Importante: Artigo apenas **para fins pedagógicos**. Os resultados **não** devem ser considerados como referência para outras finalidades, pois **podem conter erros**.

UTILIZAÇÃO DE LSTM E GRU PARA CLASSIFICAR AVALIAÇÕES DE PRODUTOS EM COMÉRCIO ELETRÔNICO

RELATÓRIO EP1 - MAC5725 - LINGUÍSTICA COMPUTACIONAL

Nahim Alves de Souza
Instituto de Matemática e Estatística
Universidade de São Paulo
São Paulo, SP, Brasil
nahim@usp.br

Outubro, 2023

RESUMO

As redes neurais recorrentes, sobretudo os modelos LSTM e GRU, apresentaram um bom desempenho em tarefas de Processamento de Linguagem Natural, devido à sua característica guardar informações do contexto e em uma memória, e utilizar essa memória, por exemplo, na predição de sequências de caracteres ou palavras. Esse trabalho apresenta uma análise da aplicação desses modelos em uma tarefa de classificação de textos, originados do corpus de avaliações de produtos da B2W. Ao final, são apresentados os resultados que demonstram o bom desempenho dessas redes nesse cenário, avaliando também o quanto os hiper-parâmetros, como o *dropout*, podem influenciar no desempenho da rede.

Keywords RNN · LSTM · GRU · NLP

1 Introdução

Nos últimos anos, as redes neurais têm sido utilizadas em um grande número de aplicações LeCun et al. [2015]. Em especial, na área de Processamento de Linguagem Natural (em inglês, *Natural Language Processing* - NLP), as redes neurais recorrentes trouxeram ótimos resultados em tarefas de predição de sequências de caracteres e palavras em um texto, principalmente com a introdução dos modelos LSTM e GRU.

O objetivo desse trabalho é avaliar a capacidade de predição de redes neurais LSTM e GRU nesse cenário, utilizando o corpus de avaliações de produtos da B2W[1]. Nesse corpus existem cerca de 130 mil avaliações de produtos, fornecidas pelos consumidores. Cada avaliação possui um texto com a opinião do cliente sobre o produto e outros campos que informam, por exemplo, a nota de avaliação do produto (1 a 5 estrelas) e uma indicação se o cliente recomendaria ou não o produto para outras pessoas.

Para avaliar o desempenho das redes LSTM e GRU nesse cenário, foi proposto utilizar estas redes para predizer a nota de avaliação a partir do texto de avaliação do produto. As seções seguintes apresentam uma breve introdução sobre esses tipos de redes neurais recorrentes, uma análise dos dados utilizados, a descrição dos experimentos, e por fim, uma avaliação dos resultados obtidos.

[1] O corpus da B2W está disponível em: https://github.com/americanas-tech/b2w-reviews01

2 Redes neurais recorrentes

A redes neurais recorrentes (em inglês, *RNNs - Recurrent Neural Networks*) possuem uma arquitetura baseada em uma célula de memória, composta de uma ou mais *hidden layers*, alimentada recorrentemente com as saídas de cada iteração (Figura 1). Essa construção permite que a rede capture e extraia informações relacionadas à sequência em os dados são inseridos na rede [Lindemann et al., 2021].

As RNNs serviram de base para a criação dos modelos *Seq2Seq* e *Encoder-Decoder*, muito utilizados no processamento de linguagem natural [Lindemann et al., 2021] como, por exemplo, predição de sequências de caracteres ou palavras, ou ainda, tradução de textos para diversos idiomas [LeCun et al., 2015].

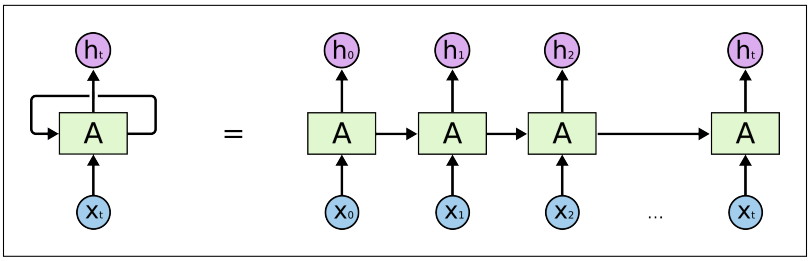


Figura 1: Representação de uma RNN "desenrolada"[2]. Conforme mostrado do lado esquerdo da figura, a rede possui uma única célula (representada pelo retângulo verde com a letra **A**), mas a cada iteração, a célula recebe uma entrada nova (x_t) e a saída do estado anterior (h_{t-1}).

