Universidade Federal de Juiz de Fora
Relatório Técnico do Departamento de Ciência da Computação

RelaTeDCC 002/2023

Implementação de Redes Neurais Convolucionais

Bruno Martins Bartolomeu

Prof. Dr. Luiz Maurílio da Silva Maciel

Setembro 2023

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

Agradecimentos

Agradeço primeiramente a Deus pela minha vida e por ter me ajudado a superar todos os obstáculos ao longo dela, aos meus pais e irmão que sempre me incentivaram nos momentos mais difíceis, e a minha esposa por todo apoio e suporte.

Trabalho Prático 02

.

1. Introdução
2. Redes Neurais Convolucionais
3. Estratégias de implementação
4. Descrição dos experimentos
5. Conclusão
6. Referências

RESUMO

Redes neurais são sistemas computacionais que simulam o sistema nervoso humano e foram desenvolvidas para ajudar na resolução de problemas que requerem análises complexas e ponderadas, considerando diferentes variáveis.

Pertencentes ao universo de Machine Learning, considerado o coração dos algoritmos de deep learning.

Na prática, as redes replicam nas máquinas a maneira como o ser humano pensa. Ou seja, elas permitem que as máquinas aprendam e entendam padrões, adaptem-se a mudanças e imitem processos de pensamento humano em situações que precisam de respostas ou soluções.

Palavras-Chave: Redes Neurais. Reconhecimento de imagem. Deep Learning. CNN. Aprendizado de Máquina. FashionMNIST. CIFAR-10. AlexNet. LeNet. VGG.

ABSTRACT

Neural networks are computational systems that simulate the human nervous system and were developed to help solve problems that require complex and weighted analysis, considering different variables.

Belonging to the universe of Machine Learning, considered the heart of deep learning algorithms.

In practice, networks replicate the way humans think in machines. That is, they allow machines to learn and understand patterns, adapt to changes, and mimic human thought processes in situations that need answers or solutions.

Keywords: Neural networks. Image recognition. Deep Learning. CNN. Machine Learning. FashionMNIST. CIFAR-10. AlexNet. LeNet. VGG.

1 Introdução

Um breve histórico sobre Redes Neurais deve começar por três das mais importantes publicações iniciais, desenvolvidas por: McCulloch e Pitts (1943), Hebb (1949), e Rosemblatt (1958). Estas publicações introduziram o primeiro modelo de redes neurais simulando "máquinas", o modelo básico de rede de auto-organização, e o modelo Perceptron de aprendizado supervisionado, respectivamente.

Alguns históricos sobre a área costumam de forma equivocada "pular" os anos 60 e 70 e apontar um início da área com a publicação dos trabalhos de Hopfield (1982) relatando a utilização de redes simétricas para otimização.

Redes Neurais são um subconjunto de machine learning e estão no centro de importância relacionado aos algoritmos de deep learning. Seu nome e estrutura são inspirados no cérebro humano, imitando a maneira como os neurônios biológicos enviam sinais uns para os outros através de nós interconectados em uma estrutura de camadas.

Através de um sistema adaptativo, usam para aprender com os erros e se aprimorar continuamente. São utilizadas especificamente para criação de Inteligência Artificial, reconhecimento de imagens e entender informações não lineares.

TOP 3 das principais arquiteturas de Redes Neurais:

Perceptrons

Convolucionais

Recorrentes

2 Redes Neurais Convolucionais

Em 1998, Yann LeCun e seus colaboradores desenvolveram um reconhecedor, realmente bom, para dígitos manuscritos chamado LeNet. Ele usou o backpropagation em uma rede feed forward com muitas camadas ocultas, muitos mapas de unidades replicadas em cada camada, agrupando as saídas de unidades próximas, formando uma rede ampla que pode lidar com vários caracteres ao mesmo tempo, mesmo se eles se sobrepõem e uma inteligente maneira de treinar um sistema completo, não apenas um reconhecedor. Mais tarde, esta arquitetura foi formalizada sob o nome de **Redes Neurais Convolucionais**.

