

Desempenho de Modelos de Redes Neurais Recorrentes para Análise de Sentimentos *

Cristiano A. Künas¹, Leandro P. Heck¹, Edson L. Padoin¹

¹Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI)
Santa Rosa – RS – Brasil

{cristiano.kunas, leandro.h}@sou.unijui.edu.br, padoin@unijui.edu.br

Resumo. Cada vez mais usuários expressam seus sentimentos em forma de opinião sobre diversos assuntos. Analisar os sentimentos expostos pode auxiliar no entendimento do que as pessoas estão falando sobre uma determinada empresa, marca, evento ou até mesmo sobre outras pessoas, funcionando como uma forma de se obter um feedback. Este artigo apresenta uma comparação de acurácia e tempo de execução de modelos de Recurrent Neural Network baseados em LSTM e BiLSTM para aplicação de análise de sentimento. Com a aplicação da RNN desenvolvida sobre uma base de dados pública com 50.000 registros de filmes foi possível aumentar a acurácia da RNN para 89,34% e reduzir o tempo de treinamento em até 7,22 vezes.

1. Introdução

As opiniões exercem um papel fundamental em quase todas as atividades humanas, pois são indicadores de comportamento. Sempre que precisamos tomar uma decisão, queremos ouvir as opiniões dos outros [Liu et al. 2010]. Nos últimos anos, mais e mais usuários começam a compartilhar livremente seus sentimentos na forma de opiniões sobre uma ampla variedade de tópicos, o que resulta em uma enorme quantidade de dados. Entretanto, analisar estes dados e filtrar as informações relevantes contidas nestes é uma tarefa difícil devido à rápida disseminação de textos *online*. Deste modo, sistemas inteligentes capazes de automatizar o processo de análise de sentimento são fundamentais.

Para a análise de sentimento, várias abordagens baseadas em processamento de linguagem natural e aprendizado de máquina foram usadas no passado. No entanto, os métodos baseados em aprendizagem profunda estão se tornando muito populares devido ao seu desempenho nos últimos tempos [Yadav and Vishwakarma 2020]. No que diz respeito ao domínio da aplicação, *Recurrent Neural Networks* (RNNs) se adaptam melhor a problemas de modelagem, tais como dados e análise de textos.

Existem diversas variações de modelos de RNN. A maioria destes modelos diferem principalmente devido às suas capacidades de memorização dos dados de entrada. Em geral, um modelo básico de RNN não tem a capacidade de lembrar os dados do passado. Modelos baseados em *Long Short-Term Memory* (LSTM) e *Bidirectional Long Short-Term Memory* (BiLSTM) são um tipo especial de RNN capaz de aprender a partir de dados passados. Sendo assim, este artigo apresenta uma comparação de acurácia e tempo de treinamento de modelos de RNN baseada em LSTM e BiLSTM para análise da polaridade (positivo, negativo) do sentimento de sentenças. O restante do trabalho esta organizado da seguinte forma. A Seção 2 discute

*Trabalho desenvolvido com recursos do edital MCTIC/CNPq - Universal 28/2018 sob número 436339/2018-8 e do edital da VRPGPE bolsa PIBIC/UNIJUI.

os trabalhos relacionados. A Seção 3 descreve a metodologia aplicada na implementação e o ambiente de execução. Na Seção 4 são discutidos os resultados, seguidos das Conclusões e Trabalhos Futuros.

2. Trabalhos Relacionados

O trabalho de Siami-Namini *et al.* (2019) relata uma análise e comparação comportamental dos modelos BiLSTM e LSTM. O objetivo é explorar se a incorporação de camadas adicionais de formação na arquitetura de um LSTM melhora a sua previsão. Os resultados mostram que a utilização de camadas adicionais de formação ajudaria a melhorar a precisão em 37,78% em média, portanto, a modelação baseada em BiLSTM oferece melhores previsões do que os modelos regulares baseados em LSTM. Observou-se também que os modelos BiLSTM atingem o equilíbrio muito mais lentamente do que os modelos baseados em LSTM.

