

Convolutional Neural Networks and Long Short-Term Memory Networks for Textual Classification of Information Access Requests

E. S. Paiva, A. B. Paim and N. F. F. Ebecken

Abstract—The e-sic system aims to centralize requests for access to information addressed to the Brazilian Federal Executive. However, the volume of requests received can be an impediment to responses to those requests. The purpose of this article is to create an automatic classifier for these requests. For that, they were analyzed as architectures of the Convolutional Neural Network (CNN) and of Long Short Term Memory (LSTM), as well as a combination of these two architectures in order to identify the best architectures to this problem. The metrics used to evaluate the results were the area under curve roc and accuracy, and the error function used was cross entropy. The study concluded that the CNN network performed the best. Thus, the main contribution of this article is the identification of the most appropriate network architecture for classifying texts of interaction between citizens and government written in Portuguese.

Index Terms—Convolution Neural Network, Long Short-Term Memory, Government Data, Text Classification.

I. INTRODUÇÃO

Um regime democrático pressupõe transparência por parte dos governantes e dos órgãos governamentais. A transparência e a participação popular são princípios basilares da gestão democrática, que permitem aos cidadãos se informarem sobre a agenda proposta pelo governo e participarem das decisões sobre assuntos públicos e relacionados a seus legítimos interesses particulares [5]. Visando assegurar essa transparência, a Constituição Federal de 1988 previu no inciso XXXIII do artigo 5º que todo cidadão brasileiro teria direito a receber dos órgãos públicos informações de seu interesse particular ou de interesse geral ou coletivo. Dessa forma, em novembro de 2011, foi publicada a Lei Federal nº 12.527/11 ou Lei de Acesso à Informação (LAI), que tem o objetivo de regular o referido dispositivo constitucional, definindo assim a forma como o cidadão pode ter acesso a essas informações. A LAI define como as solicitações de informações devem ser feitas, bem como as condições e prazos para as respostas dessas solicitações.

Nesse contexto, foi criado o Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC). Esse sistema tem o

objetivo de centralizar as entradas e saídas de todos os pedidos de acesso à informação dirigidos ao Poder Executivo Federal. Sendo assim, o e-SIC organiza e facilita os procedimentos de acesso tanto para os cidadãos quanto para a Administração Pública. Através do e-SIC, qualquer pessoa – seja física ou jurídica – pode encaminhar pedidos de acesso à informação para órgãos e entidades do Poder Executivo Federal. Por meio do sistema também é possível consultar as respostas recebidas; entrar com recursos; apresentar reclamações; entre outras ações.

No entanto, o volume de solicitações recebidas pode ser um impeditivo para que as respostas de tais pedidos sejam feitas de forma tempestiva, visto que, mensalmente são recebidos mais de 10.000 pedidos de informações, com diferentes características. De acordo com suas características, os pedidos podem ser encaminhados para uma área ou outra, a fim de identificar a resposta que está sendo solicitada. Sendo assim, os pedidos de informações podem ser enquadrados em 17 classes distintas.

A existência de um sistema capaz de fazer a leitura das descrições textuais dos pedidos de informação e classificá-los de forma automatizada dentro das classes a que eles se referem traria grande agilidade ao processo como um todo.

Sendo assim, o objetivo desse trabalho é criar um classificador automático das solicitações de acesso à informação, de forma que a partir do texto da solicitação, já se saiba a que classe a referida solicitação pertence, sem a necessidade de interação humana.

Atualmente, têm-se obtido bons resultados com a utilização de redes neurais profundas para o processamento de linguagem natural, pois, a melhoria da capacidade de processamento (principalmente com a possibilidade de processamento paralelo por meio da utilização de GPUs) aliada a grande quantidade de dados disponíveis para o processo de treinamento, tem possibilitado a utilização de redes neurais mais complexas, o que traz uma melhora considerável nos resultados obtidos.

Nesse contexto, conforme citado por [9], duas arquiteturas de redes neurais profundas (*deep learning*) têm se destacado, as *Convolutional Neural Network-CNN* [18] e as *Recurrents Neural Network-RNN* [16], sendo que, algumas limitações das RNNs convencionais propiciaram o surgimento de duas outras subcategorias de RNNs: as *Long Short-Term Memory-LSTM* [19] e *Gated Recurrent Unit – GRU* [14].

