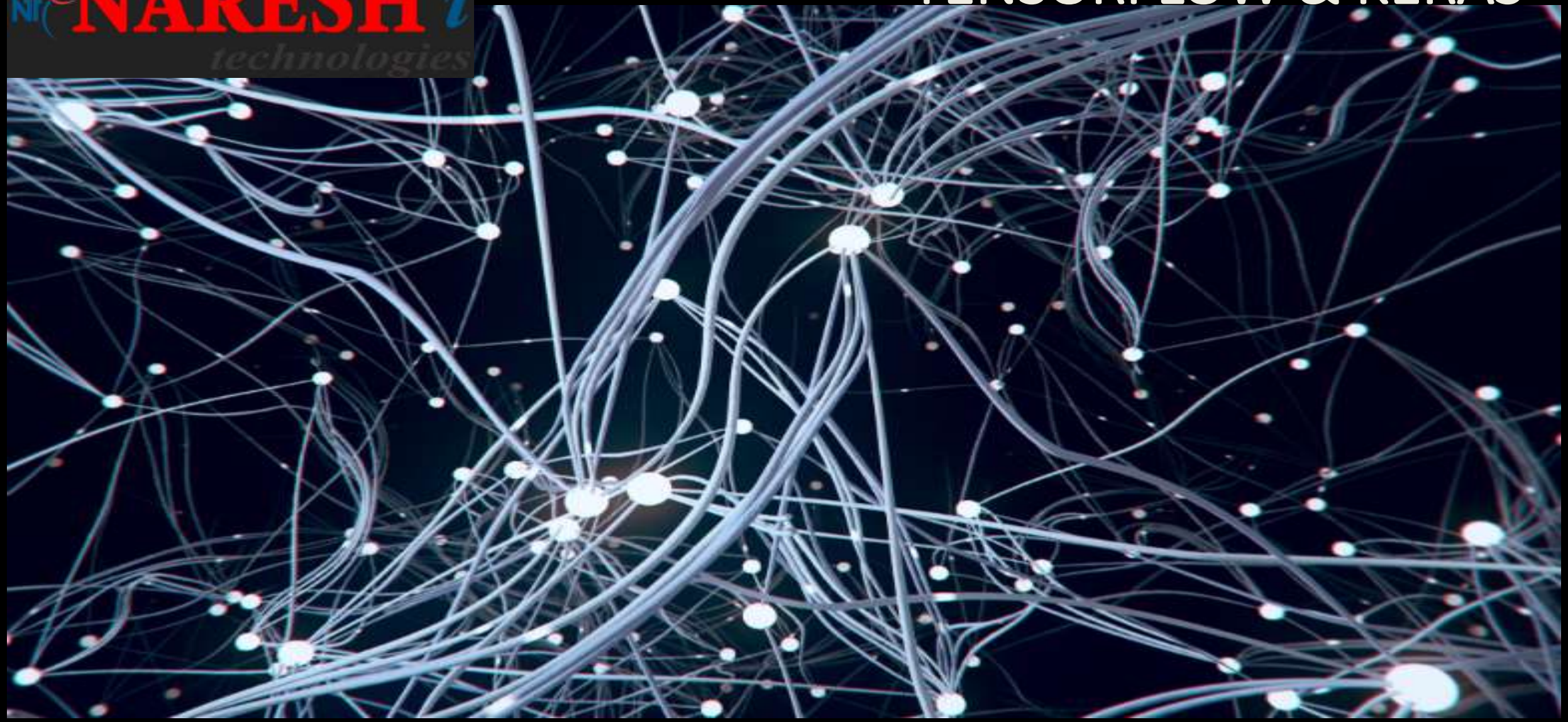
Workshop on Deep Learning & Neural Network Using TENSORFLOW & KERAS

NARESH i technologies



K. Prakash Senapati

Agenda for todays workshop :-

1- What is Deep learning

2- Deep Learning Framework - TensorFlow

3- Installation of TensorFlow | Keras | Pytorch | OpenCV |

4- What is Neural Network Architecture

5- ANN (Artificial Neural Network)

- Neuron
- Signal
- Forward propagation
- Backward propagation
- Epoch
- Activation Function

- ❑ **PRACTICLE – CUSTOMER CHURN PREDICION USING TENSORFLOW & KERAS**
- ❑ **PRACTICLE – FASHION MNIST DATASET TRINING USING TENSORLFOW & KERAS**

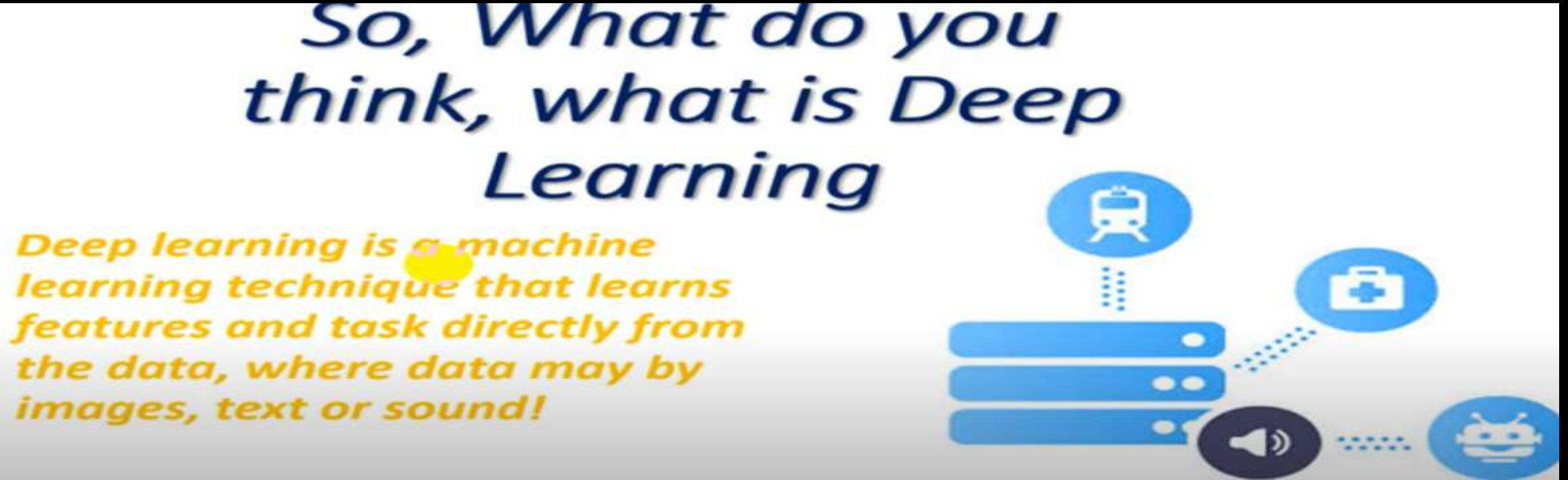
6- CNN (Convolution Neural Network)

- Convolution
- Max Pooling
- Flattening
- Fully connected Layer

- ❑ **PRACTICLE - MOOD CLASSIFICATION USING TENSORFLOW & KERAS**

1- What is Deep Learning -

- **Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way of human brain.**
- **Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions.**
- **Deep Learning is collection of neural network & These neural networks helps to simulate the behavior of the human brain.**
  - **ANN (Artificial Neural Network)**
  - **CNN (Convolution Neural Network)**
  - **RNN (Recurrent Neural Network)**



2- Deep Learning Framework -

- **What is mean by Framework –**
- **Framework can refer to a structure. It could be the structure of a system, a building, a project.**
- **Frameworks are a huge help to developers in designing products.**
- **They provide an infrastructure that has already figured out the low-level details, allowing the developer to focus on the unique details of the specific project.**
- **The developer also has to write less code.**

3- Installation of TensorFlow | Keras | Pytorch | OpenCV :

1- Anaconda command prompt

2- conda create -n tensorflow_env tensorflow

3- activate tensorflow_env

4- pip install keras==2.3.1 (((TENSORFLOW - BACKEND || KERAS - FRONTEND)))

5- pip install opencv-python (you can install under base & tensorflow_env)

6- pip install torch==1.5.1+cpu torchvision==0.6.1+cpu -f https://download.pytorch.org/whl/torch_stable.html (INSTLL PYTORCH)

7- Install basic library or package (eg – sklearn, numpy, pandas, matplot, nltk, spacy, gensim etc.)

8- Open Anaconda Navigator -- change base to tensorflow_env

9- Check in Notebook -- Setting -- install specific version of (Spyder & Jupiter)

