Assignment No. 03

Problem Statement

Build the Image Classification model by dividing the model into following 4 stages:

1. Import the necessary packages
2. Loading and preprocessing the image data
3. Defining the model's architecture
4. Training the model
5. Estimating the model's performance

Solution Expected

Implement and train a model for image processing and classification using CNN on a image dataset and improve model generalization by achieving increased accuracy and descresesd loss where model gains good confidence with the prediction.

Objectives to be achieved

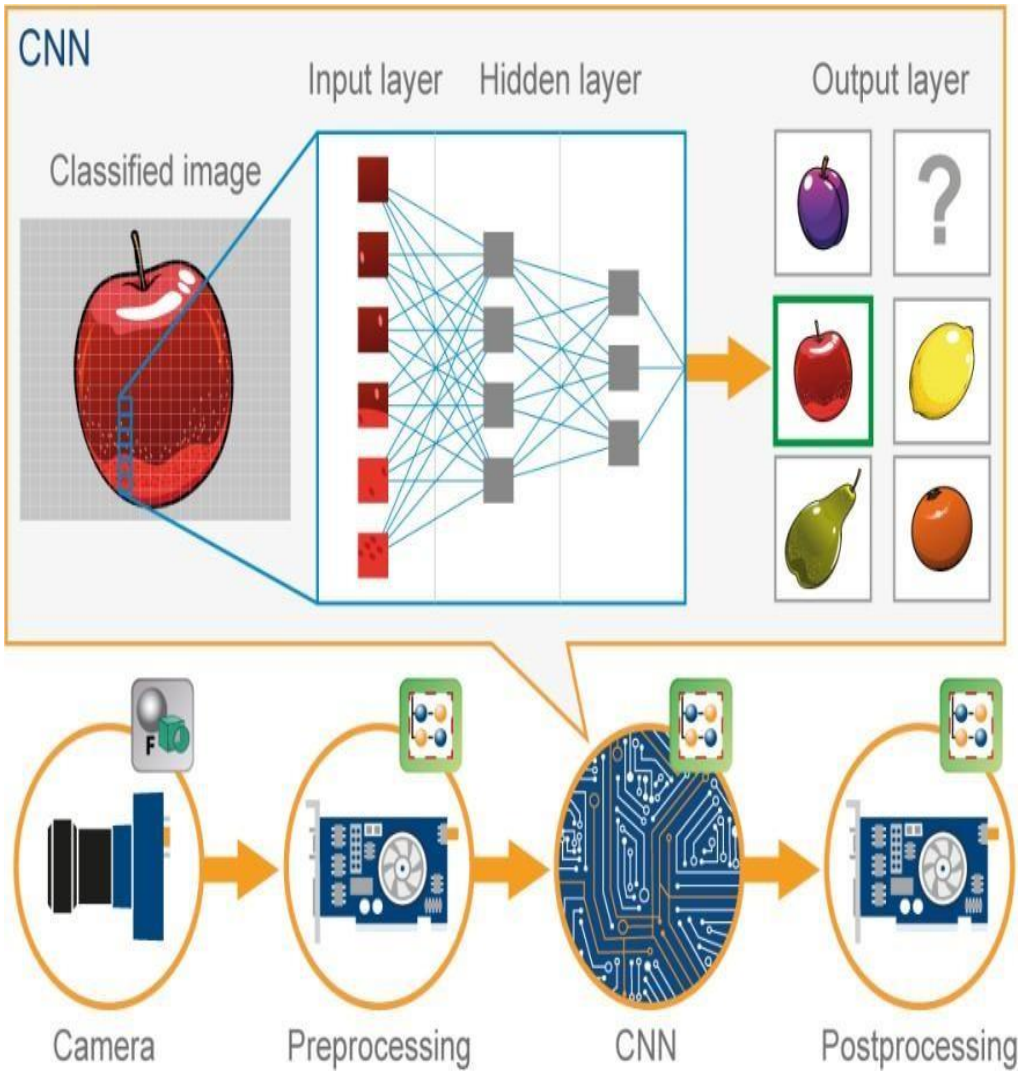
1. Understand how to use Tensorflow Eager and Keras Layers to build neural network architecture.
2. Understand how a model is trained and evaluated.
3. Identify digits from images.
4. Our main goal is to train a neural network (using Keras) to obtain > *90%* accuracy on MNIST dataset.
5. Research at least 1 technique that can be used to improve model generalization.