A recorrência presente nas RNNs, no entanto, pode causar alguns problemas como o de gradiente desvancente (*vanishing gradient*), que ocorre quando o valor do gradiente é muito pequeno e se torna cada vez menor na retro-propagação [Lindemann et al., 2021]. Esse problema não é facilmente tratável e requer a adoção de abordagens sofisticadas para ser mitigado.

2.1 LSTM - *Long Short-Term Memory*

As LSTMs foram projetadas por [Hochreiter and Schmidhuber, 1997] com o objetivo de minimizar o problema de *vanishing gradient* das RNNs, através da combinação de circuitos que permitem selecionar as informações que serão guardadas ou esquecidas em cada iteração ao longo da rede. Como pode ser visto na Figura 2, a grande diferença da LSTM está no fato de que as células de memória são diferentes daquelas de uma RNN simples. Nas LSTMs, existem três portas lógicas (*input gate*, *forget gate* e *output gate*) que permitem controlar o fluxo de informações que entra e sai em cada iteração da rede (também referido como *estado* da rede).

2.2 GRU - *Gated Recurrent Unit*

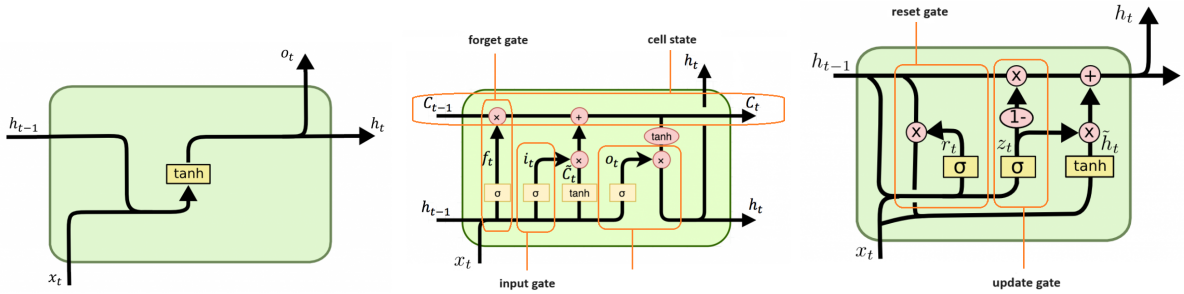
As GRUs foram propostas por [Cho et al., 2014] como uma simplificação das LSTMs que busca manter as mesmas propriedades, mas com menor exigência de poder de processamento. Cada célula de uma GRU é composta por um circuito com duas portas: *update gate* (uma combinação do *input gate* e *forget gate*, que decide se a informação do estado anterior será descartada ou preservada e quanto de informação nova será acrescentada) e *reset gate* (que determina quanto da informação do estado anterior deve ser preservada). A Figura 2 ilustra a diferença entre uma célula de RNN simples, uma LSTM e uma GRU. Em geral GRUs e LSTMs apresentam desempenho semelhante para várias tarefas, no entanto, as LSTMs ainda se mostram mais poderosas em tarefas mais complexas [Lindemann et al., 2021].

2.3 Redes bidirecionais

Uma outra variação das RNNs são as redes neurais bidirecionais, introduzidas por [Schuster and Paliwal, 1997]. As redes bidirecionais possuem uma camada extra, responsável por analisar a sequência na direção reversa à original (Figura 3). Esse modelo permite que o contexto seja analisado de forma mais completa, para os estados passados e

[2]Imagem disponível em: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

[3]Imagem adaptada de `http://dprogrammer.org/rnn-lstm-gru`



(a) RNN Simples (b) LSTM (c) GRU

Figura 2: Representação do funcionamento das células de memória de redes RNN, LSTM e GRU[3]. Note que em uma RNN simples, apenas uma função $tanh(x)$ é usada como função de ativação, enquanto nas outras duas redes as funções de ativação são mais complexas.

futuros de uma sequência de dados. Ao tentar predizer uma palavra em uma sequência, esse modelo permite que todo o contexto ao redor de uma palavra seja analisado, fornecendo mais informações para a rede que está sendo treinada, e consequentemente melhorando a sua acurácia.