Sendo assim, esse trabalho analisa as arquiteturas de redes

Este trabalho foi submetido para revisão em 20/04/2020

E. S. Paiva, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brasil, eduardo.paiva@coc.ufrj.br

A. B. Paim, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brasil, andrea.paim@coc.ufrj.br.

N. F. F. Ebecken, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brasil, nelson@ntt.ufrj.br.

CNN e LSTM, assim como uma combinação dessas duas arquiteturas, para identificar aquela que é mais apropriada para o problema em questão. O artigo também avalia o comportamento das soluções propostas em ambientes de processamento em paralelo com GPU e em ambiente apenas com CPU. Tais avaliações foram executadas utilizando como métricas de comparação a área sob a curva ROC e a acurácia, e a função de erro utilizada foi a entropia cruzada.

Dessa forma, as principais contribuições desse artigo são a identificação da arquitetura de rede mais apropriada para classificação de textos de interação entre cidadãos e governo escritos em português, além das justificativas da diferença de comportamento das arquiteturas testadas nos diferentes ambientes analisados.

O restante desse estudo está organizado da seguinte forma: a Seção II apresenta alguns trabalhos relacionados; já a Seção III define o problema a ser tratado e a seção IV descreve os experimentos realizados. Por fim, a Seção V traz a conclusão do artigo.

II. TRABALHOS RELACIONADOS

Atualmente, as redes neurais profundas têm sido uma das principais técnicas utilizadas no processamento de linguagem natural, sendo que, as CNNs e as RNNs têm apresentado excelentes resultados nessas tarefas.

Nesse sentido, [10] aplicou redes CNNs em diversas tarefas de classificação de sentenças. [10] explica que os núcleos convolucionais inicializados aleatoriamente se tornam detectores específicos de recursos de n-grama úteis para tarefas de classificação.

[3] tentaram tratar o problema da dependência de longa duração, presente em textos mais longos, com a utilização de uma rede neural convolucional dinâmica (DCNN) para modelagem semântica de sentenças. Essa rede usava o *Dynamic k-Max Pooling*, uma operação de *pooling* global sobre sequências lineares. Isso fazia com que a rede conseguisse lidar com frases de comprimentos variados, tornando-a capaz de capturar tanto relações próximas, quanto distantes em uma determinada sentença.

Os autores testaram a DCNN em quatro experimentos: previsão de sentimentos binários e multiclasse em pequena escala, classificação de perguntas e previsão de sentimentos no Twitter por supervisão distante, obtendo resultados satisfatórios.

[21] propuseram o uso da CNN para modelar representações de textos curtos, que sofrem com a falta de contexto disponível e, portanto, exigem esforços extras para criar representações significativas. Os autores propuseram um agrupamento semântico, que introduzia unidades semânticas em várias escalas para serem usadas como conhecimento externo para os textos curtos. Nesse caso, a CNN foi usada para combinar essas unidades e formar a representação geral.

[4] propuseram a detecção de sarcasmo em textos do Twitter usando uma rede CNN. Os autores utilizaram redes pré-treinadas, treinadas em conjuntos de dados de emoções, sentimentos e personalidade, e conseguiram obter um desempenho satisfatório.

[1] combinaram as arquiteturas LSTM e CNN e propuseram um modelo unificado, chamado C-LSTM, para representação de sentenças e classificação de texto. A arquitetura proposta utiliza rede CNN para extrair uma sequência de representações de frase de nível superior e alimenta uma rede LSTM para obter a representação das sentenças. O C-LSTM é capaz de capturar características locais, assim como a semântica global e temporal das frases. Essa proposta foi avaliada em tarefas de classificação de sentimentos e classificação de perguntas, sendo que os resultados superaram os obtidos pela aplicação isolada de redes CNNs e LSTMs.

[6] analisaram a CNN e a RNN básica para tarefas de classificação. Os autores relatam maior desempenho da CNN em relação a RNN, mas, eles indicam que as CNNs e as RNNs fornecem informações complementares. Segundo os autores, enquanto as RNNs calculam uma combinação ponderada de todas as palavras da sentença, as CNNs extraem os ngrams mais informativos e consideram apenas as ativações resultantes.

Considerando que ambas as arquiteturas apresentam resultados satisfatórios na atividade de processamento de linguagem natural, [9] se propôs a fazer um estudo comparativo entre as redes CNN e RNN para esse tipo de tarefa. O objetivo do estudo era fornecer orientações básicas para a seleção da rede a ser utilizada.

Segundo os autores, a arquitetura com melhor desempenho depende da importância de entender semanticamente toda a sequência. [9] concluem que as RNNs apresentam bom desempenho em uma ampla gama de tarefas, exceto quando a tarefa é essencialmente uma tarefa de reconhecimento de frase-chave, como em algumas configurações de detecção de sentimentos e correspondência de perguntas e respostas.

