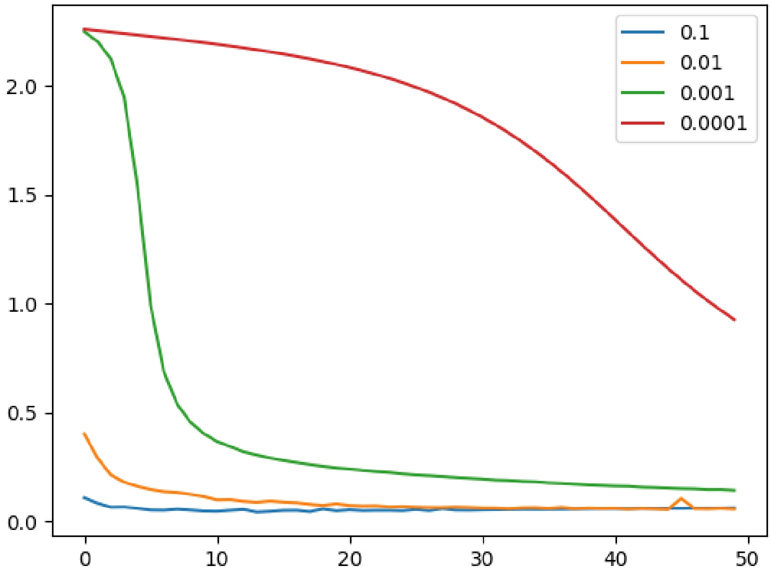
Exercise 2 Report **Ahmed Abdelhadi**
Deep Learning Lab Course
Lab Date: 30/10/2018
Due Date: 12/11/2018

Introduction

In this exercise, a ConvNet was developed using Tensorflow framework. The ConvNet aimed to solve multiclass image classification problem, namely the MNIST dataset. The architecture used was similar to the famous LeNet architecture where convolution layers were followed by ReLu activation layers then max pooling and at the end a full dense layer.

Different Learning Rates

Different learning rates were experimented and the performance was observed for each in terms of the learning curve (i.e validtion loss). the following graph shows the results

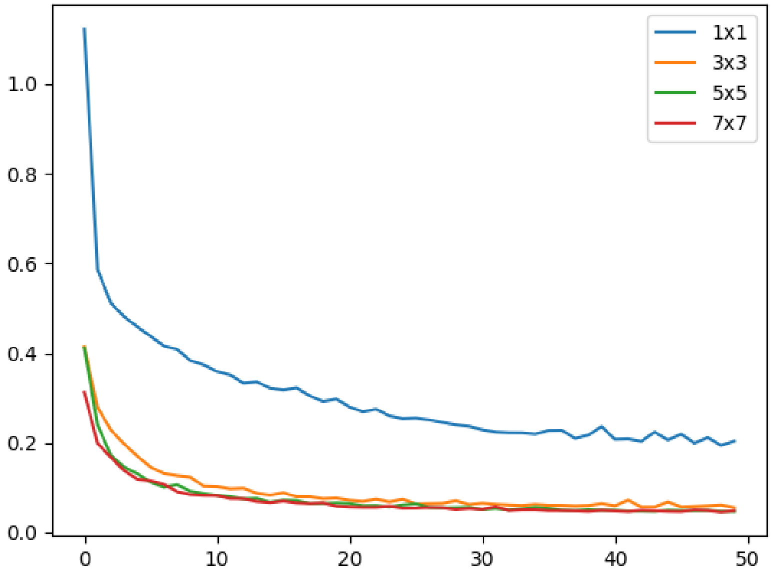


Do you have an idea what is happening if the learning rate is too small / high?
For each optimizer, there is a hill shaped performance for learning rates: too small a learning rate never progresses, too high a learning rate causes instability and never converges. In between there is a range of just right learning rates that yields the best performance.

Which conclusions could be drawn from this figure? Which value for the learning rate works best?
0.1 and 0.01 achieved a similar performance on this dataset and model, they started to diverge and the validation loss began to increase after nearly 10 epochs. While, 0.0001 never converged. The best was 0.001 which took around 10 epochs to converge and stabilize.

Different Filter Sizes

Different filter sizes for the convolution layers were experimented as well and the performance was observed for each in terms of the learning curve (i.e validation loss). the following graph shows the results

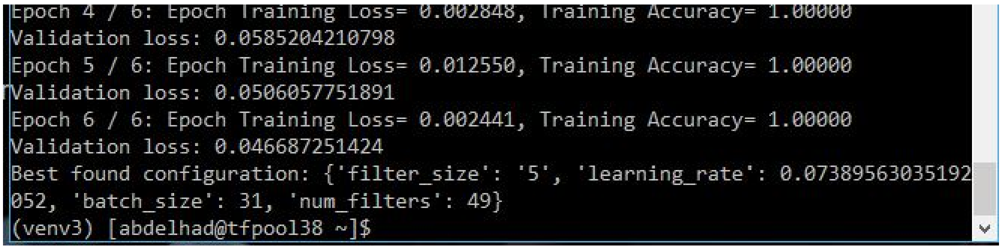


Do you have an idea in which scenarios smaller filters might be better than larger filters and vice versa?

Intuitively, at first layers, small filters are better because they are supposed to capture edges,lines and so on. But for deeper layers, bigger layers are more useful because they do object detection.

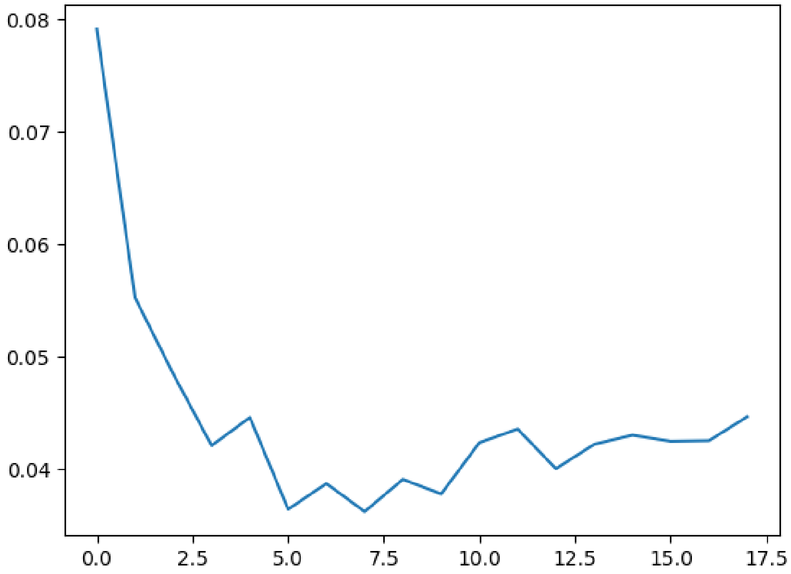
Random Search

After running Random Search for 50 iterations, that is the best configuration that I have got:



```
Epoch 4 / 6: Epoch Training Loss= 0.002848, Training Accuracy= 1.00000
Validation loss: 0.0585204210798
Epoch 5 / 6: Epoch Training Loss= 0.012550, Training Accuracy= 1.00000
Validation loss: 0.0506057751891
Epoch 6 / 6: Epoch Training Loss= 0.002441, Training Accuracy= 1.00000
Validation loss: 0.046687251424
Best found configuration: {'filter_size': '5', 'learning_rate': 0.07389563035192
052, 'batch_size': 31, 'num_filters': 49}
(venv3) [abdelhad@tfpool38 ~]$
```

The test accuracy of this configuration was 99.25%
The learning curve of this configuration is:


0.08
0.07
0.06
0.05
0.04
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5