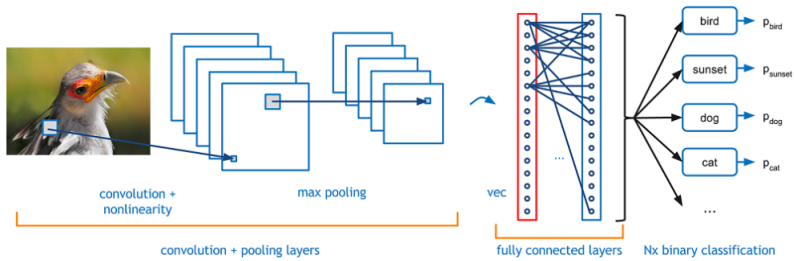
As Redes Neurais Convolucionais (ConvNets ou CNNs) são redes neurais artificiais profundas que podem ser usadas para classificar imagens, agrupá-las por similaridade (busca de fotos) e realizar reconhecimento de objetos dentro de cenas.

As Redes Convolucionais realizam o reconhecimento óptico de caracteres (OCR) para digitalizar texto e tornar possível o processamento de linguagem natural em documentos analógicos e manuscritos, onde as imagens são símbolos a serem transcritos. As CNNs também podem ser aplicadas a arquivos de áudio quando estes são representados visualmente como um espectrograma. Mais recentemente, as redes convolucionais foram aplicadas diretamente à análise de texto, bem como dados gráficos.

A eficácia das redes convolucionais no reconhecimento de imagem é uma das principais razões pelas quais o mundo testemunhou a eficácia do aprendizado profundo. Este tipo de rede está impulsionando grandes avanços em Visão Computacional, que tem aplicações óbvias em carros autônomos, robótica, drones, segurança, diagnósticos médicos e tratamentos para deficientes visuais.

As Redes Convolucionais ingerem e processam imagens como tensores e tensores são matrizes de números com várias dimensões. Eles podem ser difíceis de visualizar, então vamos abordá-los por analogia. Um escalar é apenas um número, como 7; um vetor é uma lista de números (por exemplo, [7,8,9]); e uma matriz é uma grade retangular de números que ocupam várias linhas e colunas como uma planilha. Geometricamente, se um escalar é um ponto de dimensão zero, então um vetor é uma linha unidimensional, uma matriz é um plano bidimensional, uma pilha de matrizes é um cubo tridimensional e quando cada elemento dessas matrizes tem uma pilha de mapas de recursos ligados a ele, você entra na quarta dimensão.

A primeira coisa a saber sobre redes convolucionais é que elas não percebem imagens como os humanos. Portanto, você terá que pensar de uma maneira diferente sobre o que uma imagem significa quando é alimentada e processada por uma rede convolucional.



As Redes Convolucionais percebem imagens como volumes; isto é, objetos tridimensionais, em vez de estruturas planas a serem medidas apenas por largura e altura.

Isso porque as imagens de cores digitais têm uma codificação vermelho-verde-azul (RGB – Red-Green-Blue), misturando essas três cores para produzir o espectro de cores que os seres humanos percebem. Uma rede convolucional recebe imagens como três estratos separados de cores empilhados um em cima do outro.

Assim, uma rede convolucional recebe uma imagem como uma caixa retangular cuja largura e altura são medidas pelo número de pixels ao longo dessas dimensões e cuja profundidade é de três camadas profundas, uma para cada letra em RGB. Essas camadas de profundidade são referidas como canais.

À medida que as imagens se movem através de uma rede convolucional, descrevemos em termos de volumes de entrada e saída, expressando-as matematicamente como matrizes de múltiplas dimensões dessa forma: 30x30x3. De camada em camada, suas dimensões mudam à medida que atravessam a rede neural convolucional até gerar uma série de probabilidades na camada de saída, sendo uma probabilidade para cada possível classe de saída. Aquela com maior probabilidade, será a classe definida para a imagem de entrada, um pássaro por exemplo.

As Redes Neurais Convolucionais basicamente foram criadas para resolverem uma tarefa específica: o reconhecimento de imagens, que é um problema da área de visão computacional.

