

Estudo Comparativo de Arquiteturas de Redes Neurais em Análise de Sentimentos

Thiago S. Farias, Rafael G. Rossi

¹Universidade Federal do Mato Grosso do Sul (UFMS)
UNID. II: Av. Ranulpho Marques Leal, 3484 – CEP 79620-080 - Cx Postal n° 210
Mato Grosso do Sul – MS – Brazil

tsffarias@gmail.com, rafael.g.rossi@ufms.br

Abstract. *Sentiment Analysis is the process in which the main goal is to extract the polarities of the sentiments expressed in opinions toward a topic of interest. This research field keeps gaining increasing attention in the web and the academic world, since institutions, people and companies are interested to know the real opinion of a group of people regarding a specific topic. The consequence of this interest is the development of applications for opinion monitoring of products and brands, and the repercussion of important events on social media like presidential elections and sports events. Currently, neural networks have been gaining notoriety in general Text Mining tasks and in Sentiment Analysis; and in the face of challenges, researchers use several neural network architectures in order to obtain better results. As there is no established consensus on the best architecture for the Sentiment Analysis tasks, it is up to the researcher to choose between the approaches. In this context, the objective of this paper is to present a comparative performance analysis on the Sentiment Analysis of the following neural network architectures: Long-Short Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), and Bi-directional Long Short-Term Memory (Bi-LSTM). The goal is to indicate to the reader which one has the greatest performance when performing the task of extracting the sentiment polarity expressed in textual data: a neural network that extracts global or sequential patterns. The approach applied in this comparative analysis is based on supervised machine learning and has no limitations regarding its domain as well as its language.*

Resumo. *A Análise de Sentimentos é um processo que tem por objetivo principal extrair as polaridades dos sentimentos expressos nas opiniões em relação a um tópico de interesse. Essa área de pesquisa vem ganhando atenção, tanto na Web quanto na academia, pois instituições, pessoas e companhias se interessam em saber a opinião real de um grupo de pessoas a respeito de um tema específico. A consequência deste interesse é o desenvolvimento de aplicações de monitoramento de opiniões de produtos e marcas, além da análise da repercussão de eventos importantes em redes sociais como eleições presidenciais e eventos esportivos. Atualmente, as redes neurais vêm ganhando notoriedade em tarefas e Mineração de Textos em geral e na Análise de Sentimentos, e frente aos desafios, pesquisadores utilizam diversas arquiteturas de redes neurais presentes na literatura com objetivo de obter melhores resultados. Como não há um consenso estabelecido sobre qual seria a melhor arquitetura*

para a classificação de sentimentos, fica a cargo do pesquisador escolher entre os métodos. Neste contexto, este trabalho de conclusão de curso visa realizar uma análise comparativa do desempenho de arquiteturas de redes neurais em Análise de Sentimentos das seguintes arquiteturas: Rede neural recorrente LSTM (Long Short-Term Memory), Rede neural convolucional (CNN), Rede neural recorrente Gated Recurrent Unit (GRU), e Rede neural recorrente Bi-LSTM (Bi-Directional Long-Short Term Memory). O objetivo é indicar ao leitor, dentre tais abordagens, qual tem maior performance ao realizar a tarefa de extrair a polaridade dos sentimentos expressos em bases textuais: uma rede neural que extrai padrões globais ou sequenciais. A abordagem empregada nesta análise está baseada na aprendizagem de máquina supervisionada, e não tem limitação em relação ao seu domínio, nem tão pouco à língua.

1. Introdução

Em uma sociedade interconectada como a atual, a influência opinativa social, tanto de amigos, familiares, pessoas desconhecidas, ou de especialistas, transmitidas a todo momento pelos veículos de comunicação, tem um importante papel na formação da opinião, autocrítica, crenças, comportamentos e até mesmo na tomada de decisão de cada indivíduo [18, 24]. A procura por opiniões de outras pessoas é um hábito comum e devido o advento da *web 2.0* [32], onde o usuário se tornou também um gerador de conteúdo, as opiniões ali proferidas são amplamente propagadas por meio de redes sociais, *blogs* e *sites* [38]. A aprovação destas influências opinativas pode servir de estímulo e motivação para uma melhora de desempenho em uma tarefa específica. Porém, o contrário também é verdadeiro. Quando percebemos que a opinião social baseada em informações negativas nos desaprova, a reação mais comumente observada é um certo recuo de ideias e convicções diante à negatividade imposta [24]. Essa influência, seja ela positiva ou negativa, pode ainda ter efeito em diversas outras instâncias, como no processo de tomada de decisão da compra de um produto [28], escolha de voto em um político [3] e investimento no mercado financeiro [15]. Diante destes dois polos da influência social, ignorá-los seria um erro tanto em sentido pessoal, perdendo assim o seu poder estimulativo e motivacional característicos, como também profissional, perdendo informações essenciais providas pelos consumidores.

Como consequência dos fatos mencionados acima, o volume de informação gerada para determinadas personalidades, companhias, produtos ou serviços é muitas vezes impossível de ser analisada manualmente e em tempo hábil para uma tomada de decisão. As redes sociais, atualmente, são um destes ambientes geradores de grandes volumes de dados. Elas permitem, após extração e análise, compreender as opiniões e sentimentos sobre as mais diferentes entidades e até mesmo prever fenômenos sociais complexos [14]. Esta potencialidade vem despertando o interesse e assim o desenvolvimento de técnicas e ferramentas que visam a análise e extração de conhecimento de tais textos de maneira automatizada [7, 19]. Porém, a realização da extração de conhecimentos contém muitos desafios, pois em uma mensagem de texto, há muitas nuances que podem induzir os resultados destas técnicas e ferramentas a resultados errôneos. Estes desafios se devem à complexidade da linguagem humana, além do uso de gírias, sarcasmo, ironia, erros ortográficos e dialetos.

Neste contexto de extração de conhecimento, são comumente utilizadas técnicas

de Mineração de Textos (MT), a qual realiza a extração de padrões de textos não estruturados e escritos em linguagem natural [1]. Surge, então, a área de Análise de Sentimentos (AS), também conhecida como Mineração de Opinião, a qual é a área de estudo responsável por analisar opiniões, sentimentos, emoções e atitudes expressas por pessoas sobre entidades como produtos, serviços, eventos, organizações e seus atributos [19].

Dada a definição acima, basicamente as técnicas de AS visam a identificação de sentimentos que usuários expressam a respeito de alguma entidade de interesse (seja ela um produto, empresa, lugar, pessoa ou evento) em textos compartilhados na *web*, sem que seja necessário a busca e leitura manual de todas as opiniões e notícias. Segundo Bing Liu [19], *startups* e empresas de pequeno, médio e grande porte, estão utilizando aplicações de análise de sentimentos em ampla escala em muitas áreas como varejo, hotelaria, turismo, saúde, eventos esportivos e até mesmo eleições presidenciais. Dentre estas empresas, se encontram *Google, Microsoft, Amazon, eBay, Oracle e Adobe* [19], mostrando assim o interesse e a ascensão nos últimos anos desta área, como é apresentado na Figura 1.

Figura 1. Resultado de pesquisa do termo “Sentiment Analysis”



Fonte: Google Trends, 2021

Muitos dos esforços destas grandes empresas neste ramo do conhecimento, é empregado em solucionar desafios que são próprios da linguagem humana para se analisar sentimentos. Na literatura, encontram-se diversas abordagens para analisar sentimentos em bases textuais, como a abordagem baseada em léxico e as baseadas em aprendizado de máquina [38]. Na abordagem léxica, as palavras individuais possuem uma polaridade prévia, a qual é uma orientação semântica independente de contexto e que pode ser expressada com um valor numérico ou classe. O léxico é constituído por um dicionário de termos, onde cada termo está associado a um sentimento específico. Após o método receber uma sentença de entrada, a classificação é realizada por meio da co-ocorrência entre alvo e sentimento com uma pesquisa no dicionário léxico de cada termo que forma a mensagem, ao qual não leva em consideração a ordem dos termos dentro de um documento nem suas relações léxico-sintáticas. Nessa abordagem, cada documento da coleção de dados é representado por um vetor de palavras, e a polaridade do documento é baseada na polaridade mais frequente presente nos termos. Assim, esta abordagem tem por limitação, além do fato da dependência de um léxico, a não análise por meio do contexto, o que pode resultar em uma classificação de polaridade errônea ao ser aplicada em um domínio sensível ao contexto [22].

Assim como em outros ramos do conhecimento, algoritmos de Aprendizado de Máquina (AM) têm sido utilizados para automatizar tarefas envolvendo a análise de textos, e, conseqüentemente, podendo ser empregados na Análise de Sentimentos. Estes

algoritmos são capazes de aprender a generalizar ou extrair padrões com base em experiências acumuladas (exemplos) por meio de solução bem sucedida de problemas anteriores, além de ser uma abordagem que evita a limitação de dependência de um léxico [38]. As categorias de aprendizado de máquina envolvem: (i) aprendizado supervisionado, onde faz uso de exemplos previamente rotulados por um “professor”, com objetivo de induzir um modelo de classificação para classificar novos objetos; (ii) aprendizado semissupervisionado, que realiza a tarefa de classificação através de uma base de treinamento rotulada e uma base não-rotulada; e (iii) aprendizado não supervisionado, onde são fornecidos implicitamente padrões de entrada para o sistema, ou seja, não é preciso a existência de um “professor” que forneça exemplos para a aprendizagem, pois o próprio sistema reconhece e avalia os conceitos implícitos [26].

Em Redes Neurais Artificiais (RNAs), ao qual é uma abordagem da área de Aprendizado de Máquina, busca-se resolver problemas complexos como o da Análise de Sentimentos, por simular o comportamento de redes neurais biológicas. As RNAs são compostas por nós interconectados (que atuam como neurônios) com pesos e arestas (que por sua vez atuam como sinapses) para realizar o processamento de dados. Por meio de avaliações experimentais, a abordagem de rede neural vem apresentando melhor performance quando comparada ao léxico, mesmo sendo aplicada em domínios sensíveis ao contexto [22]. Atualmente, as redes neurais vêm ganhando notoriedade, e frente aos desafios, pesquisadores utilizam diversas arquiteturas de redes neurais presentes na literatura com objetivo de minimizar os problemas em Mineração de Opiniões e Análise de Sentimentos. Como não há um consenso estabelecido sobre qual seria a melhor arquitetura para a classificação de sentimentos, fica a cargo do pesquisador escolher entre os métodos.

Neste contexto, este trabalho se propõe a realizar uma análise comparativa da performance de arquiteturas de redes neurais em aprendizado de máquina supervisionada que têm se destacado na classificação de textos, e assim, por meio de experimentos empíricos, indicar ao leitor, dentre tais abordagens, qual tem maior desempenho ao realizar a tarefa de extrair a polaridade dos sentimentos expressos em bases textuais; uma rede neural que extrai padrões globais ou sequenciais. As arquiteturas a serem comparadas são: Rede Neural Recorrente do tipo LSTM (*Long Short-Term Memory*), Rede Neural Convolutiva (CNN), Rede Neural Recorrente do tipo *Gated Recurrent Unit* (GRU), Rede Neural Recorrente do tipo Bi-LSTM (*Bidirectional Long Short-Term Memory*) [2]. As arquiteturas aqui analisadas, têm como característica não ter limitação em relação ao seu domínio e linguagem, podendo assim serem aplicadas em textos de quaisquer áreas de conhecimento e idioma desejados.

Para melhor apresentar esta área em ascensão, juntamente com a elaboração e a análise comparativa das arquiteturas, este trabalho está dividido em 5 seções. Na Seção 2 será apresentada a fundamentação teórica necessária para o entendimento deste trabalho de conclusão de curso juntamente com alguns trabalhos relacionados e de grande relevância. Na Seção 3 será detalhada a abordagem utilizada para realização da análise comparativa de cada arquitetura de rede neural profunda para Análise de Sentimentos supervisionada, com todas as suas etapas de elaboração, técnicas, limitações e estratégias. Na Seção 4 serão apresentados os detalhes e resultados da avaliação experimental, considerando as arquiteturas de redes neurais profundas utilizadas nesse trabalho para 2 bases de Análise de Sentimentos. Por fim, na Seção 5 serão apresentadas as considerações fi-

nais juntamente com as contribuições deste projeto, assim como algumas sugestões para trabalhos futuros.

2. Fundamentação Teórica e Trabalhos Relacionados

Para a melhor compreensão das diretrizes propostas neste trabalho, nesta seção são apresentados os principais conceitos envolvidos e relacionados a este trabalho de conclusão de curso como Mineração de Textos, Aprendizado de Máquina, Análise de Sentimentos, Redes Neurais Artificiais (RNAs), RNAs recorrentes, RNAs convolucionais, juntamente com a revisão bibliográfica referente ao tema deste trabalho.

2.1. Mineração de Textos

Devido ao avanço da informatização, a sociedade está cada vez mais interconectada e por consequência produzindo grandes volumes de dados [21]. Esses dados, muitas vezes, contêm informações valiosas como tendências e padrões que poderiam ser utilizadas para ajudar nas tomadas de decisões de empresas públicas ou privadas.

Segundo [36], 80% das informações corporativas estão armazenadas em documentos textuais. Chen [5] diz ainda que 80% do conteúdo *on-line* estão em formato textual. Este volume de dados gerados diariamente, é tão grande que dificulta ou impossibilita a análise manual e em tempo hábil para uma tomada de decisão.

