

AVALIAÇÃO DOS FRAMEWORKS TENSORFLOW, PYTORCH E KERAS PARA DEEP LEARNING

GREICE ALINE WELTER¹, ANTHONY VANZAN¹, GABRIEL RUSTICK FIM¹, MATHEUS CÉSAR SAUSEN¹, DALVAN GRIEBLER².

¹ Alunos, Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC), Faculdade Três de Maio (SETREM), Três de Maio/RS. greicewelter8@gmail.com, anthonyvanzan0202@gmail.com, gabifimtm@gmail.com, matheus.sausen00@gmail.com.

² Professor, Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC), Faculdade Três de Maio (SETREM), Três de Maio, RS. dalvangriehler@setrem.com.br

RESUMO

Os *frameworks* de programação são um conjunto de classes sobre a qual uma ferramenta é constituída. Esse esqueleto disponibilizará classes para construir, por exemplo, uma solução que permita criar e trabalhar com vários modelos heterogêneos de RNAs (Redes Neurais Artificiais) interligadas. Como existem diversas opções disponíveis, não se sabe qual delas é a mais adequada. Em uma pesquisa prévia no contexto do projeto de pesquisa AGROCOMPUTAÇÃO (parceria entre SETREM e TECNICON), identificou-se que os principais *frameworks* para o desenvolvimento de RNAs profundas (Deep Learning) são TensorFlow, PyTorch e Keras. A partir disso, o objetivo foi avaliar qual delas é a melhor para o desenvolvimento de modelos de RNAs. Além de contribuir com o projeto AGROCOMPUTAÇÃO, os resultados dessa pesquisa poderão auxiliar na tomada de decisão de outros projetos de *software* que visam criar modelos de RNAs profundas. Os testes efetuados apontaram que o PyTorch é melhor no quesito acurácia e é mais flexível para o desenvolvimento de modelos de RNAs se comparado ao TensorFlow e Keras.

Palavras-chave: Redes Neurais, Inteligência Artificial, *Frameworks* de Programação.

1 INTRODUÇÃO

O uso de Inteligência Artificial (IA) tem se tornado cada vez mais importante para a descoberta de conhecimento em dados. No contexto de imagens, as RNAs são os algoritmos de aprendizado de máquina que estão obtendo o melhor desempenho, através da implementação de arquiteturas com camadas profundas (MACHADO,2020). Essa característica de aprendizado profundo é chamada de *Deep Learning*.

Para simplificar o desenvolvimento desses modelos de *Deep Learning*, foram criados *frameworks* de programação, onde diferentes arquiteturas e tipos de RNAs podem ser estendidos ou customizados com maior facilidade e produtividade. Com o passar do tempo, várias opções foram disponibilizadas com diferentes funcionalidades e propósitos. Fica difícil escolher com assertividade qual deverá ser usada. Por isso, uma pesquisa teórica no projeto AGROCOMPUTAÇÃO foi conduzida com o intuito de chegar nos três principais frameworks, onde concluiu-se que são TensorFlow, PyTorch e Keras.

O objetivo deste trabalho é avaliar na prática quais dessas três opções possui maior acurácia e flexibilidade no desenvolvimento de modelos para *Deep Learning*. Isso é o primeiro passo antes de iniciar qualquer projeto de software que almeja o

desenvolvimento de modelos de RNAs. Por isso, essa pesquisa também poderá contribuir na tomada de decisão de outras pessoas que desejam atuar nesta área.

Dentre os trabalhos encontrados na literatura, o artigo (FLORENCIO, *et al.*, 2019) fez uma comparação entre as bibliotecas TensorFlow e PyTorch, em uma CNN onde os autores constataram uma ausência de desempenho da biblioteca PyTorch na análise de processos de uma *Lenet* utilizando o dataset MNIST. Diferente dele, este estudo visou avaliar três opções (TensorFlow, Keras e PyTorch) usando uma RNA diferente e outro dataset. O restante deste documento está organizado da seguinte forma. A Seção 2 faz uma breve apresentação dos principais conceitos sobre a área e sobre os *frameworks* estudados. A Seção 3 descreve a metodologia utilizada para o desenvolvimento do trabalho. Na Seção 4 é apresentada uma análise de desenvolvimento dos resultados obtidos. Por fim na seção 5 conclui-se o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

RNAs são compostas de neurônios interligados em camadas, que se assemelha ao comportamento do cérebro humano. Cada neurônio é ativado com uma função matemática. As camadas podem ter características diferentes e o comportamento desta rede também varia. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento (neurônios), enquanto que o cérebro de um mamífero pode ter bilhões de neurônios (ICMC, 2020). Os tipos de redes mais usados são:

- RNN (*Recurrent Neural Network*): tem como objetivo fazer previsões sobre dados sequenciais utilizando uma arquitetura baseada em memória. Com o uso de um estado de memória, a arquitetura de RNN consegue resolver perfeitamente os problemas na rede neural (KOSTADINOV, 2018).
- CNN (*Convolutional Neural Network*): transforma uma imagem em um vetor de características. Os pesos dos filtros são calculados automaticamente (BackPropagation), e cada filtro aprendido é um extrator de características (UFPR, 2020).

