Comparing Neural Network accuracies based on dataset granularities, hidden neuron counts and network complexities

Irtza Suhail,[1]

[1]u6269726@anu.edu.au

Abstract. In real life complex problems, we are often given large amounts of data to use to process and use in training neural networks. In this paper, we used the dataset for predicting forest cover types to investigate how using a partial dataset from the data provided affected our neural networks accuracies in both training and testing. We also studied how changing the number of hidden neurons affected both training and testing accuracies. The accuracies obtained were less than those obtained by Blackard and Dean. With our basic network, we achieved accuracies of upto 51.12% for training and 49.32% for testing. We also investigated how changing the network topography affected our accuracies, with the addition of a rectified linear layer improving our accuracies by approximately 2%.

Keywords: Artificial intelligence; Forest cover types; Geographic information systems (GIS); Neural networks

1 Introduction

When we are tasked with creating accurate neural networks to classify data for us, it is a complex task that often involves a large amount of data. This high volume is often attributed both to a very large collection of data points, as well as to extensive variables being monitored per data point. To this end, at times it is feasible to perform some preprocessing before handing the data to the neural network for classification and training. In this paper we will be using the forest cover type data set for training and testing. The reason we are using this data set is that it is a sufficiently large data set with approximately 58000 data points available. Furthermore, each data point has 54 variables stored, thus increasing the complexity of the data set and as such giving us multiple ways we can approach the data manipulation and preprocessing.

To investigate how data manipulation and preprocessing affects the neural network's performance, we conducted several different experiments. Blackard limited the dataset to approximately 11000 training and 4000 testing data points and removed certain columns from the data or tried to combine them into more generalized variables (Blackard & Dean, 1999). Likewise, we also limited our data set and removed or combined columns to reduce the data complexity. We ran the data through a simple two-layer neural network with a varying number of hidden neurons, while keeping the learning rate constant at 0.05 and with 1000 epochs. We reviewed the training and testing accuracies of our neural network for all the different data sets we ran through it.

The motivation behind this neural network is to classify large swathes of forest automatically and accurately. To this end, the data collected contains 54 points of interest on each sample. These include the elevation, aspect, slope, vertical and horizontal distances to hydrology, horizontal distances to roadsides and fire points, hill shade at 9am, noon and 3pm, as well as 4 binary wilderness area designations and 40 binary soil designations. These are used to classify the area into one of the following seven different forest cover types, lodgepole pine, spruce/fir, ponderosa pine, Douglas-fir, aspen, willow, krummholz (Blackard & Dean, 1999).

2 Method

The dataset used is the cover type data set. These include the elevation, aspect, slope, vertical and horizontal distances to hydrology, horizontal distances to roadsides and fire points, hill shade at 9am, noon and 3pm, as well as 4 binary wilderness area designations and 40 binary soil designations. These are used to classify the area into one of the following seven different forest cover types, lodgepole pine, spruce/fir, ponderosa pine, Douglas-fir, aspen, willow, krummholz (Blackard & Dean, 1999).

To investigate how using a smaller dataset would affect the neural network's accuracies in training and testing, we conducted several small experiments based on the work performed by Blackard. The first step was to create a simple two-layer neural network. To achieve this, we created a neural network with one hidden layer comprising of varying numbers of hidden sigmoid neurons. We used the same values used by Blackard, which are 6, 12, 18, 24, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300 (Blackard & Dean, 1999). For all iterations, we kept the learning rate constant at 0.01 and used a fixed number of epochs (1000). Once this network was set up, we ran all the data with all data points through it to get a baseline for the training and testing accuracies. Once this was done, we moved on to the main step of pre-processing the data.

We also investigated how changing the network itself affects the training and testing accuracies. To this end, after selecting one fixed value for our hidden neuron count, we ran the dataset through 2 further networks. One of these networks contained a rectified linear unit layer after the sigmoid layer, while the other contained 2 sigmoid and 1 rectified linear unit layers. As before, our learning rate and epochs were held constant at 0.01 and 1000 respectively.

Before the preprocessing of the data could begin, we culled the data. To do this, we selected the least common cover type, willow, and selected ~60% of its data which came out to be 1620. Then to match this, we selected an equal number of data points from the other cover types. This was done as using 60% of the available data for training is the most efficient way to use the data provided (Blackard & Dean, 1999). From the remaining data, 540 data points (20% of the least common cover type) were selected as our testing data set from each cover type. This gave us a total of approximately 11000 training and 400 testing data points. This data was also run through our neural network and its results recorded.

Once we had the required data, we could start with the preprocessing. From Blackard's paper, we selected 3 different methods of preprocessing. In the first method, the 40 binary soil types were dropped from the dataset and the remaining 14 variables were used in the training and testing of the neural network. In the second method, not only were the 40 binary soil types dropped, the 4 binary wilderness types were also dropped, further reducing the number of variables to 10. This data set was also run through the neural network and its results recorded. In the final variation, the 40 binary soil types were combined into 6 variables, thus reducing the complexity of the data and making the training and testing less computationally expensive.

For each of these data sets, we ran them through the neural network with varying numbers of hidden neurons so that the best fit could be found without making any assumptions (Marzban & Stumpf, 1996). The results are given below and compared to the results obtained by Blackard.

3 Results and Discussion

We processed the data through the neural network and recorded the results below.

