Anthony Gray
EN.605.649.81
Intro to Machine Learning
Project 6

Abstract

Multilayer Perceptron networks were implemented and trained with backpropagation to classify examples from five public datasets. A tuning step was used to determine the best network structure for each dataset. Networks had between zero and two hidden layers, and each hidden layer had several nodes ranging between one and the number of attributes in the dataset. Classification accuracy averaged about 95% on three of the datasets, about 68% on one dataset, and about 26% on the last dataset. It is likely that the implementation of momentum in the training process, as well as more time spent tuning and training the networks would result in greater classification accuracy on all datasets. A correlation was also observed between the number of attributes and a reduced need for hidden layers.

Problem Statement and Algorithms

This project performs classification on five different datasets using multilayer perceptrons trained with backpropagation. The perceptron networks are structured with an input layer, an output layer, and a variable number of hidden layers in between. The input layer contains a node for each attribute in the dataset, and those nodes correspond to the instance of data to be classified. The output layer contains several nodes equal to the number of possible classes in the dataset, and the network assigns the class value to an instance corresponding to the output node with the highest activation after the values of the instance propagate through the network. The variable number of hidden layers also contain a variable number of nodes in each layer. In this implementation, all hidden layers had the same number of nodes, but that number was tuned for classification accuracy based on the dataset being tested. The nodes of each layer are connected to the nodes of each adjacent layer by weighted edges. The input values are multiplied by these weights, added to a bias term, and passed through the logistic function to provide the activation values of the nodes in the next layer. These activation values are then treated as the input values for the next layer, and the process repeats until the activation values for the output layer are generated and the corresponding class is assigned.

The networks in this project were initialized with random weights and trained with backpropagation (6). Backpropagation works by calculating the error of the networks output when compared to the expected output for a specific test instance, and minimizing that error using gradient descent by adjusting the weights between nodes according to that gradient. The weights connecting the output layer to the penultimate layer are adjusted, and the weight adjustment is propagated back through the network by repeating this process for the next layer back. The weights connecting the penultimate layer to the next layer back are adjusted, and so on until the weights connecting the input layer to the second layer are adjusted. A learning rate is used to control the rate of gradient descent, and this process is repeated for each input until a convergence is reached. For the tuning phase, a maximum number of epochs was also implemented, since perfect accuracy was not necessary to compare the performance of different network structures. Additionally, a momentum term is often used to influence the rate of gradient descent, though it was not implemented for this project.

It is likely that these trained networks will be able to successfully classify items from all five datasets without using the maximum number of layers and nodes. Considering that several of these datasets have been successfully classified using linear discriminant methods in previous projects, it is expected that networks with no hidden layers, effectively generating a linear discriminant, will be sufficient in some cases. Some datasets have been more challenging to classify than others, and even if a network with no hidden layers is effective, at least one hidden layer will likely improve its classification accuracy. The number of nodes in the hidden layers will likely be less than the total number of attributes, as the data is already generally linearly separable, and the maximum number of nodes will probably not be necessary.

Procedure/Tuning

Dataset values before treatment:

