Mohamed Boudiaf University of M'sila
Faculty of Mathematics and Computer Science

Introduction to Artificial Intelligence

Chapter 5. Deep Learning

Dr. SAID KADRI

Associate Professor "Class A"

Department of Computer Science, Faculty of Mathematics and Informatics,

University Mohamed Boudiaf of M'sila

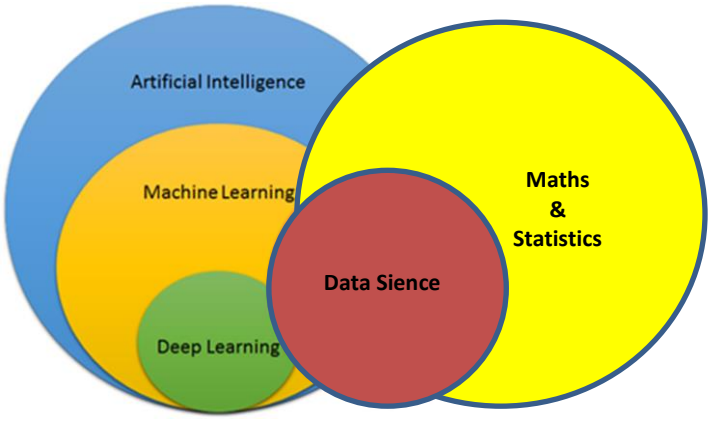
E-mail:kadri.said28@gmail.com

Website:https://kadrisaid28.wixsite.com/sgadri

2020– 2021

Deep Learning

- Deep learning DL is a subfield of Machine Learning ML.
- Build a model that learns to perform classification tasks based on images or texts.



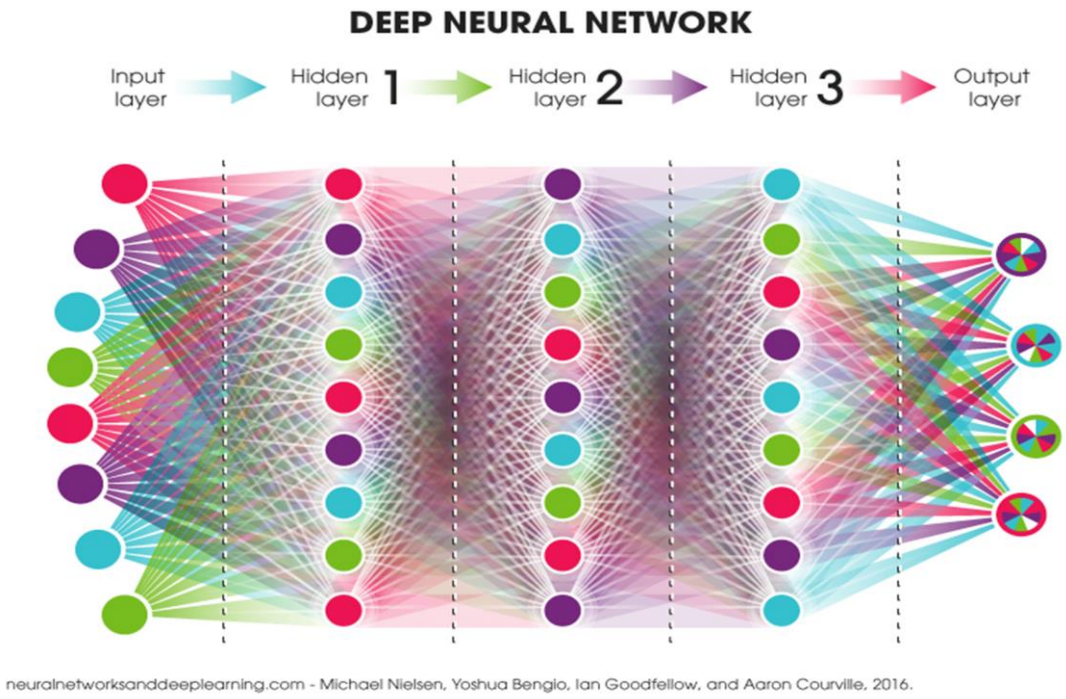
- Only Artificial Neural Networks are used to build the appropriate model.

- The term “Deep” comes from the use of several layers of neurons.

  (Using more layers ==> the network is deeper).

- A DL architecture consists of an input layer, several hidden layers, is an output layer.

- The different layers are interconnected, so that the outputs of each layer become inputs for the next layer.



Machine Learning Vs Deep Learning

- Deep Learning is a subfield of Machine Learning

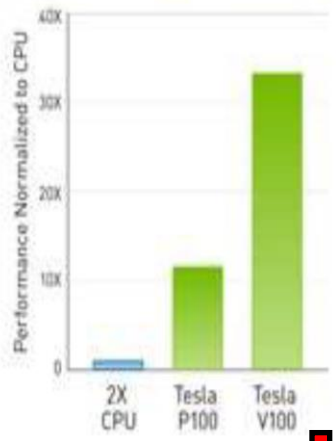
- In machine learning, the feature extraction task is performed manually by experts.

- In deep learning, the built network which takes care of this task automatically when processing thousands or millions of occurrences (Ex: pixels of an image).

Deep Learning Motivation

- Wide availability of Datasets.
- Evolution of graphics equipment (GPUs)
- Relatively reasonable price of Hardware.
- Evolution of processing techniques and methods

Nvidia Tesla V100 avec processeur GV100 :
21,1 milliards de transistors, 12 nm, 120 TFLOPS, 5376 coeurs, 815 mm2



La société Nvidia vient d'annoncer une nouvelle génération de GPU dédié au deep learning, le Tesla GV100. Il intègre 21,1 milliards de transistors dans 5376 cœurs, le tout ayant une puissance de calcul de 120 TFLOPS.

Application Fields

- Image Classification.
- Automatic translation: from a given language to another.
- Games (ex: Chess Game, AlphaGO).
- Recognition Tasks (Face, Gesture, Speech, ...).
- Video Search (Search & Analysis)
- Recommendation engines (Robots, Cars)
- Indexing and Search (Ex: search for documents on the Web)

Deep Learning Requirements

- More data (Large datasets, Big Data)
- Better learning models (e.g., DNN, CNN, RNN, Auto-encoders, transformers, ...)
- Powerful GPU accelerators (Nvidia drive™ px auto-pilot car computer, tegra x1 mobile superchip 256-core GPU | 8-core 64-bit CPU | 1 Tera-FLOPS)

Deep Learning Platforms

1. Caffe

- Interface: C++, Python
- One of the first platforms to support a GPU
- Focused mainly on the field of computer vision (and CNNs)

2. TensorFlow (Google, 2015)

- Interface: Python, C++
- Uses Multi GPUs
- Very popular

3. Keras (Google, 2015)

- Set up on Tensorflow and Theano
- Interface: Python
- Purpose: Provide a simple interface

4. Theano

- Interface: Python
- One of the first platforms to support GPU card

5. Other platforms: Microsoft CNTK, Torch, MXN, Deeplearning4j

Datasets Used for Deep Learning

- ⇨ **ImageNet:** 14 million images (21841 categories), size: 150GB, used for image recognition.
- ⇨ **Open Images**: 9 million URLs for images (size: 500 GB (Compressed), training set: 9,011,219 images, a validation set: 41,260 images, test set: 125,436 images)
- ⇨ **CIFAR10** (size: 170 MB, 60,000 images of 10 classes, 50,000 training images, 10,000 test images).
- ⇨ CIFAR100: (size: 175 MB, 60,000 images of 100 classes, 500 training images and 100 testing images per class)
- ⇨ Fashion-Mnist (size: 30 MB, 70,000 images in 10 classes, 60,000 training images, 10,000 test images)
- ⇨ **IMDB Reviews (NLP, size: 80 MB, 25,000 highly polar movie** reviews for training, and 25,000 for testing)
- ⇨ **WordNet (NLP, Size: 10 MB, 117,000 synsets)**

Some types of deep neural networks

1. A simple Deep Neural Networks DNN

A simple deep neural network called "Deep Neural Network" is a neural network with more than two layers (one input layer, several hidden layers, one output layer). Neural networks of this type use sophisticated mathematical modeling to process data. These models adjust outputs based on inputs.