RNAs de *Deep learning* são um tipo de aprendizado de máquina usado para treinar computadores a realizar tarefas humanas, o que inclui reconhecimento de fala, identificação de imagem e previsões. Ao invés de organizar os dados para serem executados através de equações pré-definidas, o deep learning configura parâmetros básicos sobre os dados e treina o computador para aprender sozinho através do reconhecimento de padrões em várias camadas de processamento (SAS, 2020). Para isto, são necessários frameworks próprios para seu desenvolvimento, algumas das mais populares são: TensorFlow, PyTorch, Theano e Keras. Os frameworks são um conjunto de classes sobre a qual uma ferramenta é constituída. Esse esqueleto disponibilizará classes para construir uma solução que permita criar e trabalhar com vários modelos heterogêneos de RNAs (Redes Neurais Artificiais) interligadas (REINALDO, 2013). A seguir, são apresentados os principais para implementação de modelos de *Deep Learning*.

Keras é uma API que funciona sob o Tensorflow ou o Theano, atuando de forma frontend. O Keras foi desenvolvido com foco em permitir experimentação rápida. Todos os objetos retornados de funções neste pacote são objetos R nativos

ou ponteiros brutos para objetos python, possibilitando que os usuários acessem toda a API do Keras (CRAN, 2020). A API Keras possui uma interface simples e otimizada, facilitando na criação e o entendimento de um modelo de rede neural profunda.

O Tensorflow é uma biblioteca open source para computação numérica e Machine Learning. O framework utiliza a linguagem de programação python provendo uma API front-end para o desenvolvimento de aplicações executadas em C++ de alto desempenho. O TensorFlow é capaz de realizar a execução e treinamento de Deep Neural Networks para diversas aplicações como processamento de linguagem, reconhecimento de imagens e classificação de dígitos (INFOWORLD, 2020). O TensorFlow possui como benefício a sua abstração e flexibilidade, fala facilitar o entendimento do usuário existem APIs que ajudam a utilizar as tecnologias do TensorFlow.

O PyTorch é uma biblioteca open source de machine learning, utilizada em aplicações como processamento de linguagem natural. Foi inicialmente desenvolvida pelo grupo de pesquisa em inteligência artificial do Facebook, como um invólucro para o framework baseado em Torch LuaJIT (VASUDEVAN, 2018). O PyTorch possui uma grande flexibilidade e customização para a escrita de uma rede neural e tem como desvantagem a falta de certas funções e métodos presentes nos outros frameworks comparados, como o método *fit* que realiza o treinamento e validação da rede neural e a função *flatten* que recebe uma saída das camadas convolucionais da rede e as passa para camadas lineares.

3 METODOLOGIA

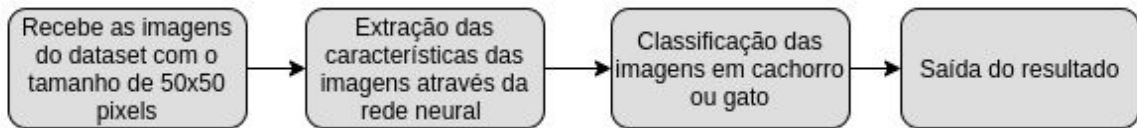
Para efetuar a avaliação da acurácia, foi adotado o procedimento experimental. A flexibilidade foi avaliada de forma qualitativa durante o desenvolvimento do programa de teste. No caso, foi implementado o mesmo tipo de rede nos três frameworks. Esse programa de teste teve como objetivo classificar imagens de gatos e cachorros.

Os algoritmos desenvolvidos utilizaram o mesmo *dataset* com 25000 imagens contendo gatos e cachorros, que está disponível em (MICROSOFT, 2020). Definiu-se que 30% do dataset será usado para a validação. O treinamento do modelo foi realizado em 10 épocas e os testes foram executados 3 vezes para certificar da corretude. Esses testes foram realizados no mesmo computador composto de um Processador Intel Core i5-7200U CPU @ 2.50GHz x 4, GPU Intel HD Graphics 620 (Kaby LakeGT2), Memória RAM 8GB, HDD de 1TB e tendo como S.O. Linux Ubuntu 18.04.2LTS.

4 APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS

O programa de teste desenvolvido é o mesmo para os três frameworks testados usando a linguagem Python. O que muda são as classes e rotinas relativas aos frameworks para implementação da rede neural convolucional. No caso, foi desenvolvido um fluxograma para melhorar o entendimento do algoritmo como ilustra a Figura 1. Nele descreve a estrutura onde o algoritmo recebe as imagens do *dataset* redimensionadas com o tamanho de 50x50 pixels, passando pelas camadas de extração da rede e classificando as imagens de gatos e cachorros e por fim obtendo os resultados da rede.