3 Estratégias de implementação

Este trabalho foi totalmente implementado com a linguagem **Python** e contou com as seguintes bibliotecas: **Numpy, Tensorflow, Pytorch, Matplotlib, sys**.

Utilizado também a **API Keras** e o ambiente de desenvolvimento escolhido foi o **Google Colab**.

3 arquiteturas de Redes Neurais **(AlexNet, LeNet, VGG)** e os cojuntos de dados escolhidos: **FahsionMnist e CIFAR-10**.

FashionMnist: O conjunto de dados FashionMNIST é um grande banco de dados de imagens de moda disponível gratuitamente , comumente usado para treinar e testar vários sistemas de aprendizado de máquina .O FashionMNIST foi concebido para servir como um substituto para o banco de dados MNIST original para benchmarking de algoritmos de aprendizado de máquina, pois compartilha o mesmo tamanho de imagem, formato de dados e estrutura de divisões de treinamento e teste.

O conjunto de dados contém 70.000 imagens em escala de cinza 28x28 de produtos de moda de 10 categorias de um conjunto de dados de imagens de artigos da Zalando , com 7.000 imagens por categoria.

O conjunto de treinamento consiste em 60.000 imagens e o conjunto de teste consiste em 10.000 imagens.

CIFAR-10: é um acrônimo que significa Canadian Institute For Advanced Research (Instituto Canadense de Pesquisa Avançada) e o conjunto de dados CIFAR-10 foi desenvolvido junto com o conjunto de dados CIFAR-100 por pesquisadores do instituto CIFAR.

O conjunto de dados é composto por 60.000 fotografias coloridas de 32×32 pixels de objetos de 10 classes, como sapos, pássaros, gatos, navios, etc. São imagens muito pequenas, muito menores do que uma fotografia típica, e o conjunto de dados foi planejado para pesquisa de visão computacional.

O CIFAR-10 é um conjunto de dados bem conhecido e amplamente utilizado para referência na medição de algoritmos de visão computacional no campo de aprendizado de máquina.

AlexNet: A arquitetura AlexNet consiste em 5 camadas convolucionais, 3 camadas de pooling máximo, 2 camadas de normalização, 2 camadas totalmente conectadas e 1 camada softmax. 2. Cada camada convolucional consiste em filtros convolucionais e uma função de ativação não linear RELU.

LeNet: Em um alto nível, a arquitetura LeNet consiste em duas partes: um codificador convolucional que consiste em duas camadas convolucionais e um bloco denso que consiste em três camadas totalmente conectadas.

VGG: A rede VGG original tinha 5 blocos convolucionais, entre os quais os dois primeiros têm uma camada convolucional cada e os três últimos contêm duas camadas convolucionais cada. O primeiro bloco tem 64 canais de saída e cada bloco subsequente dobra o número de canais de saída, até que esse número chegue a 512.

4 Descrição dos experimentos

Propósito: Observar diferenças de resultados alterando os hiperparâmetros entre as 3 arquiteturas escolhidas utilizando os 2 conjuntos de dados citados acima.

Hiperparâmetros: batchsize / epochs / learningrate

Pontos para entendimento: / dropout / regularização / data augmentation

Legenda:

- hiperparâmetros: Variáveis fornecidas através de valores precisos para regular como os algoritmos aprendem e modificam o desempenho de um modelo.
- batchsize: Tamanho do lote (qtd de imagens do conjunto de dados de treinamento analisadas pelo algoritmo).
- epochs: Número de épocas (vezes que o algoritmo funcionará em todo o conjunto de dados de treinamento).
- learningrate: Taxa de aprendizagem (controla o quanto o modelo deve ser alterado em resposta ao erro estimado cada vez que os pesos do modelo são atualizados).
- dropout: Refere-se a desistência, eliminação dos nós em uma rede neural, considerado uma ténica de regularização.
- regularização: Técnicas utilizadas para reduzir o overfitting, mesmo quando temos uma rede de tamanho fixo e dados de treinamento em quantidade limitada.
- data augmentation: Técnica de Machine Learning usada para reduzir também o overfitting ao treinar um modelo.