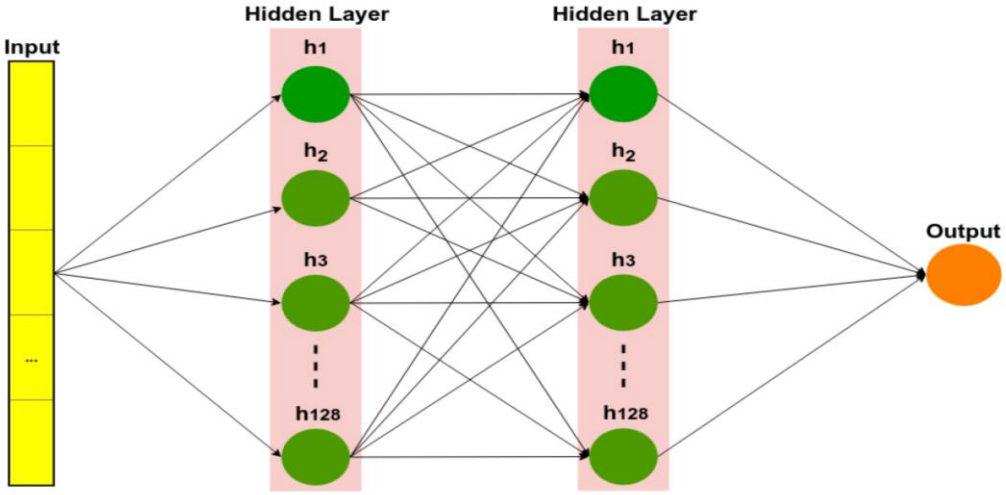


Figure 2. Deep neural network (DNN).

2. Convolutional Neural Networks CNN

A Convolutional Neural Network CNN is a special model of feed-forward neural networks used in different fields such as: computer vision CV, recommender systems RS, and Natural Language Processing NLP. It is a deep neural network architecture. A CNN network is composed of several layers of different types: convolutional layers, sampling layers (pooling or subsampling layers) and a fully-connected layer. The role of the convolution layers is to filter the data they receive to extract the most relevant elements, the role of sampling layers (Pooling Layers) is to reduce the resolution (the dimension) of the elements selected by the layers of convolution which reduces complexity, avoids the phenomenon of overfitting and therefore increases the robustness and performance of the CNN network, especially in the face of parasites.

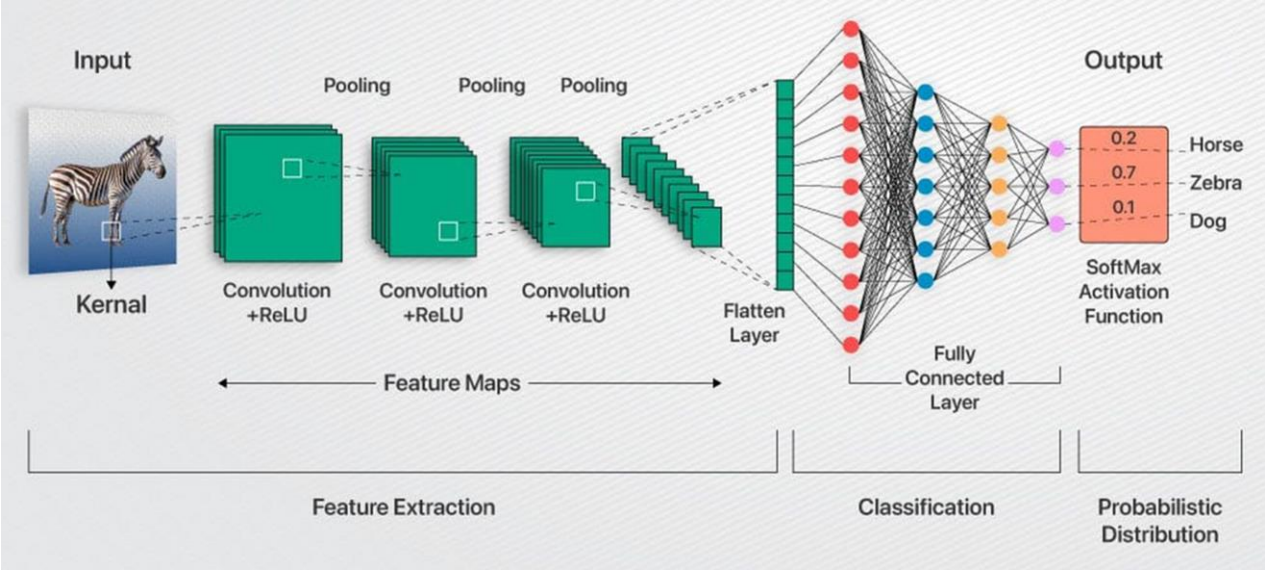


Figure 3. Convolutional Neural Network CNN

3. Recurrent Neural Networks (RNN)

Recurrent neural networks are a class of neural networks whose connections between neurons form a directed cycle which creates feedback loops in the RNN network. The main function of an RNN is the processing of sequential information based on the internal memory captured by the directed cycles. Unlike simple neural networks, RNNs can recall previous

calculations of information and reuse them by applying them to the next element in the sequence of inputs. A special type of RNNs is the so-called LSTM (Long Short-Term Memory) network, which is capable of using long memory as an activation function in hidden layers. The LSTM architecture was proposed by Hochreiter and Schmidhuber (1997).

In Fig 4, the input data is preprocessed to resize it (the same principle of CNNs). The next layer is composed of LSTMs made up of 200 neurons. The final layer is a fully connected layer, composed of 128 neurons for text classification. The last layer uses a sigmoid activation function to reduce a 128-dimensional vector to a single output, knowing that we have two classes to predict (positive, negative).In Fig 4, the input data is preprocessed to resize it (the same principle of CNNs). The next layer is composed of LSTMs made up of 200 neurons. The final layer is a fully connected layer, composed of 128 neurons for text classification. The last layer uses a sigmoid activation function to reduce a 128-dimensional vector to a single output, knowing that we have two classes to predict (positive, negative).

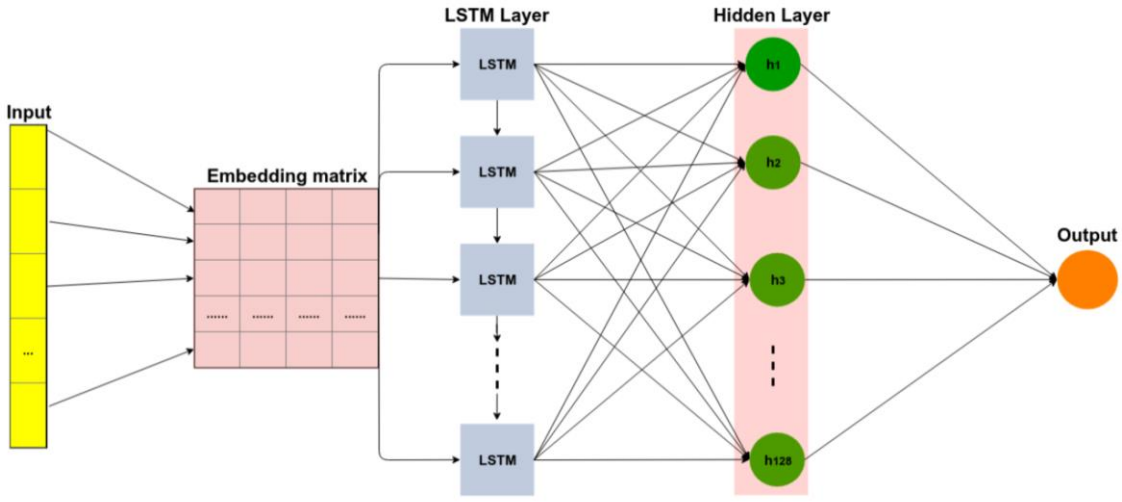


Figure 4. A Long Short-Term Memory Network LSTM

4. Recursive Neural Networks (Recursive Neural Network RecNN):

Is a neural network which can be considered as a generalization of RNNs networks. RecNNs are typically used to learn a directed acyclic graph structure from data. The hidden state vectors of the left and right child nodes in the graph can be used to calculate the hidden state vector of the curre