10- click on anaconda -- check - 2 files are created under anconda folder

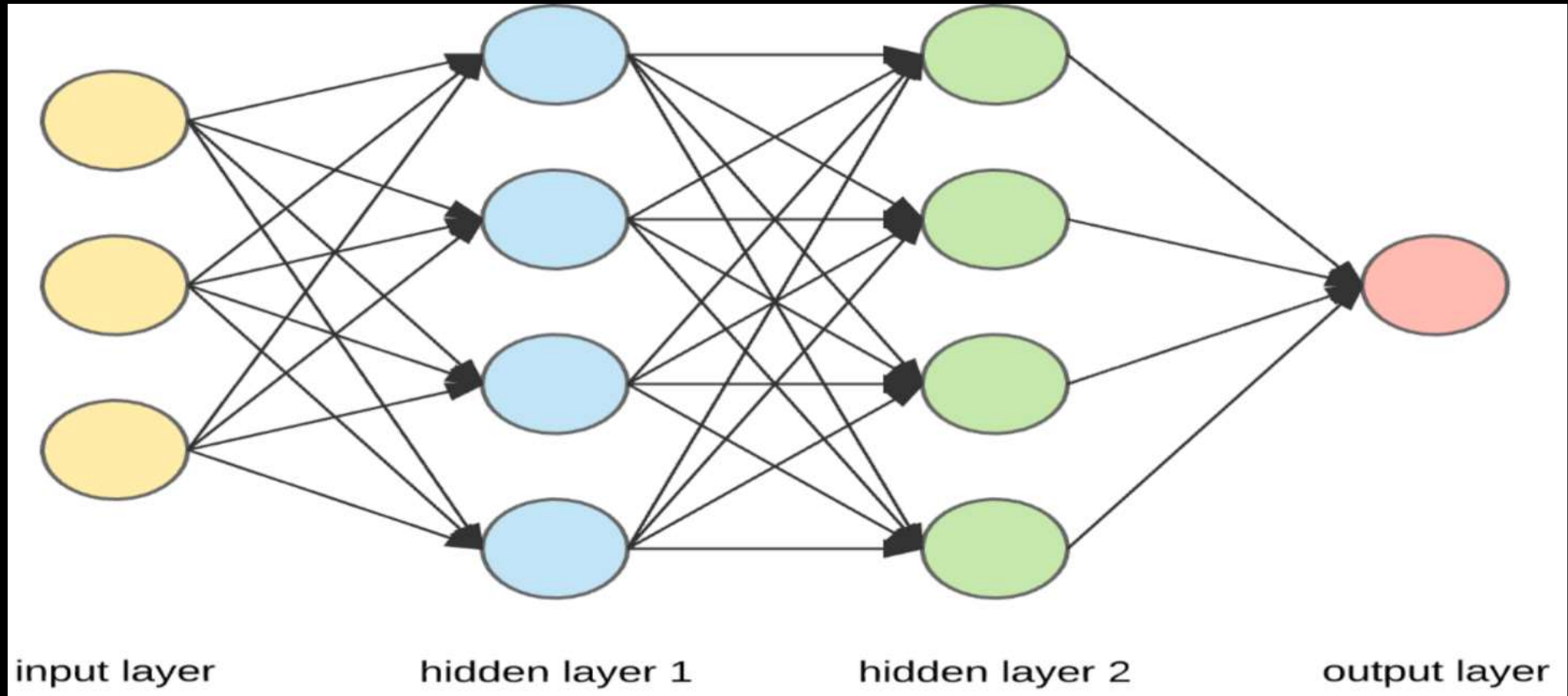
11- click on jupyter tensorflow_env

12- test wheather the framework installed on your machine or not .

```
import tensforflow as tf (tensorflow)
from keras.preprocessing.image import ImageDataGenerator(keras)
import cv2 (opencv)
```

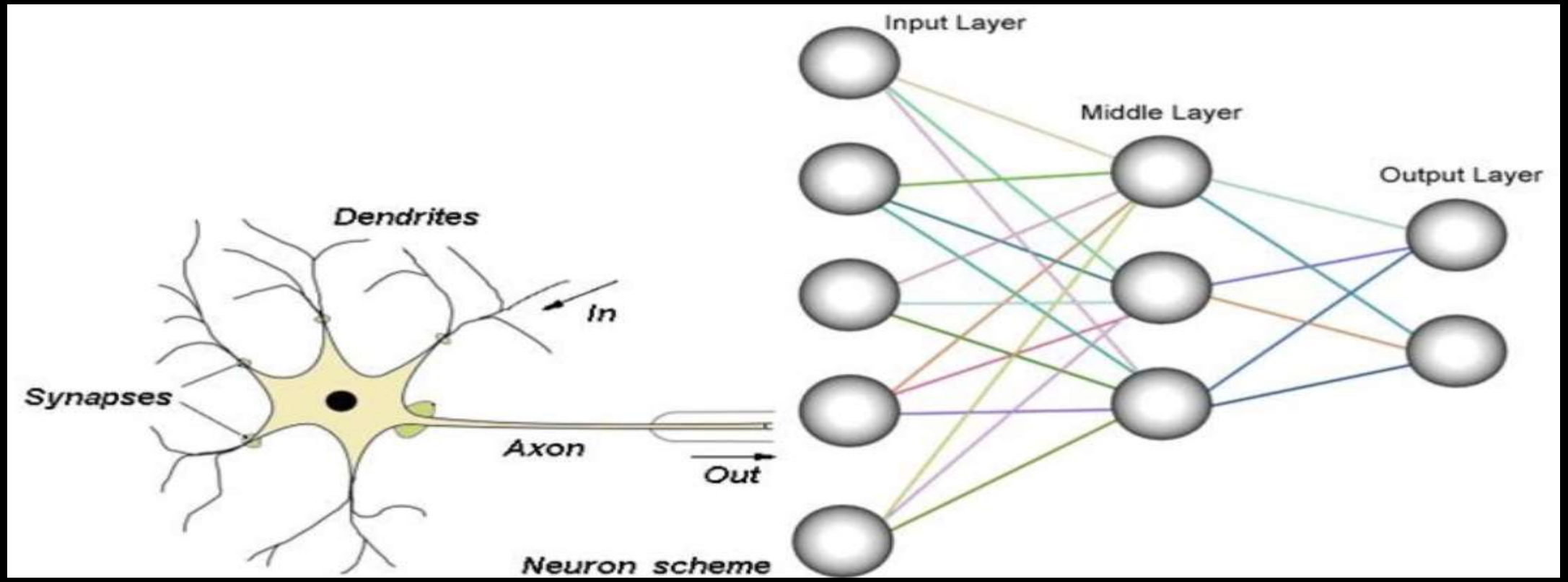
4- What is Neural Network Architecture -

- **The architecture of neural networks is made up of an input, output, and hidden layer.**
- **Neural networks themselves, or artificial neural networks (ANNs), are a subset of machine learning designed to mimic the processing power of a human brain.**



5- ANN (Artificial Neural Network):-

- **Lets understand what is Human Neural Network**
- **Human Neural Network vs Artificial Neural Network**
- **Artificial neural networks, usually simply called neural networks or neural nets**
- **ANN is based on a collection of connected units or nodes called artificial neurons,**
- **Each node, or artificial neuron, connects to another and has an associated weight and threshold.**
- **Signal process from (I/P layer → Hidden Layer → O/P layer)**

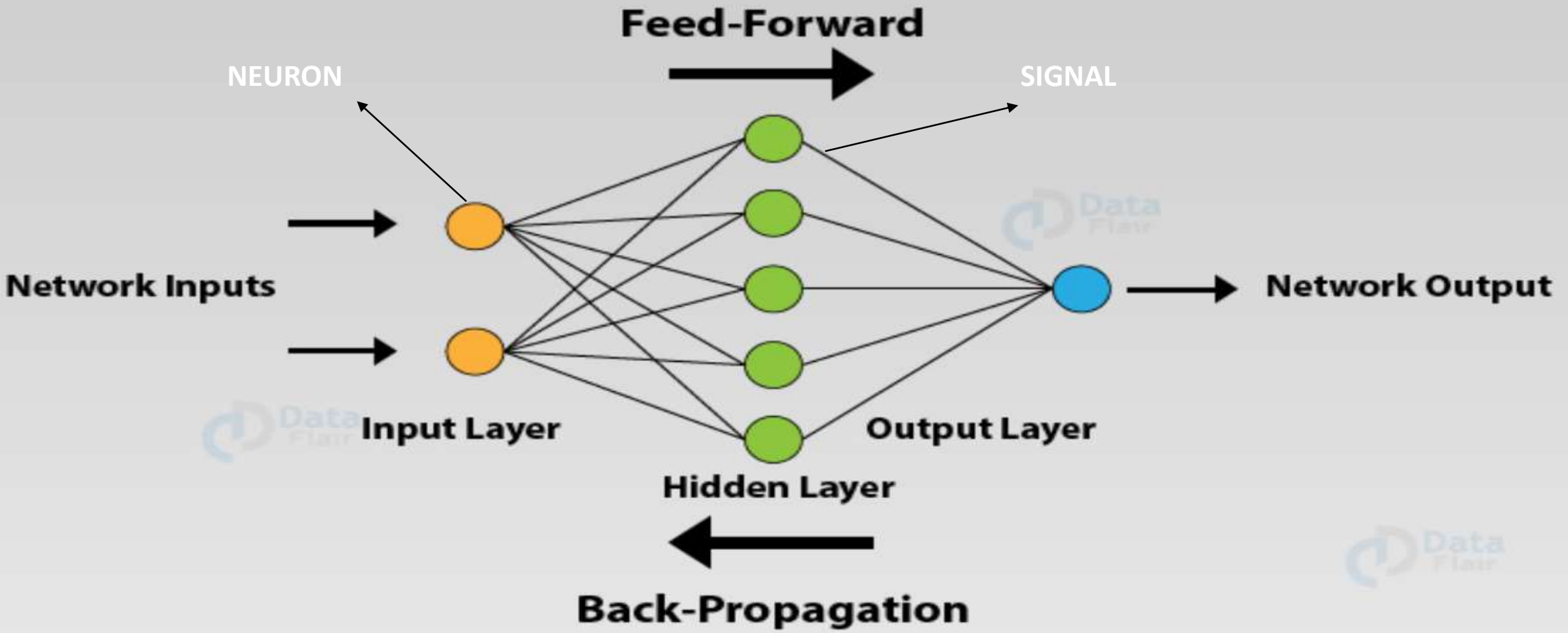


FORWARD PROPAGATION ->

Forward propagation is the way data moves from left (input layer) to right (output layer) in the neural network. A neural network can be understood by a collection of connected input/output nodes.

BACK PROPAGATION ->

Backward Propagation is the process of moving from right (output layer) to left (input layer).



ACTIVATION FUNCTION ->

In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs.

