COMPARAÇÃO ENTRE MODELOS RESNET NA CLASSIFICAÇÃO DE DÍGITOS MANUSCRITOS

Renan Marcili Lemons[1]; Rafael Duarte dos Santos[2]; Ruhan Avila da Conceição[3]

[1]*Universidade Federal de Pelotas – rmlemons@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – rrdsantos@inf.ufpel.edu.br*
[3]*Instituto Federal Sul-rio-grandense – ruhanconceicao@ifsul.edu.br*

1. INTRODUÇÃO

A inteligência artificial tem se popularizado cada vez mais nas últimas décadas, sendo cada vez mais utilizada nos mais diversos setores, desde assistentes pessoais, a indústria de transporte e até na área da saúde (OUR WORLD IN DATA, 2023). Um dos grandes motivos responsáveis por essa evolução rápida é o aprendizado profundo (*deep learning*), sendo uma subárea do aprendizado de máquina, que trabalha a ideia de algoritmos que possibilitem ao computador aprender diretamente a partir dos dados, sem a necessidade de uma programação específica para cada uma das tarefas.

De acordo com Raschka (2017), existem três principais tipos de aprendizado de máquina:

- Aprendizado supervisionado: o modelo é treinado usando dados rotulados, ou seja, tanto as entradas quanto as saídas corretas são fornecidas. O objetivo é aprender uma função que mapeia as entradas e as saídas levando em conta os dados fornecidos durante o treinamento. Geralmente utilizados em tarefas de classificação.
- Aprendizado não supervisionado: onde o modelo é treinado sem os dados rotulados, então, neste caso o objetivo é que o modelo aprenda a identificar padrões ou estruturas, sem necessidade de descobrir uma saída correta.
- Aprendizado por reforço: este tipo de aprendizado envolve um agente que vai aprendendo a tomar decisões ao interagir por meio de tentativa e erro, sendo recompensado ou penalizado por cada ação tomada, reforçando as ações corretas através de uma recompensa, ou atenuando ações incorretas por meio de penalidades. Ele busca sempre maximizar a recompensa acumulada.

Redes neurais são uma classe de algoritmos de aprendizado de máquina que buscam simular o funcionamento do cérebro humano, visando reconhecer padrões complexos nos dados fornecidos. O tamanho das redes neurais podem variar de apenas um neurônio (perceptron) até centenas de camadas, onde surge o termo aprendizado profundo (*deep learning*). Devido a capacidade de aprendizado elevado das redes neurais, somado ao fato do crescimento do poder computacional e da quantidade de dados disponíveis atualmente, o aprendizado profundo tem se tornado a técnica principal para tarefas como o reconhecimento de imagem, processamento de linguagem natural, sendo capaz até mesmo de superar o desempenho humano em algumas tarefas(GOODFELLOW; BENGIO; COURVILLE, 2016).

As redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) são uma classe especial de redes neurais profundas focadas principalmente para

o processamento de dados tipicamente organizados em matrizes, como imagem e vídeo. Este tipo de rede busca extrair características hierárquicas dos dados para permitir a identificação de padrões em imagens. (RUN.AI, 2024).

Dentre os diversos algoritmos de classificação de imagens baseados em CNNs, destacam-sem as ResNets, cuja nomenclatura vem de redes residuais, que foram introduzidas no artigo Deep Residual Learning for Image Recognition, que tratam de uma nova forma de treinar redes profundas, sendo mais fácil e eficiente treinar uma rede se ela aprender a ajustar “resíduos” em vez de tentar aprender uma nova função completa do zero (HE et al., 2016).A ResNet possui diversos tamanhos de camadas, variando de 18 até 152, onde o maior número de camadas é indicado para tarefas de classificação mais complexas. Tendo isto em vista, este trabalho visa comparar os modelos *ResNet* 18, *ResNet* 34 e *ResNet* 50 em termos de eficiência de classificação e tempo de execução na classificação de dígitos numéricos sobre o dataset MNIST (LECUN; CORTES; BURGES, 1998).

2. METODOLOGIA

Para a implementação e execução dos modelos foi escolhida a linguagem de programação Python, devido a popularidade da mesma, o que acaba por fornecer uma grande quantidade de material online disponível sobre a mesma. Para o desenvolvimento do trabalho específico no que tange às redes neurais, a biblioteca escolhida foi a PyTorch, pois ela já possui a implementação de funções e ferramentas para o desenvolvimento do trabalho, facilitando o processo de treinamento dos modelos escolhidos.

A execução dos modelos realizou-se através de ambientes de computação na nuvem. Utilizando o Google Colab, aproveitando o acelerador de hardware GPU-T4 para acelerar o treinamento das redes. Devido ao Google Colab ter um limite de uso, alternativamente, utilizou-se o *Kaggle*, utilizando a configuração de GPU T4x2 no treinamento das redes. É importante salientar que embora o ambiente de execução do Kaggle disponibilize duas GPUs simultaneamente, apenas uma foi utilizada nas tarefas de treino e inferência.