Outra conclusão desse artigo foi que a escolha da quantidade de camadas ocultas e o tamanho do “*batch*” podem ter grande influência no desempenho da rede. Isso sugere que a otimização desses dois parâmetros é crucial para o bom desempenho das CNNs e RNNs.

Logo, pode-se concluir que não há de se falar em uma arquitetura mais apropriada para todas as tarefas de processamento de linguagem natural, sendo necessários estudos específicos para cada tipo de caso.

Diante disso, alguns outros trabalhos também se propuseram a realizar tal tipo de comparação, sendo que em cenários mais específicos. [7] avaliaram as RNNs e as CNNs na classificação de textos em relatórios de radiologia. O trabalho utilizou essas arquiteturas para sintetizar informações sobre embolia pulmonar em mais de 7370 relatórios clínicos de radiologia coletados em quatro grandes centros de saúde dos Estados Unidos da América.

Os autores concluem que os resultados apontam para a viabilidade de CNNs e RNNs na classificação automatizada dos relatórios textuais e sugerem a aplicação dessas técnicas em vários casos de uso tais como: priorização de pacientes em radiologia, geração de corte para pesquisa clínica, triagem de elegibilidade para ensaios clínicos e avaliação da utilização de imagens.

Seguindo essa linha, [8] e [2] também realizam estudos

comparativos entre as redes CNNs e LSTMs para tarefas específicas de processamento de linguagem natural. [8] comparam o desempenho dessas arquiteturas em problemas ligados a pesquisas de opinião. [2] avalia a performance dessas redes no contexto de textos jurídicos.

Como pode ser visto, não existe uma arquitetura ideal para todos os tipos de textos e tarefas, sendo que, a arquitetura mais apropriada varia de acordo com as características dos dados e do problema a ser resolvido. Nesse sentido, não foi identificado na literatura trabalhos que fizessem esse tipo de comparação para textos na língua portuguesa, e mais especificamente para textos ligados a atividades de interação entre cidadãos e governo. Sendo assim, a principal contribuição desse trabalho é a identificação da arquitetura mais apropriada para classificação de textos de interação entre cidadãos e governo escritos em português. No entanto, cabe ressaltar que esse trabalho também avalia o desempenho dessas arquiteturas, para o problema em questão, em ambientes de processamento paralelo com GPU e em ambiente apenas com o processamento com CPU.

III. DEFINIÇÃO DO PROBLEMA

Quando um cidadão pretende solicitar alguma informação do governo federal, ele deve entrar no site do Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC), fazer um cadastro (caso não tenha), redigir a sua solicitação e escolher o órgão de destino. A partir daí, ele recebe um número de protocolo para acompanhar o andamento desse pedido.

Diariamente, o e-SIC recebe centenas de descrições de pedidos de informações. Esses pedidos devem passar por uma triagem, de acordo com o seu conteúdo, a fim de serem encaminhados para o setor que tenha condições de respondê-los de forma apropriada.

Tais pedidos podem ser enquadrados em 17 categorias, de acordo com o seu conteúdo. A Fig. 1 apresenta 2 exemplos de solicitações de informações enquadrados na categoria “Habitação, Saneamento e Urbanismo”.

①

Desejo saber qual o endereço de agente financeiro (imobiliária) da CEF, para fins de tratamento de financiamento habitacional, no bairro de Campo Grande/Rio de Janeiro/RJ. Ou seria na própria agência do bairro as tratativas sobre financiamento habitacional?

②

Qual o percentual de financiamento do PMCMV (em todas as faixas) realizado pela CEF, BB e demais bancos?

Fig. 1. Exemplos de Solicitações de Informações da Categoria Habitação Saneamento e Urbanismo.

Sendo assim, o problema a ser tratado consiste no processamento dos textos das descrições de pedidos de informação e na identificação da categoria a que cada uma dessas descrições pertence. Logo, o problema em questão pode ser considerado como uma questão de classificação textual e é definido da seguinte forma:

Dada uma coleção de Solicitações de Informações $S = \{s_1, s_2, s_3, \dots, s_k\}$ e um conjunto de classes $C = \{c_1, c_2, c_3, \dots, c_n\}$. Deseja-se definir uma função $F: S \times C \rightarrow \{0, 1\}$, ou seja, uma função F que atribui um valor 0 ou 1 para cada par (s_i, c_j) , tal que $s_i \in S$ e $c_j \in C$. Se o valor atribuído for 1, diz-se que a solicitação de informação s_i pertence à classe c_j , caso contrário, diz-se que a solicitação de informação s_i não se refere a classe c_j . Cabe ressaltar que uma solicitação de informação só pode pertencer a uma única classe.