Neste cenário, a mineração de textos surge da necessidade de se extrair, de forma automática, padrões ou conhecimentos úteis a partir de dados em formato texto [7]. O processo de mineração de textos pode ser dividido em várias etapas, com suas diversas especificidades de acordo com o objetivo em questão. Na Figura 2, é apresentado um modelo de processo de mineração de textos dividido em 5 etapas proposto por Aranha e Passos [6].

Figura 2. Processo de Mineração de Textos



Fonte: Aranha and Passos, 2009 [6]

A primeira etapa, "coleta", tem como objetivo coletar informações de uma base textual, a qual será utilizada pelos demais passos de processo e impactará diretamente nos resultados obtidos. Na etapa de pré-processamento, são aplicadas técnicas para estruturar e padronizar os documentos em um formato apropriado para que possam ser submetidos aos algoritmos de extração de conhecimento. Já na etapa de indexação, os documentos são identificados e indexados para possibilitar consultas posteriores, facilitando o seu acesso e recuperação [7]. Na etapa de mineração, são utilizadas técnicas para identificar padrões e extrair conhecimento destas bases textuais. Finalmente, na última etapa, o objetivo é avaliar e interpretar os resultados, e descobrir se o algoritmo de mineração de textos atingiu as expectativas segundo alguma métrica tais como taxa de erro e tempo.

2.2. Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA), que estuda como os algoritmos computacionais são capazes de automaticamente executarem tarefas através da experiência acumulada de soluções anteriores. Estes algoritmos são utilizados na etapa de mineração, e se baseiam em várias áreas do conhecimento, como estatística e otimização, para aprender padrões complexos a partir de algum *corpus* [23]. Os principais tipos de aprendizado de máquina são:

- **Aprendizado Supervisionado:** este método faz uso de exemplos previamente rotulados, fornecidos por um “professor”, com intuito de induzir um modelo de classificação para classificar novos objetos reproduzindo o comportamento do professor. Alguns exemplos de aplicações que fazem uso desse tipo de aprendizado são: reconhecimento facial, previsão do tempo, categorização de textos [26].
- **Aprendizado Semisupervisionado:** este método realiza a tarefa de classificação através de um conjunto de treino rotulado e um conjunto não-rotulados [30]. As mesmas aplicações do aprendizado supervisionado podem ser utilizadas com este tipo de aprendizado.
- **Aprendizado Não Supervisionado:** neste método são fornecidos exemplos sem rótulos, ou seja, não é preciso a existência de um “professor” que forneça as categorias dos exemplos para a aprendizagem, pois o próprio sistema reconhece e avalia os conceitos implícitos no conjunto de dados, como grupos de objetos semelhantes e associações entre os atributos. [26].

Em relação ao presente trabalho, foi utilizado o método de aprendizagem supervisionada no desenvolvimento dos modelos das arquiteturas de redes neurais profundas em Análise de Sentimento para a realização da análise comparativa.

2.3. Análise de Sentimentos

A Análise de Sentimentos é a área de estudo que analisa as opiniões expressas sobre uma determinada entidade (seja ela um produto, empresa, lugar, pessoa ou evento) em textos escritos em linguagem natural compartilhados na *web*, e computacionalmente extrair e classificar se a opinião transmitida possui polaridade positiva, negativa ou neutra [19, 37].

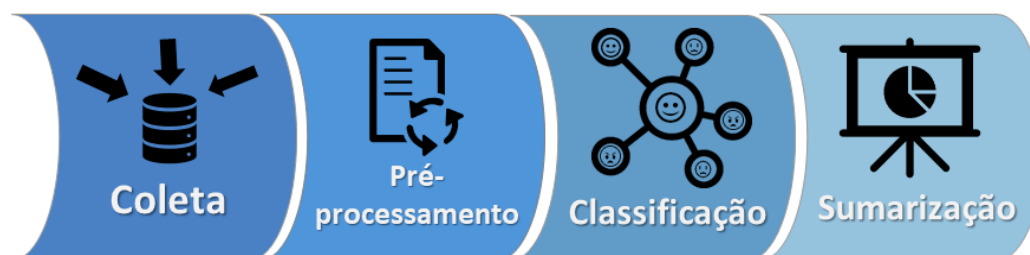
Esta classificação é possível através de técnicas de Aprendizado de Máquina, análise léxica e Mineração de Dados, ao qual visam detectar e extrair padrões para construção de um modelo ou função que descreva diferentes classes de dados. Segundo [38], a Análise de Sentimento pode ser realizada em três níveis de granularidade:

- **A nível de documento:** este nível tem como objetivo classificar um documento como positivo, negativo ou neutro, observando o sentimento global expresso no texto.
- **A nível de sentença:** este nível objetiva classificar a polaridade de cada sentença de um determinado documento.
- **A nível de aspecto:** visa analisar e identificar textos e sentenças que contenham determinada característica que se deseja buscar, como um componente de um produto ou serviço, e então o classifica com a polaridade positiva, negativa ou neutra.

Da mesma forma que na Mineração de Textos, o processo de Análise de Sentimentos também pode ser dividido em etapas. Na Figura 3 é apresentado um modelo de processo

global de Análise de Sentimentos dividido em 4 etapas, baseado no modelo proposto por Contratres e Souza [8]. Esta figura será utilizada como um modelo didático ao descrever cada etapa da abordagem proposta deste presente trabalho, pois este modelo de processo foi utilizado no desenvolvimento da abordagem proposta deste trabalho de conclusão de curso.

Figura 3. Processo de Análise de Sentimentos.



Fonte: Contratres e Souza, 2018 [8]

Na primeira etapa da Análise de Sentimento se realiza a extração e coleta de dados, onde os dados são obtidos das fontes de interesse e armazenados em uma base de dados. Após a criação desta base de dados, vêm a etapa de pré-processamento, na qual são realizados os tratamentos dos dados. Alguns dos tratamentos de dados mais aplicados em coleções de dados textuais são: lematização, padronização de caixas, tratamento de caracteres repetidos em textos de redes sociais, eliminação de identificadores de usuários e remoção de caracteres não úteis para a discriminação do texto. Assim, este tratamento consiste na padronização e limpeza dos dados, objetivando eliminar redundâncias e informações desnecessárias para facilitar a procura do conhecimento que se deseja extrair. Após esta etapa, será gerada uma representação de dados estruturada da coleção de documentos, seja para documentos focados em sentenças ou aspectos, de acordo com os métodos de extração de padrões que serão aplicados. A seguir, na etapa de classificação, é definida qual tipo de Aprendizagem de Máquina melhor se aplica ao problema em questão, como AM supervisionado ou não supervisionado, para que então seja construído um modelo de classificação capaz de determinar se um dado textual pertence à classe positiva, negativa ou neutra. Por fim, na etapa de sumarização é realizada uma ordenação, categorização e sumarização dos resultados, de forma que os resultados sobre as respectivas polaridades possam ser melhor apresentados, possibilitando uma melhor interpretação das informações e conhecimentos obtidos com a etapa de classificação [38].

2.4. Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é uma estrutura lógico-matemática que busca simular a forma, comportamento e funções de redes neurais biológicas. Em geral, uma RNA é composta por nós que são denominados neurônios, que são distribuídos em camadas interconectadas com pesos por conexões inspiradas nas relações formadas por axônios e dendritos, ao qual são componentes essenciais de células neurais biológicas [2].

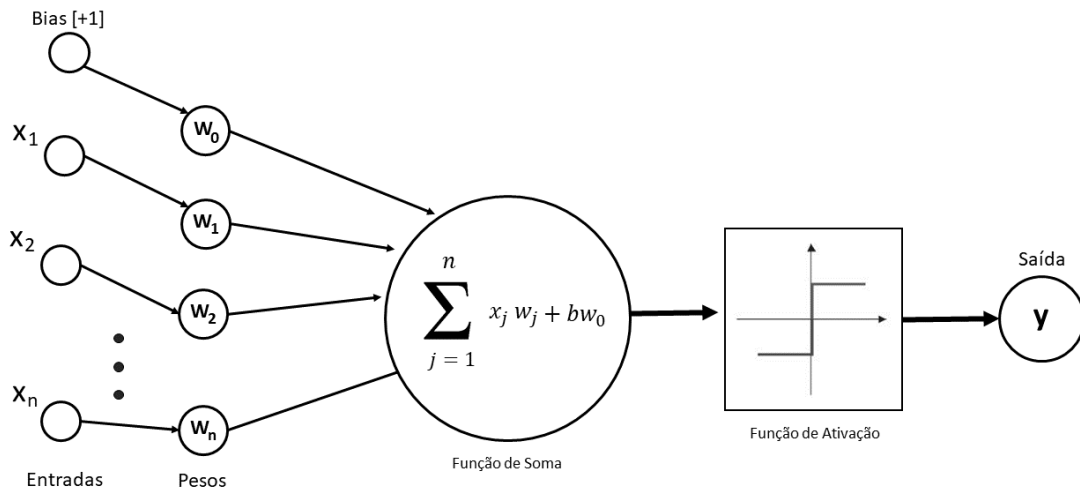
As RNAs possuem capacidade de aprendizado e de generalização a partir da experiência análoga aos seres humanos, podendo assim realizar classificações e reconhecimento de padrões. A ideia geral dessa aprendizagem é a seguinte: ao fornecer dados classificados, será calculado a diferença entre a saída gerada pela RNA e a saída real

(dado classificado), e então será ajustado os pesos (conexões entre os neurônios) da rede baseado nessa diferença. Isso será repetido até que a diferença calculada seja pequena o suficiente [2]. Este processo é dividido em duas etapas diferentes chamadas *feedforward* e *backpropagation*, ao qual será explanada ainda nesta seção. Para o entendimento das RNAs utilizadas neste trabalho, vamos apresentar inicialmente um neurônio tipicamente utilizado em uma rede neural para depois explicar as demais arquiteturas que serão apresentadas subsequentemente.

2.5. Perceptron

O Perceptron é a arquitetura mais simples de redes neurais usada para a classificação e reconhecimento de padrões. Foi introduzida em 1958 por Frank Rosenblatt, ao qual se inspirou nos trabalhos de Walter Pitts e Warren Sturgis McCulloch. Esta arquitetura corresponde à um único neurônio, conforme ilustrado na Figura 4.

Figura 4. A arquitetura básica do Perceptron



Fonte: Autoria Própria, 2021

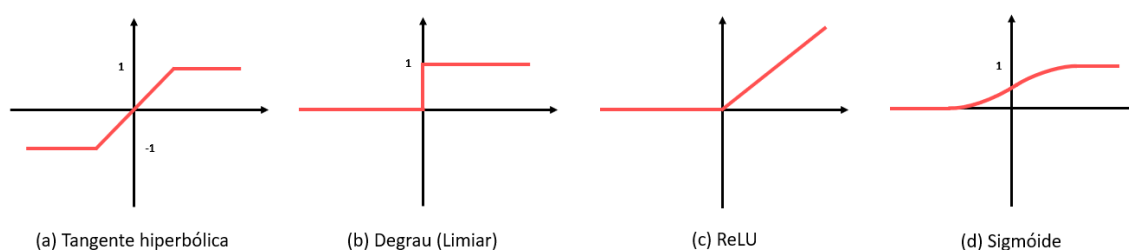
Cada entrada x_j está conectada à um peso w_j . Há também uma entrada adicional denominada *bias* (b) que sempre recebe como entrada o valor 1. O valor w_0 é o peso que pondera o *bias*. *Bias*, ou viés, funciona como uma entrada sempre ativa para cada uma das camadas da rede. As estradas e os seus respectivos pesos são submetidos à função soma que realiza a soma das entradas ponderadas pelos pesos, conforme apresentado na Equação 1 [1].

$$\sum_{j=1}^n x_j w_j + b w_0 \quad (1)$$

Logo após o somatório ponderado, a próxima tarefa do neurônio é a de verificar se o valor resultante da função soma atingiu ou não um valor predeterminado, chamado de limiar. Se o resultado atingiu o limiar, então um valor será transmitido adiante pela saída (y). Caso contrário, outro valor será transmitido [2]. O objetivo é limitar a amplitude de saída do neurônio, ou seja, o valor resultante da função soma é normalizado

dentro de um intervalo fechado, por exemplo $[-1,1]$, como é apresentado na Figura 5. Esse processo de verificação é comumente chamado de função de ativação, ou função de transferência. Existem diversas funções de ativação atualmente, como as funções lineares degrau e identidade, e as não lineares como a gaussiana, tangente hiperbólica, ReLU e sigmóide [2]. Em relação as funções lineares, a função mais popular para servir como função de ativação no perceptron é a função degrau. Entre as funções não lineares, a técnica logística sigmoideal (ou apenas sigmóide), e a ReLU estão entre as mais utilizadas.