Configuração da máquina utilizada para os testes:

- Nome dos dispositivo: LAPTOP-HAJNFCD0
- Processador: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
- Memória RAM: 8,00 GB (utilizável: 7,88 GB)
- Placa de Vídeo: NVIDIA GeForce MX110
- Armazenamento: SSD - 237 GB

Arquiteturas e hiperparâmetros avaliados:

1. LeNet - CIFAR10
    - learning rate = 0.001
    - batch size = 128
    - epochs = 10
    - Duração = 15m 17s
    - Gráfico figura 1
2. LeNet - CIFAR10
    - learning rate = 0.005
    - batch size = 200
    - epochs = 15
    - Duração = 15m 22s
    - Gráfico figura 2
3. LeNet - MNIST
    - learning rate = 0.001
    - batch size = 128
    - epochs = 10
    - Duração = 11 minutos
    - Gráfico figura 3
4. LeNet - MNIST
    - learning rate = 0.004
    - batch size = 180
    - epochs = 20
    - Duração = 12m 13s
    - Gráfico figura 4

5. AlexNet - MNIST
    - learning rate = 0.001
    - batch size = 128
    - epochs = 10
    - Duração = 9m 44s
    - Gráfico figura 5
6. AlexNet - MNIST
    - learning rate = 0.005
    - batch size = 200
    - epochs = 15
    - Duração = 12m 53s
    - Gráfico figura 6
7. AlexNet - CIFAR10
    - learning rate = 0.001
    - batch size = 32
    - epochs = 5
    - Duração = 2hrs 4m
    - Gráfico figura 7
8. AlexNet - CIFAR10
    - learning rate = 0.005
    - batch size = 100
    - epochs = 3
    - Duração = 55m 34s
    - Gráfico figura 8

9. VGG - CIFAR10
    - learning rate = 0.001
    - batch size = 100
    - epochs = 3
    - momentum = 0.9
    - Duração = 11m 20s
    - Gráfico figura 9
10. VGG - CIFAR10
    - learning rate = 0.003
    - batch size = 180
    - epochs = 5
    - momentum = 0.9
    - Duração = 15m 10s
    - Gráfico figura 10

11. VGG - MNIST
    - learning rate = 0.001
    - batch size = 128
    - epochs = 12
    - momentum = 0.9
    - Duração = 11 minutos
    - Gráfico figura 11
12. VGG - MNIST
    - learning rate = 0.005
    - batch size = 200
    - epochs = 15
    - momentum = 0.9
    - Duração = 17 minutos
    - Gráfico figura 12

Segue abaixo visualizações referentes as experimentações:

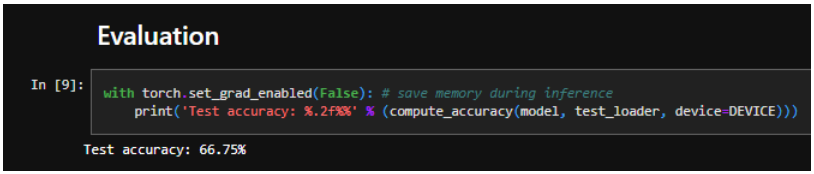


Figura 1- Nível de Acurácia gerada pela arquitetura LeNet - Cojunto de dados: CIFAR10

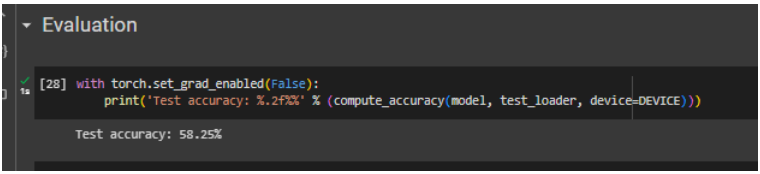


Figura 2- Nível de Acurácia gerada pela arquitetura LeNet - Cojunto de dados: CIFAR10 - **Hiperparâmetros alterados**

