Deep Learning Homework 1- Report

Prashanth Reddy Kadire
Dr. Feng Luo
CPSC 8430-Deep Learning
21 February 2022

Part 1: Deep vs Shallow

Task 1: Simulate a Function

#Bonus Attempted - Used more than 2 models
#Bonus attempted-Used more than 1 function

-Three separate models, each with different number of parameters were trained on two separate functions. The models named as model0, model1, model2 with 571, 572, and 572 parameters are fully connected neural networks with seven, four, and one layers respectively. The number of nodes of each layer are varied. The functions that were simulated were

i)y=(sin(5*(pi*x)))/((5*(pi*x))
ii) y=sign(sin(5*pi*X1))

all learning rates were 0.001.

- -All models for both functions had convergence in loss functions within a reasonable number of training iterations. In Figure 1 below, we can see the Mean Squared Error (MSE) of three models (deep, middle, shallow) for each epoch during the training for **(sin(5*(pi*x)))/((5*(pi*x))** simulation. Convergence is achieved after 2000 iterations. In Figure 2 displays the predicted and actual values for the **(sin(5*(pi*x)))/((5*(pi*x))** function.

-

prashanth 2

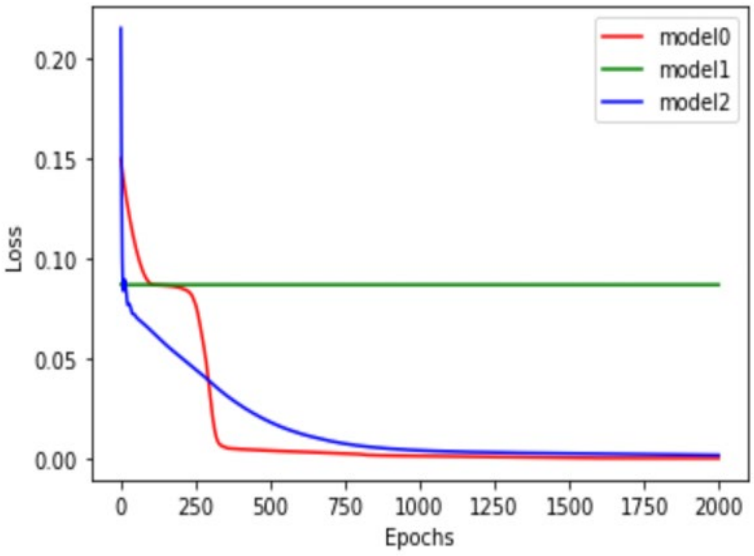

Figure 1

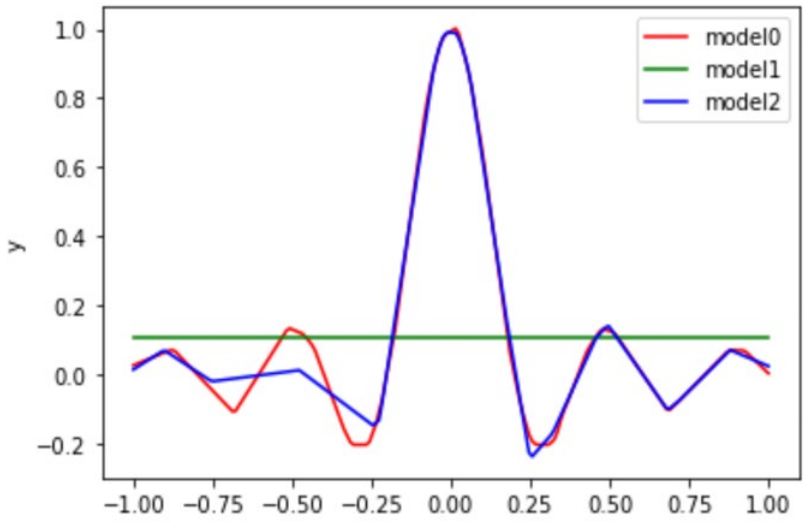

Figure 2

Figure 3 displays the Mean Squared Error (MSE) loss occurred during the training of three models (deep, middle, shallow) for **sign(sin(5*pi*X1))**

simulation. Figure 4 displays the predicted and actual values for the **sign(sin(5*pi*X1)) function.** All three models learned the function accuracy could provide correct output when the unseen input is given. In the center of the graph models were performed well, extreme ends of x-axis have misclassifications. The deeper the model lesser the failure at the edges of the graph.

