Universidade Estadual de Ponta Grossa

Deep Learning

Jonathan de Matos

Universidade Estadual de Ponta Grossa

Conteúdo

- Pytorch + Anaconda
- OpenCV
- PIL (Python Imaging Library)
- Convoluções
- Multi Layer Perceptron
  - Fully Connected Layers
- Redes mais populares
- Treinamento
- Avaliação de resultados

Universidade Estadual de Ponta Grossa

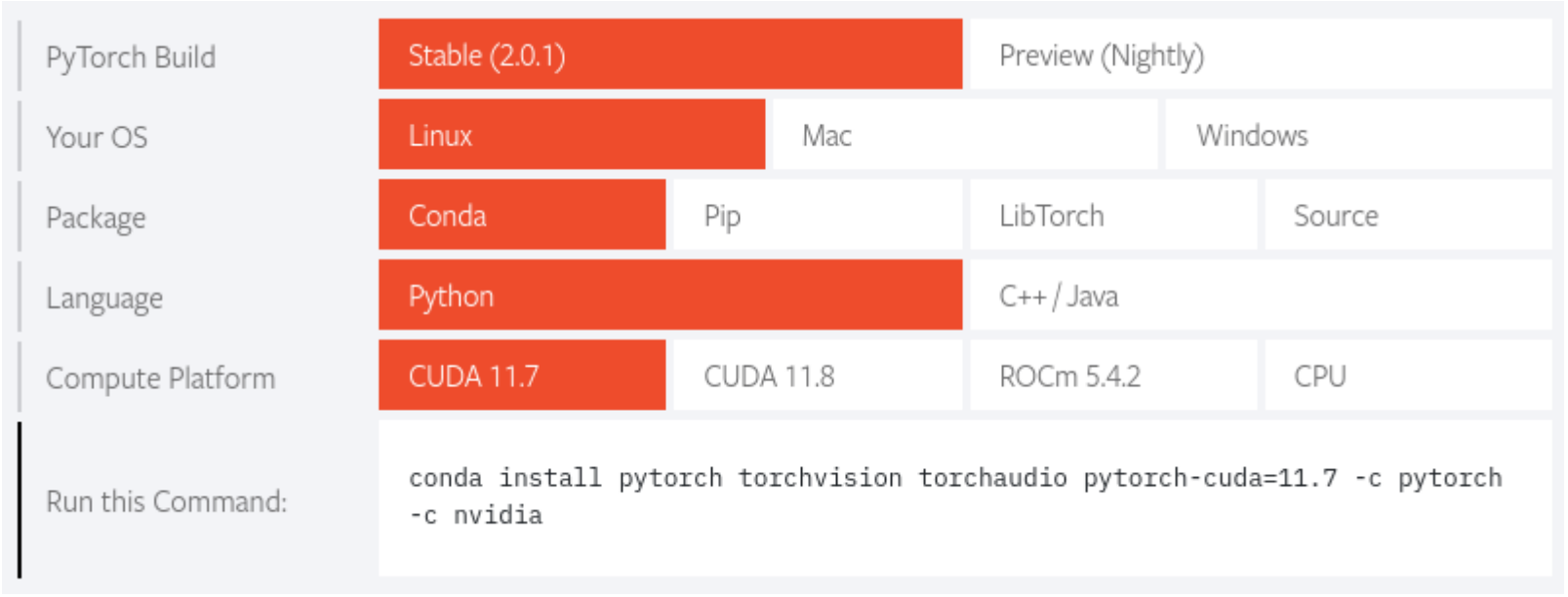
Pytorch

- É uma biblioteca que facilita o uso de tensores tanto em CPU quanto em GPU
- Tensores são entidades geométricas criadas para generalizar a noção de escalares, vetores e matrizes
- É possível realizar operações sobre os tensores, simples ou complexas, tanto no CPU quanto GPU de forma transparente
- Quando usado apenas com processamento na CPU, a instalação é simples, porém o desempenho é reduzido
- A instalação usando GPU requer a instalação das versões corretas do CUDA (Compute Unified Device Architecture) ou ROC (Radeon Open Compute)

