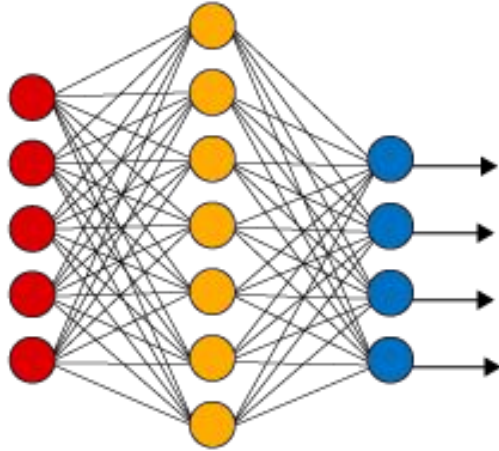# Convolutional Neural Networks and Deep Learning

Overview

- Deep Networks

- Convolution Neural Networks

- Introduction to Google Colab

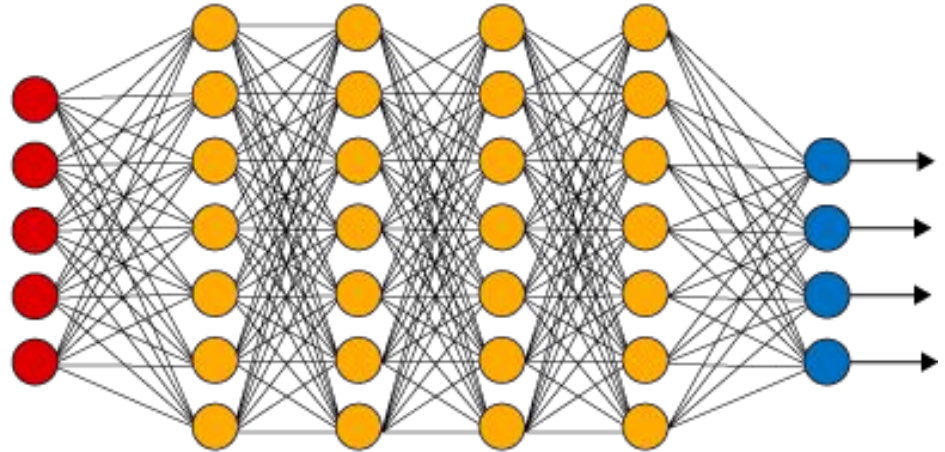- Design and train a simple CNN on MNIST dataset

- Do it yourself

Deep Neural Network

Simple Neural Network

Deep Learning Neural Network

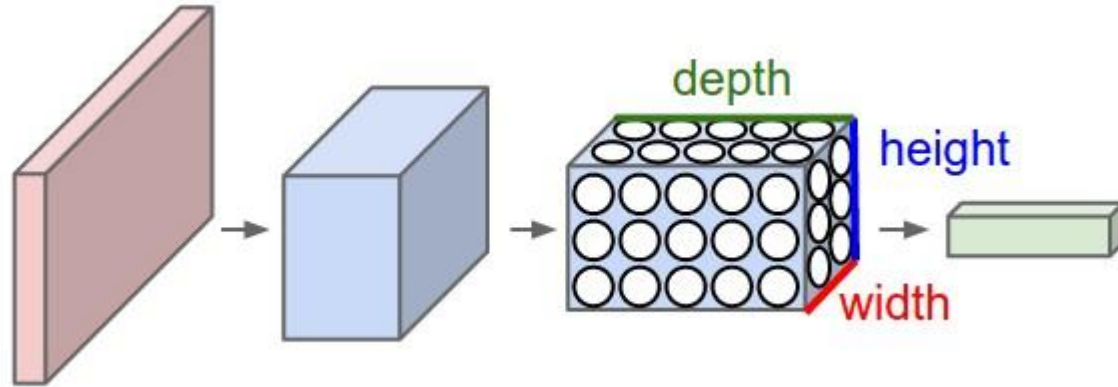● Input Layer   ● Hidden Layer   ● Output Layer