Figura 5. Exemplo de comportamento de algumas funções de ativação



Fonte: Autoria Própria, 2021

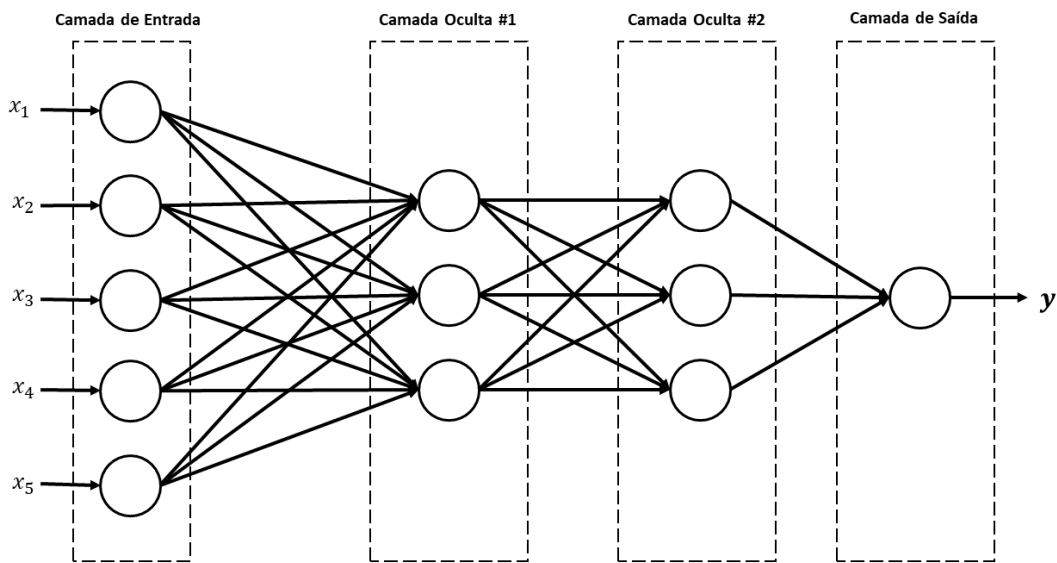
O uso de um único *perceptron* é capaz de criar uma única superfície de decisão entre exemplos de diferentes classes, limitando, portanto, o seu desempenho em situações em que exemplos de diferentes classes necessitam de mais de uma superfície de decisão (ex: a função XOR). Para solucionar problemas como XOR por exemplo, seria necessário o desenvolvimento de uma rede neural mais robusta [2]. Com isso em mente, foi proposta a rede neural artificial perceptron de múltiplas camadas.

2.6. Perceptron de Múltiplas Camadas

Com o intuito de lidar com problemas não linearmente separáveis, foram adicionadas camadas de neurônios ocultas no modelo de Rosenblatt, formando assim a rede neural artificial Perceptron de Múltiplas Camadas (PMC ou *MLP* — *Multi Layer Perceptron*) [1]. A arquitetura de uma rede MLP consiste no arranjo topológico dos neurônios com os respectivos valores de pesos associados às conexões. O treinamento da rede se resume no ajustamento dos pesos sinápticos para fazer com que a resposta real da rede se aproxime da resposta desejada [2]. A MLP vem sendo aplicada com sucesso em diversos problemas complexos, através do seu treinamento supervisionado usando o algoritmo de retro-propagação de erro (*error backpropagation*) baseado na regra de aprendizagem por correção do erro. A Figura 6 apresenta um exemplo de uma rede perceptron de múltiplas camadas, ao qual possui uma camada de entrada com cinco neurônios, duas camadas ocultas (intermediárias) com três neurônios cada, e uma camada de saída com um neurônio. Vale ressaltar que quando uma rede possui mais de uma camada oculta, esta é denominada rede neural profunda (*DNN* - *Deep Neural Network*) e o aprendizado em tal rede é chamado de aprendizado profundo (*deep learning*).

Em uma rede perceptron de múltiplas camadas, a camada de entrada é composta por neurônios sensoriais, ou seja, unidades que não modificam os sinais externos, apenas os distribuem para a primeira camada oculta. As camadas ocultas (*hidden layers*), que

Figura 6. Arquitetura MLP com Duas Camadas Ocultas



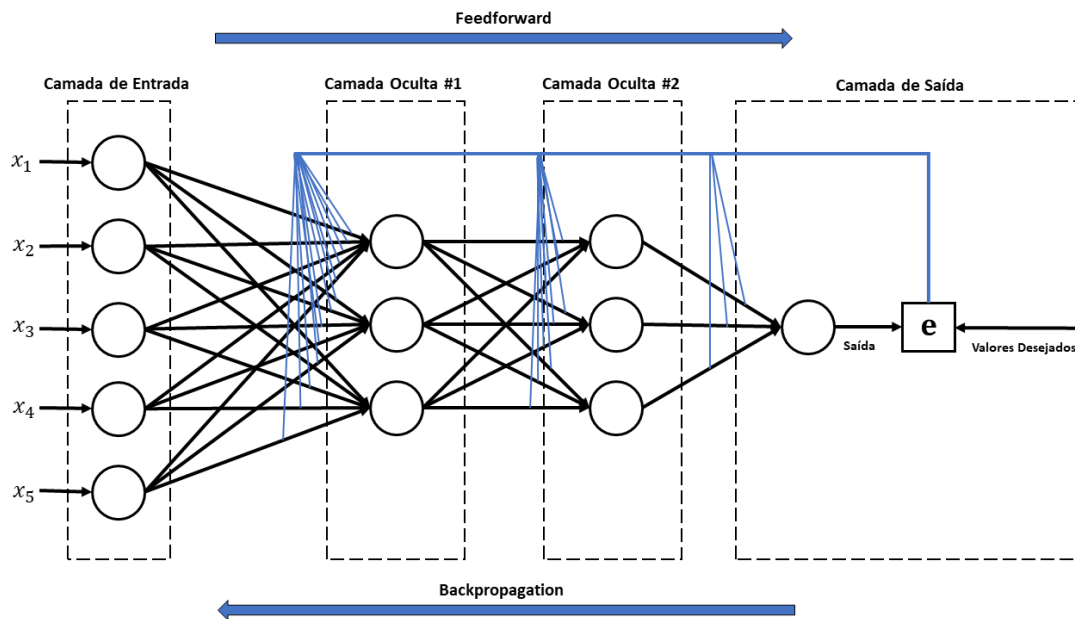
Fonte: Autoria Própria, 2021

também podem ser chamadas de camadas intermediárias ou escondidas, são camadas que se situam entre a camada de entrada e a camada de saída da rede neural. Assim como nas camadas externas (entrada e saída), as camadas ocultas, além de possuírem seus respectivos pesos, também possuem neurônios artificiais ao qual os sinais são transferidos de um neurônio para outro obedecendo as funções de ativação (transferência) que cada neurônio possui [1].

A introdução do algoritmo de retro-propagação por David Rumelhart em 1986, possibilitou o treinamento de RNAs com diversas camadas [4]. Basicamente o processo ocorre em duas etapas. Na primeira etapa, ao qual é chamada de *feedforward*, os pesos sinápticos da rede recebem valores aleatórios e são propagados em uma única direção dentro das camadas internas até que o sinal seja reproduzido na camada de saída. Na segunda etapa, ocorre efetivamente o aprendizado da rede. Utilizando a saída desejada/esperada e a saída fornecida pela camada de saída da rede, é calculada a diferença entre os dois valores e, caso o resultado não esteja no padrão aceitável, a rede calcula o erro e propaga a correção para as demais camadas internas até a entrada, ajustando os seus pesos sinápticos por meio da técnica retro-propagação de erro (*error backpropagation*). A técnica *backpropagation* propaga os sinais de erro na direção oposta ao *feedforward*, como ilustra a Figura 7. Este processo é repetido por um número determinado de vezes para cada padrão de dados durante o treinamento, até que o erro total chegue a um valor mínimo, ou até que um limite pré-determinado de iterações tenha sido atingido.

O número de neurônios em uma RNA exerce uma grande influência no aprendizado da rede. Com isso em mente, é importante antes de desenvolver a estrutura da rede neural, conhecer as características dos dados com que se está trabalhando. Quando uma rede neural profunda possui um alto número de neurônios na camada oculta, aumenta-se a capacidade de mapeamento não-linear da rede, o que pode ocasionar um *overfitting* (sobre ajustamento dos dados), diminuindo sua capacidade de extrair características gerais (generalização). Por outro lado, uma RNA com poucos neurônios na camada oculta, pode

Figura 7. Arquitetura MLP e sua Retro-propagação de Erro



Fonte: Autoria Própria, 2021

não ser capaz de realizar o mapeamento desejado, o que é denominado de *underfitting*. A definição do número ideal de neurônios presentes nas camadas ocultas é definida de forma empírica, e normalmente provém da distribuição dos padrões de treinamento e validação da rede.

A rede Perceptron de Múltiplas Camadas tem a limitação de apenas tratar os dados de entrada de maneira independente, o que faz com que não seja possível o tratamento de padrões sequenciais ou locais. Esta limitação fez surgir diversas propostas de redes neurais profundas para os mais variados problemas, ao qual algumas serão abordadas a seguir.

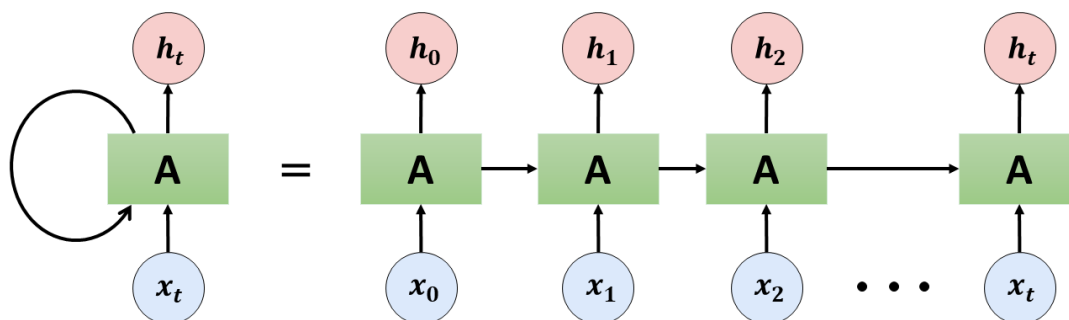
2.7. Rede Neural Recorrente

As Redes Neurais Recorrentes (RNR ou *RNN* — *Recurrent Neural Networks*) são uma família de redes neurais profundas voltadas para o processamento de dados sequenciais [13]. Da mesma forma como um ser humano, ao ler um texto, a rede RNN busca entender cada palavra baseada no entendimento das palavras anteriores ou posteriores (contexto). Uma rede neural recorrente objetiva manter um dado de uma entrada anterior para que possa servir para o entendimento do dado atual. Este é um dos grandes motivos destas arquiteturas receberem tanta adesão no campo de previsões de séries temporais e em tarefas envolvendo Processamento de Linguagem Natural (NLP) [2].

Uma RNN pode ser representada pelo diagrama ilustrado na Figura 8. O *loop* no diagrama representa a memória interna da rede neural. A função do *loop*, é permitir que as informações trafegadas nele persistam, e sejam passadas de uma etapa da rede para a próxima. Elas fornecem a rede um *feedback* constante a respeito do estado das entradas processadas previamente. Essas conexões recorrentes permitem que a RNN persista em memória os estados destas entradas anteriores, influenciando no que é obtido como saída. Além disso, como é ilustrado na Figura 7, em uma rede neural recorrente, cada camada

(célula de memória) fornece informações para a camada sucessora, caracterizando assim a dependência da sequência para realizar o aprendizado [2].

Figura 8. Ilustração de Rede Neural Recorrente (RNN)



Fonte: Autoria Própria, 2021

Este princípio presente na RNN é aplicado em uma variedade de problemas atualmente. Porém, as RNN são muito limitadas quando se deparam com a necessidade de processar longas sequências de informação. Em sua forma padrão, elas são incapazes de persistir informações contextuais de longo prazo por conta de um problema chamado *Vanishing gradient problem* [2]. Isso se dá pelo fato que quando n camadas ocultas usam uma ativação como a função *sigmoid*, n pequenas derivadas são multiplicadas juntas. Dessa forma, os gradientes da função de perda se aproximam de zero, tornando a rede difícil de treinar, pois os pesos e vieses (*bias*) das camadas iniciais não serão atualizados efetivamente. Isso pode levar a uma imprecisão geral de toda rede, pois as camadas iniciais são cruciais para reconhecer os elementos centrais dos dados de entrada, e no pior cenário, isso até pode interromper completamente o treinamento de uma rede neural. Este problema é o principal causador do “esquecimento” das informações mais antigas em uma RNN.

Com este problema em mente, diversas variantes da rede neural recorrente vêm sendo propostas para lidar com perda de memória, como é o caso das redes *Long Short-Term Memory* (memória de curto e longo prazo), comumente chamadas de “LSTMs”, e também das redes *Gated Recurrent Unit*, comumente chamadas de “GRUs”.

2.8. Rede Neural Recorrente LSTM (Long Short-Term Memory)

As redes *Long Short-Term Memory* (memória de curto e longo prazo), ou apenas LSTM, são um tipo especial de RNN proposto por *Hochreiter & Schmidhuber* (1997) para resolver o tratamento de sequências longas por uma rede recorrente [31]. Esta arquitetura é bastante utilizada em diversas áreas na aprendizagem profunda como previsão de séries temporais, reconhecimento de escrita, reconhecimento de fala, composição musical, tradução de idiomas [9].

