Handwritten Digit Classification

Programming Assignment No. 2
CSE574 Introduction to Machine Learning

Ashwini Aher, Pratik Patil

Department of Electrical Engineering
{aaher, ppatil2}@buffalo.edu
State University of New York at Buffalo

April, 18 2018

Contents

1 Hyper Parameters **1**
1.1 Hidden Units in Hidden Layer 1
1.1.1 Observations 1
1.2 Regularization Parameter (*Lambda*) 3
1.2.1 Observations 3
1.3 Accuracy 4
1.4 Execution Summery 4

2 Neural Networks Vs. Deep Neural Networks **6**
2.1 Comparison between NN and DNN 6
2.1.1 Observations 6

3 Convolutional Neural Networks(CNN) **8**
3.1 Observations 8

4 Reference for graphs **11**

Chapter 1

Hyper Parameters

1.1 Hidden Units in Hidden Layer

The accuracy of the *Neural Network* depends on the value of following two parameters:

Number of Hidden Units
Regularization Parameter (*Lambda*)

To get the highest accuracy we have to choose the optimal value of these both parameters. If values are two high or too low then there is a chance of *overfitting* and *underfitting* respectively. To find this optimal value we varied the value of Regularization Parameter (*Lambda*) from 0 to 60 with the interval of 5. On the other hand number of hidden nodes takes the values as 10, 15, 25, 50 and 75. We plot the graph in Figure 1.1 of Regularization Parameter (*Lambda*) Vs. Accuracy of the *Neural Network*. Each line in the graph represents the number of hidden layers. We can clearly observe in the graph that as the number of hidden units in the hidden layer increases, accuracy of detecting correct labels increases Figure 1.2. If the number of hidden layers are below 25 accuracy of the model is not so good but there is sudden significant improvement when the number of hidden units increased from 25 to 50.

1.1.1 Observations

We have observed following points from the graphs in figure 1.1 and 1.2.

1. The accuracy performance of the *Neural Network* varies significantly with varying number of units in hidden layer and Regularization Parameter (*Lambda*).

2. As the number of hidden units increases in the hidden layer the accuracy increases as well.

3. For the Regularization Parameter (*Lambda*), this is not the case. Accuracy of the *Neural Network* does not linearly increases with increasing value of Regularization Parameter (*Lambda*). Accuracy increases till certain

value of the *Lambda* and then accuracy decreases. This is the result of *overfitting*.

4. From the graph it is clear that 75 hidden units in hidden layers gives best accuracy hence we choose **75 hidden units** in hidden layer.

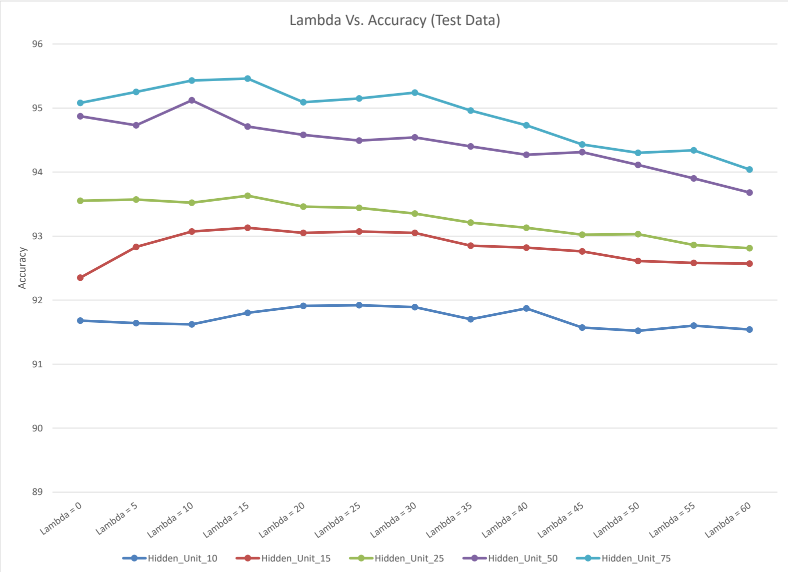

Figure 1.1: Lambda Vs Accuracy(Test Data)