Convolution on Images

$$I = \begin{bmatrix} 5 & 2 & 1 & 0 \\ 4 & 3 & 5 & 2 \\ 2 & 8 & 2 & 2 \\ 8 & 10 & 5 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$
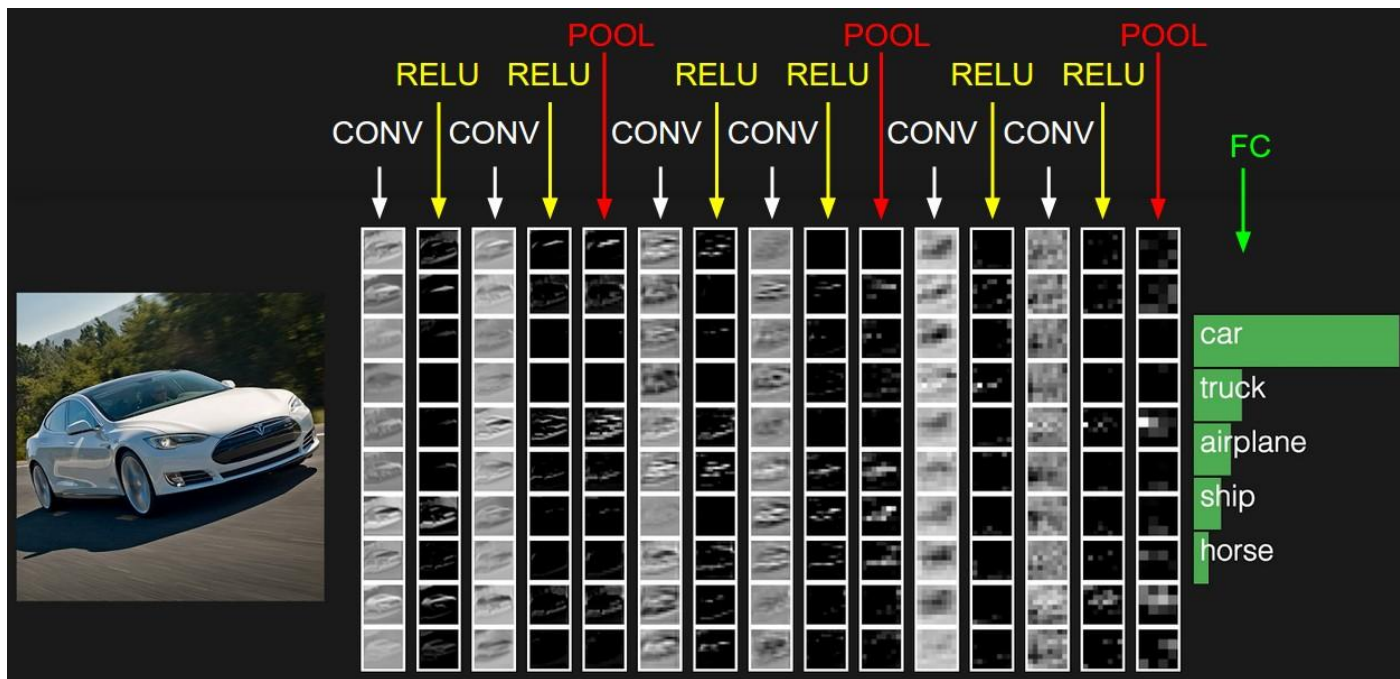
Convolutional Neural Networks