A ideia básica da rede LSTM é a introdução das representações de estado (ou neurônios de memória) que preservam os gradientes com o decorrer do tempo. Assim como em um RNN padrão, esta arquitetura é composta por um conjunto de sub-redes conectadas recorrentemente chamada bloco de memória, onde cada bloco contém células de

memória com auto conexões que permitem armazenar o estado temporal da rede. Além das células de memória, cada bloco também possui estruturas chamadas portas (*gates*), responsáveis por controlar o fluxo de informações. Cada porta possui pesos e um *bias* associado, sendo estas grandezas responsáveis por controlar cada uma das portas. Portas de uma célula de memória LSTM permitem o armazenamento e o acesso de informação por um longo período de tempo. Esta arquitetura conseguiu contornar o problema apresentado na seção anterior, pois quando a porta *forget gate* (explicação a seguir) está ativo e assim próxima de 1, o gradiente não irá dissipar, e devido a impossibilidade da porta *forget gate* ser maior que 1, o gradiente não irá explodir, resolvendo assim o problema do *vanishing gradient problem* [31].

Cada bloco de memória da arquitetura LSTM contém uma ou mais células de memória conectadas com três portas (*gates*): porta de entrada (*input gate*), porta de esquecimento (*forget gate*) e porta de saída (*output gate*). Todas as portas possuem uma função que as permite escrever, ler e redefinir operações dentro do bloco de memória, além também de utilizar uma função *sigmoide* (σ) para achatando os valores dos vetores no intervalo fechado entre 0 (porta fechada) e 1 (porta aberta) [31].

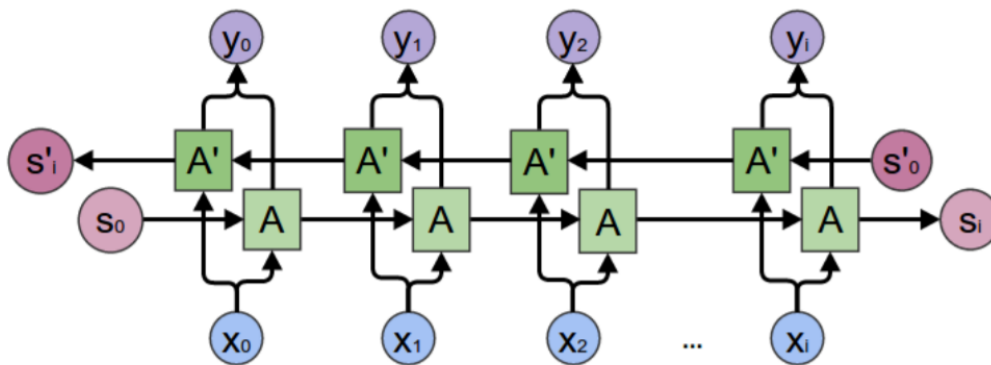
A porta de esquecimento (*forget gate*) determina quais valores do vetor de saída da célula anterior serão aproveitadas na memória, através da função σ , que retorna valores entre 0 e 1. Assim, caso tenha sido detectado características úteis, a porta será aberta e as informações poderão ser carregadas para os próximos intervalos de tempo, caso contrário, a informação contida na célula de memória anterior será redefinida. A porta de entrada (*input gate*) é responsável por definir o quanto do novo estado calculado para a entrada será aproveitado, ou seja, ele realiza a adição de informações úteis ao estado da célula. A porta de saída (*output gate*) realiza a tarefa de extrair informações úteis do estado interno da célula atual para serem expostas para o resto da rede [27]. Nesta arquitetura há também uma camada tangente hiperbólica, ao qual cria um vetor de novos valores, que serviram como candidatos para serem adicionados no novo estado da célula de memória.

Porém, a rede neural recorrente LSTM possui a limitação de apenas ser capaz de fazer uso da informação de contexto anterior. No entanto, a tarefa de modelagem sequencial, permite o acesso a informações de entrada do passado e do futuro por um determinado tempo. Com isso em mente, foi então proposto uma variante da rede LSTM para lidar com esta limitação, nomeada rede neural recorrente *Bidirectional LSTM* (Bi-LSTM) [31].

2.9. Rede Neural Recorrente Bi-LSTM (Bidirectional Long Short-Term Memory)

A rede neural *Bidirectional LSTM* (Bi-LSTM) foi proposto por Schuster e Paliwal (1997), e sua arquitetura é formada por duas camadas de rede LSTM, onde uma camada lida com os estados passados, e a outra com os estados futuros [2]. Como apresentado na Figura 9, as palavras de uma sentença são as entradas X, aplicadas para duas redes recorrentes simultâneas A e A', onde uma analisa a sentença em ordem de leitura, sentido S, e a outra com a ordem das palavras lidas de trás para frente, sentido S'. A saída da rede Y, é a concatenação dos resultados das duas direções. Dessa forma, esta bidirecionalidade dos dados faz com que este tipo de rede possa compreender melhor o contexto [11].

Figura 9. Arquitetura da Rede Bi-LSTM



Fonte: Guarise [11], 2019

2.10. Rede Neural Recorrente Gated Recurrent Unit (GRU)

A rede neural recorrente *Gated Recurrent Unit* (GRU) foi proposta por Kyunghyun Cho em 2014, e pode ser considerada uma arquitetura relativamente nova, especialmente quando comparada a arquitetura amplamente adotada LSTM, ao qual foi introduzida em 1997. A motivação por trás dos esforços da criação da arquitetura GRU, além da melhor eficiência e redução da complexidade em relação a arquitetura LSTM, era permitir a captura de dependências de forma adaptativa de grandes sequências de dados sem descartar informações de partes anteriores da sequência [2].

Estes objetivos foram alcançados por meio de suas unidades de portas (*gates*), semelhantes às dos LSTMs, que controlam o fluxo de informação dentro de uma unidade, e que resolvem o *vanishing gradient problem* das RNNs tradicionais.

A célula da arquitetura GRU possui apenas duas portas (*gates*): porta de atualização (*update gate*) e a porta de redefinição (*reset gate*) [2]. Por meio destas portas (*gates*) e cálculos que o estado oculto e os dados de entrada são submetidos, o estado oculto da GRU se torna capaz de manter ao mesmo tempo as dependências de curto e longo prazo. E assim como as portas das LSTMs, as portas da rede GRU também são treinadas para efetivamente selecionar informações úteis, e descartar informações irrelevantes. De maneira básica, as portas são vetores que contém valores em um intervalo de 0 e 1, ao qual serão multiplicados com os dados de entrada e/ou estado oculto. Caso houver um valor 0 nos vetores, isso mostra que as informações no estado de entrada ou oculto não são importantes, e então retornarão como zero. No entanto, quando houver um valor 1 no vetor, isso indica que as informações são úteis e serão utilizadas. Quando toda a rede GRU é treinada através do método *backpropagation*, os pesos são ajustados de forma que o vetor aprenda a reter apenas os recursos úteis.

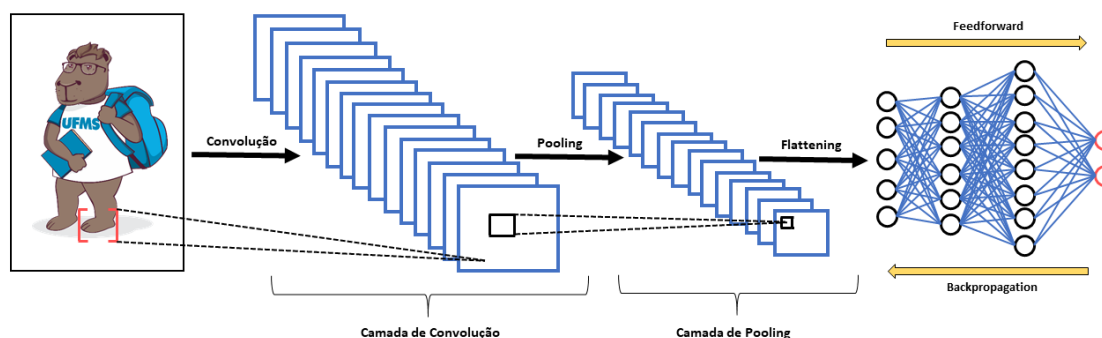
2.11. Rede Neural Convolutiva (CNN)

As Redes Neurais Convolucionais (*CNN – Convolutional Neural Network*), também conhecidas como ConvNets, formam uma classe de redes neurais artificiais do tipo *feed-forward* muito utilizada na área de visão computacional para classificação de imagens e detecção de objetos. Este nome se dá pelos filtros de convolução utilizados por este tipo de técnica [12]. A inspiração desta arquitetura, ao qual é uma variação da rede MLP, veio

por meio dos estudos realizados em 1962 por Hubel e Wiesel sobre o funcionamento do córtex visual dos gatos [2].

Na Figura 10 é apresentada uma ilustração de arquitetura geral de uma CNN. Uma CNN possui diversos tipos de camadas, onde as camadas mais básicas são: camadas de convolução, camadas de agrupamento (*pooling*) e camadas totalmente conectadas (*fully connected*) [2]. Basicamente, uma convolução consiste na aplicação de uma série de “filtros” em toda a extensão da matriz de entrada para uma posterior aplicação de uma rede neural densa (*multilayer perceptron*). Para isso, na arquitetura CNN, há diversas sequencias de camadas de convolução ao qual é composta por diversos neurônios, onde cada neurônio é responsável por aplicar um filtro em uma determinada região da matriz de entrada; onde o resultado da aplicação destes filtros é comumente conhecido como mapa de características *feature maps*. Estes mapas indicam onde estão localizadas as características importantes extraídas pelos filtros. De modo geral, pode-se considerar os parâmetros presentes nas camadas convolucionais como um conjunto de n filtros, com pesos previamente inicializados que serão ajustados através do método *backpropagation* conforme a rede aprende a identificar regiões significantes com intuito de minimizar a função custo.

Figura 10. Arquitetura Geral de uma Rede Neural Convolutiva (CNN)



Fonte: Autoria Própria, 2021

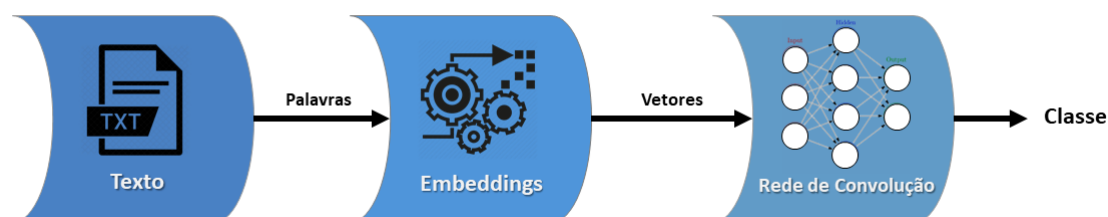
A próxima camada em uma rede CNN é a camada *pooling*, ao qual realizam um processo de *down-sampling* [2]. Esta técnica tem o objetivo de diminuir a dimensionalidade das matrizes resultantes da convolução, diminuindo assim a quantidade de parâmetros treinados pela rede. Existem dois tipos de operações de *pooling* que são amplamente utilizadas: *max-pooling* e *mean-pooling*. Ambas seguem o mesmo princípio, se diferenciando apenas na forma com que é calculado o valor final. A operação de *max-pooling* retira o maior valor de uma determinada região da matriz, de acordo com o tamanho do *pool* aplicado. Esta operação parte da análise do maior valor de uma região, pois os valores máximos no mapa de características (*feature map*) representam onde estão localizadas as características, e ao aplicar a camada de *pooling* uma porcentagem de informação que não é característica é desprezada (*down-sampling*). Desta forma, a matriz resultante desta operação é o mapa de características agrupadas (*pooled feature map*). Em relação a operação *mean-pooling*, também chamada de *average-pooling*, tem a característica de realizar o cálculo médio de uma determinada região limitada pelo filtro. Após o processo de

pooling, é realizada a operação *flattening*. Esta operação realiza a conversão das matrizes de características resultantes da camada de *pooling* em um vetor coluna, o qual servirá de entrada para uma RNA clássica.

As camadas totalmente conectadas (*FCL - Fully Connected Layer*), onde normalmente são usadas como camadas finais, e diferentemente das camadas convolucionais, são camadas onde todos os pesos da camada são conectados com a camada anterior [20]. Nesta camada, toda uma rede neural densa ou perceptrons em múltiplas camadas (Seção 2.6) é adicionada ao final da rede CNN, onde cada valor presente no vetor coluna criado pela operação de *flattening* se torna uma entrada para a RNA, para em seguida serem ligadas as camadas ocultas totalmente conectadas até a camada de saída. Na camada de saída (*output layer*) ocorre efetivamente o processo de classificação. Devido a sua simplicidade e bons resultados, é comumente utilizada a função de ativação *SoftMax* para prever a classe do objeto de entrada. A função *SoftMax* transforma a saída da rede neural em probabilidades de o exemplo pertencer a cada uma das classes definidas. Então, o erro é calculado e retro-propagado, de forma que alguns parâmetros como os pesos e os detectores de características (filtros) são ajustados como uma RNA clássica.

Embora originalmente a arquitetura CNN tenha sido construída para processamento de imagens, hoje ela também é utilizada efetivamente em áreas de classificação de textos como Análise de Sentimentos, categorização de tópicos e filtro *spam* [13]. Na Figura 11 é apresentada o diagrama de funcionamento de uma rede neural convolucional em classificação de textos.

