Deep Learning Lab 2018
Exercise 2 : Convolutional Neural Network

Name: Jatin Dhawan
Matriculation No. : 452322

Introduction: Convolutional Neural Networks are a special kind of Multi layer perceptrons (MLPs) which are built up on learnable learning weights and biases. Each Neuron in the network receives some input, performs a dot product and optionally add some non linearity. These network assumes inputs as Images and vastly reduces the number of parameters in the network. CNN also has a fully connected layer and softmax function which helps us to predict the final output. Figure 1 shows the model of 2 layer Convolutional Network also called as LeNet.

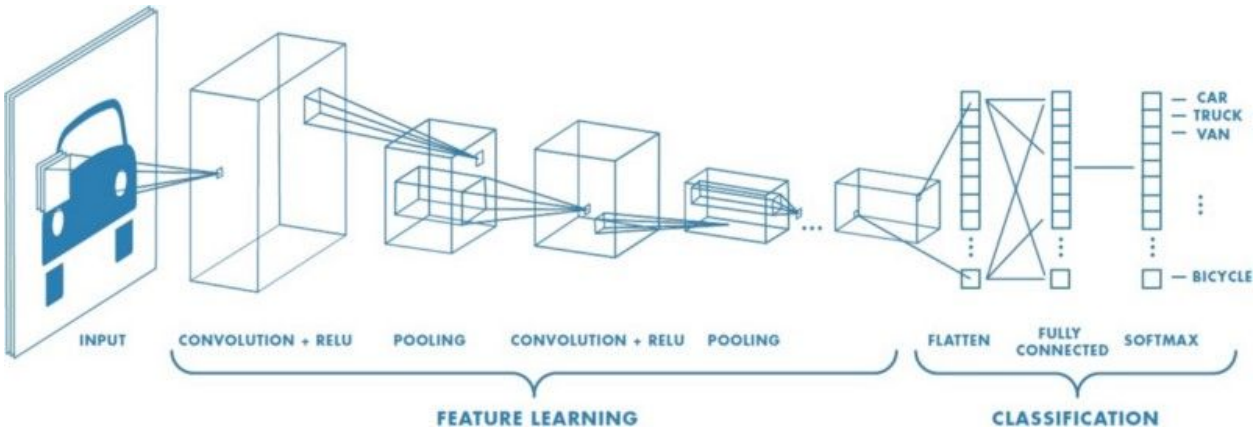
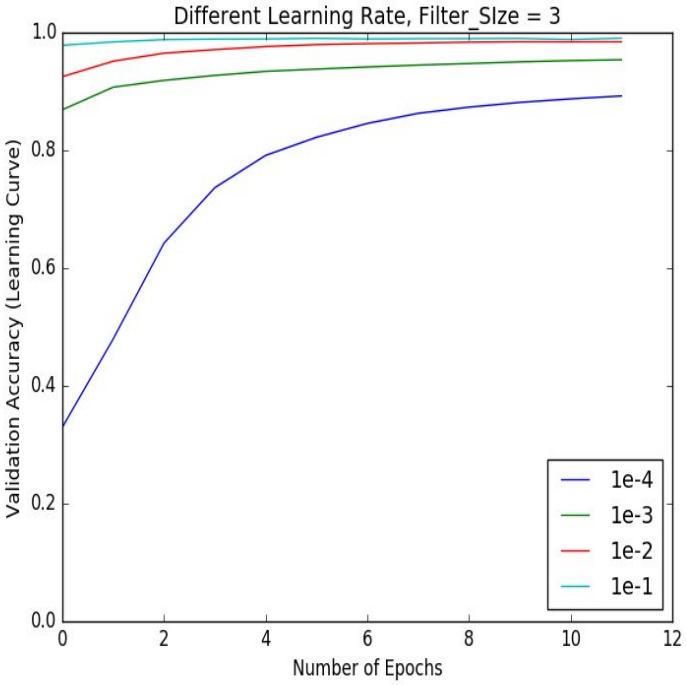