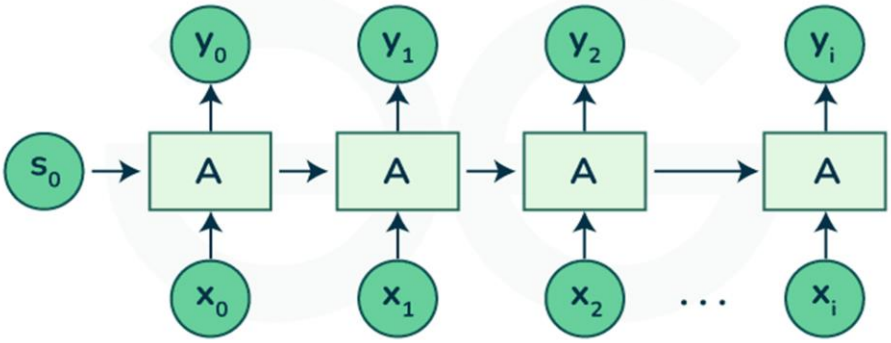


Figure 5. A Recursive Neural Network RecNN

5. Deep Belief Networks (Deep Belief Network DBN)

A DBN network contains several layers consisting of graphical models having both directed and undirected edges. Each network is composed of several layers, each of which is made up of a collection of neurons connected to neurons in the next layer, but not connected to each other. Learning a DBN network is carried out by a specific algorithm called a "greedy layer-wise learning algorithm".

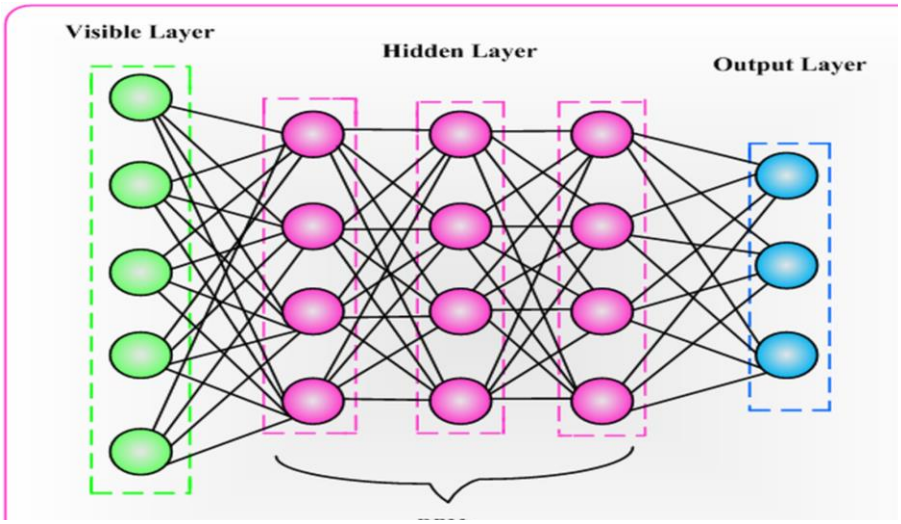


Figure 6. Deep Belief Network DBN

6. Hybrid Deep Learning Network HDLN

Another category of neural network is called a “Hybrid Deep Learning Network”, which combines two or more deep models together, such as: a CNN model, an LSTM model, a probabilistic PNN model or even a RBM model (RBM).

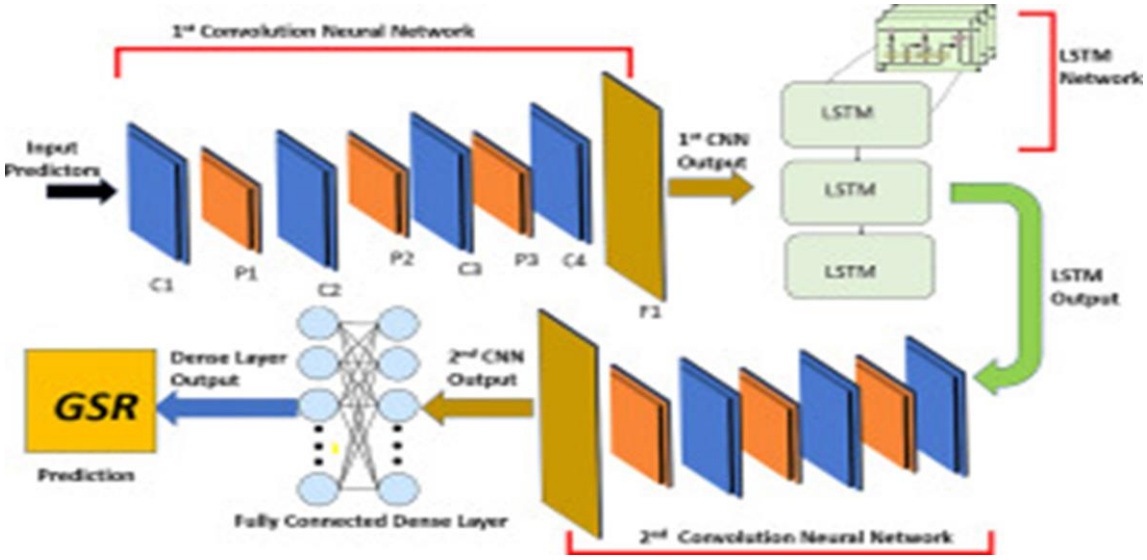


Figure 7. Hybrid Deep Learning Network HDLN

Other Types of ANN

- ⇨ AutoEncoders
- ⇨ Transformers
- ⇨ GAN’s
- ⇨ Others

Convolutional Neural Networks CNNs

- CNN is a deep learning algorithm that often has an image as input (or text).
- CNNs are used to classify a collection of images ➔ detect what each image represents (car, human, cat, boat, plane, ...)

Motivations of CNNs

- For an RGB image of resolution 224X224X3=150.528 pixels ➔ Design more complicated ANN model to process it!!

  ➔ We can focus on the most significant parts (pixels) of the image, which is achieved by CNNs.