How does a simple CNN look?
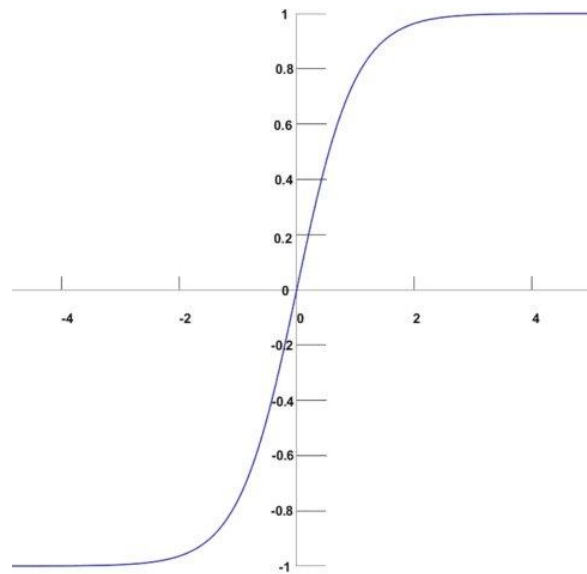
INPUT --> CONV --> ReLU --> POOL --> Fully Connected
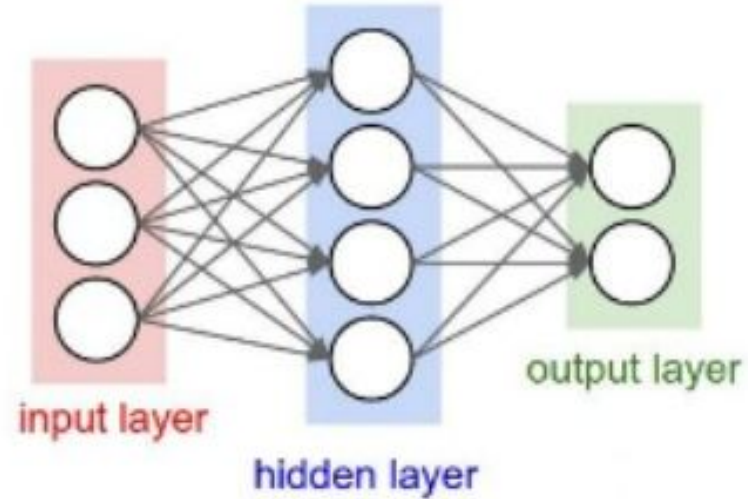
Convolutional Layer

ReLU

$$R(z) = max(0, \ z)$$
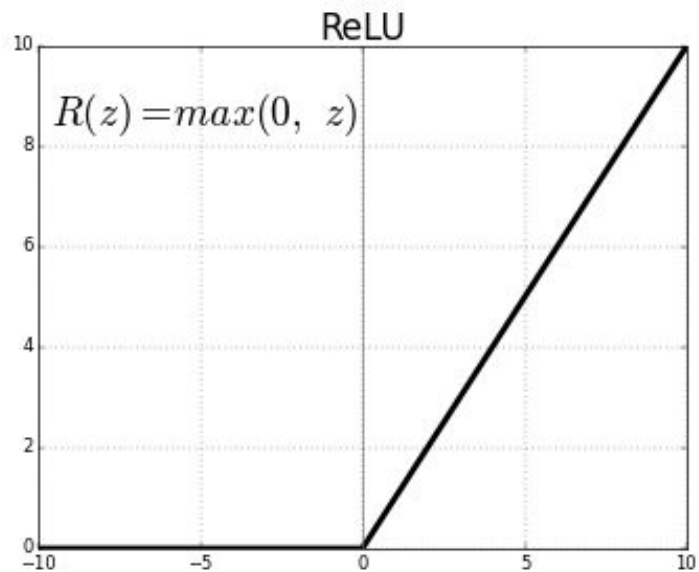
Softmax

Pooling

Max Pool

Filter - (2 x 2)
Stride - (2, 2)

Fully Connected

input layer

hidden layer

output layer

Google Colab

- Cloud based open source service to execute python code

- Developed by Google

- Requires

  - a Google account, Google Chrome

  - stable internet connectivity

- Provide backends with

  - CPU

  - GPU

  - TPU

Machine Learning Libraries

- Data Processing (NumPy, Pandas)

- Data Visualization (Matplotlib)

- Neural Network Architecture (Keras)

- Neural Network Computation (Tensorflow, Theano, CNTK)

Tensorflow Vs. Keras

- **Keras (Frontend)**
    - Open source neural network library
    - written in python

