N_C, \tag{4}$$

onde $y_j^{(c)}$ é o sinal de saída de cada neurônio, $x_i^{(c)}$ é sinal de entrada de cada neurônio, $w_{ij}^{(c)}$ é peso de cada neurônio, $b_j^{(c)}$ é o *bias* de cada neurônio e $f^{(c)}$ é a função de ativação de cada neurônio.

Os pesos e os parâmetros *bias* então são atualizados utilizando o algoritmo da descida do gradiente de forma a minimizar o erro quadrático para o treinamento que pode ser definido conforme a Equação 5:

$$E = ||y(t) - o(t)||^2, \tag{5}$$

onde $y(t)$ é o vetor de saída da rede em um tempo t do treinamento e $o(t)$ é o vetor alvo objetivo de classificação supervisionada em um tempo t do treinamento.

O erro quadrático E é uma função de todos os pesos e dos parâmetros *bias* da rede neural já que $y(t)$, que é um vetor, e depende destes parâmetros. Logo, substituindo $y(t)$ da Equação 4 na Equação 5 é definida a Equação 6:

$$E = \left|\left| f^{(c)} \left(\sum_{i=1}^{N_{c-1}} x_i^{(c-1)} w_{ij}^{(c)} + b_j^{(c)} \right) - o(t) \right|\right|^2. \tag{6}$$

Visando minimizar a função erro, calcula-se as derivadas parciais de E com relação aos parâmetros $w_{ij}^{(c)}$ e $b_j^{(c)}$ usando a regra da cadeia conforme Equação 7. Resolvendo as derivadas da Equação 7 é possível definir as Equações 8 e 9 para a compreensão do processo da descida gradiente.

$$\frac{\partial E}{\partial w_{ij}^{(c)}} = \sum_{n=1}^{C} \frac{\partial E}{\partial y_n^{(c)}} \frac{\partial y_n^{(c)}}{\partial w_{ij}^{(c)}} \tag{7}$$

$$w_{ij}^{(c)}(t+1) = w_{ij}^{(c)}(t) - \eta \, \frac{\partial E}{\partial w_{ij}^{(c)}(t)} \tag{8}$$

$$b_j^{(c)}(t+1) = b_j^{(c)}(t) - \eta \, \frac{\partial E}{\partial b_j^{(c)}(t)}, \tag{9}$$

onde η é taxa de aprendizado. Reescrevendo as Equações 8 e 9 é possível definir as Equações 10 e 11.

$$w_{k+1} = w_k - \eta \, \nabla E(w_k) \tag{10}$$

$$b_{k+1} = b_k - \eta \, \nabla E(b_k), \tag{11}$$

onde w_k é o peso em uma iteração no aprendizado e b_k é o *bias* em uma iteração no aprendizado. Com isto é possível definir a descida do gradiente como iterações buscando a minimização da função erro $E(w)$, onde η_k é a taxa de aprendizado a cada iteração.

for $k = 0, 1, 2, \ldots$
$E_k \leftarrow \nabla E(w_k)$
$w_{k+1} \leftarrow w_k - \eta_k E_k$
end

Existem três tipos principais de gradiente descendente: descida do gradiente estocástico, descida do gradiente em lote e descida do gradiente em mini-lotes. Na descida do gradiente, há um lote que indica o número total de amostras de um conjunto de dados usado para calcular o gradiente para cada iteração.

Na descida em gradiente em lote, o lote é considerado o conjunto de dados inteiro antes de atualizar os pesos. No entanto, com o número de grandes amostras de treinamento, o uso da descida do gradiente em lote pode demorar muito tempo, ou seja, o algoritmo demora a convergir. O treinamento em mini-lote realiza o treino com um número reduzido de conjunto de treinamento atualizando os parâmetros do modelo em cada iteração.

A descida do gradiente estocástico refere-se a um processo que está vinculado a uma probabilidade aleatória. Algumas amostras são selecionadas aleatoriamente em vez de todo o conjunto de dados para cada iteração. O uso da descida do gradiente estocástico é mais rápido porque a melhoria de desempenho ocorre na primeira amostra.

3.2.2 Sobreajuste e subajuste

O processo de subajuste em uma rede neural é quando o modelo de aprendizado de máquina não consegue reduzir o erro no conjunto de teste ou treinamento e não se adapta bem com dados com os quais foi treinado. Por outro lado, o processo de sobreajuste ocorre quando o modelo se ajustou bem aos dados com os quais está sendo treinado mas não generaliza para novos dados. Desta forma o modelo decora o conjunto de dados de treino, mas não classifica bem novas amostras.

Para controlar o processo de *overfitting* uma das estratégias consiste em adicionar camadas de *pooling* e *dropout*. Conforme observado em (KO; KIM; CHOI, 2017) a camada de *dropout* permite minimizar o processo de *overfitting* aumentando a eficiência da rede neural, aumentando o tempo de processamento no treinamento. As camadas de *dropout* funcionam zerando aleatoriamente uma percentagem de neurônios nas camadas da rede neural e as camadas de *pooling* reduzem a dimensionalidade das matrizes e a complexidade da rede. A camada de *dropout* zera uma certa quantidade de ativações da camada baseando-se em uma tomada de decisão probabilística, agindo mediante ativação por um limiar pré-definido. A Figura 9 ilustra o processo de *dropout*.

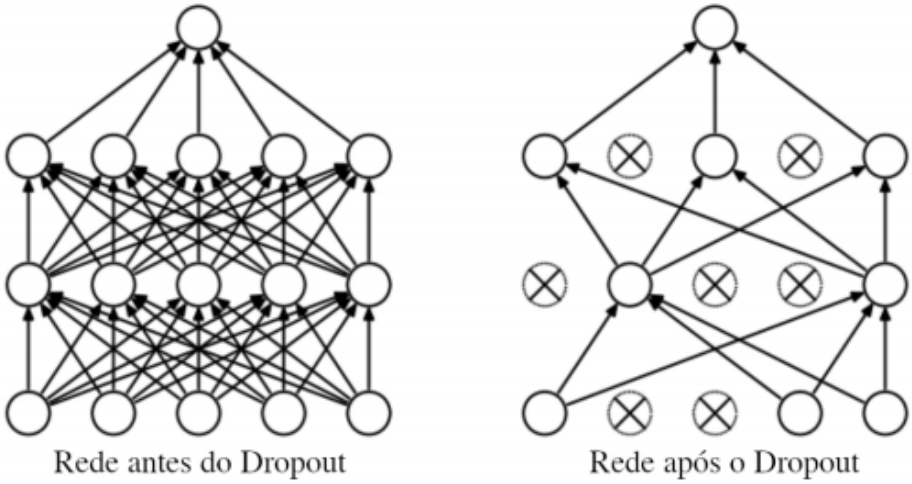


Figura 9: Ilustração do processo de *Dropout*
[fonte: (KO; KIM; CHOI, 2017)]

3.2.3 Otimização do gradiente descendente

Existem três formas de otimizar o algoritmo do gradiente descendente: alterar a taxa de aprendizado η, alterar a componente $\frac{\partial E}{\partial w}$ de gradiente e alterar as componentes do gradiente e da taxa de aprendizado.

Os otimizadores que alteram a componente de gradiente ajustam a taxa de aprendizado realizando a multiplicação por um fator que é uma função dos gradientes. Enquanto os otimizadores da taxa de aprendizado multiplicam a taxa de aprendizado por um fator que é uma constante ou uma função do intervalo de tempo.

A Figura 10 apresenta o relacionamento e origem dos algoritmos de otimização onde GD são algoritmos de otimização do gradiente descendente e TX são algoritmos de otimização da taxa de aprendizado. O SGD (*Stochastic gradient descent* - SGD) se ramificou em dois tipos principais de otimizadores: os que atuam sobre o componente gradiente, a partir do algoritmo do *Momentum* e os que atuam sobre a componente da taxa de aprendizado, por meio do algoritmo de gradiente adaptativo (*Adaptive gradient* - *AdaGrad*). Na sequência houve o surgimento do algoritmo de propagação da raiz do valor quadrático médio (*Root mean square propagation* - *RMSprop*) e o método de taxa de aprendizado adaptativa (*Adaptive Learning Rate Method* - AdaDelta) originados a partir do Adagrad. Nota-se que o algoritmo Nesterov é originado do *Momentum*. Em seguida surge o algoritmo de estimativa de momento adaptativo (*Adaptive Moment Estimation* - Adam) que é uma combinação do *Momentum* com o *RMSprop*. É possível observar que os algoritmos Adam, algoritmo variante de média móvel exponencial do gradiente (*Exponential moving avarage variant* - AMSGrad), estimativa de momento adaptativa acelerada por Nesterov (*Nesterov accelerated Adam* - Nadam) e o algoritmo de gradiente adapatativo baseado na norma infinita (*adaptive stochastic gradient descent* - Adamax) utilizam ambas as técnicas. Os algoritmos de otimização possuem as seguintes características principais:

- *SGD*: amostras são selecionadas aleatoriamente durante o treinamento levando a convergência mais rápida de forma ruidosa devido ao caráter estocástico.
- *AdaGrad*: reduz a taxa de aprendizado dos pesos que recebem gradientes altos aumentando a taxa de acurácia.

- *RMSProp*: modifica a taxa de aprendizado antes de aplicar os gradientes convergindo mais rápido.
- *Adam*: modifica o *AdaGrad* suavizando os ruídos e faz com que a taxa de aprendizado não diminua agressivamente.
- *AdaMax*: modifica o *Adam* para aumentar a velocidade de convergência.

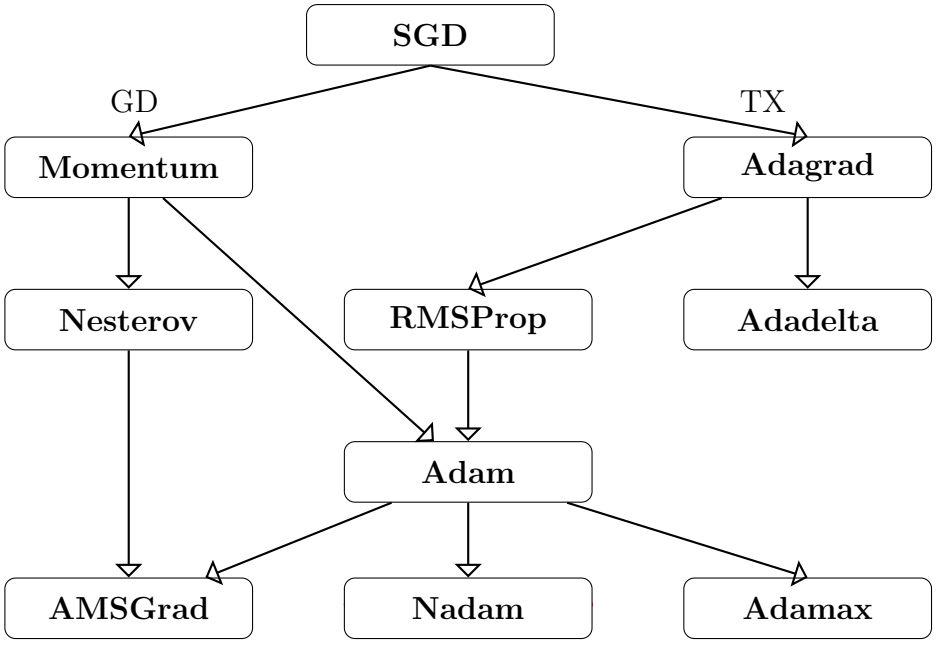


Figura 10: Algoritmos de otimização do gradiente descendente

O gradiente descendente estocástico é um algoritmo que atualiza os parâmetros de treinamento utilizando apenas uma única amostra de treinamento em cada iteração selecionada aleatoriamente e converge rapidamente de forma ruidosa devido ao caráter probabilístico (ROBBINS; MONRO, 1951). O gradiente descendente é um processo que utiliza a amostra completa e se o tamanho da amostra for muito grande o processo de aprendizado pode se tornar lento.

A descida do gradiente de estocástico para um instante t denominado atual realiza a atualização do peso atual w_t utilizando o gradiente atual $\nabla E(w_t)$ multiplicado pela taxa de aprendizado η definida conforme Equação 12:

$$w_{t+1} = w_t - \eta \, \nabla E(w_t), \tag{12}$$

onde w_{t+1} é o peso no instante t+1, w_t é o peso no instante t atual, η é a taxa de aprendizado e $\nabla E(w_t)$ é o gradiente da função erro em um instante t atual.

3.2.3.1 Momentum

Para acelerar o gradiente descendente um fator denominado *momentum* representado por μ é adicionado nas equações de treinamento como uma variação do algoritmo do gradiente descendente estocástico (POLYAK, 1964). A descida do gradiente por momento substitui o gradiente atual por uma média móvel exponencial dos gradientes atuais por meio de um

parâmetro de velocidade denotado por V_t. Atua suavizando atualizações, aumentando a estabilidade e a velocidade do algoritmo. O fator de atualização de velocidade é definido conforme Equação 13:

$$V_{t+1} = \mu V_t - \eta \nabla E(w_t), \tag{13}$$

onde V_{t+1} é o vetor velocidade de atualização dos pesos em instante t+1 da iteração, μ é o fator de momento, V_t é o fator de velocidade de atualização em um instante t, η é a taxa de aprendizado, $\nabla E(w_t)$ é o gradiente atual. A atualização do peso para este algoritmo é definida na Equação 14:

$$w_{t+1} = w_t + V_{t+1}, \tag{14}$$

onde w_{t+1} é o peso no instante t+1, w_t é o peso no instante t atual e V_{t+1} é o vetor velocidade de atualização em instante t+1 da iteração.

3.2.3.2 Nesterov

O gradiente acelerado de Nesterov (NAG) foi proposto em (NESTEROV, 1983) para problemas de otimização convexa. A diferença deste algoritmo para o algoritmo do *momentum* é a atualização parcial com o termo $(+\mu V_t)$ no gradiente aumentando a estabilidade e a velocidade do algoritmo. O fator de velocidade de atualização dos pesos da Equação 15 é utilizado na Equação 14 para atualização dos pesos neste algoritmo. O fator de velocidade de atualização dos pesos é definido na Equação 15:

$$V_{t+1} = \mu V_t - \eta \nabla E(w_t + \mu V_t), \tag{15}$$

onde V_{t+1} é o vetor velocidade de atualização em instante t+1 da iteração, μ é o fator de momento, V_t é o fator de velocidade de atualização em um instante t, η é a taxa de aprendizado, $\nabla E(w_t + \mu V_t)$ é o gradiente da função erro em um instante atual com a atualização de Nesterov.

3.2.4 Taxa de aprendizado

A taxa de aprendizado é um parâmetro de treinamento que influencia em como a rede irá aprender. Taxas de aprendizado altas permitem que o algoritmo de treinamento tenha uma convergência mais rápida. Taxas de aprendizados baixas podem levar ao excesso de ajustes de pesos e lentidão no aprendizado no treinamento. Se a taxa de aprendizado for muito alta existe ainda a possibilidade do algoritmo de treinamento não convergir.

3.2.5 Otimização da taxa de aprendizado

Nos algoritmos de taxa de aprendizado adaptativa o valor da taxa de aprendizado, que corresponde à velocidade do aprendizado, é modificado por um fator definido em cada algoritmo.

A taxa de aprendizado adaptativa permite comparar o desempenho de vários algoritmos em termos dos aspectos de convergência, rapidez e precisão. É possível analisar também a robustez e desempenho para diferentes dados em relação a complexidade computacional.

3.2.5.1 AdaGrad

O algoritmo de gradiente adaptativo Adagrad foi proposto em (DUCHI; HAZAN; SINGER, 2011) e tem o efeito de reduzir a taxa de aprendizado dos pesos que recebem gradientes altos aumentando a acurácia. É um método onde a taxa de aprendizado é normalizada pela raiz quadrada da soma dos valores quadrados passados. Ou seja, a soma acumula os gradientes quadrados atuais e passados. A atualização do peso para este algoritmo é definida na Equação 16:

$$w_t = w_{t-1} - \eta \frac{\nabla E_t(w_{t-1})}{\sqrt{\sum_{t'=0}^{t} \nabla E_{t'}(w_{t'-1})^2}}, \tag{16}$$

onde w_t é o peso no instante t, w_{t-1} é o peso no instante t-1, t' é o instante das iterações históricas variando no intervalo $\in [0, 1, \ldots, t]$, η é a taxa de aprendizado, $\nabla E_t(w_{t-1})$ é o gradiente da função erro no instante t-1 e $\nabla E_{t'}(w_{t'-1})^2$ é o gradiente da função erro histórico quadrático.

Possui a desvantagem de que todos os gradientes atuais e anteriores têm o mesmo peso. A acumulação de gradientes ao quadrado desde o início do treinamento pode resultar em uma diminuição da taxa de aprendizado.

3.2.5.2 AdaDelta

O método da taxa de aprendizado adaptativa (*Adaptive Learning Rate Method* - AdaDelta) originado do AdaGrad foi proposto por (ZEILER, 2012). Este algoritmo busca reduzir a taxa de aprendizado e restringe a janela dos gradientes acumulados anteriores a um tamanho fixo. A Equação 17 apresenta o gradiente do erro em um instante t:

$$\delta w_t = w_t - w_{t-1}, \tag{17}$$

onde δw_t é o gradiente da função erro em um instante t. A Equação 18 define o termo D_t como uma acumulação dos gradientes:

$$D_t = \beta D_{t-1} + (1 - \beta)\delta w_t^2, \tag{18}$$

onde β é a taxa de decaimento , t é o tempo em que ocorre a atualização, D_{t-1} é a acumulação do gradiente no instante t-1 e δw_t^2 é o gradiente quadrático no instante t. O termo S_t está relacionado ao gradiente quadrático do peso e define a Equação 19:

$$S_t = \beta S_{t-1} + (1 - \beta).\left(\frac{\partial E}{\partial w_t}\right)^2, \tag{19}$$

onde β é a taxa de decaimento. A atualização do peso para este algoritmo pode então ser descrita na Equação 20:

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} + \epsilon}}{\sqrt{S_t + \epsilon}}.\frac{\partial E}{\partial w}, \tag{20}$$

onde ϵ é o termo constante de decaimento para manter a estabilidade numérica.