Figura 11. Classificação Textual Usando Redes Neurais Convolucionais



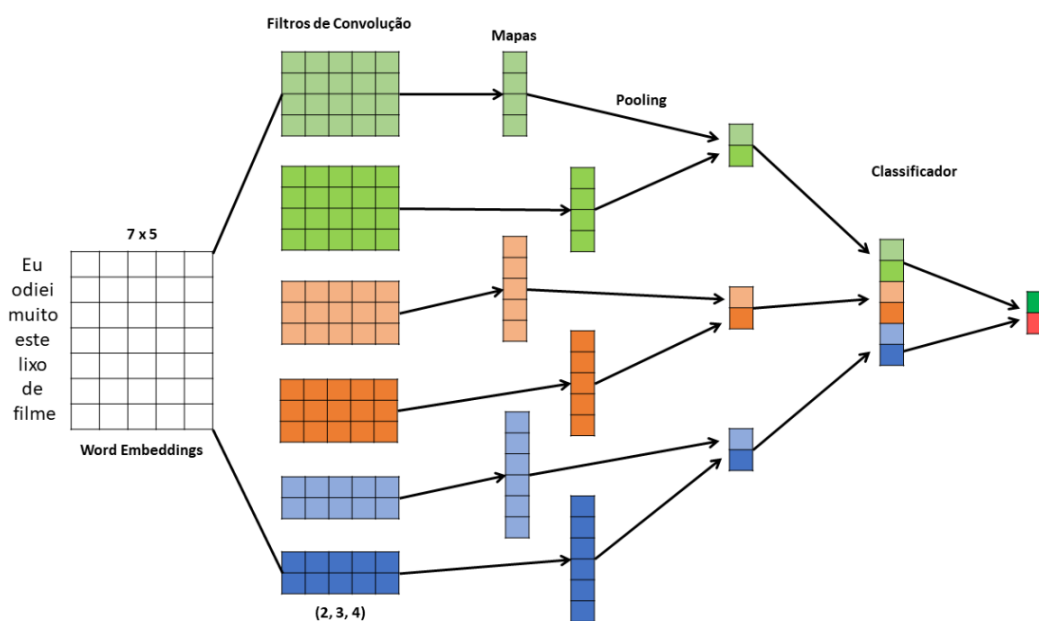
Fonte: Autoria Própria, 2021

Cada dado de entrada passa então pelo bloco *Embeddings*, ao qual realiza a transformação de cada palavra em um vetor com dimensões fixas, de forma que palavras semanticamente relacionadas possuam representações vetoriais semelhantes, pois as redes neurais possuem a necessidade de trabalhar com representações numéricas de cada palavra. Os vetores resultantes servirão então de entrada para rede de convolução, e a rede assim classificará o texto segundo as classes do problema que estão sendo abordados [2]. Vale ressaltar que os filtros convolucionais da CNN aplicada a imagens são 2D, enquanto que em textos são 1D.

Diferentemente de como a rede convolucional realiza o processamento de imagens, em que cada posição de uma matriz representa um pixel da imagem, no caso das tarefas de Processamento de Linguagem Natural (NLP), as entradas são frases ou documentos representados como uma matriz, onde cada linha corresponde a uma palavra ou

caractere. Estes vetores, ao qual são o resultado da passagem pelo bloco *Embeddings* como mostra a Figura 9, podem ser representações de baixa dimensionalidade produzidos por algoritmos como *word2vec* e *GloVe*, ou vetores de alta dimensionalidade como *one-hot* [13]. No caso de uma frase contendo sete palavras usando uma representação de 100 dimensões, a matriz dessa forma se torna uma matriz 7 x 100 ao qual servirá como entrada para a rede. Em relação a textos, estes terão a mesma largura dos vetores de entrada, porém diferentes alturas.

Figura 12. Arquitetura de Rede Neural Convolutacional para Classificação Textual



Fonte: Autoria Própria, 2021

Na Figura 12, é apresentado um exemplo de arquitetura de rede neural convolutacional sendo aplicada a um texto. A sentença utilizada (“Eu odiei muito este lixo de filme”) contém 7 palavras e foi representada por vetores de 5 dimensões. Foram usados três tamanhos de região de filtro: 2, 3 e 4, cada um com 2 filtros onde cada filtro realiza convolução na matriz de sentença, e gera 6 *feature maps* (mapas de características) de comprimento variável. Os mapas resultantes tem tamanho 4x1, 5x1 e 6x1, e passam pelo processo de *pooling* (max-pooling) e depois são concatenados formando um vetor de tamanho 6x1. Por fim, a camada final de *Softmax* então recebe este vetor de características como entrada e o usa para classificar a sentença, obtendo duas classes como saída [20].

2.12. Trabalhos Relacionados

Nesta seção, são apresentados alguns dos trabalhos correlatos relevantes para a análise comparativa de redes neurais profundas deste presente trabalho envolvendo *deep learning* (Aprendizagem Profunda). Devido a crescente quantidade de trabalhos publicados envolvendo o tema Análise de Sentimento em textos, se fez necessário a seleção de trabalhos mais recentes ao qual consideram as arquiteturas utilizadas neste trabalho, para fazer uma apresentação das abordagens existentes do estado da arte, e que apresentam características diretamente relacionadas com este trabalho.

Devido a rápida popularização do *e-commerce*, o número de opiniões em relação aos produtos e serviços também vem crescendo. O número de *reviews* (avaliações) de clientes para produtos chega a ser centenas ou até milhares, dificultando assim a leitura e tomada de decisão por parte do cliente de comprar ou não aquele produto, assim como por parte do vendedor de conhecer a real opinião dos clientes a respeito daquele produto. Deste modo, o trabalho de Bing Liu e Minqing Hu [16] estuda o problema de sumarização baseado em características (*feature-based summaries - FBS*) de *reviews* de clientes. O trabalho visa propor um modelo FBS para minerar e sumarizar todos os *reviews* de clientes para cada característica (*features*) específica de determinado produto, além de informar a polaridade da opinião expressada. A tarefa é realizada em três passos: (1) utilizar mineração de dados e NLP (*Natural Language Processing*) e regras de associação (*Association Mining*) com CBA (*Classification Based on Associations*) para coletar características (*features*) frequentes, atributos e funções expressadas em opiniões de clientes, (2) identificar sentenças que expressam opiniões e classificar cada sentença de um *review* por sua polaridade (positivo ou negativo), (3) sumarizar os resultados. Estes passos são proferidos em múltiplos subpassos. Em geral, adjetivos compartilham a mesma orientação semântica dos seus sinônimos, e orientação semântica oposta dos seus antônimos. No trabalho realizado por Bing Liu e Minqing Hu [16], é usado esta premissa para determinar a orientação semântica de um adjetivo presente na sentença, além de utilizar da orientação semântica de palavras opinativas predominante nas sentenças. Foi comparado as características (*features*) de produtos gerados pelo método apresentado com os termos gerados pelo método FASTR (*FAst Syntactic Term Recognizer*). Os resultados do experimento mostram que o método FASTR tem menor precisão para a tarefa de extração de características (*features*) de produtos em relação ao método proposto pelo trabalho. Em relação a classificação de orientação semântica, o método proposto mostra uma boa acurácia de 84%.

No trabalho de Sindhwani [33], é proposta uma abordagem de aprendizado semissupervisionado transdutivo para a Análise de Sentimentos em documentos por meio de redes bipartidas utilizando um método de classificação denominado *Semi-supervised Lexical Regularized Least Squares*, e um dicionário léxico desenvolvido pela *IBM India Research Labs* para classificar documentos como positivo, negativo ou neutro. É apresentado um *framework* unificado em que informações léxicas, dados não rotulados e dados de treinamento já rotulados são combinados de forma eficiente para melhorar a acurácia do método. Também são apresentados os resultados da abordagem em três diferentes domínios, *blogs* de discussões de produtos de *softwares* empresariais, *blogs* de discussões da eleição presidencial americana, e *reviews* de filmes *online*. O Algoritmo proposto foi comparado com outras abordagens comumente aplicadas ao problema em questão. Os algoritmos comparados foram: classificação lexical não supervisionada (LEX), máquina de vetor de suporte linear (*Linear SVM*), e duas implementações de máquina de suporte transdutivo (*Transductive SVM*), uma baseada em *label switching* (LS) e outra baseada em *deterministic annealing* (DA). Os experimentos empíricos nos três datasets, mostram que o algoritmo proposto superou todos os concorrentes em todos os conjuntos de dados, obtendo acurácia de 85%.

Em Xia Hu [17], é discutido o quão complexo e importante são os sinais emocionais (*Emotional Signals*) expressos em palavras ou *posts*, por exemplo *emoticons* e avaliações de produtos através de estrelas, para se fazer uma análise de sentimentos em

base textuais eficiente. Neste trabalho é apresentado a proposta de um algoritmo de aprendizado não supervisionado para Análise de Sentimentos em dados textuais baseado em léxico, onde é incorporado duas categorias de sinais emocionais, indicação de emoção e correlação de emoção em um *framework* denominado *ESSA (Emotional Signals for unsupervised Sentiment Analysis)*, com intuito de realizar a classificação da polaridade de *posts* extraídos do *Twitter*. O *framework* proposto foi comparado com os seguintes métodos de análise de sentimentos não supervisionados: GI-Label, MPQA-Label, LBM, *k*-Means, ONMTF, MoodLens e CSMF. Dois *datasets* foram utilizados nos experimentos deste trabalho, o *dataset Stanford Twitter Sentiment (STS)* e o *Obama-McCain Debate (OMD)*, onde o primeiro contém 40.216 *tweets* com suas polaridades, e o segundo contém 3.269 *tweets* (juntamente com suas polaridades) postados durante o debate presidencial americano entre Barack Obama e John McCain. Os resultados dos experimentos conduzidos com o *framework* ESSA mostram que o *framework* proposto superou os concorrentes em ambas as coleções de dados. No *dataset* STS, o *framework* ESSA obteve 72% de acurácia, e no *dataset* OMD 69%.

Em Goldberg [10], é apresentado a proposta de um algoritmo de aprendizado de máquina semissupervisionado baseado em representação em redes, utilizando método transdutivo e medida de similaridade *PSP (Positive Sentence Percentage)*, para realizar a tarefa de Análise de Sentimentos em base textuais de avaliações (*Reviews*) dadas por usuários. A partir de um grupo de documentos, por exemplo avaliações textuais de filmes feitos por críticos de cinemas, o algoritmo infere uma avaliação numérica a partir destes documentos não estruturados para demonstrar que utilizando avaliações não rotuladas é possível melhorar a performance da realização da tarefa de inferência de avaliação (*Rating inference*). Os resultados dos experimentos indicam, que o algoritmo semissupervisionado baseado na similaridade *PSP (SSL+PSP)* demonstrou uma melhor performance quando comparado aos demais métodos mencionados no estudo em todos os quatro autores quando há apenas 200, 100, 50, 25, ou 12 documentos rotulados disponíveis. Em 19 dos 20 cenários de aprendizagem, a acurácia do conjunto não rotulado pelo algoritmo *SSL + PSP* foi significativamente alta em relação aos demais métodos. Vale ressaltar que o método *SSL* não foi tão bem com grandes conjuntos já rotulados.

No trabalho de Guilherme Martins [22], foi realizado uma análise comparativa do desempenho de dois métodos para Análise de Sentimentos em notícias: léxico e redes neurais. Foi realizado a elaboração e expansão a partir da conjugação de verbos de um léxico de sentimentos para língua portuguesa, juntamente com o treinamento e avaliação de uma rede neural recorrente *LSTM (Long Short-Term Memory)* na classificação de notícias de categoria: nacional, internacional e econômica. Os experimentos mostraram que o léxico apresentou melhores resultados em situações onde a distinção entre as polaridades positivo/negativo eram mais explícitas, ou seja, baseadas em termos isolados. Porém o método falha em situações que necessitam de um maior contexto. Já a rede neural apresentou de maneira geral melhores resultados que o léxico, conseguindo também ser mais consistente em suas previsões, mesmo na categoria econômica, ao qual é um domínio sensível ao contexto em que as palavras estão inseridas.

Em Hassan [13], é proposto um modelo de aprendizagem profunda para classificação de textos que se baseia na Rede Neural Convolutiva (CNN) e Rede Neural Recorrente Bidirecional (BRNN). A proposta combina as camadas convolucionais e re-

correntes em um único método, para capturar de forma eficiente as dependências de longo prazo no documento para realizar tarefas de classificação, e também para explorar uma camada recorrente bidirecional que serve como um substituto para a camada de *pooling* na rede convolucional, para hipoteticamente reduzir a perda de detalhes nas informações locais e captura de dependências de longo prazo. O modelo proposto foi avaliado por meio de experimentos realizados em dois conjuntos de dados de Análise de Sentimentos, *Stanford Large Movie Review* (IMDB) e *Stanford Sentiment Treebank* (SSTb). O modelo de arquitetura CNN-BRNN proposto, comparado com outras abordagens como *Recursive Neural Tensor Network* (RNTN), *Multi-View Recurrent Neural Network* (MV-RNN), *CNN-multichannel*, SVMs e *Naïve Bayes*, apresentou melhores resultados em ambos os conjuntos de dados de Análise de Sentimento, chegando a obter 93.4% de performance.