- **Tensorflow (backend)**
    - Open-source library for dataflow and differentiable programming
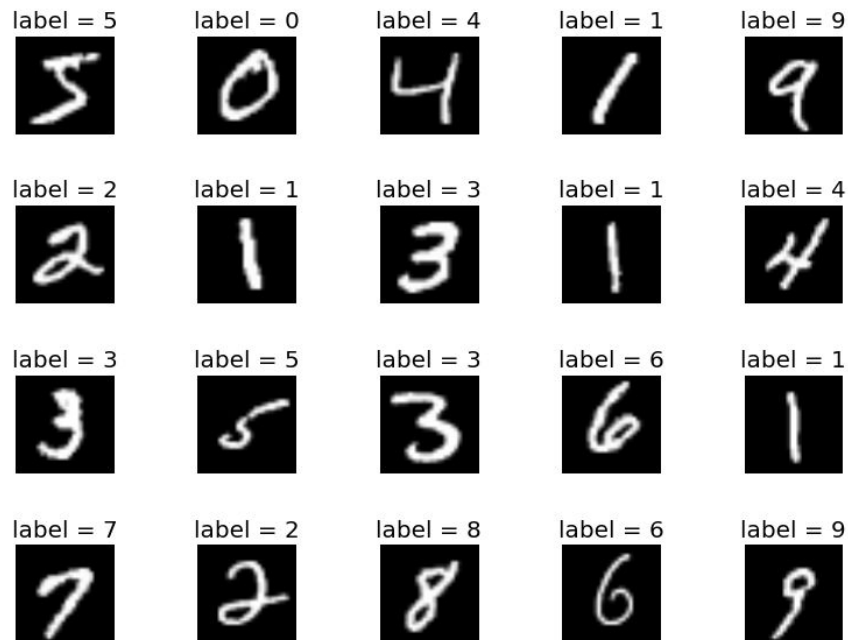   - Create graph of tensor computations