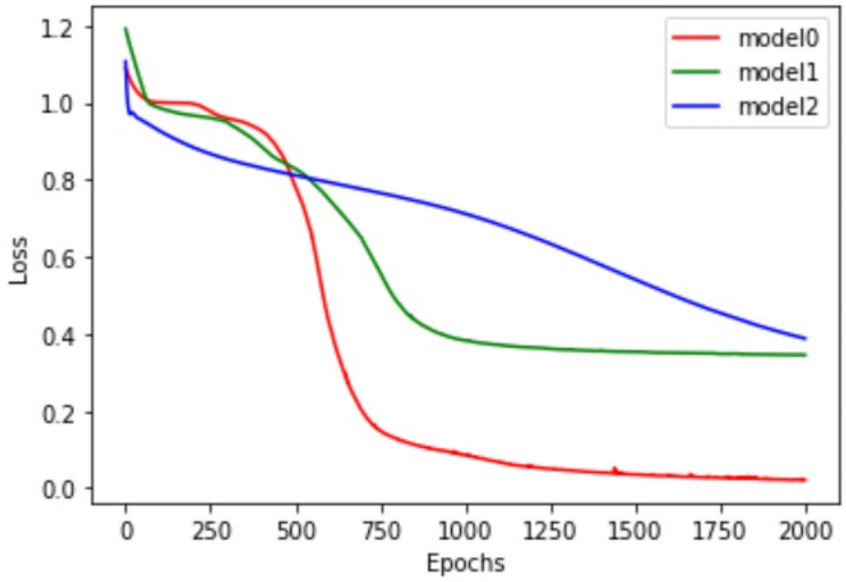

Figure 3

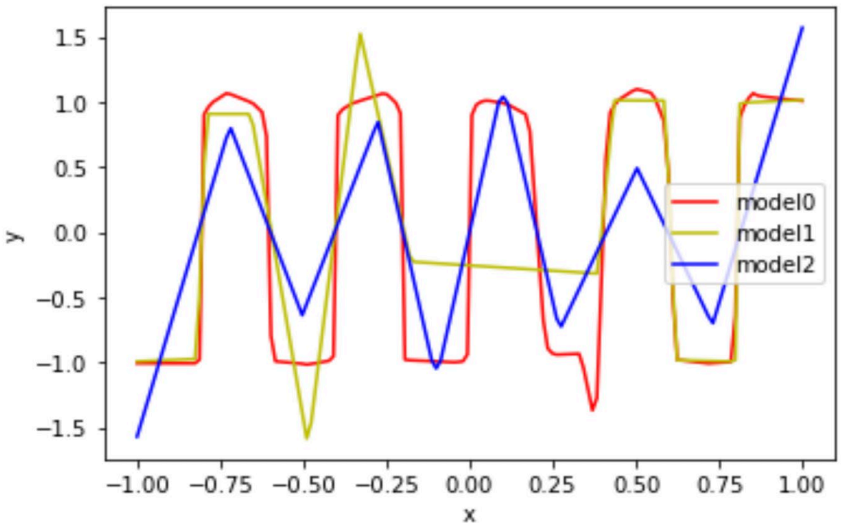

Figure 4

Task 1-2: Train on Actual Tasks

- Created two models using Convolutional Neural Networks (CNN) with MNIST dataset 60 thousand training data and 10 thousand testing data with batch size of 10. Training data is shuffled whereas testing data is not shuffled. All the two models are fully connected for pooling we have used max_pool, we have used RELU as an activation function. The number of nodes in each layer of each network is varied. All learning rates were 0.001. The figure 5 displays the loss occurred during the training of model1 and model2. Convergence is generally achieved around 50 epochs for two models.

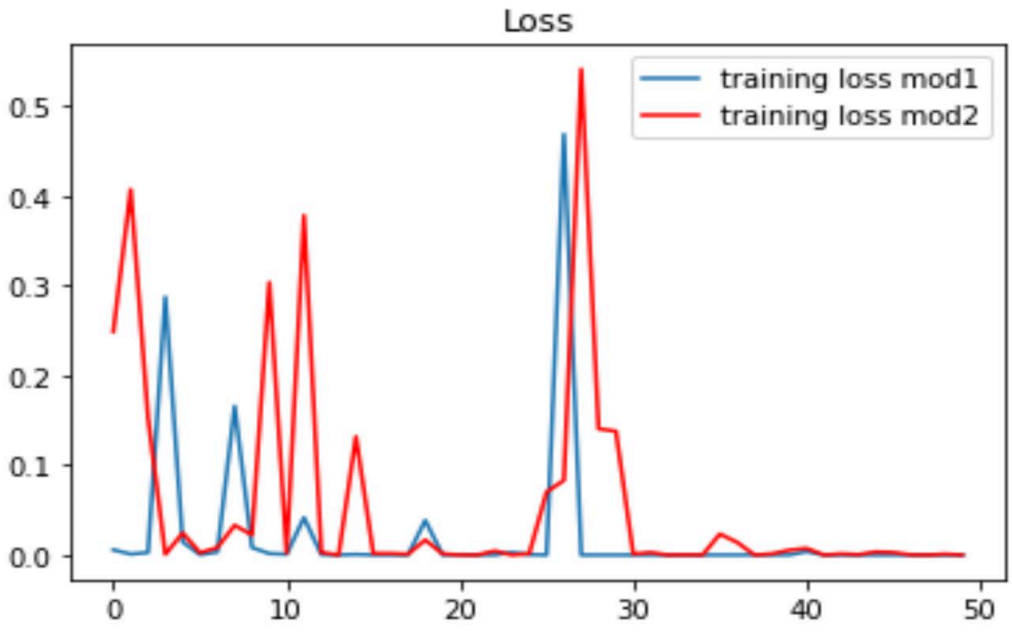

Figure 5

- Figure 6 displays the accuracy of the two models on training and test sets during the training of the models. We can observe that two models are consistently perform better on the training data than the test data, as to be expected.