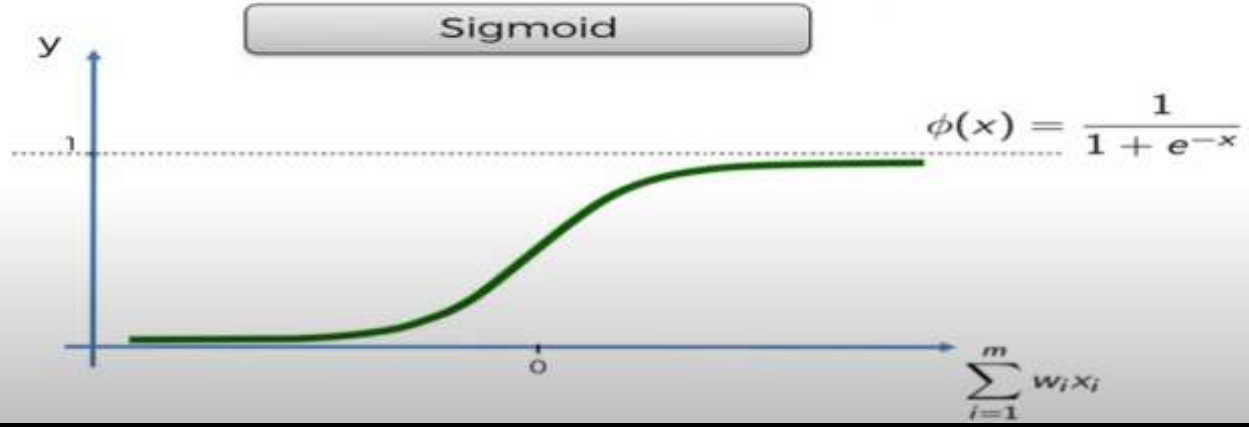
They make it easy for the model to generalize or adapt with variety of data and to differentiate between the output

The Nonlinear Activation Functions are mainly divided on the basis of their **range or curves**

1. Threshold
2. Sigmoid
3. Tanh
4. ReLU
5. Leaky ReLU
6. Softmax

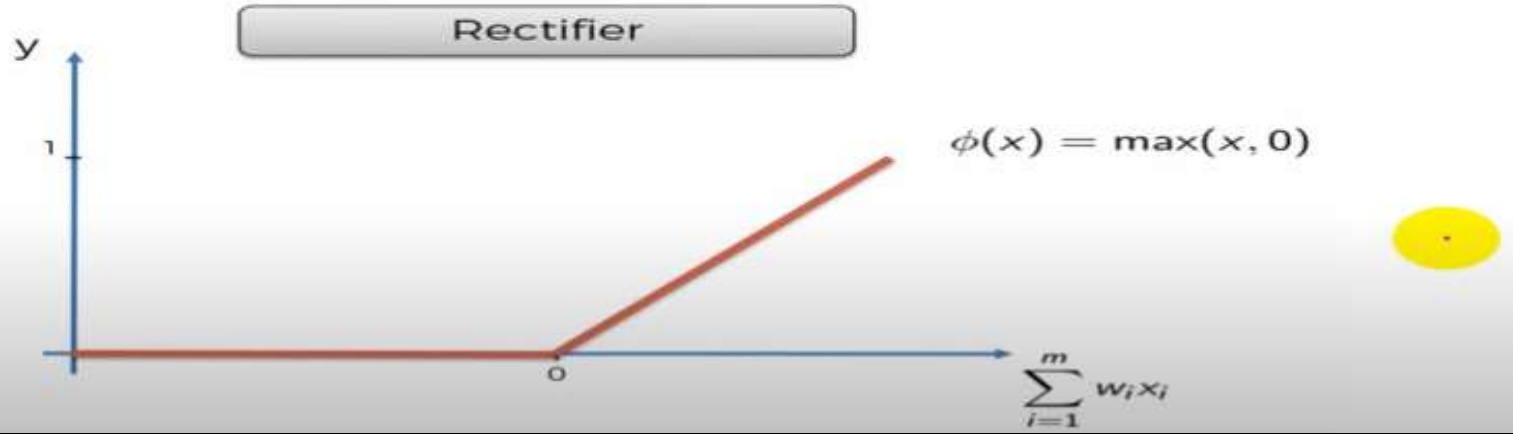
Sigmoid Function?

The Sigmoid Function curve looks like a S-shape
This function reduces extreme values or outliers in data without removing them.
It converts independent variables of near infinite range into simple probabilities between 0 and 1, and most of its output will be very close to 0 or 1.



Rectifier (Relu) Function?

ReLU is the most widely used activation function while designing networks today. First things first, the ReLU function is non linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.

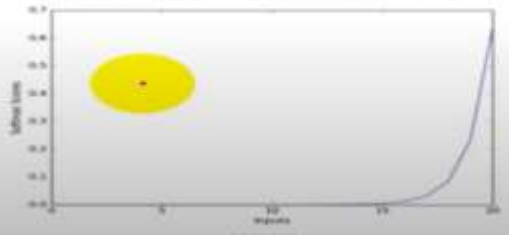


Softmax Function (for Multiple Classification)?

Softmax function calculates the probabilities distribution of the event over 'n' different events. In general way of saying, this function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

The main advantage of using Softmax is the output probabilities range. The range will 0 to 1, and the sum of all the probabilities will be equal to one. If the softmax function used for multi-classification model it returns the probabilities of each class and the target class will have the high probability.

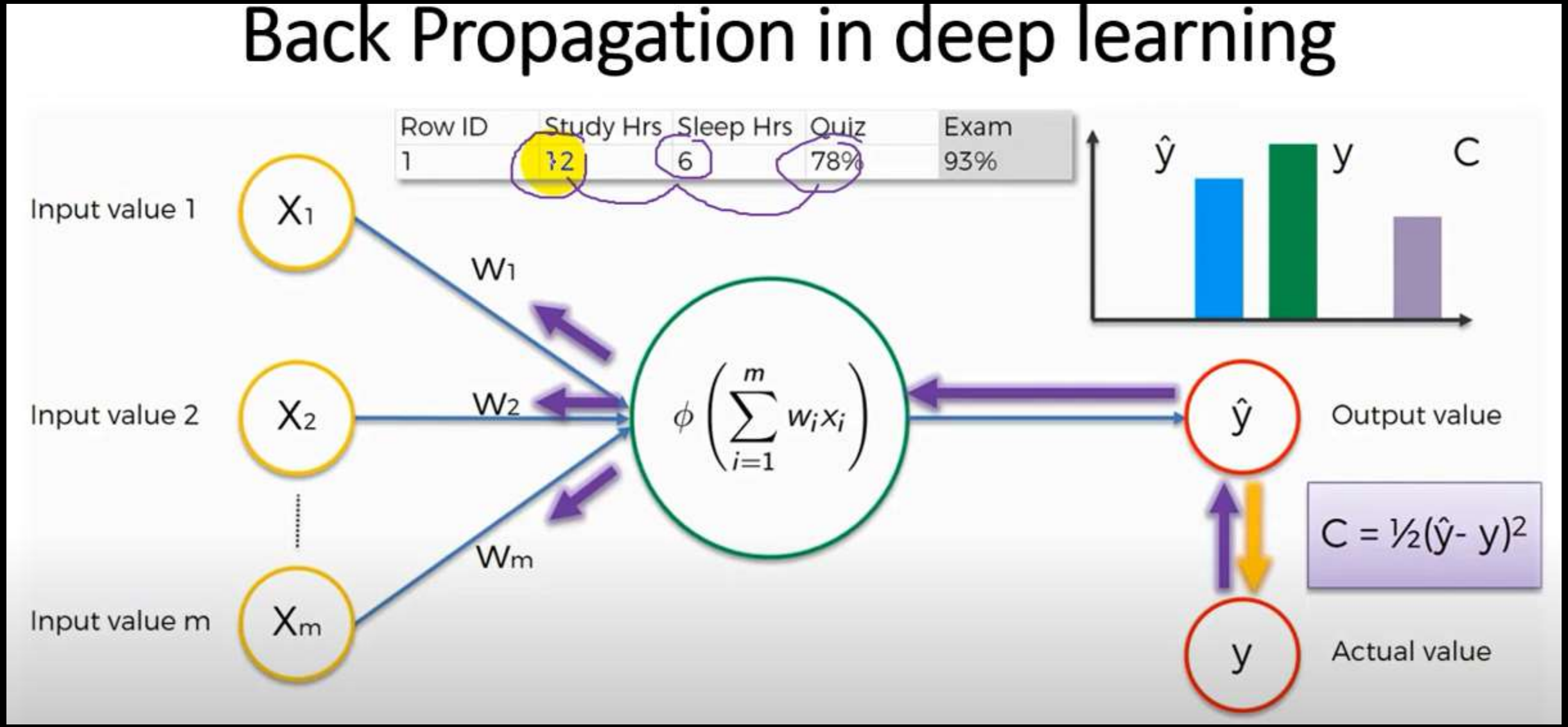
The formula computes the exponential (e-power) of the given input value and the sum of exponential values of all the values in the inputs. Then the ratio of the exponential of the input value and the sum of exponential values is the output of the softmax function.



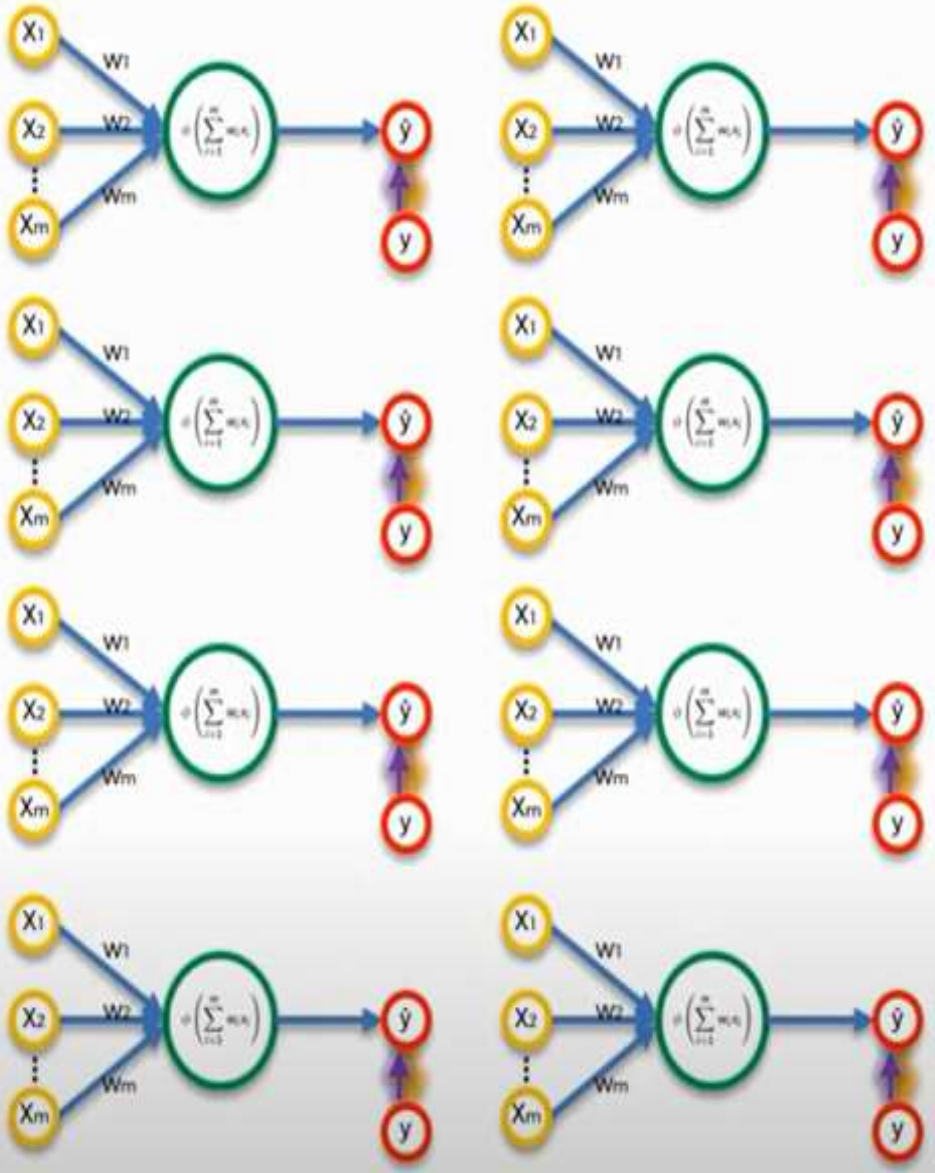
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