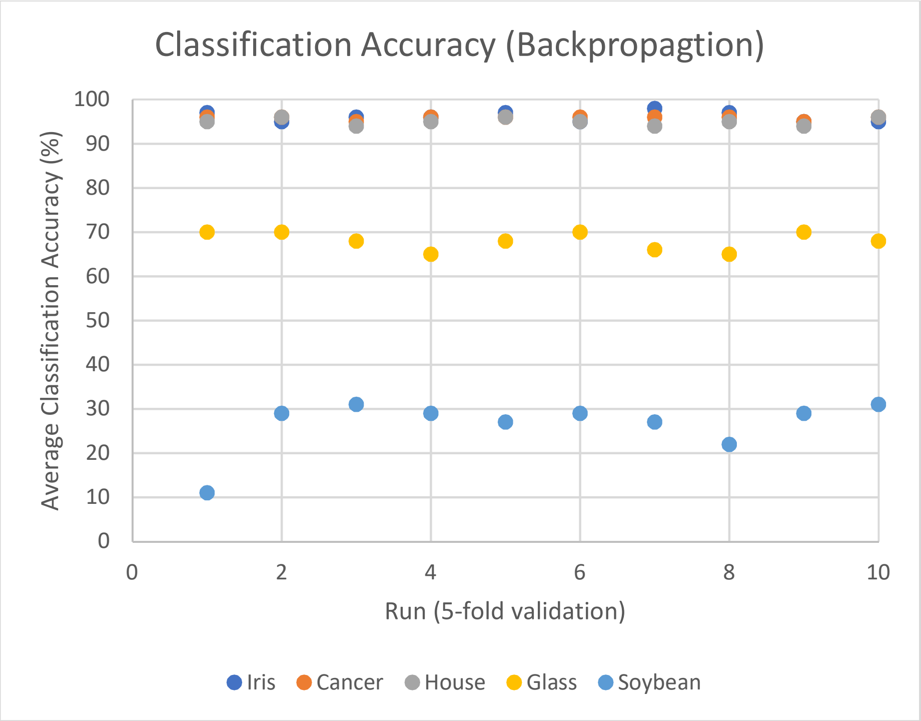
Name	Instances	Attributes	Classes	Data Type
Iris (1)	150	4	3	Continuous
Glass (2)	214	9	7	Continuous
House Votes (3)	435	16	2	Boolean
Breast Cancer (4)	699	10	2	Categorical
Soybeans (5)	47	35	4	Categorical

Each dataset was downloaded from its public repository and treated for use.

The Iris dataset contained no missing values or unlearnable attributes. The ID column of the Glass dataset was removed, as it is not a learnable attribute. Missing data in the House Votes dataset was too frequent to remove, so a random Boolean value was generated for each missing data point. Instances with missing data in the Breast Cancer dataset were dropped, as there were only 16 out of almost 700 examples with missing data. The unlearnable Sample number column was also dropped from the Cancer dataset. Column order was rearranged where necessary to ensure that the Class value was always in the last column, and the data types were changed to numeric for all values except the Class. Finally, the categorical Breast Cancer and Soybean datasets were converted to Boolean values using one-hot-encoding, resulting in 89 and 72 attributes respectively. The continuous values in the Iris and Glass datasets were normalized between 0 and 1 for each column.

The networks were tuned to each dataset by varying the learning rate, maximum number of epochs, number of layers, and number of neurons per layer. Learning rate was tested between 0.1 and 0.5 in increments of 0.1. Epochs ranged between 500 and 2000 in increments of 200. Networks were run with zero, one or two hidden layers, and the number of neurons in those layers ranged from 1 to the number of encoded attributes in the dataset. Tuning was performed by repeatedly testing the accuracy of each combination of parameters on a subset of each dataset and saving the parameters that produced the greatest accuracy. Each dataset was then evaluated with backpropagation ten times using five-fold validation and the average classification accuracy was recorded as a percentage.

Results



The trained networks were successful in classifying items from most of the datasets, with one glaring exception. Instances from the Iris, Breast Cancer, and House Votes datasets were classified reliably, with about a 95% success rate for all three datasets. The networks were also reasonably successful in classifying the instances from the Glass dataset, with about a 68% success rate, though further tuning is clearly required for reliable performance. The networks performed extremely poorly on the Soybean dataset, producing classifications with only about 25% accuracy. Considering that the Soybean dataset contains roughly equal proportions of four classes, this is no better than guessing randomly, and the networks would require much more rigorous training to approach any degree of utility with this dataset.

