Experiment No: 03

Aim: To Implement Image classification model using CNN Deep Learning Architecture.

Problem Statement: Build the Image classification model using CNN Deep Learning Architecture by dividing the model into following 4 stages:

a. Loading and preprocessing the image data

b. Defining the model's architecture

c. Training the model

d. Estimating the model's performance

Objectives:

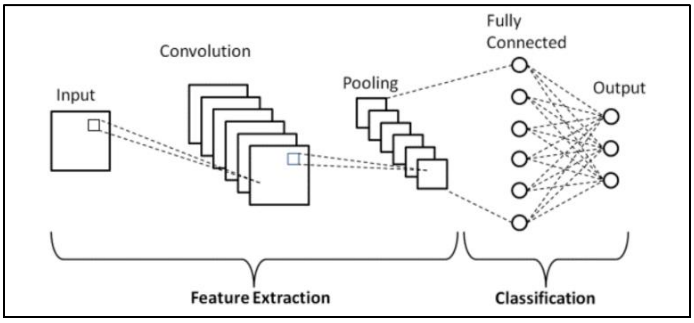
a) Apply CNN Deep Learning architecture on MNIST/CIFAR dataset for classification

b) Evaluate the performance of model in terms of accuracy, loss and number of epochs

Theory:

The goal of a CNN is to learn higher-order features in the data via convolutions. They are well suited to object recognition with images and consistently top image classification competitions. They can identify faces, individuals, street signs, platypuses, and many other aspects of visual data. CNNs overlap with text analysis via optical character recognition, but they are also useful when analyzing words6 as discrete textual units. They're also good at analyzing sound. The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning. As Figure 4-7 illustrates, CNNs are good at building position and (somewhat) rotation invariant features from raw image data. CNNs are powering major advances in machine vision, which has obvious applications for self-driving cars, robotics, drones, and treatments for the visually impaired.

<u>Architecture of CNN:</u>

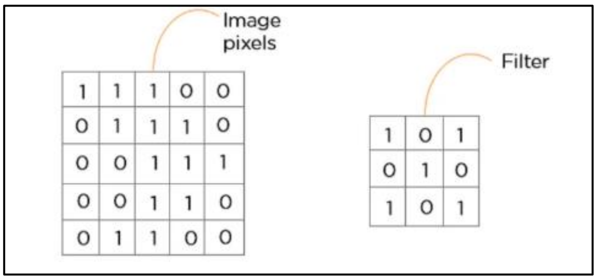


A convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer

1. Convolution Layer

This is the first step in the process of extracting valuable features from an image. A convolution layer has several filters that perform the convolution operation. Every image is considered as a matrix of pixel values. Consider the following 5x5 image whose pixel values are either 0 or 1.

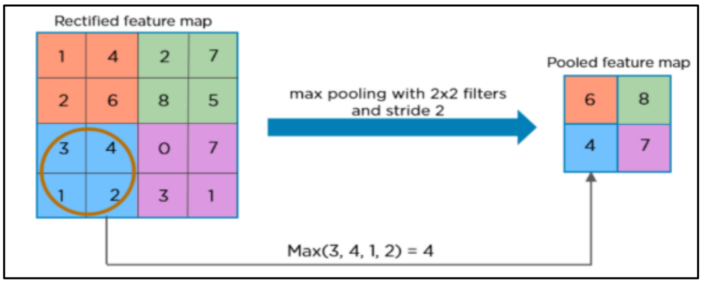


2. ReLU layer

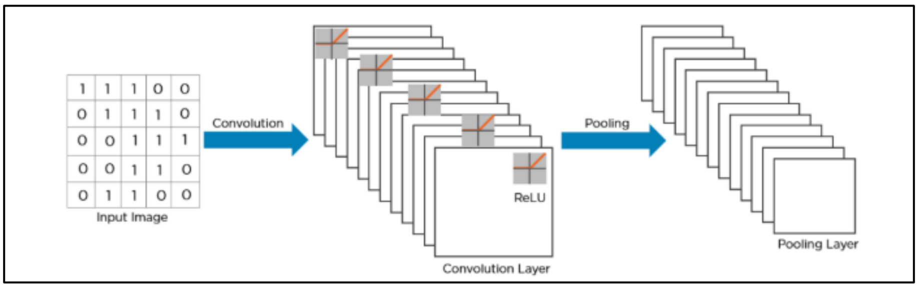
ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer. ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map.