Figura 1 - Fluxograma da Rede Neural



A Figura 2 apresenta os valores finais obtidos nos testes realizados em cada *framework*, podendo ser observado que o PyTorch obteve 0,88 de acurácia e 0,11 de perda em um tempo de 1 min e 25s, obtendo o melhor resultado em relação ao Keras e TensorFlow. Sendo este um dos fatores para a escolha do mesmo como o *framework* utilizado no projetos AGROCOMPUTAÇÃO. Outro fator contribuinte para sua escolha foi a velocidade que o PyTorch obteve quando comparado aos outros, isto se deu devido a alta quantidade de ferramentas já presente dentro dos outros *frameworks*, o que os tornam mais lentos.

Figura 2 -Comparação dos frameworks

<i>Framework</i>	Acurácia	Perda	Tempo Exec. Médio	Num. Épocas Trein.	Num. Imagens Trein.	Num. Imagens Validação
TensorFlow	0,78	0,6	6 min	10	17462	7484
Keras	0,78	0,58	7 min	10	17462	7484
PyTorch	0,88	0,11	1min 25s	10	17462	7484

5 CONCLUSÃO

Neste trabalho foi realizada uma análise comparativa do Keras, Tensorflow e PyTorch relativo ao desempenho e desenvolvimento de modelos de *Deep Learning*. Foram avaliados a acurácia e perda dos testes realizados com os algoritmos. Constatou-se que dentre as três opções o PyThorch obteve o melhor tempo de execução, melhor acurácia e menor perda. Um dos grandes desafios encontrados ao foi a curva de aprendizagem que cada framework possui e a realização dos testes dos algoritmos desenvolvidos. Vale destacar que o uso de CPU limitou a capacidade de processamento. Em modelos maiores isso pode se tornar um problema, podendo levar horas e até dias para realizar o treinamento. Outro ponto importante é que esse trabalho chegou em uma conclusão diferente do trabalho relacionado. Após concluir esta pesquisa, passou-se a adotar o PyTorch no projeto AGROCOMPUTAÇÃO, onde vem se demonstrando eficiente e mais rápido quando executado em um computador com GPU.

6 REFERÊNCIAS

REINALDO, F.A.F. 2013. **Definição e aplicação de um framework para desenvolvimento de redes neurais modulares e heterogêneas.** [Online].

[Acessado em 24/082020]. Disponível em <<http://repositorio.ufsc.br/xmlui/handle/123456789/85689>>.

ICMC. 2020. **Redes Neurais Artificiais** [Online]. [Acessado em 31/08/2020]. Disponível em <<https://sites.icmc.usp.br/andre/research/neural/>>.

SAS. 2020. **Deep Learning**. [Online]. [Acessado em 30/082020]. Disponível em <https://www.sas.com/pt_br/insights/analytics/deep-learning.html>.

CRAN. 2020. **R Interface to the Keras Deep Learning Library** [Online]. [Acessado em 31/08/2020]. Disponível em: <<https://cran.r-project.org/web/packages/kerasR/vignettes/introduction.html>>.

VASUDEVAN, C. 2018 **Concepts and Programming in PyTorch**. ISBN: 9388176057, 9789388176057.

INFOWORLD. 2020 **What is TensorFlow? The machine learning library explained** [Online]. [Acessado em 31/08/2020]. Disponível em <<https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>>.

GAEA. 2020 **Afinal, o que é Deep Learning?** [Online]. [Acessado em 02/09/2020]. Disponível em <<https://gaea.com.br/afinal-o-que-e-deep-learning/>> .

MICROSOFT. 2020 [Online]. [Acessado em: 05/09/2020]. Disponível em: <https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_3367a.zip>.

UFPR. 2020 **Redes Neurais Convolucionais** [Online]. Disponível em: http://www.inf.ufpr.br/aghochuli/caffe/CNN_PPT.pdf. Acessado 08 de set 2020.

KOSTADINOV. 2018 **Recurrent Neural Networks with Python Quick Start Guide.Packt** [Online]. [Acessado em 08/09 2020]. Disponível em <<https://www.packtpub.com/product/recurrent-neural-networks-with-python-quick-start-guide/9781789132335>> .

FLORENCIO. JUNIOR. ORDONEZ. SILVA. 2019 **Performance Analysis of Deep Learning Libraries: TensorFlow and PyTorch** [Online]. [Acessado em: 08/09 2020]. Disponível em <https://www.researchgate.net/publication/332933108_Performance_Analysis_of_Deep_Learning_Libraries_TensorFlow_and_PyTorch>.

MACHADO. 2020 **Inteligência Artificial** [Online]. [Acessado em 08/092020]. Disponível em < www.uece.br> .