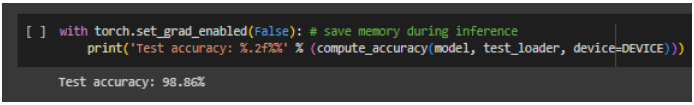


Figura 3- Nível de Acurácia gerada pela arquitetura LeNet - Cojunto de dados: MNIST

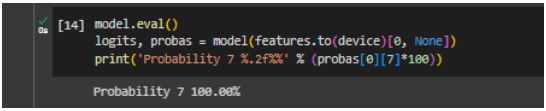


Figura 4- Nível de Acurácia gerada pela arquitetura LeNet - Cojunto de dados: MNIST - **Hiperparâmetros alterados**

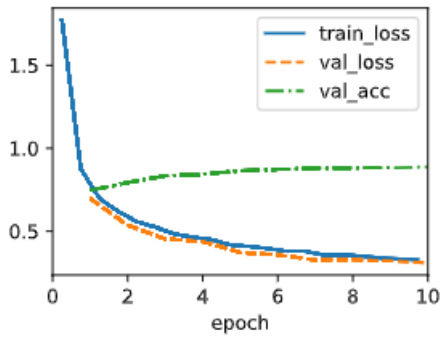


Figura 5- Exemplo de gráfico pela arquitetura AlexNet apresentando epochacc e epochloss - Cojunto de dados: MNIST

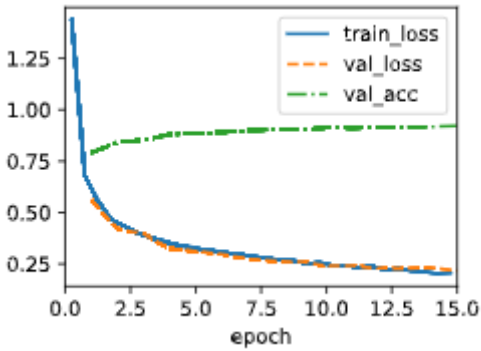


Figura 6- Exemplo de gráfico gerado pela arquitetura AlexNet apresentando epochacc e epochloss - Cojunto de dados: MNIST - **Hiperparâmetros alterados**

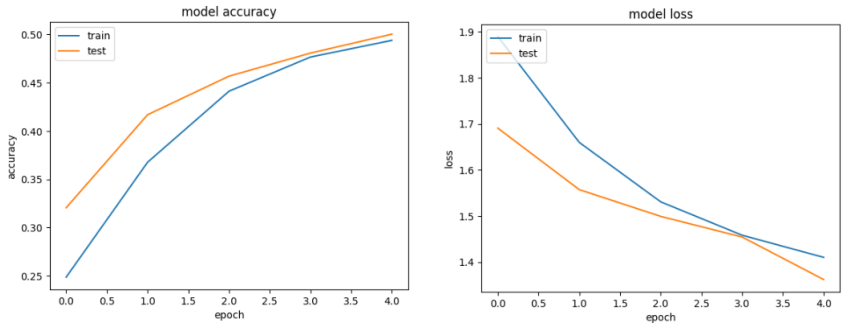


Figura 7- Exemplo de gráfico gerado via matplotlib pela arquitetura AlexNet apresentando epochacc e epochloss - Cojunto de dados: CIFAR10

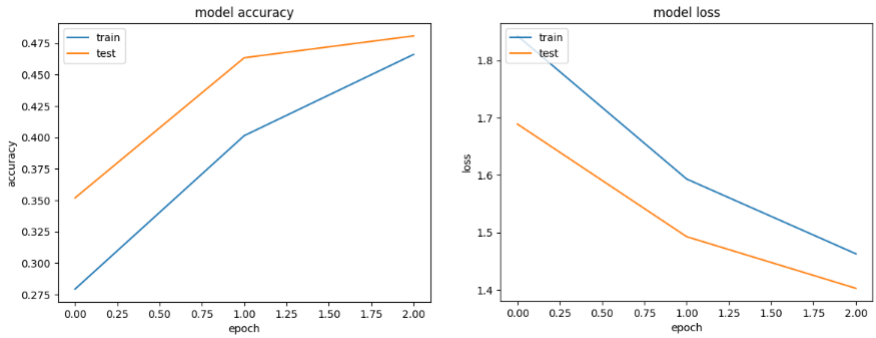


Figura 8- Exemplo de gráfico gerado via matplotlib pela arquitetura AlexNet apresentando epochacc e epochloss - Cojunto de dados: CIFAR10 - **Hiperparâmetros alterados**