3. Pooling Layer

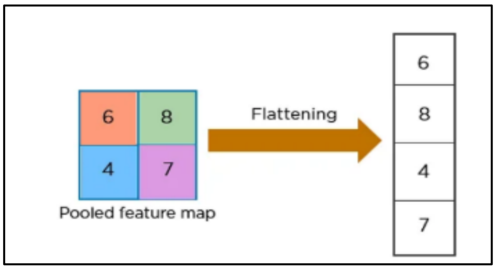
Pooling is a down-sampling operation that reduces the dimensionality of the feature map. The rectified feature map now goes through a pooling layer to generate a pooled feature map.



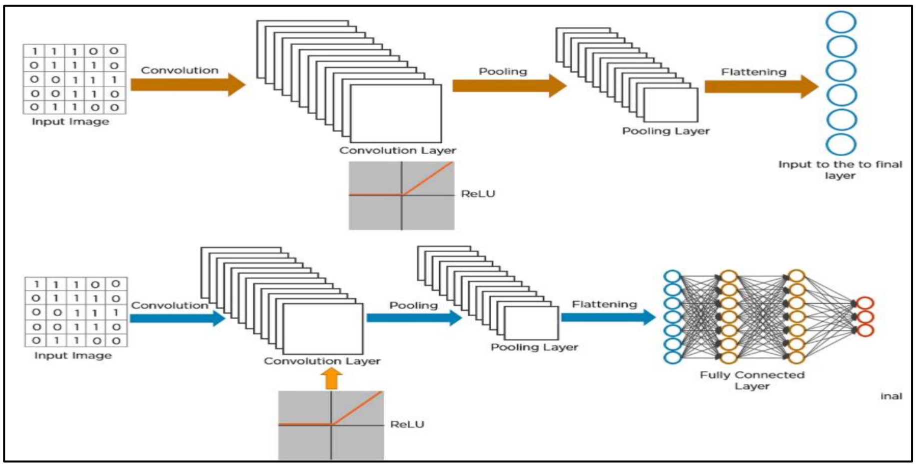
The pooling layer uses various filters to identify different parts of the image like edges, corners, body, feathers, eyes, and beak. Here's how the structure of the convolution neural network looks so far:



The next step in the process is called flattening. 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.



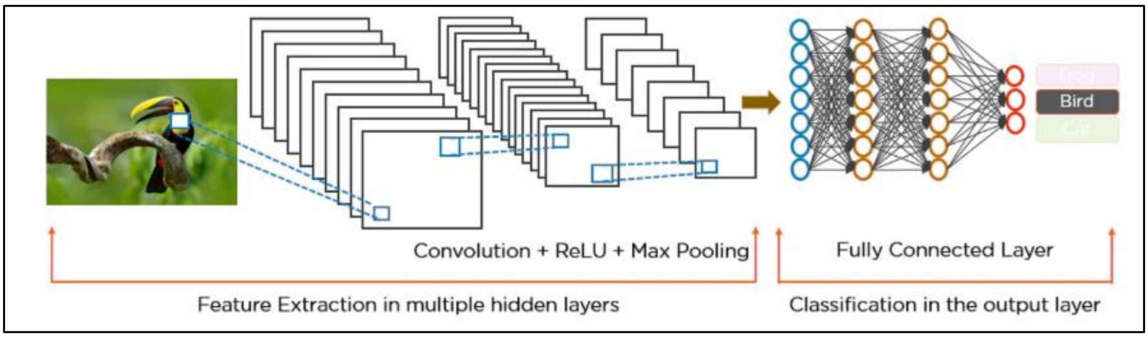
The flattened matrix is fed as input to the fully connected layer to classify the image.



The pixels from the image are fed to the convolutional layer that performs the convolution operation

1. It results in a convolved map
2. The convolved map is applied to a ReLU function to generate a rectified feature map
3. The image is processed with multiple convolutions and ReLU layers for locating the features
4. Different pooling layers with various filters are used to identify specific parts of the image

The pooled feature map is flattened and fed to a fully connected layer to get the final output



Conclusion:

Thus, we have implemented the Image classification model using CNN. With above code we can see that sufficient accuracy has been met. Throughout the epochs, our model accuracy increases and loss decreases that is good since our model gains confidence with our prediction

This indicates the model is trained in a good way:

1. The loss is decreasing and the accuracy is increasing with every epoch.
2. The test accuracy is the measure of how good the model is predicting so, it is observed that the model is well trained after 10 epochs