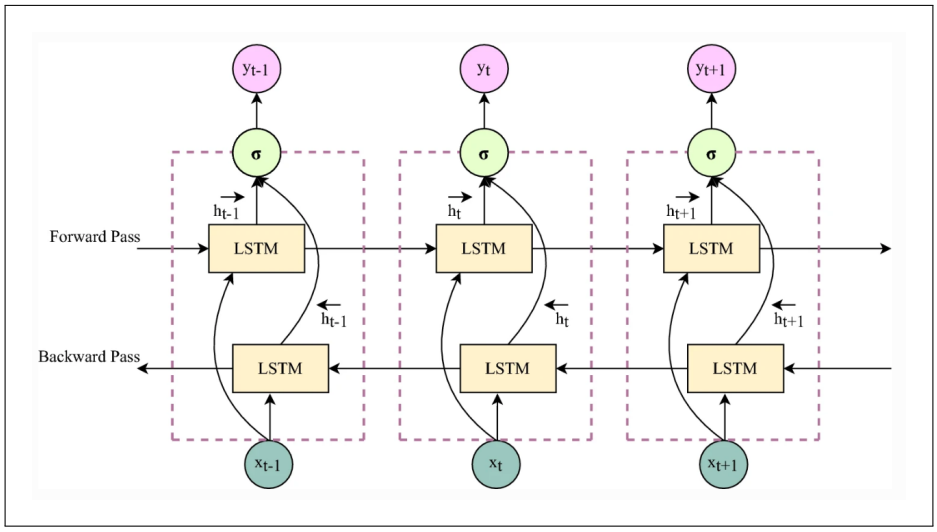


Figura 3: Ilustração representando uma rede LSTM bidirecional[4]. As células LSTM são utilizadas em duas direções, uma recebendo a sequência original (*Forward Pass*) e outra recebendo a sequência reversa (*Backward Pass*).

3 Análise do corpus de avaliações da B2W

O corpus da B2W utilizado é composto por uma série de avaliações de produtos, fornecidas por consumidores de uma plataforma de comércio eletrônico da B2W, a *americanas*. Para o propósito desse trabalho, dois atributos dessas avaliações serão analisados: `review_text` (referido como `texto`) e `overall_rating` (referido como `rotulo`), que correspondem, respectivamente, ao texto de avaliação do produto e à nota dada pelo consumidor (1 a 5 estrelas).

A base original é composta por mais de 130 mil avaliações, mas um processo de limpeza e filtragem foi realizado para remover amostras com valores inválidos. Foram removidas amostras onde um dos campos possuía valor nulo e onde o

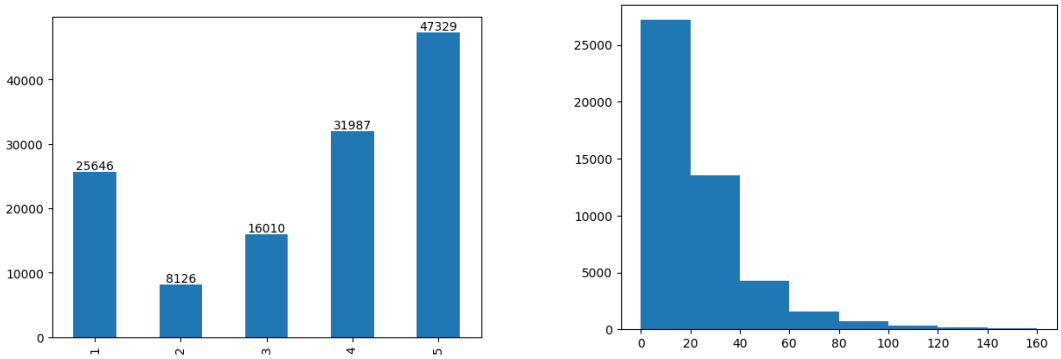
[4]Imagem disponível no trabalho de [Naik and Jaidhar, 2022]. Disponível em: `https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00664-6`

`rotulo` era diferente de um valor numérico entre 1 e 5. Após esse processamento, a distribuição das avaliações em relação às notas foi a seguinte:

- 5 estrelas - 47329 avaliações
- 4 estrelas - 31987 avaliações
- 3 estrelas - 16010 avaliações
- 2 estrelas - 8126 avaliações
- 1 estrela - 25646 avaliações

A partir disso é possível observar que as avaliações não estão distribuídas uniformemente entre as classes (Figura 4a). Para evitar que esse desbalanceamento causasse problemas nos experimentos, uma subamostragem dos dados foi realizada, removendo aleatoriamente amostras das classes com mais avaliações. Desse modo, na base usada nos experimentos, todas as classes ficaram com um total de 8126 avaliações (correspondente ao tamanho da menor classe).