3.2.5.3 RMSprop

No algoritmo de propagação da raiz do valor quadrático médio denominado *RMSprop* é realizado o cálculo das médias da magnitude dos gradientes para cada parâmetro. Estas médias são utilizadas para modificar a taxa de aprendizado individualmente antes de aplicar os gradientes. Este algoritmo modifica o *AdaGrad* com uma média móvel exponencial controlada pela taxa $\gamma \in [0, 1)$ que possui uma ênfase em gradientes recentes para atingir um melhor desempenho. O valor quadrático médio dos gradientes R_t é definido na Equação 21:

$$R_t = \gamma R_{t-1} + (1 - \gamma)\nabla E_t(w_{t-1})^2, \tag{21}$$

onde γ é a taxa que controla a escala da média móvel. A atualização do peso para este algoritmo pode então ser descrita na Equação 22:

$$w_t = w_{t-1} - \eta \frac{\nabla E_t(w_{t-1})}{\sqrt{R_t}}. \tag{22}$$

3.2.6 Combinação de otimizadores

A combinação de algoritmos que variam a taxa de aprendizado e momento permite obter novos algoritmos visando utilizar as vantagens de cada um. A combinação permite melhorar a convergência e instabilidade dos algoritmos. Neste contexto os algoritmos que combinam otimizadores são o *Adam* que é uma combinação do *Momentum* e do *RMSprop*, *Adamax* e *Nadam* que são variantes do *Adam* e *AMSGrad* que é uma combinação do *Adam* e do *Nesterov*.

3.2.6.1 Adam

O algoritmo de treinamento de estimativa de momento adaptativo (*Adaptive moment estimation* - Adam) proposto por (KINGMA; BA, 2014) é um método derivado do *RMSprop* que ajusta o método *Adagrad* para que a taxa de aprendizado não diminua agressivamente. A contribuição do método Adam é adicionar um momento na atualização e suavizar os ruídos do gradiente antes de fazer essa operação. Ele herda do *RMSprop* a adição de uma taxa de decaimento na soma dos gradientes de cada parâmetro enquanto a taxa de aprendizado é reduzida a cada passo. Este método combina as vantagens do *AdaGrad* e o *RMSprop* e calcula as taxas de aprendizado a partir da estimativa dos momentos de primeira e segunda ordem dos gradientes. Para estimar os momentos, Adam utiliza médias móveis exponencialmente calculadas no gradiente avaliado em um mini lote no instante t atual. A atualização do peso para este algoritmo pode então ser descrita na Equação 23:

$$w_t = w_{t-1} - \eta \frac{\hat{M}_t}{\sqrt{\hat{R}_t + \epsilon}}, \tag{23}$$

onde η é a taxa de aprendizado, ϵ é o fator de decaimento. Vale ressaltar que η e ϵ devem ser maiores que zero. A atualização da primeira estimativa do momento é definida na Equação 24:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)\nabla E_t(w_{t-1}), \tag{24}$$

onde $\beta_1 \in [0, 1)$ é taxa de decaimento do primeiro momento. A atualização da segunda estimativa do momento é definida na Equação 25:

$$R_t = \beta_2 R_{t-1} + (1 - \beta_2)\nabla E_t(w_{t-1})^2, \tag{25}$$

onde $\beta_2 \in [0, 1)$ é a taxa de decaimento do segundo momento. Calcula-se a estimativa do primeiro momento corrigida pelo *bias* na Equação 26:

$$\hat{M}_t = \frac{M_t}{(1 - \beta_1^t)}, \tag{26}$$

onde o segundo momento corrigido pelo *bias* é definido na Equação 26:

$$\hat{R}_t = \frac{R_t}{(1 - \beta_2^t)}. \tag{27}$$

3.2.6.2 Adamax

O algoritmo de gradiente adaptável baseado na norma infinita (*Adaptive stochastic gradient descent* - Adamax) é uma variante do *Adam* onde o momento de segunda ordem é substituído pelo momento de ordem infinita (KINGMA; BA, 2014). Este algoritmo geralmente tem performance superior ao Adam devido à sua simplificação e estabilidade com a inserção do momento de ordem infinita. Neste algoritmo as atualizações do algoritmo Adam são generalizadas substituindo o termo do gradiente quadrático por um termo p. A atualização da primeira estimativa do momento é definida na Equação 28:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)\nabla E_t(w_{t-1}), \tag{28}$$

onde $\beta_1 \in [0, 1)$ é a taxa de decaimento do primeiro momento. A atualização da segunda estimativa do momento é definida na Equação 25:

$$R_t = \beta_2 R_{t-1} + (1 - \beta_2)\nabla E_t(w_{t-1})^p, \tag{29}$$

onde $\beta_2 \in [0, 1)$ é a taxa de decaimento do segundo momento e p é a constante de atualização. Calcula-se a estimativa do primeiro momento corrigida pelo *bias* na Equação 30:

$$\hat{M}_t = \frac{M_t}{(1 - \beta_1^t)}. \tag{30}$$

Calcula-se a estimativa do segundo momento corrigida pelo *bias* na Equação 31:

$$\hat{R}_t = \frac{R_t}{(1 - \beta_2^p t)}. \tag{31}$$

A atualização do peso parametrizado pela constante de atualização para este algoritmo pode então ser descrita na Equação 32:

$$w_t = w_{t-1} - \eta \frac{\hat{M}_t}{(\hat{R}_t + \epsilon)^p}, \tag{32}$$

onde p é a constante de atualização. Quando $p \to \infty$ os termos são modificados, conforme observado em (KINGMA; BA, 2014), gerando novas equações com a estimativa do momento

de ordem infinita. A atualização da primeira estimativa do momento de ordem infinita é definida na Equação 33:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)\nabla E_t(w_{t-1}). \tag{33}$$

Quando $p \to \infty$ na Equação 31 define-se um novo termo de maximização do gradiente na Equação 34:

$$U_t = \max\left\{\beta_2 U_{t-1}, |\nabla E_t(w_{t-1})|\right\}. \tag{34}$$

A atualização do peso para este algoritmo pode então ser descrita na Equação 35:

$$w_t = w_{t-1} - \eta \frac{\hat{M}_t}{U_t}, \tag{35}$$

onde η é a taxa de aprendizado.

3.2.6.3 Nadam

Nadam é uma fusão de conceitos dos otimizadores Nesterov e Adam conforme proposto em (DOZAT, 2016). Porém o componente Nesterov é modificado para ser mais eficiente do que sua implementação original. A partir das Equações 23, 24, 25, 26, 27 a otimização do algoritmo Adam pode ser escrita na Equação 36:

$$w_t = w_{t-1} - \eta \left(\frac{\beta_1 M_{t-1} + (1 - \beta_1)\nabla E_t(w_{t-1})}{\sqrt{\hat{R}_t + \epsilon}} \right), \tag{36}$$

onde $\beta_1 \in [0, 1)$ é taxa de decaimento do primeiro momento, M_{t-1} é a primeira estimativa do momento em um instante t-1, $\beta_2 \in [0, 1)$ é a taxa de decaimento do segundo momento e $\hat{R}_t$ é a atualização da segunda estimativa do momento e ϵ é o fator de decaimento.

Nadam utiliza o algoritmo de Nesterov para atualizar o gradiente em relação ao instante seguinte, substituindo o termo M_{t-1} na Equação 36 pelo atual M_t conforme a Equação 37:

$$w_t = w_{t-1} - \eta \left(\frac{\beta_1 M_t + (1 - \beta_1)\nabla E_t(w_{t-1})}{\sqrt{\hat{R}_t + \epsilon}} \right). \tag{37}$$

3.2.6.4 AMSGrad

O algoritmo AMSGrad é uma variação do Adam e foi proposto em (REDDI; KALE; KUMAR, 2018). Essa variante modifica o componente da taxa de aprendizado adaptável no Adam e melhora sua convergência garantindo que o termo R atual seja sempre maior que o passo anterior. A atualização da primeira estimativa do momento para o algoritmo AMSGrad é definida na Equação 38:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)\nabla E_t(w_t). \tag{38}$$

A atualização da segunda estimativa do momento para o algoritmo AMSGrad é definida na Equação 39:

$$R_t = \beta_2 R_{t-1} + (1 - \beta_2)\nabla E_t(w_t)^2. \tag{39}$$

Calcula-se a estimativa do primeiro momento corrigida pelo *bias* para o algoritmo AMSGrad na Equação 40:

$$\hat{R}_t = max(\hat{R_{t-1}}, R_t). \tag{40}$$

A atualização dos pesos para este algoritmo pode então ser descrita na Equação 41:

$$w_{t+1} = w_t - \eta \frac{M_t}{\sqrt{\hat{R}_t + \epsilon}}, \tag{41}$$

onde η é a taxa de aprendizado, ϵ é um fator de decaimento β_1 é a taxa de decaimento do primeiro momento, β_2 é a taxa de decaimento do segundo momento.

3.3 Considerações finais

Neste capítulo foram descritos os conceitos para fundamentação teórica deste trabalho no contexto de redes neurais artificiais e técnicas de otimização da descida do gradiente. Foram abordadas informações sobre os algoritmos de otimização *SGD*, *Momentum*, *AdaGrad*, *RMSprop*, *Adadelta*, *Nesterov*, *Adam*, *AdaMax*, *Nadam* e *AMSGrad*. Também foram apresentados conceitos para evitar o sobreajuste da rede neural, conceitos de funções de ativação e taxa de aprendizado. Para melhorar o desempenho de um sistema de aprendizagem profunda é possível otimizar estrutura e arquitetura do modelo, a inicialização dos parâmetros do modelo ou adotar um algoritmo de aprendizagem mais eficiente. Este trabalho utiliza as informações dos algoritmos de otimização descritos neste capítulo para o desenvolvimento da solução de classificação de imagens de redes sociais.

Capítulo 4

MODELO PROPOSTO E SUA IMPLEMENTAÇÃO

NESTE capítulo, são apresentados os procedimentos para o desenvolvimento e implementação dos modelos e a arquitetura da rede neural convolucional proposta para a classificação de imagens em categorias populares em redes sociais. Em seguida são apresentadas as técnicas utilizadas para criar, treinar e testar o modelo classificador de imagens utilizando diferentes algoritmos de otimização da descida do gradiente. São abordados também os conceitos das métricas de desempenho utilizadas para medir a performance e comparar os diferentes algoritmos utilizados.

4.1 Modelo proposto

Este trabalho avalia diferentes algoritmos de otimização de redes neurais como uma abordagem para a utilização do algoritmo da descida do gradiente no contexto de classificação de imagens. A comparação dos algoritmos e a performance de cada um permite a escolha do melhor algoritmo para classificação das imagens de redes sociais.

A primeira etapa do desenvolvimento do método de classificação consiste na busca de conjuntos de imagens nas categorias pré-definidas como animais, eletrônicos, jogos, veículos e vestuário. Vale ressaltar que a definição destas categorias foi baseada em (HU; MANIKONDA; KAMBHAMPATI, 2014) onde é feita a análise e categorização das imagens mais postadas por usuários do *Instagram*. De posse das imagens foram realizadas rotinas para leitura e tratamento das imagens conforme Seção 4.3. Após as imagens serem processadas a rede neural convolucional é implementada. Em seguida os algoritmos SGD, AdaGrad, Adamax, RMSProp e Adam de otimização de redes neurais são aplicados durante o treinamento da CNN. Para análise dos resultados é feita a comparação do desempenho de cada algoritmo testado na aplicação. São realizados treinos com todos os algoritmos de otimização com o objetivo de encontrar o mais adequado para a identificação do perfil de usuários. A Figura 11 ilustra as etapas percorridas durante toda a implementação da solução proposta.

4.1.1 Desenvolvimento do método de treinamento

Para o desenvolvimento do método de treinamento foi criado um ambiente virtual com o *container docker* (MAKSIMOV, 2019). Um *container docker* é um ambiente encapsulado que executa aplicativos por meio da virtualização em nível do sistema operacional. Uma imagem do *docker* é um arquivo, composto de várias camadas, usado para executar código

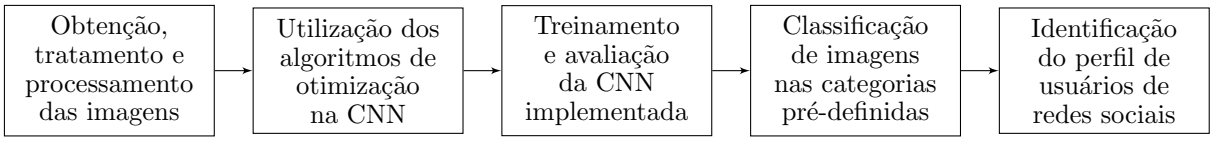


Figura 11: Fluxo proposto para identificação do perfil de usuários com treinamento utilizando algoritmos de otimização[1]

em um *container* do *docker*. A virtualização baseada em *container* usa um único núcleo para executar várias instâncias e a camada de virtualização é executada como um aplicativo no sistema operacional. Nessa abordagem, o *docker* pode ser subdivido em *docker client* onde são executados comandos docker, o *docker host* onde são criados os *container* baseados nas imagens e o *docker registry* onde são obtidas as imagens com as bibliotecas necessárias para execução de uma aplicação conforme ilustrado na Figura 12.

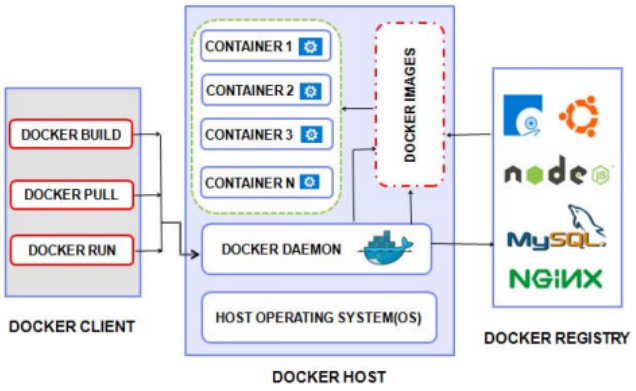


Figura 12: Arquitetura do *Docker*
[fonte: (Singh; Singh, 2016)]

Foi utilizada a imagem (MAKSIMOV, 2019) de um container *docker* com as seguintes ferramentas: *Jupyter*, *Matplotlib*, *Pandas*, *Tensorflow*, *Keras* e *OpenCV*. Neste trabalho foi utilizado o *Tensorflow* e o *Keras* com a linguagem *Python* para realização do treinamento. O *Tensorflow* é uma biblioteca de *software* de código aberto para a inteligência de máquina criado pelo Google em 2015 (MARTÍN et al., 2015). Com o *Tensorflow* é possível definir modelos de aprendizado de máquina, treiná-los com dados e exportá-los. O *Tensorflow* opera com tensores. Tensores são vetores multidimensionais que percorrem pelos nós de um grafo. Este grafo, definido pelo *Tensorflow*, é composto pelos seguintes elementos: um conjunto de objetos *tf.Operation*, que representam as unidades com as operações, um conjunto de *tf.Tensor*, que representam as unidades com os dados e uma sessão *tf.Session* que encapsula o ambiente onde as operações do grafo são executadas e os tensores são avaliados. A Figura 13 ilustra o grafo da biblioteca *Tensorflow* que implementa a CNN. Vale ressaltar que os algoritmos de otimização implementados neste trabalho são representados pelo bloco A da Figura 13.

[1]Os algoritmos de otimização utilizados estão disponíveis nos otimizadores da biblioteca *Keras*.

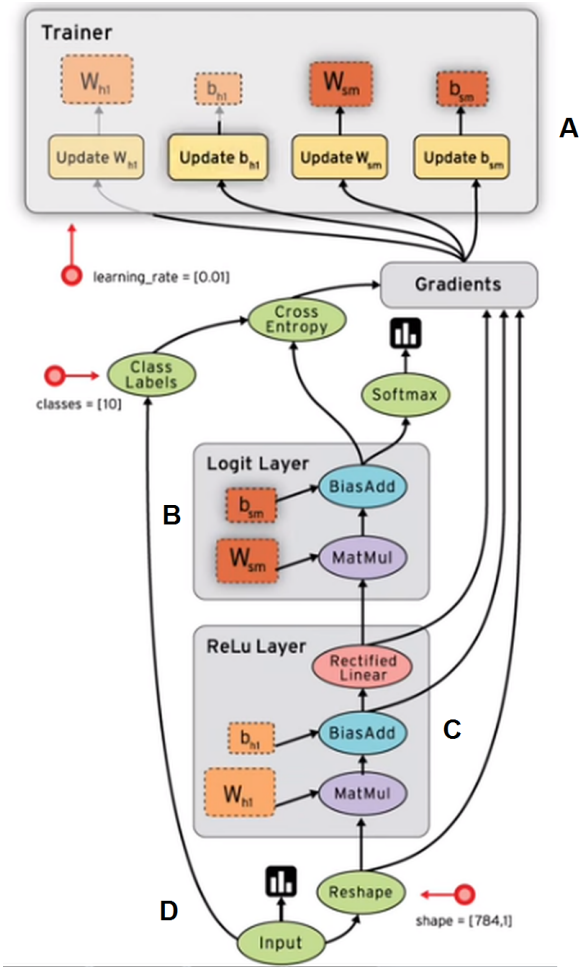


Figura 13: Arquitetura da biblioteca *Tensorflow*

De posse das imagens tratadas é realizada a execução do modelo de treinamento. A função objetivo para o treinamento pode ser definida como a função erro a ser minimizada no processo. O erro calculado foi o erro médio quadrático (*mean squared error*). O cálculo do erro é apresentado na Equação 42:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\Big(y(t) - o(t)\Big)^2, \tag{42}$$

onde y é o vetor de saída da rede, n é o número de exemplos de treinamento e o é o vetor alvo de classificação supervisionada no treinamento.

Foram criadas funções para a implementação dos otimizadores da descida do gradiente para cada execução e treinamento da rede neural. A Figura 14 ilustra o procedimento de criação do modelo.

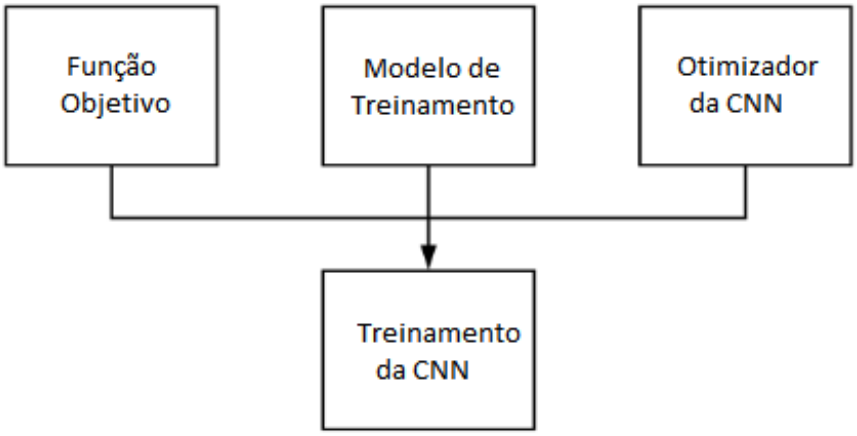


Figura 14: Execução do treinamento da rede neural

De posse do modelo foi possível treinar o modelo com os dados de treino. A Figura 15 ilustra o procedimento de ajuste dos pesos da rede neural convolucional durante o treinamento.