IV. EXPERIMENTOS

Essa sessão descreve os experimentos realizados, os resultados obtidos, e as análises desses resultados. Durante a execução dos experimentos, utilizou-se validação cruzada (com 5 iterações), considerando-se sempre 80% dos dados para o treinamento e 20% para o teste, sendo que, cada tipo de rede foi testado tanto em configurações com a utilização de GPUs, quanto em configurações que utilizavam apenas o processamento com CPUs. A Fig. 4 ilustra a rede CNN utilizada, e a Fig. 5 demonstra a estrutura da rede LSTM. A configuração da rede CNN com LSTM é apresentada na Fig. 6. A implementação das redes neurais foi realizada na linguagem python, com o auxílio do framework tensorflow.

Todos os modelos foram gerados com a utilização de 25 épocas de treinamento, porém, colheu-se informações dos resultados preliminares a cada 5 épocas, para se ter uma ideia de quão rápido o treinamento convergiria para o mínimo da função de erro, sendo que, a função de erro utilizada foi a entropia cruzada.

Cabe ressaltar que todas as redes tinham como camada de entrada uma camada de embedding que utilizava uma word embedding obtida a partir do processamento de textos em português [20]. Essa camada de embedding foi inicializada com o CBOW [13], uma variante do modelo word2vec, sendo que, tal word embedding representava cada uma das palavras com vetores de 50 dimensões.

A. Base de Dados

Os dados utilizados nesse projeto estão disponíveis para download no próprio site do e-sic, sendo que, as informações relativas a cada solicitação estão divididas em 21 colunas, que trazem dados como o identificador da solicitação, a situação desse pedido, a data de registro, o prazo de atendimento, dentre outros. No entanto, esse trabalho só utilizou dois campos desse conjunto de dados: o texto da solicitação da informação e a categoria em que essa solicitação se enquadra. Sendo que, o texto da solicitação é preenchido pelo cidadão que está fazendo a solicitação de informação, e a categoria é definida pelo servidor público que está dando tratamento a essa solicitação.

Para a execução dos experimentos, utilizou-se dados de pedidos de acesso à informação referentes ao ano de 2019. Dessa forma, selecionou-se uma amostra aleatória de 1.000 solicitações de informação por classe. Logo, como existem 17 classes distintas, os experimentos foram realizados com um total de 17.000 registros.

Dentre esses 17000 registros, as sentenças tinham características distintas, sendo que a sentença de menor tamanho possuía apenas 1 palavra, e a sentença de tamanho máximo era composta por 351 palavras. No entanto, o tamanho médio das sentenças era de 64,5 palavras. A Fig. 2 apresenta um histograma do número de palavras por sentença.

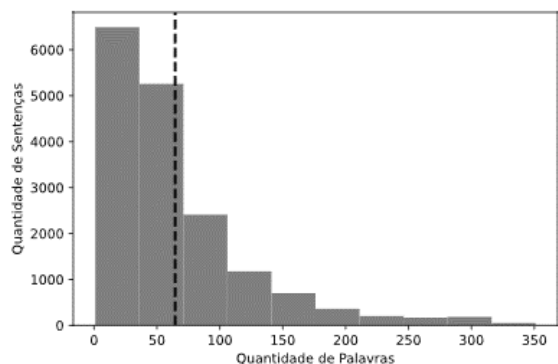


Fig. 2. Histograma da quantidade de palavras por sentença.

A linha tracejada (na Fig. 2) representa o tamanho médio das sentenças (quantidade de palavras). Como pode ser observado, a maioria das sentenças são curtas, e poucas sentenças possuem quantidades maiores de palavras.

B. Métricas de Análise

Para a análise dos resultados, utilizou-se como métrica principal a área sob a curva ROC (ROC-AUC). Essa é uma métrica de classificação que mede a área sob a curva do gráfico entre a taxa de verdadeiros positivos (casos que foram corretamente classificados como sendo de uma determinada classe) e a taxa de falsos positivos (casos que foram erroneamente classificados como sendo de uma determinada classe). Dessa forma, quanto maior o valor dessa métrica, melhor é a performance do modelo, sendo que, o seu valor máximo é 1. Cabe ressaltar que, um modelo que gerasse uma classificação totalmente aleatória obteria um valor de 0,5 para a ROC-AUC. Logo, qualquer valor acima de 0,5 já representa uma classificação melhor que uma escolha aleatória.