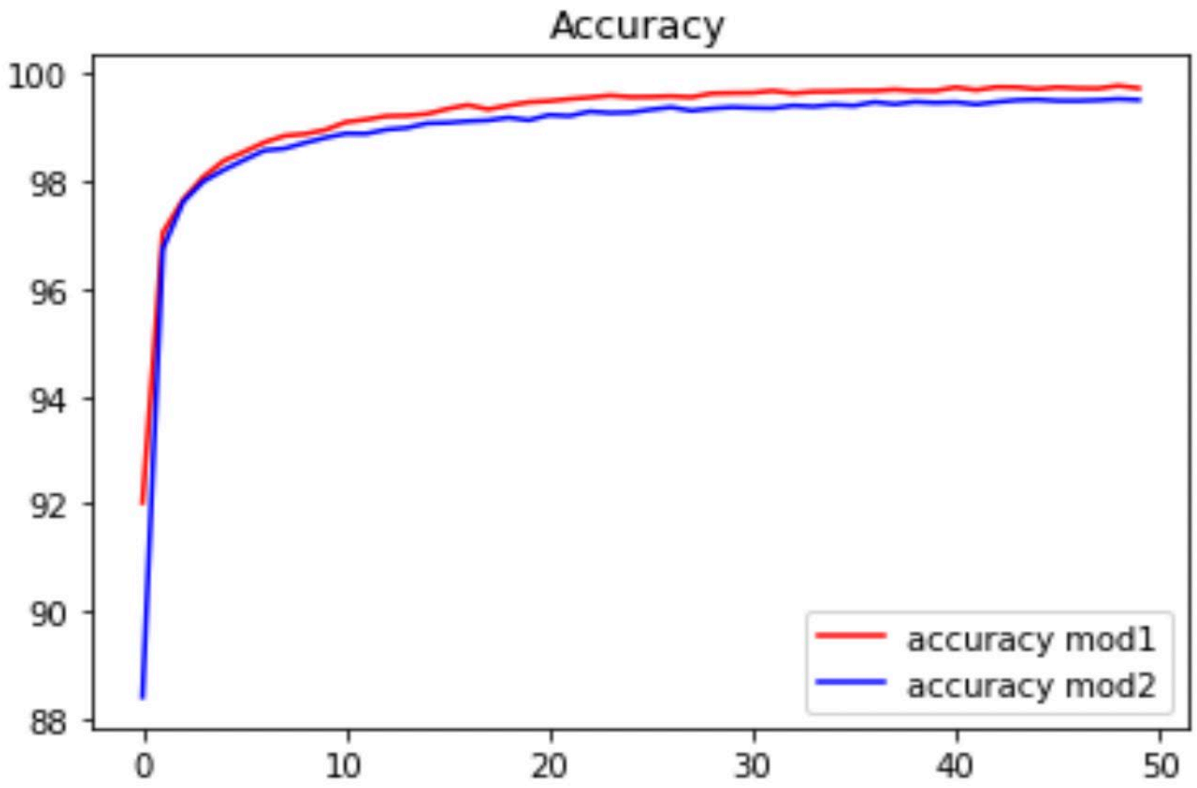

Figure 6

Part 2: Optimization

Task 1: Visualize the Optimization Process

A Deep Neural Network consists of three fully connected layers with 57 parameters was trained to simulate the function **(sin(5*(pi*x)))/((5*(pi*x))** . Figure 7 shows the second layer of the network and Figure 8 shows the whole model optimization. The Adam optimizer was used for the optimization of the network. Eight different training series, each of 30 training epochs was carried out, during which weights of the model were periodically collected. The dimension reduction is achieved by PCA implementation. The learning rate of all models is 0.001.

