

Introduction to the basics of AI - S9-10

Z. TAIA-ALAOUI

Outline for today's course

- Revision Of Deep Learning Principles
- Convolutional Neural Networks
- Implementation

Vocabulary

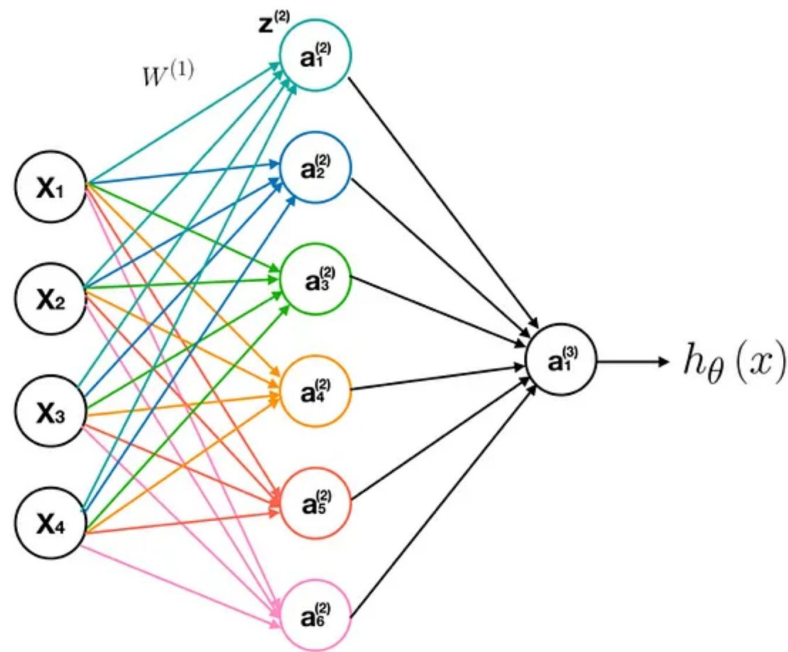
Learning:

A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E. [Tom Mitchell] ([link](#))

Learning

- Forward Propagation

$$W^T X = \begin{bmatrix} \theta_{11}^{(1)} & \theta_{12}^{(1)} & \theta_{13}^{(1)} & \theta_{14}^{(1)} \\ \theta_{21}^{(1)} & \theta_{22}^{(1)} & \theta_{23}^{(1)} & \theta_{24}^{(1)} \\ \theta_{31}^{(1)} & \theta_{32}^{(1)} & \theta_{33}^{(1)} & \theta_{34}^{(1)} \\ \theta_{41}^{(1)} & \theta_{42}^{(1)} & \theta_{43}^{(1)} & \theta_{44}^{(1)} \\ \theta_{51}^{(1)} & \theta_{52}^{(1)} & \theta_{53}^{(1)} & \theta_{54}^{(1)} \\ \theta_{61}^{(1)} & \theta_{62}^{(1)} & \theta_{63}^{(1)} & \theta_{64}^{(1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \\ z_4^{(1)} \\ z_5^{(1)} \\ z_6^{(1)} \end{bmatrix} = Z^{(2)}$$

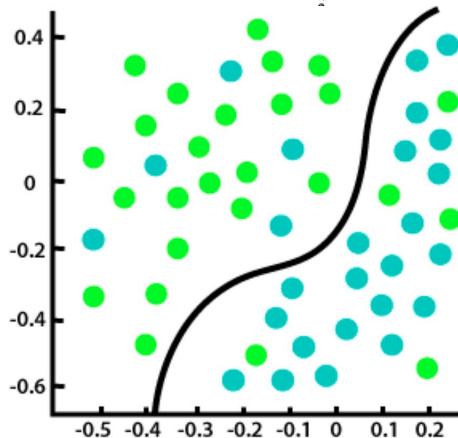


Loss Function

- Output Estimate: $\hat{y} = \sigma(w^T x + b)$

Cross-Entropy

$$H(p, q) = - \sum_i p_i \log q_i = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