No trabalho de Trofimovich [35], é realizado um estudo comparativo de três soluções de aprendizagem supervisionada para a tarefa de Análise de Sentimentos em mensagens postadas na rede social *Twitter*, sobre entidades como bancos e empresas de telecomunicações. As duas primeiras soluções são baseadas em *Gated Recurrent Unit* (GRU) e na Rede Neural Convolucional (CNN), respectivamente. A última solução é uma abordagem *ensemble* ao qual consiste em três classificadores. Ele utiliza uma máquina de vetores de suporte (SVM) com adaptação de domínio, juntamente com RNN e CNN. Foram usados dois conjuntos de dados neste trabalho, um para treinamento ao qual contém cerca de 9.000 *tweets* que mencionam tanto bancos como empresas de comunicações, e o outro para teste que contém cerca de 19.000 mensagens. Os resultados de avaliação mostram que a solução baseada na rede neural de tipo GRU obteve a melhor performance *F1-score* em ambas as coleções de dados (bancos e empresas de telecomunicações).

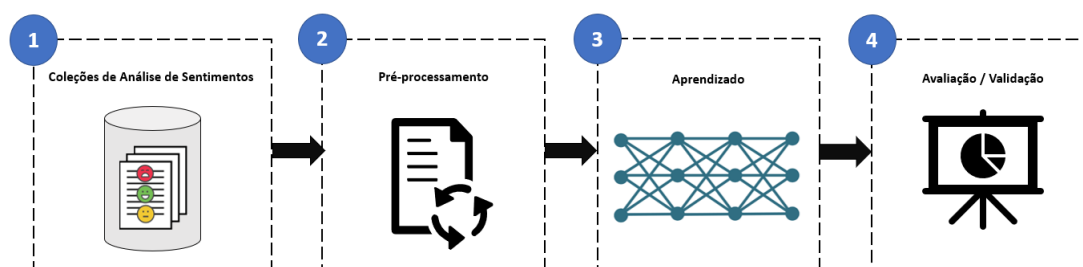
Em Nowak [27], é realizada uma análise comparativa entre três arquiteturas de redes neurais profundas para a tarefa de classificação de sentimentos em textos curtos. As redes neurais analisadas são: Rede Neural Recorrente LSTM (*Long Short-Term Memory*), Rede Neural *Gated Recurrent Unit* (GRU) e Rede Neural Recorrente Bi-LSTM (*Bidirectional Long-Short Term Memory*). Para se realizar a avaliação de cada arquitetura, foram realizados experimentos utilizando três coleções de dados: *Spambase Data Set* (conjunto de mensagens rotuladas como SPAM), *Farm Advertisement* (conjunto de textos de anúncios relacionados a animais de fazenda) e *Amazon book reviews* (conjunto de dados de resenhas de livros para 8 livros diferentes). Os resultados dos experimentos mostram que, as arquiteturas LSTM e Bi-LSTM foram as arquiteturas que obterão melhor acurácia utilizando as coleções de dados selecionadas. Observa-se também nos resultados que, a arquitetura GRU não é uma boa escolha para classificação em textos curtos, como títulos de resenha de livros, porém se sai melhor em casos simples onde se objetiva obter resultados rápidos, como nos dois primeiros *datasets*.

Com relação aos trabalhos relacionados, este trabalho se difere ao realizar um estudo comparativo do desempenho entre quatro arquiteturas de redes neurais aplicadas na tarefa de Análise de Sentimentos. O objetivo é indicar ao leitor, dentre tais abordagens, qual tem maior desempenho ao realizar a tarefa de extrair a polaridade dos sentimentos expressos em bases textuais, uma rede neural que extrai padrões globais ou sequenciais. As arquiteturas aqui analisadas têm a característica de não possuir limitação em relação ao seu domínio e linguagem, podendo assim serem aplicadas em textos de quaisquer áreas de conhecimento e idioma.

3. Método de Pesquisa

Para alcançar o objetivo deste trabalho, nesta seção são apresentadas as abordagens utilizadas para a realização da análise comparativa de arquiteturas de redes neurais profundas para Análise de Sentimentos supervisionada. O processo de Análise de Sentimentos aplicado nas arquiteturas de RNA aqui analisadas, foi dividido em 4 etapas conforme ilustrado na Figura 13: (1) obtenção de coleções de Análise de Sentimento, (2) pré-processamento das coleções, (3) aprendizado, (4) avaliação/validação. Cada etapa desempenha um papel específico para o resultado final da análise.

Figura 13. Esquema Geral do Método de Pesquisa



Fonte: Autoria Própria, 2021

3.1. Coleções de Análise de Sentimento

Neste presente trabalho, as arquiteturas de redes neurais profundas foram aplicadas em duas coleções de Análise de Sentimentos. A primeira coleção utilizada se chama *IMDB Dataset of 50K Movie Reviews*, disponibilizado pela plataforma *Kaggle*, a qual contém 50.000 documentos de resenhas (*reviews*) de filmes da plataforma *Internet Movie Database* (IMDB), juntamente com suas polaridades positiva e negativa, sendo 25.000 resenhas positivas e 25.000 resenhas negativas [25]. Isso significa que a base de dados contém um número igual de avaliações positivas e negativas, evitando assim algum tipo de enviesamento dos resultados por conta do desbalanceamento das classes.

A segunda coleção utilizada tem como domínio notícias do mercado financeiro. O *dataset* chama-se *Sentiment Analysis for Financial News* (Análise de Sentimento para Notícias Financeiras), disponibilizado pela plataforma *Kaggle*, a qual contém as polaridades positiva, negativa e neutra para as manchetes de notícias financeiras através da perspectiva de um investidor de varejo [34]. A coleção possui 4837 documentos, nos quais 2871 são notícias neutras, 1363 positivas e 604 negativas. Esta coleção foi escolhida pois o domínio de notícias do mercado financeiro é um domínio no qual, diferentemente da base IMDB, apresenta características como textos escritos em linguagem formal, grande quantidade de palavras, além da expressão dos sentimentos ser menos explícita. Além disso, é um domínio interessante, pois muitas vezes o sentimento embutido nas notícias pode afetar o valor de empresas e *commodities* no mercado financeiro.

3.2. Pré-processamento

Na etapa de pré-processamento das coleções de dados deste trabalho, foram aplicadas diversas técnicas para remover ruídos e elementos textuais irrelevantes, objetivando melhorar a qualidade dos dados para assim alcançar melhores resultados na classificação

do sentimento dos documentos [1]. Em ambos *datasets* utilizados, foram aplicadas as seguintes técnicas:

- **Padronização da coleção textual:** foram removidos caracteres irrelevantes, como sinais de pontuação, caracteres especiais e alfanuméricos, assim como padronização dos textos em caixa baixa, ou seja, passaram a ter apenas caracteres minúsculos, para que palavras como “Subiu” e “subiu” sejam vistas como palavras diferentes.
- **Remoção de stopwords:** nesta etapa ocorre a remoção de palavras pouco discriminativas, como artigos, pronomes, e advérbios, visto que estas palavras não expressam nenhum sentimento, e não contribuem para o significado de uma frase [1]. Neste contexto, foi utilizado a lista de *stopwords* disponibilizada pela biblioteca de processamento de linguagem natural *Spacy*¹, que contém as palavras consideradas irrelevantes em diversos idiomas.
- **Lematização:** este processo tem como objetivo agrupar as variantes de um termo em um único lema, ou seja, transformar verbos para sua forma no infinitivo, e substantivos e adjetivos para o masculino singular [1]. Por exemplo, as palavras “casa”, “casas”, “casinha” e “casarão” são todas formas do mesmo lema “casa”. Assim como as palavras “tiver”, “tinha”, “tenho” e “terei” são do mesmo lema “ter”. Bem como a técnica *stemming*, a lematização auxilia na redução da dimensionalidade dos atributos. Para aplicar a técnica lematização (*lemmatisation*), também foi utilizado a biblioteca *Spacy* em ambos os *datasets*.
- **Label encoding:** modelos de Aprendizado de Máquina e Aprendizado Profundo exigem que todas as variáveis de entrada e saída da rede sejam numéricas. Dessa forma, foi utilizada a biblioteca *Sklearn* para aplicar a técnica *label encoding* nas coleções de dados, para transformar os dados na coluna onde as polaridades (positivo, negativo e neutro) estão presentes, para um valor numérico (0, 1, 2).
- **Word embeddings:** essa técnica objetiva representar uma palavra por meio de um vetor numérico de baixa dimensionalidade. A obtenção desse vetor é feita utilizando o contexto das palavras, isto é, palavras ao redor, de forma que palavras que ocorram em contexto parecidos serão representadas por vetores parecidos. Há diversos métodos para criação de *word embeddings*, assim como a possibilidade de utilizar *word embeddings* pré-treinadas, como o *Word2Vec* ou *GloVe* [1]. Neste presente trabalho foi utilizado o *word embedding* pré-treinado *GloVe* (*Global Vectors for Word Representation*) disponibilizado pela universidade *Stanford*², a qual possui 400 mil palavras e dimensionalidade 300.

3.3. Aprendizado

Neste trabalho, foram utilizadas em todas as arquiteturas de RNA aqui analisadas a *word embedding* pré-treinada *GloVe* na camada de *embedding*, função ReLU nas camadas densas e a função de ativação Softmax na camada de classificação. Especificamente nas células de memória das redes recorrentes, células essas que pertencem à uma camada escondida, foi utilizada a função de ativação Tangente Hiperbólica (TanH). A função de ativação Softmax costuma ser aplicada na camada de saída da rede neural em problemas de classificação multiclasse, e tem como objetivo inferir a probabilidade de um exemplo

¹<https://spacy.io/>

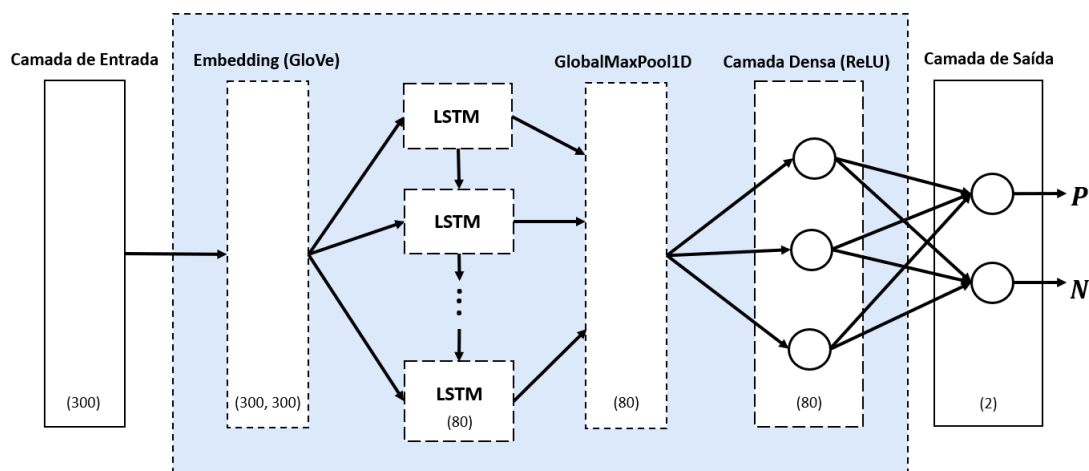
²<https://nlp.stanford.edu/projects/glove/>

de entrada pertencer a cada uma das classes [1]. Por exemplo, em um problema que contenha 3 classes, positiva, negativa e neutra, a função Softmax irá produzir 3 valores [0.42, 0.31, 0.27], que somam 1, onde cada valor representa a probabilidade de cada instância pertencer a uma das 3 possíveis classes. Assim, pode-se determinar que a classe estimada pela rede neural é a que possui a maior probabilidade, ao qual no exemplo anterior seria a classe positivo.

A função de ativação *Rectified Linear Unit* (ReLU), ou unidade linear retificada, é hoje uma das funções de ativação mais utilizadas no treinamento de redes neurais, e costuma ser utilizada nas camadas ocultas (*hidden layers*). A função ReLU, produz valores no intervalo $[0, \infty[$, como é apresentado na Figura 5, e retorna 0 para todos os valores negativos, e o próprio valor para valores positivos, ou seja, esta função de ativação não propaga valores negativos para a camada subsequente da rede [2].

Em relação ao aprendizado, na etapa de treinamento de todas as arquiteturas de RNAs foi utilizada a função de perda (*Loss Function*) chamada Entropia Cruzada Categórica Esparsa (*Sparse Categorical Crossentropy*), a qual é utilizada quando há dois ou mais neurônios na camada de saída (classificação multiclasse) e a representação das classes é do tipo esparsa (cada classe é representada por um único valor). Também foi utilizada a função de otimização Adam [2], disponibilizadas pela biblioteca Keras³.