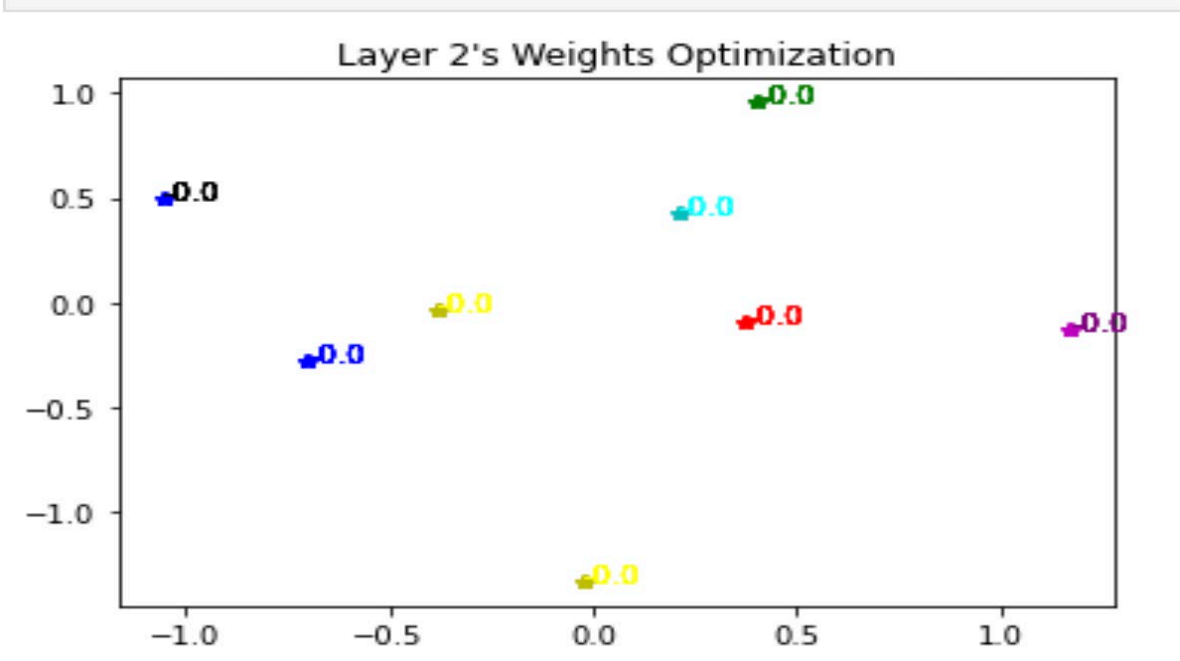

Figure 7

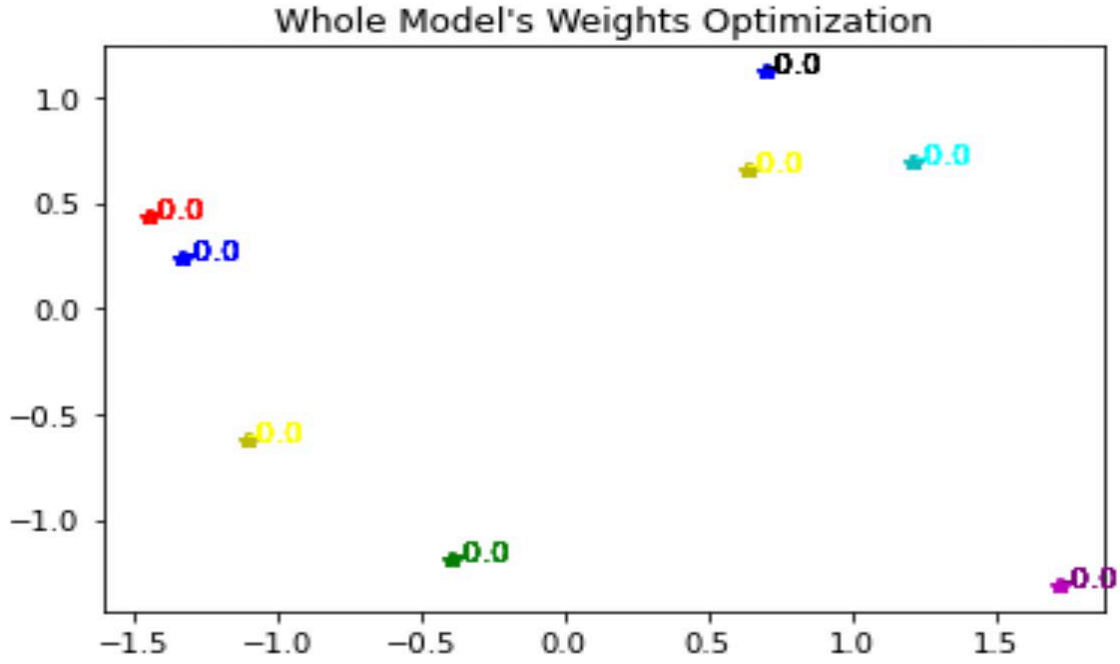

Figure 8

- Figure 7 and 8 both shows the optimization process of the weights within the network learns during the training process, its weights are optimized by backpropagation. The figures show the fine tuning over time as the weights are slowly optimized.

Task 2: Observe Gradient Norm during Training

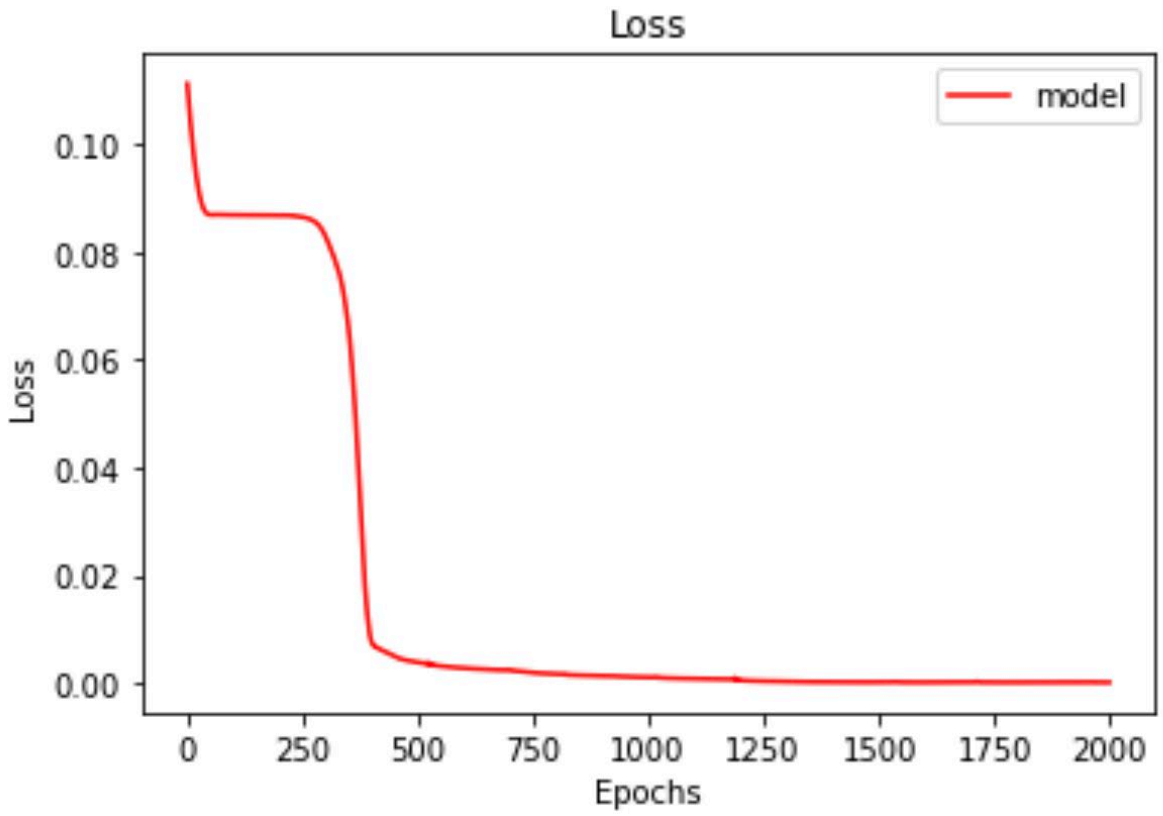

Figure 9

- Figure 9 shows the model's loss occurred for each epoch of training. While figure 10 shows the gradient norm for each epoch. Each of the spikes occurred in figure 10 corresponds to the slope changes in figure 9. There is abnormality in the graph 10.