Methodology to be used

- Deep Learning
- Convolutional Neural Network

Introduction

In the past few years, Deep Learning has been proved that it's a very powerful tool due to its ability to handle huge amounts of data. The use of hidden layers exceeds traditional techniques, especially for pattern recognition. One of the most popular Deep Neural Networks is Convolutional Neural Networks (CNN).



A convolutional neural network (CNN) is a type of **Artificial Neural Network (ANN)** used in image recognition and processing which is specially designed for processing data (pixels).

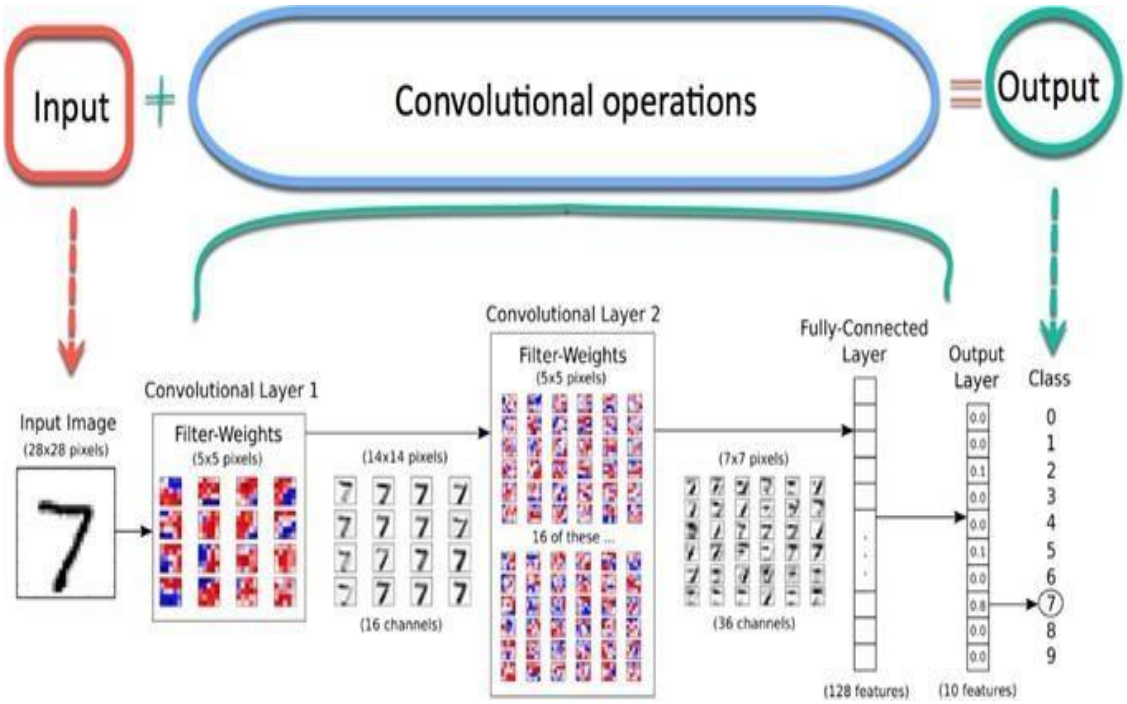


Image Source: Google.com

Before moving further we need to understand what is the neural network? Let's go…

Neural Network:

A neural network is constructed from several interconnected nodes called **"neurons"**. Neurons are arranged into the **input layer, hidden layer, and output layer.** The input layer corresponds to our predictors/features and the Output layer to our response variable/s.