Discussion

Parameters used for each dataset:

Dataset	Layers	Neurons	Number of Inputs
Iris	2	4	4
Cancer	0	-	89
House	1	11	16
Glass	1	8	9
Soybean	0	-	72

The expectation that the networks would not need to contain the maximum number of hidden layers and nodes seems to hold. The only dataset that benefitted from the maximum number of layers and nodes was the Iris dataset, which achieved the best accuracy of all five sets with a network containing two hidden layers made of four nodes each. Even though this is the set highest number of hidden layers, it is also the set with the fewest number of nodes in those layers. However, that is likely because the Iris dataset only had four attributes, the fewest of all five datasets. It is possible that additional hidden nodes may have increased accuracy, though for the purposes of these tests, the number of nodes per hidden layer was capped at the number of nodes in the input layer. The other datasets produced the best results when classified with networks containing less than two layers. The House and Glass datasets used one hidden layer, while the Cancer and Soybean datasets didn't use a hidden layer at all. There does seem to be a negative correlation between the number of inputs and the number of hidden layers necessary for classification, as the datasets that were classified without hidden networks also had the greatest number of attributes after encoding. As mentioned above, these datasets are known to be linearly separable, so it is possible that the hidden layers are only necessary for these datasets when there are a limited number of attributes. There may also be a similar relationship between the number of attributes in the dataset and the number of nodes required in the hidden layers. The two datasets that used a single hidden layer also used fewer than the maximum possible number of nodes in that layer.

Average Classification Accuracy (%) by Dataset

Dataset	Backpropagation	Logistic Regression
Iris	96.1	93.7
Cancer	95.8	95.5
House	95.0	95.2
Glass	68.0	47.2
Soybean	26.5	100.0

The extremely poor performance of the networks on the Soybean dataset is surprising, considering the accuracy on the other datasets and how accurately the same dataset was classified with Logistic Regression in a previous project. The relative classification difficulty of the other datasets was essentially the same between the two projects, with accuracies of around 95% for the Iris, Cancer and House sets and the accuracy of the Glass dataset lagging behind the other three. In fact, the backpropagated networks performed significantly better than logistic regression on the Glass dataset. Considering that the Soybean dataset is known to be linearly separable and considering that the networks were able to classify the other four datasets with at least acceptable levels of accuracy, it is likely that the Soybean networks were improperly tuned. Allowing more time for convergence on each tuning step, as well as relaxing some of the ranges for parameters like the learning rate or the maximum number of nodes in each hidden layer could result in better accuracy.

Summary

In general, the networks trained with backpropagation performed well. Three of the five datasets were classified with very high accuracy, and one of the datasets with poor performance was easy to classify in the past. Given these factors, the inaccuracies on the Glass and Soybean datasets are likely due to poor tuning. The tuning process was not as rigorous as it could have been, since, for the sake of time, a maximum number of epochs was used instead of convergence for this step, and only a few values for the learning rate were tested. It only took a few runs of the tuning process to settle on parameters for the three datasets with high accuracy, but a long time was spent tuning networks for the two datasets with lower accuracy. The process was repeated several times, but it became clear that a more flexible, longer tuning period would be needed to find the best parameters for both datasets. Additionally, momentum was not implemented in the gradient decent step of the training process. It is possible that the inaccuracy associated with the Soybean dataset was caused by the weight updates getting caught in a local minimum, and never truly converging on a configuration that learned the data. With more time taken in the tuning and training phases, it is likely that the networks could classify all five datasets with a reliable degree of accuracy.

References

1) Creator: R.A. Fisher Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov) Date: July, 1988
2) Creator: B. German -- Central Research Establishment Home Office Forensic Science Service Aldermaston, Reading, Berkshire RG7 4PN Date: September, 1987
3) Title: 1984 United States Congressional Voting Records Database Source: Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc. Washington, D.C., 1985. Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu) Date: 27 April 1987
4) O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.
5) Michalski,R.S. Learning by being told and learning from examples: an experimental comparison of the two methodes of knowledge acquisition in the context of developing an expert system for soybean desease diagnoiss", International Journal of Policy Analysis and Information Systems, 1980, 4(2), 125-161. Donor: Doug Fisher (dfisher%vuse@uunet.uucp) Date: 1987
6) Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.