Table 1. All data (54 variables, 58000 data points) trained and tested against the neural network

Number of Hidden Neurons	Training Accuracy (%)	Testing Accuracy (%)
6	44.40	42.18
12	43.19	47.16
18	45.76	47.96
24	44.21	49.32
30	46.75	45.20
60	41.59	43.82
90	44.34	41.92
120	47.53	45.58
150	48.38	41.00
180	47.23	47.12
210	45.19	47.38
240	49.92	43.36
270	49.13	47.90
300	49.73	46.72

As we can see from the results, there is a general trend towards better results as the number of hidden neurons increases. This is likely due to the fact that as the number of hidden neurons increase, the network is better able to perceive the differences between the samples and generate more accurate predictions based on the results. Compared to Blackard's accuracy of 70.58%, we can see that our network is lacking in certain aspects that reduce its accuracy (Blackard & Dean, 1999).

Table 2. Limited data (54 variables, 15000 data points) trained and tested against the neural network

Number of Hidden Neurons	Training Accuracy (%)	Testing Accuracy (%)
6	45.31	42.82
12	45.31	42.88
18	45.38	47.78
24	43.42	47.62
30	44.23	48.41
60	46.25	49.37
90	46.78	48.86
120	47.17	49.44
150	50.66	45.48
180	49.77	47.80
210	50.57	45.16
240	50.44	45.87
270	50.48	45.82
300	51.12	44.71

can see that reducing the data slightly improved the overall training accuracies, as there was less chance of overlearning the information and generating too specific models for classification. Similarly, there was some improvement in the testing accuracies due to having less data to learn and compare against.

Table 3. Limited data (14 variables, 15000 data points) trained and tested against the neural network

Number of Hidden Neurons	Training Accuracy (%)	Testing Accuracy (%)
6	20.48	20.29
12	22.83	22.99
18	24.77	25.98
24	25.42	22.96
30	30.18	29.71
60	30.06	29.66
90	31.89	31.06
120	36.06	36.59
150	34.50	34.58
180	36.00	36.85
210	37.51	36.24
240	36.76	35.74
270	37.24	36.75
300	37.35	36.98

We can immediately see a significant loss of accuracy in both training and testing due to a loss of data. This meant that there was less data for the network to learn from and as such it suffered from poor accuracies in testing and training. Similarly, Blackard lost some accuracy as well, dropping to 58.38% from 70.58% (Blackard & Dean, 1999).

We can see that having 300 hidden neurons gives us the best accuracies, regardless of data used. Therefore, for the next part, we will be using 300 hidden neurons in our testing.

Table 4. All data (54 variables, 58000 data points) trained and tested against the neural network with a ReLU layer added

Number of Hidden Neurons	Training Accuracy (%)	Testing Accuracy (%)
300	53.46	82.02

We can see that adding a rectified linear unit layer increases our accuracies for both testing and training. However, the testing accuracy is abnormally high and possibly incorrect.

Next, we added a second layer of sigmoid and relu neurons to the network and observed their accuracies.

Table 5. All data (54 variables, 58000 data points) trained and tested against the neural network with a ReLU layer and a sigmoid layer added

Number of Hidden Neurons	Training Accuracy (%)	Testing Accuracy (%)
300	40.42	79.28

Here we see that adding a second layer of sigmoid neurons reduces the accuracies of the network, both in training and testing. However, Blackards results are still superior to the results we obtained due to our neural network not being as well programmed as theirs. This is something we can improve in our future iterations of the work.

4 Conclusion and Future Works

In conclusion we can infer two theories from the experiments performed above. One, that increasing the number of hidden neurons improves training and testing accuracies, as the network is better able to discern the minute differences between the different data points made available to it. While Blackard achieved much more impressive results when compared to ours, this can be explained by us using a much simpler neural network with a sigmoid activation function. If we were to implement a network more closely resembling the one created by Blackard, we too could achieve similar results from the data.

Secondly, we can see that reducing the data, can have both good and bad consequences. Reducing the number of data points made available means that the network does not overlearn and make mistakes that could reduce its testing accuracy. Furthermore, using less data means that the network is able to process the data faster and come to a conclusion in a much more reasonable time frame.

However, reducing the number of variables stored per data point can negatively impact the training and testing, as the network may lose out on crucial information it may need to discern the at times minute differences between different classes. We can see from our data that reducing the number of variables severely impacted our results, with drastically reduced accuracies in both training and testing. However, with better preprocessing, and careful consideration of which variables to prune out, we can improve our accuracies and achieve better results.

For future works, there are several improvements that can be made to the current experiment. Firstly, we can improve the neural network used. We can implement a back propagating neural network like the one described in the paper, this would mean that our results would more closely mirror the results of the paper. Furthermore, it would improve the overall accuracy of our neural network and give us better results as a consequence.

Secondly, we can improve how we preprocess the data. By carefully examining the data and pruning variables deemed insignificant, we can improve the speed of the classification while not compromising on the quality of the results obtained. Furthermore, by using normalization to bring certain data columns within a zero to one range we can again further improve our accuracy.

Lastly, the paper mentioned combining the forty binary soil types into six types, thereby reducing the number of variables and the complexity of the data. This would reduce the training and testing time while keeping the accuracy the same as it was before. If we were to implement this, it could improve our network’s performance as well as its accuracy.

References

Blackard, J. A., & Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 131-151.

Marzban, C., & Stumpf, G. J. (1996). A neural network for tornado prediction based on Doppler radar derived attributes. *Journal of Applied Meteorolgy*, 617-626.