Universidade Estadual de Ponta Grossa

Pytorch

- Instalação
  - https://pytorch.org/features/



Universidade Estadual de Ponta Grossa

Pytorch + Anaconda

- É uma distribuição de Python para uso científico
  - Facilidade de uso multiplataforma (Linux, Windows e macOS)
- Vantagens
  - Gerenciamento simples de pacotes e dependências
  - Criação de ambientes congelados com versões de pacotes
  - JupiterLab e Jupiter Notebook
  - Instalação em servidores e programação remota

Universidade Estadual de Ponta Grossa

OpenCV

- Biblioteca para visão computacional
  - Publicada inicialmente em 2000
- C++, Python e JS
- Fornece suporte a diversos formatos de imagens
- Permite várias formas de pré-processamento de imagens
  - Transformações
  - Thresholding
  - Smoothing
  - Transformações morfológicas
  - Gradientes
  - Bordas
  - Contornos
  - Histogramas
  - Segmentação
  - Características
  - Estereoscopia

Universidade Estadual de Ponta Grossa

PIL

- Python Imaging Library
- Leitura e gravação de imagens
- Conversão entre numpy e imagens
- Conversão entre tipos de imagens
- Geração de imagens
- Propriedades de imagens
- Filtros
- Visualização

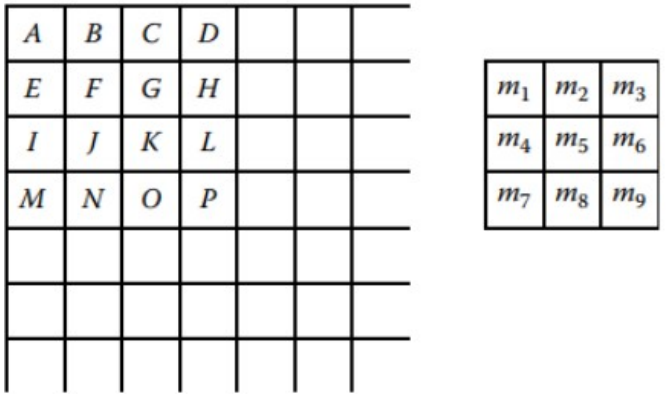
Universidade Estadual de Ponta Grossa

Deep learning

- A tecnologia de machine learning tem auxiliado em muitos aspectos a sociedade atual
  - Identificação de objetos, speech to text, busca, CBIR, etc
- Métodos tradicionais de ML são limitados na habilidade de processar dados naturais em sua forma pura
  - Extração de característica
- Representation learning → Deep Learning, múltiplos níveis de representações com crescente abstração
  - 1ª camada: bordas em várias orientações e suas localizações
  - 2ª camada: arranjos de bordas, padrões
  - 3ª camada: combinações de padrões que podem ser encontrados em objetos
  - 4ª camada … : combinações de objetos em objetos
- Não precisam ser criados ou configurados por pessoas, são alterados de acordo com os dados recebidos
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Convoluções

- É uma função resultante da soma do produto de duas outras funções em uma determinada região
- Convolução, máscara ou filtro
- Afeta diretamente a imagem
- Kernel de filtro ou máscara (matrix 3x3, por exemplo)



Universidade Estadual de Ponta Grossa

Convoluções

- Sobreposição da máscara sobre a imagem

$$R = (A \times m_1 + B \times m_2 + C \times m_3 + E \times m_4 + F \times m_5 + G \times m_6 + I \times m_7 + J \times m_8 + K \times m_9) \Big/ S$$

- O valor de R é colocado no lugar de F da nova imagem
- A mesma operação e repetida movendo-se para o próximo conjunto de pixels ao lado e o próximo e próximo, até chegar a borda, o processo então se repete na próxima linha
- Os valores de *m* resultam em efeitos diferentes na imagem de destino

Universidade Estadual de Ponta Grossa

Convoluções

- OpenCV
  - cv.filter2D(src=[imagem], ddepth=-1, kernel=[kernel])

```
import cv2
import numpy as np


image = cv2.imread('uepg_sm.jpg')


if image is None:
    print('Could not read image')

kernel1 = np.array([[-1,-1,-1],
                    [-1, 8,-1],
                    [-1,-1,-1]])


filtered = cv2.filter2D(src=image, ddepth=-1, kernel=kernel1)


cv2.imwrite('filtered.jpg', filtered)
```

Universidade Estadual de Ponta Grossa

Convoluções

- Exemplos (filter2D) (filter2d_opencv.py)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Identity kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge detection