Originalmente a ROC-AUC é utilizada como uma métrica para problemas de classificação binária, porém, pode-se estender essa métrica para problemas com várias classes, considerando-se vários classificadores binários.

Adicionalmente, utilizou-se a acurácia como métrica secundária. Ou seja, apesar das análises serem guiadas pela métrica principal, a acurácia é utilizada como um segundo balizador, a fim de possibilitar uma outra visão para os resultados obtidos.

C. Benchmarking

O fato de inexistirem trabalhos anteriores com a base de dados em questão fez com que esse estudo não tivesse um ponto de referência para medir os seus resultados. No entanto, considerando a premissa de que o valor de 0,5 da métrica ROC-AUC representaria uma classificação totalmente

aleatória, qualquer valor acima desse parâmetro já significaria um ganho para o processo atual.

Porém, a fim de se ter um parâmetro um pouco mais realista do que um classificador totalmente aleatório, utilizou-se o resultado de um classificador gerado por uma Máquina de Vetor de Suporte (SVM) como base de comparação para os modelos estudados. O objetivo da utilização do algoritmo SVM nessa parte do trabalho não é fazer nenhum tipo de comparação com os modelos de *deep learning* aqui estudados. Sendo assim, a única função do modelo gerado por esse algoritmo é fornecer uma base de comparação.

Adicionalmente, a fim de se utilizar um parâmetro obtido por um modelo de *deep learning*, utilizou-se a base de dados textuais do sistema ouvidoria do governo federal brasileiro, cujos textos possuem características semelhantes aos do problema tratado nesse trabalho, para classificá-la em 5 classes distintas: Comunicação, Denúncia, Solicitação, Sugestão, Elogio e Reclamação.

Sendo assim, gerou-se duas linhas de base para comparação: uma obtida pelo algoritmo SVM a partir da mesma base de dados aqui estudada, e outra obtida a partir de uma rede CNN, utilizando um conjunto de dados semelhante aos estudados nesse artigo.

O modelo gerado pela SVM, para o problema em questão, gerou um valor de 0,6132 para a ROC-AUC, e acurácia de 43,71%. Já o classificador gerado pela rede CNN, para a base de dados do sistema de ouvidoria, obteve 0,7804 para a ROC-AUC e 85,15% para a acurácia.

D. Pré-Processamento

Para se processar dados textuais em redes neurais, é necessário que esses dados sejam previamente tratados, visto que, as redes neurais não têm capacidade de processar o texto em seu formato original. A solução mais usual de para esse tipo de problema é a transformação do texto em vetores numéricos. Dessa forma, antes de se iniciar o processamento, realiza-se um processo de vetorização textual. Esse processo consiste na transformação do texto em tensores numéricos. Sendo assim, o texto pode ser separado por palavras, por letras ou por n-gramas, de forma que, cada palavra, letra ou n-gran seja representado por um vetor

Essas unidades em que o texto pode ser dividido (palavras, caracteres ou n-gramas) são chamadas de tokens e a divisão do texto em tokens é chamada de tokenização. Sendo assim, a vetorização de textos consiste na aplicação de algum processo de tokenização a esse dado textual, e na atribuição de vetores numéricos aos tokens gerados. Logo, esses vetores numéricos gerados funcionam como a entrada da rede neural.

As duas maneiras mais usuais de se representar uma palavra como vetor são a representação one-hot encoding (vetores esparsos) e a representação word embeddings (vetores densos).

- One-hot Encoding

Esse tipo de vetorização considera um vocabulário de tamanho N e atribui um índice i para cada um dos tokens do

obtidos no ambiente com CPU e com GPU foram respectivamente 0,7072 e 0,7111.

G. Convolutional Neural Network com Long Short-Term Memory

A rede CNN com LSTM era formada por 6 camadas. A primeira era a camada de embedding. A segunda camada era uma camada de convolução de 1 dimensão, formada por 8 filtros, com kernel de tamanho 5 e função de ativação RELU. Depois tinha-se uma camada de pooling e uma camada LSTM com 8 unidades. A camada seguinte era a de dropout com taxa de dropout de 10%. A última camada era uma camada Densa, com função de ativação softmax. A Fig. 6 ilustra a configuração dessa rede.

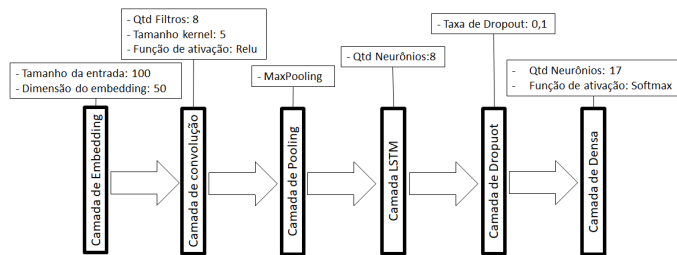


Fig. 6. Estrutura da rede CNN com LSTM.