EPOCH:-

- Forward propagation + Backward propagation = 1 iteration (1 Epoch)
- Weights are adjusted at Back propagation . This is also called as backpropagation of error

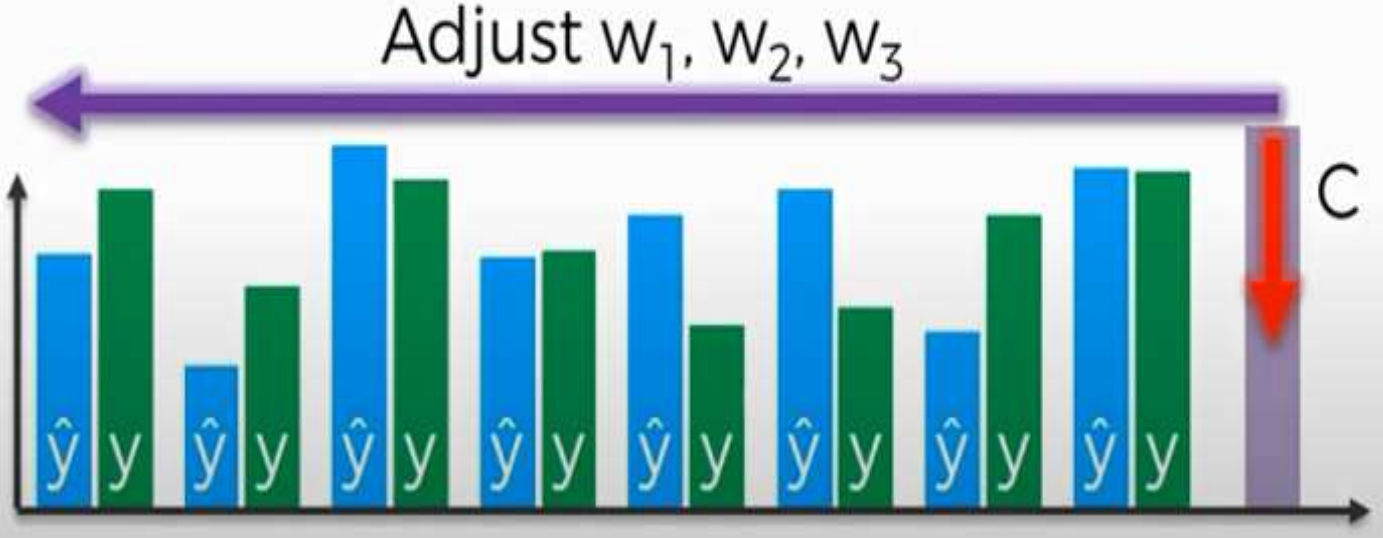


Back Propagation in deep learning (epoch)



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$



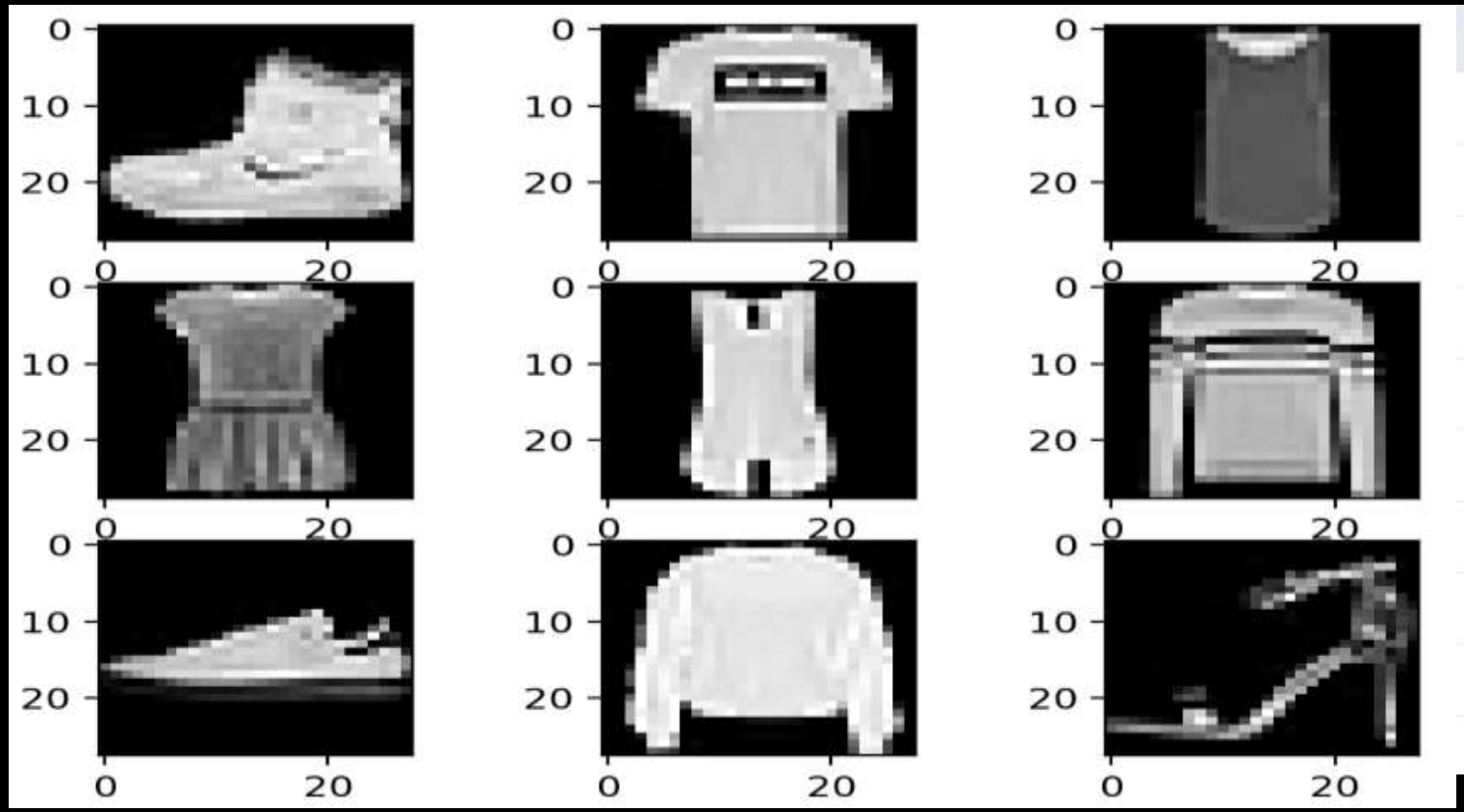
PROJECT-1: Customer Churn Prediction using TensorFlow (ANN)

- Customer churn means the customer who want to stopped to use the product.
- Your customer churn rate indicates how many of your existing customers are not likely to make another purchase from your business.
- **OVERVIEW:** This project is for BANKING domain and to build this project we used few customer historical sample records
- We build the deep learning model on historical data & we will see the model accuracy

Number	ID	Name	Cibil Score	State	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Akash	619	Bangalore	42	2	0	1	1	1	101348.88	1
2	15647311	Ashok	608	Pune	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Ben	502	Bangalore	42	8	159660.8	3	1	0	113931.57	1
4	15701354	Montt	699	Bangalore	39	1	0	2	0	0	93826.63	0

PROJECT-2: FASHION MNIST using TensorFlow (ANN)

- The Fashion MNIST dataset is database of fashion images that is commonly used for training and testing
- MNIST Database (Modified National Institute of Standards and Technology) is a database of fashion items which commonly used for various image processing & Neural network
- **OVERVIEW:** The Fashion-MNIST dataset contains 60,000 training images (and 10,000 test images) of fashion and clothing items, taken from 10 classes. Each image is a standardized 28x28 size in grayscale (784 total pixels).
- No need to import any dataset. If we import this dataset then it take more to extract 60000 images . so this dataset is prebuild in KERAS library & we are importing direct from keras.



Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Convolutional Neural Networks

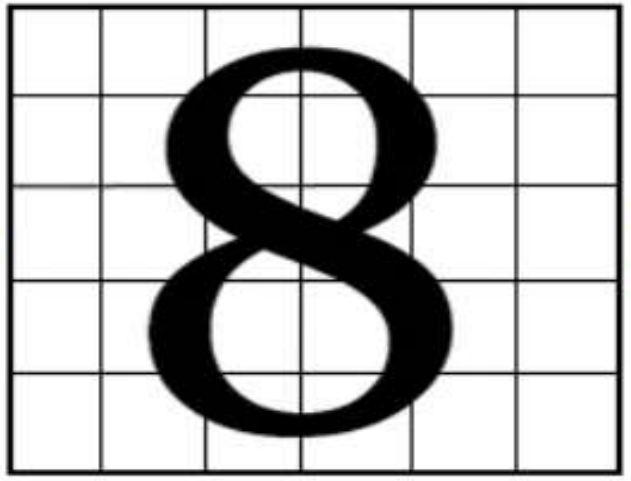
STEP 1: Convolution

STEP 2: Max Pooling

STEP 3: Flattening

STEP 4: Full Connection

Real Image of the digit 8

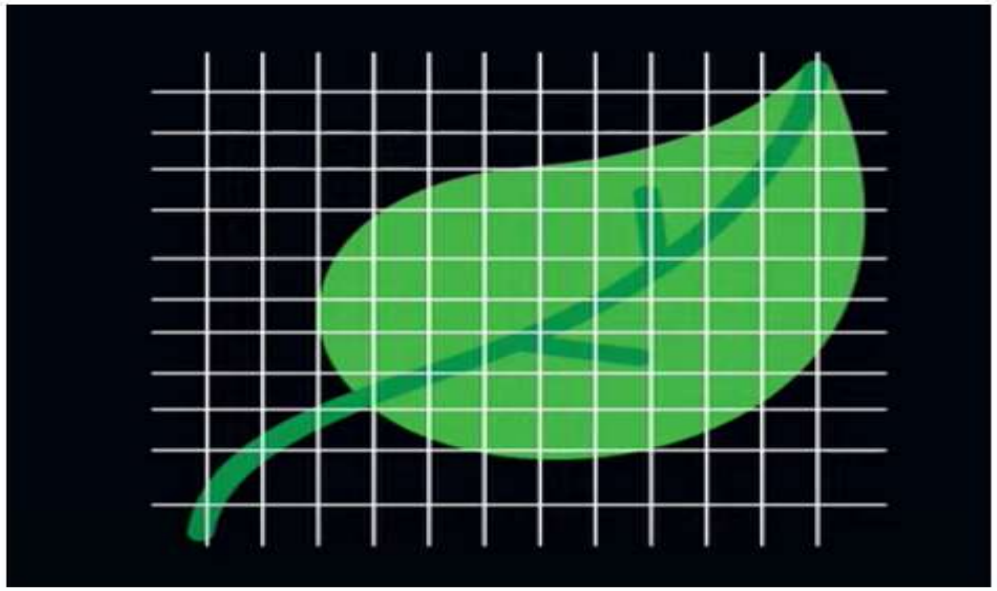


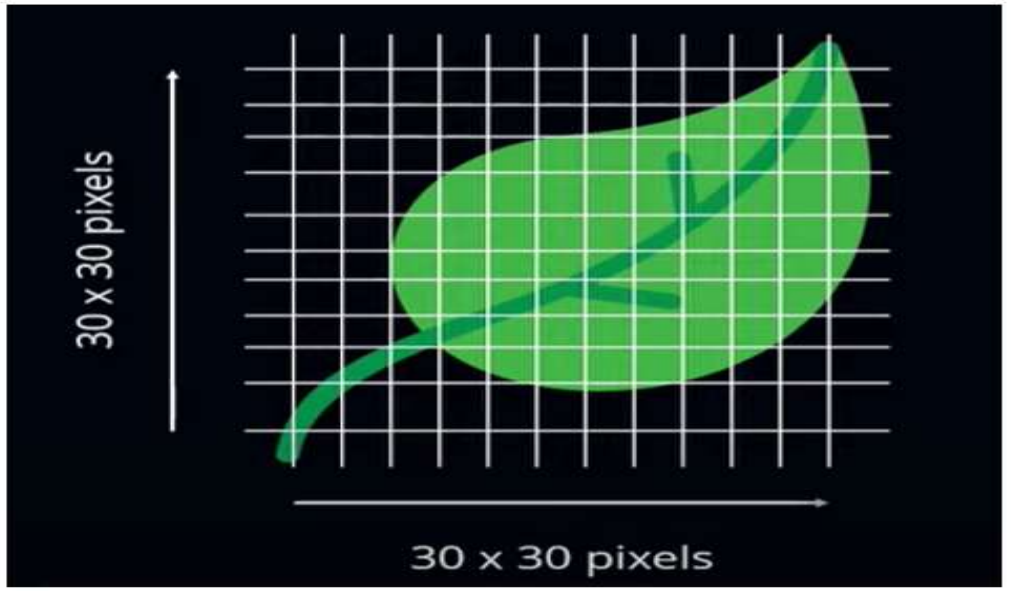
Represented in the form of an array

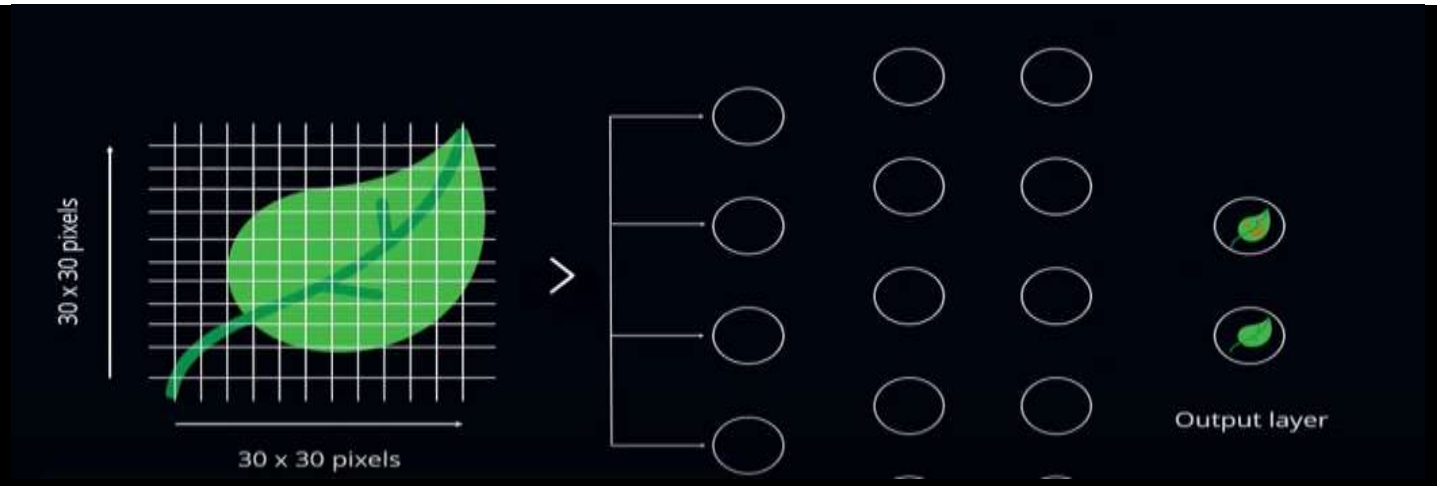
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0

Digit 8 represented in the form of pixels of 0's and 1's

Each leaf image will be broken down into pixels depending on the dimension of the image e.g., if the image is completed by
30 * 30 pixels then the total number of pixels will be 900 ,