Outro ponto analisado foi o tamanho do vocabulário e dos textos das avaliações. O vocabulário completo possui pouco mais de 11 mil palavras (*tokens*), no entanto, os experimentos preliminares mostraram que um vocabulário menor apresentou melhores resultados, assim, o tamanho do vocabulário foi limitado a 600 palavras. O tamanho dos lotes (*batches*) utilizados na etapa de tokenização foi fixado em 64, pois como se observa no histograma da Figura 4b, poucas avaliações possuem mais que 60 palavras.



(a) Número de amostras por classe

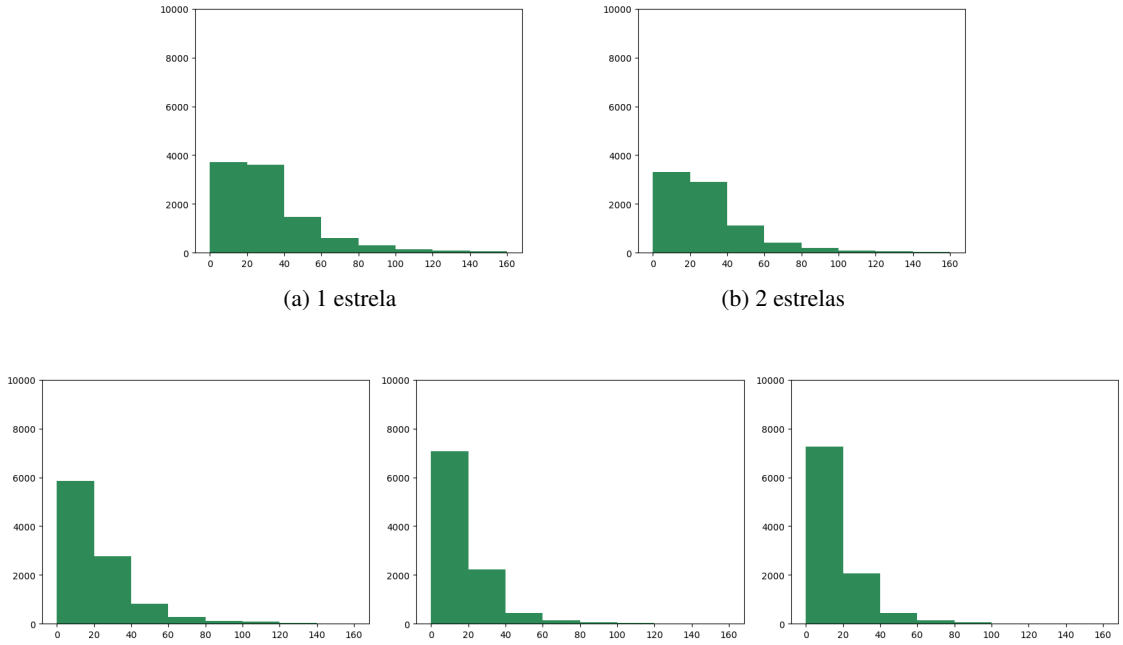
(b) Número de palavras por avaliação

Figura 4: Do lado esquerdo, um gráfico da distribuição das avaliações por classe (nota da avaliação) antes do balanceamento. Do lado direito, o histograma do número de palavras por amostra (base balanceada).

A distribuição do comprimento do texto das avaliações de acordo com a nota dada pelo consumidor também foi avaliada com o objetivo de verificar se há correlação entre tamanho do texto e a nota dada ao produto. Na Figura 5, pode-se verificar que as avaliações que receberam notas maiores (3 a 5 estrelas) possuem um número menor de palavras (a maioria delas, menos de 20 palavras). Por outro lado, dentre as avaliações que receberam notas menores (1 ou 2 estrelas), é possível observar um número maior de avaliações com comprimento entre 20 e 60 palavras, o que pode indicar que os consumidores menos satisfeitos tendem a escrever avaliações maiores sobre os produtos.

4 Experimentos

O objetivo principal dos experimentos desse trabalho é avaliar o comportamento de redes LSTM e GRU, unidirecionais e bidirecionais, na classificação de textos, juntamente com a utilização de diferentes valores de *dropout* e a sua influência sobre o modelo.