O experimento da rede CNN com LSTM em CPU obteve acurácia de 94,12% e levou 573 segundos para ser executado. Já o experimento com a rede CNN com LSTM em GPU obteve acurácia de 94,09% e levou 256 segundos para ser executado.

Os resultados da métrica ROC AUC para rede CNN com LSTM, obtidos no ambiente com CPU e com GPU foram respectivamente 0,7781 e 0,7720.

H. Análise dos Resultados

A partir da análise dos resultados obtidos pelas redes CNN, LSTM e CNN com LSTM, em ambientes com CPU e em GPU, pode-se concluir que todas elas alcançaram bons resultados, quando comparadas com os modelos utilizados como base de comparação. Enquanto o modelo gerado pela SVM obteve a métrica ROC-AUC de 0,6132 e acurácia de 43,72%, e o modelo gerado pela CNN com uma base semelhante a utilizado nos experimentos obteve ROC-AUC de 0,7804 e acurácia de 85,15 %. Os modelos gerados pela CNN, LSTM e pela CNN com LSTM obtiveram os valores médios de ROC-AUC de 0,8782, 0,7091 e 0,7750 respectivamente. Esses modelos também obtiveram acurácia média de 94,56 %, 93,60% e 94,10 %.

Como observado, todos os experimentos apresentaram resultados superiores ao modelo gerado pelo SVM. No entanto, com relação aos resultados obtidos com o conjunto de dados do sistema de ouvidorias (com a aplicação de um modelo de deep learning), constatou-se que o experimento realizado com a rede CNN obteve valores superiores com relação a ambas as métricas analisadas, mas quanto as redes LSTM e CNN com LSTM, essas conseguiram acurácias

superiores, e valores de ROC-AUC similares, porém inferiores. Tal constatação pode ser um indicativo de que a rede CNN apresentaria uma melhor capacidade de generalização para esse tipo de dados.

Com relação aos modelos analisados nos experimentos, constatou-se que o modelo gerado pela rede CNN apresentou performance superior aos modelos da LSTM e da CNN com LSTM. Apesar de todos os modelos apresentarem valores médio de acurácia semelhantes, o modelo da CNN obteve melhor valor para a métrica ROC-AUC, além de ter sido o modelo que convergiu de forma mais rápida, conforme pode ser observado na Fig. 7, que ilustra o valor da métrica ROC AUC a cada 5 épocas para cada uma das redes testadas.

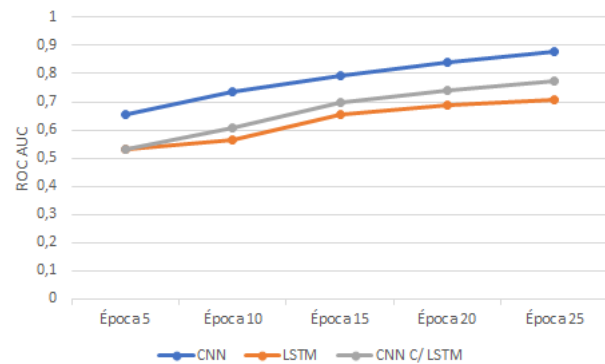


Fig. 7. Evolução da métrica ROC AUC por rede durante o treinamento.

Essa melhor performance da rede CNN pode ser justificada pelo fato das sentenças analisadas no experimento serem curtas, ou seja, composta por uma pequena quantidade de palavras.

Levando-se em consideração o aspecto de tempo, a eficiência dos modelos durante o treinamento variou de acordo com o tipo de processamento utilizado (CPU e GPU). Para o processamento em CPU, o modelo que treinou utilizando menos tempo foi a rede CNN, seguido pela rede CNN com LSTM e por último ficou a LSTM. O tempo gasto para a execução dos treinamentos utilizando CPU para as redes CNN, LSTM e CNN com LSTM foram respectivamente 172 segundos, 229 segundos, e 247 segundos.

Quando os modelos foram treinados utilizando GPU, o modelo que treinou em menor tempo também foi gerado pela rede CNN, seguido pela rede CNN com LSTM e por último também ficou a rede LSTM. Sendo que, os tempos gastos nos treinamentos das redes CNN, CNN com LSTM e LSTM foram de 24 segundos, 98 segundos e 146 segundos, respectivamente. A Fig. 8 ilustra a diferença de tempo de treinamento no ambiente em CPU e em GPU para cada uma das redes analisadas.