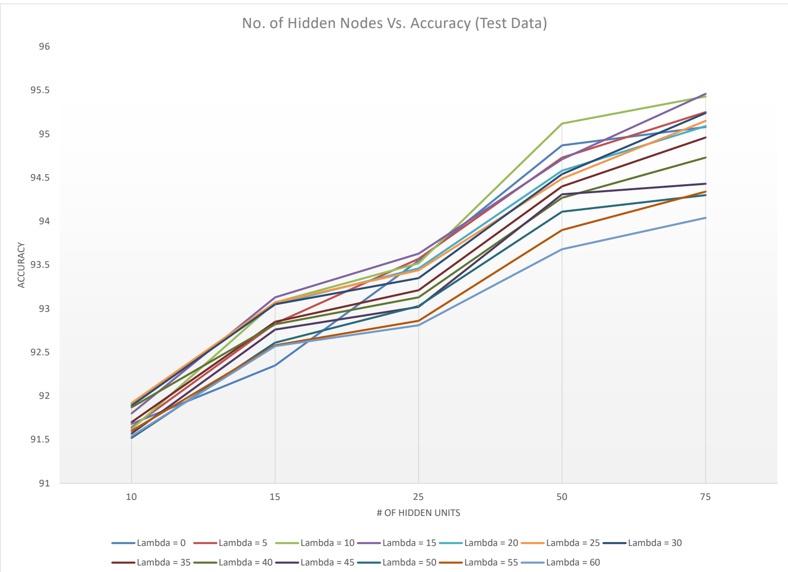

Figure 1.2: # of Hidden Units Vs Accuracy(Test Data)

1.2 Regularization Parameter (*Lambda*)

In the previous section we have fixed the number of hidden units to 75. Hence we are only focusing on the accuracy for different values of *Lambdas* for 75 hidden units in hidden layer. We plot the graph Lambda Vs. Accuracy in figure 1.3. In this graph hidden units are kept constant to 75 and *Lambda* varies from 0 to 60 with step size of 5.

1.2.1 Observations

We have observed following points from the graphs in figure 1.3.

1. From the graph we can conclude that with the increase in the value of lambda, accuracy increases upto certain value and then starts decreasing. This is the effect of *overfitting* where model tries to find weights which increases accuracy for training data but does not necessarily a good fit for test data.

2. We can see that maximum accuracy is achieved at lambda equal to 15 after which there is a constant decrease in the accuracy except for lambda values from 20 to 30.

3. From the above observations we select our Regularization Parameter (*Lambda*) equal to 15.

4. We have selected hyper parameters as follows to get the maximum accuracy

   (a) Number of Hidden Units = 75

   (b) Regularization Parameter (*Lambda*) = 15

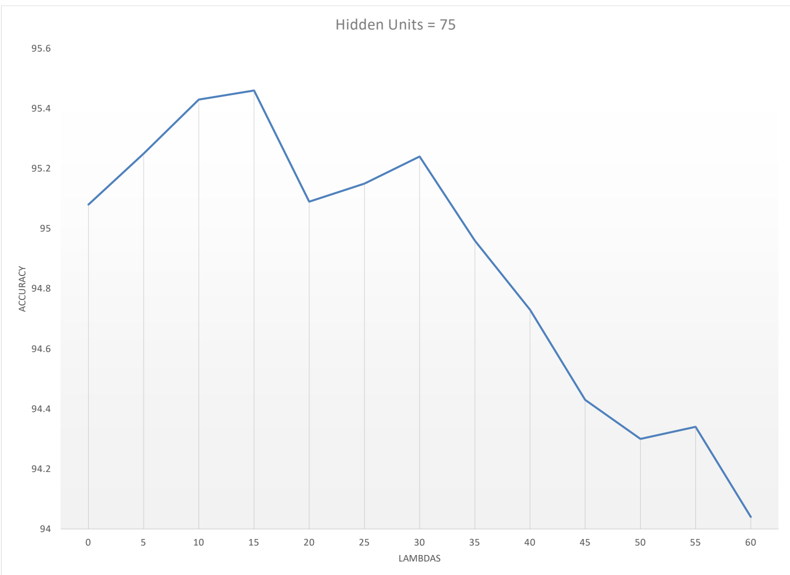

Figure 1.3: Lambda Vs Accuracy(Hidden Units = 75)

1.3 Accuracy

When we plot the graph in Figure 1.4 for value of Regularization Parameter ($Lambda$) = 15 we can clearly see that maximum accuracy is achieved when hidden units are 75 and for this combination we have achieved the accuracy of 95.5%.
As the number of hidden units in hidden layer increases the training time of algorithm increases as well because of the more complexity. But if we reduced the number of hidden units then the accuracy of Neural Network decreases. Hence there is a trade off between choosing number of hidden units and time required to train the weights.

