CSE4/574 Programming Assignment 2- Handwritten Digits Classification (Team 50)

Abhishek Kumar
50419133
akumar58@buffalo.edu

Utkarsh Kumar
50419745
utkarshk@buffalo.edu

Abstract

In this report we examine different application of neural networks/deep learning for supervised tasks. In particular we attempt at obtaining the optimal networks classification tasks. The project has been implemented using Tensorflow library on a GCP machine instance (for nnScript.py, facennScript.py), Metallica server for deepnnScript.py & Timberlake server (for cnnScript.py)

1 Introduction

In the first phase of the project we fine tune a multilayer perceptron network for a high accuracy detection of handwritten digit data set: MNIST. Through multiple iterations we obtain the optimal hyper-parameter values for the task. Using these values we apply the same network on celebrity face classification (spectacles vs no spectacles) using CelebA data set, and compare the results. The comparison is then taken forward by switching to deep networks and then to convolutional networks and observing the change in training accuracy.

2 Neural network on MNIST:

The MNIST data set has 60,000 train data points and 10,000 for test. We further split the test data set to create a 10,000 point validation data set. For the experimentation we vary the number of hidden nodes (n) and the regularization coefficient (λ) to fine tune our model to classify the numbers. The range for variations are:

Hidden nodes (n): 4,8,12,16,20,30,40,50

Regularization coeff. (λ): 0,10,20,30,40,50,60

We found the optimum n,λ pair for our data set to be [50,0] reporting the following accuracy: [Train, Validation, Test] = [95.38%, 95.12%, 95.04%]
Figures 1,2 &3 show the the errors vs the parameters and the runtime vs number of hidden layers.
In Fig. 1, for the optimum λ value, the accuracy increases suddenly from 4 to 8 hidden nodes. After this there is a decrease in the rate at which the accuracy improves. While the progress is slower, we achieve the highest accuracy when n=50.
Fig. 2 shows the variation of accuracy with respect to λ when the hidden nodes are kept constant. The highest possible accuracy has been consistently shown for λ=0.
In Fig. 3, we see an increase in training time with increase in complexity, i.e., increase in number of nodes. The time however stabilizes as we cross 30 hidden nodes.

With these we draw the conclusion that for MNIST, the optimum hyperparameter values are [n=50,λ=0]