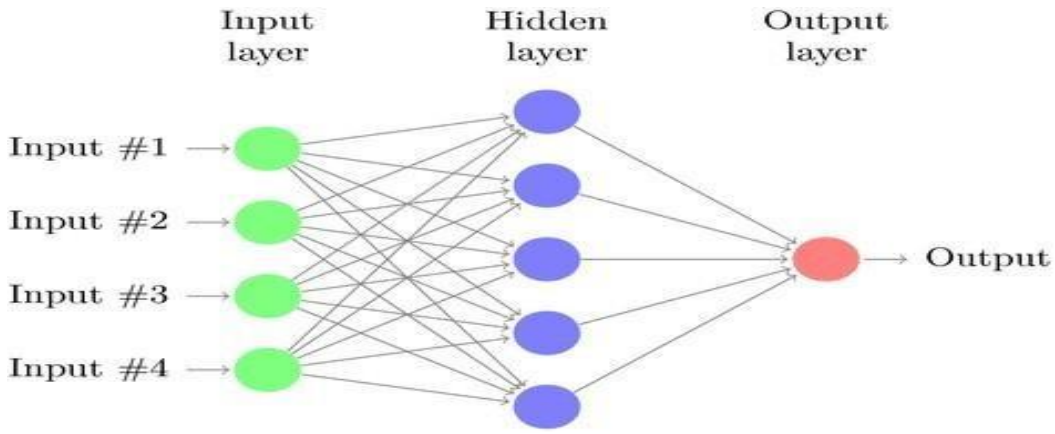


Image Source: Google.com

Multi-Layer Perceptron (MLP):

The neural network with an input layer, one or more hidden layers, and one output layer is called a **multi-layer perceptron (MLP).** MLP is invented by **Frank Rosenblatt** in the year of 1957. MLP given below has 5 input nodes, 5 hidden nodes with two hidden layers, and one output node

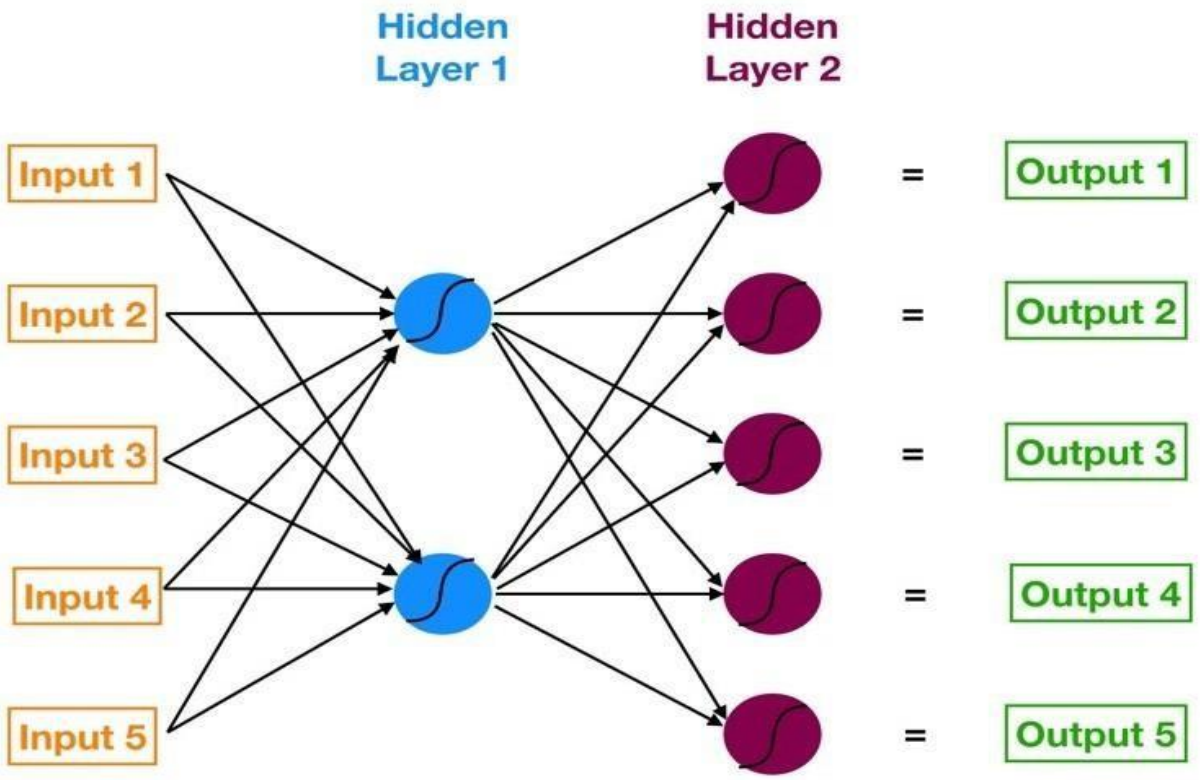


Image Source: Google.com

How does this Neural Network work?

