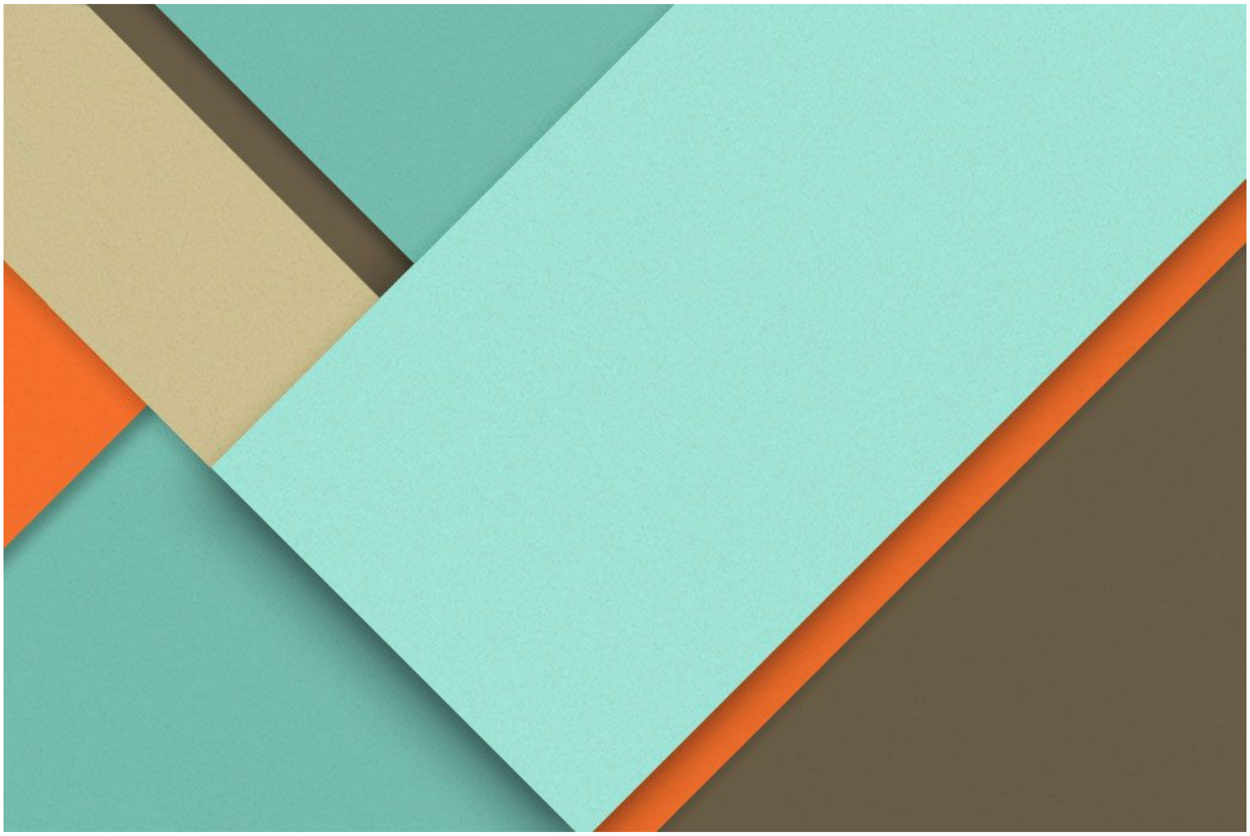


Image Classification using Convolutional Neural Networks

25.11.2019

Aman Jain | 160050019

Rishabh Kumar | 160040077

Objective

To classify an image using the concept of convolutional neural networks.

Data Used

We've used CIFAR-10 to train the model.

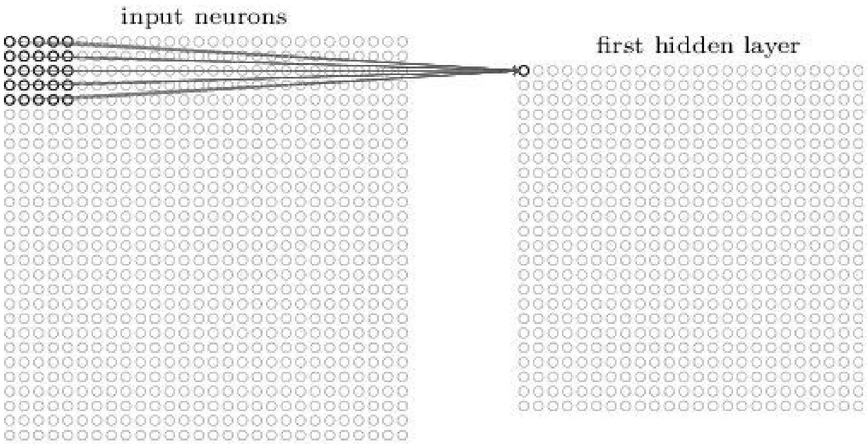
The CIFAR-10 dataset (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. The 10 different classes represent aeroplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.

