**IBM Naan Mudhalvan**

**Cloud Application Development**

**(Image recognition and IBM**

**Cloud visual recognition)**

**Phase-3 Project**

**Image recognition using CNN tool**

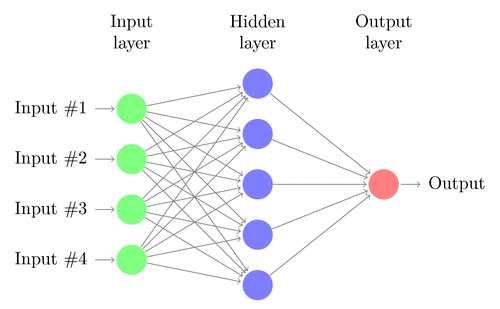
**Introduction:**

Deep Learning has been proved that its 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).

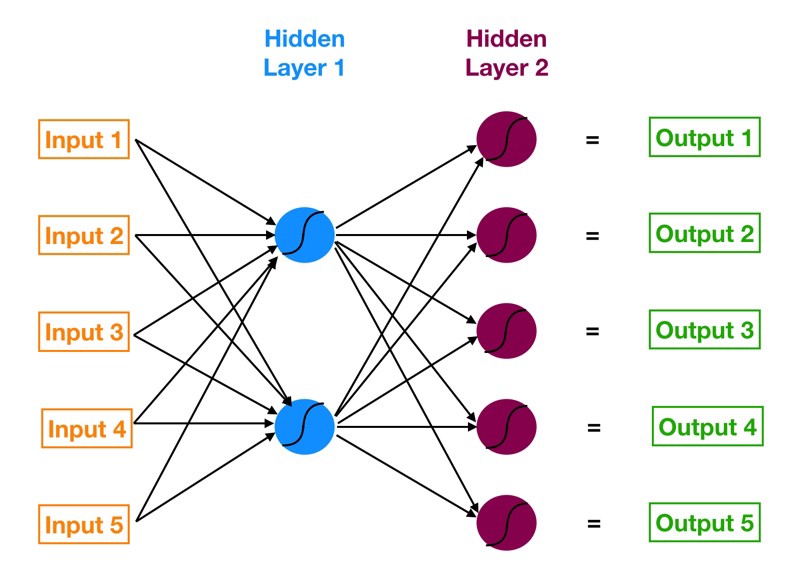
## Neural Network:

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.



**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



**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.

# image recognition

Image recognition, in the context of [machine vision](https://www.techtarget.com/searchenterpriseai/definition/machine-vision-computer-vision), is the ability of software to [identify objects](https://www.techtarget.com/whatis/definition/object-recognition), places, people, writing and actions in digital images. Computers can use machine vision technologies in combination with a camera and artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) software to achieve image recognition.

Steps to achieve Image recognition:

* Collecting the Dataset
* Importing Libraries and Splitting the Dataset
* Building the CNN
* Full Connection
* Data Augmentation
* Training our Network
* Testing

# **Step 1 — Collecting the Dataset**

In order to train our machine, we need a huuuuggge amount of data so that our model can learn from them by identifying out certain relations and common features related to the objects.

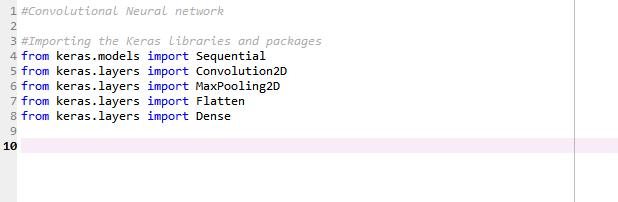
For example consider:



In above we have images of apples and oranges these set of images are required for the image recognition program to be identify as one of them

# **Step 2 — Importing Libraries and Splitting the Dataset**

To use the powers of the libraries, we first need to import them.



After importing the libraries, we need to split our data into two parts- taining\_set and test\_set.

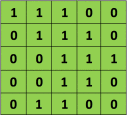
# **Step 3 — Buliding the CNN**

This is most important step for our network. It consists of three parts

* Convolution
* Polling
* Flattening

The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

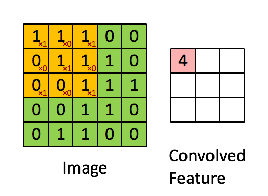
Since every image can be considered as a matrix of pixel values. Consider a 5 x 5 image whose pixel values are only 0 and 1 (note that for a grayscale image, pixel values range from 0 to 255, the green matrix below is a special case where pixel values are only 0 and 1):



Also, consider another 3 x 3 matrix as shown below:

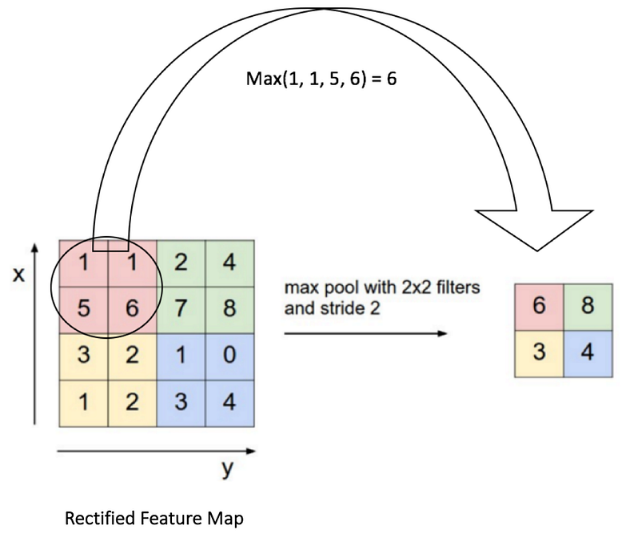


Then, the Convolution of the 5 x 5 image and the 3 x 3 matrix can be computed as shown in the animation in **Figure 5** below:



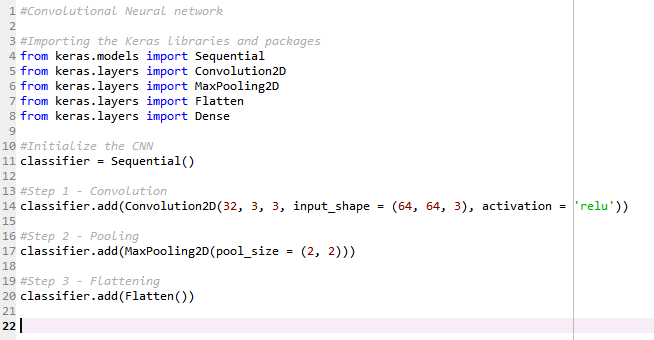
The obtained matrix is also known as the feature map. An additional operation called ReLU is used after every Convolution operation. The next step is of pooling.

Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. In case of Max Pooling, we define a spatial neighborhood (for example, a 2×2 window) and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average (Average Pooling) or sum of all elements in that window. In practice, Max Pooling has been shown to work better.

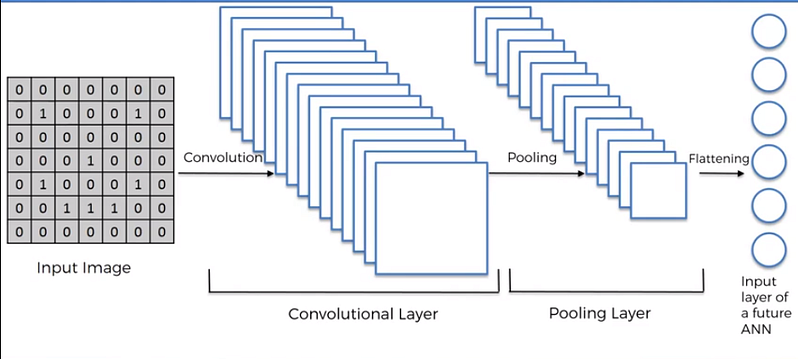


After pooling comes flattening. Here the matrix is converted into a linear array so that to input it into the nodes of our neural network.

Let’s come to the code.

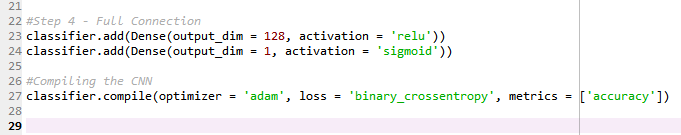


So now our CNN network looks like this



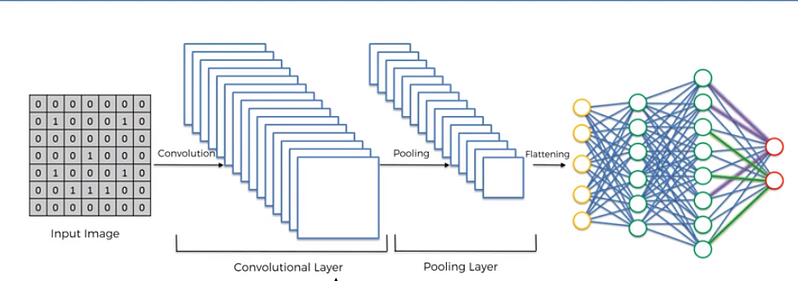
# **Step 4 — Full Connection**

Full connection is connecting our convolutional network to a neural network and then compiling our network.



Here we have made 2 layer neural network with a sigmoid function as an activation function for the last layer as we need to find the probability of the object being a cat or a dog.

So now the final network looks something like this -



# **Step 5 — Data Augmentation**

While training your data, you need a lot of data to train upon. Suppose we have a limited number of images for our network. What to do now??

You don’t need to hunt for novel new images that can be added to your dataset. Why? Because, neural networks aren’t smart to begin with. For instance, a poorly trained neural network would think that these three tennis balls shown below, are distinct, unique images.



The same tennis ball, but translated.

So, to get more data, we just need to make minor alterations to our existing dataset. Minor changes such as flips or translations or rotations. Our neural network would think these are distinct images anyway.

Data augmentation is a way we can reduce overfitting on models, where we increase the amount of trainingdata using information only in our training data. The field of data augmentation is not new, and in fact, various data augmentation techniques have been applied to specific problems.

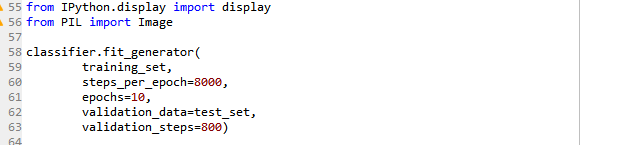
And here goes the code



Now we have a huge amount of data and its time for the training.

# **Step 6 — Training our Network**

So, we completed all the steps of construction and its time to train our model.



If you are training with a good video card with enough RAM (like an Nvidia GeForce GTX 980 Ti or better), this will be done in less than an hour. If you are training with a normal cpu, it might take a lot longer.

With increasing number of epochs, the accuracy will increase.

# Step 7 — Testing

Now lets test a random image.



And the code will be