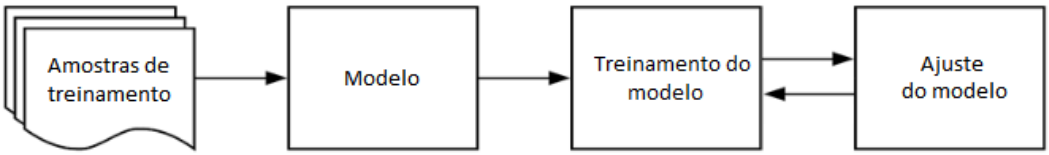


Figura 15: Ajuste dos pesos no treinamento da rede neural

A rede processa as imagens na primeira etapa de obtenção das características. O resultado da primeira etapa são mapas de característica que definem o que existe na imagem. Na segunda etapa os mapas de características definem onde estão as características na imagem. Em resumo, a primeira fase se preocupa em encontrar o que existe na imagem e a segunda em definir onde estão essas características na imagem.

4.1.2 Análise dos dados

A rede neural foi treinada para cada um dos algoritmos de otimização do gradiente descendente com o objetivo de avaliar e comparar a performance de cada um. Após a seleção do melhor algoritmo foram realizados testes para avaliação da generalização do modelo treinado com o melhor algoritmo. A Figura 16 ilustra o procedimento de predição e análise dos dados após o treinamento da CNN.

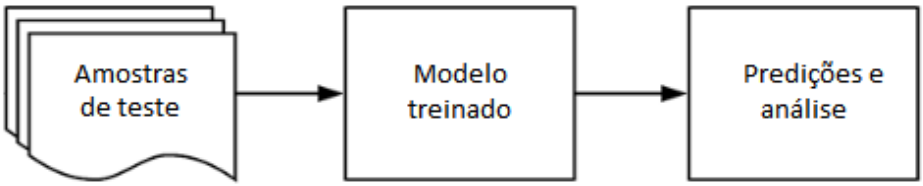


Figura 16: Procedimento de predição e análise das imagens de teste

Após o treinamento com os otimizadores foram realizadas comparações entre as redes resultantes utilizando métricas de desempenho. Dessa forma, foi possível realizar a seleção do melhor algoritmo para classificar imagens de redes sociais.

4.2 Arquitetura da rede neural convolucional

A arquitetura da CNN utilizada foi introduzida em (SIMONYAN; ZISSERMAN, 2015) e é uma variante menor da rede VGGNet (*Very Deep Convolutional Networks for Large Scale Image Recognition*). Esta arquitetura aprimora o modelo *AlexNet* (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) substituindo filtros de dimensão superior por vários filtros (dimensão 3×3). A arquitetura aprimorada foi desenvolvida pelo VGG (*Visual Geometry Group*) da Universidade de *Oxford*. A arquitetura da rede foi definida com sete conjuntos de camadas totalizando 28 camadas. Os cinco primeiro conjuntos possuem uma camada convolucional com filtros de convolução (*kernels*) de 3x3 e funções de ativação ReLU. Camadas com redução da dimensionalidade das matrizes utilizando a função *max pooling* são utilizadas.

O primeiro conjunto utiliza uma camada *max pooling* com dimensão 3x3, o terceiro e quinto conjunto utilizam uma camada *max pooling* de 2x2. Os seis primeiros conjuntos possuem uma camada de *batch normalization* que é uma técnica para padronizar as entradas para uma camada em uma rede neural. O sexto conjunto possui uma camada totalmente conectada com função de ativação ReLU. Por fim uma camada totalmente conectada com a função *softmax* possui cinco saídas conforme as classes definidas como animais, eletrônicos, jogos, veículos e vestuário. A divisão e quantitativo das amostras dos dados serão descritos nas Seções 4.3 e 5.1 respectivamente. Em alguns conjuntos foram adicionadas camadas de *Dropout* com taxa de 0.25 que ajudam a controlar o problema de *overfitting*.

A Tabela 1 apresenta a arquitetura da rede neural convolucional com os 7 conjuntos de camadas, com a quantidade de filtros, com o filtro 3x3 utilizado nas primeiras cinco camadas, a dimensão da saída da última camada do conjunto, o tipo de função de ativação, a presença da camada de *dropout* no conjunto, a dimensão da camada de *max pooling*, a presença da rede totalmente conectada, a presença de camada de *batch normalization*, a presença de camada convolucional e a quantidade de camadas em cada conjunto. A Figura 17 ilustra a arquitetura da rede com a distribuição das camadas e dimensão das saídas de cada conjunto de camadas.

Tabela 1: Configuração da arquitetura da CNN[2]

C	Filtro	N.F.	Dim.	Ativ.	Drop.	M.P.	T.C.	B.N.	Conv.	Qtd.
1	3x3	32	32x32	ReLU	Sim	3x3	-	Sim	Sim	5
2	3x3	64	32x32	ReLU	-	-	-	Sim	Sim	3
3	3x3	64	16x16	ReLU	Sim	2x2	-	Sim	Sim	5
4	3x3	128	16x16	ReLU	-	-	-	Sim	Sim	3
5	3x3	128	8x8	ReLU	Sim	2x2	-	Sim	Sim	5
6	-	-	1024x1	ReLU	Sim	-	Sim	Sim	-	5
7	-	-	5x1	Softmax	-	-	Sim	-	-	2

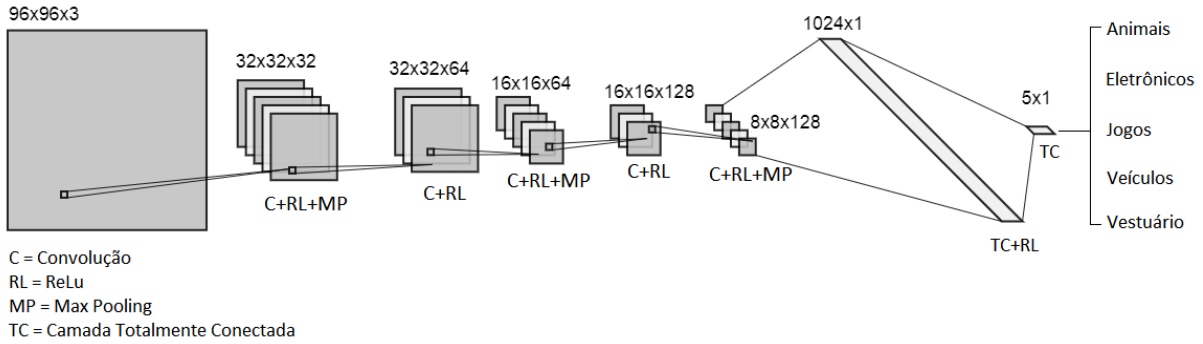


Figura 17: Arquitetura da Rede Neural Convolucional

A imagem é representada como três matrizes de dimensão 96x96, cada matriz representando um canal de cor *RGB*, na entrada da rede neural convolucional conforme ilustrado na Figura 17. Em seguida são aplicados os processos de convolução, ReLU e *max pooling* sucessivamente a fim de reduzir a dimensionalidade da matriz de entrada e complexidade computacional. O *max pooling* é aplicado com uma janela de dimensão 2x2 e 3x3. Em cada camada de convolução são aplicados os filtros 3x3 com 32 filtros no primeiro conjunto, 64 filtros no segundo e terceiro conjuntos e 128 filtros no quarto e quinto conjuntos. Após a primeira etapa de convolução no primeiro conjunto, a imagem passa de uma dimensão 96x96x3 para uma dimensão de 96x96x32 devido aos 32 filtros de convolução aplicados. Ainda no primeiro conjunto o procedimento de *max-pooling* reduz a dimensão de saída para 32x32x32 com uma janela de 3x3. Este procedimento é repetido ao longo dos conjuntos até a camada totalmente conectada. As camadas escondidas são contempladas com a função ReLU (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Por fim as camadas convolucionais são seguidas de duas camadas totalmente conectadas densas: uma com 1024 canais e a camada final com função de ativação *softmax* com as cinco classes de saída. Conforme relatado em (SIMONYAN; ZISSERMAN, 2015), os filtros 3x3 aumentam consideravelmente a performance do treinamento.

4.3 Processamento das imagens

Foram desenvolvidos scripts *python* para o carregamento e processamento das imagens. O procedimento de tratamento das imagens inicia com o carregamento das imagens de pastas para a memória RAM conforme ilustrado na Figura 18.

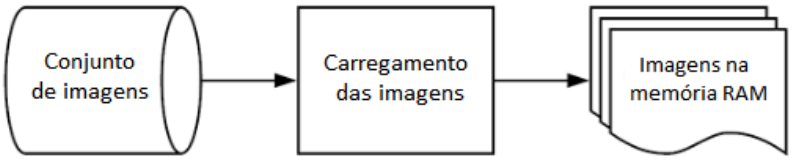


Figura 18: Carregamento das imagens

[2]Conjuntos de camadas (C), quantidade de filtros (N.F.) de dimensão (Dim.), camada de ativação (Ativ.), dropout (Drop.), Max pooling (M.P), camada totalmente conectada (T.C.), *batch normalization* (B.N), Convolução (Conv.) e quantidade de camadas (Qtd.).

As imagens utilizadas no treinamento passaram por técnicas de processamento visando aumentar a performance do modelo classificador e diminuir o esforço computacional necessário para processá-las. As imagens foram redimensionadas para um padrão de 96px de largura e 96px de altura.

Foi realizado o procedimento de *data augumentation* que é uma técnica para aumentar diversidade de dados disponíveis. Este processo foi realizado com rotações variando em 25 graus em conjunto com *zoom* nas imagens a fim de aumentar a capacidade de generalização da rede.

A intensidade dos *pixels* foi normalizada dividindo seus valores escalares por 255 de forma para variarem num intervalo de 0 a 1. As matrizes das imagens foram representadas com uma dimensão de 96x96x3 com 3 canais de cores conforme a escala de cores RGB (*Red Green Blue*). Os dados foram divididos em treino e teste sendo 75% para treino e 25% para teste. A Figura 19 ilustra o procedimento de divisão dos dados.

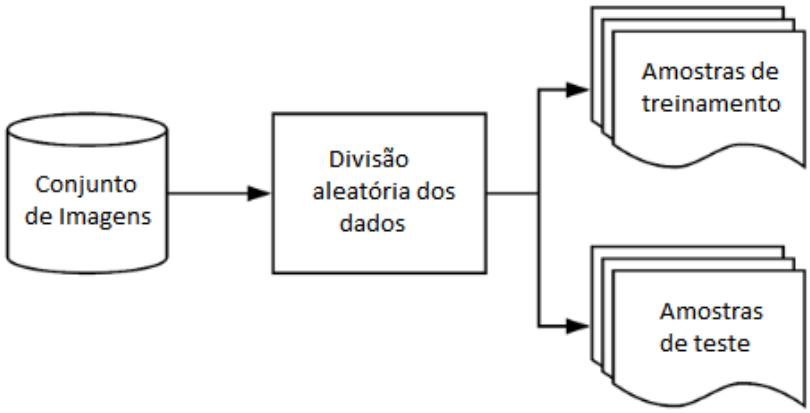


Figura 19: Divisão dos dados em treino e teste

4.4 Métricas de qualidade

Ao final dos experimentos, o desempenho de cada algoritmo foi analisado quanto ao erro, acurácia, sensibilidade, precisão e a medida F1. Estas métricas são descritas como as mais utilizadas algoritmos de classificação conforme observado em (HOSSIN; M.N, 2015). Para medição destas métricas de desempenho foi utilizada a biblioteca de código aberto *sklearn*. A acurácia é a taxa de acertos que o modelo obteve. O cálculo da acurácia (AC) é apresentado na Equação 43:

$$AC = \frac{TP + TN}{TP + FN + TN + FP}, \tag{43}$$

onde TP é o número de verdadeiros positivos, TN é o número de verdadeiros negativos, FN são os falsos negativos e FP são os falsos positivos. A precisão (*precision*) é definida como a taxa de acertos positivos dentre os classificados como positivos. O cálculo da precisão (PS) é apresentado na Equação 44:

$$PS = \frac{TP}{TP + FP}. \tag{44}$$

A sensibilidade é definida como a taxa de acertos positivos dentre todos os positivos. Pode ser definida como a frequência em que o classificador encontra os exemplos de uma classe. O cálculo da sensibilidade (RC) é apresentado na Equação 45:

$$RC = \frac{TP}{TP + FN}. \tag{45}$$

A especificidade é usada para medir a fração de padrões negativos que são classificados corretamente. O cálculo da métrica especificidade é apresentado na Equação 46:

$$SP = \frac{TN}{TN + FP}. \tag{46}$$

A métrica F1 combina a precisão e a sensibilidade de modo a trazer um número único que indique a qualidade geral do modelo treinado, ou seja, quanto maior melhor o modelo. Essa métrica representa a média harmônica entre valores de sensibilidade e precisão. O cálculo da métrica F1 é apresentado na Equação 47:

$$F1 = \frac{2 \times PS}{PS + RC}. \tag{47}$$

O modelo foi testado com 5 variações de algoritmos. Os resultados de cada abordagem foram então comparados e ordenados de acordo com sua performance em relação às métricas de desempenho.

4.5 Identificação de perfis

Após a análise do desempenho dos algoritmos de otimização, é realizada a escolha do algoritmo que gerou a melhor rede de classificação. Com o melhor modelo realiza-se classificação de 700 imagens de 20 perfis públicos da rede social *Instagram*. Propõe-se uma rotina que faça a leitura de um conjunto de imagens e classifique cada uma delas utilizando o modelo treinado. Em seguida, essa rotina realiza a medida do nível de interesse s_i^u (LEWENBERG; BACHRACH; VOLKOVA, 2015) denotada como *score* para cada usuário e definida como a probabilidade média atribuída por um classificador. A medida baseada em um *score* é uma abordagem relevante para predição do perfil de usuários (LAZZEZ; OUARDA; ALIMI, 2018). Conforme observado em (MERLER; CAO; SMITH, 2015), a distribuição de categorias representadas nas imagens postadas por um usuário está correlacionada ao seu interesse. Com base nisto, foi possível estabelecer uma abordagem baseada em contagem (MERLER; CAO; SMITH, 2015) definindo uma medida $C_i(u)$ denominada conceito ou frequência normalizada que quantifica o interesse do usuário em determinada área de interesse. Com a análise dos resultados realiza-se a identificação do perfil de interesse em relação às categorias de treinamento animais, eletrônicos, jogos, veículos e vestuário.

4.6 Considerações finais

Neste capítulo foram abordados todos os passos referentes ao desenvolvimento do método de treinamento da rede neural convolucional para classificação de imagens de redes sociais. Foram apresentados os procedimentos para processamento das imagens bem como a arquitetura da rede neural utilizada. Em seguida foram descritos os passos seguidos para o

desenvolvimento da solução e análise dos resultados. No próximo capítulo serão apresentados e discutidos todos os resultados provenientes das rotinas de treinamento e validação dos modelos desenvolvidos, comparando o desempenho de cada algoritmo de otimização da descida do gradiente e analisando o desempenho de cada um para a classificação de imagens de redes sociais e identificação do perfil de interesse de usuários.

Capítulo 5

EXPERIMENTOS E RESULTADOS

NESTE capítulo são apresentados e analisados os resultados obtidos com a metodologia aplicada no desenvolvimento deste trabalho. São abordados os processos para obtenção das imagens, definição dos hiperparâmetros para o treinamento do modelo e são apresentados os desempenhos de cada técnica de otimização de redes neurais utilizada. Com base nisto é realizada a escolha do melhor algoritmo e modelo apropriado para a classificação das imagens de usuários de redes sociais. Por fim os resultados são analisados e perfis de usuários são identificados utilizando o melhor modelo eleito.

5.1 Obtenção dos dados

A seleção das imagens para treinamento dos algoritmo foi baseada nos bancos de imagens do *Kaggle*, *pyimagesearch* e do *Computational Vision of Caltech*. Para os animais foi utilizado o *dataset* (SERMANET, 2014) e imagens públicas das redes sociais *Instagram* e *Flickr*, para os dispositivos eletrônicos foi utilizado o *dataset* (LI; ANDREETTO; RANZATO, 2003), para os jogos e dispositivos foram utilizadas imagens públicas obtidas diretamente da rede social *Instagram*, para os veículos foi utilizado o *dataset* (KHAYRULLIN, 2012), para o vestuário foram utilizados os *datasets* (ROSEBROCK, 2018) e (GAUTAM, 2017). Foram utilizadas 1.600 imagens com 5 categorias balanceadas dividas em 17 subcategorias. Foi realizada a divisão dos dados com 75% dos dados para treinamento e 25% para teste. Totalizam-se 1.200 imagens para treino que após o processo de *data augumentation* com a rotação, espelhamento e *zoom* das imagens foram geradas 11.840 imagens para o treinamento da rede neural convolucional. Para o teste totalizam-se 400 imagens. As imagens foram organizadas em pastas para realização de aprendizado supervisionado, sendo uma pasta para cada uma das categorias.