Methodology

We are using a convolutional neural network followed by flattening and two fully connected layers. The final output of this whole network is a vector of size 10. The vector index with the highest value is the predicted label of the model.

All these components are explained below -

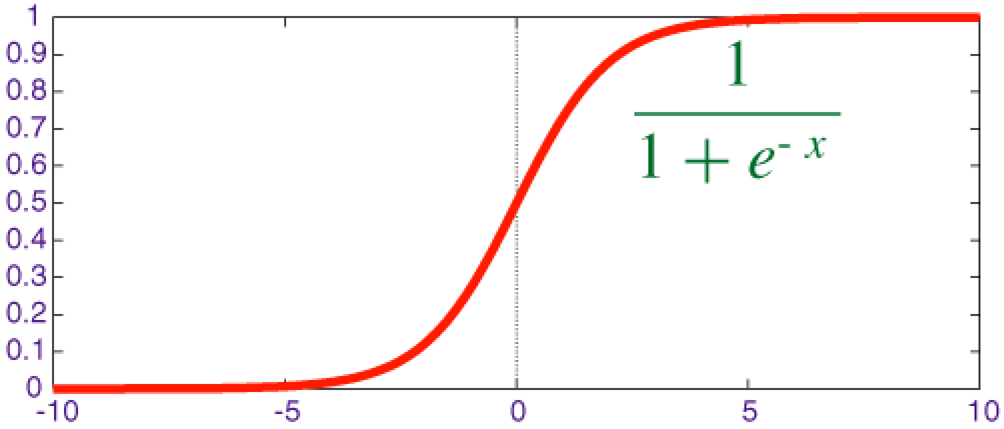
1. **Convolution**



In our implementation, we are using four kernels of size 8x8 with stride length being 3. This results in the convolution of 32x32 pixel values from the input image into four 9x9 pixel values matrices.

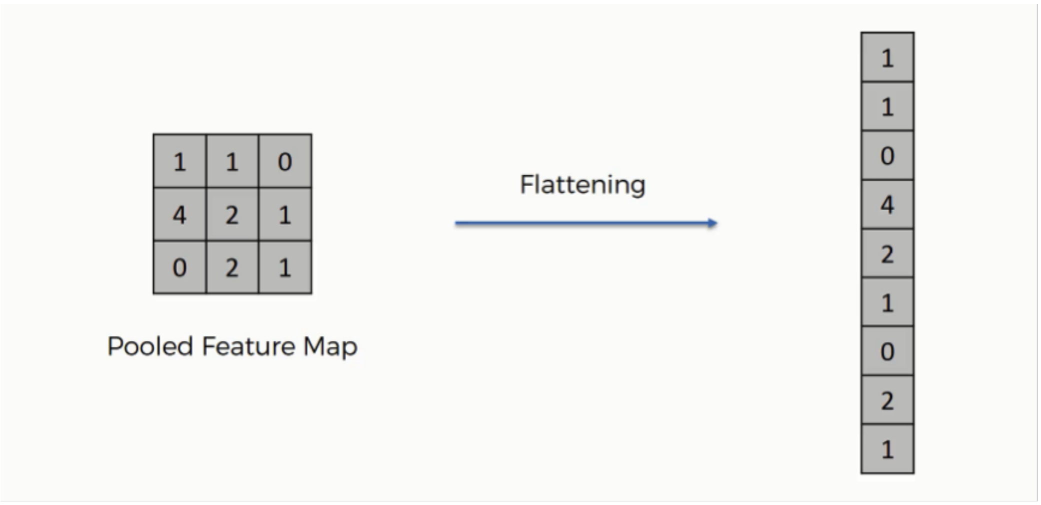
The weights in these kernels are learned by training using 5000 images from CIFAR dataset.