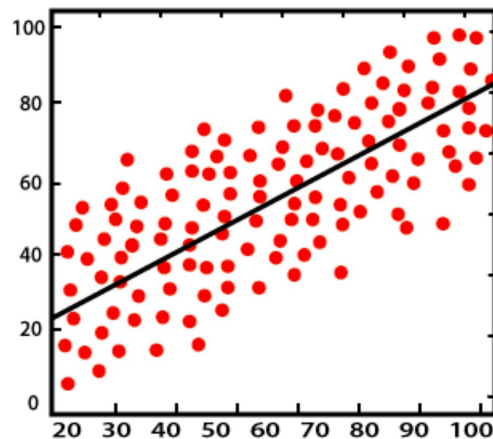


$$p \in \{y, 1 - y\} \text{ and } q \in \{\hat{y}, 1 - \hat{y}\}$$

Classification

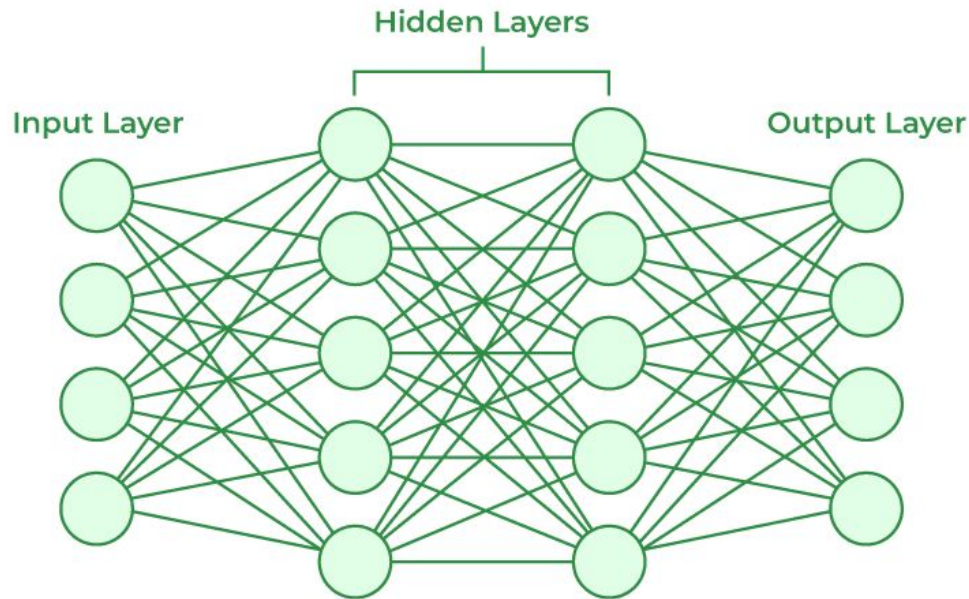
Mean Absolute Error

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$



Regression

Deep Networks



$$\mathbf{z} = \mathbf{W}^{(1)}\mathbf{x}$$

$$\mathbf{h} = \phi(\mathbf{z})$$

$$\mathbf{o} = \mathbf{W}^{(2)}\mathbf{h}$$

Loss

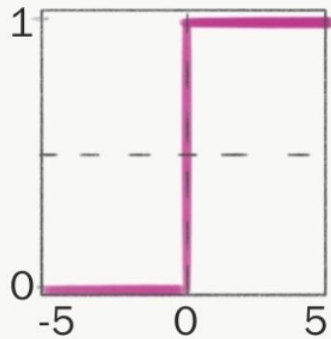
$$L = l(\mathbf{o}, y)$$