(a) 1 estrela (b) 2 estrelas

(c) 3 estrelas (d) 4 estrelas (e) 5 estrelas

Figura 5: Histograma da distribuição do número de palavras por avaliação, de acordo com a nota da avaliação (1 a 5 estrelas).

4.1 Definição da topologia da rede

Para realizar esta avaliação, alguns experimentos preliminares foram feitos inicialmente com a finalidade de auxiliar na definição da topologia da rede neural. A topologia inicial foi definida com duas camadas LSTM bidirecionais, sendo a primeira com 128 e a segunda com 64 neurônios[5], seguida pela camada de *dropout* (com valor 0.25) e uma camada final densa, com 5 neurônios e uma função de ativação *softmax* para classificar as amostras (Figura 6).

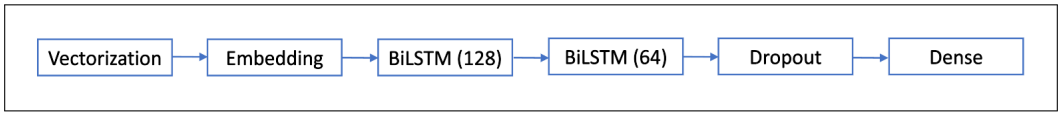


Figura 6: Topologia inicial da rede neural utilizada nos experimentos.

O desempenho da rede foi avaliado com base na acurácia e na função de *loss* `sparse_categorigcal_crossentropy`. Um otimizador Adam, com `learning_rate=0.001` e `clipvalue=1` foi utilizado para o treinamento. A proporção de dados utilizados para treinamento, validação e teste foram, respectivamente, 75%, 10% e 15%. Os experimentos também demonstraram que um número menor do que 20 épocas é suficiente para treinar o modelo. Para reduzir o tempo de treinamento e prevenir *overfitting*, um método de *Early Stopping* foi acrescentado ao processo.

Nessa configuração inicial, o modelo conseguiu obter uma acurácia razoável, 52%, que é um valor duas vezes maior comparado ao *baseline* de 20% (valor esperado de um modelo que retorna respostas aleatórias).

Adicionalmente, foram realizados experimentos alterando parâmetros da rede inicial. A taxa de aprendizado do otimizador foi alterada para valores maiores (0.005 e 0.01), no entanto, isso resultou em um subajustamento (*underfitting*) do modelo. Outros otimizadores como SGD e RMSprop foram testados, mas como não houve melhoras na acurácia, o otimizador Adam foi mantido.

[5]Arquiteturas semelhantes são frequentemente utilizadas em tarefas de classificação de texto. A página `https://neptune.ai/blog/text-classification-tips-and-tricks-kaggle-competitions` apresenta uma série de exemplos dessa utilização em competições na plataforma Kaggle.

Visto que nenhuma das alterações provocou um aumento significativo na acurácia, a topologia inicial foi mantida para os demais testes, alterando apenas a camada de *dropout*, e posteriormente, modificando as camadas bidirecionais para utilizarem GRUs. Por fim, também foram realizados testes com redes unidirecionais LSTM e GRU para verificar se o desempenho seria semelhante em redes mais simples.

4.2 Avaliação da camada de *dropout*

A Tabela 1 mostra os resultados obtidos em **uma** das execuções dos experimentos com redes bidirecionais. O método de *Early Stopping* manteve o número de épocas menor que 10. O melhor resultado nesta execução foi obtido pelo modelo LSTM com um *dropout* de 0.25, mas é possível observar que, apesar da rede LSTM ter tido maior acurácia que a rede GRU, os valores não variaram muito.

Tabela 1: Resultados de **uma** execução dos experimentos com redes bidirecionais na base balanceada.

Modelo + Dropout	Acc Treino	Acc Teste	Epochs
BiLSTM + 0.0	0.63	0.52	9
BiLSTM + 0.25	0.62	**0.53**	**6**
BiLSTM + 0.5	0.62	0.52	7
BiGRU + 0.0	0.62	0.50	8
BiGRU + 0.25	0.63	**0.52**	**6**
BiGRU + 0.5	0.65	0.51	8

Apesar de ser esperado que a utilização de *dropout* reduzisse o *overfitting* [Srivastava et al., 2014], esse comportamento não foi observado nos experimentos. A mudança dos valores de *dropout* também não provocou grandes mudanças na acurácia, no entanto, em ambos os modelos, o valor de *dropout* de 0.25 foi o que o obteve o melhor resultado em um número menor de épocas de treinamento. A Figura 7 mostra uma comparação do histórico das execuções.