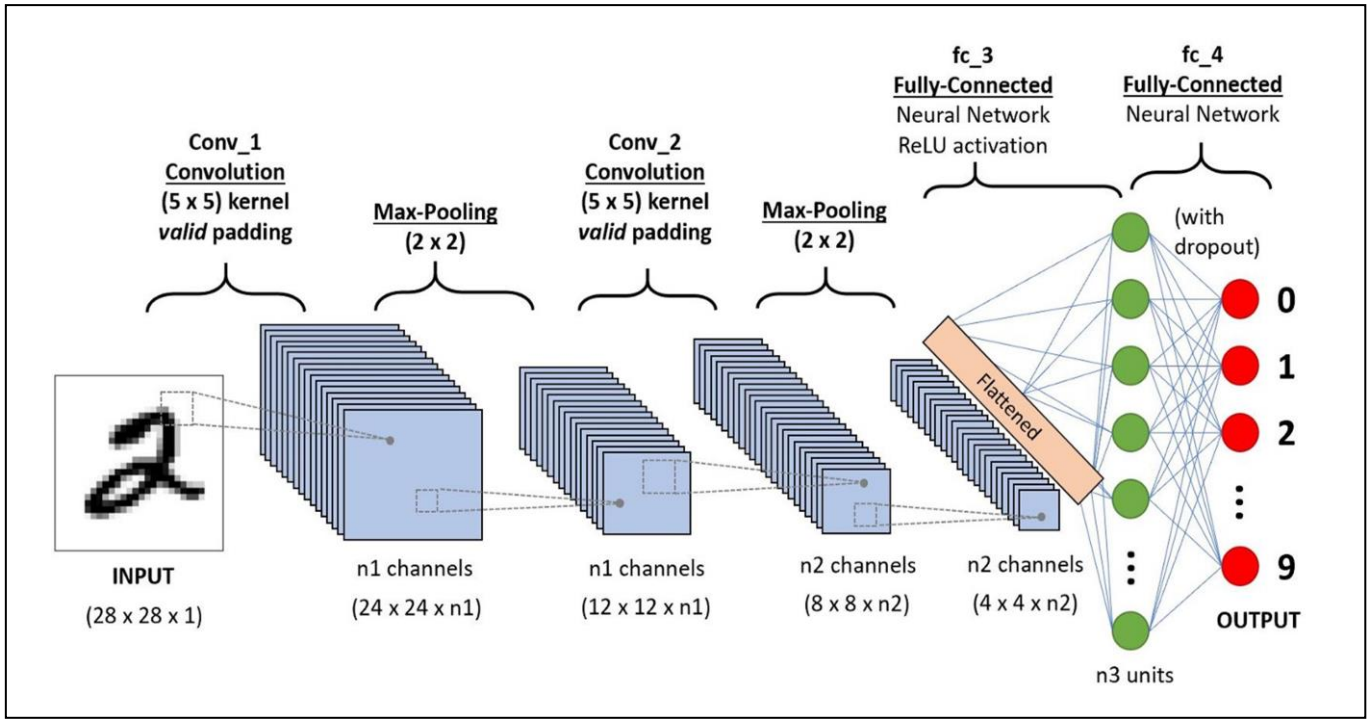
Areas of application of CNNs

1. Image classification (Computer Vision)
2. Object detection (Computer Vision, robots, autonomous cars)
3. Language processing (NLP)

Architecture of a CNN network

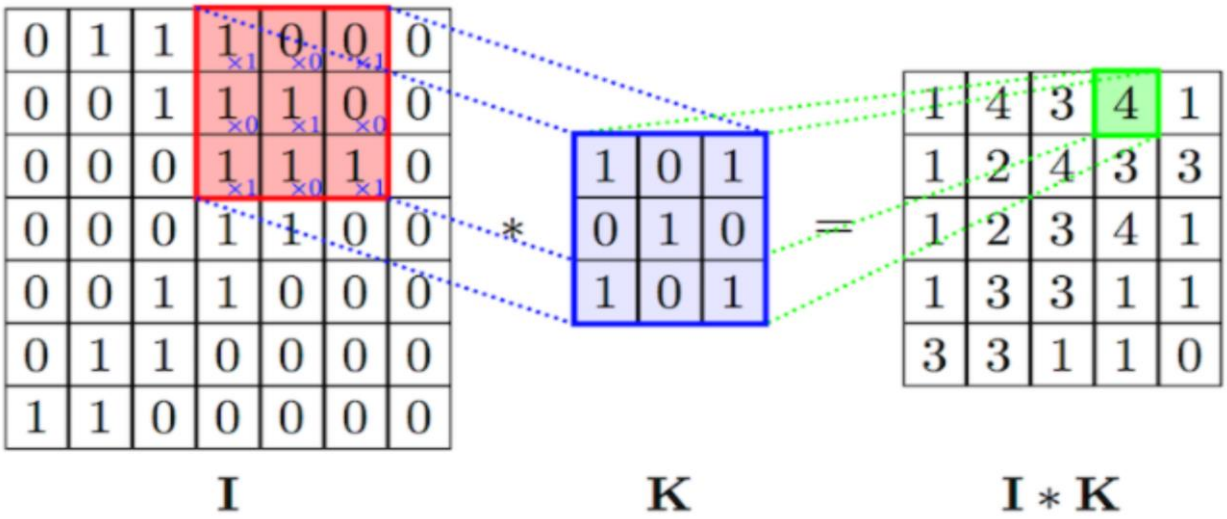
A CNN model is made up of several types of layers, including:

- CONVolution layers (Conv Layers)
- Sampling layers (Pooling Layers)
- A Fully Connected layer (Fully Connected Layer).



CONV-Layer:

- Is based on the mathematical operation convolution.
- Uses a set of filters where each filter (also called kernel) is a two-dimensional 2D matrix containing integer values.



Advantages of a CONV-layer

- Reduce the dimensions of the input images to facilitate their processing (we can keep the dimensions if necessary).
- Extract the most significant elements in the image (edges, colors, ...) without losing information.

How to apply convolution on images?

We apply this filter to an input image to produce a reduced image (this is very important) by convolving the filter with the image following the following procedure:

1. Superimpose the filter on the input image at a specific location.
2. Perform an element-by-element multiplication between the filter values and the corresponding image values.
3. Calculate the sum of all the elementary products. The result of this sum is the output value for the pixel of the resulting output image.
4. Move the filter on the image to another location
5. Repeat (2) and (3)

Example

We consider a 4x4 grayscale (black and white) input image and a 3x3 filter as follows:

0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

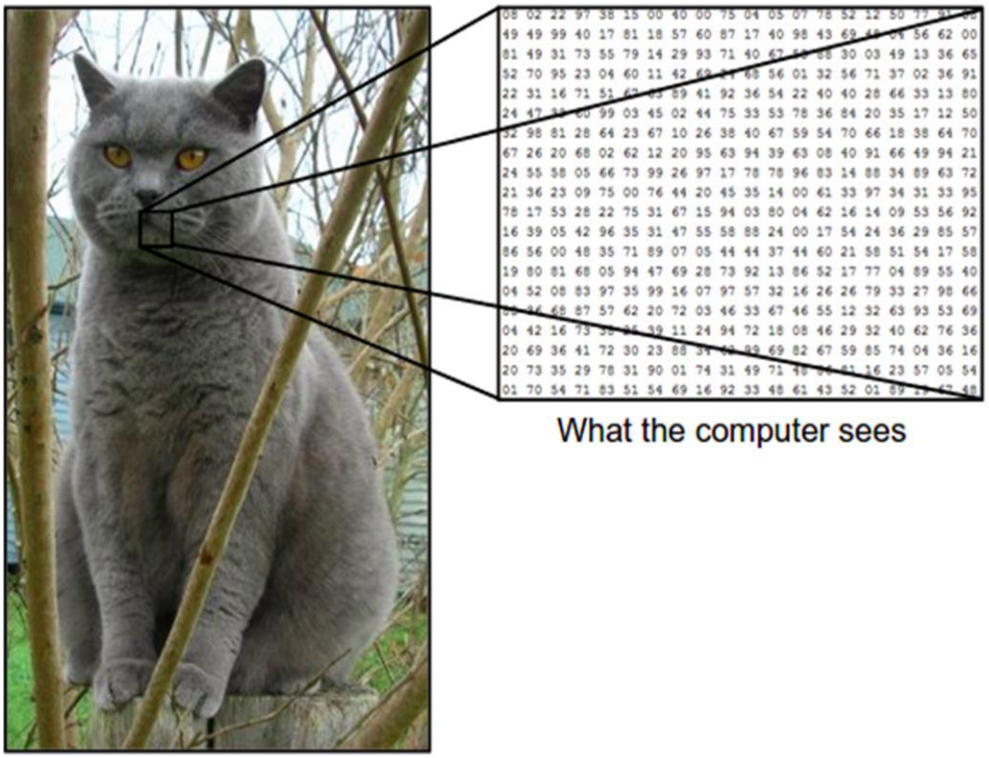
A 4X4 Image

-1	0	1
-2	0	2
-1	0	1

A 3X3 filter

The numbers in the image represent pixel intensities (colors). Eg, the value 0 represents the black color, while the value 255 represents the white color.

In general, pixels with high values represent the lightest part of the image and pixels with low values represent the darkest parts.



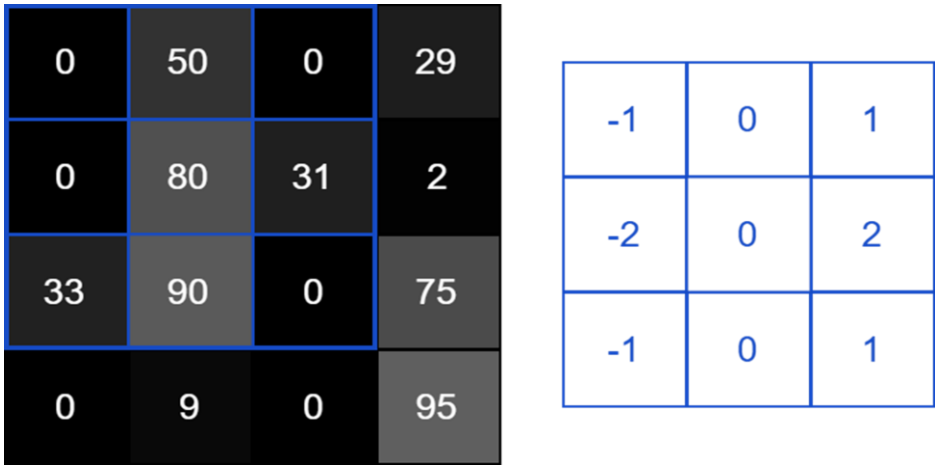
Representation of an image on the machine

We want to convolve the image and the filter to have an output image of dimension 2x2

?	?
?	?

