Alexander Gibbons

CMPT 465

Due: December 13, 2023

Dr. Igor Aizenberg

MLP Simulator

The final project I chose for this class was the design of a MLP simulator that employs three learning techniques: minimization of Learning Error, minimization of Validation Error, and incremental learning in conjunction with minimization of Validation Error. This project was completed with a partner, so I will not be covering the entire design of the project, but rather the parts that I contributed to. In this paper I will discuss the design of LearningMLPWithValidation, the decision to add an Early Stop to the functions, as well as performing tests on all the functions.

To design LearningMLPWithValidation I used LearningMLP.m as a base. Not many changes needed to be made. For starters, a parameter was added to the function call for the Validation Set. This Validation set would be used to validate the weights the MLP was obtaining over the course of its learning. The results of this validation were then used to compute the Root Mean Square Error of Validation Error. This RMSE was checked against the error threshold and the function would exit if the threshold was passed. If not, the process would start again. A computer code with black text

Description automatically generated

When designing and testing the functions, I noticed that often times the MLPs for LearningMLPWithValidation and IncLearningwValidation would stop learning at a certain point or would become over fitted. This resulted in comparably worse classification rates when they were tested. To solve this issue, I employed an Early Stop. The function keeps track of iterations\_without\_improvement. If the RMSE of Validation Error from the current iteration did not perform better than the previous best RMSE, iterations\_without\_improvement would be incremented. I tested a range of thresholds for this check and ultimately decided upon 2500 iterations as this resulted in the least overfitting. A computer screen shot of a code

Description automatically generated

I decided to perform 5 experiments on 5 different neural network structures. These neural network structures ranged from 2 hidden input neurons to 10. The mean classification rate for each neural network structure was then calculated. The means of all neural network structures for each test was also calculated and put in a table.

A screenshot of a table

Description automatically generated

After performing a series of tests I was able to draw a conclusion. Our original hypothesis was that Incremental Learning in conjunction with minimization or Validation Error would be the best performing MLP. In two of the tests this was true. In those tests, the error threshold for the two functions with validation was not the same as the error threshold for LearningMLP, instead the error threshold that was used came from the testing RMSE of the LearningMLP. The best(smallest) testing error for each neural network structure was passed to the two functions with validation. The data below shows the difference between passing an error threshold of 0.1 to all three functions and passing 0.1 to LearningMLP, then passing the best testing error to the latter two functions.

A table with numbers and text

Description automatically generated

Passing 0.1 error threshold to all three

A screenshot of a table

Description automatically generated

Passing best testing error to latter two functions

The decision to use the best testing error allowed the validation functions to not become over fitted, as well as gave us a bench mark to see if the functions were performing better. When this was not used, the validation functions fell short.



Additional data is provided below.

Running the MLP Simulator performing 5 experiments with 8 hidden input neurons:

Minimization of Learning Error with LearningMLP.m

A screenshot of a table

Description automatically generated

Minimization of Validation Error with LearningMLPWithValidation.m

A screenshot of a table

Description automatically generated

Incremental Learning in conjunction with Validation Error

A screenshot of a data

Description automatically generated

Mean Results

A screenshot of a computer error

Description automatically generated

Running the MLP Simulator performing 1 experiment with 60 hidden input neurons, iteration limit of 10,000 and Early Stopping at 2500:

Minimization of Learning Error with LearningMLP.m

A screenshot of a computer

Description automatically generated

Minimization of Validation Error with LearningMLPWithValidation.m

A close-up of a computer screen

Description automatically generated

Incremental Learning in conjunction with Validation Error

A screenshot of a computer

Description automatically generated

Mean Results

A white rectangular sign with black text and numbers

Description automatically generated