As categorias das imagens utilizadas com as respectivas subcategorias são descritas a seguir, sendo 5 categorias e 17 subcategorias:

- Animais domésticos: cachorros, gatos e pássaros;
- Dispositivos eletrônicos: celulares, notebooks e televisões;
- Jogos: dispositivos e cenas de jogos;
- Veículos: carros, motos, bicicletas e avião;
- Vestuário: calças, vestidos, blusas, sapatos e tênis;

5.2 Obtenção de hiperparâmetros

Após a definição da arquitetura da rede neural convolucional foram realizados experimentos para encontrar a combinação dos hiperparâmetros conforme procedimento realizado em (PINHEIRO, C. A. DE P.; NEDJAH, N.; MOURELLE, L. DE M., 2019). Tais hiperparâmetros são a taxa de aprendizado e o número de épocas. Em seguida são realizados os os treinamentos com os algoritmos de otimização. Foi utilizado o método clássico de inicialização dos pesos com valores baixos e aleatórios. Foram selecionados os valores de 0,001, 0,01 e 0,1 para a taxa de aprendizado realizando experimentos com 10, 20, 50, 100 e 200 épocas repetidos 10 vezes. Para a busca dos hiperparâmtros foi utilizado o algoritmo de otimização SGD nos experimentos. Os resultados médios para cada cenário, denotado como R, são apresentados na Tabela 5 do Apêndice A. O desempenho de cada experimento foi medido e foi realizada a escolha da melhor combinação para a classificação. A Tabela 5 do Apêndice A apresenta os resultados obtidos no treinamento contemplando o resultado, a taxa de aprendizado, número de épocas, erro mínimo, acurácia máxima, desvio padrão do erro (σ_E) e desvio padrão da acurácia (σ_A). Com o resultado dos experimentos, o melhor conjunto de hiperparâmetros para o treinamento foi de 0,1 de taxa de aprendizado e 100 épocas sendo estes utilizados no processo de treinamento e otimização da rede neural. Baseado na técnica de parada antecipada, a escolha de 100 épocas foi para evitar o processo de *overfitting*. Nota-se a acurácia alcançando 0,92 e o erro 0,11 para 100 épocas e 0,1 de taxa de aprendizado. A Figura 20 apresenta a matriz de confusão para o treinamento com 100 épocas e 0,1 de taxa de aprendizado. A matriz de confusão é uma tabela que apresenta as frequências de classificação para cada classe do modelo comparando as classes efetivas com as classes previstas pelo modelo (DIEZ, 2018). Na Figura 20, é possível verificar valores altos na diagonal principal da matriz conforme a escala de cores o que demonstra boa previsibilidade para o conjunto de testes. A matriz de confusão, calculada com os dados de teste, demonstra que a combinação dos hiperparâmetros é uma escolha apropriada para os treinamentos subsequentes. Notam-se 47 casos de acertos para a categoria de animais, 59 para eletrônicos, 71 para jogos, 63 para veículos e 55 para vestuário.

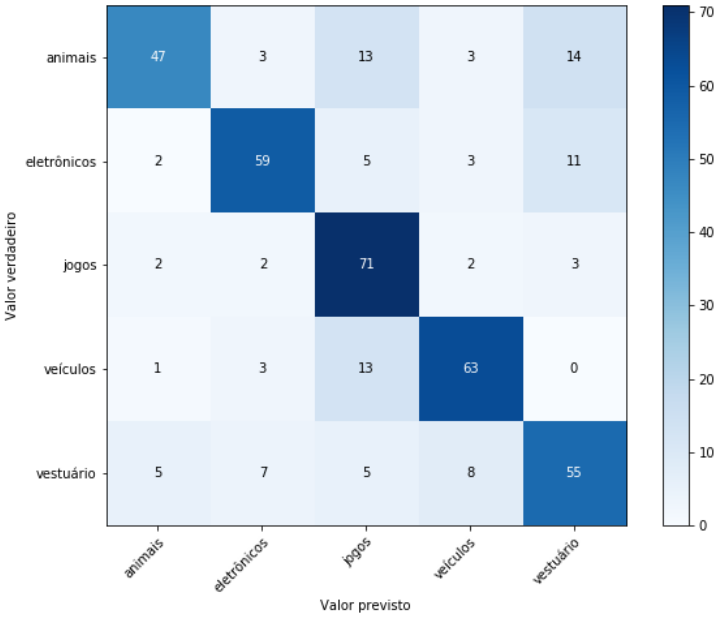


Figura 20: Matriz de confusão com 100 épocas e taxa de aprendizado 0,1

5.3 Validação cruzada

Foram realizados experimentos utilizando a validação cruzada (*k-Fold*) que é uma técnica para avaliar a capacidade de generalização de um modelo a partir de um conjunto de dados. Foi aplicada uma validação cruzada com 5 subconjuntos (*folds*) utilizado o algoritmo de otimização SGD a fim de verificar a qualidade dos dados utilizados. A biblioteca *sklearn* possui um método para validação cruzada que retorna um *score* de cada execução do *k-Fold* utilizando a acurácia como avaliação. O *score* varia de 0 até 1, onde 1 representa o melhor resultado. O resultado que ilustra o processo de validação cruzada com 5 subconjuntos é apresentado na Figura 21 com o *score* em cada retângulo preenchido, onde os retângulos brancos representam os conjuntos de treino e os retângulos destacados representam os conjuntos de testes. Os valores obtidos na validação cruzada (Figura 21) bem como os valores da matriz de confusão (Figura 20) demonstram uma qualidade nos dados utilizados bem como um balanceamento adequado.

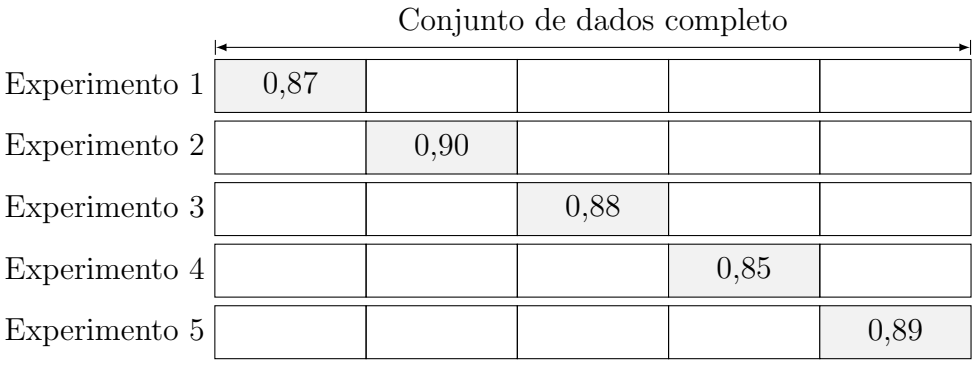


Figura 21: Resultado da validação cruzada com 5 subconjuntos

5.4 Otimização do treinamento

Após a definição dos hiperparâmetros foram utilizados os algoritmos de otimização da rede neural convolucional em busca de maiores desempenhos. Foram realizados experimentos com o total de 100 épocas e taxa de aprendizado igual a 0,1.

Foram realizados experimentos com os algoritmos de otimização SGD, AdaGrad, Adamax, RMSProp e Adam e analisados os resultados de cada um para a eleição do melhor algoritmo a ser utilizado.

5.4.1 Treinamento via algoritmo SGD

O algoritmo SGD possui um caráter estocástico em seu comportamento. É chamado de estocástico porque as amostras são selecionadas aleatoriamente ao invés de um único grupo em que aparecem no conjunto de treino. A Tabela 6 do Apêndice A apresenta os resultados médios, máximos, mínimos e o desvio para o erro de treino e teste, acurácia de treino e teste e das métricas de qualidade F-média, sensibilidade, especificidade e precisão. Nota-se valores moderados das métricas qualidade F-média, sensibilidade e precisão e especificidade com uma variação similar. As métricas de qualidade apresentam valores médios próximos e moderados. Nota-se um erro mínimo de 0,01 para o treino e 0,02 para o teste. A acurácia máxima para o treino é de 0,97 e para o teste é 0,93. A Figura 22 apresenta as curvas de acurácia para o treino e teste para o algoritmo SGD. Na Figura 22 é possível visualizar o crescimento da acurácia atingindo valores de 0,97 para treino e 0,93 para teste. Na Figura 23 nota-se o decaimento do erro ao longo das épocas para o treinamento e teste ambas com a curva de treino alcançando 0,01 e a curva de teste alcançando o valor de 0,02. A curva de erro de teste demonstra algumas instabilidades pontuais relacionadas ao caráter estocástico do algoritmo.

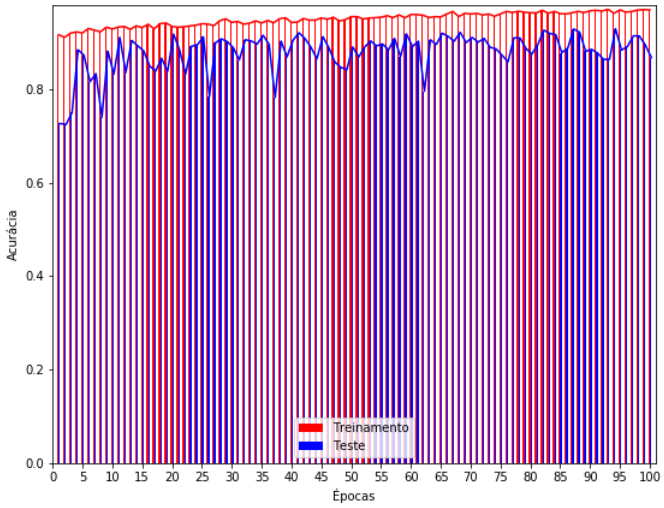


Figura 22: Crescimento da acurácia no treinamento e no teste com o algoritmo SGD

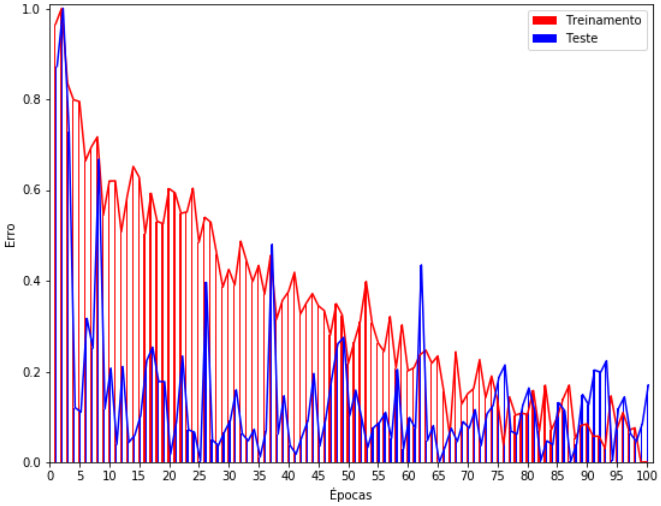


Figura 23: Decaimento do erro no treinamento e no teste com o algoritmo SGD

A Figura 24 apresenta as curvas de precisão e sensibilidade para o algoritmo SGD. Na Figura 24 é possível observar o aumento da taxa de acerto com o aumento da precisão e sensibilidade.

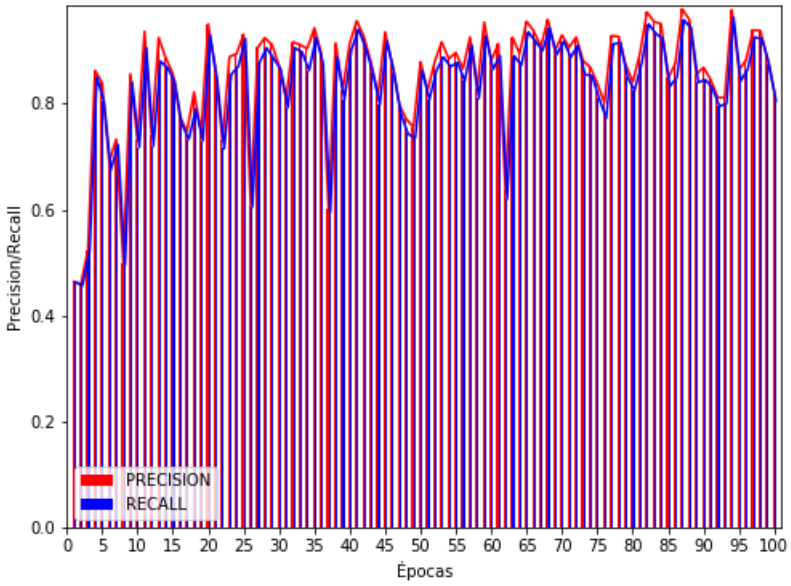


Figura 24: Métricas Precisão e Sensibilidade para o algoritmo SGD

Na Figura 25 é possível observar o aumento da taxa de acerto com o aumento das métricas F1 e especificidade.

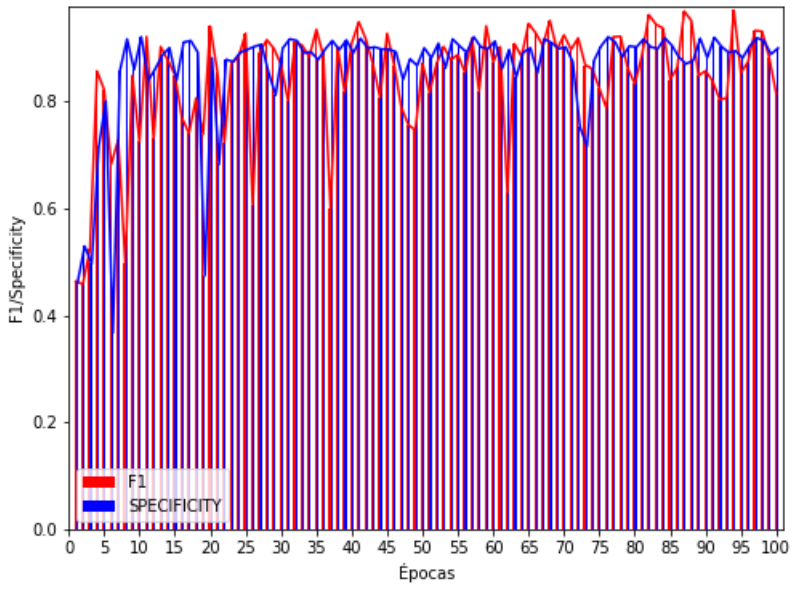


Figura 25: Métricas F1 e Especificidade para o algoritmo *SGD*

A Figura 26 apresenta a matriz de confusão para o SGD com a diagonal principal com valores elevados conforme escala de cores indicando boa capacidade de predição.

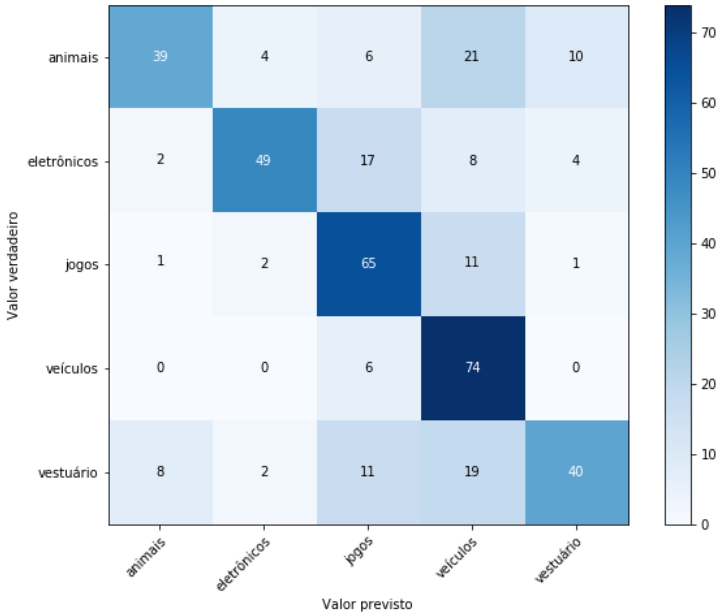


Figura 26: Matriz de confusão para o algoritmo *SGD*

Na Figura 26, nota-se uma uma assertividade de 39 casos para os animais, 49 casos para os eletrônicos, 65 casos para os jogos, 74 casos para os veículos e 40 casos para os vestuário.

5.4.2 Treinamento via algoritmo AdaGrad

O algoritmo *AdaGrad* tem o efeito de reduzir a taxa de aprendizado dos pesos que recebem gradientes altos e ao mesmo tempo aumentar a taxa de aprendizado nas atualizações durante o treinamento do algoritmo.

A Tabela 7 do Apêndice A apresenta os resultados médios, máximos, mínimos e o desvio para o erro de treino e teste, acurácia de treino e teste e das métricas de qualidade F-média, sensibilidade, especificidade e precisão. As métricas de qualidade apresentam valores médios próximos exceto a precisão que apresenta resultados superiores. Nota-se um erro mínimo de 0,005 para o treino e 0,004 para o teste. A acurácia máxima para o treino é de 0,92 e para o teste é 0,91 indicando o AdaGrad como um otimizador que aumenta o desempenho da rede neural.

A Figura 27 apresenta as curvas de acurácia para o treino e teste para o algoritmo AdaGrad. O AdaGrad demonstra uma estabilidade maior relação ao SGD. Conforme a Figura 27 a curva de acurácia do algoritmo AdaGrad demonstra boa estabilidade a atingindo valores elevados de 0,92 para o treino e 0,91 para o teste.

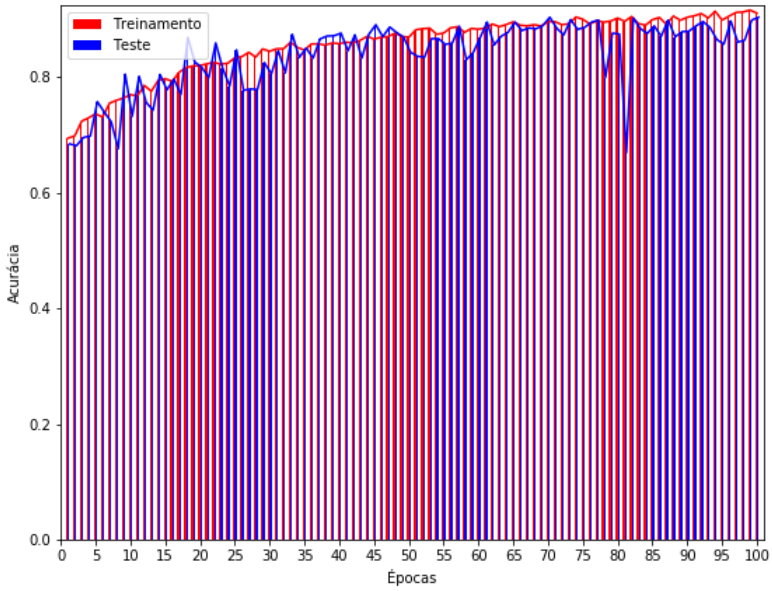


Figura 27: Crescimento da acurácia no treinamento e no teste com o algoritmo *AdaGrad*

A Figura 28 apresenta o decaimento do erro ao longo das épocas tanto nas curvas de treinamento e na curva de teste. Nota-se o erro decaindo para um mínimo de 0,005 para treinamento e 0,004 para teste. A curva de erro de teste demonstra algumas instabilidades pontuais porém bem inferior ao SGD. Com isso, observa-se uma estabilidade maior ao longo das épocas tanto para o crescimento da acurácia quanto para o decaimento do erro.