2. Sigmoid



After the convoluted matrices are obtained they are passed through a sigmoid layer. The sigmoid layer does an elementwise operation on matrices as shown in the above figure.

3. Flattening

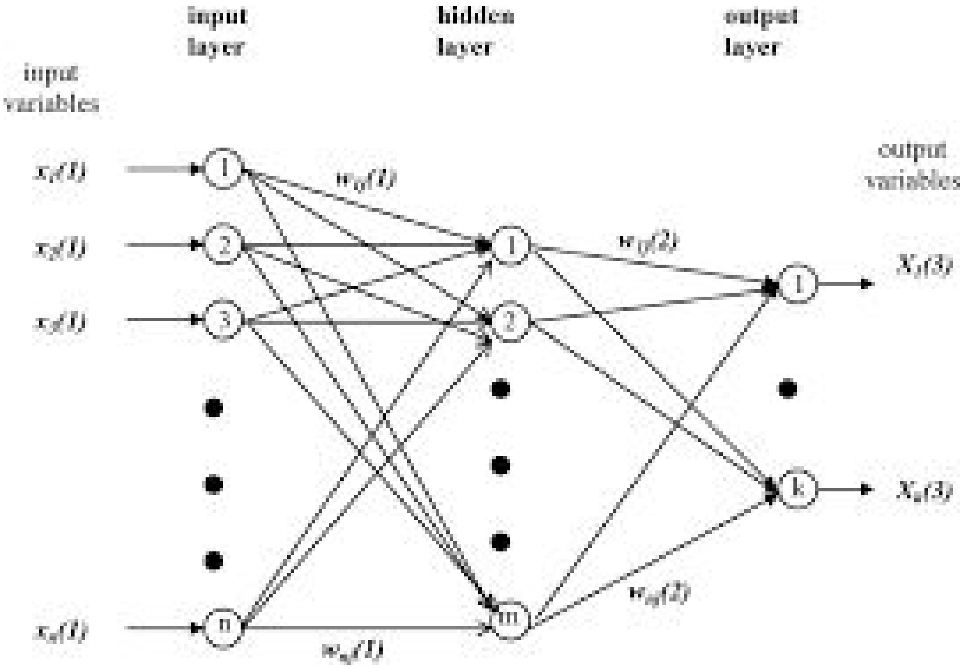


1	1	0
4	2	1
0	2	1

1
1
0
4
2
1
0
2
1

In this process, we are flattening the four 9x9 matrices produced from convolution and passing through sigmoid layer into 4 vectors of size 81 each. The conversion from matrices to vector is similar as shown in figure

4. Fully Connected Network

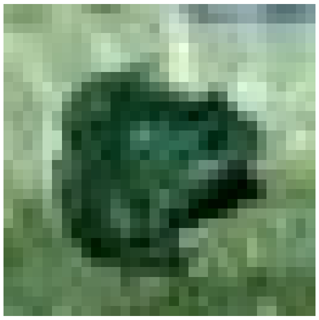


The results obtained after flattening (4*81) are now fed in a fully connected network. In fully connected network, all the inputs from one layer are connected to the every activation unit of the next layer. At every activation point, a weighted sum is done and sigmoid function is applied to finally give results. We have used two fully connected layers with number of nodes as follows - 4*81 -> 20 -> 10.

Results

We achieved an accuracy of 39.8% with training on only 5000 images on just a CPU (we didn’t use GPUs).

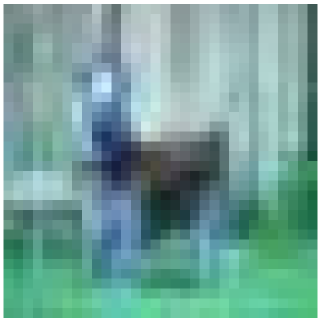
Some sample images and their predictions are given below -



Frog

Bird

Airplane



Bird

Discussion

The benchmark on CIFAR 10 dataset gives an accuracy of 99%. Our accuracy was 39.8%. This is because of the following reasons -

- We have a very simple architecture, the benchmarks use very advanced machine learning techniques such as residual networks, poolings etc.
- We had only limited computational power of CPU. Even training our simple model took 5-6 hours and even then we trained on only 5000 images.
- We wanted to build a deep understanding of each component listed above. So we did not use any library (from pytorch or tensorflow) to build the model, and instead wrote everything from scratch using only numpy library (which provides basic functionalities like matrix multiplication, etc.)