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Sharpen kernel

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Box blur

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Gaussian blurr kernel

Universidade Estadual de Ponta Grossa

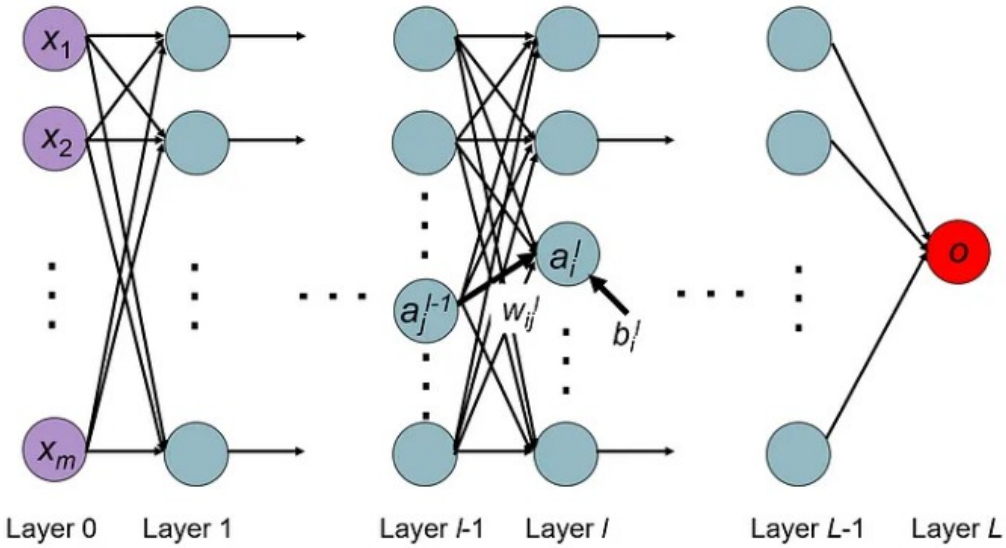
PIL

- Python Imaging Library (pil_example.py)
  - Muito utilizada para abrir, fechar, verificar valores, converter, gerar imagens
  - Diferente do OpenCV que é voltado para aplicar certo processamento sobre as imagens, a PIL nos ajuda muito junto com o Pytorch para verificar o que acontece durante todo processamento

Universidade Estadual de Ponta Grossa

Multi Layer Perceptron

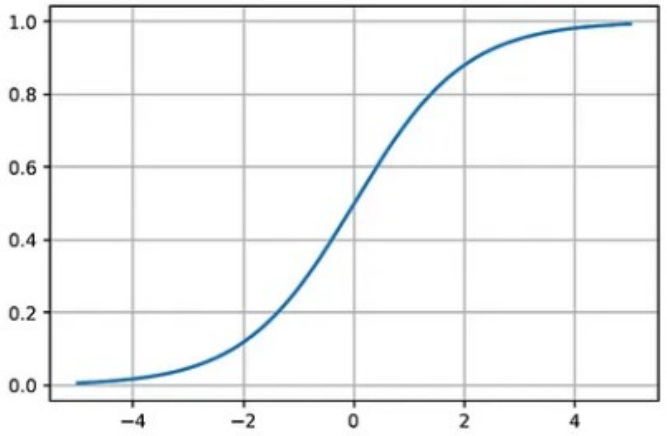
- É uma rede neural com ao menos três camadas: entrada, camada oculta e saída [1]