$$\hat{o} = \sigma(w^T x + b)$$

Activation Functions

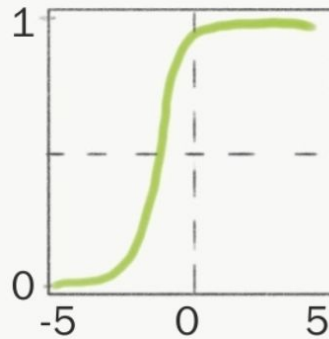
Activation Functions

Threshold

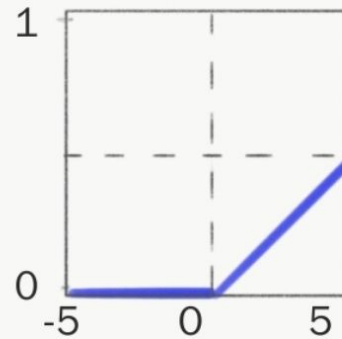


No defined
gradient

Sigmoid

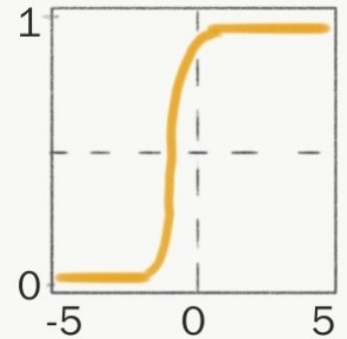


Relu*



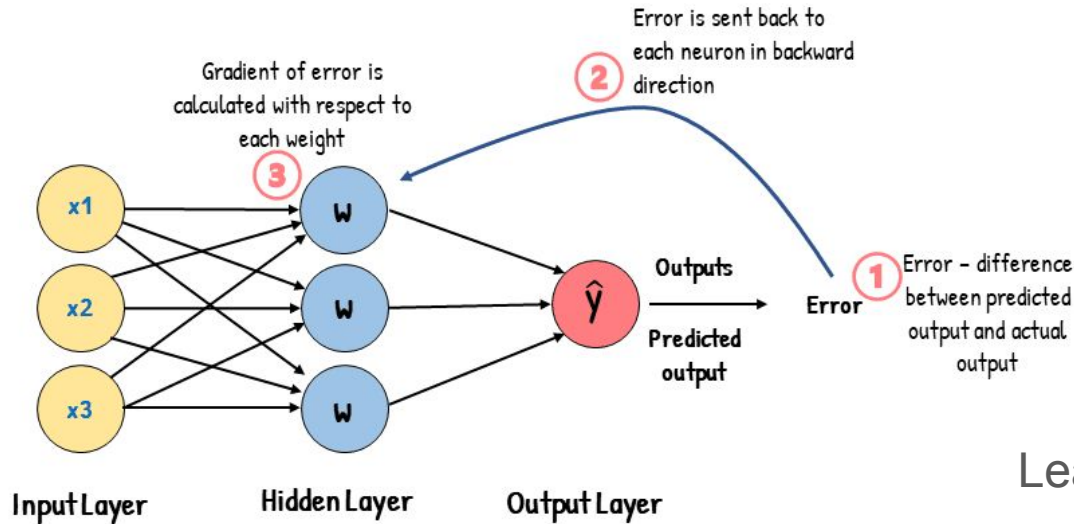
Default

Tan H



Learning with Gradient Descent

Backpropagation



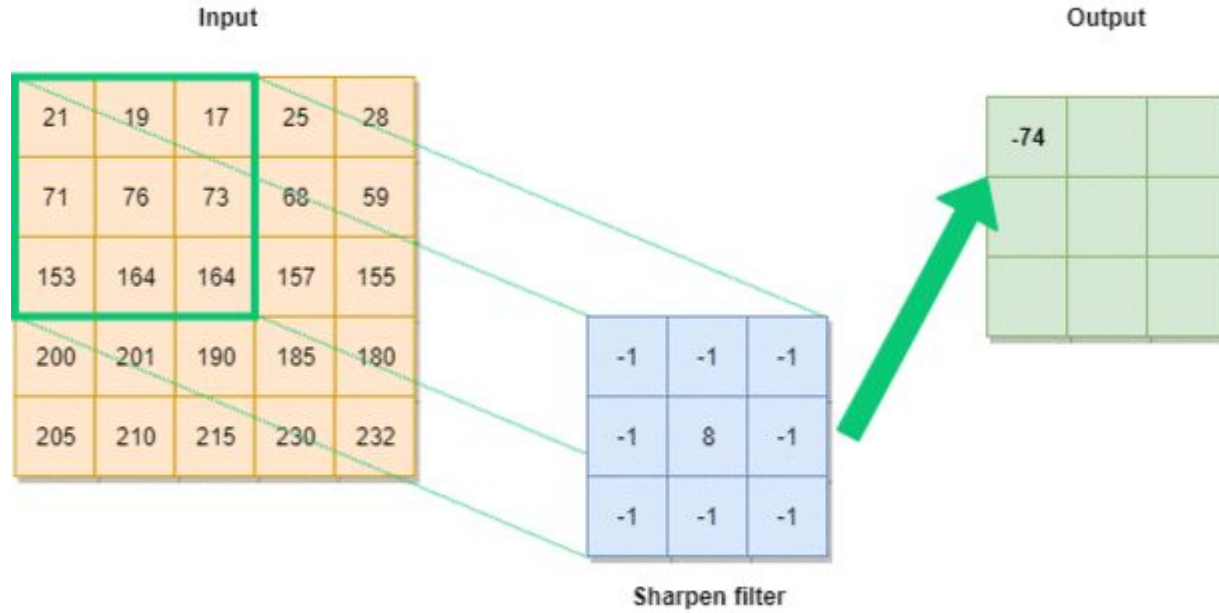
$$w_{ij}^{(k)} = w_{ij}^{(k)} - \eta \frac{\partial L}{\partial w_{ij}^{(k)}}$$