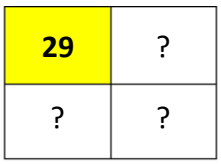
A 2X2 output image

As explained previously, we superimpose our filter on the upper left corner of the image



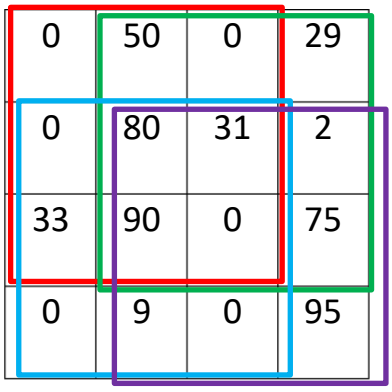
Then we perform an element-by-element product between the image values and the overlapping filter values starting from the upper left corner and going to the right, then down.

0x(-1) + 50x0 + 0x1 + 0x(-2) + 80x0 + 31x2 + 33x(-1) + 90x0 + 0x1
0+0+0+0+0+62-33+0+0 = 29

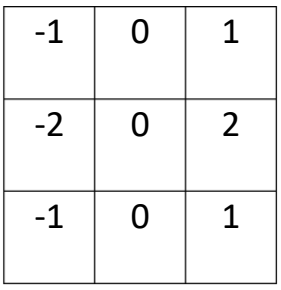


A 2X2 output image

We do the same thing by moving the filter horizontally and vertically on the input image by calculating the corresponding sums to find the values of the destination pixels of the output image.



A 4X4 Image



A 3X3 filter

29	-192
-35	-22

A 2X2 output image

Types of filters used in convolution?

The type of filters we choose allows us to detect the vertical or horizontal edges of the image.

1	0	-1
1	0	-1
1	0	-1

A vertical 3X3 filter

1	1	1
0	0	0
-1	-1	-1

A horizontal 3X3 filter

Examples of the most used filters

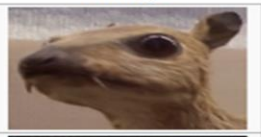
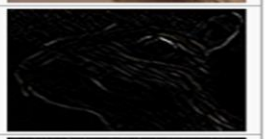
-1	0	1
-2	0	2
-1	0	1

A vertical Sobel filter

3	0	-3
10	0	-10
3	0	-3

A Scharr filter

Filter influence

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

What does the convolution of an image give?

By applying the 3x3 Sobel vertical filter that we have already used previously on the image, we obtain:

-1	0	1
-2	0	2
-1	0	1

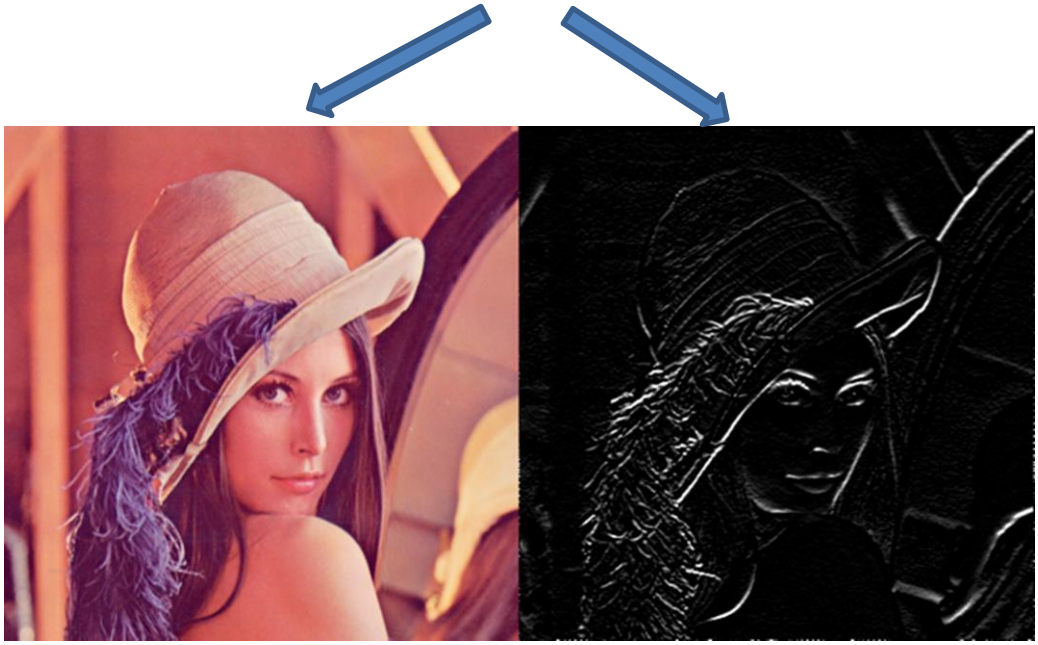
A vertical Sobel filter

An image convolved with a vertical Sobel filter

Likewise, if we apply a horizontal Sobel filter, we obtain the following image:

1	2	1
0	0	0
-1	-2	-1

A horizontal Sobel filter



An image convolved with a horizontal Sobel filter

Can we easily see that this is what is happening?

- Sobel filters are edge detectors. The vertical Sobel filter detects vertical contours.
- On the other hand the horizontal Sobel filter detects horizontal contours. The resulting images are now clearly interpreted, a bright pixel (having a high value) in the resulting image indicates that there is a strong edge around that area in the input (original) image.

Padding

Recall that convolving a 4x4 image with a 3x3 filter can produce a 2x2 output image.

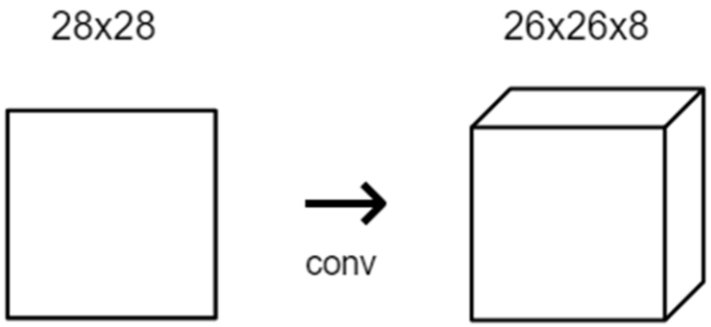
If we want to obtain an output image of the same dimensions as those of the input image (4x4), we must add 0s around the image in order to be able

to superimpose the filter in several places. Note that a 3x3 filter requires a padding pixel. In this case we are talking about the same padding, because the input and output images have the same dimensions. We can also have padding which reduces the dimensions of the output image, this is the case of valid padding.

1. Convolution layers (Conv.Layers)

As explained previously, CNNs include convolution layers (Conv.Layers) which use a set of filters to transform input images into output images. So, the main parameter of a Conv layer is the number of filters used.

For example a CNN model applied to an MNIST image (representing a number), we use a small initial CONV layer with 08 filters, which transforms the 28x28 input image into an output volume 26x26x8 (08 26x26 images stacked together) .