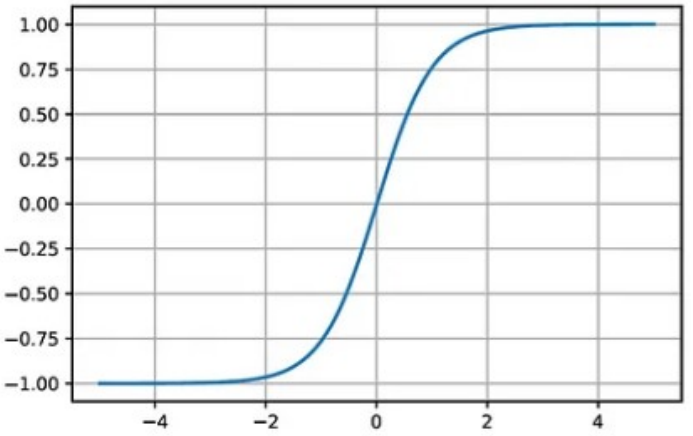
Universidade Estadual de Ponta Grossa

Multi Layer Perceptron

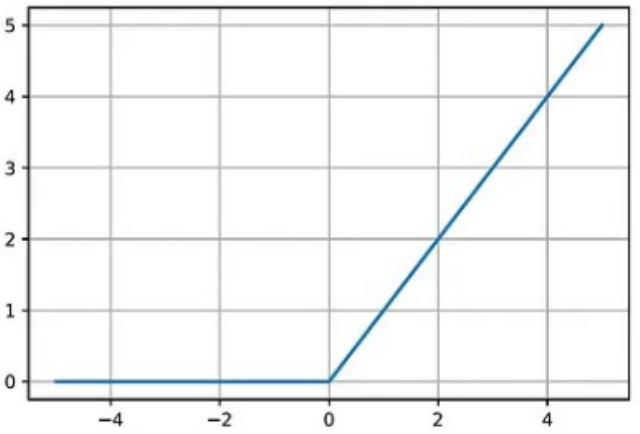
- Camadas de ativação



$$f(z) = \frac{1}{1 + e^{-z}}$$



$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$



$$f(z) = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Universidade Estadual de Ponta Grossa

Multi Layer Perceptron

- Implementação usando Sklearn (mlp_example.py)
  - https://scikit-learn.org/stable/index.html
- Criacao de dois datasets (treino e teste) baseados em “luas”
- Treino do classificador
- Relatório
- Representação gráfica dos dados
  - Foi possível representar graficamente pois os dados tinham apenas duas características
  - Imagens puras tem H x W x C características
  - A camada de entrada teria H x W x C neurônios

Universidade Estadual de Ponta Grossa

Redes convolucionais

- Aplicam convoluções como se fossem os filtros2D do OpenCV sucessivamente (convolucao.py)
- É possível verificar o efeito de um filtro de uma rede (pré-treinada) usando o próprio OpenCV
- A VGG (Visual Geometric Group) é uma Rede Neural Convolucional proposta em 2015 [2]
- Ela possui entre 133 e 144 milhões de parâmetros e somente filtros 3x3