$$b_i^{(k)} = b_i^{(k)} - \eta \frac{\partial L}{\partial b_i^{(k)}}$$

Learning Rate





Convolutional Neural Networks

- Convolution

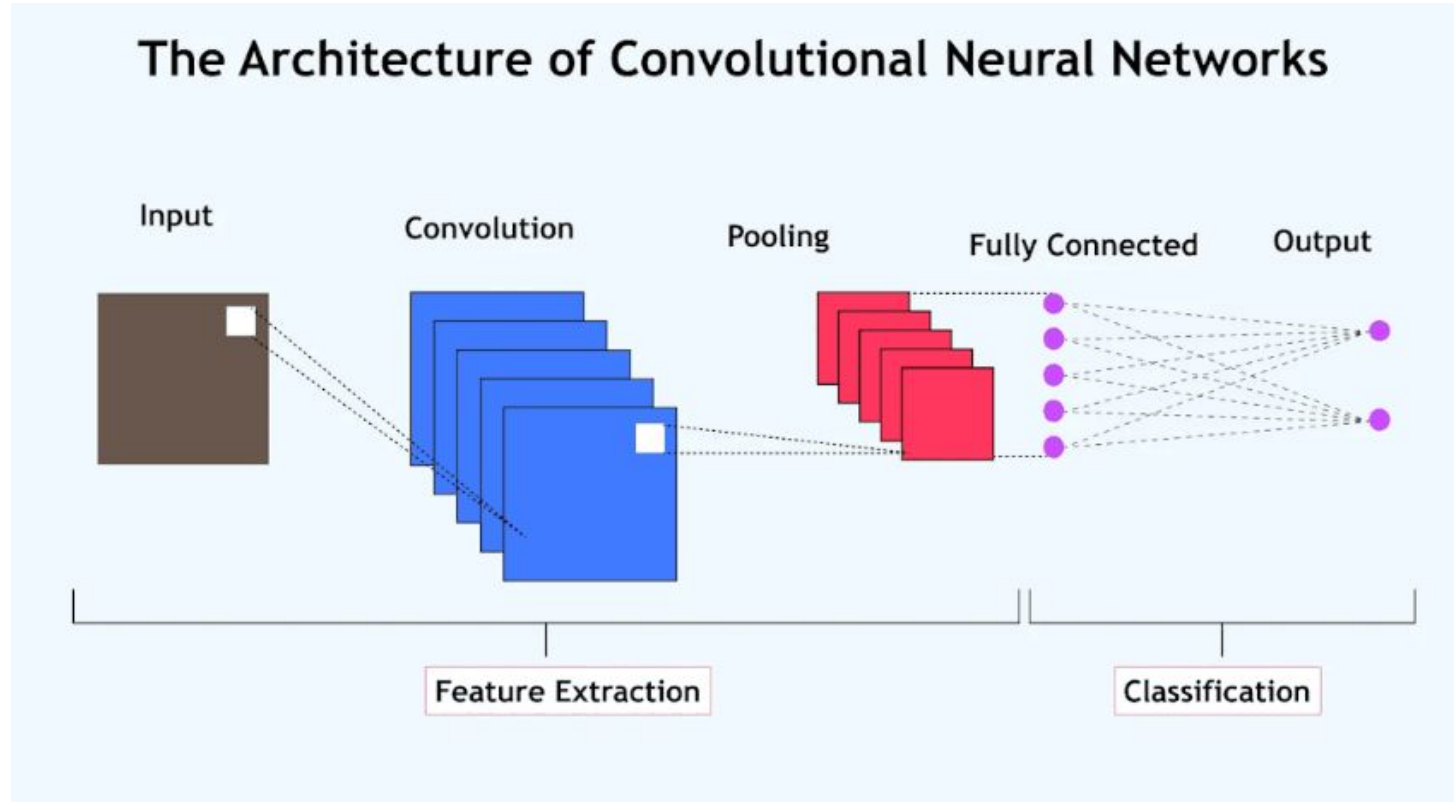


Convolutional Neural Networks

- Convolution Filters = Kernels

<i>Original</i>	<i>Gaussian Blur</i>	<i>Sharpen</i>	<i>Edge Detection</i>
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

Convolutional Neural Networks



Convolutional Neural Networks

- Pooling
 - Dimensionality Reduction
 - Prevents Overfitting
 - Feature Extraction

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

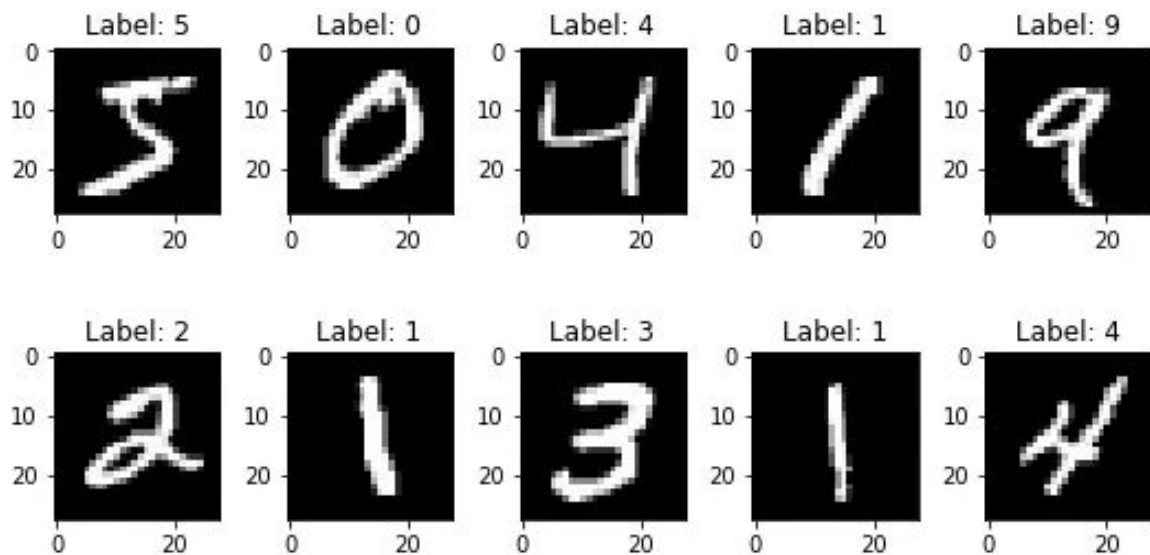
Max Pool
→
Filter - (2 x 2)
Stride - (2, 2)

9	7
8	6

Convolutional Neural Networks

- How to Train a CNN according to your problem ?
 - What is the nature of your data ? is it sequential or not ? If not, should you impose an order in the variables ?
 - What is the optimal size for the filters ?
 - How many Layers ?
- Bibliography is the key to answering these questions !

Time for Coding



Resources

<https://botpenguin.com/glossary/deep-neural-network>

<https://fidle.cnrs.fr/done/02-CNN-MNIST==K3MNIST2.done==.html>

<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>

<https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e>

<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/>