2. Pooling/Sampling Layers

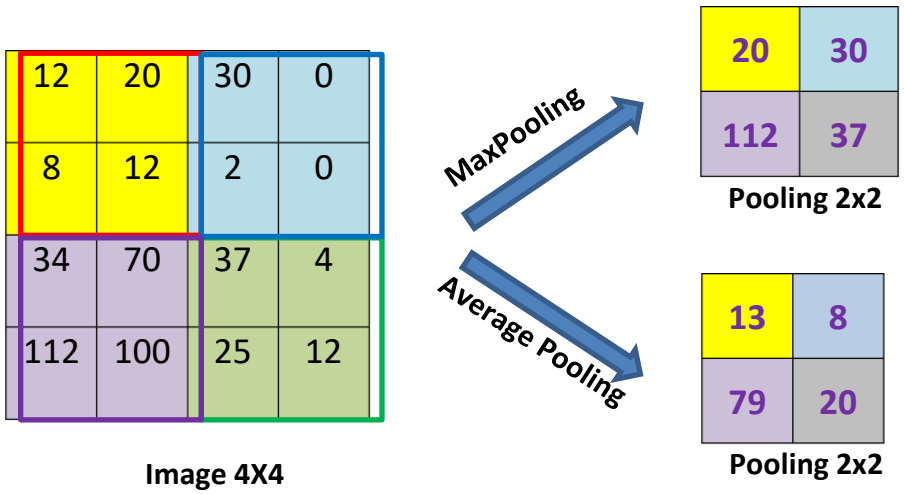
Generally, neighboring pixels in images have similar values hence the convolution also produces pixels of similar values. So why not group these similar pixels together and take only one pixel representative of these pixels. This is exactly the idea behind sampling (pooling).

The objective is to reduce the dimensions of the resulting images in the CONV layer in order to take only the significant pixels and speed up image processing.

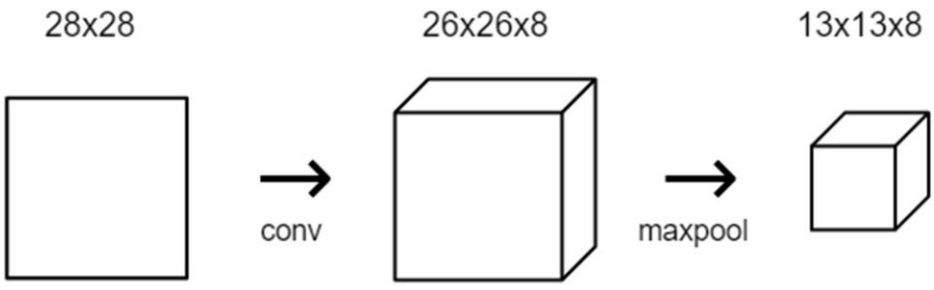
There are 2 types of sampling (pooling):

1. **Maximum sampling (MaxPooling):** take the maximum value from each part of the image
2. Average Sampling (Average Pooling): Take the average value of each part of the image.

Example: a pooling layer (MaxPooling–Average Pooling) of size 2x2

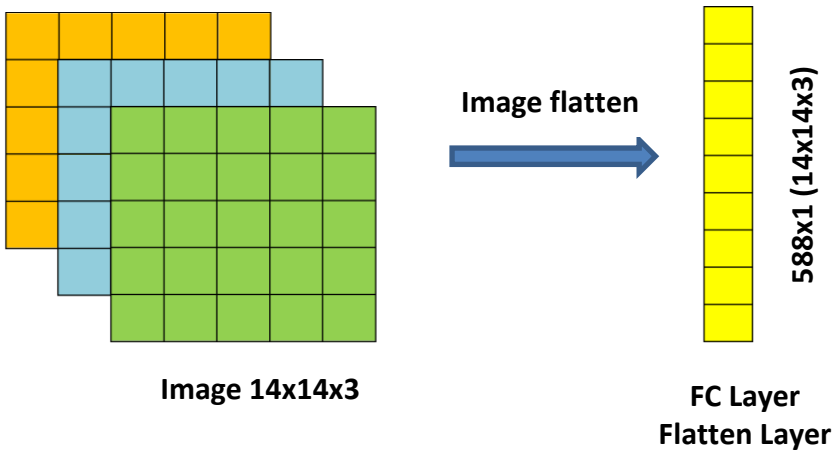


Note that pooling divides the width and height of the input image by the size of the pool. For the previous example (the MNIST model), we place a Maxpooling layer of size 2x2 just after our initial CONV layer, which will transform the 26x26x8 input image into a 13x13x8 output.



3. Fully Connected Layer/FC Layer/Flatten Layer

- Transforms the image into a one-dimensional vector (1-dim vector) before starting the classification phase



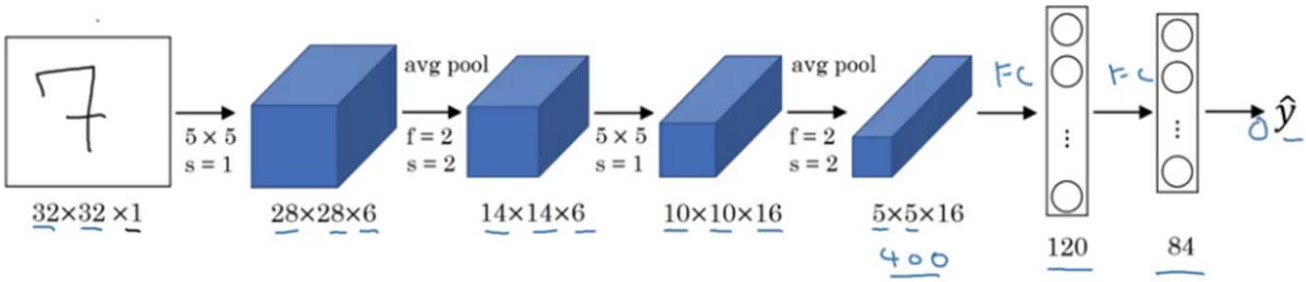
Popular architectures of CNNs

In this section, we will see the most popular architectures of CNN networks, note:

1. LeNet-5 architecture
2. The AlexNet architecture
3. The VGG architecture

1. LeNet-5 (LeCun, 1998)

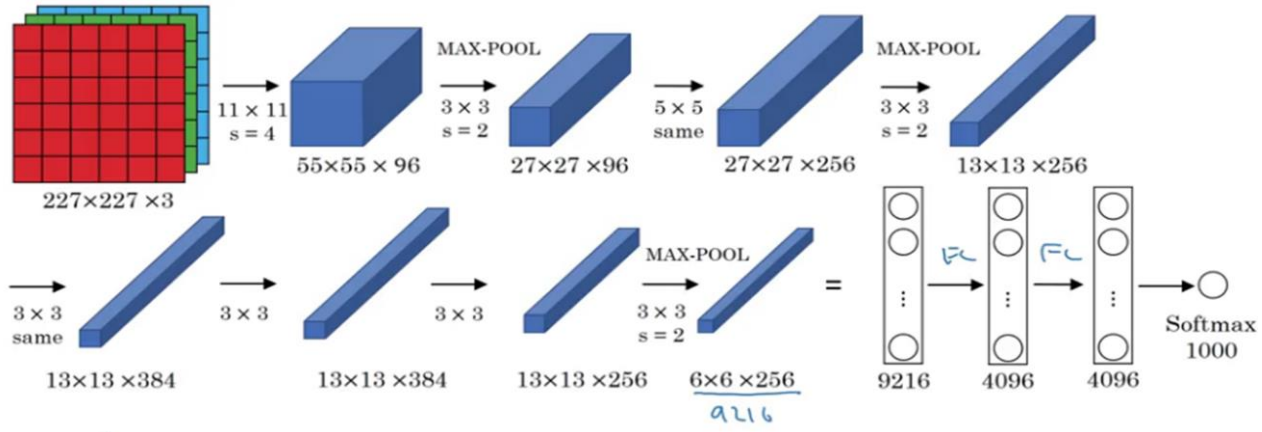
Can be represented as follows:



- **Settings:**60k
- **Layers used:** Conv -> Pool -> Conv -> Pool -> FC -> FC -> Output
- **Activation function:** Sigmoid/tanh and ReLu

2. AlexNet (2012)

Can be illustrated as follows:

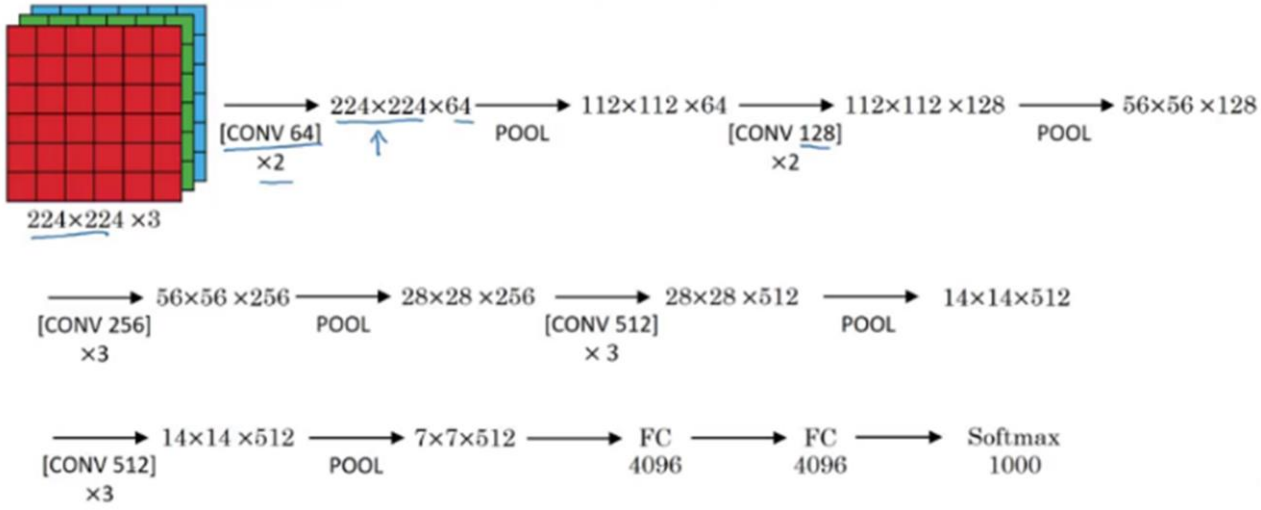


This model is similar to the LeNet-5 model with more convolution and pooling layers

- **Settings:**60 million
- **Layers used:**as explained in the figure.
- **Activation function:**ReRead

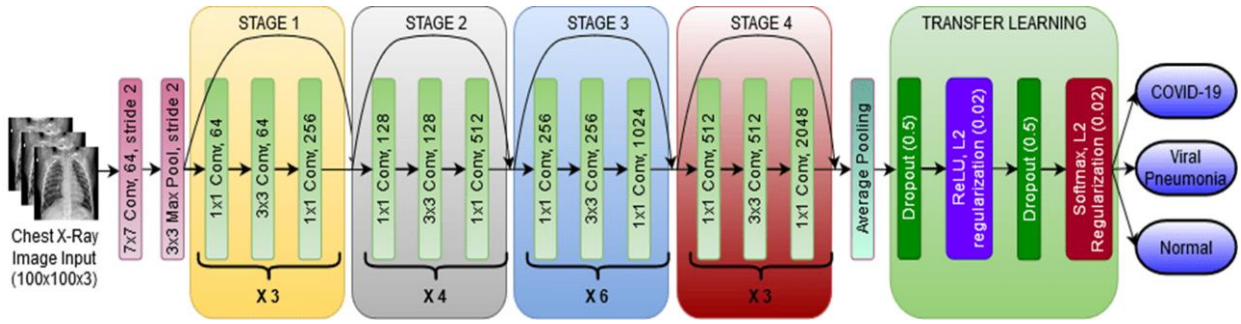
3. *VGG-16 (2014)*

The VGG-16 architecture is based on a simpler network that emphasizes sized CONV layers (also uses the same padding). A MaxPooling layer of size 2x2 is used after each convolution layer. The VGG-16 architecture can be summarized as follows:



. **Settings:**138 million

4. *ResNet (2015)*

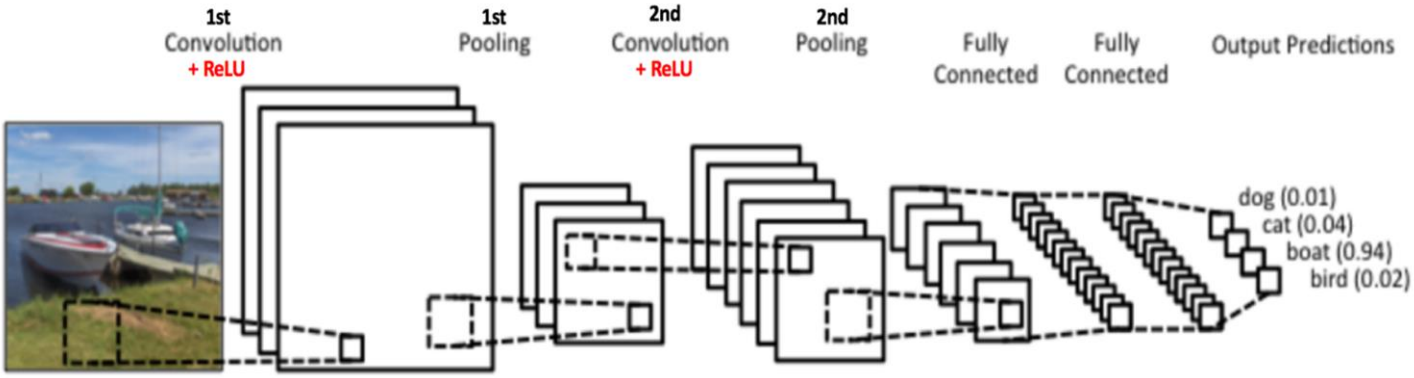


5. Other Architectures

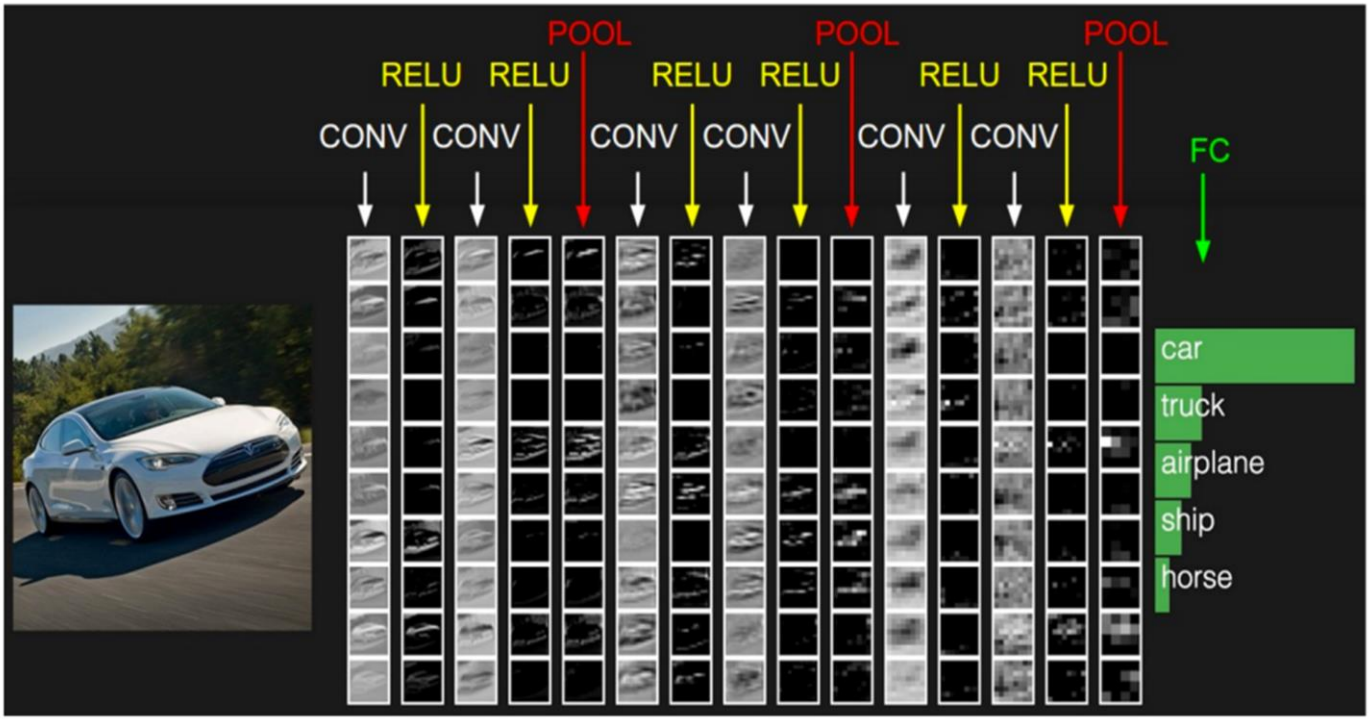
- ✓ ZFNet (2013)
- ✓ GoogleNet (2014)
- ✓ CUImage (2016)
- ✓ Squeeze and Excitation Networks (2017)