Universidade Estadual de Ponta Grossa

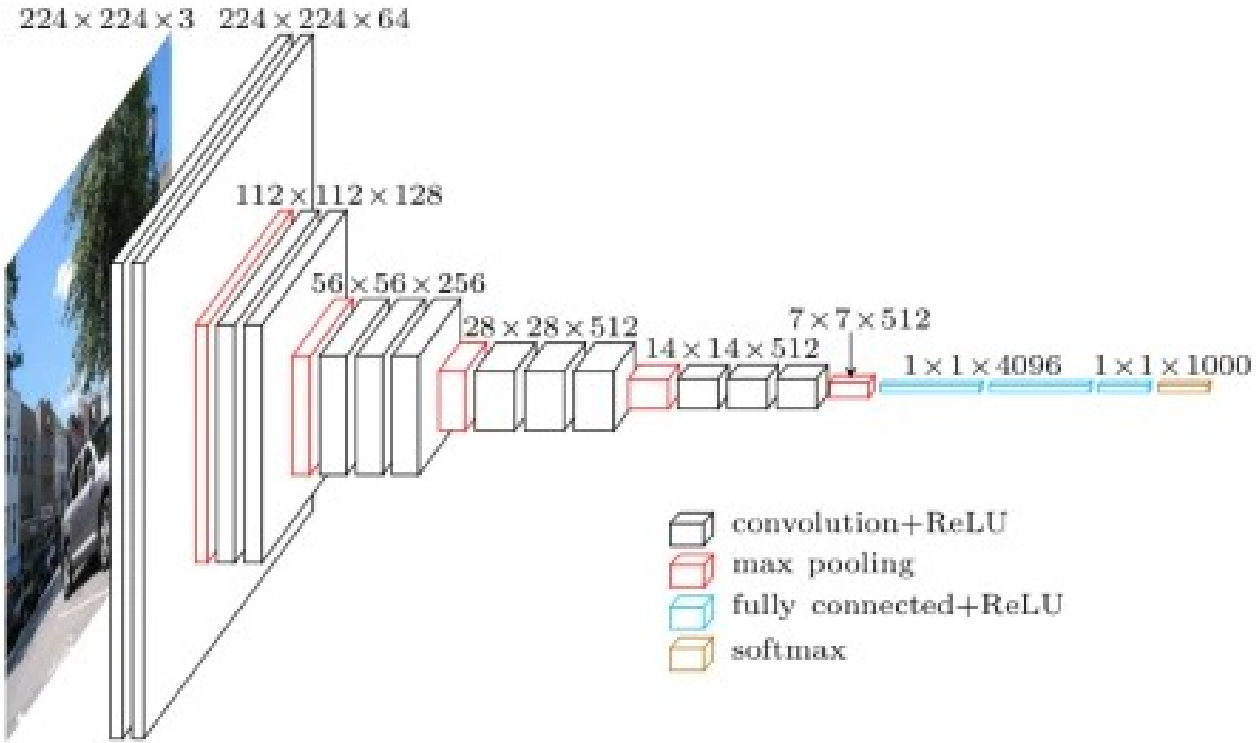
Redes convolucionais

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Universidade Estadual de Ponta Grossa

VGG

- Formato



Universidade Estadual de Ponta Grossa

VGG

- A sequência de camadas convolucionais mais as camadas de MaxPooling vão diminuindo o tamanho da imagem (H x W), porém aumentando a quantidade de “canais”
- Os “canais” no meio da VGG acabam sendo as combinações de bordas, padrões e objetos que no final são colocados em uma espécie de MLP (Fully Connected Layers) como se fossem as características do nosso exemplo de MLP
- Quando uma imagem tem N objetos de um tipo, M objetos de outro tipo, L objetos de outro tipo, … ela é um gato, um cachorro ou um carro
- No exemplo do código “all_vgg.py” podemos ver a imagem diminuindo, a filtragem de bordas e a formação de padrões e combinações deles até a última camada convolucional

Universidade Estadual de Ponta Grossa

Outros modelos

- A VGG é uma rede relativamente antiga, grande e com limitações
- Existem outros modelos, inclusive disponíveis no próprio pytorch já pré-treinados:
  - https://pytorch.org/vision/stable/models.html
  - Exemplos: AlexNet, ConvNeX, DenseNet, EfficientNet, EfficientNetV2, GoogLeNet, Inception V3, MaxVit, MNASNet, MobileNet V2, MobileNet V3, RegNet, ResNet, ResNeXt, ShuffleNet V2, SqueezeNet, SwinTransformer, VisionTransformer, Wide ResNet

Universidade Estadual de Ponta Grossa

Treinamento completo

- O treinamento completo é realizado fornecendo todo o conjunto de treinamento para a rede em modo treinamento, ou seja, calculando as derivadas parciais em relação a todos os pesos a cada imagem fornecida
- A cada pedaço do conjunto de treino (**batch**) os pesos da rede são ajustados com base nas derivadas parciais e a taxa de aprendizado
- Como o conjunto de treino é muito grande, pode ser que nos últimos pedaços dele, a rede já tenha “esquecido” como classificar as primeira imagens fornecidas, então todas as imagens de treino devem ser fornecidas novamente, porém, em geral, em outra ordem
- Cada vez que todas as imagens de treino passam pela rede, temos uma **época**

Universidade Estadual de Ponta Grossa

Treinamento completo

- Ao final de cada época, podemos fornecer um conjunto de imagens que o modelo ainda não viu, assim verificamos se ele está aprendendo ou não, este é o conjunto de validação
- Podemos treinar o modelo por um número grande de épocas ou então parar o treinamento quando a taxa de acerto no conjunto de validação não conseguir melhorar mais

Universidade Estadual de Ponta Grossa

Treinamento completo

- O exemplo “mnist.py” é um código simplificado do site do pytorch que mostra o treinamento completo para o conjunto de dados MNIST, que reconhece caracteres escritos a mão
- Usando a biblioteca PIL, é possível ler novas imagens e testar a capacidade da rede treinada

Universidade Estadual de Ponta Grossa

Referências

- [1] https://towardsdatascience.com/multi-layer-perceptrons-8d76972afa2b
- [2] https://arxiv.org/pdf/1409.1556.pdf
- [3] https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/