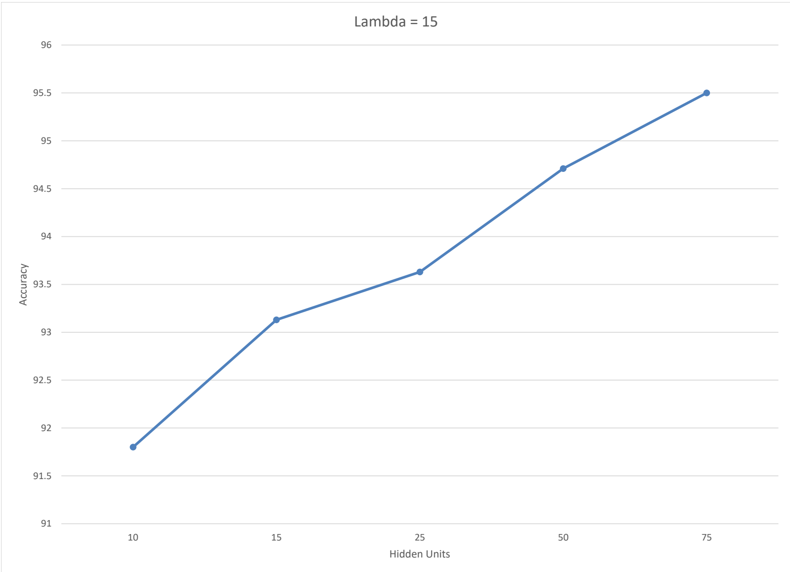

Figure 1.4: Hidden Units Vs Accuracy(Lambda = 15)

1.4 Execution Summery

We used personal laptop to run all the algorithms and did not use *mettalica* or *springteen* server. Configuration and version of software use to execute the *Neural Network* are as follows.

CPU : Intel Core i7-7700HQ CPU @ 2.80GHz
GPU : NVIDIA GeForce GTX 1050
RAM : 8 GB
VRAM : 4 GB
Python : v3.6.3
Tensorflow : v1.7.0
CUDA : v9.0.176

Following graph Figure 1.5 shows the execution time for varying number of hidden units in hidden layer.

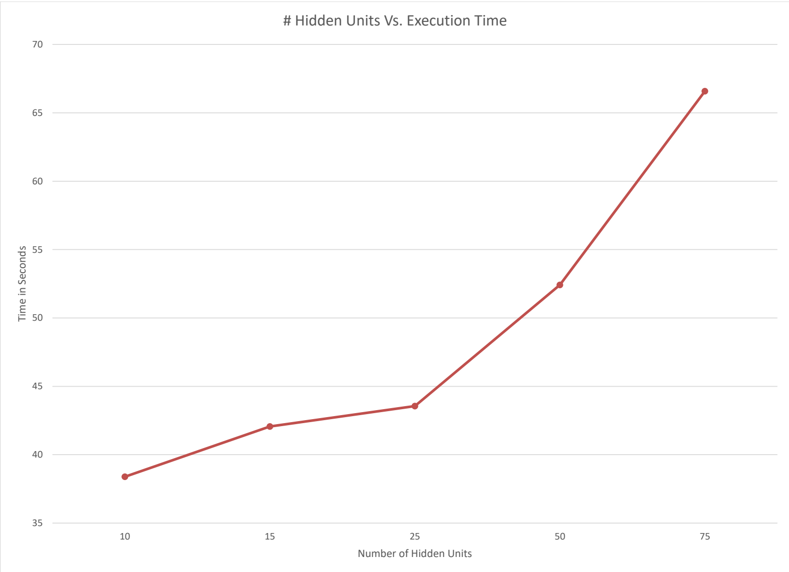

Figure 1.5: Hidden Units Vs Execution time

Chapter 2

Neural Networks Vs. Deep Neural Networks

2.1 Comparison between NN and DNN

We learned the hyper parameters in the previous section on the MNIST dataset. We will use the same hyper parameters i.e. Number of hidden layers equal to 75 and Regularization Parameter (*Lambda*) equal to 15 on the CelebA dataset to find the accuracy for single hidden layer in *Neural Network*. We will also compare the performance and accuracy of multiple hidden layers on the same dataset using tensorflow. Here we computed the accuracy and duration of execution for 1, 2, 3, 5, 7 hidden layers using tensorflow. We used the same configuration as that of mentioned in section 1.4 to execute the deep neural network.

2.1.1 Observations

From the following graph in Figure 2.1 we have observe the following things:

1. As the number of hidden layers increases, the execution time of the algorithm also increase.

2. Even if the number of hidden layers are increased, it does not guarantee that accuracy of prediction will increase. Instead in case of CelebA dataset its inverse, i.e. accuracy decreases with increasing number of hidden layers.

3. As stated by Occam's Razor that "more complicated neural network does not necessarily mean that better accuracy", this can be observed in the graph Figure 2.1

4. Four point stars represent the single layer neural network implemented by us. Green star represents the accuracy of the neural network while red star represents the execution time taken by neural network.