Deve-se ressaltar que entre as execuções, o resultado dos experimentos variaram – a acurácia de teste variou entre 0.51 e 0.53 e o número de épocas entre 6 e 13, para todos os modelos. Portanto, não foi possível concluir que algum dos modelos foi consistentemente melhor ou pior que os outros.

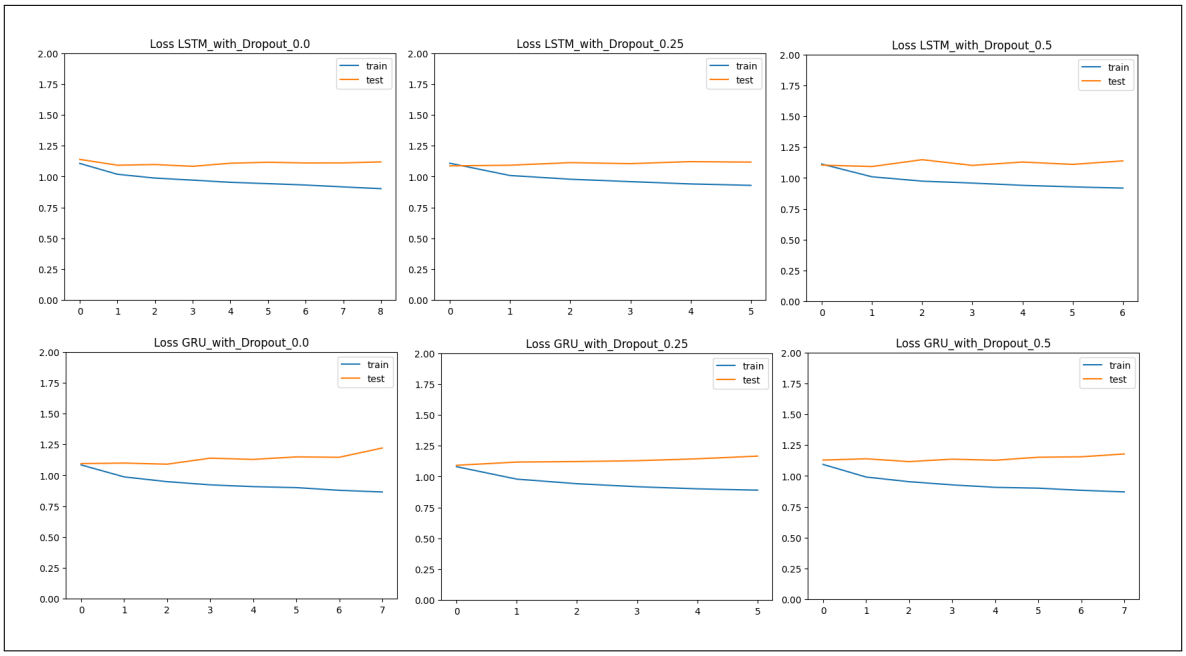


Figura 7: Avaliação da função de *loss* nos experimentos com redes bidirecionais.

4.3 Avaliação das redes unidirecionais

A Tabela 2 apresenta os resultados de uma execução dos experimentos com redes unidirecionais. Nota-se que, na média, o número de épocas de treinamento foi levemente maior do que nos experimentos com redes bidirecionais. Apesar disso, os valores da acurácia permaneceram praticamente iguais.

Tabela 2: Resultados de **uma** execução dos experimentos com redes unidirecionais LSTM e GRU.

Modelo + Dropout	Acc Treino	Acc Teste	Epochs
LSTM + 0.0	0.58	0.53	14
LSTM + 0.25	0.57	**0.53**	**10**
LSTM + 0.5	0.59	0.52	9
GRU + 0.0	0.60	0.52	10
GRU + 0.25	0.62	0.52	12
GRU + 0.5	0.61	0.52	9

Na Figura 8 são apresentados os gráficos das execuções das redes unidirecionais onde pode-se observar um comportamento muito semelhante ao comportamento das redes bidirecionais. Os modelos atingiram o melhor desempenho logo nas épocas inicias e começaram a sofrer *overfitting*. Assim como nas redes bidirecionais, os valores diferentes de *dropout* não causaram impacto na acurácia dos modelos.