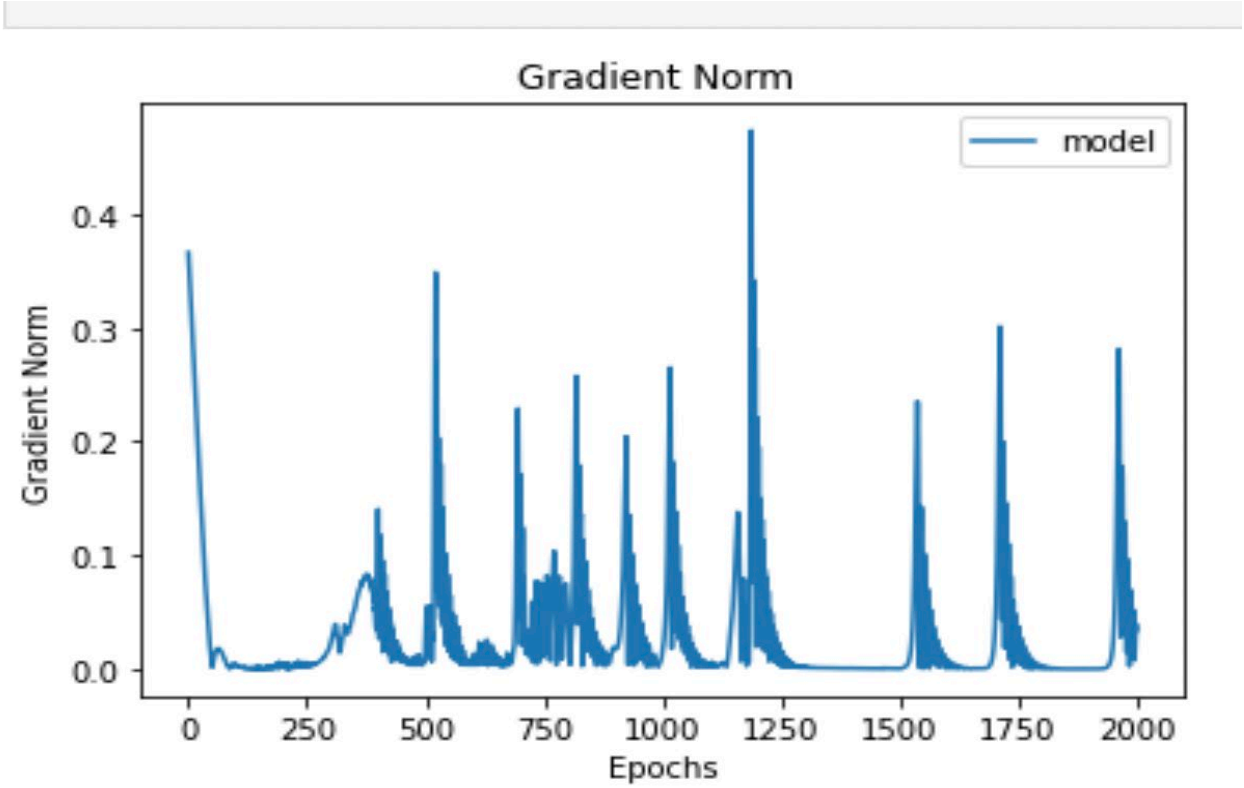

Figure 10

Part 3: Generalization

Task 1: Can Network fit random Variables

-The MNIST dataset was chosen as the training and testing dataset. A feed forward Deep Neural Network was implemented with 3 hidden layers. It was trained for 2000 times on the MNIST dataset. All

models have learning rate as 0.001. The Adam optimizer was used for the optimization process of the neural network.

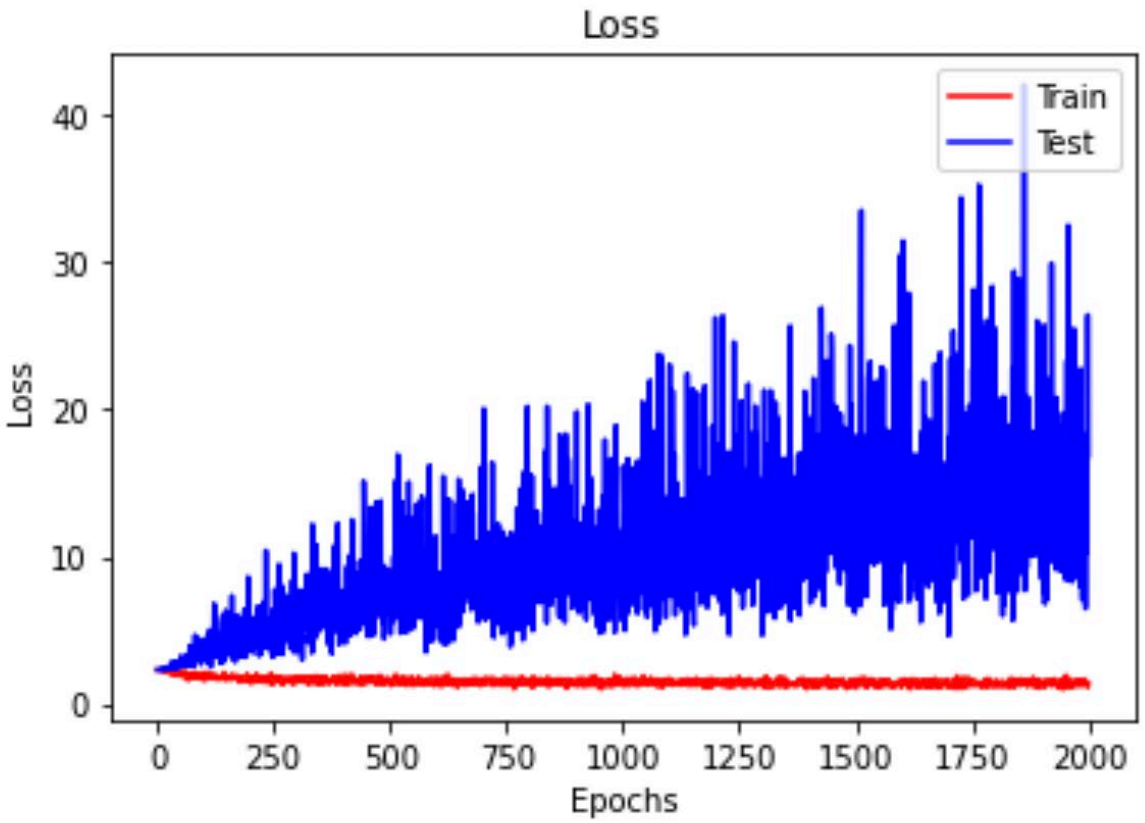

Figure 11

-If we observe in the figure 11, during the training loss is very less but test data which is unseen earlier has high loss comparatively. Therefore, we cannot fit the random labels.