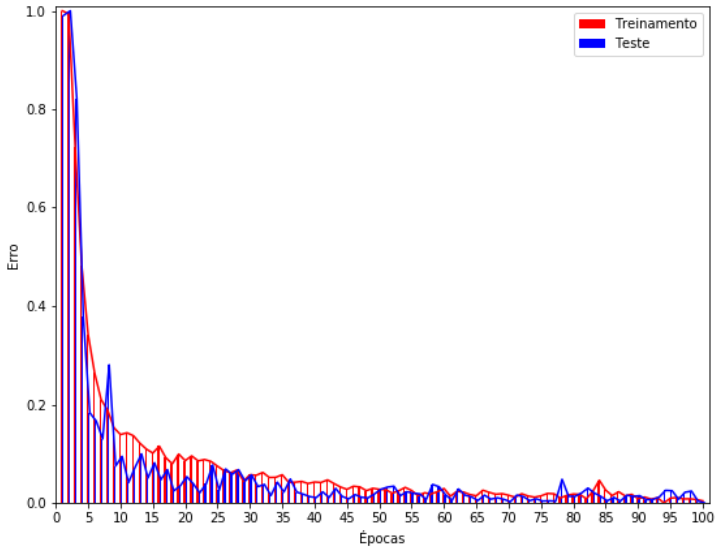


Figura 28: Decaimento do erro no treinamento e no teste com o algoritmo *AdaGrad*

A Figura 29 apresenta o aumento da taxa de acerto com o aumento da precisão e sensibilidade.

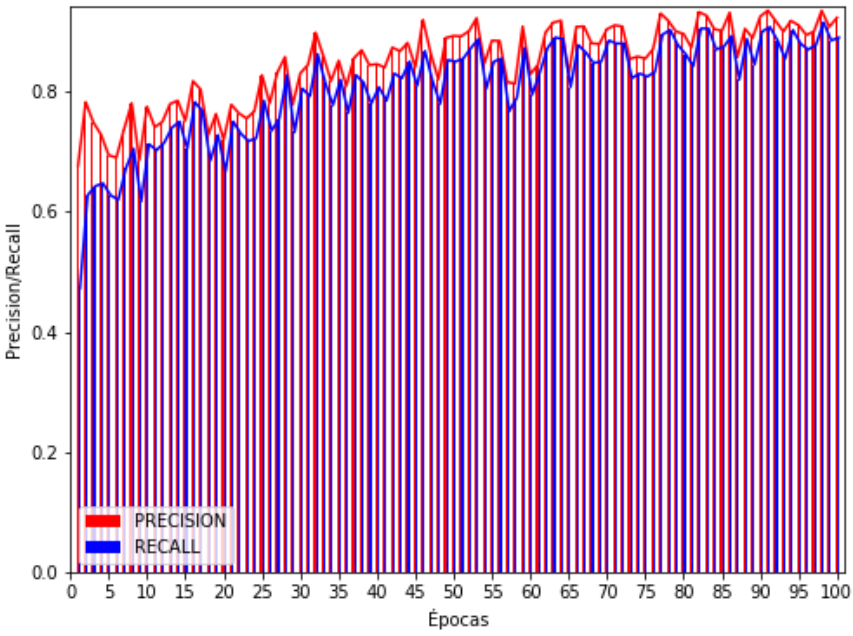


Figura 29: Métricas Precisão e Sensibilidade para o algoritmo *AdaGrad*

A Figura 30 apresenta o aumento da taxa de acerto com o aumento das métricas F1 e especificidade para o algoritmo AdaGrad.

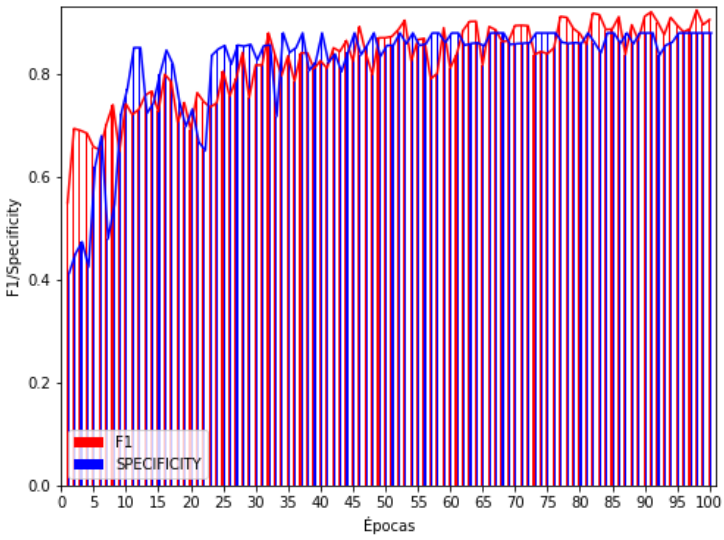


Figura 30: Métricas F1 e Especificidade para o algoritmo *AdaGrad*

A Figura 31 apresenta a matriz de confusão com valores mais elevados do que no algoritmo SGD na diagonal principal indicando uma maior taxa de acertos do algoritmo.

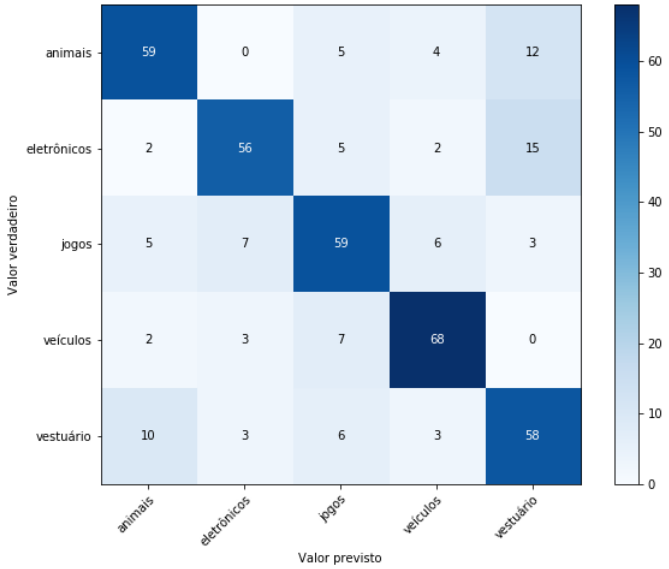


Figura 31: Matriz de confusão para o algoritmo *AdaGrad*

5.4.3 Treinamento via algoritmo RMSprop

A Figura 32 apresenta o crescimento da acurácia atingindo um desempenho de 0,98 no treinamento e 0,93 no teste. Ambas as curvas de treinamento e teste apresentam boa estabilidade no aumento da acurácia.

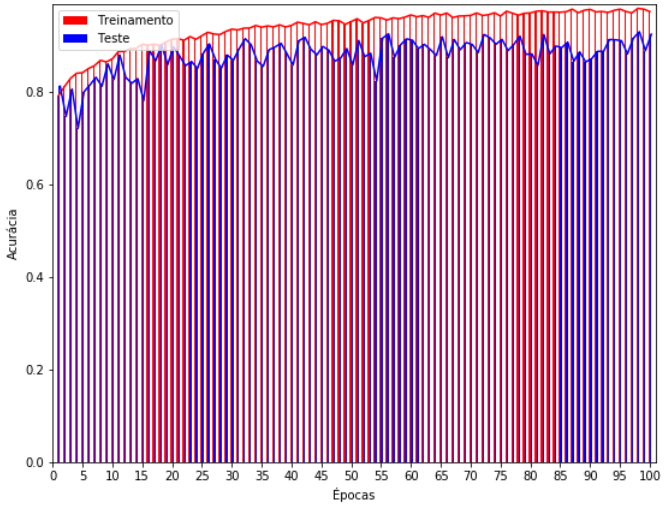


Figura 32: Crescimento da acurácia no treinamento e no teste com o algoritmo *RMSprop*

A Figura 33 apresenta o erro decaindo para o mínimo de 0,003 para treinamento e 0,004 para teste. A curva de teste apresenta algumas instabilidades porém um decaimento geral com estabilidade moderada ao longo das épocas.

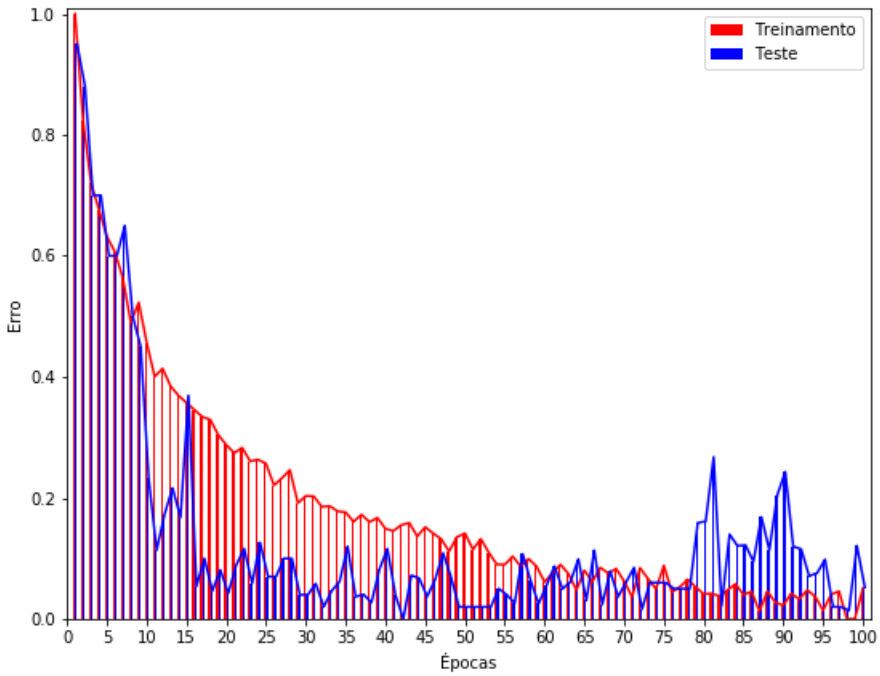


Figura 33: Decaimento do erro no treinamento e no teste com o algoritmo *RMSprop*

A Tabela 8 do Apêndice A apresenta os resultados médios, máximos, mínimos e o desvio para o erro de treino e teste, acurácia de treino e teste e das métricas de qualidade

F-média, sensibilidade, especificidade e precisão.

Na Tabela 8 do Apêndice A notam-se valores superiores para as métricas de qualidade F-Média, sensibilidade e precisão e especificidade em relação aos algoritmos anteriores demonstrando a qualidade do algoritmo *RmsProp*.

Nota-se uma acurácia de treino que alcança o valor de 0,98 e de teste 0,93. Esse resultado demonstra a superioridade desse algoritmo em relação ao AdaGrad e SGD. O erro alcança valores de 0,003 para o treino e 0,004 para o teste conforme Figura 33.

A Figura 34 apresenta o aumento da taxa de acerto com o aumento das métricas precisão e sensibilidade para o algoritmo *RmsProp*.

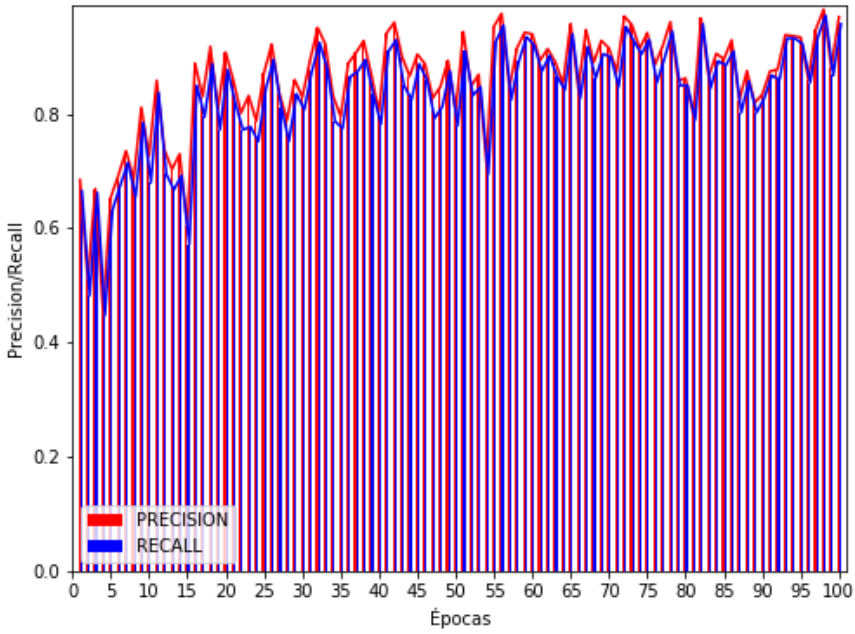


Figura 34: Métricas Precisão e Sensibilidade para o algoritmo *RmsProp*

A Figura 35 apresenta o aumento da taxa de acerto com o aumento das métricas F1 e especificidade.

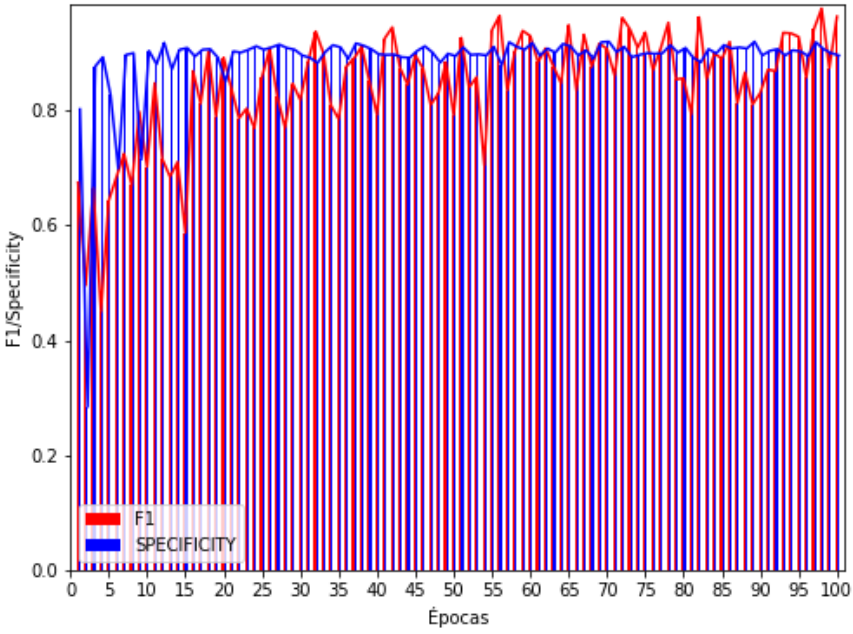


Figura 35: Métricas F1 e Especificidade para o algoritmo *RmsProp*

A matriz de confusão apresenta valores elevados em sua diagonal principal com superioridade em relação aos algoritmos anteriores indicando boa performance de predição.

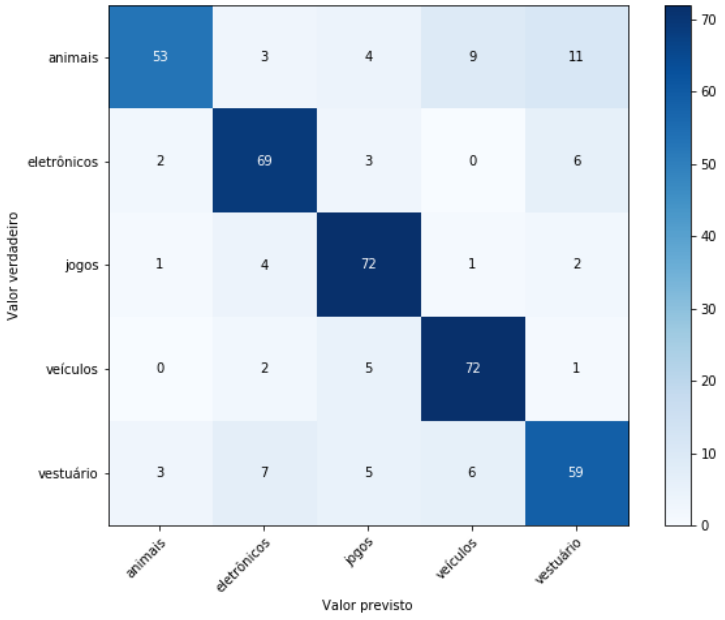


Figura 36: Matriz de confusão para o algoritmo *RMSprop*

5.4.4 Treinamento via algoritmo Adam

O algoritmo de treinamento de estimativa de momento adaptativo é um método derivado do *RMSprop* que ajusta o método *Adagrad* para que a taxa de aprendizado não diminua agressivamente.

A Tabela 9 do Apêndice A apresenta os resultados médios, máximos, mínimos e o desvio para o erro de treino e teste, acurácia de treino e teste e das métricas de qualidade F-média, sensibilidade, especificidade e precisão. Nota-se valores moderados das métricas qualidade F-Média, sensibilidade e precisão e especificidade variando em torno de 0,83 em seu valor máximo. Nota-se uma acurácia de treino alcançando o valor de 0,98 para o treino e 0,93 para o teste demonstrando um bom desempenho com a utilização de otimização da rede neural.

A Figura 37 apresenta o crescimento da acurácia atingindo valores de 0,97 para treino e 0,93 para teste demonstrando um bom desempenho no uso de otimizadores de redes neurais artificiais. Ambas as curvas de treinamento e teste apresentam boa estabilidade no aumento da acurácia.

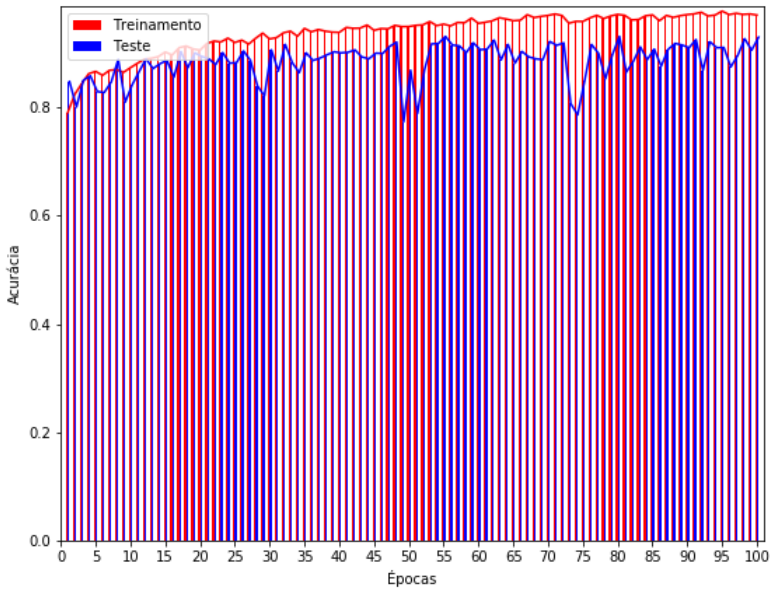


Figura 37: Crescimento da acurácia no treinamento e no teste com o algoritmo *Adam*

A Figura 38 apresenta o decaimento do erro ao longo das épocas para o treinamento e teste utilizando o algoritmo Adam como otimizador. Na Figura 38 o erro alcança o valor de 0,002 para treinamento e 0,004 para teste. A curva de erro de teste apresenta algumas instabilidades porém as duas apresentam um claro decaimento ao longo das épocas.