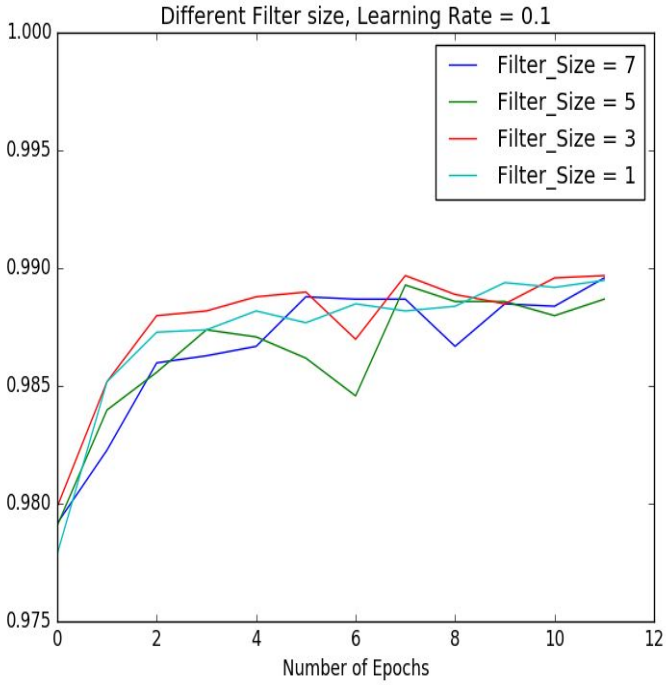
Figure 1: General architecture of LeNet

Implementation: In this exercise, we have used Mnist data which are images of handwritten digits from 0 to 9. We implemented our CNN network which contains 2 convolutional layers with 16 filters of size 3x3. Each layer is followed by max-pooling of 2 and stride of 1. To add non linearity a Relu activation function is used after max-pooling. After which a fully connected layer with 128 units is applied. It flattens our tensor to a vector. And finally another fully connected layer which will apply softmax loss and then use cross entropy to predict our final output. The final layer will have an output size of 26x26x16.

Validation Accuracy wrt different learning rates and filter size :

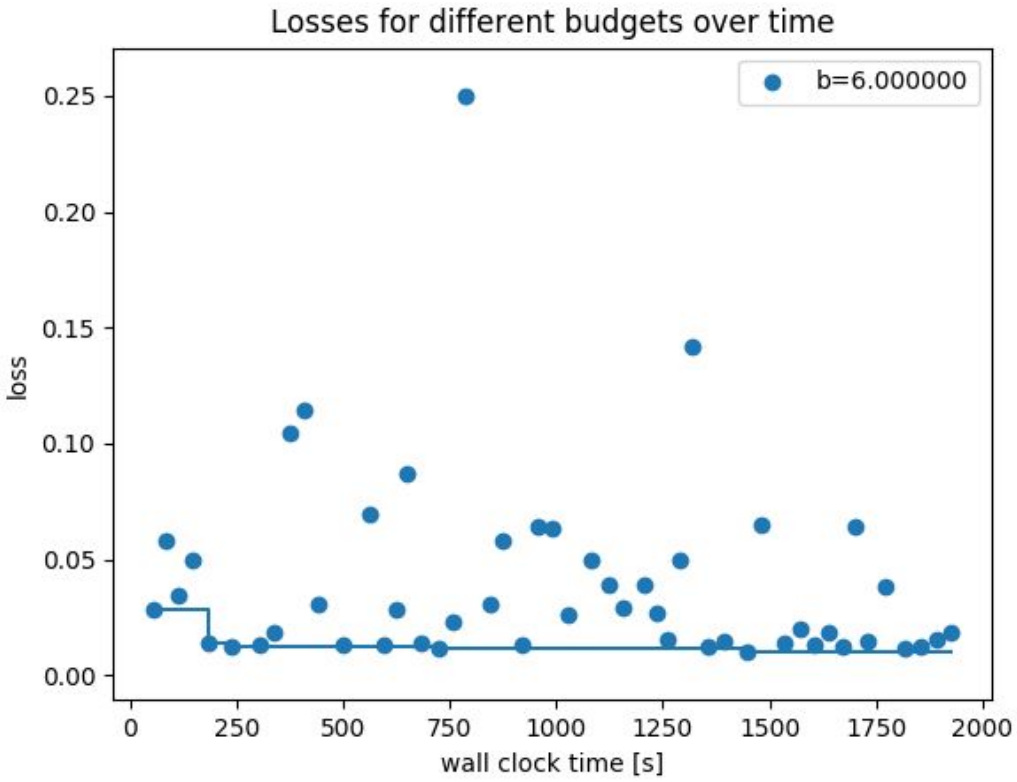
In order to get the best performance of our model, we have trained it with different hyperparameter(learning rates, filter sizes). With learning rate of 0.1, we get the maximum accuracy in the initial epochs, whereas with very small learning rate, we need more epochs to reach a higher accuracy.





Optimization of the hyperparameters: After training the model and noticing different performance of our hyperparameters (learning rate and filter size), we further optimize our learning rate, batch size, number of filters and filter size using random search method. The best configuration values after 6 epochs and 50 iterations after optimizing our model are:

- Learning rate : 0,0622664780088128
- Filter size : 3
- Number of filters : 36
- Batch size : 27



Problems Faced:

1. The matplot library was installed but it was still showing an error due to which the random_search optimization was run on friend's computer.
2. The tensorflow session was not getting restored properly due to placeholders only being defined in cnn_mnist file. So I had to define it again in my random search file.
3. The for loop was not running for different learning rates and filter sizes, due to which I had to run for each learning rate manually.