MNIST

Figure : Some images from MNIST Dataset with labels

Pre-processing and Training

- Resizing
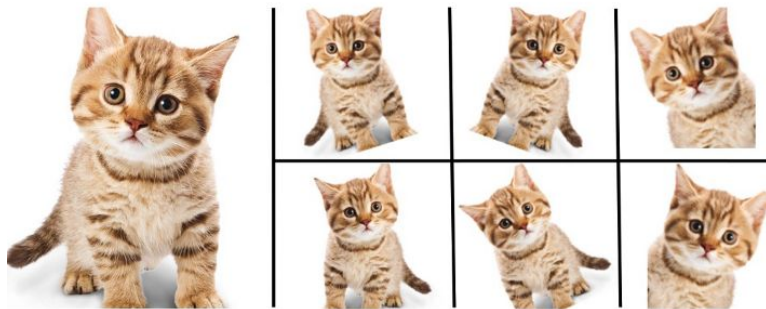- Normalisation of data
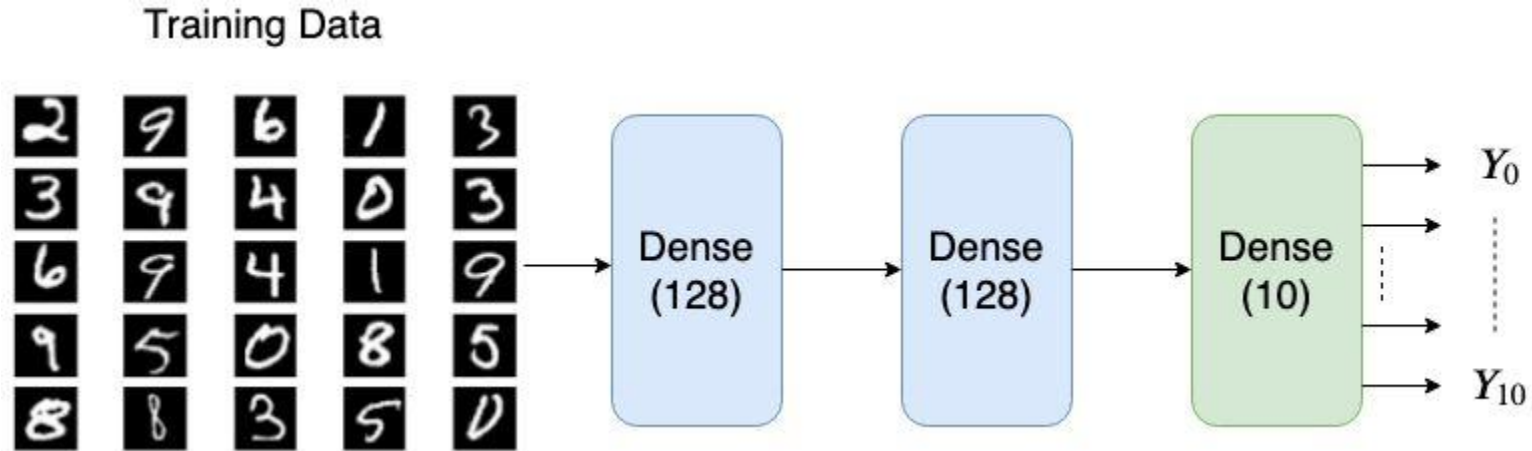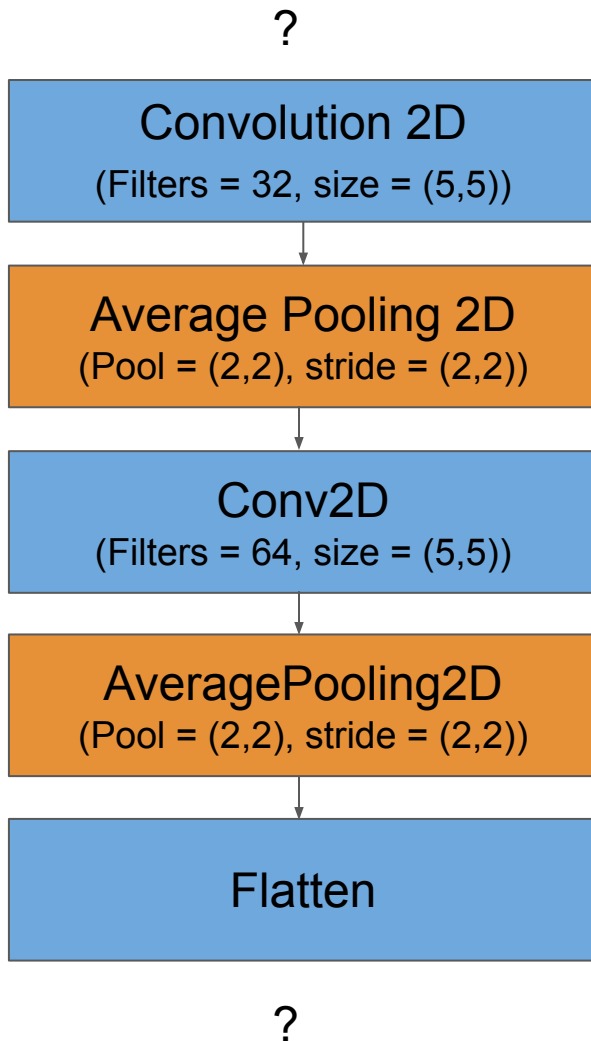- Denoising
- Data augmentation

Figure : Example of Data Augmentation

Simple DNN Model

- Both Hidden layers use **'ReLu'** as activation function

DIY

?

Convolution 2D
(Filters = 32, size = (5,5))

Average Pooling 2D
(Pool = (2,2), stride = (2,2))

Conv2D
(Filters = 64, size = (5,5))

AveragePooling2D
(Pool = (2,2), stride = (2,2))

Flatten

?

Transfer Learning

- Instead of training our own model from scratch

- Take the feature extractor from previously trained network and fine tune the classifier

- Other models:

    - ResNet, VGG16, VGG19, Inception etc.

Machine Learning with Custom Dataset

- Create a dataset

- Apply some preprocessing (if required)

- Design a network as per your application

- Train model

Appendix

Figure. Weights

Thank You