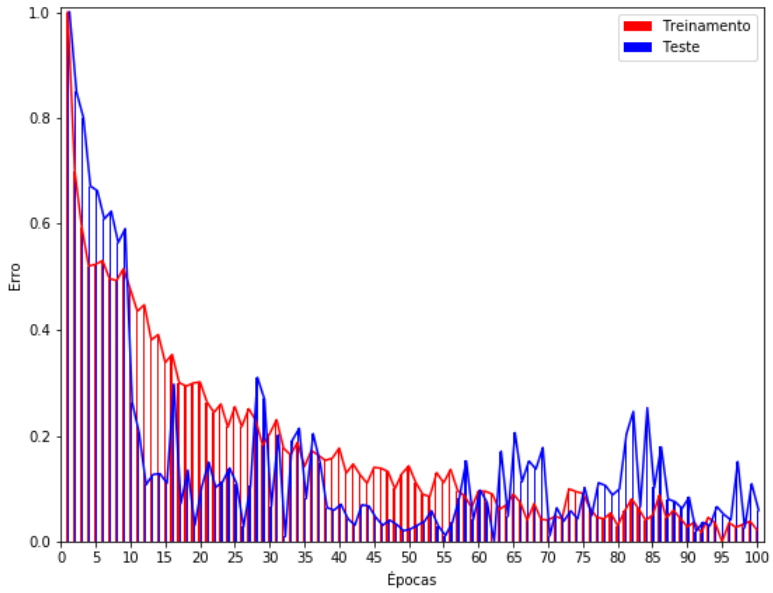


Figura 38: Decaimento do erro no treinamento e no teste com o algoritmo *Adam*

A Figura 39 apresenta o aumento da taxa de acerto com o aumento da precisão e sensibilidade.

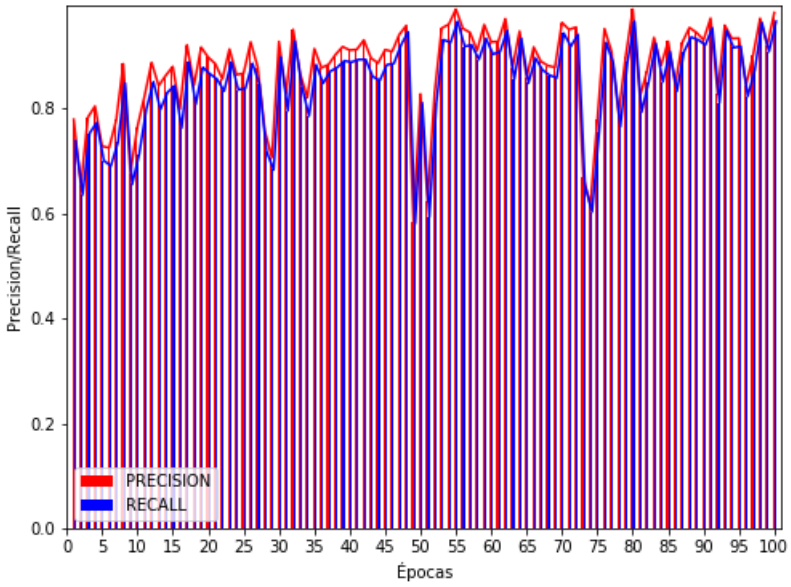


Figura 39: Métricas Precisão e Sensibilidade para o algoritmo *Adam*

Na Figura 40 é possível observar o aumento da taxa de acerto com o aumento das métricas F1 e especificidade.

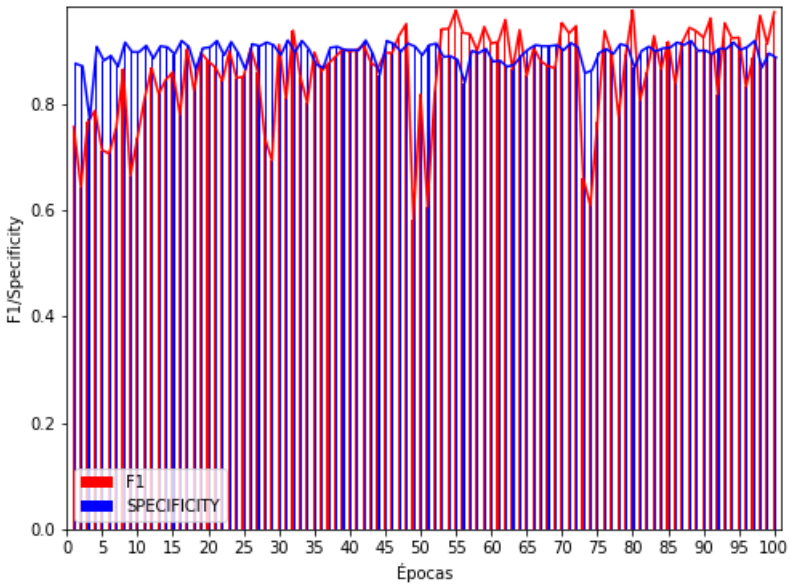


Figura 40: Métricas F1 e Especificidade para o algoritmo *Adam*

A Figura 41 apresenta a matriz de confusão para o algoritmo Adam com a diagonal da matriz com valores elevados evidentes pela escala de cores indicando boa predição.

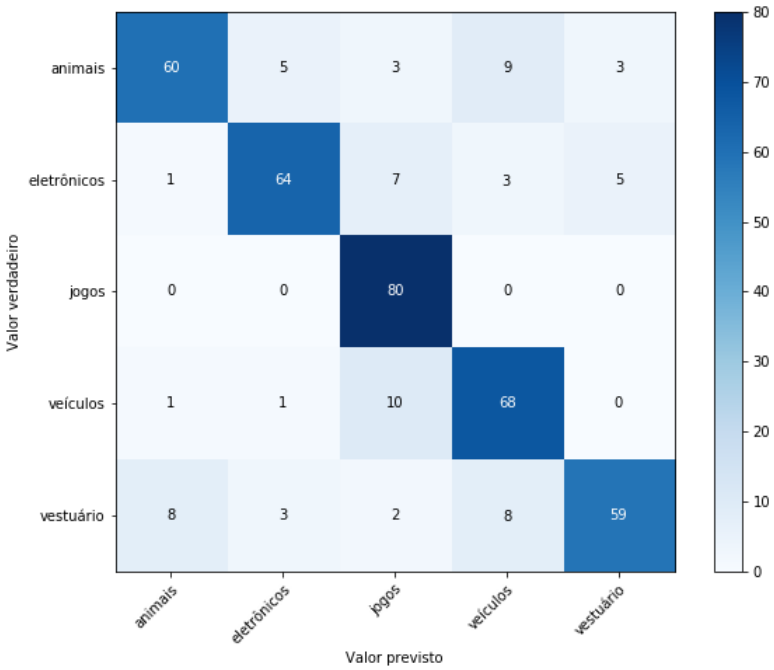


Figura 41: Matriz de confusão para o algoritmo *Adam*

5.4.5 Treinamento via algoritmo Adamax

O Adamax é um algoritmo variante do Adam onde o momento de segunda ordem é substituído pelo momento de ordem infinita (KINGMA; BA, 2014). A Tabela 10 do Apêndice A apresenta os resultados médios, máximos, mínimos e o desvio para o erro de treino e teste, acurácia de treino e teste e das métricas de qualidade F-média, sensibilidade, especificidade e precisão.

Nota-se valores moderados das métricas qualidade F-Média, sensibilidade e precisão e especificidade variando em torno de 0,82 em seu valor máximo. Nota-se uma acurácia de treino alcançando o valor de 0,96 para o treino e 0,93 para o teste demonstrando um bom desempenho com a utilização de otimização da rede neural.

O erro alcança valores de 0,005 para o treino e 0,003 para o teste conforme Figura 38. A Figura 42 apresenta o crescimento da acurácia nas curvas de treinamento e teste alcançando o valor máximo de 0,96 para treino e 0,93 para teste.

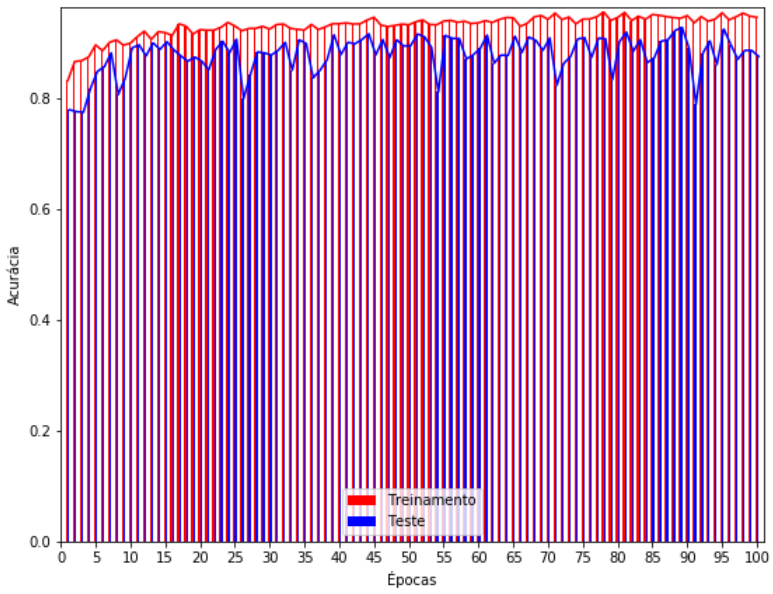


Figura 42: Crescimento da acurácia no treinamento e no teste com o algoritmo *Adamax*

A Figura 43 apresenta as curvas de decaimento do erro e do teste ao longo das 100 épocas. Notam-se instabilidades em ambas as curvas porém um claro um decaimento do erro ao longo das épocas demonstrando a boa utilização de algoritmos de otimização.

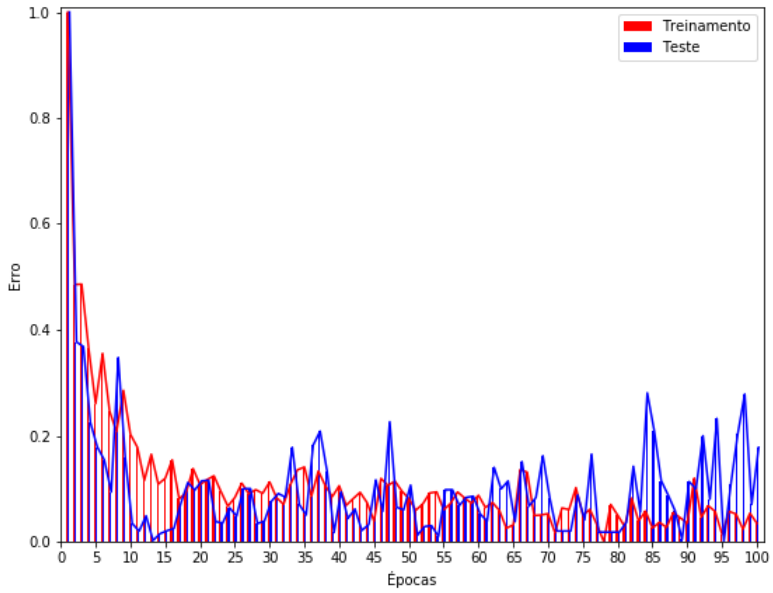


Figura 43: Decaimento do erro no treinamento e no teste com o algoritmo *Adamax*

A Figura 44 apresenta o aumento da taxa de acerto com o aumento da precisão e sensibilidade.

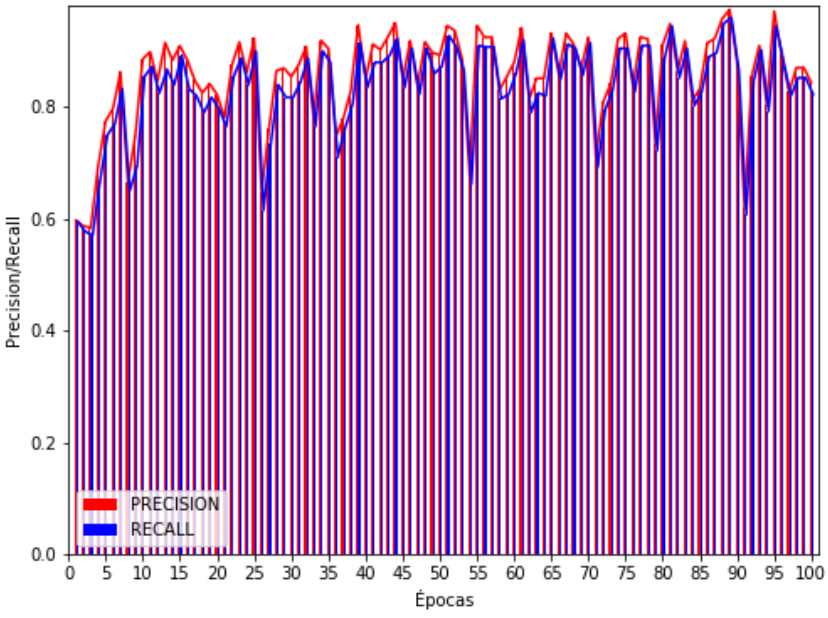


Figura 44: Métricas Precisão e Sensibilidade para o algoritmo *Adamax*

A Figura 45 apresenta o aumento da taxa de acerto com o aumento das métricas F1 e especificidade.

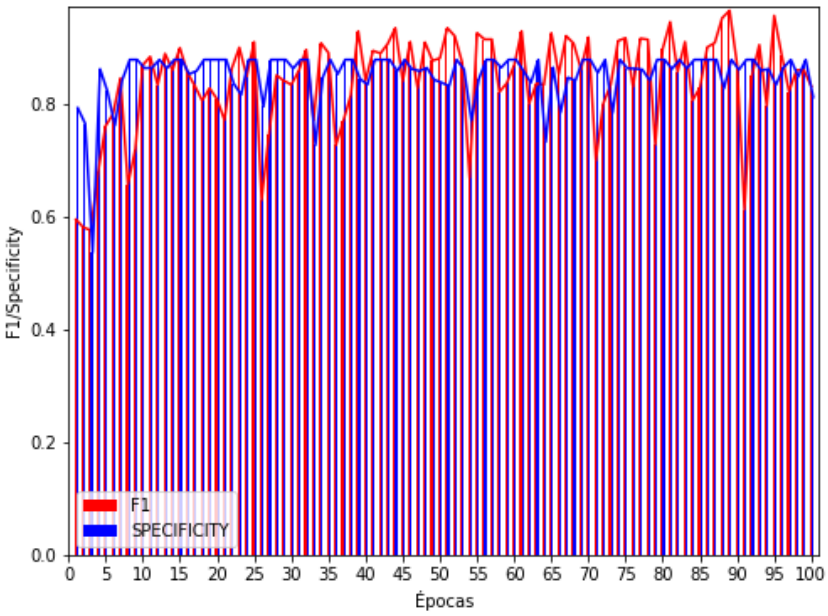


Figura 45: Métricas F1 e Especificidade para o algoritmo *Adamax*

A Figura 46 ilustra a matriz de confusão para o algoritmo *Adamax*. Nota-se que a diagonal principal da matriz de confusão possui valores elevados segundo a escala de cores.

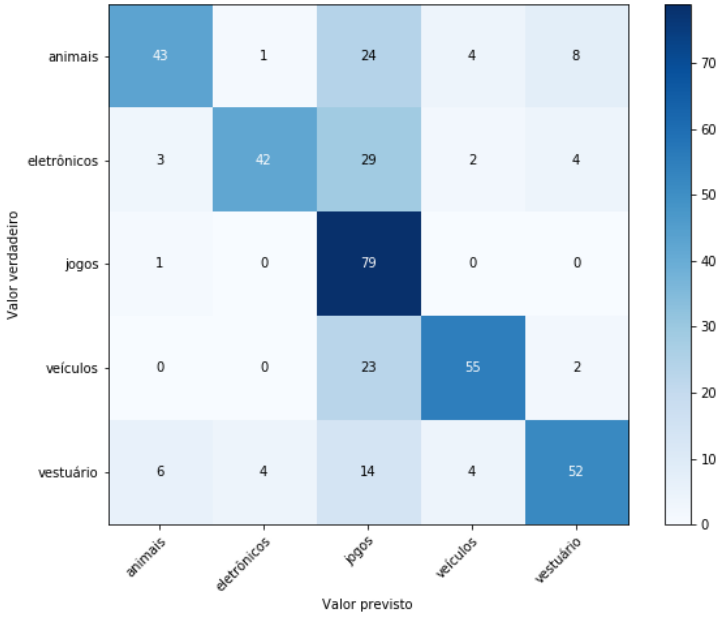


Figura 46: Matriz de confusão para o algoritmo *Adamax*

O algoritmo Adamax algoritmo apresentou resultados inferiores aos resultados do *Adam*, conforme observado na diagonal principal da matriz de confusão na Figura 46, demonstrando menor taxa de acertos de cada classe.

5.5 Eleição do melhor método

A Tabela 11 do Apêndice A apresenta a comparação entre o desempenho dos algoritmos de otimização em relação a acurácia média para o treinamento (Acc. Treino), erro médio de treinamento, a acurácia média para o teste (Acc. Teste), erro médio no teste, e somatório da diagonal principal da matriz de confusão (Conf.) referente aos resultados de teste. A Tabela 11 do Apêndice A permite verificar a diferença de desempenho dos algoritmos na etapa de treino e teste. Nota-se que o algoritmo *Adam* obteve alta acurácia sendo 0,94 para treino e 0,90 para teste. O algoritmo Adam apresentou erro médio de 0,10 para teste sendo estes inferiores aos erros do SGD, RMSProp, Adamax e AdaGrad. Nota-se que o algoritmo AdaGrad também obteve bom desempenho em alta acurácia no teste de 0,84 e erro médio no teste de 0,12 conforme a Tabela 11 do Apêndice A.

Em relação a matriz de confusão da Figura 41, nota-se o alto somatório da diagonal principal do algoritmo Adam de 331 sendo superior aos demais. O algoritmo Adam apresentou 0,90 na acurácia de teste e 0,10 no erro de teste conforme a Tabela 11 do Apêndice. De posse destes dados, a eleição do melhor algoritmo de otimização e treinamento é definido como o Adam conforme alta acurácia, baixo erro e alta assertividade na classificação, sendo o mais indicado para a predição de perfis de usuários.