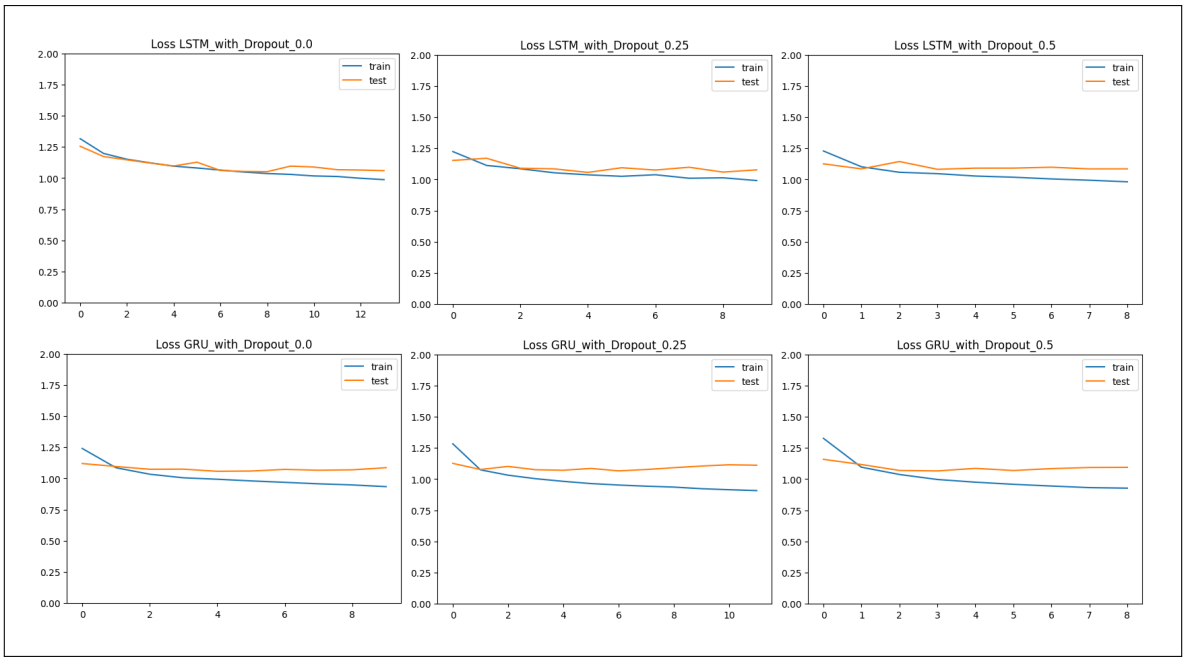


Figura 8: Avaliação da função de *loss* nos experimentos com redes unidirecionais.

4.4 Resultados com a base completa e desbalanceada

Adicionalmente, foram realizados experimentos com a base completa e desbalanceada com o fim de verificar o quanto o balanceamento da base influenciou nos resultados. Como se observa na Tabela 3, a acurácia na partição de testes foi de 0.59 em todos os modelos e valores de *dropout*. Nota-se que os valores de *dropout* influenciaram no número de épocas de treinamento, assim como ocorreu nos experimentos anteriores.

A acurácia maior neste caso pode estar relacionada a dois fatores: (1) o aumento da quantidade de dados, que fornece mais informações para o treinamento modelo (a base balanceada possui cerca de 40 mil amostras, enquanto a

Tabela 3: Resultados de **uma** execução dos com redes bidirecionais na base desbalanceada.

Modelo + Dropout	Acc Treino	Acc Teste	Epochs
BiLSTM + 0.0	0.67	0.59	9
BiLSTM + 0.25	0.67	0.59	8
BiLSTM + 0.5	0.66	0.59	7
BiGRU + 0.0	0.68	0.59	8
BiGRU + 0.25	0.67	0.59	7
BiGRU + 0.5	0.67	0.59	7

desbalanceada possui em torno de 129 mil); (2) um *baseline* naturalmente maior, pois neste caso, 37% das amostras estão na classe com rótulo 5, e o modelo tem mais chances de acertar caso classifique aleatoriamente uma amostra com esse rótulo.