Kim e Moon (2019) propuseram um modelo de LSTM bidirecional com base em dados de séries temporais multivariadas. Aplicaram o modelo proposto nas previsões da área de comércio. Os experimentos e avaliação de desempenho foram realizados com base no *Root Mean Square Error*. Os autores relatam que resultados experimentais mostram que o modelo BiLSTM supera o desempenho da LSTM unidirecional. No entanto, este estudo não considera a velocidade de aprendizagem.

Cui *et al.* (2018) propuseram o empilhamento de redes LSTM bidirecionais e unidirecionais para prever a velocidade do tráfego em toda a rede. Eles relatam que a arquitetura empilhada supera tanto a BiLSTM como a LSTM.

Estes trabalhos contemplam importantes contribuições na implementação de modelos LSTM unidirecionais e bidirecionais. Seguindo a mesma linha de raciocínio, o presente trabalho realiza uma comparação de acurácia e tempo de treinamento entre modelos baseados em LSTM e BiLSTM aplicados na análise de sentimentos.

3. Implementação da RNN

A implementação da RNN se dá a partir da linguagem de programação *Python*. Os principais módulos utilizados são *Keras*, *Tensorflow* e *Scikit-Learn*. A base de dados usada nos experimentos foi elaborada no trabalho de Maas *et al.* (2011), e possui uma coleção de 50.000 avaliações do IMDb escritas no idioma inglês. As críticas estão balanceadas em 2 classes: positiva e negativa. O pré-processamento destes dados é feito em etapas com objetivo de remover ruído presente nas sentenças. São elas: i) caracteres especiais e sinais de pontuação são removidos; ii) todas as palavras são escritas em letras minúsculas; iii) as *stopwords* são removidas; iv) as sequências são limitadas a um tamanho fixo (300 palavras). O *dataset* é dividido em treino (80%) e teste (20%).

A criação do modelo de RNN é definido como uma sequência de camadas. Implementamos dois modelos, o primeiro com uma camada LSTM (entrada da esquerda para a direita) e o segundo com uma camada BiLSTM (entrada da esquerda para a direita e da direita para a esquerda). A classe *Model* da API funcional da biblioteca *Keras* é utilizada no desenvolvimento do modelo de dados. A compilação do modelo configura o processo de aprendizado. Ele define o otimizador (*adam*), a função de perda (*binary_crossentropy*) e as métricas (*accuracy*). Para treinar o modelo, os dados são divididos em lotes (*batch_size*), iterando repetidamente em todo o conjunto de dados para um determinado número de épocas. O modelo é avaliado por meio

da função *evaluate()*. Essa função gerará uma predição para cada entrada e saída e coletará pontuação, realiza uma média sobre o valor de perda e acurácia.

O ambiente de execução é composto de um equipamento com um processador *Intel Core i7-9750H* com 6 *cores* (12 *threads*) de 2.60 GHz de frequência. Este equipamento possui 16 GB de Memória RAM DDR4, GPU NVIDIA GeForce RTX 2060 com 6 GB de memória GDDR6 e 1920 CUDA *cores*, utilizou-se do sistema operacional *Linux Ubuntu 18.04.5 LTS* com versão de *kernel 5.4.0-58-generic*. A versão do NVIDIA CUDA *Compiler* utilizada foi a 10.1.243.

4. Resultados

Na Figura 1 é apresentada a acurácia (eixo y) obtida, variando-se o *batch_size* nos modelos com LSTM e BiLSTM executando em GPU. A acurácia informa o percentual que nosso modelo acertou na predição em relação ao rótulo. Estes testes comprovam que a acurácia dos dois modelos implementados se mantém estável, variando entre 87 ~ 89%. Também é possível observar que os modelos BiLSTM são mais lentos para atingir o equilíbrio. A melhor taxa de acurácia para o modelo com LSTM unidirecional é obtida utilizando *batch_size=32*, que representa 88,32%. Já para o modelo com LSTM Bidirecional, a melhor acurácia é de 89,34%, obtida para *batch_size=2048*.