Figura 14. Arquitetura da Rede Neural Recorrente LSTM utilizada neste trabalho e instanciada para duas classes, positiva e negativa



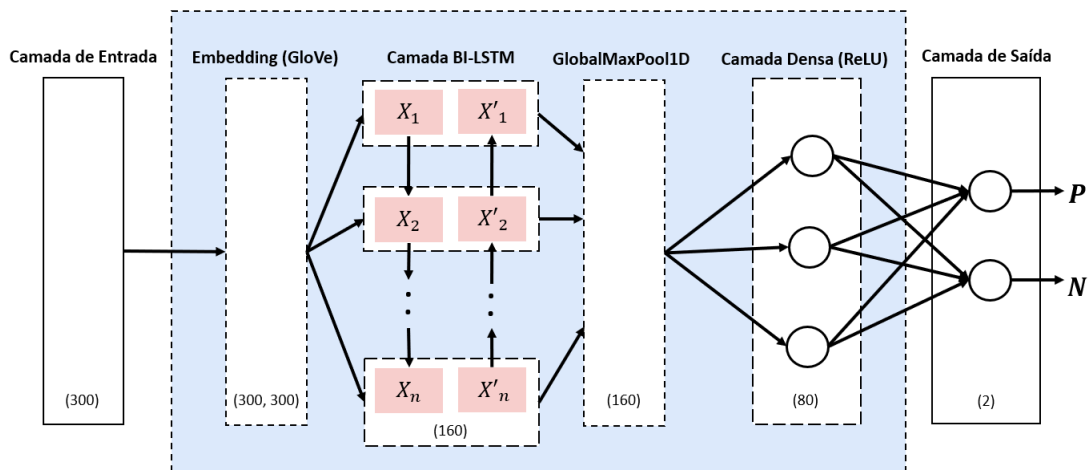
Fonte: Autoria Própria, 2021

Todos os modelos desenvolvidos utilizaram parâmetros de treinamento iguais, como *batch-size*=128 (tamanho do lote de treinamento), *epochs*=10 (número máximo de épocas). Tanto a arquitetura de rede neural recorrente LSTM (Figura 14), Bi-LSTM (Figura 15) e GRU (Figura 16) usam uma camada de entrada possuindo 300 nós, seguida de uma camada *embedding* com formato de entrada 300 e de saída 300, mudando apenas a configuração das células de memória das redes recorrentes segundo suas especificidades com formato de entrada 300 e de saída 80, e função de ativação Tangente Hiperbólica

³<https://keras.io/>

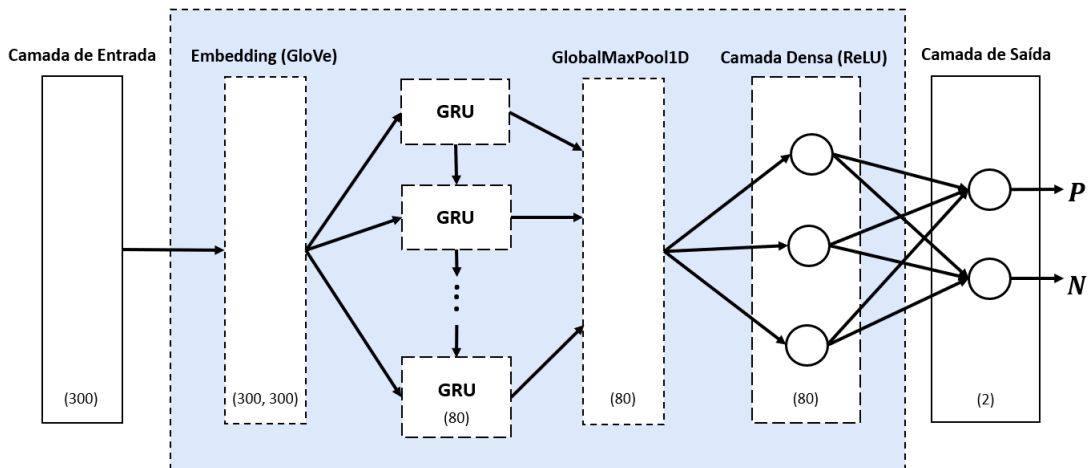
(TanH). Por fim, é utilizada uma camada GlobalMaxPool1D, seguida por uma camada densamente conectada com função de ativação ReLU, com a saída conectada à uma última camada densa com dois neurônios de saída e função de ativação *Softmax*, onde serão produzidas a saída de dois valores no intervalo entre [0, 1] que representam a probabilidade de o texto ser positivo ou negativo.

Figura 15. Arquitetura da Rede Neural Recorrente Bi-LSTM utilizada neste trabalho e instanciada para duas classes, positiva e negativa



Fonte: Autoria Própria, 2021

Figura 16. Arquitetura da Rede Neural Recorrente GRU utilizada neste trabalho e instanciada para duas classes, positiva e negativa

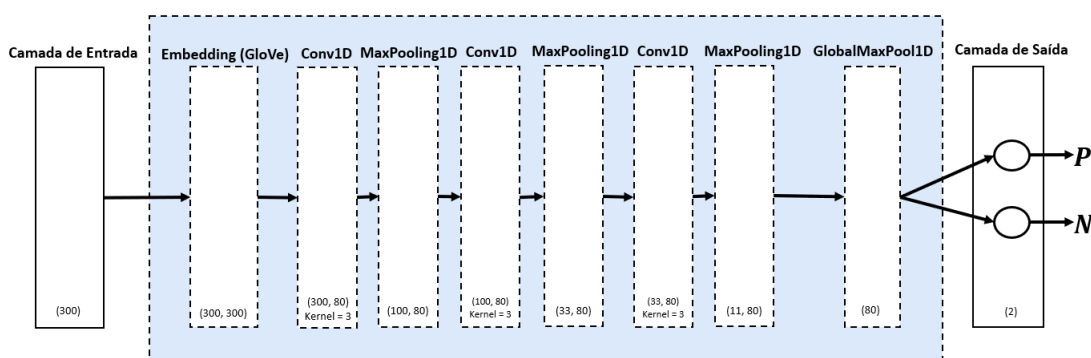


Fonte: Autoria Própria, 2021

Em relação a arquitetura de rede neural convolucional CNN desenvolvida neste trabalho, o modelo possui uma camada de entrada com 300 nós, em seguida uma camada *embedding* com formato de entrada 300 e de saída 300. As próximas 6 camadas

da arquitetura, são compostas por uma camada Conv1D, seguida por outra camada Max-Pooling1D. A camada convolucional Conv1D possui formato de entrada (300, 80) com 80 filtros, *kernel* de tamanho 3 e função de ativação ReLU. Em seguida há uma camada de MaxPooling1D com formato de saída (100, 80). Por fim, é utilizada uma camada GlobalMaxPool1D, seguida por uma camada densamente conectada com dois neurônios de saída e função de ativação *Softmax*, onde serão produzidas a saída de dois valores no intervalo entre [0, 1] que representam a probabilidade de o texto ser positivo ou negativo (no caso de uma classificação com apenas duas classes), conforme apresentado na Figura 17.

Figura 17. Arquitetura da Rede Neural Convolucional CNN utilizada neste trabalho e instanciada para duas classes, positiva e negativa



Fonte: Autoria Própria, 2021

3.4. Critérios de Avaliação / Validação

Para realizar a avaliação da performance de classificação, foram utilizadas métricas e esquemas de avaliação comumente utilizadas em problemas de classificação, como matriz de confusão e *k-fold cross-validation* [1]. Uma matriz de confusão é um artefato usado para avaliar a performance de modelos destinados a realização de classificação em *Machine Learning* (Aprendizado de Máquina). Os valores que compõem a matriz são obtidos após a fase de treinamento do modelo, e a eficiência é medida ao fornecer os segmentos do conjunto de testes ao método de classificação e comparando as previsões com os rótulos corretos dos exemplos de teste.

Os itens apresentados na Figura 18 correspondem respectivamente à [29]:

- **Verdadeiros Positivos (VP):** o classificador previu o exemplo como sendo da classe positiva e o exemplo pertence à classe positiva, ou seja, o classificador previu corretamente.
- **Falsos Positivos (FP):** o classificador previu o exemplo como sendo da classe positiva, porém, o exemplo pertence à classe negativa.
- **Falsos Negativos (FN):** o classificador previu o exemplo como sendo da classe negativa, porém, o exemplo pertence à classe positiva.
- **Verdadeiros Negativos (VN):** o classificador previu o exemplo como sendo da classe negativa, e o exemplo pertence à classe negativa, ou seja, o classificador previu corretamente.

Figura 18. Matriz de confusão contendo duas classes

| | | Valor Verdadeiro (confirmado por análise) | |
|--|-----------|--|------------------------------------|
| | | Positivos | Negativos |
| Valor Previsto (predito pelo teste) | Positivos | VP Verdadeiros Positivos | FP Falsos Positivos |
| | Negativos | FN Falsos Negativos | VN Verdadeiros Negativos |

Fonte: Autoria Própria, 2021

Após a realização da contagem dos termos e assim a geração da matriz de confusão, é possível então calcular métricas para avaliar o desempenho do modelo [29]. A seguir, são apresentadas as métricas que foram utilizadas para avaliar as diferentes arquiteturas de redes neurais consideradas neste trabalho:

- **Acurácia** (*accuracy*): indica o desempenho geral do modelo. Esta métrica diz basicamente, dentre todas as classificações realizadas, a porcentagem de classificações realizadas corretamente. Para realizar o cálculo da acurácia de um modelo, basta simplesmente realizar a divisão de todas as predições corretas pelo total.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

- **Precisão** (*precision*): esta métrica indica, dentre todas as classificações positivas que o modelo realizou, quantas delas estão corretas. Para efetuar este cálculo, é necessário dividir os valores VP (Verdadeiro Positivo) pela soma entre VP e o FP (Falso Positivo).

$$Precision = \frac{VP}{VP + FP}$$

- **Revocação** (*Recall*): é uma métrica que visa medir o desempenho do modelo em prever ou recuperar dos exemplos que pertençam a uma determinada classe. Para efetuar este cálculo, é necessário dividir os valores VP (Verdadeiro Positivo) pela soma entre VP e o FN (Falso Negativo).

$$Recall = \frac{VP}{VP + FN}$$

- **F₁-Score**: o *F₁-Score* é basicamente uma forma de observar apenas uma métrica ao invés de duas em alguma situação, pois esta métrica é uma média harmônica entre precisão e *recall*, ao qual gera um número único que indica a qualidade geral do modelo. A seguir é mostrado a formula para calcular a métrica *F₁-Score*.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

As explicações apresentadas anteriormente para a matriz de confusão referem-se à classificação binária. Porém, devido a utilização da coleção de dados *Sentiment Analysis for Financial News*, ao qual diferentemente da coleção *IMDB Reviews*, possui não apenas as polaridades positivo e negativo, mas também a polaridade neutra, ou seja, o problema passa a ser multiclasse (três classes), e requer algumas adaptações da matriz de confusão. A matriz de confusão para um problema n -classes se torna uma matriz $n \times n$, como é apresentado na Figura 19. Para uma melhor visualização e compreensão do leitor, foi adicionado à matriz mais uma linha chamada FP (Falsos Positivos), e mais uma coluna chamada FN (Falsos Negativos). Pode-se observar na Figura 19 que os elementos Verdadeiros Positivos (VP), em cor verde, e os elementos Verdadeiros Negativos (VN), em cor vermelha, compõem um mesmo conjunto (mesma linha). Após obtida a matriz de confusão para o problema multiclasse, existem diferentes estratégias para calcular as métricas apresentadas anteriormente [30]. Neste trabalho, foram adotadas duas estratégias: *micro-averaging*, em que é feito um somatório dos termos de cada métrica para cada classe, e *macro-averaging*, ao qual é feita uma média da métrica calculada para cada classe.

Figura 19. Exemplo Matriz de Confusão para Problemas de Multiclasse

| | | Previsto | | | FN |
|-------------|----------|----------|----------|--------|----|
| | | Positivo | Negativo | Neutro | |
| Classe Real | Positivo | 6 | 4 | 0 | 4 |
| | Negativo | 3 | 4 | 2 | 5 |
| | Neutro | 0 | 1 | 11 | 1 |
| FP | | 3 | 5 | 2 | 10 |

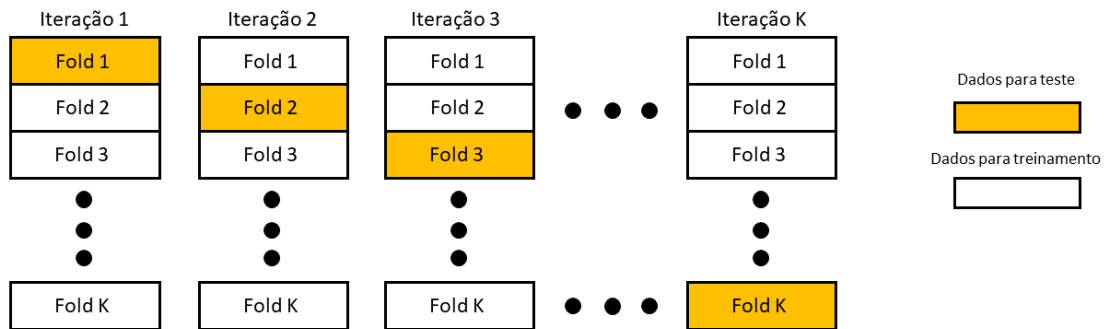
Fonte: Autoria Própria, 2021

Para estimar a performance dos modelos na prática, foi utilizada a técnica *k-Fold Cross-Validation*, ao qual é uma técnica que avalia a capacidade de generalização de um modelo, ou seja, como o modelo se comporta ao tentar prever um dado que nunca viu [1]. O método *k-Fold Cross-Validation* consiste em dividir a base de dados k vezes, chamadas de *folds* – por isso o nome *k-Fold*. Após a divisão, uma das partes é escolhida para testar o modelo e o restante é utilizado para treinamento. Essa tarefa é repetida k iterações até que o modelo seja treinado e testado com todas as partes anteriormente divididas. Ao final de cada iteração, pode-se calcular todas as métricas de avaliação anteriormente apresentadas, como a acurácia do modelo, e realizar uma média dos resultados obtidos para todas as pastas de teste, de forma a se obter um resultado geral do processo de avaliação [29]. Na Figura 20 é apresentada uma ilustração dos conjuntos de dados para treino e teste gerados pelo procedimento de validação cruzada *k-Fold*. Nos experimentos realizados na base de dados deste trabalho foram utilizados 10 *folds*.