5. Blue and Orange line represents the accuracy variation and execution time variation respectively for varying hidden layers.

6. We can observed that performance of single layer neural network is slightly better than the deep neural networks. Also the execution time of the single

layer neural network implemented by us is significantly lower than that of the single layer neural network implemented using tensorflow.

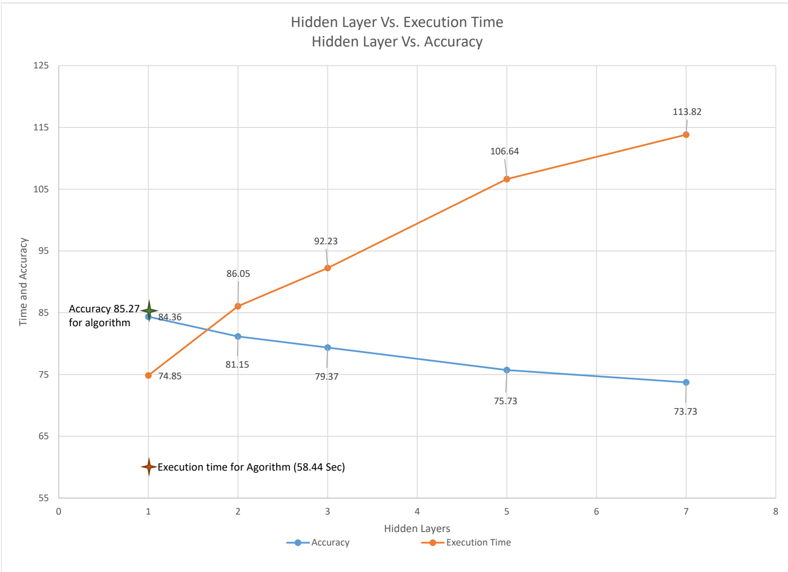

Figure 2.1: Hidden Layers Vs Execution time, Hidden Layers Vs Accuracy

Chapter 3

Convolutional Neural Networks(CNN)

In this script CNN is implemented and MNIST dataset is used to identify the digits written in the image. In this algorithm two 16 and 32 convolutional layers are used with size 25*25 each. Images are converted to gray scale so instead of three layers of RGB, only one layer of image is present . Necessary changes have been made to print out the accuracy, confusion matrix and error examples. Updated script is kept in folder with this submission. We plot the graph (as shown in Figure 3.1) of the number of iterations Vs. Accuracy of prediction.

3.1 Observations

From the following graph in Figure 3.1 we have observe the following things:

1. We can clearly observe from the graph that accuracy of the digit recognition increases with the increasing number of iterations.

2. The confusion matrix represents the Predicted value Vs. True label of the image. By observing it, we can say that, initially most of the images were labeled with same digit. Thus, one or two columns are filled with values and other matrix elements are zero. But confusion matrix improves as the number of iterations increases. As a result of which more and more values moved towards diagonal as shown in matrix below.

3. We also observed that accuracy improves rapidly from 1 to 99 iterations. But after that weights are improved slowly with iterations and after 9000 iterations, accuracy is almost 99%.

4. We have observed one more most important thing that the time required for 9900 iterations was almost close to 48 seconds which is much lower than what we required for deep neural network and the results were more accurate too as mentioned above. Graph in Figure 3.2

Confusion matrix after first iteration can be written as follows:

$$\text{Confusion Matrix} = \begin{array}{cc} & PredictedLabels \\ \rotatebox{90}{$TrueLabels$} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 253 & 0 & 0 & 0 & 727 \\ 0 & 0 & 0 & 0 & 0 & 396 & 0 & 0 & 0 & 739 \\ 0 & 0 & 0 & 0 & 0 & 228 & 0 & 0 & 0 & 804 \\ 0 & 0 & 0 & 0 & 0 & 214 & 0 & 0 & 0 & 796 \\ 0 & 0 & 0 & 0 & 0 & 475 & 0 & 0 & 0 & 507 \\ 0 & 0 & 0 & 0 & 0 & 283 & 0 & 0 & 0 & 609 \\ 0 & 0 & 0 & 0 & 0 & 117 & 0 & 0 & 0 & 841 \\ 0 & 0 & 0 & 0 & 0 & 111 & 0 & 0 & 0 & 917 \\ 0 & 0 & 0 & 0 & 0 & 200 & 0 & 0 & 0 & 774 \\ 0 & 0 & 0 & 0 & 0 & 273 & 0 & 0 & 0 & 736 \end{bmatrix} \end{array}$$

