Multi-Layer Perceptron

Logan Ladd logan.ladd@ecat1.montana.edu
Gianforte School of Computing
Montana State University
Bozeman, MT 59717-3880, USA

Asher Worley asher.worley@ecat1.montana.edu
Gianforte School of Computing
Montana State University
Bozeman, MT 59717-3880, USA

Editor: Logan Ladd

Abstract

The MLP neural network is a supervised training method used for classification or regression. While applicable in various problems, it requires tuning of parameters to perform optimally. When tuned correctly, the network is capable of high performance. In this paper we test the MLP model on six different data sets, three classification sets and three regression sets. We hypothesize that the model will perform better on the classification data sets than on the regression data sets. In the end, we reject this hypothesis. We also found that certain data sets perform far worse than expected. These data sets likely required more hidden nodes than what we tested with.

1. Introduction

Neural Networks are historical in the Machine Learning context, originating with Warren McCulloch and Walter Pitts, two neuroscientists in 1943. The original purpose was to model the neurological functions of the brain, however as time progressed the focus in the field shifted to computational applications. The backpropagation rule was published to move out of the realm of linear problems. The MLP, multi-layer perceptron, is a simple yet powerful model that is composed of nodes that take in inputs with weights and outputs to another node in the network. It is composed of at least three layers, an activation layer, a hidden layer, and an output layer, with additional hidden layers if necessary. The MLP network is often used as a universal function approximator. The network will be tuned to handle the dataset and performance will be evaluated. The task at hand is to find a model that predicts a real value or class based on a feature set.

We have two different types of data sets used in the paper, classification sets and regression sets. For this paper we will be evaluating three different sets of each type. We will be attempting to evaluate the efficacy of the MLP model on each set.

Our hypothesis is that the MLP will perform better on the classification data sets.

We test each data set with an MLP with zero, one, and two hidden layers. Of note is that, even though a true MLP always has at least one hidden layer, we will be testing with zero in order to assess the difference that it makes.

1.1 Data Sets and Preprocessing

All examples in the data set are scrambled at random and then assigned to sets for ten-fold CV same as the last project. All categorical variables are converted to integers for convenience. Our preprocessor also generates a similarity matrix for each variable to determine distances between categorical variables. All numerical variables are normalized between 0 and 1.

Data Set	# of Features	Model Type Needed
abalone[5]	4177	Regression
glass[3]	214	Classification
soybean-small[4]	47	Classification
breastcancer[6]	2310	Classification
machine[2]	209	Regression
forestfires[1]	512	Regression

1.2 Model

The MLP has the basic structure of an input layer, hidden layers, and an output layer. Functionality of layers, nodes weights, and backpropagation are generic, in that each node and layer does not represent a specific feature or process in analysis. Each layer may use linear or sigmoidal activation function. The sigmoidal activation function in our case will be the logistic function. Is its defined as:

$$f(x) = \frac{1}{1 + e^{1-x}}$$

Again, MLP networks may have an arbitrary number of hidden layers with an arbitrary number of hidden nodes. Each MLP will require the number of both of these hidden entities as well as an activation function to store each layer. Weights of the layers are initialized to a value between -0.01 and 0.01. Tests will mostly use sigmoidal activation functions in the hidden layers to ensure the network functions as a universal approximator. Training with an arbitrary amount of examples is done in batches using backpropagation from the output through hidden layers.

Backpropagation is the process by which the weights of each node are adjusted in order to minimize error in the model. It works by calculating the gradient of the error function with respect to the nodes that precede each individual node. This is calculated via the product rule. This gradient is then multiplied by the learning rate, n, which is a hyperparameter. After this, the resulting value is added to the previous gradient for that weight multiplied by the momentum factor, which is an optional hyperparameter. Each individual weight must be calculated separately.

The networks detect the type of dataset we are using and the output layer is chosen upon training. A classification network will have an output node tied to each class in the dataset, each with a sigmoidal activation function. Predicted class will be chosen based on the node with the maximum activation value. A regression network will have a single output node with a linear activation function. Predicted values will be the activation value of this node given an input.

1.3 Hyperparameter Tuning

Hyperparaemeter tuning done with the learning rate n which is the step size at each iteration in gradient descent. For classification sets, the initial value was set to 0.1, the other values tested include 0.0001, 0.001, 0.01, 0.2, 0.3, 0.4, 0.5, 1, 2, and 4. Learning rates below 0.1 generally performed the worst and rates above 0.1 did not see better performance except on larger classification data sets because they contain significantly more points, so using the same number of iterations the learning rate can be larger to find the local minimum. The regression sets performed best with a learning rate of 0.1, generally the same as for the classification sets.
Another value tuned was the momentum multiplier. This value was tested at 0.5, 0.25, and 0. There was negligible change in the mean squared error when momentum was increased from 0.25 to 0.5, and accuracy practically experienced no change. Regression data sets as a whole performed the best with a momentum multiplier of 0.25.