5.6 Análise de desempenho

Com a escolha do melhor algoritmo de otimização sendo o *Adam* foram realizados testes com imagens de cada subcategoria a fim de avaliar o desempenho na classificação unitária de imagens. Após estes resultados foi possível utilizar o algoritmo para classificação de um conjunto maior de imagens com o intuito de identificar perfis. Na Tabela 2 são apresentados os resultados de taxa de acerto de cada categoria nos testes utilizando o algoritmo Adam com o melhor desempenho de treino para imagens das 5 categorias.

Tabela 2: Resultado da classificação de imagens nas categorias de treinamento

Anim.(%)	Eletr.(%)	Jog.(%)	Veíc.(%)	Vest.(%)	Imagem
96,71	0,94	0,03	0,11	2,21	Cachorro
98,92	0,09	0,01	0,24	0,75	Gato
99,64	0,01	0,01	0,33	0,01	Pássaro
0,00	**99,92**	0,05	0,01	0,02	Notebook
0,00	**98,89**	0,76	0,34	0,01	TV
0,01	**98,60**	0,07	0,00	1,32	Celular
0,00	6,58	**93,13**	0,00	0,29	Console
0,00	0,01	**99,93**	0,07	0,00	Jogo
0,00	0,00	0,03	**99,96**	0,00	Carro
0,00	0,00	0,08	**99,91**	0,01	Moto
0,00	0,02	0,58	**99,19**	0,21	Bicicleta
0,00	0,53	0,16	**99,26**	0,04	Avião
0,24	2,06	0,20	0,81	**96,69**	Tênis
6,27	0,87	0,30	1,06	**91,50**	Sapato
0,05	0,01	0,00	0,00	**99,94**	Vestido
1,90	1,31	0,01	0,00	**96,78**	Calça
8,31	31,21	1,58	30,85	28,05	Cadeira
0,11	0,10	48,17	51,26	0,37	Jardim
13,29	46,64	33,39	4,31	2,37	Torre

Os resultados da Tabela 2 foram obtidos utilizando amostras aleatórias das imagens pertencentes a dados de testes que não foram utilizados pelos algoritmo no treinamento. Em relação a classificação das categorias cadeira, jardim e torre os resultados indicaram baixas probabilidades pois essas classes não foram utilizadas no treino. Notam-se valores superiores a 99% de desempenho demonstrando boa capacidade de generalização.

5.7 Identificação de perfis de usuários

De forma semelhante ao procedimento realizado em (YOU et al., 2014) e em (HU; MANIKONDA; KAMBHAMPATI, 2014), foram realizadas predições de uma seleção de amostras de imagens de determinados perfis em redes sociais. Foi desenvolvida uma rotina que faz a leitura de um conjunto de imagens, classifica cada uma delas e realiza a média das probabilidades do pertencimento a cada classe ao final do processo. Foram selecionadas 700 imagens de 20 perfis públicos do *Instagram* para análise. Conforme relatado em (LEWENBERG; BACHRACH; VOLKOVA, 2015) é possível definir uma medida de interesse de um usuário. Sendo U uma amostra de usuários e cada usuário $u \in U$, define-se i como interesse $i \in I$, onde I são determinados interesses. Cada usuário u de conjunto usuários U de uma rede social pode ser analisado de forma a observar o grau de pertinência de interesse i dos potenciais interesses I definidos neste trabalho como animais, eletrônicos, jogos, veículos e vestuário. A medida do nível interesse s_i^u, comumente denotada como *score*, para um determinado usuário alvo $u \in U$ para a área de interesse $i \in I$, pode ser definida como a probabilidade média atribuída por um classificador conforme a Equação 48:

$$s_i^u = \frac{\sum_{j=1}^{N_u} p_i^u(j)}{N_u}, \tag{48}$$

onde p_i^u é a probabilidade resultante de um classificador para um usuário u na área de interesse i e N_u é o número de imagens pertencentes a um usuário u com $j = 1, ..., Nu$.

A medida baseada em um *score* é uma abordagem relevante para predição do perfil de usuários conforme relatado em (LAZZEZ; OUARDA; ALIMI, 2018). Conforme observado em (MERLER; CAO; SMITH, 2015), as categorias representadas nas imagens postadas por um usuário estão relacionadas ao seu interesse. Em (MERLER; CAO; SMITH, 2015) é realizada uma abordagem baseada em contagem. Baseado nesta abordagem é possível definir uma medida $C_i(u)$ denominada conceito ou frequência normalizada. Esta medida quantifica o interesse do usuário u conforme probabilidade $C_i(x_j)$ atribuída por um classificador ao classificar uma dada imagem x, caso esta probabilidade seja maior que determinado limite t (*threshold*) especificado conforme Equação 49:

$$C_i(u) = \frac{\sum_{j=1}^{N_u} T_i(x_j)}{N_u}, \tag{49}$$

onde o termo $T_i(x_j)$ é definido conforme Equação 50:

$$T_i(x_j) = \begin{cases} C_i(x_j) & \text{se } C_i(x_j) > t \\ \\ 0 & \text{senão.} \end{cases} \tag{50}$$

Conforme relatado em (LAZZEZ; OUARDA; ALIMI, 2018) o entendimento das imagens compartilhadas em redes sociais pelos usuários permite a análise do seu perfil de interesse. Primeiramente é feita a análise da característica visual de suas imagens compartilhadas individualmente. Em seguida, com a agregação das informações no nível do conjunto de imagens é possível prever o interesse do usuário em determinada área.

Os avanços no número de usuários de redes sociais, por sua vez, tornam necessários estudos na área de visão computacional, com o objetivo de prever o seu interesse em determinada área baseado na classificação de imagens postadas.

A medida de conceito $C_i(u)$ está diretamente relacionada à frequência de ocorrência de classificações superiores ao *threshold* estabelecido permitindo avaliar melhor cada perfil de usuário. A medida $C_i(u)$ é uma métrica de avaliação relacionada à frequência de ocorrência de uma categoria porém normalizada variando de 0 até 1. Conforme realizado em (MERLER; CAO; SMITH, 2015) foi definido um limite *threshold* de 90% para realização da classificação das imagens.

O resultado do *score* obtido para as 700 imagens de 20 perfis para cada categoria de classificação das imagens dos usuários é apresentado na Tabela 3. Com os dados desta tabela é possível o início da análise para identificar o perfil de usuários por interesse nas categorias de treinamento. Valores elevados acima de 80% podem ser observados para as categorias. Notam-se *scores* elevados para a categoria animais para os usuários 1,2 e 3 respectivamente. Notam-se *scores* elevados para a categoria eletrônicos nos usuários 3,4 e 5. Nos usuários 16,17,18,19 e 20 notam-se baixos *scores* para as categorias pré-definidas devido ao interesse em outras categorias.

Tabela 3: Identificação do perfil de interesse de usuários baseado na métrica *score*

Usuário	**Anim.(%)**	**Eletr.(%)**	**Jog.(%)**	**Veíc.(%)**	**Vest.(%)**	**Interesse**
1	**97,18**	0,33	0,02	0,23	2,23	Animais
2	**96,56**	1,46	0,10	0,79	1,08	Animais
3	**84,35**	1,22	0,44	6,35	7,65	Animais
4	0,18	**81,82**	6,61	11,05	0,33	Eletrônicos
5	0,31	**83,69**	5,89	4,88	5,23	Eletrônicos
6	0,00	**77,56**	0,80	21,62	0,02	Eletrônicos
7	0,04	0,20	**92,80**	1,24	5,72	Jogos
8	0,03	13,55	**79,36**	6,78	0,29	Jogos
9	9,31	9,20	**71,43**	9,81	0,25	Jogos
10	0,42	0,52	4,96	**93,72**	0,38	Veículos
11	0,02	0,65	0,08	**98,81**	0,43	Veículos
12	0,14	4,18	0,03	**95,24**	0,40	Veículos
13	1,44	1,46	0,56	0,88	**95,67**	Vestuário
14	1,68	1,44	0,71	14,3	**81,86**	Vestuário
15	0,52	2,75	6,04	0,41	**90,28**	Vestuário
16	14,52	5,88	33,98	39,91	5,7	Esportes
17	10,61	17,84	8,8	48,35	14,39	Decoração
18	11,0	10,44	23,17	44,43	10,95	Alimentos
19	8,96	4,51	20,63	45,08	20,83	Viagens
20	2,95	5,57	37,15	45,75	8,59	Natureza

É possível observar *scores* para a categoria jogos para os usuários 7,8 e 9. Notam-se *scores* para veículos elevados para os usuários 10,11 e 12. Observam-se *scores* elevados para vestuário para os usuários 13,14 e 15. O resultado da medida conceito $C_i(u)$ calculado para as 700 imagens de 20 perfis para cada categoria obtido por meio da classificação das imagens dos usuários é apresentado na Tabela 4. Com os dados da Tabela 2 é possível complementar a análise para identificar o perfil de usuários por interesse nas categorias de treinamento. Notam-se *conceitos* elevados para animais para os usuários 1,2 e 3 respectivamente e *conceitos* elevados para eletrônicos para os usuários 3,4 e 5 respectivamente. Nos usuários 16,17,18,19 e 20 é possível observar baixos *conceitos* para as categorias pré-definidas devido ao interesse em outras categorias. Observam-se *conceitos* elevados para jogos para os usuários 7,8 e 9 respectivamente e para veículos para os usuários 10,11 e 12 respectivamente. Observam-se *conceitos* elevados para a categoria vestuário para os usuários 13,14 e 15 respectivamente. É possível observar valores elevados para as categorias demonstrando uma previsibilidade satisfatória para a identificação do perfil de clientes.

Tabela 4: Identificação do interesse de usuários baseado na frequência normalizada

Usuário	Anim.	Eletr.	Jog.	Veíc.	Vest.	Interesse
1	**0,91**	0,06	0,00	0,00	0,03	Animais
2	**0,85**	0,00	0,00	0,11	0,04	Animais
3	**0,97**	0,00	0,00	0,03	0,00	Animais
4	0,00	**0,85**	0,06	0,09	0,00	Eletrônicos
5	0,05	**0,81**	0,05	0,05	0,05	Eletrônicos
6	0,00	**0,88**	0,00	0,12	0,00	Eletrônicos
7	0,00	0,06	**0,83**	0,06	0,06	Jogos
8	0,00	0,09	**0,87**	0,04	0,00	Jogos
9	0,09	0,06	**0,71**	0,12	0,03	Jogos
10	0,00	0,00	0,04	**0,96**	0,00	Veículos
11	0,07	0,00	0,00	**0,93**	0,00	Veículos
12	0,00	0,00	0,05	**0,95**	0,00	Veículos
13	0,03	0,06	0,00	0,03	**0,89**	Vestuário
14	0,00	0,00	0,00	0,03	**0,97**	Vestuário
15	0,00	0,09	0,03	0,03	**0,86**	Vestuário
16	0,00	0,03	0,06	0,17	0,06	Esportes
17	0,00	0,00	0,04	0,19	0,00	Decoração
18	0,00	0,00	0,04	0,19	0,00	Alimentos
19	0,00	0,00	0,03	0,3	0,03	Viagens
20	0,00	0,02	0,19	0,26	0,02	Natureza

Com a classificação de imagens de redes sociais de perfis de usuários é possível identificar potenciais consumidores para produtos de determinada categoria conforme dados obtidos da classificação. Procedimentos desta natureza permitem o apoio à decisão e no desenvolvimento de sistemas de recomendação.

5.8 Considerações finais

Neste capítulo foram descritos os procedimentos para o treinamento da rede neural artificial e os resultados de cada algoritmo de otimização SGD, AdaGrad, Adamax, RMSProp e Adam. Foram descritos os procedimentos para obtenção dos hiperparâmetros e da escolha do melhor modelo para classificação de imagens e predição do perfil de usuários de redes sociais. Os resultados indicam que a utilização de algoritmos de otimização de redes neurais artificiais pode aumentar a performance na predição e classificação. Foi possível observar o bom desempenho e estabilidade na utilização do algoritmo AdaGrad, não obstante, foi possível constatar a superioridade no algoritmo Adam em relação aos demais sendo este o mais adequado para classificação de imagens. De posse do melhor modelo foram realizadas análises de 20 perfis de usuários de redes sociais com a classificação de imagens de suas redes sociais utilizando o melhor modelo treinado. Com os resultados da classificação das imagens foi possível identificar os perfis de usuários obtendo resultados satisfatórios.

Capítulo 6

CONCLUSÃO E TRABALHOS FUTUROS

ESTE capítulo completa esta dissertação, apresentando as principais conclusões obtidas durante o desenvolvimento do presente trabalho. São apresentadas também as possíveis direções para outros trabalhos futuros envolvendo aprendizagem profunda para a classificação de imagens e identificação do perfil de usuários utilizando inteligência computacional.

6.1 Conclusões

Neste trabalho foram utilizadas redes neurais convolucionais e aprendizagem profunda como técnicas de inteligência computacional, com o objetivo de realizar a predição do interesse de usuários de redes sociais em determinadas categorias utilizando a classificação de imagens. Esta análise é válida como fonte de conhecimento na identificação de consumidores por interesse e no desenvolvimento de ferramentas de apoio a decisão tais como sistemas de recomendação de produtos. Este trabalho é uma evolução da abordagem realizada em (AZEVEDO, V. R.; NEDJAH, N.; MOURELLE, L. DE M., 2019) onde foi demonstrada a relevância deste estudo.

Este estudo foi motivado pela crescente demanda de desenvolvimento de sistemas capazes de detectar o interesse de usuários de redes sociais. Com a crescente expansão das mídias sociais torna-se necessário investigar novas formas para a classificação das imagens e identificação do perfil de usuários. As redes sociais mais usadas do mundo são o *Facebook, Youtube, Whatsapp, Facebook Messenger, WeChat e Instagram.*

O *Instagram* é uma das redes sociais mais populares e possui mais de 1 bilhão de usuários ativos mensais. Todos os dias, os usuários do *Instagram* publicam em média mais de 100 milhões de fotos o que indica uma grande relevância para o estudo deste trabalho. Acredita-se que este trabalho contribui para a comunidade científica, no ramo de visão computacional e classificação de imagens, com seu caráter exploratório e experimental na análise do desempenho de redes neural artificial. As redes sociais oferecem oportunidades para investigar o comportamento do usuário por meio de informações contidas em seus perfis. Os critérios podem ser extraídos pelo conteúdo gerado pelo usuário, considerando a previsão de personalidade através de informações coletadas das mídias sociais, devido ao seu potencial em muitos aplicativos existentes atualmente.

Neste trabalho foram abordados os conceitos e definições de redes neural artificial, redes neurais convolucionais, aprendizagem profunda, descida do gradiente e os algoritmos de otimização da descida do gradiente para o treinamento de redes neurais artificiais. Foi

realizada a análise do desempenho de diferentes algoritmos de otimização da descida de gradiente para classificação de imagens. Com a comparação dos algoritmos de otimização é possível melhorar a qualidade da solução final de um modelo utilizando os métodos mais eficientes para aprendizagem de máquina.

Inicialmente buscou-se um banco de dados de imagens nas categorias animais, eletrônicos, jogos, veículos e vestuário divididas em 17 subcategorias sendo estas cachorros, gatos, pássaros, celulares, notebooks e televisões, consoles de jogos, cenas de jogos, carros, motos, bicicletas, avião, calças, vestidos, blusas, sapatos e tênis. Em seguida criou-se um ambiente virtual usando *Python* e *Keras* sendo estas as principais ferramentas utilizadas para análise de dados e aprendizado de máquinas na atualidade. O ambiente foi desenvolvido utilizando o *Docker* para inicialmente agir sobre as imagens da base de dados, aplicando rotinas de pré-processamento como a normalização e redimensionamento visando a otimização de esforço computacional. Além disto foram realizados procedimentos de *data augumentation* a fim de aumentar a diversidade de dados. Foram utilizadas imagens para treinamento divididas nas categorias animais, eletrônicos, jogos, veículos e vestuário sendo 75% para treino e 25% para teste. As métricas de desempenho e qualidade foram analisadas. As métricas analisadas foram a acurácia, erro médio quadrático, precisão, sensibilidade, especificidade e F-Média. De posse das métricas de qualidade foram desenvolvidas matrizes de confusão para ajudar na eleição do melhor modelo adequado para a classificação das imagens de redes sociais. O conhecimento de como os hiperparâmetros podem influenciar o desempenho da rede neural convolucional é muito importante ao treinar uma rede. Visando enriquecer a pesquisa realizada e eventualmente encontrar resultados ainda mais expressivos, investigou-se o desempenho dos modelos quando operando sob diferentes configurações de hiperparâmetros da rede neural convolucional. A combinação dos hiperparâmetros mais adequada de 0,1 para a taxa de aprendizado e 100 épocas demonstrou um maior desempenho. Vale ressaltar a relevância do balanceamento de classes realizado entre cada categoria, a validação cruzada realizada bem como a matriz de confusão para avaliação do modelo. De posse desses dados foi possível avaliar o desempenho e capacidade de generalização dos algoritmos bem como sua capacidade de predição.

O modelo proposto foi treinado e testado utilizando o algoritmo *backpropagation* variando os métodos de otimização da rede neural artificial. Algoritmos de treinamento e otimização são uma parte crucial da rede neural. Entender seu desempenho pode ajudar a escolher a melhor opção para os sistemas. A comparação dos algoritmos de treinamento SGD, AdaGrad, Adamax, RMSProp e Adam permitiu verificar que o algoritmo de treinamento Adam obteve acurácia e desempenhos superiores aos demais e estabilidade elevada. Foi possível constatar que o algoritmo Adam é uma boa opção para otimização de redes neurais artificiais, sendo um algoritmo estável e alcançando desempenhos elevados. Foi possível constatar também o bom desempenho do algoritmo AdaGrad e estabilidade elevada sendo um algoritmo com desempenho moderado para aplicações de classificação de imagens. Não obstante, o algoritmo Adam demonstrou ser o melhor algoritmo pois obteve a maior taxa de acertos gerais sendo o modelo eleito como mais adequado para a classificação das imagens de redes sociais. Com o algoritmo Adam foi possível alcançar uma acurácia de treino de valor de 0,98 para o treino e 0,93 para o teste, o erro alcança valores de 0,002 para o treino e 0,004 para o teste demonstrando um bom desempenho com a utilização de otimização da rede neural. Vale ressaltar que o somatório da diagonal principal da matriz de confusão para o algoritmo Adam demonstrou valores elevados indicando melhor taxa de acerto em relação aos demais algoritmos.