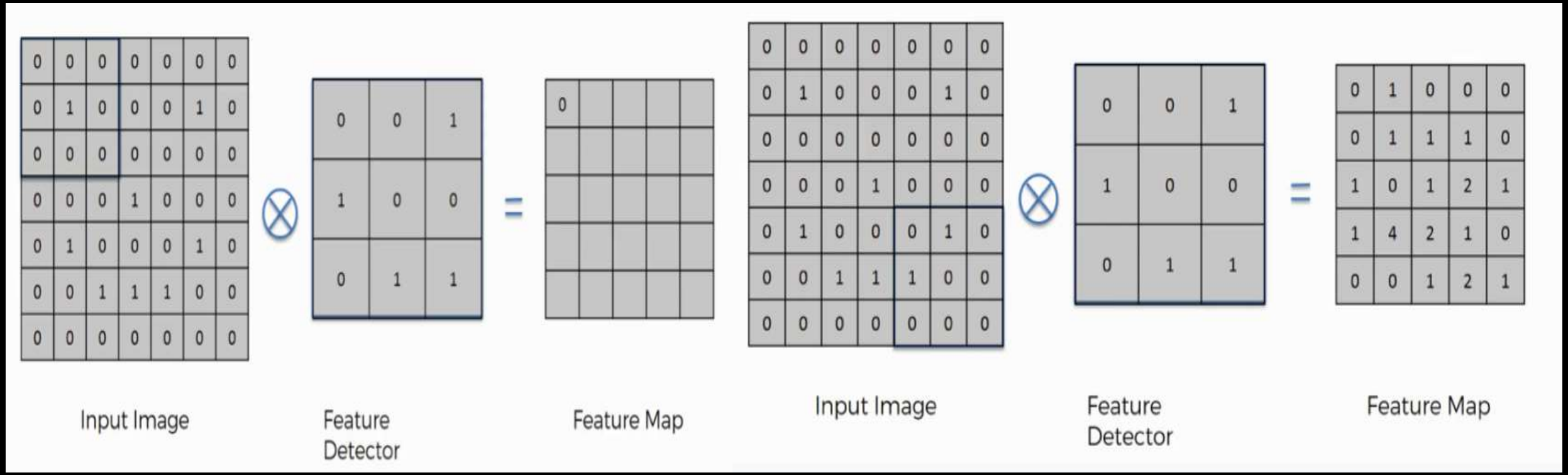






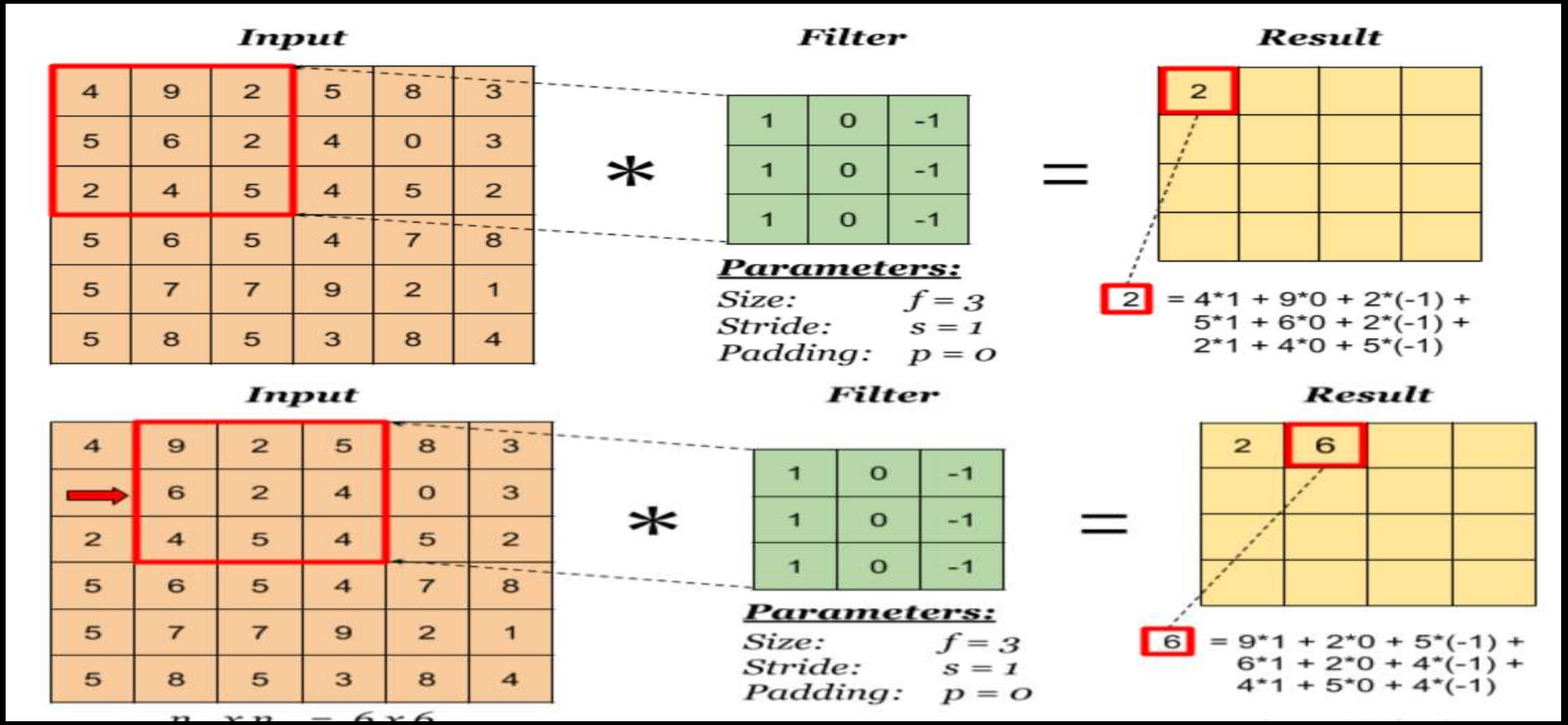
What is CONVOLUTION LAYER -

- **Convolution is a mathematical operation that allows the merging of two sets of information.**
- **In the case of CNN, convolution is applied to the input data to filter the information and produce a feature map.**
- **This filter is also called a kernel, or feature detector.**
- **The term convolution refers to the mathematical combination of two functions to produce a third function.**
- **It merges two sets of information & in CNN, the convolution is performed on the input data with the use of a filter or kernel (these terms are used interchangeably) to then produce a feature map.**



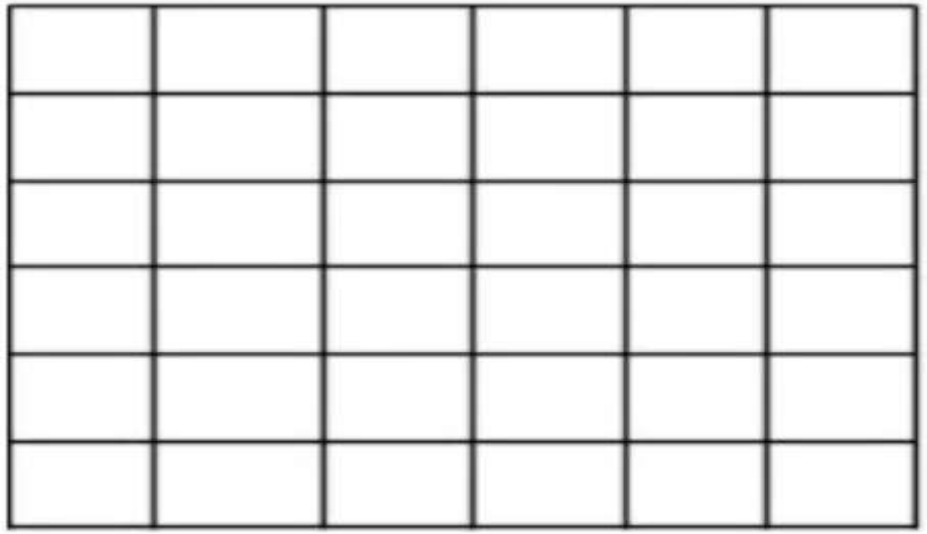
STRIDING -

- **Stride denotes how many steps we are moving in each steps in convolution. By default it is one. We can observe that the size of output is smaller that input.**

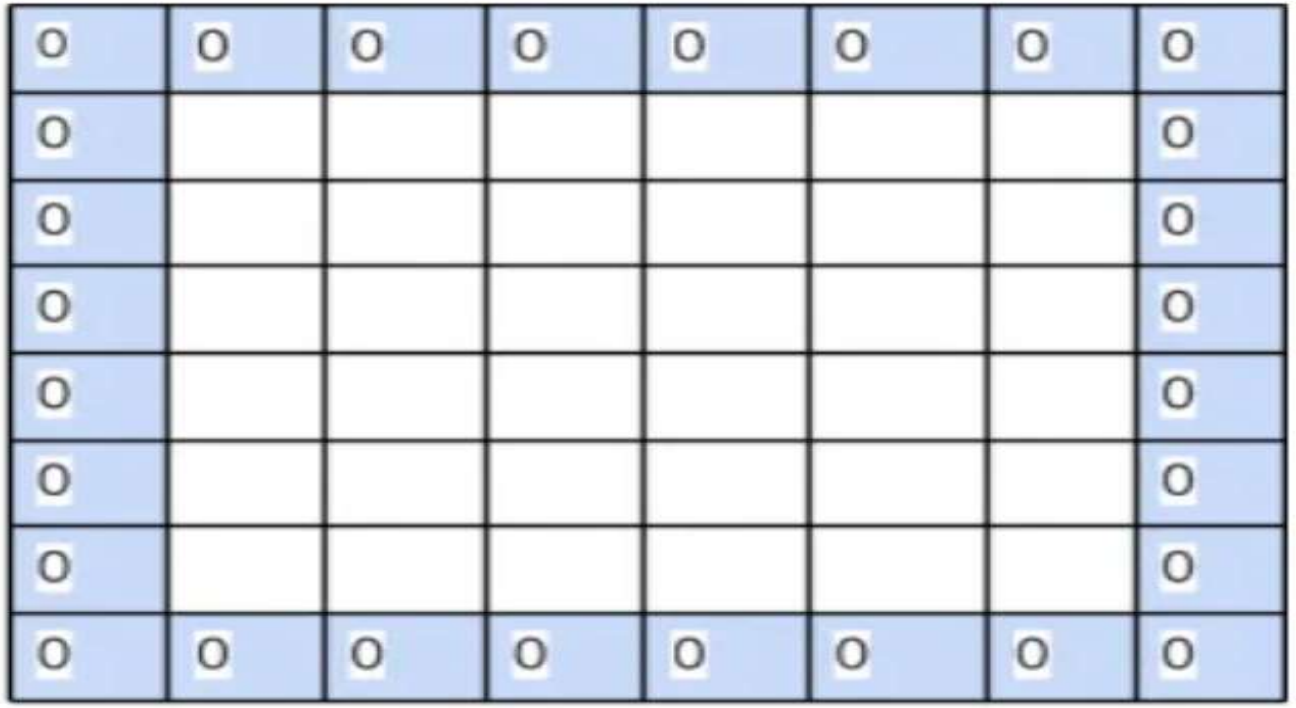


PADDING -

- **Stride denotes how many steps we are moving in each steps in convolution & By default it is one.**
- **We can observe that the size of output is smaller that input. To maintain the dimension of output as in input , we use padding. Padding is a process of adding zeros to the input matrix symmetrically.**
- **An element's padding area is the space between its content and its border.**
- **Padding creates extra space within an element. In contrast, margin creates extra space around an element.**