Confusion matrix after 9901 iteration is as follows:

$$\text{Confusion Matrix} = \begin{array}{cc} & PredictedLabels \\ \rotatebox{90}{$TrueLabels$} & \begin{bmatrix} 972 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 3 & 1 \\ 0 & 1127 & 3 & 0 & 0 & 1 & 2 & 0 & 2 & 0 \\ 1 & 4 & 1017 & 0 & 1 & 0 & 0 & 4 & 5 & 0 \\ 0 & 0 & 1 & 1002 & 0 & 2 & 0 & 2 & 2 & 1 \\ 0 & 0 & 2 & 0 & 978 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 5 & 0 & 878 & 1 & 1 & 1 & 3 \\ 4 & 2 & 0 & 1 & 1 & 4 & 945 & 0 & 1 & 0 \\ 0 & 4 & 5 & 2 & 0 & 0 & 0 & 1013 & 1 & 3 \\ 2 & 0 & 3 & 3 & 3 & 0 & 1 & 2 & 958 & 2 \\ 0 & 3 & 0 & 1 & 7 & 2 & 0 & 2 & 2 & 992 \end{bmatrix} \end{array}$$

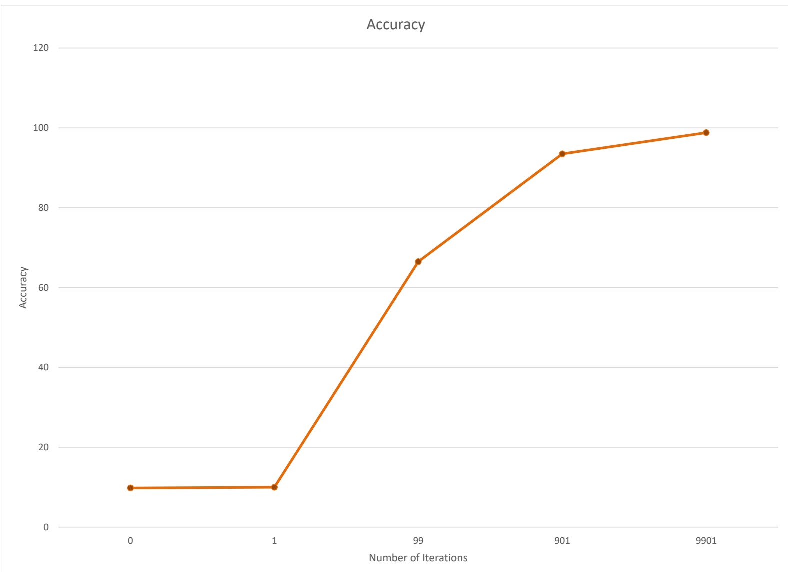

Figure 3.1: Number of Iterations Vs. Accuracy

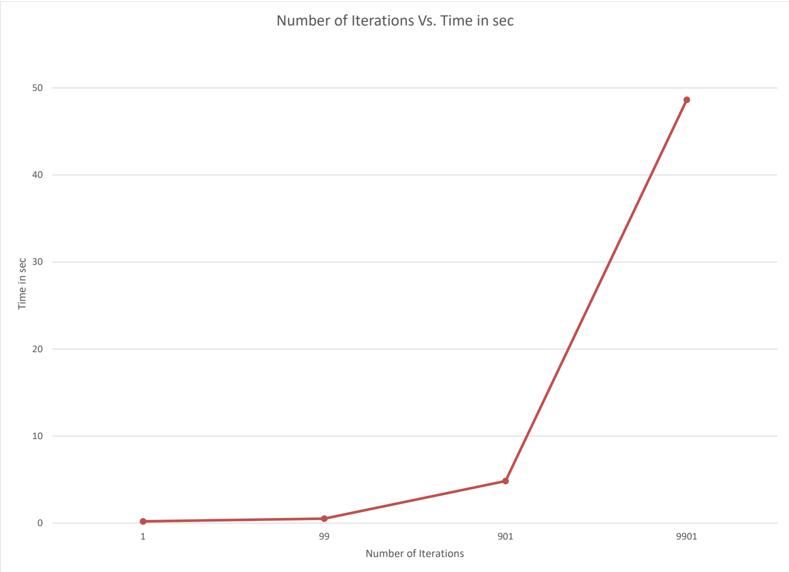

Figure 3.2: Number of Iterations Vs. Time Required for execution

Chapter 4

Reference for graphs

Following are the links to access the excel files, using which we have generated all the graphs in this report.

Click here to access all the graphs related to MNIST data set.
Click here to access all the graphs related to CelebA data set.