MLP model for self driven cars

Anirudh Sharma
ID : 2015CSB1007

Aditya Ranjan
ID : 2015CSB1004

***Abstract*—We experimented with MLP(multi layer perceptron in this lab. We implemented a mode to predict the driving angle of a car on the road, given the image of the road as input. The MLP we implemented consists of 2 hidden layers. We experimented with different batch sizes and learning rates and plotted graphs of training error and validation error with number of epochs and noted the observations.)**

***Keywords*—MLP, epochs, learning rate, mini-batch sizes, training error, validation error.**

I. Question1 : MLP warmup exercise

We had to complete the pre-provided matlab script in this question. The forward and backward pass was implemented for a single hidden layer MLP model. A function to calculate the output was also implemented.
After the asked functionalities had been implemented, experiments were run on the model.
The graphs for the experiment are included in the graphs folder.

II. Basic neural network to predict the steering angle the road image for a self-driving car application

The input images of size 32 X 32 were provided by the instructor. A MLP model with 2 hidden layers of size 512 and 64 were implemented using mini-batch approach. Number of epochs, mini-batch sizes and learning rates were varied and the training and validation error were observed and noted with number of training iterations.

A. Plot of errors VS epochs

learning rate was set to 0.01. mini-batch size was fixed to 64 and a graph of training and validation mean squared VS number of training iterations(for 5000 epochs) was obtained. It was observed that the training error decreased very fast till about 1500 epochs and after that the training error started oscillating. However it was still decreasing.
The validation error decreased till about 500 epochs but it then started increasing at a slow rate. This observation is justified because the model overfitted with the increasing number of training iterations. As a result, training error was still decreasing while the testing error started increasing after 500 epochs. The graph is shown below :

B. Errors VS training iterations for different mini-batch size

Plots of error and training iterations was obtained for different mini batch sizes. Learning rate was fixed to 0.01 and number of epochs were set as 1000. Plots for mini-batch sizes of 32, 64 and 128 were obtained.
It was observed that the training and validation errors were least for mini-batch size of 32, a little more for 64 and most for 128.
This observation is justified because as the mini-batch size decreases, the model behaves as a simple gradient descent type model (absolute gradient descent model when the mini-batch size is set to 1). The plots are shown below.

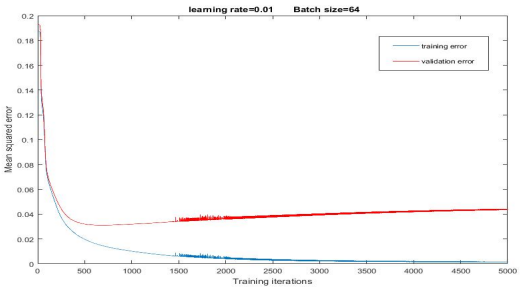

Fig. 1: plot of MSE VS training iterations

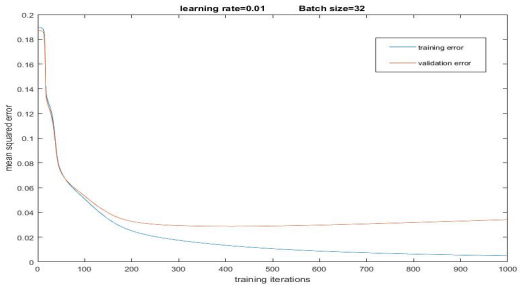

Fig. 2: plot of MSE VS training iterations for BS = 32

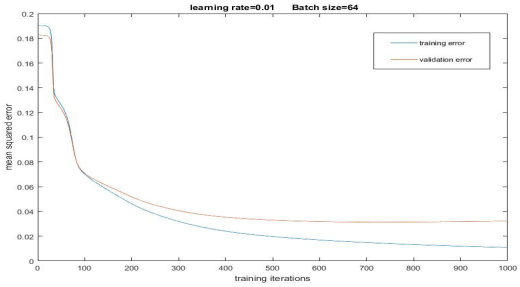

Fig. 3: plot of MSE VS training iterations for BS = 64

C. Implementing Dropout

Dropout was implemented. For that, random nodes according to the dropout probability was selected and were not

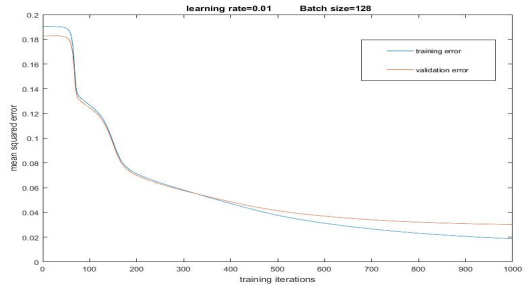

Fig. 4: plot of MSE VS training iterations for BS = 128

updated during backward propagation. It was observed that the training and testing ware both almost constant and oscillating. There was no convergence.
The reason for this because the learning rate was very low and number of training iterations were very less. The graph for the same is shown below :