Task 2: Parameters Vs Generalization

-The MNIST dataset was chosen as the training and testing data set. 10 feedforward Deep Neural Networks were implemented with two hidden layers. The number of parameters in the models are varied from hundreds to millions. All the model's learning rate was set to 0.001. The Adam Optimizer was used for the optimization of the Neural Network.

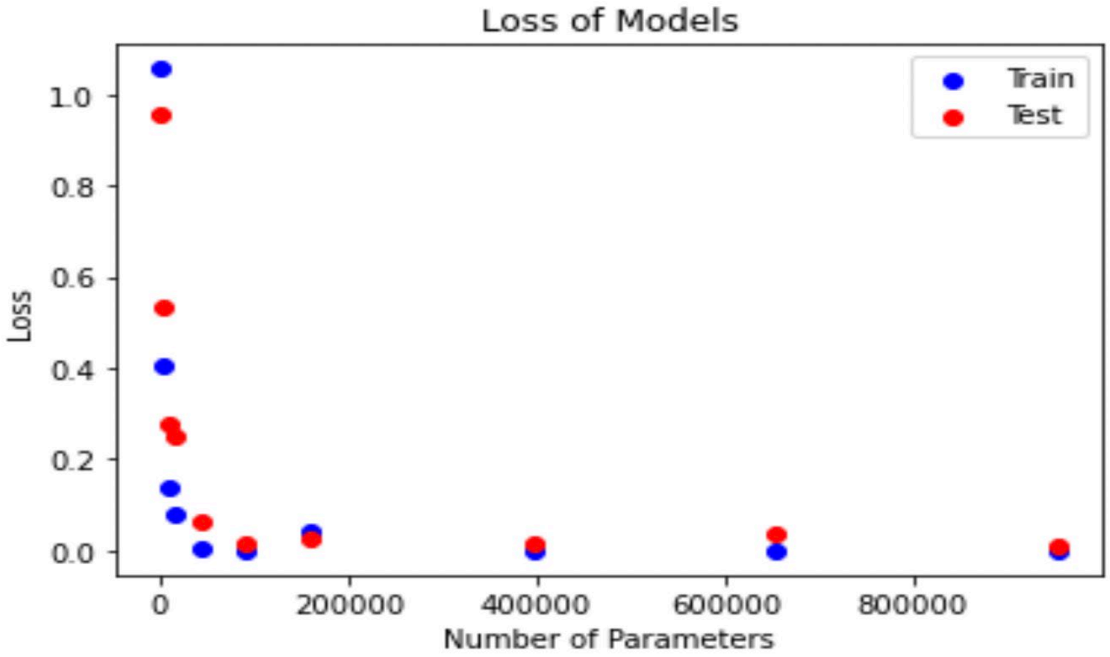

Figure 12

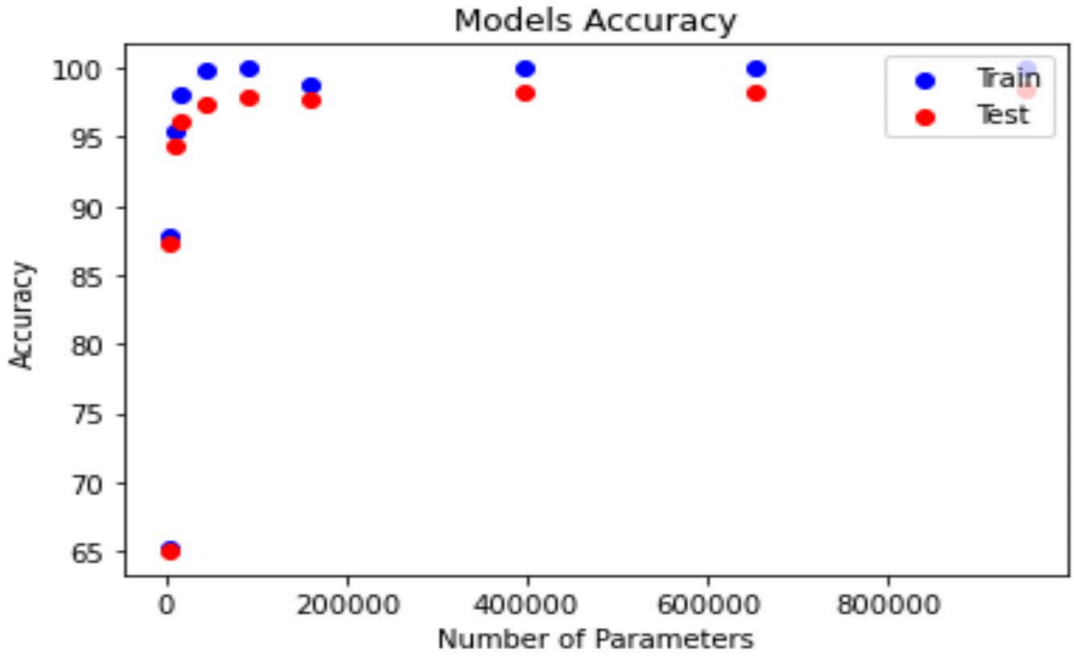

Figure 13

-if we observe Figure 12 and 13, increasing the number of parameters in the model decreases its loss and also increase its accuracy. However, after particular number of parameters in a model, the model improvement from iteration to iteration is negligible adding additional parameters improves the model very barely.