Com o melhor modelo definido com o Adam foram realizadas classificações de 700 imagens de 20 perfis públicos da rede social *Instagram*. Foi realizada uma rotina que faz a leitura de um conjunto de imagens e classifica cada uma delas utilizando o modelo treinado. Após todas as classificações foi realizada a análise dos resultados com base na métrica *score* e frequência normalizada geralmente utilizadas na identificação de perfis. Com a análise dos resultados foi possível realizar a identificação e dedução do perfil de interesse dos usuários em relação às categorias do treinamento. Após a finalização de todos os experimentos, foi realizada a classificação de imagens das categorias definidas e identificação dos perfis dos usuários. Foi possível validar o método proposto e obter desempenhos elevados de classificação das imagens. Este estudo permite o desenvolvimento de sistemas de apoio a decisão e de recomendação de produtos.

6.2 Trabalhos futuros

Nesta seção são citadas algumas possíveis modificações à metodologia proposta, com o intuito de melhorar o desempenho dos modelos desenvolvidos neste trabalho. Também são levantadas propostas para futuros trabalhos na área de identificação do perfil de interesse de usuários de redes sociais.

Para trabalhos futuros, uma outra abordagem seria variar as categorias das imagens de forma a obter classificações com maior capacidade de generalização. Com outras categorias de imagens é possível identificar outros tipos de perfis de usuários com enfoque em outros tipos de produtos ou serviços.

Para ampliar a gama de técnicas utilizadas e fornecer mais uma comparação aos algoritmos de otimização de redes neural artificial a técnica de *transfer learning* poderia ser utilizada. O *transfer learning* com algum conhecimento prévio faz uso de redes neurais complexas e previamente treinadas tais como *ResNet* (HE et al., 2015) desenvolvida pela *Microsoft* ou *ImageNet* (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) para simplificar o processo de desenvolvimento de novos modelos.

Uma outra possível extensão deste trabalho é variar a arquitetura da rede neural utilizada e avaliar a performance obtida na classificação. Além disto para identificar perfis, outra abordagem a ser utilizada são algoritmos Neuroevolucionários que combinam algoritmos genéticos com redes neural artificial. Visando ainda ampliar a verificação da superioridade do treinamento de redes neurais convolucionais com algoritmos de otimização, outros algoritmos poderiam ser testados, como por exemplo os algoritmos de otimização Nesterov, AMSGrad ou AdaDelta.

REFERÊNCIAS

AGGARWAL, C. C. et al. *Recommender systems*. São Paulo, Brasil: Springer, 2016.

ASLAM, S. *OmniCoreAgency: Instagram statistics for 2019*. 2019. Instagram by the numbers: usage stats, demographics and facts you need to know. Disponível em: <https://www.websitehostingrating.com/instagram-statistics/>.

AZEVEDO, V. R.; NEDJAH, N.; MOURELLE, L. DE M. Identificação do perfil de clientes utilizando redes neurais convolucionais. ABRICOM, CBIC, Belém, PA, p. 1–8, Novembro 2019.

BAATARJAV, E.; DANTU, R. Current and future trends in social media. In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. Boston, Massachusetts, USA: IEEE, 2011. p. 1384–1385.

BALLAN, L. et al. Combining generative and discriminative models for classifying social images from 101 object categories. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. Tsukuba, Japan: IEEE, 2012. p. 1731–1734. ISSN 1051-4651.

BENGIO, Y. Deep learning of representations: Looking forward. *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 1–37, 2013. ISSN 1611-3349.

BERGMAN, R.; NACHLIELI, H. Perceptual segmentation: Combining image segmentation with object tagging. *IEEE Transactions on Image Processing*, v. 20, n. 6, p. 1668–1681, June 2011. ISSN 1057-7149.

BIRCANOGLU, C.; ARICA, N. A comparison of activation functions in artificial neural networks. In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. Izmir, Turkey: SIU, 2018. p. 1–4.

CAMPANA, M.; DELMASTRO, F. Recommender systems for online and mobile social networks: A survey. *Online Social Networks and Media*, Volumes 3–4, p. 75–97, 10 2017.

CASTILLO, D. J. K. D. R. D. D. Recommender systems as a mobile marketing service. *Journal of Service Science and Management*, 2013.

CHATZILARI, E. et al. Leveraging social media for scalable object detection. *Pattern Recognition*, v. 45, n. 8, p. 2962 – 2979, 2012. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320312000684>.

CHUNPRUGSUK, C. et al. Inspection weapon's image on social network using haartraining. In: *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*. Pattaya, Thailand: ECTI, 2017. p. 330–335.

CINAR, Y. G.; ZOGHBI, S.; MOENS, M. Inferring user interests on social media from text and images. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. Atlantic City, NJ, USA: IEE, 2015. p. 1342–1347. ISSN 2375-9259.

CROSS, S.; HARRISON, R.; KENNEDY, R. Introduction to neural networks. *The Lancet*, v. 346, n. 8982, p. 1075 – 1079, 1995. ISSN 0140-6736.

DIAS, P. A.; TABB, A.; MEDEIROS, H. Apple flower detection using deep convolutional networks. *Computers in Industry*, v. 99, p. 17 – 28, 2018. ISSN 0166-3615. Disponível em: <http://www.sciencedirect.com/science/article/pii/S016636151730502X>.

DIEZ, P. Chapter 1 - introduction. In: DIEZ, P. (Ed.). *Smart Wheelchairs and Brain-Computer Interfaces*. Academic Press, 2018. p. 1 – 21. ISBN 978-0-12-812892-3. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780128128923000017>.

DOZAT, T. Incorporating nesterov momentum into adam. In: STANFORD. *International Conference on Learning Representations*. San Diego, CA, USA, 2016. p. 1–6.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, v. 12, n. 7, p. 2121–2159, 2011.

EDGAR, T. W.; MANZ, D. O. Chapter 6 - machine learning. In: EDGAR, T. W.; MANZ, D. O. (Ed.). *Research Methods for Cyber Security*. Cambridge, MA: Syngress, 2017. p. 153 – 173. ISBN 978-0-12-805349-2.

GAUTAM, S. *Clothes Classification*. 2017. `https://www.kaggle.com/salil007/caavo/metadata`.

GREENFIELD, M. Social media trends. 2019. Disponível em: <https://inteligencia.rockcontent.com/social-media-trends-2019-panorama-das-empresas-e-dos-usuarios-nas-redes-sociais/>.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994. ISBN 0023527617.

HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015.

HECHT-NIELSEN. Theory of the backpropagation neural network. In: *International 1989 Joint Conference on Neural Networks*. Washington, D.C: IEEE, 1989. p. 593–605 vol.1.

HECHT-NIELSEN, R. Theory of the backpropagation neural network. *International 1989 Joint Conference on Neural Networks*, DC, USA, p. 593–605 vol.1, 1988.

HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 22, n. 1, p. 5–53, 2004.

HOSSIN, M.; M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, v. 5, p. 01–11, 03 2015.

HU, Y.; MANIKONDA, L.; KAMBHAMPATI, S. What we instagram: A first analysis of instagram photo content and user types. In: *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*. Ann Arbor, United States: The AAAI Press, 2014. p. 595–598. ISBN 9781577356578.

HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, v. 195 1, p. 215–43, 1968.

KEYES, D. *Instagram rolls out shoppable posts for more merchants*. 2017. `https://www.businessinsider.com/instagram-rolls-out-shoppable-posts-for-more-merchants-2017-10`.

KHAYRULLIN, E. *Image Classification: car, motorbike, bicycle*. 2012. `https://www.kaggle.com/c/image-classification2/data`.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

KO, B.; KIM, H.; CHOI, H. Controlled dropout: A different dropout for improving training speed on deep neural network. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Bari, Italy: IEEE, 2017. p. 972–977.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105. Disponível em: <http://dl.acm.org/citation.cfm?id=2999134.2999257>.

KUNCORO, B. A.; ISWANTO, B. H. Tf-idf method in ranking keywords of instagram users' image captions. In: IEEE. *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*. Cambridge, United States, 2015. p. 1–5.

LAZZEZ, O.; OUARDA, W.; ALIMI, A. M. *DeepVisInterests: CNN-Ontology Prediction of Users Interests from Social Images*. 2018.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

LENG, B.; YU, K.; QIN, J. Data augmentation for unbalanced face recognition training sets. *Neurocomputing*, v. 235, p. 10 – 14, 2017. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231216314886>.

LEWENBERG, Y.; BACHRACH, Y.; VOLKOVA, S. Using emotions to predict user interest areas in online social networks. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Paris, France: IEEE, 2015. p. 1–10.

LI, F.-F.; ANDREETTO, M.; RANZATO, M. A. Caltech101 image dataset. 2003. Disponível em: <http://www.vision.caltech.edu/Image$_D atasets/Caltech$101/>.

LI, R. et al. Improving deep convolutional neural networks for real-world clothing image. In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. Guilin, China: IEEE, 2017. p. 837–843.

LING, C. X. Overfitting and generalization in learning discrete patterns. *Neurocomputing*, v. 8, n. 3, p. 341 – 347, 1995. ISSN 0925-2312. Optimization and Combinatorics, Part I-III.

LIU, N. et al. Exploiting convolutional neural networks with deeply local description for remote sensing image classification. *IEEE Access*, v. 6, p. 11215–11228, 2018. ISSN 2169-3536.

LOURIDAS, P.; EBERT, C. Machine learning. *IEEE Software*, v. 33, n. 5, p. 110–115, Sep. 2016.

MAGLOGIANNIS, I. et al. Emerging artificial intelligence applications in computer engineering. In: *Emerging Artificial Intelligence Applications in Computer Engineering*. Athens, Greece: Engineering Applications of Artificial Intelligence, 2007.

MAKSIMOV, A. *Docker container with Jupyter, Matplotlib, Pandas, Tensorflow, Keras and OpenCV*. 2019. `https://hub.docker.com/r/amaksimov/python_data_science/`.

MARTÍN, A. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <http://tensorflow.org/>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, v. 52, p. 99–115, 1988.

MERLER, M.; CAO, L.; SMITH, J. You are what you tweetpic! gender prediction based on semantic analysis of social media images. IBM TJ Watson, Yorktown Heights, NY, v. 2015, 08 2015.

MOLENAAR, C. *Why Customers Would Rather Have a Smartphone Than a Car: Relationship Retailing as an Opportunity*. [S.l.]: Routledge, 2015. ISBN 9781315547237.

NESTEROV, Y. A method for solving the convex programming problem with convergence in a hilbert space. In: . Island, EUA: AMS, 1983.

NWANKPA, C. et al. Activation functions: Comparison of trends in practice and research for deep learning. *ArXiv*, abs/1811.03378, 2018.

PATHAK, M. P. A. R.; RAUTARAY, S. Application of deep learning for object detection. *Procedia Computer Science*, Elsevier, v. 132, p. 1706 – 1717, 2018.

PIMKOTE, P.; KANGKACHIT, T. Classification of alcohol brand logos using convolutional neural networks. *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*, p. 135–138, 2018.

PINHEIRO, C. A. DE P.; NEDJAH, N.; MOURELLE, L. DE M. Detection and classification of pulmonary nodules using deep learning and swarm intelligence. Multimedia Tools and Applications, Mar 2019. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-019-7473-z>.

POLYAK, B. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, v. 4, p. 1–17, 12 1964.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, v. 29, p. 1–98, 06 2017.

REDDI, S.; KALE, S.; KUMAR, S. On the convergence of adam and beyond. In: *International Conference on Learning Representations*. New York, NY, USA: eprint arXiv:1904.09237, 2018.

ROBBINS, H.; MONRO, S. A stochastic approximation method. *Ann. Math. Statist.*, The Institute of Mathematical Statistics, v. 22, n. 3, p. 400–407, 09 1951.

ROLIM, V. et al. Um estudo sobre sistemas de recomendação de recursos educacionais. In: . Recife, PE: CBIE, 2017. p. 724.

ROSEBROCK, A. *Multi-label classification with Keras*. 2018. `https://www.pyimagesearch.com/2018/05/07/multi-label-classification-with-keras/`.

ROSENBLATT, F. F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958.

ROSER, H. R. M.; ORTIZ-OSPINA, E. Internet. *Our World in Data*, 2019. Https://ourworldindata.org/internet.

ROSER, M. Internet access. *Our World in Data*, 2019. Https://ourworldindata.org/internet.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015.

SCHWENKER, F.; TRENTIN, E. Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 37, p. 4–14, fev. 2014. ISSN 0167-8655.

SEGALIN, C. et al. The pictures we like are our image: Continuous mapping of favorite pictures into self-assessed and attributed personality traits. *IEEE Transactions on Affective Computing*, v. 8, n. 2, p. 268–285, April 2017. ISSN 1949-3045.

SERMANET, P. *Competition distinguish dogs from cats*. 2014. `https://www.kaggle.com/c/dogs-vs-cats/overview`.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations*. Chennai, India: IEEE, 2015.

Singh, S.; Singh, N. Containers docker: Emerging roles future of cloud technology. In: *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. Pune, India: IEEE, 2016. p. 804–807.

UHRIG, R. E. Introduction to artificial neural networks. In: *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*. Orlando, USA: IEEE, 1995. v. 1, p. 33–37 vol.1.

VIEIRA, M. G.; MOREIRA, J. Classification of e-commerce-related images using hierarchical classification with deep neural networks. In: *2017 Workshop of Computer Vision (WVC)*. Natal, Brazil: IEEE, 2017. p. 114–119.

YOU, Q. et al. The eyes of the beholder: Gender prediction using images posted in online social networks. In: *2014 IEEE International Conference on Data Mining Workshop*. Rochester, NY: IEEE, 2014. p. 1026–1030. ISSN 2375-9232.

ZEILER, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, p. 01–06, 12 2012.

ZHENG, J.; WANG, Z. J.; JI, X. Superpixel-based image recognition for food images. In: *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. London, Ontario: IEEE, 2016. p. 1–4.

APÊNDICE A – Resultados dos experimentos

Este apêndice apresenta os valores obtidos nos experimentos de treinamento e teste dos modelos classificadores.

A.1 Obtenção de hiper-parâmetros

Tabela 5: Resultado dos experimentos variando os hiper-parâmetros

Experimento	**Taxa Aprendizado**	**Épocas**	**Erro**	**Acurácia**	σ_E	σ_A
E1	0,001	10	0,58	0,75	0,02	0,01
E2	0,001	20	0,55	0,79	0,02	0,01
E3	0,001	50	0,46	0,81	0,06	0,03
E4	0,001	100	0,49	0,82	0,05	0,02
E5	0,001	200	0,48	0,79	0,04	0,01
E6	0,01	10	0,36	0,49	0,05	0,04
E7	0,01	20	0,22	0,75	0,03	0,02
E8	0,01	50	0,42	0,83	0,05	0,02
E9	0,01	100	0,49	0,82	0,04	0,03
E10	0,01	200	0,51	0,78	0,02	0,03
E11	0,1	10	0,53	0,80	0,05	0,02
E12	0,1	20	0,22	0,83	0,02	0,04
E13	0,1	50	0,34	0,87	0,02	0,04
E14	**0,1**	**100**	**0,11**	**0,92**	**0,01**	**0,04**
E15	0,1	200	0,23	0,89	0,02	0,04

A.2 Métricas de desempenho

Tabela 6: Métricas de desempenho do SGD

Métrica	Média	Desvio	Máximo	Mínimo
Erro Treino	0,34	0,23	1,00	0,01
Erro Teste	0,14	0,17	1,00	0,02
Acurácia Treino	0,95	0,01	0,97	0,91
Acurácia Teste	0,88	0,04	0,93	0,72
F-Média	0,80	0,10	0,82	0,32
Sensibilidade	0,80	0,10	0,81	0,30
Precisão	0,81	0,11	0,83	0,31
Especificidade	0,88	0,08	0,89	0,45

Tabela 7: Métricas de desempenho do AdaGrad

Métrica	Média	Desvio	Máximo	Mínimo
Erro Treino	0,08	0,16	1,00	0,005
Erro Teste	0,12	0,16	0,96	0,004
Acurácia Treino	0,86	0,05	0,92	0,69
Acurácia Teste	0,84	0,06	0,91	0,67
F-Média	0,81	0,16	0,82	0,17
Sensibilidade	0,83	0,14	0,84	0,18
Precisão	0,82	0,17	0,85	0,17
Especificidade	0,87	0,08	0,89	0,62

Tabela 8: Métricas de desempenho do RMSprop

Métrica	Média	Desvio	Máximo	Mínimo
Erro Treino	0,19	0,19	1,00	0,003
Erro Teste	0,13	0,15	0,70	0,004
Acurácia Treino	0,94	0,04	0,98	0,79
Acurácia Teste	0,88	0,04	0,93	0,72
F-Média	0,80	0,10	0,83	0,32
Sensibilidade	0,80	0,10	0,82	0,30
Precisão	0,81	0,10	0,83	0,30
Especificidade	0,88	0,07	0,87	0,39

Tabela 9: Métricas de desempenho do Adam

Métrica	**Média**	**Desvio**	**Máximo**	**Mínimo**
Erro Treino	0,18	0,17	1,00	0,002
Erro Teste	0,11	0,19	1,00	0,004
Acurácia Treino	0,93	0,04	0,98	0,79
Acurácia Teste	0,90	0,03	0,93	0,77
F-Média	0,81	0,09	0,83	0,43
Sensibilidade	0,82	0,09	0,82	0,43
Precisão	0,82	0,09	0,84	0,43
Especificidade	0,87	0,02	0,89	0,88

Tabela 10: Métricas de desempenho do Adamax

Métrica	**Média**	**Desvio**	**Máximo**	**Mínimo**
Erro Treino	0,11	0,12	1,00	0,005
Erro Teste	0,12	0,12	1,00	0,003
Acurácia Treino	0,93	0,02	0,96	0,83
Acurácia Teste	0,88	0,03	0,93	0,77
F-Média	0,82	0,09	0,84	0,43
Sensibilidade	0,80	0,08	0,82	0,42
Precisão	0,81	0,09	0,83	0,43
Especificidade	0,86	0,05	0,89	0,66

Tabela 11: Comparativo do desempenho dos algoritmos

Algoritmo	**Acc. Treino**	**Erro Treino**	**Acc. Teste**	**Erro Teste**	**Conf.**
SGD	0,95	0,34	0,88	0,14	267
AdaGrad	0,86	0,08	0,84	0,12	300
RMSProp	0,93	0,19	0,88	0,13	325
Adam	**0,94**	**0,18**	**0,90**	**0,10**	**331**
Adamax	0,93	0,11	0,88	0,12	271