– Input layer neurons receive incoming information from the data which they process and distribute to the **hidden layers**.

– That information, in turn, is processed by hidden layers and is passed to the output **neurons**.

– The information in this artificial neural network(ANN) is processed in terms of one **activation function**. This function actually imitates the brain neurons.

– Each neuron contains a value of **activation functions** and a **threshold value.**

– The **threshold value** is the minimum value that must be possessed by the input so that it can be activated.

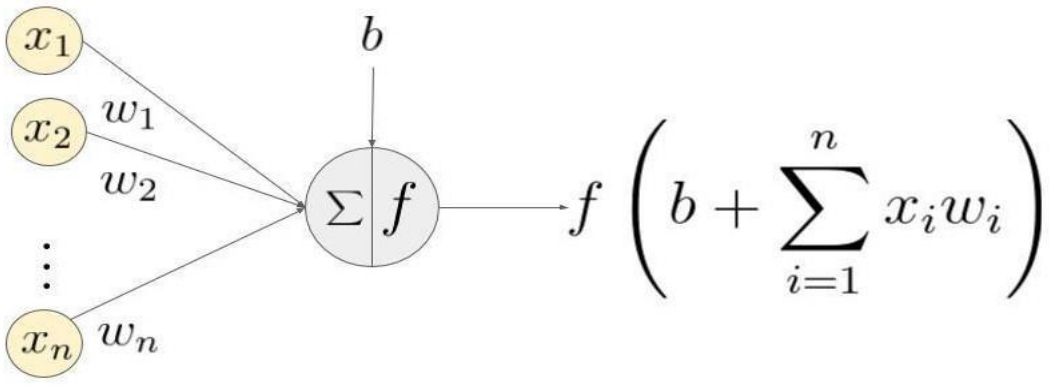
– The task of the neuron is to perform a weighted sum of all the input signals and apply the activation function on the sum before passing it to the next(hidden or output) layer.

Let us understand what is weightage sum?

Say that, we have values $a1$, $a2$, $a3$, $a4$ for input and weights as w1, w2, w3, w4 as the input to one of the hidden layer neuron say nj, then the weighted sum is represented as

j = **σ =1to4** w ***a** + bj

where bj : bias due to node



An example of a neuron showing the input (x_1 - x_n), their corresponding weights (w_1 - w_n), a bias (b) and the activation function f applied to the weighted sum of the inputs.

Image Source: Google.com

What are the Activation Functions?

These functions are needed to introduce a non-linearity into the network. The activation function is applied and that output is passed to the next layer.

Possible Functions

- **Sigmoid:** Sigmoid function is differentiable. It produces output between 0 and 1.
- **Hyperbolic Tangent:** Hyperbolic Tangent is also differentiable. This Produces output between -1 and 1.
- **ReLU:** ReLU is most popular function. ReLU is used widely in deep learning.
- **Softmax:** The softmax function is used for multi-class classification problems. It is a generalization of the sigmoid function. It also produces output between 0 and 1

Convolutional Neural Nework:

Now imagine there is an image of a bird, and you want to identify it whether it is really a bird or something other. The first thing you should do is feed the pixels of the image in the form of arrays to the input layer of the neural network (MLP networks used to classify such things). Thehidden layers carry Feature Extraction by performing various calculations and operations. There are multiple hidden layers like the convolution, the ReLU, and the pooling layer that performs feature extraction from your image. So finally, there is a fully connected layer that you can see which identifies the exact object in the image. You can understand very easily from thefollowing figure:

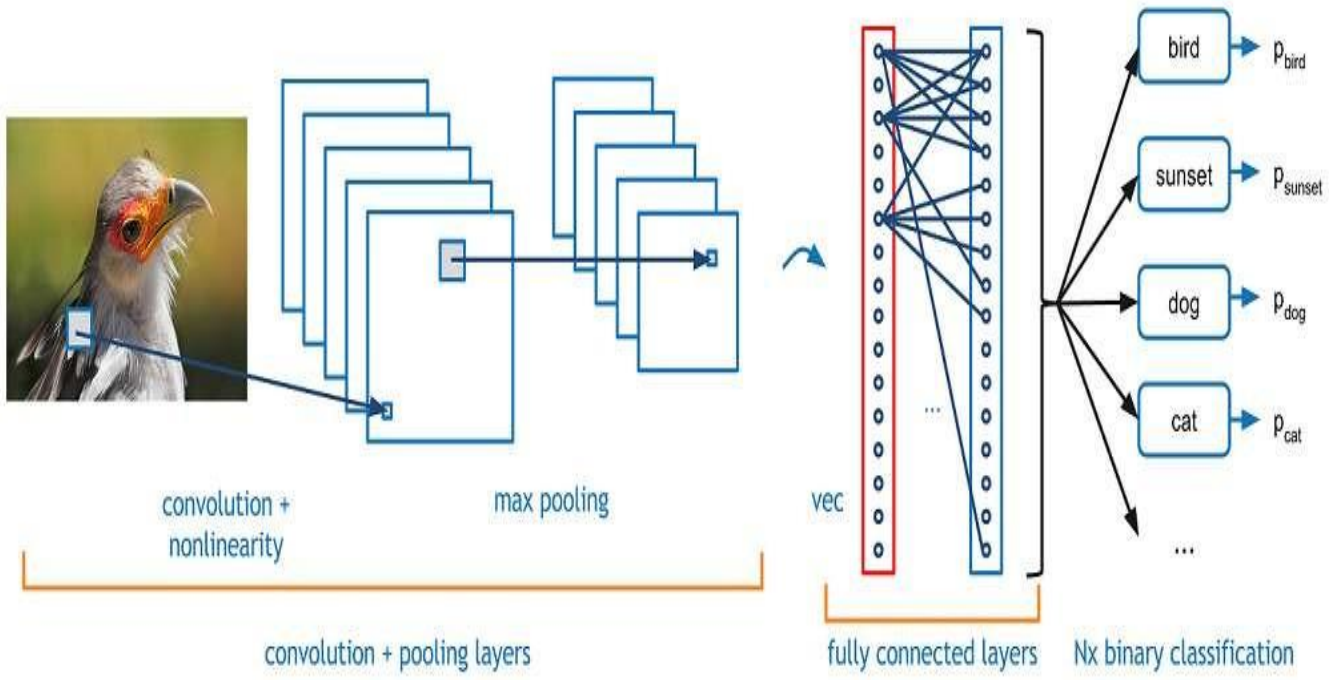


Image Source: Google.com

Convolution:-

Convolution Operation involves matrix arithmetic operations and every image is represented in the form of an array of values (pixels).

Let us understand example:

a = [2, 5, 8, 4, 7, 9]

b = [1, 2, 3]

In Convolution Operation, the arrays are multiplied one by one element-wise, and the product is grouped or summed to create a new array that represents **a*b**.

The first three elements of matrix **a** are now multiplied by the elements of matrix **b**. The product is summed to get the result and stored in a new array of **a*b.**

This process remains continuous until the operation gets completed.