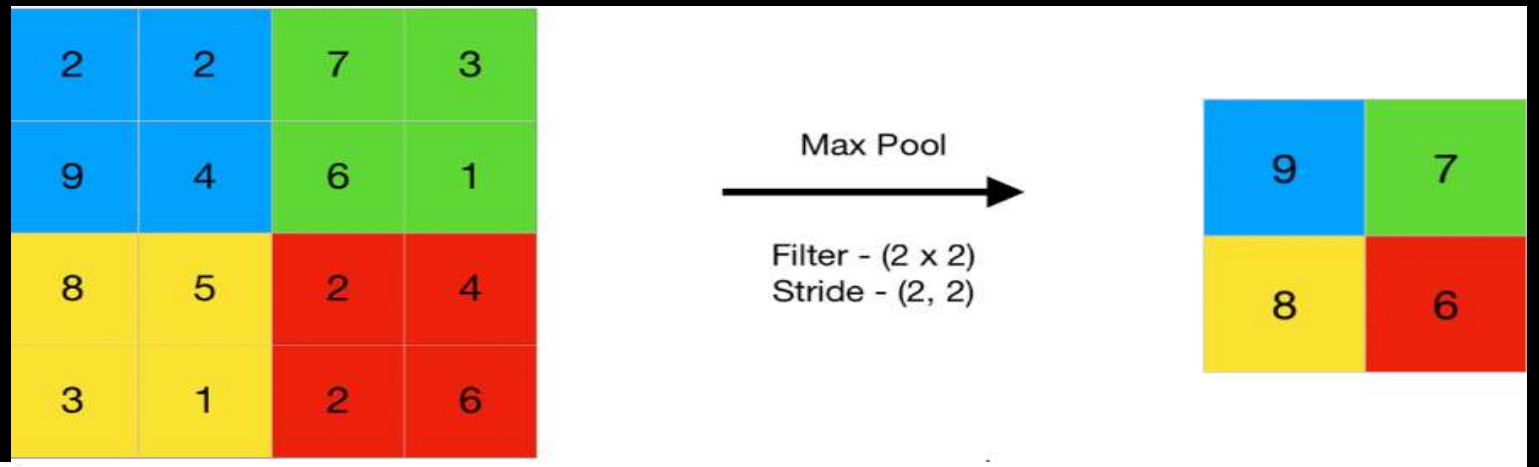
6x6 image

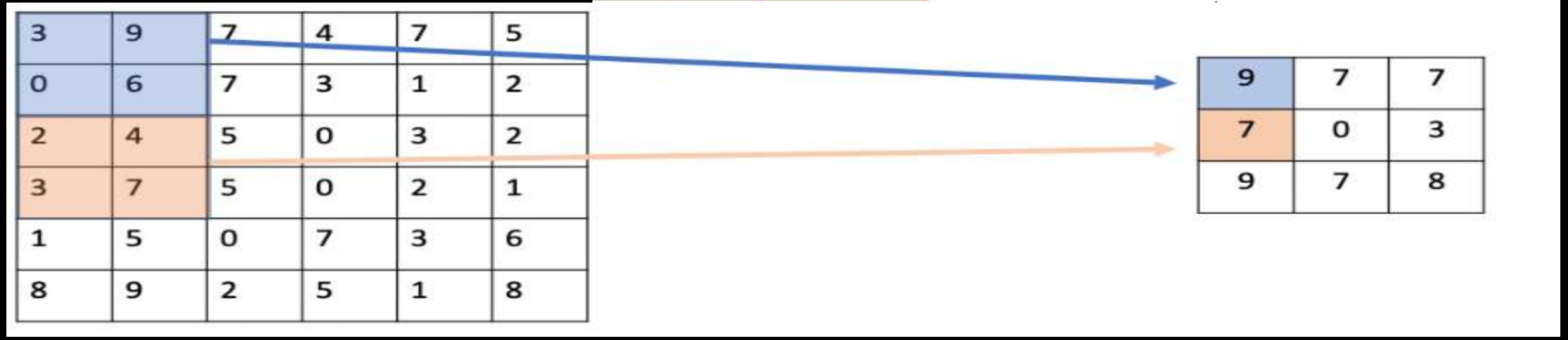


6x6 image with 1 layer of zero padding

MAX POOLING →

- **This works by selecting the maximum value from every pool.**
- **Max Pooling retains the most prominent features of the feature map, and the returned image is sharper than the original image.**
- **Max Pooling is a pooling operation that calculates the maximum value for patches of a feature map, and uses it to create a down sampled (pooled) feature map.**
- **It is usually used after a convolutional layer.**

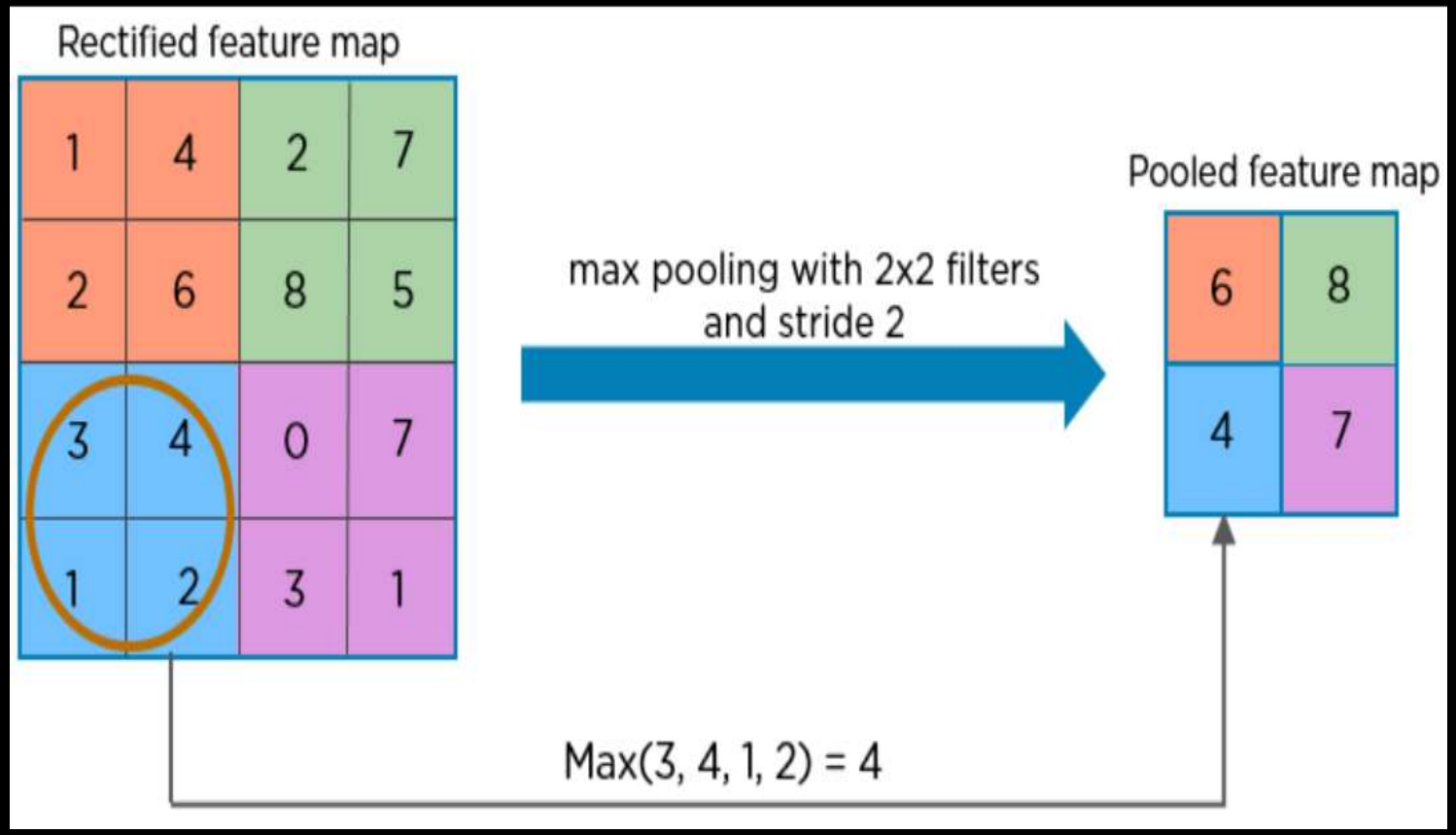


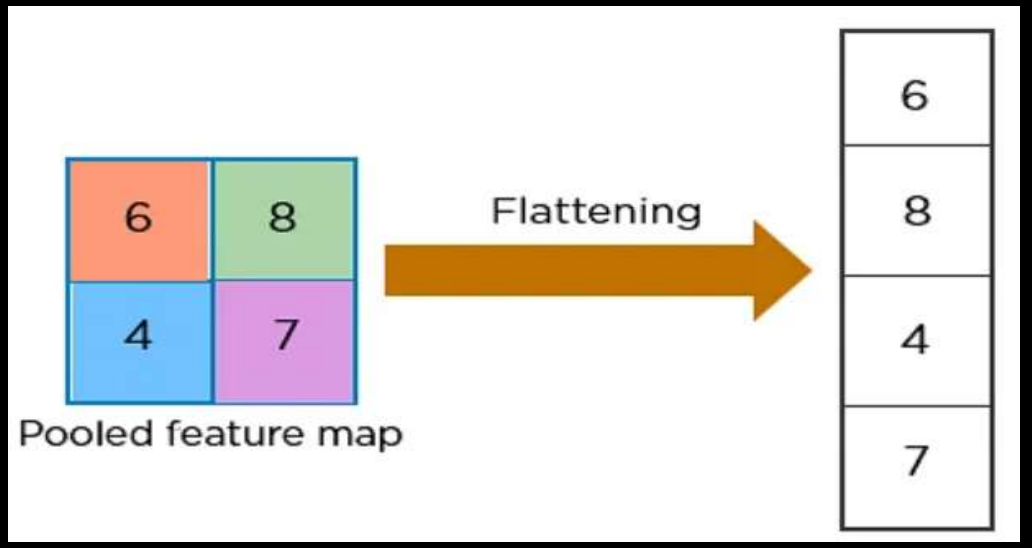


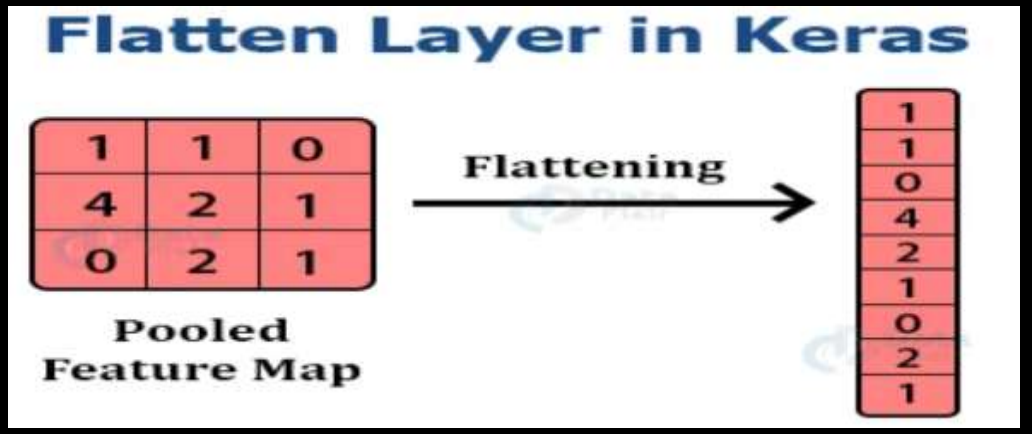
FLATTENING LAYER ->

- Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

- **The flattened matrix is fed as input to the fully connected layer to classify the image.**
- **Flatten layer is used to make the multidimensional input one-dimensional, commonly used in the transition from the convolution layer to the full connected layer.**