Apesar da ordem dos tempos de treinamento dos modelos ter sido mantida (tanto para o processamento em CPU quanto em GPU), o ganho de processamento em GPU para a rede CNN foi muito maior. A melhora de performance para os modelos, comparando-se os processamentos em CPU e GPU, foram de 7,17 vezes, 2,34 vezes, e 1,69 vezes para as redes CNN, CNN com LSTM e LSTM respectivamente. Essa diferença pode ser justificada pelo fato da rede LSTM

necessitar que parte do seu processo de treinamento seja sequencial, o que faz com que esse tipo de rede não usufrua totalmente dos benefícios trazidos pela paralelização obtida com as GPUs.

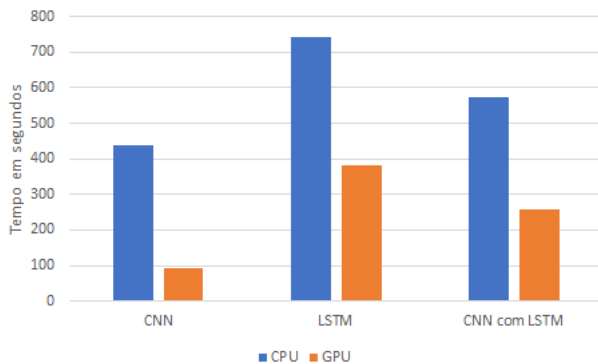


Fig. 8. Tempo de Treinamento em CPU e em GPU por rede.

Diante desses fatos, pode-se dizer que o melhor modelo para classificar os pedidos de acesso à informação do sistema e-sic é o modelo gerado pela rede CNN, pois, além de ter sido o modelo que treinou em menor tempo (tanto em GPU quanto em CPU), foi o modelo que convergiu de forma mais rápida e obteve melhores valores para as métricas avaliadas.

V. CONCLUSÃO

Esse trabalho testou o desempenho das redes CNN, LSTM e de uma combinação de CNN com LSTM para o problema de classificação de solicitações textuais de informações governamentais.

Os experimentos foram executados tanto em ambientes com processamentos com GPU quanto em ambientes apenas com CPU, sendo que, em ambos os casos a rede CNN foi a que treinou em menor tempo e obteve melhores resultados, sendo essa a arquitetura mais apropriada para os dados em questão.

Uma possível explicação para essa melhor performance da rede CNN é o fato dos textos analisados serem em sua maioria textos curtos.

Outro fato observado foi que o ganho de tempo de treinamento no ambiente com GPU em relação ao ambiente apenas com CPU, foi muito maior para a rede CNN, quando comparado com as redes LSTM e CNN com LSTM. Uma explicação para esse comportamento é o fato de que parte do processamento das células LSTM ter que ser feito obrigatoriamente de forma sequencial, o que impede uma melhor utilização do paralelismo proporcionado pelas GPUs.

Como trabalhos futuros, pretende-se testar outras arquiteturas de rede e outras formas de representação textual para o problema de classificação de textos de interação entre cidadãos e governo escritos na língua portuguesa.