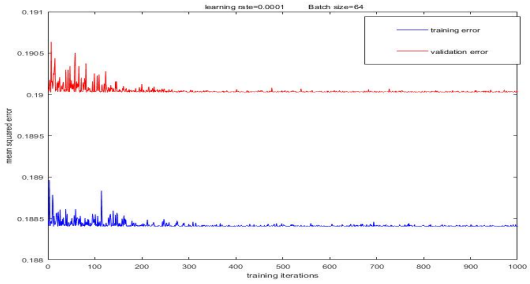

Fig. 5: plot of MSE VS training iterations for BS = 128

D. Error VS training iterations for different learning rates

Plots of training and validation errors VS the number of training iterations were obtained. The number of epochs was fixed to 1000 and mini-batch size was fixed to 64. learning rate was varied to 0.05,0.001,0.005.
It was observed that for the learning rate 0.05, the error decreased at a faster rate. But for 0.001 and 0.005, the error would have decreased even less if we have enough training iterations, more than 1000 as asked in question. The Plots are shown below.

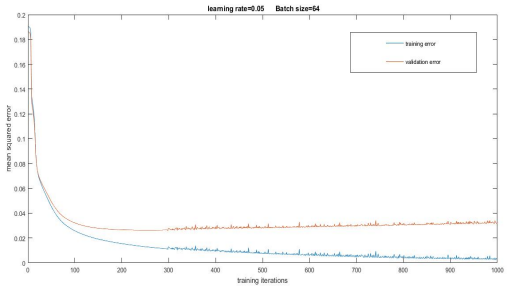

Fig. 6: plot of MSE VS training iterations for LR = 0.05

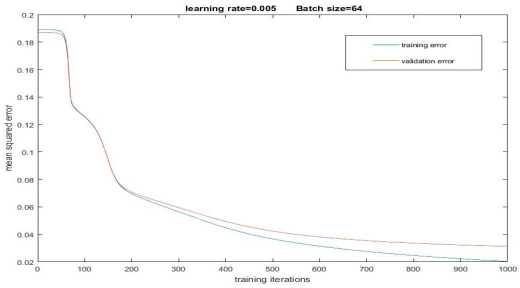

Fig. 7: plot of MSE VS training iterations for LR = 0.005

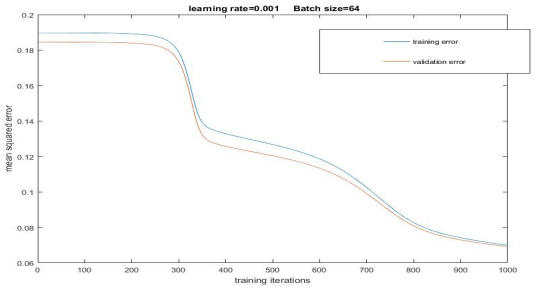

Fig. 8: plot of MSE VS training iterations for LR = 0.001

III. The competition part

For the competition part, help of the observations from above experiments was taken. First the learning rate was decreased for training and set to 0.0005. number of epochs was set to 10,000 for better convergence. batch size was set to 50 because it was observed above that smaller the batch size, better is the accuracy.
Now the model was trained and the validation error was noted for each training iteration for all 10,000 iterations. The value of the u,v and w matrix for which the validation error came out to be least was noted. Those values were then used to predict the output of the test data. The least MSE for validation set was found to be 0.0307. The plot for the trained model for competition is shown below.

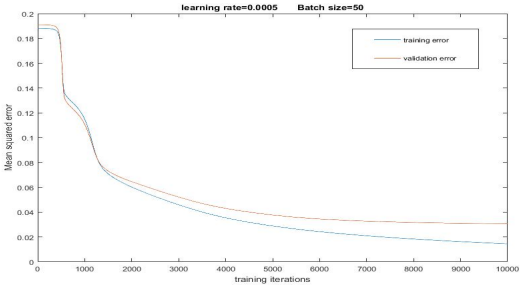

Fig. 9: plot of MSE VS training iterations for LR = 0.001, BS = 50, epochs = 10000 For competetion part

References

[1] http://cse.iitrpr.ac.in/ckn/courses/f2017/csl603/csl603.ht