Some illustrations of model CNNs

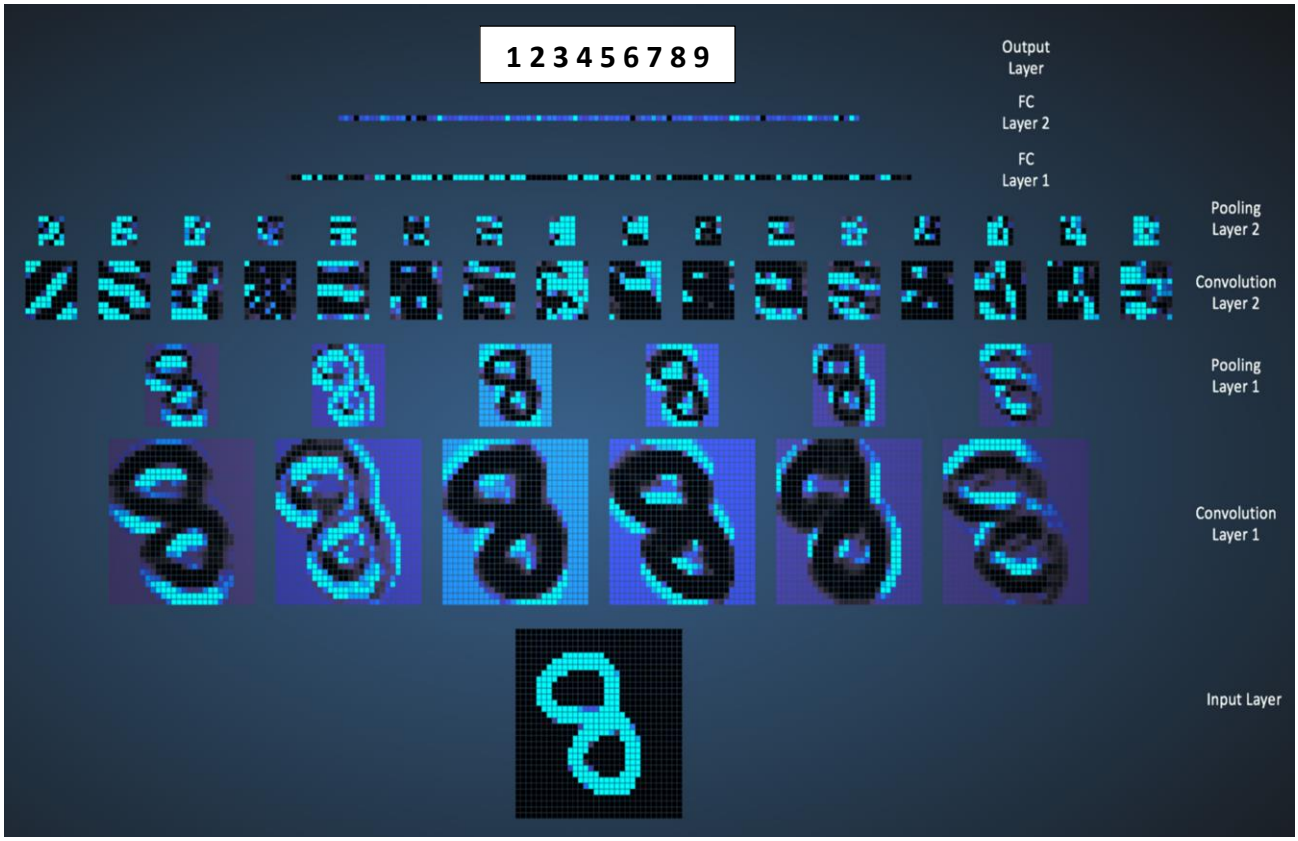
1. Classification of animals



2. Classification of different objects



3. *Number recognition*



Published papers and articles

Conference Papers

1. Said G. Diabetic Patient Classification: An Efficient Algorithm Based on Deep Learning Approach, International Conference on Science, Engineering&Technology (ICSET) held in Istanbul, Turkey, 20-21 December, 2019 (Winner of Best Conference Paper)
2. Said G. Efficient Arabic handwritten character recognition based on machine learning and deep learning approaches, The 2nd International Conference on Advances in Intelligent Systems, Soft Computing and Optimization Techniques 2020 (ICAISCO 2020) held in Penang, Malaysia, 01 - 02, April 2020 (**Scopus Indexed Conference**).
3. Said G., Erich N: Building Best Predictive Models Using ML and DL Approaches to Categorize Fashion Clothes, The 19th International Conference on Artificial Intelligence and Soft Computing ICAISC 2020 (H5-index = 20) (**Springer Conference**). held in Zakopane, Poland, 12 - 14, October 2020.

Available on:https://link.springer.com/book/10.1007/978-3-030-61401-0

4. Hassina T, Said Gadr, Baya L, Nour El-Houda A, Nadjet B: Handwritten Digit Recognition: Developing an Efficient ML and DL Model to Recognize Handwritten Digits, CNIATI'21 National Conference on Artificial Intelligence and Information Technologies information, May 24, 2021, University Chadli Ben Djedid, Al-Taref.
5. Said G, Sara OM, Khadidja H, Safia C. An Efficient System to Predict Customers' Satisfaction on Touristic Services Using ML and DL Approaches. 22nd Arabic International Conference on Information Technology (ACIT 2021), 21-23 Dec,**IEEE Conference**, Quabos University, Sultanate Oman. Available on:https://ieeexplore.ieee.org/document/9677167.

DOI: 10.1109/ACIT53391.2021.9677167

6. Gadri S., Adouane NE (2022) Efficient Traffic Signs Recognition Based on CNN Model for Self-Driving Cars. In: Vasant P., Zelinka I., Weber GW. (eds) Intelligent Computing & Optimization. ICO 2021. Dec 30-31, 2021 (**LNNS Springer Conference)**. Lecture Notes in Networks and Systems, vol 371. Springer, Cham.https://doi.org/10.1007/978-3-030-93247-3_5

Available on:https://link.springer.com/chapter/10.1007/978-3-030-93247-3_5

7. Gadri S., Chabira S., Ould Mehieddine S., Herizi Kh. (2022) Sentiment Analysis: Developing an Efficient Model Based on Machine Learning and Deep Learning Approaches. In: Vasant P., Zelinka I., Weber GW. (eds) Intelligent Computing & Optimization. ICO 2021. Dec 30-31, 2021 (**LNNS Springer Conference)**. Lecture Notes in Networks and Systems, vol 371. Springer, Cham.https://doi.org/10.1007/978-3-030-93247-3_24

Available on:https://link.springer.com/chapter/10.1007/978-3-030-93247-3_24

Newspaper Articles:

8. Said G. Efficient Arabic handwritten character recognition based on machine learning and deep learning approaches, Journal of Advanced Research in Dynamical & Control Systems, (Scopus indexed Journal), online version Vol. 12, 07-Special Issue, 2020, Pages: 9-17,DOI:10.14311/NNW.2017.27.011, August 2020, DOI: 10.5373/JARDCS/V12SP7/20202076, ISSN 1943-023X,https://www.jardcs.org/archivesview.php?volume=3&issue=39
9. Said G. Diabetic patient classification: an efficient algorithm based on deep learning approach, International Journal of Advances in Electronics and Computer Science IJAECS, ISSN(p): 2394-2835, Volume-7, Issue-3, April 2020,http://iraj.in
10. Said G. Developing an efficient Predictive model based on ML and DL approaches to detect diabetes, The International Journal of Computing and Informatics Informatica, ISSN(p): 0350-5596, Volume-45, Issue-3, 2021, https://www.informatica.si/index.php/informatica/article/view/3041