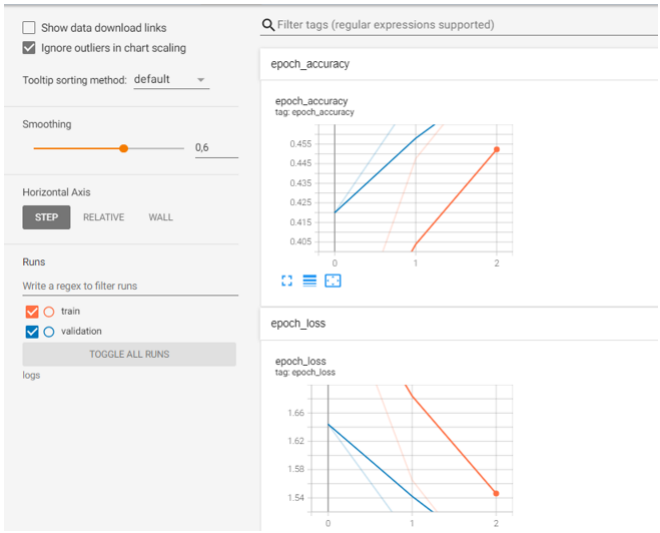


Figura 9 - Gráfico gerado via tensorboard pela arquitetura VGG apresentando epochacc e epochloss - Cojunto de dados: CIFAR10

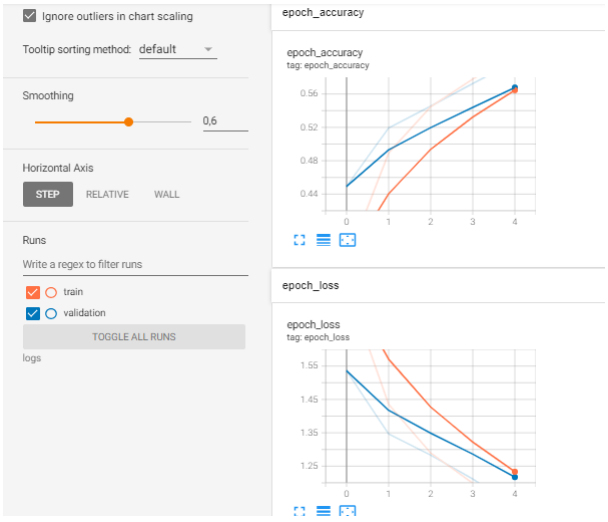


Figura 10 - Gráfico gerado via tensorboard pela arquitetura VGG apresentando epochacc e epochloss - Cojunto de dados: CIFAR10 - **Hiperparâmetros alterados**

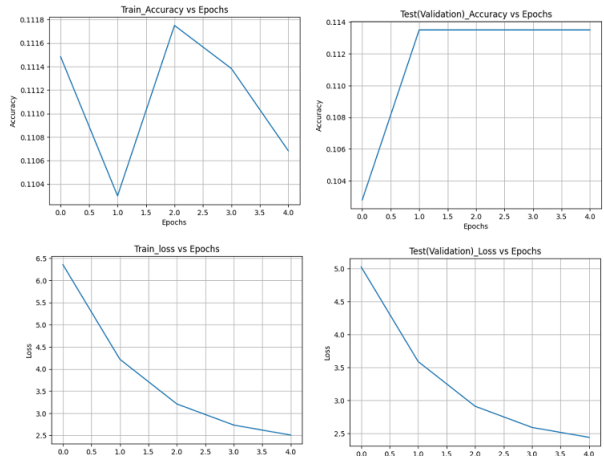


Figura 11 - Gráfico gerado via Matplotlib pela arquitetura VGG apresentando epochacc e epochloss - Conjunto de dados: MNIST

