Neural Networks and Backpropogation

Amrith Deepak

Abstract:

The breast cancer data showing tumors were malignant, glass data showing type of glass, Iris data showing species of Iris plants, soybean data examining the class of soybeans, and vote data examining the party of house congress votes were analyzed using a feedforward network trained with backpropagation. All the datasets except the soybean dataset performed better with 1 and 2 hidden layers than 0 hidden layers. Between 1 and 2 hidden layers, the performance was similar, in some cases slightly higher and in other cases slightly lower. The breast cancer, glass, and vote data had very high performance (over 90% accuracy) and the iris data has moderate performance(74%) and the soybean data had the lowest performance (27%).

Hypothesis:

It is predicted that more hidden layers will improve performance so performance for 1 hidden layer will be higher than 0 hidden layers, and performance for 2 hidden layers will be higher than that of 1 hidden layer. The performance will be high in all the datasets because the assumptions of backpropagation hold that the cost function can be written as an average over cost functions for individual example and the cost function can be written as a function of the outputs from the neural network.

Algorithms:

Backpropagation creates a network with nodes. It has hidden layers and output layers. First, forward propagation is used with initial weights. In the backpropagation step, the error function is calculated for each of the nodes. Then the weights are updated based on these errors, and the learning rate. This happens for each node. There is an activation function which calculates a weighted sum and then a sigmoid function similar to logistic regression is used to put the data point into a class.

Experimental Approach:

For each of the five datasets, the data was split into training and testing data using 5-fold cross validation. The algorithms described above were followed.

For all of the datasets, the variables were changed to binary numerical variables based on whether they were greater than or less than the mean. Missing values from the house votes dataset were randomly filled in. For the breast cancer data, missing values were removed since there were small in quantity.

To tune the number of hidden nodes per layer, I just experimented with a couple values, including 1, 3, 10, and 50 hidden nodes per layer. For the 2 hidden layers, I used 3 nodes per layer and for the one layer I used 10 nodes. The performance worsened for 50 nodes and for using 10 nodes per layer, the performance was almost identical for 2 hidden layers compared to 3 nodes so for saving time, I used 3 nodes per layer.

Results:

Dataset	Hidden Layers	Performance
Breast Cancer	0	0.6485
Breast Cancer	1	0.9706
Breast Cancer	2	0.9706
Glass	0	0.6529
Glass	1	0.9676
Glass	2	0.9662
Iris	0	0.2933
Iris	1	0.6667
Iris	2	0.74
Soybean	0	0.3556
Soybean	1	0.2667
Soybean	2	0.2667
Vote	0	0.6163
Vote	1	0.9349
Vote	2	0.9372

The performance for the breast cancer data was 0.6485 for no hidden layers and 0.9706 for both 1 and 2 hidden layers. For the glass data, the performance was 0.6529 for 0 hidden layers, 0.9676 for 1 hidden layer, and 0.9662 for 2 hidden layers. The iris data had 0.2933 performane with 0 hidden layers. 0.6667 with 1 hidden layer, and 0.74 with 2 hidden layers. The soybean data had performance 0.3556 with 0 hidden layers, but the performance dropped to 0.2667 for 1 and 2 hidden layers. The vote data had performance 0.6163 for 0 hidden layers, 0.9349 for 1 hidden layer, and 0.9372 for 2 hidden layers. These performance values are the average over 5 fold cross validation. The performance increased from 0 to 1 hidden layer for all except the soybean data and the performance increased a small amount from 1 to 2 hidden layers for the iris and vote data, stayed the same for the breast cancer and soybean data, and decreased a small amount for the glass data.

Discussion:

All the datasets except the soybean dataset performed significantly better with 1 and 2 hidden layers than 0 hidden layers. The first hidden layer improves performance since without a hidden layer, there is no layer between the input and output layer. This layer is very useful in going from a set of weighted inputs to outputs. Having a second hidden layer in most cases made an insignificant change in performance. The soybean dataset was small and the backpropagation did not work well. Even adding more hidden layers did not help performance. For the breast cancer, vote, and glass data, the performance was strong with over 90% accuracy with 1 and 2 hidden layers, and over 60% accuracy with 0 hidden layers. The iris data also showed a increase in performance from 29.33% with 0 hidden layers to 66.67% accuracy with 1 hidden layer and 74% accuracy with 2 hidden layers. The soybean data not only had the lowest performance but also saw a reduction in performance from 35.56% to 26.67%. from 0 to 1 hidden layer. This shows that for most of the datasets, backpropagation worked effectively, except for the soybean dataset. This is probably due to the small dataset size, so the training was not sufficient.

Summary:

All the datasets except the soybean dataset performed better with 1 and 2 hidden layers than 0 hidden layers. Between 1 and 2 hidden layers, the performance was similar, in some cases slightly higher and in other cases slightly lower. The breast cancer, glass, and vote data had very high performance (over 90% accuracy) and the iris data has moderate performance(74%) and the soybean data had the lowest performance (27%). The code trained using 5-fold cross validation where each time 20% of the data was used for testing and 80% for training.

Datasets:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

https://archive.ics.uci.edu/ml/datasets/Glass+Identification

https://archive.ics.uci.edu/ml/datasets/Iris

https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29

https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

Works Cited

Cun, Yann Le. *A Theoretical Framework for Back Propogation*. Department of Computer Science University of Toronto, 1998, yann.lecun.com/exdb/publis/pdf/lecun-88.pdf.

Gotesman, Ryan. *Learning Backpropagation from Geoffrey Hinton*. Towards Data Science, 17 Apr. 2018, towardsdatascience.com/learning-backpropagation-from-geoffrey-hinton-619027613f0.