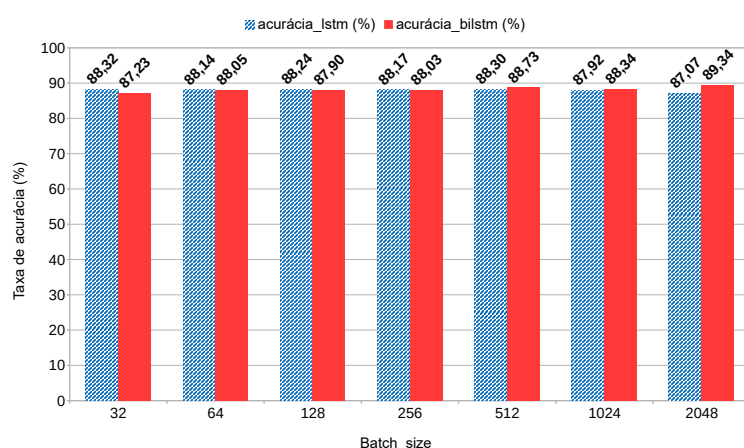


Figura 1. Métricas de acurácia dos modelos LSTM e BiLSTM.

A Figura 2 apresenta os tempos de treinamento conforme o *batch_size* definido. Percebe-se que conforme o tamanho do *batch* aumenta, o tempo de execução é reduzido consideravelmente em ambos modelos. As execuções do modelo baseado em LSTM unidirecional apresenta um ganho de 6,06 vezes, reduzindo o tempo de execução de 970 segundos para 160 segundos. Para o modelo LSTM Bidirecional o ganho chegou a 7,22 vezes, partindo de 1799 segundos para 249 segundos. O custo computacional com a transferência dos dados de treinamento é perceptível ao usar lotes de tamanho menor e, como consequência, não é possível utilizar toda a capacidade que o ambiente de execução oferece.

5. Conclusões e trabalhos futuros

Este trabalho abordou o uso de RNN LSTM e BiLSTM para análise de sentimentos em sentenças. No que tange a taxa de acurácia, no modelo baseado em LSTM, a melhor taxa é registrada para *batch* de tamanho 32 (88,32%). E para modelo BiLSTM, a melhor taxa é obtida

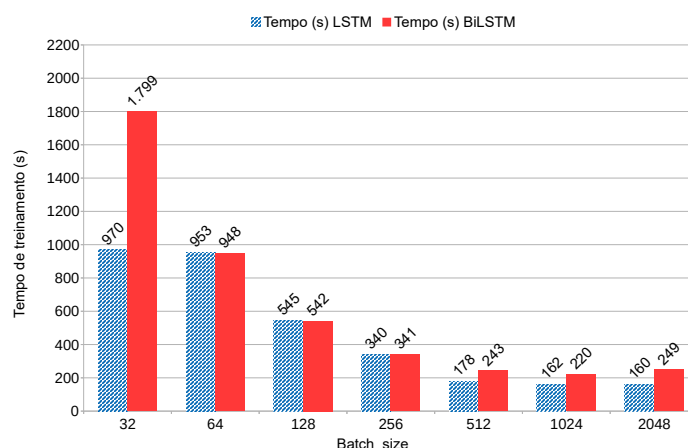


Figura 2. Tempos de treinamento dos modelos LSTM e BiLSTM.

para *batch* de tamanho 2048 (89,34%). O tempo de execução é reduzido consideravelmente em ambos os modelos ao aumentar o tamanho do *batch*. O tempo de execução do modelo com LSTM apresentou ganho de 6,06 vezes. Já para o modelo com BiLSTM, o ganho é de 7,22 vezes.

Como trabalhos futuros, se estuda a utilização de *word embeddings* pré-treinados. Além disso, estamos elaborando um novo *dataset* no idioma português, utilizando o *twitter* como fonte de dados. Outra iniciativa é a possibilidade de adicionar outras classes de classificação além de positivo e negativo, como neutro, muito positivo e muito negativo.

Referências

- Cui, Z., Ke, R., and Wang, Y. (2017). Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *The 6th International Workshop on Urban Computing (UrbComp 2017)*.
- Kim, J. and Moon, N. (2019). Bilstm model based on multivariate time series data in multiple field for forecasting trading area. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–10.
- Liu, B. et al. (2010). Sentiment analysis and subjectivity. In *Handbook of natural language processing*, volume 2, pages 627–666. CRC Press.
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Siarni-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. IEEE.
- Yadav, A. and Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385.