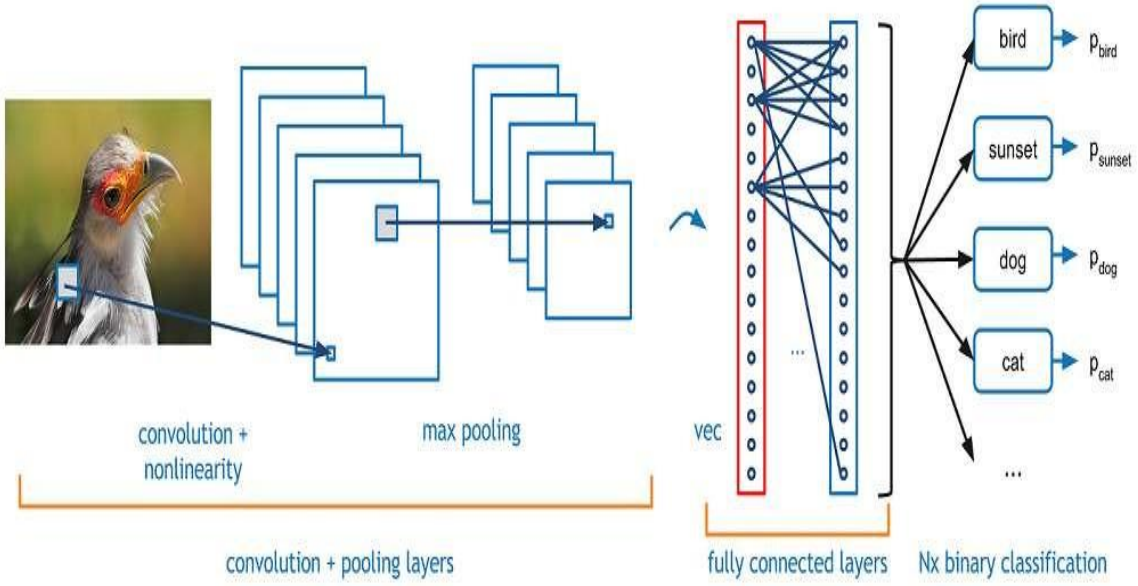


Image Source: Google.com

Pooling:

After the convolution, there is another operation called pooling. So, in the chain, convolution and pooling are applied sequentially on the data in the interest of extracting some features from the data. After the sequential convolutional and pooling layers, the data is flattened into a feed- forward neural network which is also called a Multi-Layer Perceptron.

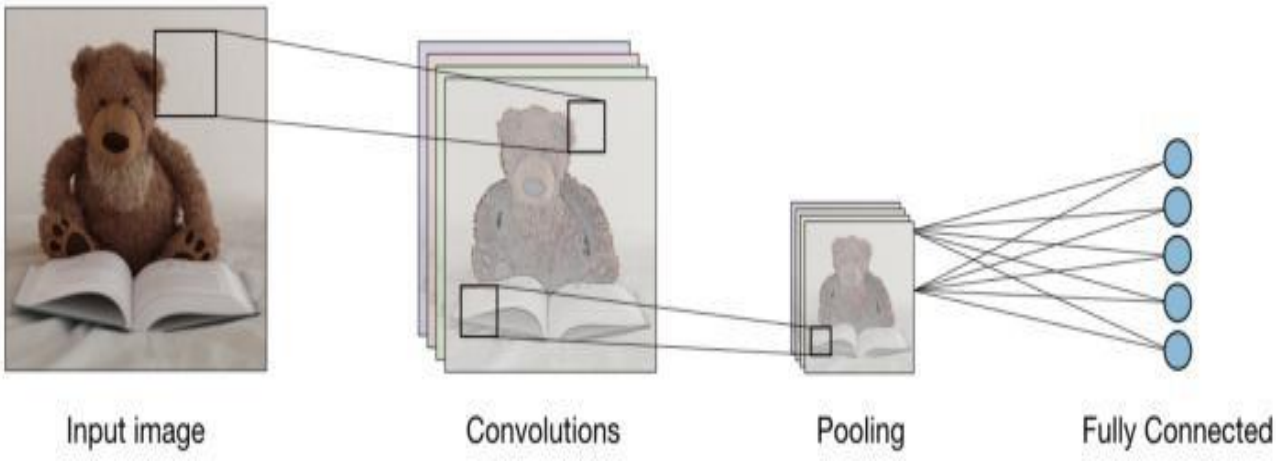


Image Source: Google.com

Conclusion: Thus we have created a machine learning model for image processing and classification using CNN and found sufficient accuracy.