4.5 Avaliação sem *early stopping*

Como uma tentativa de melhorar a acurácia do modelo, foram realizados testes com o modelo LSTM, uni e bidirecional, com *dropout* de 0.25, utilizando 60 épocas sem *early stopping*. A Figura 9 mostra os resultados desse teste. Apesar do modelo ter tido um desempenho melhor sobre a partição de treinamento, o modelo sofreu *overfitting* nas primeiras épocas de treinamento e a acurácia na partição de testes piorou, ficando abaixo de 0.50 na base balanceada. Como recurso adicional, foram adicionados parâmetros de regularização nas camadas de LSTM, o que produziu uma redução no *overfitting*, mas não foi suficiente para melhorar a acurácia do modelo.

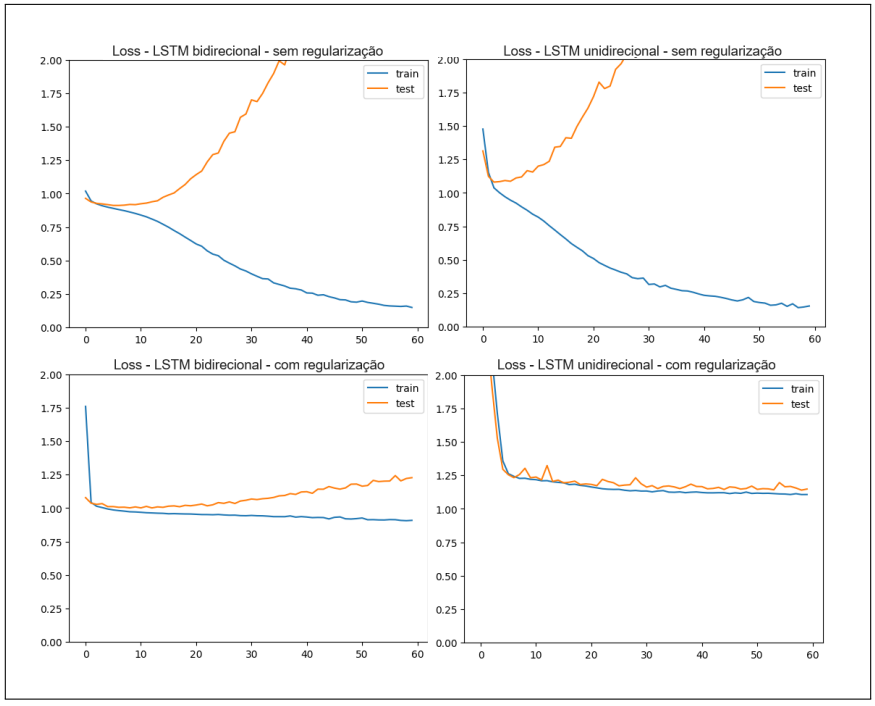


Figura 9: Avaliação do modelo sem *early stopping*. Na primeira linha estão os resultados do modelo original (uni e bidirecional), e na segunda linha os resultados com adição de parâmetros de regularização nas camadas de LSTM.

5 Conclusão

A partir dos resultados obtidos, concluiu-se que os modelos LSTM e GRU obtiveram um bom resultado na tarefa de classificação de texto, em comparação ao *baseline* original. A camada de *dropout* não influenciou na melhoria da acurácia, mas ajudou a reduzir o número de épocas de treinamento em algumas execuções. Também não foram observadas diferenças significativas na comparação entre redes unidirecionais e bidirecionais.

Acredita-se que limite de acurácia das redes LSTM e GRU (nessa base de dados) foi atingido, pois as alterações nos diversos parâmetros (número de épocas, otimizadores, taxa de aprendizado, *dropout*, etc.) não foram provocaram melhoria no desempenho dos modelos na partição de testes.

Para obter melhores resultados nesse cenário, uma possível abordagem seria a utilização de outros modelos, mais sofisticados. Essa avaliação não foi realizada devido ao escopo reduzido do trabalho, mas pode ser feita em trabalhos futuros.

Referências

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi:10.1038/nature14539.

Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655, 2021. ISSN 2212-8271. doi:https://doi.org/10.1016/j.procir.2021.03.088. URL `https://www.sciencedirect.com/science/article/pii/S2212827121003796`. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

S Hochreiter and J Schmidhuber. Long short-term memory. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 1735–1780. PubMed, 1997. doi:10.1162/neco.1997.9.8.1735.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45 (11):2673–2681, 1997. doi:10.1109/78.650093.

Dinesh Naik and C. D. Jaidhar. A novel multi-layer attention framework for visual description prediction using bidirectional lstm. *Journal of Big Data*, 2022.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.