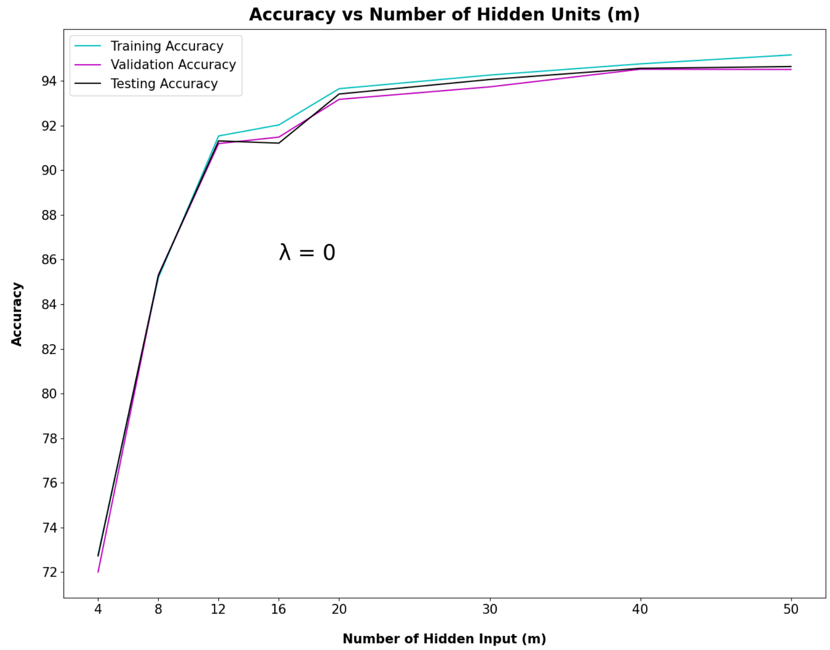

Figure 1: Accuracy vs n-hidden

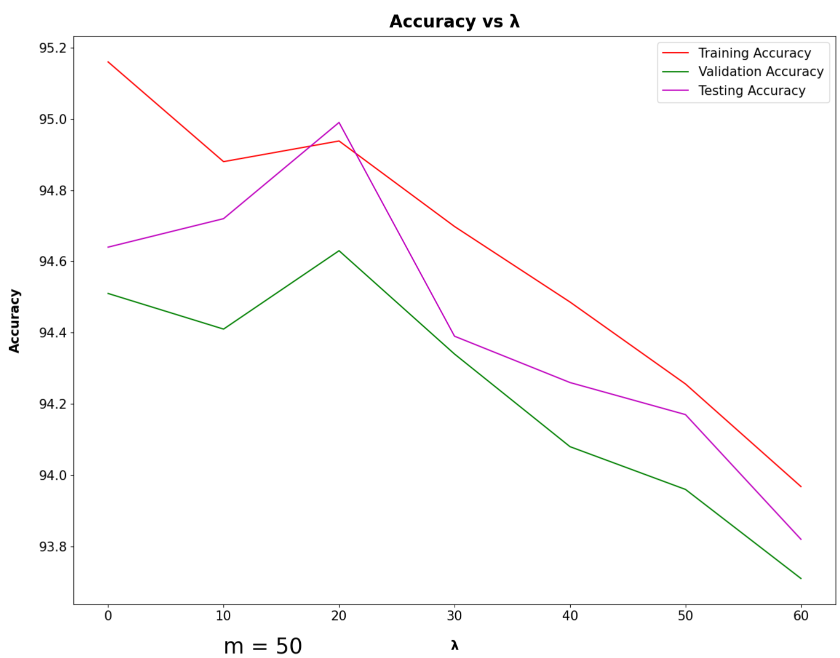

Figure 2: Accuracy vs λ

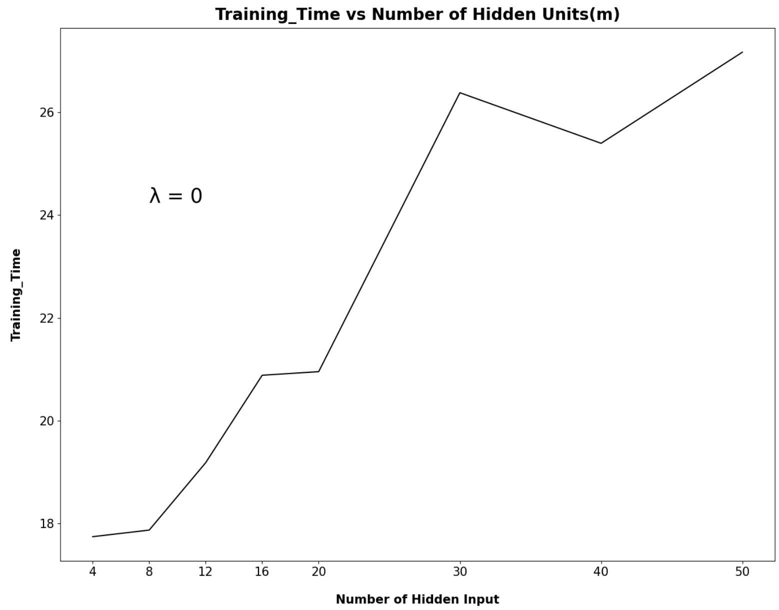

Figure 3: Time vs n

3 Neural network on CelebA:

We have determined the optimum parameters in the previous step and using the same values, we train the neural network on celebrity data set (facennScript.py).

We observe the following accuracy:

λ	Hidden nodes	Run time (s)	Train	Validation	Test
0	50	71	85.65%	83.75%	85.61%

4 Neural Network vs DeepNN:

In this setup we continue with the CelebA data set, solving the problem using two setups- Neural Network from the previous setup and Deep Neural Network (deepnnScript.py) using the tensorflow library and learning rate of 0.0003. We compare against accuracy and runtime by varying the complexity of the DNN by increasing the number of hidden layers: [2,3,5,7]. The following is our observation:

No. of hidden layers (256 nodes / layer)	Script	Test Accuracy (%)	Training time (sec)
1	facennScript.py	85.616	202.34
2	deepnnScript.py	80.204	89.45
3	deepnnScript3L.py	82.134	120.34
5	deepnnScript5L.py	81.491	136.79
7	deepnnScript7L.py	82.286	163.42

The accuracy of the deep neural network model is lower to that of neural network model. This is primarily due to the fact that the more complex model overfits the problems therefore under-performing on test data set. At the same time, a more complex model takes a higher time to train, hence there is a proportional increase in the time taken vs the number of hidden layers.

5 Application of CNN:

Finally, we apply a CNN network on CelebA classification:

Training time (Timberlake)	Test Accuracy (%)
43m 49s	98.6 (9863/10000)

The CNN model has a substantial improvement over a neural network or a deep neural network for the following reasons:

1. The CNN retains spatial information. By using filter banks and convoluting on image blocks, CNN retains the spatial information of a pixel which is a defining feature of an image data
2. The CNN adjusts itself locally according to the image features present in the region

However, training such a model comes at a computational expense as there are a lot of calculations in a convolution