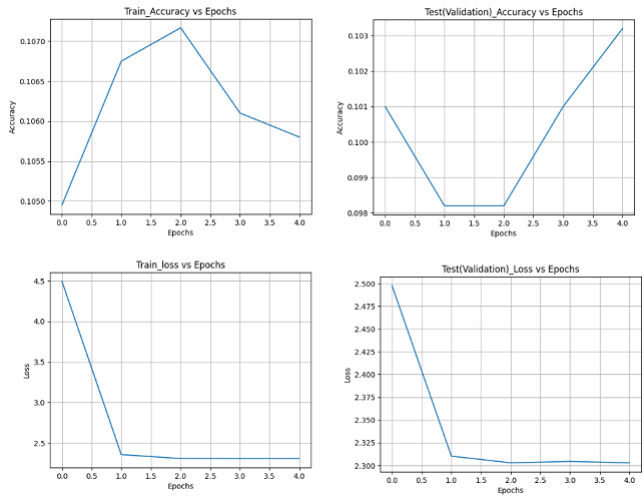


Figura 12 - Gráfico gerado via Matplotlib pela arquitetura VGG apresentando epochacc e epochloss - Conjunto de dados: MNIST - **Hiperparâmetros alterados**

5 Conclusão

Alguns pontos interessantes que pudemos identificar com o trabalho:

- LeNet é uma arquitetura com uma certa média de performance independente do conjunto de dados. (Algo em torno de 10 a 15m)

- Na arquitetura LeNet analisando o conjunto de dados MNIST, quanto maior for o nº de epochs e batch size, maior o nível de acurácia obtido. Não acontecendo o mesmo cenário com o outro conjunto de dados escolhido para o trabalho.

- AlexNet analisando o conjunto de dados CIFAR10 foi a mais demorada em questão de finalização de modelos e geração de gráficos. (2 horas e 4 minutos / epochs = 5 - Conjunto de dados: CIFAR10)

- Na arquitetura VGG houve uma certa singularidade entre os dados de treino e teste de acordo com o gráfico 10.

- A arquitetura VGG apresentou problemas de incompatibilidade com a biblioteca tensorflow.(Necessário comando "!kill PID"na porta 6006 para visão final em forma de gráfico)

- O conjunto de dados CIFAR-10 devido a sua composição é bem dificil de se conseguir um bom nível de acurácia.

De todo modo e de acordo com os resultados, podemos constatar que as arquiteturas escolhidas atendem perfeitamente para esse tipo de atividade ou a um propósito maior da área.

Link para repositório: `https://github.com/Bmartins25/Redes_Neurais_Modelos`

6 Referências

[1] - Introdução as Redes Neurais Convolucionais - http://deeplearningbook.com.br/introducao-as-redes-neurais-convolucionais

[2] - Gradient-based learning applied to document recognition - http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

[3] - Dive into Deep Learning - https://pt.d2l.ai/chapter_convolutional-modern

[4] - Redes Neurais Convolucionais - https://www.linkedin.com/pulse/introdu%C3%A7%C3%A3o-%C3%A0s-redes-neurais-convolucionais-ruan-costa/?origina
pt

[5] - 10 principais arquiteturas de Redes Neurais - https://www.deeplearningbook.com.br/

[6] - AWS - Neural Network - https://aws.amazon.com/pt/what-is/neural-network

[7] - FahsionMNIST and Cifar-10 - https://en.wikipedia.org/wiki

[8] - Breve historia das redes neurais - https://www.deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/#:~:text=Em%201943%2C%20o%20neurofisiologista%20Warren,neural%20simples%20usando%20circuitos%20el%C3%A9tricos.

[9] - Hiperparâmetros - https://blog.dsbrigade.com/hiperparametros-por-que-s

[10] - Overfitting e regularizacao - https://www.deeplearningbook.com.br/overfitting-e-regularizacao-parte-2/