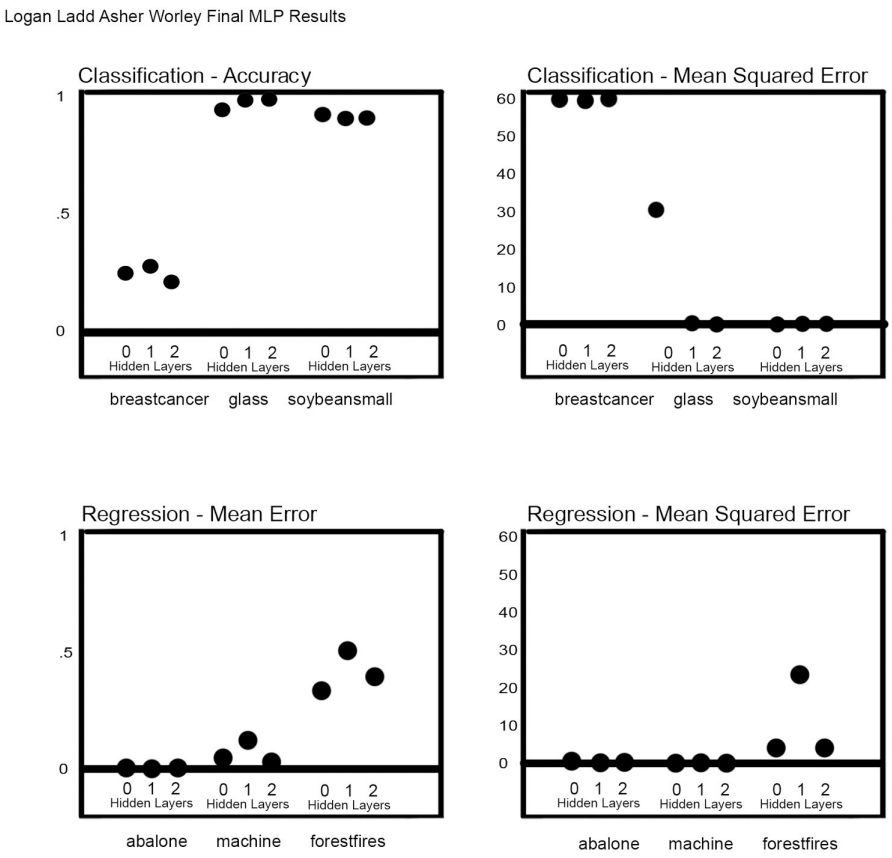
1.4 Evaluation Metrics

In order to evaluate the results of our tests, we used accuracy, error, and mean squared error. For the classification problems, accuracy and mean squared error were used. For the regression problems, we used mean error and mean squared error. Accuracy tells us what percentage of the testing set was classified correctly, mean error tells us how far, based on Euclidean distance with the regression sets, or based on the similarity matrix for the classification sets, and mean squared error is simply the square of the mean error.

1.5 Results

Data Set	#of Hidden Nodes	Step Size	Accuracy	Mean Squared Error
Breast Cancer[6]	0	0.2	25%	316.80
Breast Cancer[6]	1	0.2	26%	248.81
Breast Cancer[6]	2	0.2	17%	4835.91
Glass[3]	0	0.1	90%	29.45
Glass[3]	1	0.1	99%	0.75
Glass[3]	2	0.1	100%	0.30
Soybean-Small[4]	0	0.1	90%	0.60
Soybean-Small[4]	1	0.1	88%	1.05
Soybean-Small[4]	2	0.1	88%	1.11

Data Set	#of Hidden Nodes	Step Size	Mean Error	Mean Squared Error
Abalone[5]	0	0.1	0.00	0.65
Abalone[5]	1	0.1	-0.03	0.67
Abalone[5]	2	0.1	-0.02	0.71
Machine[2]	0	0.1	0.04	0.30
Machine[2]	1	0.1	0.06	0.28
Machine[2]	2	0.1	0.01	0.23
ForestFires[1]	0	0.1	0.31	2.76
ForestFires[1]	1	0.1	0.48	26.40
ForestFires[1]	2	0.1	0.40	2.23



We found that, taking the best number of hidden nodes for each data set, the categorical data sets had an average mean squared error of 83.24, whereas the regression data sets had an average mean squared error of 1.04. It is obvious upon inspection that the breast cancer data set performed so badly that it skewed the rest of the categorical data. Taking this into account and dismissing this data, we get an average mean squared difference of 0.45.

If we were to ignore the breast cancer data, it would be very easy to suggest that the

difference in means of 0.64 is significant. However, the breast cancer data cannot simply be ignored. It is possible that the MLP found a shallow local minimum for the breast cancer data, which is why is performed the way it did. However, in theory the addition of a non-zero momentum value should counteract that potential. Indeed, when testing using different values for momentum, the same mean squared error was found. This implies that something else is occurring within the data. It is interesting to note that a better performing model would have been to flip a coin to determine whether the patient had malignant cancer. Another possibility is that the data set required an MLP with more that two hidden layers in order to perform optimally.

Of note is that the regression data sets also included an outlier. While not to the same extreme, the forestfires data set performed significantly worse than the rest of the regression data sets. When taking this into account, the average mean squared error, again taking the best number of hidden nodes, was 0.51. Without the outliers, the difference in averages was only 0.06. This is an insignificant amount, given that that amount of difference in mean squared error won't effect the prediction accuracy very much. Given this, we are forced to reject the hypothesis.

2. Summary

The difference in mean squared error when not accounting for outlying data implies that MLP's perform worse on categorical data sets than on regression data sets. When they are accounted for, the difference was found to be negligible. Taking these into account, we rejected the hypothesis. The fact that some of the data lay extremely outside the normal range of the other data sets implies that there is something special about the data. It was determined that, since the addition of a momentum hyperparameter didn't change the results, this was not because the MLP found a local minimum. More tests with more than two hidden layers are required to conclusively determine why the model performed outside the expected range on that data.

References

[1] Paulo Cortez and Anibal Morais. *UCI Forest Fires*. 2007.

[2] Phillip Ein-Dor and Jacob Feldmesser. *UCI Relative CPU Performance Data*. 1987.

[3] B. German. *UCI Glass Identification Data Set*. 1987. URL: `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`.

[4] R.S Michalski. *UCI Small Soybean Database*. 1987.

[5] Dr. Sam Waugh. *UCI Abalone Data*. 1995.

[6] Dr. William H. Wolberg. *UCI Wisconsin Breast Cancer Database*. 1991.