-If we observe fig 13 higher accuracies and lower loss values are obtained for the models when run on training dataset compared to the testing dataset.

Task 3: Flatness VS Generalization

-Part 1: -The MNIST dataset was chosen as the training and testing data set. Two Deep Neural Networks were implemented with two different batch sizes 64 and 1024. All the model's learning rate was set to 0.001. The Adam Optimizer was used for the optimization of the Neural Network. Alpha is the linear interpolation between theta1 and theta 2, theta1 is the model 1 parameters and theta 2 is the model2 parameters.

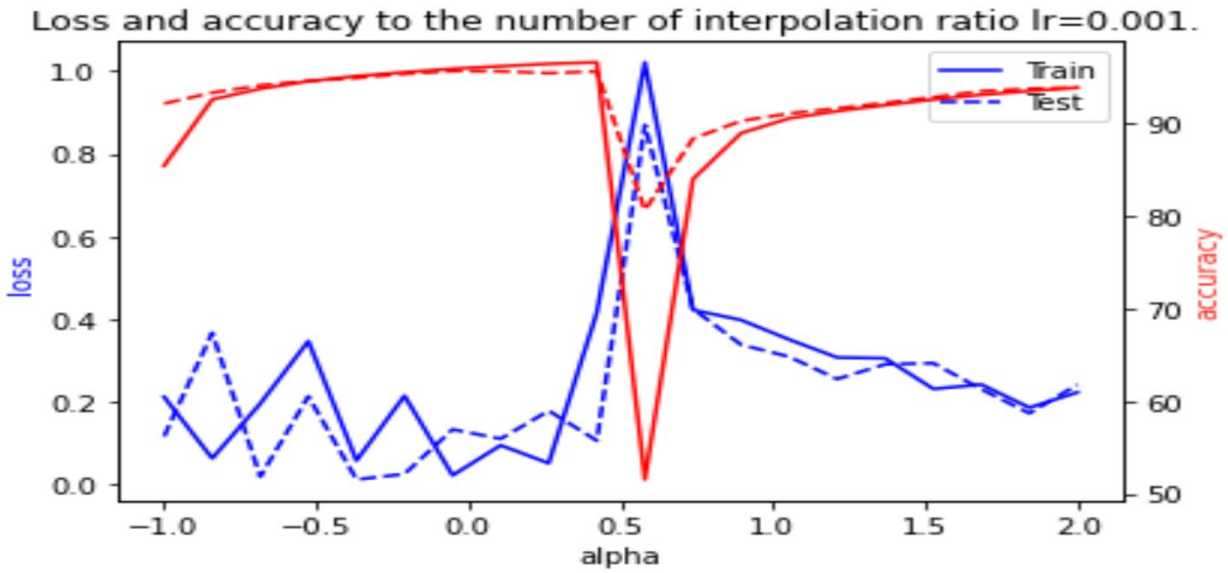

Figure 14

-Figure 14 shows the loss, accuracy, and linear interpolation alpha during the training of two models with learning rate 0.001, whereas Figure 15 with learning rate as 0.01.

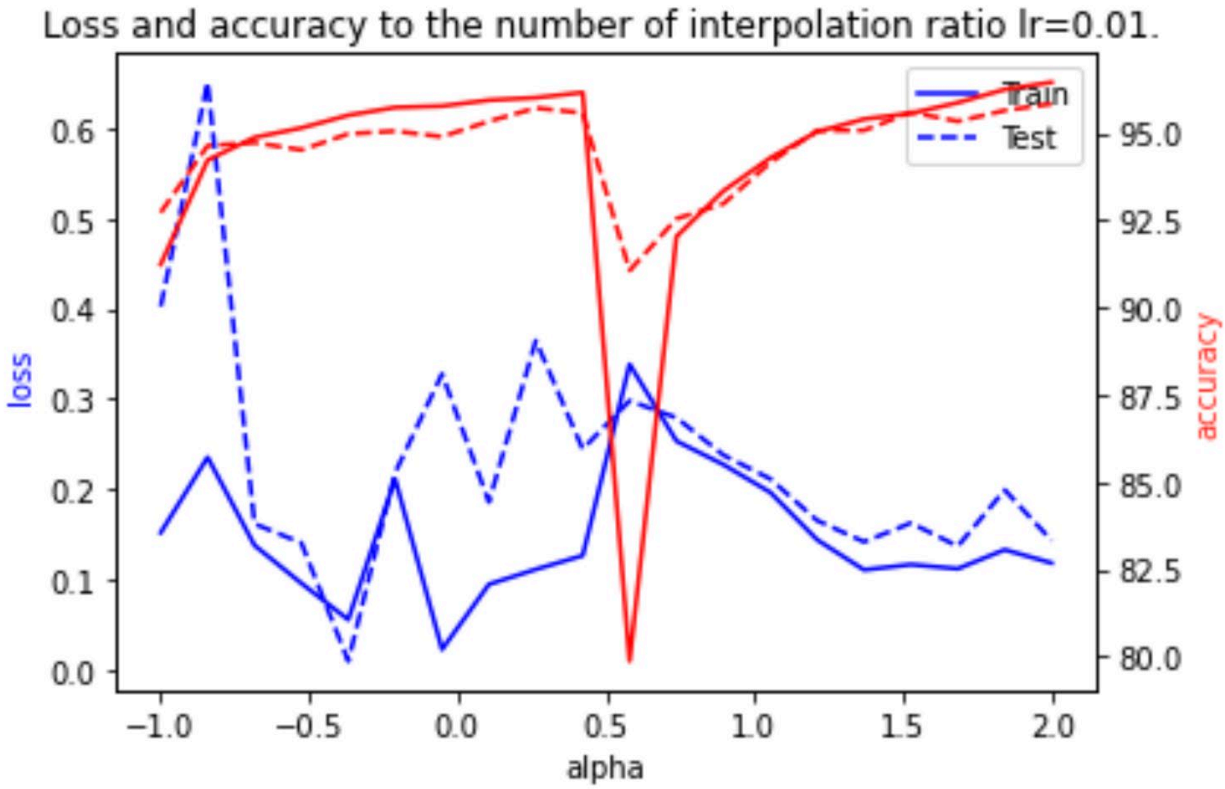

Figure 15

Part 2: Flatness Vs Generalization

-The MNIST data set was used to train and test the module. Five identical Deep Neural Networks, consists of two hidden layers with 16630 parameters, were training different batches

