DEEP LEARNING LAB COURSE

ASSIGNMENT 2 – IMPLEMENTATION OF A CNN USING TENSORFLOW

SUBMITTED BY: SANIEA AKHTAR

MATRICULATION NUMBER: 4156541

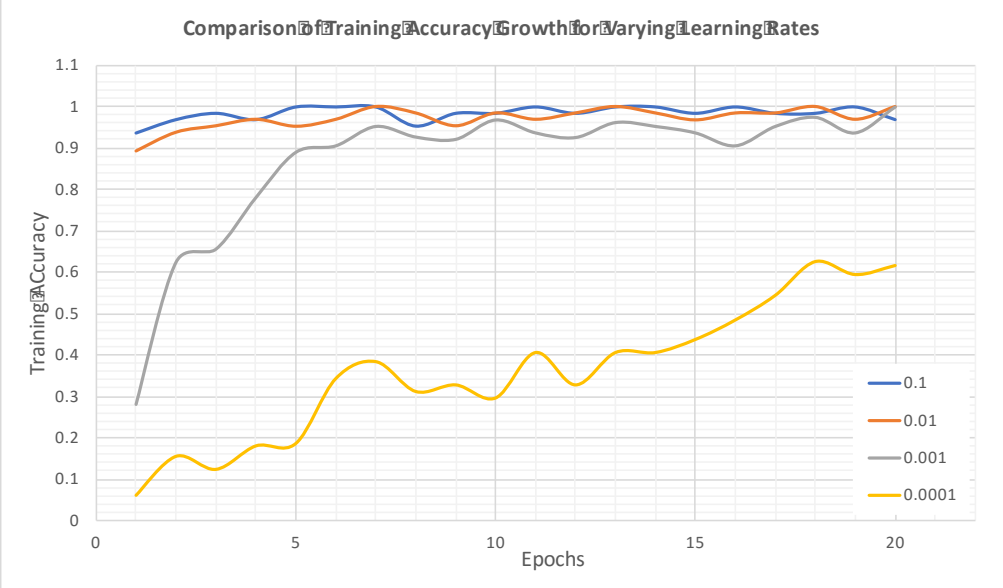
TASK

The aim of this task was to implement a convolutional neural network using the Tensorflow software library. The basic form of the system was provided in the tutorials on the Tensorflow website. The rest of the system was to be built using that basic framework and extracting useful information. This network also used the MNIST handwriting database which consists of a total of 60k training images and 10k testing images. The network should be able to compare the effect of different learning rates of the gradient descent algorithm and runtime variations based on the number of filters and computing power available.

DISCUSSION

Unfortunately, I severely underestimated the amount of time this assignment would require. I couldn't compute everything required in the script directly or elegantly. Not knowing if everything has been implemented correctly or not until the entire network has finished it's run, also made the task more tedious.

For **exercise 2**, we were required to vary the learning rate of the network while keeping all the other parameters constant. As expected, the training rate of the network slowed down immensely as a result of lowering the learning rate. A comparison is shown in the plot below:



In the case of a learning rate of 0.0001, the network was not able to finish training and reach an acceptable level of accuracy. Since the learning rate was so slow, ever step taken was extremely small. The execution of the number of epochs finished before a local minimum was reached.

As for a higher learning rate, the network achieves almost 100% accuracy from the very start. This is because, taking longer steps, the local minimum was reached possibly within the first iteration of training. I think a learning rate between 0.01 and 0.01 would be a good value since it would ensure that the minimum is reached at an appropriate speed, neither too fast nor so slow that the epochs are all executed without ever reached an optimum value.

Exercise 3 was a comparison of runtimes on a GPU and a CPU while changing some parameters of the CNN. The number of filters in the system were to be varied as {8, 16, 32, 64,128, 256} on the GPU and as {8, 16, 32, 64} on the CPU. The changes were observed as the parameters of the network were changed:

- As the number of filters was increased, the number of parameters that were calculated increased significantly.
- Keeping the same hardware specifications, increasing the number of filters increased the runtime almost two-fold, i.e. the increase in time was ~ proportional to the increase in the number of filters.
- Executing the same parameters on the CPU took significantly more time than on the GPU. Executing 256 filters on the CPU would either take a couple of hours or burn down the machine, probably the latter!