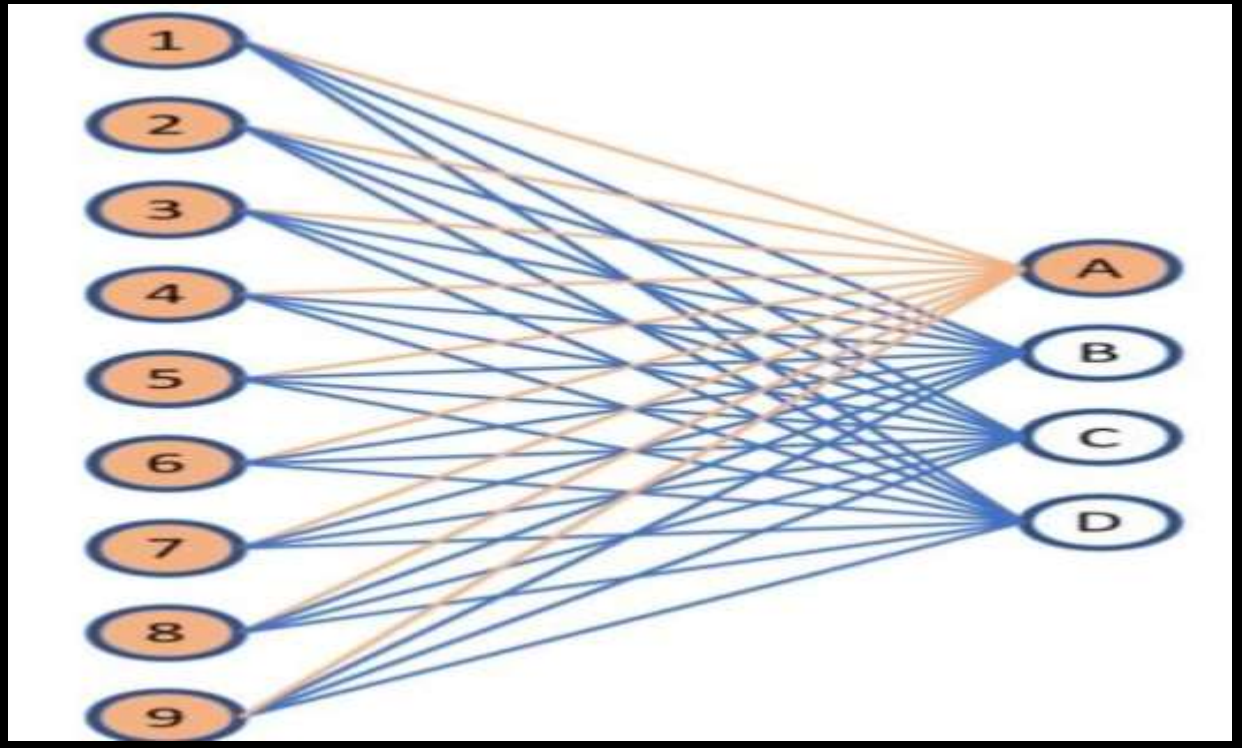


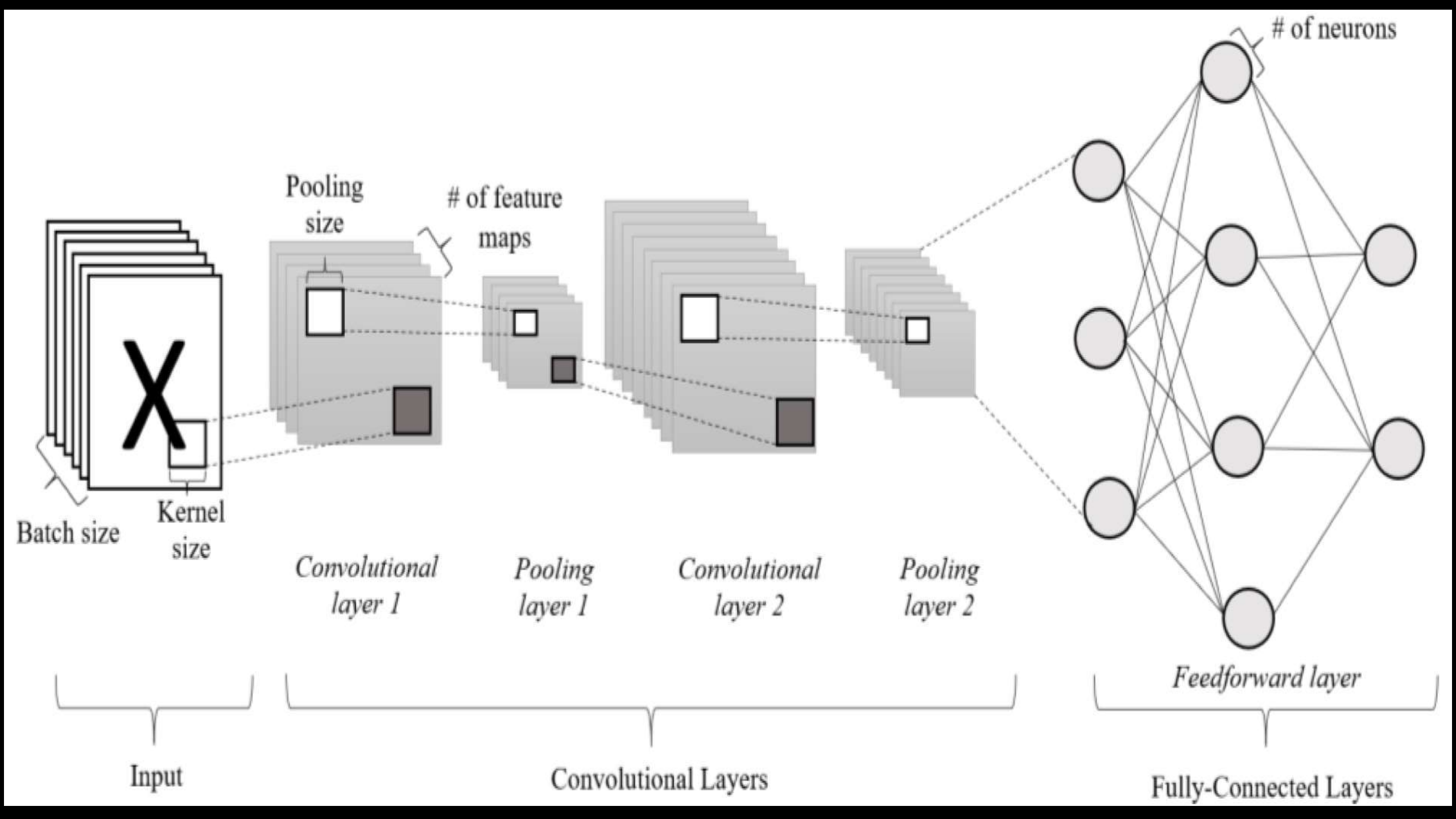
FULLY CONNECTED LAYER -

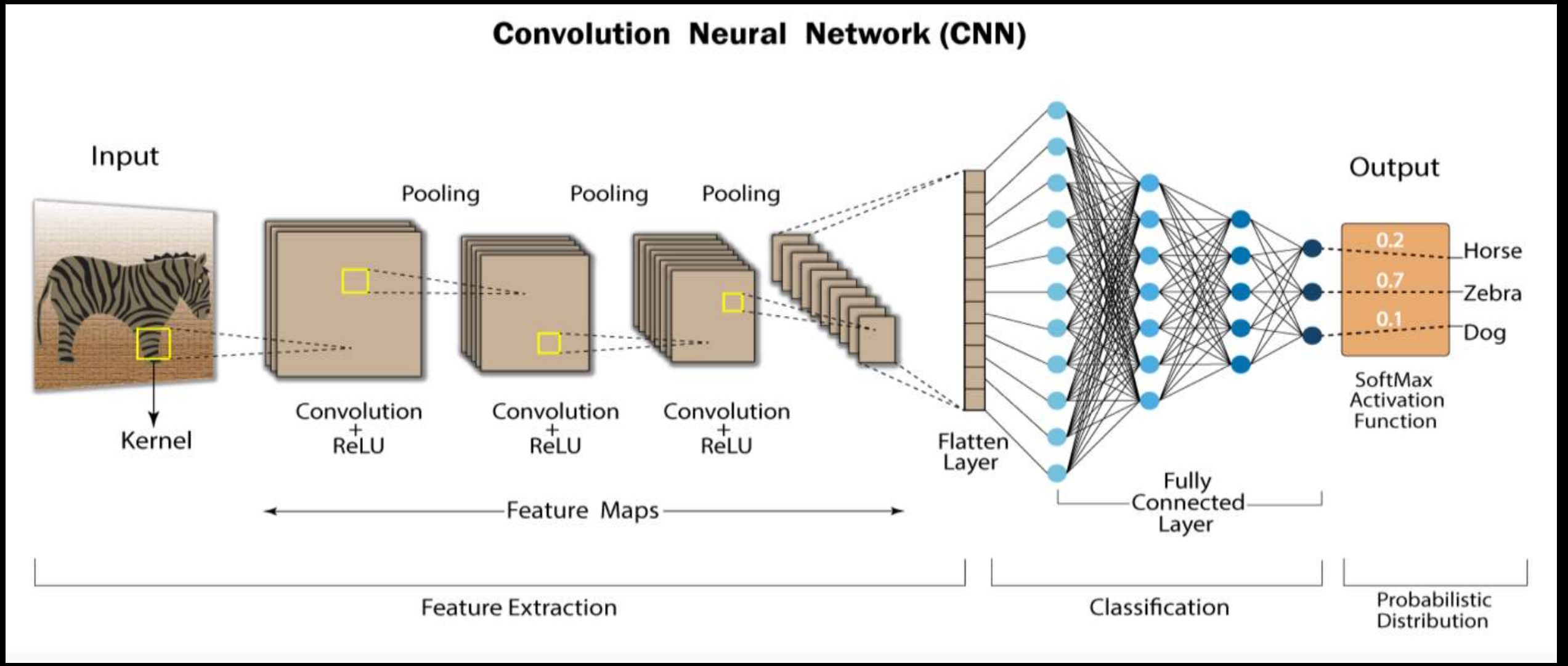
- A fully connected layer refers to a neural network in which each neuron applies a linear transformation to the input vector through a weights matrix. as a result, all possible connections layer-to-layer are present, meaning every input of the input vector influences every output of the output vector.

- Fully connected layers are an essential component of Convolutional Neural Networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision. The CNN process begins with convolution and pooling, breaking down the image into features, and analyzing them independently.

- The Fully connected layer (as we have in ANN) is used for classifying the input image into a label. This layer connects the information extracted from the previous steps (i.e Convolution layer and Pooling layers) to the output layer and eventually classifies the input into the desired label




Pooling size
# of feature maps
# of neurons
Batch size
Kernel size
Convolutional layer 1
Pooling layer 1
Convolutional layer 2
Pooling layer 2
Feedforward layer
Input
Convolutional Layers
Fully-Connected Layers



PROJECT-3: MOOD CLASSIFICATION USING TENSORFLOW & KERAS (CNN)

- This project helps to classify the mood as per trained image
- We will work using CNN model


?
Q
&
!
A