4. Resultados

Nesta seção são apresentados os resultados considerando as coleções, arquiteturas de redes e esquemas de avaliação apresentados na Seção 3. Nas Figuras 21 e 22 são apresentados os resultados para as coleções *IMDB* e *Financial New* respectivamente.

Figura 20. Processo de validação cruzada k-Fold



Fonte: Autoria Própria, 2021

Como pode-se observar na Figura 21, o modelo Bi-LSTM (*Bidirectional Long Short-Term Memory*) se destacou em relação aos demais modelos desenvolvidos nos experimentos utilizando o conjunto de dados IMDB, obtendo a acurácia de 97.84%. Por conter duas camadas de rede LSTM, uma responsável por lidar com os estados passados, e a outra com os estados futuros, esta bidirecionalidade dos dados faz com que este tipo de arquitetura possa compreender melhor o contexto, e assim, isso pode ter contribuído com a alta acurácia na tarefa de classificação. Vale destacar o resultado do modelo de rede neural convolucional (CNN), ao qual ficou em segundo lugar nos testes com a acurácia de 97.46%, obtendo assim uma melhor performance que a arquitetura de rede neural recorrente GRU com acurácia de 95.42% e até mesmo da arquitetura LSTM com acurácia 97.14%.

Figura 21. Resultado da Avaliação de Desempenho Dataset IMDB

| | Acurácia | Micro Precisão | Macro Precisão | Micro Revocação | Macro Revocação | Micro F1 Score | Macro F1 Score |
|----------------|---------------|----------------|----------------|-----------------|-----------------|----------------|----------------|
| LSTM | 97.14% | 97.14% | 97.15% | 97.14% | 97.14% | 97.14% | 97.14% |
| BI-LSTM | 97.84% | 97.84% | 97.85% | 97.84% | 97.84% | 97.84% | 97.84% |
| GRU | 95.42% | 95.42% | 95.45% | 95.42% | 95.42% | 95.42% | 95.42% |
| CNN | 97.46% | 97.46% | 97.46% | 97.46% | 97.46% | 97.46% | 97.46% |

Fonte: Autoria Própria, 2021

Na Figura 22 são apresentados os resultados da avaliação de desempenho das arquiteturas utilizando o conjunto de dados *Financial News*. Destaca-se nessa avaliação o modelo de rede neural convolucional (CNN), ao qual apresentou os melhores resultados em relação as outras arquiteturas com a acurácia de 90.04%, e em segundo lugar, a rede neural recorrente Bi-LSTM com a acurácia de 89.96%. Infere-se que a melhor performance da CNN nessa situação é que no domínio de notícias, não necessariamente uma sequência de palavras pode ser determinante, mas sim um conjunto de palavras que podem estar distantes, mas que são filtrados pela CNN.

Figura 22. Resultado da Avaliação de Desempenho Dataset Financial News

| | Acurácia | Micro Precisão | Macro Precisão | Micro Revocação | Macro Revocação | Micro F1 Score | Macro F1 Score |
|------------|---------------|----------------|----------------|-----------------|-----------------|----------------|----------------|
| LSTM | 85.74% | 85.74% | 84.25% | 85.74% | 81.18% | 85.74% | 82.32% |
| BI-LSTM | 89.96% | 89.96% | 88.56% | 89.96% | 85.93% | 89.96% | 86.98% |
| GRU | 83.86% | 83.86% | 81.59% | 83.86% | 79.09% | 83.86% | 80.02% |
| CNN | 90.04% | 90.04% | 87.77% | 90.04% | 84.88% | 90.04% | 85.92% |

Fonte: Autoria Própria, 2021

5. Considerações Finais e Trabalhos Futuros

Nos últimos anos, a área de Análise de Sentimentos vem ganhando atenção, tanto na Web quanto na academia, pois instituições, pessoas e companhias se interessam em saber a opinião real de um grupo de pessoas a respeito de um tema específico. Esta análise comparativa foi motivada pela crescente demanda por sistemas automatizados, capazes de analisar a polaridade da opinião com precisão e eficiência, o conteúdo de grandes volumes de documentos publicados online. Dentre os diversos métodos para se extrair a polaridade dos sentimentos expressos em bases textuais, se destacam as arquiteturas de Redes Neurais Artificiais (RNAs).

Neste contexto, este trabalho de conclusão de curso visou realizar uma análise comparativa do desempenho de arquiteturas de RNAs em Análise de Sentimentos das seguintes arquiteturas: Rede Neural Recorrente do tipo LSTM (*Long Short-Term Memory*), Rede Neural Convolutacional (CNN), Rede Neural Recorrente do tipo *Gated Recurrent Unit* (GRU), Rede Neural Recorrente do tipo Bi-LSTM (*Bidirectional Long Short-Term Memory*). O objetivo foi indicar ao leitor, dentre as arquiteturas analisadas, qual tem maior desempenho ao realizar a tarefa de extrair a polaridade dos sentimentos expressos em bases textuais.

Após uma série de experimentos utilizando técnicas de avaliação de desempenho, os resultados obtidos por meio destes experimentos nos dois conjuntos de dados, IMDB e *Financial News*, mostram que o modelo de rede neural que apresentou melhor performance no *dataset* IMDB foi a rede neural recorrente Bi-LSTM com a acurácia de 97.84%, e em segundo lugar a rede neural convolutacional CNN, com a acurácia de 97.46%. Já no *dataset Financial News*, destaca-se o modelo de rede neural convolutacional CNN com a acurácia de 90.04%, e em segundo lugar a rede neural recorrente Bi-LSTM, com a acurácia de 89.96%. Os resultados demonstram assim que, nos domínios de resenha de filmes, a rede neural que extrai padrões sequenciais, como é o caso da arquitetura Bi-LSTM, obteve maior performance ao realizar a tarefa de extrair a polaridade dos sentimentos expressos em bases textuais, em relação a rede neural que extrai padrões globais, como é o caso da CNN. Já no conjunto de dados que tem por domínio notícias financeiras, a rede neural que extrai padrões globais, como é o caso da arquitetura CNN, obteve maior performance em relação a arquitetura Bi-LSTM, que extrai padrões sequenciais. Porém, em ambos os domínios, tanto a Bi-LSTM quanto a CNN obtiveram as melhores performances de classificação, sendo assim indicadas para a utilização na prática.

Com intuito de melhorar a performance e acurácia da extração de conhecimento em Análise de Sentimentos, pretende-se como trabalho futuro (i) considerar outras arquiteturas presentes na literatura, como redes neurais que combinam camadas de atenção e

de memória, uma vez que demonstram ser promissoras em Análise de Sentimentos; e (ii) considerar diferentes *word embeddings* pré-treinadas, e novos parâmetros de configuração das arquiteturas de RNA.

Referências

- [1] Charu C. Aggarwal. *Machine Learning for Text*. Springer Publishing Company, Incorporated, 1st edition, 2018. ISBN 3319735306.
- [2] Charu C Aggarwal et al. *Neural networks and deep learning*. Springer, 2018.
- [3] Grant Michael Armstrong. *Examining the Parent-child Political Relationship*. PhD thesis, Oklahoma State University, 2015.
- [4] Jorge M Barreto. Introdução às redes neurais artificiais. *Florianópolis: Ufsc*, 2002.
- [5] Hsinchun Chen. *Knowledge management systems: a text mining perspective*. Knowledge Computing Corporation, 2001.
- [6] Aranha Christian and Passos Emmanuel. A tecnologia de mineração de textos. *Revista Eletrônica de Sistemas de Informação*, 5, 01 2009. doi: 10.21529/RESI.2006.0502001.
- [7] Thayná Conceição and Rafael Rossi. Desenvolvimento de uma ferramenta para análise de sentimentos de textos publicados no twitter - trabalho de conclusão de curso - sistemas de informação - UFMS/CPTL, 12 2017.
- [8] Felipe Contratres, Solange N A Souza, Lucia Filgueiras, and L. DeSouza. *Sentiment Analysis of Social Network Data for Cold-Start Relief in Recommender Systems*, pages 122–132. 03 2018. ISBN 978-3-319-77711-5. doi: 10.1007/978-3-319-77712-2_12.
- [9] Murilo Gazzola. Um método para avaliação automática da qualidade de recursos educacionais abertos usando deep learning. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1477, 2017.
- [10] Andrew B Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 45–52. Association for Computational Linguistics, 2006.
- [11] Lucas Guarise and Solange Oliveira Rezende. Detecção de notícias falsas usando técnicas de deep learning. 2019. URL <https://bdta.aguia.usp.br/item/002992853>.
- [12] Victor O. Gutemberg. Avaliação do desempenho das redes neurais convolucionais na detecção de ovos de esquistossomose. 2017. URL https://www.cin.ufpe.br/~tg/2017-1/vgom_tg.pdf.
- [13] Abdalraouf Hassan and Ausif Mahmood. Efficient deep learning model for text classification based on recurrent and convolutional layers. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1108–1113. IEEE, 2017.
- [14] Brian Heredia, Joseph D Prusa, and Taghi M Khoshgoftaar. Location-based twitter sentiment analysis for predicting the us 2016 presidential election. In *The Thirty-First International Flairs Conference*, 2018.
- [15] W. L. Hor, W. X. Goh, and S. H. Ow. Impact of news sentiment analysis on market price movement during the dividend period. *International Journal of Engineering & Technology*, 7(4.11):168–173, 2018. ISSN 2227-524X.

- [16] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
- [17] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, pages 607–618. ACM, 2013.
- [18] Jaron Lanier. *Ten Arguments for Deleting Your Social Media Accounts Right Now*. Henry Holt and Co., 3 edition, 4 2018. ISBN 125019668X. An optional note.
- [19] Bing Liu. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015. doi: 10.1017/CBO9781139084789.
- [20] Marc Moreno Lopez and Jugal Kalita. Deep learning applied to nlp. *arXiv preprint arXiv:1703.03091*, 2017.
- [21] Bernard Marr. How much data do we create every day? the mind-blowing stats everyone should read. <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/>, 3 2018.
- [22] Guilherme Martins, Glauber Colnago, and Juan Castañeda-Ayarza. Desenvolvimento e comparação de léxico e rede neural para a análise de sentimento de notícias. 01 2019.
- [23] Tom M Mitchell. *Machine learning*, 1997.
- [24] Mehdi Moussaïd, Juliane E Kämmer, Pantelis P Analytis, and Hansjörg Neth. Social influence and the collective dynamics of opinion formation. *PloS one*, 8(11):e78433, 2013.
- [25] Lakshmi pathi N. Imdb dataset of 50k movie reviews, 2019. URL <https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>. [Online; acessado em 26-Novembro-2020].
- [26] Peter Norvig and Stuart Russell. *Inteligência Artificial: Tradução da 3a Edição*, volume 1. Elsevier Brasil, 2014.
- [27] Jakub Nowak, Ahmet Taspinar, and Rafał Scherer. Lstm recurrent neural networks for short text and sentiment classification. In *International Conference on Artificial Intelligence and Soft Computing*, pages 553–562. Springer, 2017.
- [28] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 01 2008. doi: 10.1561/1500000011.
- [29] Alison Pereira Ribeiro and NFF Silva. Um estudo comparativo sobre métodos de análise de sentimentos em tweets. *Revista de Sistemas de Informação da FSMA*, 1 (22):35–48, 2018.
- [30] Rafael Geraldeli Rossi. *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2015.
- [31] Luciana Maiara Queiroz de Santana. Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos. 2017.
- [32] Preeti Shrivastav and Dr John. Web 2.0 technologies and its application: A study of management science students davv, indore, m. p. *IOSR Journal of Humanities and Social Science*, 19:82–87, 01 2014. doi: 10.9790/0837-19538287.
- [33] Vikas Sindhvani and Prem Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *2008 Eighth IEEE International Conference on Data Mining*, pages 1025–1030. IEEE, 2008.

- [34] Ankur Sinha. Sentiment analysis for financial news, 2020. URL <https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news>. [Online; acessado em 26-Novembro-2020].
- [35] J Trofimovich. Comparison of neural network architectures for sentiment analysis of russian tweets. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue*, pages 50–59, 2016.
- [36] N. Ur-Rahman and Jenny Harding. Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Syst. Appl.*, 39:4729–4739, 04 2012. doi: 10.1016/j.eswa.2011.09.124.
- [37] Ramesh Wadawadagi and Veerappa Pagi. Sentiment analysis with deep neural networks: comparative study and performance assessment. *ARTIFICIAL INTELLIGENCE REVIEW*, 2020.
- [38] Ivone Penque Matsuno Yugoshi. *Mineração de opiniões baseada em aspectos para revisões de produtos e serviços*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2018.