O dataset utilizado no treinamento e nos testes dos modelos foi o *MNIST*, que é um dataset de imagens sobre números escritos à mão, contendo cerca de 70.000 imagens. O objetivo deste trabalho é avaliar a eficácia de três modelos de *ResNet* na classificação destas imagens, considerando também o custo de execução e as diferenças entre os modelos.

Os seguintes hiperparâmetros foram definidos:

- *Batch Size*: é referente ao número de exemplos de treinamento processados antes de atualizar os parâmetros do modelo. Foi utilizado um batch size de 512.
- *Learning Rate*: responsável por definir a velocidade que o modelo ajusta seus parâmetros durante o treinamento. Foi utilizado um learning rate fixo de 0.0001
- *Epochs*: é o número de vezes que o modelo passa por todo conjunto de dados de treinamento. Foi utilizado um número de trinta Epochs em cada treinamento.

O processo de treinamento das redes neurais foi realizado da seguinte maneira: primeiramente, cada uma das ResNet foi obtida com os pesos

pré-treinados, alterando-se apenas a última camada totalmente conectada (fully connected - FC). Durante o treinamento, os pesos de todas as camadas convolucionais foram congelados, ou seja, o gradiente descendente foi aplicado apenas na camada de saída FC, sem atualização dos pesos nas demais camadas.

3. RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados os resultados dos treinamentos das arquiteturas *ResNet 18, ResNet 34* e *ResNet 50*, sobre o *dataset MNIST,* para a tarefa de classificação de dígitos manuscritos. Os três modelos foram treinados por trinta épocas, com um *learning rate* fixo de 0,0001 e um *batch size* de 512. Os resultados são apresentados na Tabela 1.

Tabela 1 - Resultados obtidos a partir modelos testados

	ResNet18	ResNet34	ResNet50
Loss 1ª Epoch	2,2074	2,0676	1,8783
Loss 30ª Epoch	0,2100	0,2483	0,2490
Acurácia Conjunto de Teste Fino	95,03%	93,53%	92,88%
Acurácia Conjunto de Teste	94,78%	93,50%	93,42%
Tempo de Execução	3736 segundos	4748 segundos	7002 segundos

Como mostra a tabela com os resultados obtidos, o modelo *ResNet 18,* para a classificação sobre o dataset MNIST se mostrou mais eficiente, seja no parâmetro de acurácia, quanto no tempo de execução. As possíveis razões para uma rede neural mais simples ter obtido um desempenho melhor comparado a redes com mais camadas, podem ser a simplicidade da tarefa. Como o *dataset MNIST* contém imagens de dígitos manuscritos com baixa resolução (28x28 pixels) e apenas dez classes (de 0 a 9), as características necessárias para diferenciar os números são relativamente simples e podem ser aprendidas com menos camadas de convolução.

Outra possibilidade é o *overfitting,* pois redes mais profundas têm uma quantidade significativamente maior de parâmetros, o que pode levar esses modelos a se ajustar demais aos dados de treinamento, reduzindo a capacidade de generalização no conjunto de testes. Como dito anteriormente o *dataset MNIST* é pequeno a *ResNet* 34 e *ResNet* 50 podem ter apresentado um *overfitting,* o que justificaria o modelo *ResNet* 18 ter obtido um melhor desempenho nos testes.

4. CONCLUSÕES

Os testes realizados para o desenvolvimento deste trabalho mostraram que a *ResNet* 18 apesar de ser um modelo de rede neural mais simples se mostrou mais eficiente em uma tarefa relativamente simples no campo de classificação de imagens, o que mostra que a complexidade da rede não resulta necessariamente em um desempenho melhor. Portanto, a escolha da arquitetura da rede neural deve considerar a complexidade da tarefa que deseja ser realizada, levando em conta o equilíbrio entre o desempenho e a eficiência computacional. O desenvolvimento deste trabalho mostrou-se uma importante tarefa para a aprendizagem no treinamento e desenvolvimento de redes neurais sobre a arquitetura ResNet. Para trabalhos futuros, espera-se avaliar o desempenho dessas redes em *datasets* mais complexos, ou também o desempenho das mesmas ao aplicar a quantização nesses modelos, analisando os impactos causados sobre o desempenho e eficiência computacional dos mesmos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, Las Vegas, v.770, n.778, p.770-778, 2016.

OUR WORLD IN DATA**. The brief history of artificial intelligence: the world has changed fast—what might be next?** 2023. Online. Disponível em: https://ourworldindata.org/ai. Acessado em: 19 set. 2024.

RASCHKA, Sebastian. **Python Machine Learning**. 2. ed. Birmingham: Packt Publishing, 2017. 622 p.

RUN.AI. **Deep Convolutional Neural Networks**. 2023 Online. Disponível em: https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks. Acessado em: 22 set. 2024.

LECUN, Y.; CORTES, C.; BURGES, C. J. C. **The MNIST Database of Handwritten Digits**. *1998*. Acessado em 18 set. 2024. Disponível em: http://yann.lecun.com/exdb/mnist/