from 5 to 1000. The Adam Optimizer is used for optimization process and all the models have learning rate has 0.001.

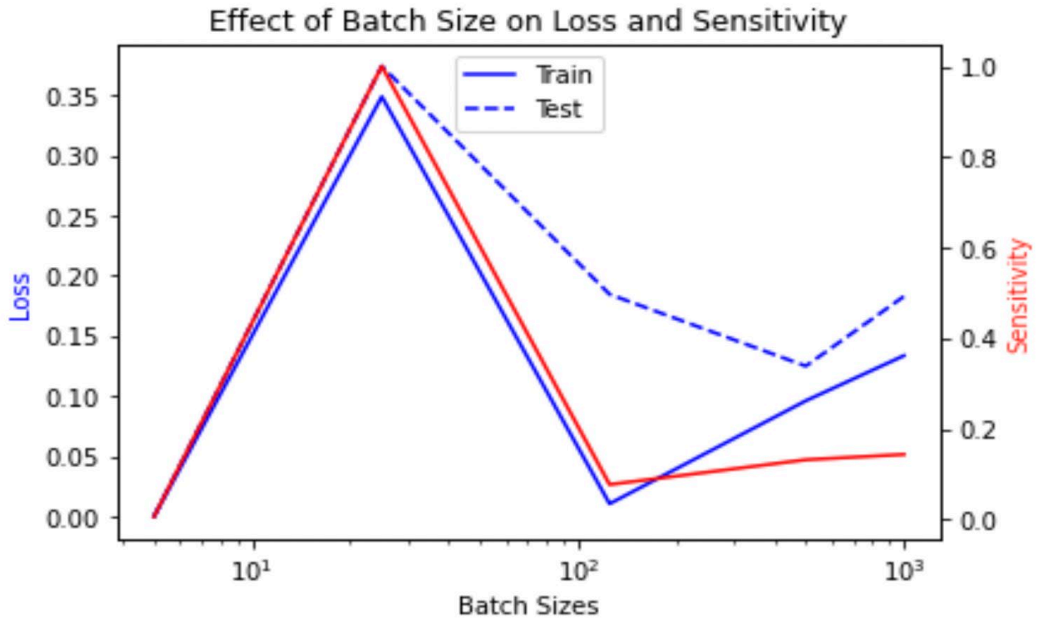

Figure 16

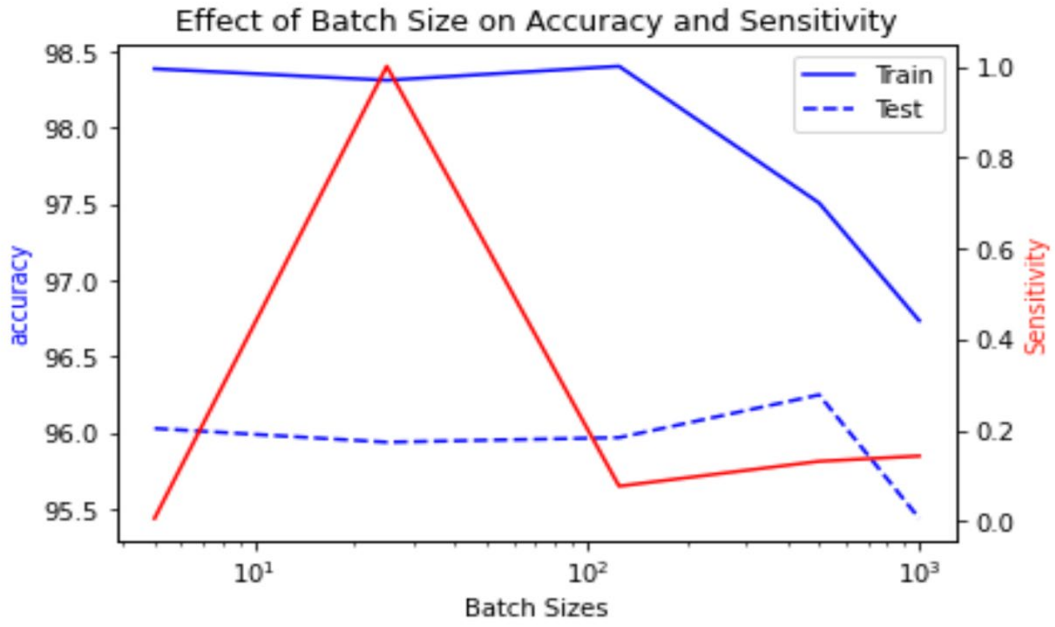

Figure 17

-After training, the accuracy and loss were calculated for the training dataset and test dataset of all five models. Then the sensitivity of the models was determined using the method forbenius norm of gradient.

-Figure 16 and 17 shows the accuracy and sensitivity are affected by batch size and loss and sensitivity. As batch size increases, sensitivity decreases.
Therefore, we conclude that the network will obtain the best results when the batch size is between 10^2 and 10^3.