REFERÊNCIAS

- [1] C. Zhou, C. Sun, Z. Liu, e F. C. M. Lau, “A C-LSTM Neural Network for Text Classification”, arXiv:1511.08630 [cs], nov. 2015, Acessado: jul. 04, 2020. [Online]. Disponível em: <http://arxiv.org/abs/1511.08630>.
- [2] S. Undavia, A. Meyers, e J. Ortega, “A Comparative Study of Classifying Legal Documents with Neural Networks”, in *Annals of Computer Science and Information Systems*, 2018, vol. 15, p. 515–522, Acessado: jul. 04, 2020. [Online]. Disponível em: <https://annals-csis.org/proceedings/2018/drp/227.html>.
- [3] N. Kalchbrenner, E. Grefenstette, e P. Blunsom, “A Convolutional Neural Network for Modelling Sentences”, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, jun. 2014, pp. 655–665, doi: 10.3115/v1/P14-1062.
- [4] S. Poria, E. Cambria, D. Hazarika, e P. Vij, “A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks”, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, dez. 2016, p. 1601–1612, Acessado: jul. 04, 2020. [Online]. Disponível em: <https://www.aclweb.org/anthology/C16-1151>.
- [5] M. Marques, *Administração Pública: uma abordagem prática*. Editora Ferreira, 2010.
- [6] N. T. Vu, H. Adel, P. Gupta, e H. Schütze, “Combining Recurrent and Convolutional Neural Networks for Relation Classification”, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, jun. 2016, pp. 534–539, doi: 10.18653/v1/N16-1065.
- [7] I. Banerjee et al., “Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification”, *Artificial Intelligence in Medicine*, vol. 97, p. 79–88, jun. 2019, doi: 10.1016/j.artmed.2018.11.004.
- [8] W. Quan, Z. Chen, J. Gao, e X. T. Hu, “Comparative study of CNN and LSTM based attention neural networks for aspect-level opinion mining”, in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 2141–2150.
- [9] W. Yin, K. Kann, M. Yu, e H. Schütze, “Comparative Study of CNN and RNN for Natural Language Processing”, *CoRR*, vol. abs/1702.01923, 2017, Acessado: ago. 24, 2018. [Online]. Disponível em: <http://arxiv.org/abs/1702.01923>.
- [10] Y. Kim, “Convolutional Neural Networks for Sentence Classification”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, out. 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.
- [11] P. Goyal, S. Pandey, e K. Jain, *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*. Apress, 2018.
- [12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, e J. Dean, “Distributed Representations of Words and Phrases and their Compositionality”, *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 3111–3119, out. 2013.
- [13] T. Mikolov, K. Chen, G. Corrado, e J. Dean, “Efficient Estimation of Word Representations in Vector Space”, *ICRL*, jun. 2013, Acessado: nov. 23, 2019. [Online]. Disponível em: <http://arxiv.org/abs/1301.3781>.
- [14] J. Chung, C. Gülçehre, K. Cho, e Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, *CoRR*, vol. abs/1412.3555, 2014, [Online]. Disponível em: <http://arxiv.org/abs/1412.3555>.
- [15] P. Bojanowski, E. Grave, A. Joulin, e T. Mikolov, “Enriching Word Vectors with Subword Information”, *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 136–146, jun. 2017.
- [16] J. L. Elman, “Finding Structure in Time”, *Cognitive Science*, vol. 14, no 2, p. 179–211, 1990, doi: 10.1207/s15516709cog1402_1.
- [17] J. Pennington, R. Socher, e C. Manning, “Glove: Global Vectors for Word Representation”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.
- [18] Y. Lecun, L. Bottou, Y. Bengio, e P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, vol. 86, no 11, pp. 2278–2324, nov. 1998, doi: 10.1109/5.726791.
- [19] S. Hochreiter e J. Schmidhuber, “Long Short-term Memory”, *Neural computation*, vol. 9, p. 1735–80, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [20] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso, J. Rodrigues, e S. Aluisio, “Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks”, *Symposium in Information and Human Language Technology*, pp. 122–131, 2017.
- [21] P. Wang et al., “Semantic Clustering and Convolutional Neural Network for Short Text Categorization”, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China, jul. 2015, pp. 352–357, doi: 10.3115/v1/P15-2058.



Eduardo Soares de Paiva Possui graduação em Engenharia da Computação pelo Instituto Militar de Engenharia (2003) e Mestrado em Informática pela Universidade Federal do Estado do Rio de Janeiro (2017). Aluno de doutorado do Programa de Engenharia Civil, na área de Sistemas Computacionais, da COPPE -

Universidade Federal do Rio de Janeiro. <http://lattes.cnpq.br/7688846833775668>.



Andréa Borges Paim graduada em Estatística pela Escola Nacional de Ciências e Estatística (1993), mestre em Estudos Populacionais e Pesquisas Sociais pela Escola Nacional de Ciências Estatísticas (2007). Atualmente é aluna de doutorado do Programa de Engenharia Civil, na área de Sistemas

Computacionais, da COPPE - Universidade Federal do Rio de Janeiro. <http://lattes.cnpq.br/7794414876694859>.



Nelson Francisco Favilla Ebecken possui doutorado em Engenharia Civil pela Universidade Federal do Rio de Janeiro (1977). Atualmente é Professor titular da Universidade Federal do Rio de Janeiro. Publicou 135 artigos em periódicos especializados e 361 trabalhos em anais de eventos. Possui 19 livros publicados. Orientou 135 dissertações de

mestrado e 138 teses de doutorado nas áreas de Ciência da Computação, Engenharia Civil e Ciência da Informação. Atua em áreas interdisciplinares da Engenharia e Engenharia de Petróleo, com ênfase em Sistemas Computacionais. As principais áreas de interesse são: Data Mining, Estruturas Offshore, Redes Neurais, Análise Não-Linear, Métodos Computacionais e Método dos Elementos Finitos. <http://lattes.cnpq.br/2703716951709834>.