Forest Cover type prediction using Deep Neural Network

Syed Ali Hussain

u6028474@anu.edu.au

Abstract. This study builds on top of the results produced previously where an Artificial Neural Network with one layer was used to predict different forest covertypes. In this study we used a Multilayer Perceptron (MLP), two-layered and three-layered, to predict the forest cover types and compare our results with a single layer neural network. It was revealed that an MLP with three layers not only improves the prediction accuracy for one set of number of neurons but also gives a more dependable multiclass accuracy classifier. In addition to that, some plots were made to give us insight into the forest cover type data to define what future work might help us get data that would results in better neural networks.

Keywords: Artificial Neural Networks, Forest Cover types, Geographic Information Systems

1 Introduction

Identifying forest cover types is often necessary for forest management departments so that they could keep in check the ecological balance of an area. Often, forest cover types are identified using field personnel or by remote sensing techniques which are most of the times, both time-consuming and expensive [1]. Moreover, applying these techniques is not practical when area being studied is remote or in another state or territory [1]. In these cases, predictive models trained on existing data can be quite useful as they can approximate cover types for unseen data.

The data belongs to four wilderness areas of Roosevelt National Forest namely Rawah, Comanche Peak, Neota and Cache la Poudre [1]. The advantage of using this data is that it belongs to an area that has been untouched by human interaction [1]. Thus, the MLPs trained will be best suited to identify forest cover types for areas that have had minimum human interaction with them. The dataset has also been chosen because it presents a complex problem to solve as it contains substantial number of instances i.e. 581,012 and 21 input variables. This means that network could be trained well and following that tested well.

Previously, work has been done where Artificial Neural Network's results were compared against Discriminant analysis and ANN outperformed [1]. In our earlier work on the forest cover type, we went a step ahead by investigating which activation function performs the best for this dataset and concluded that Relu is most effective. Aspect, which is one of the features of the cover type data, was also encoded according to the method outlined in [3] and examined if training an ANN on it produces any differences compared to the one where simple scaling between 0 and 1 is used [3, 6]. Our results indicated that encoding Aspect improved the accuracy of the ANN in classifying which cover type the data belongs to. An accuracy of 52.09% was reached with Relu without aspect encoding and 56% with encoding. We extend the work done by implementing a deep neural network in this project and studying if it further improves our accuracy of classification.

Complex machine learning algorithms have been applied on forest cover type data since Jock [1] and have shown considerable improvement over single layer neural networks. Three are particularly noteworthy in this regard and worth discussing. Of these, a distributed SVM was first applied in 2008 [8] and it showed considerable improvement in identifying cover types that had fewer number of instances in the training set. Of the remaining deep learning models, Manifold Tangent Classifier, combines three ideas in building up a new classifier [9]. It does this by exploiting three "generic" prior hypothesis, that hold for most of the problems [9]. These hypotheses are semi-supervised learning, (unsupervised) manifold, manifold hypothesis for classification [9]. The second of the deep learning models used a learned-norm pooling as an activation function.

All these classifiers however, used a subset of the original dataset for training and testing purposes. The original multi-class classification problem was thus essentially changed into a binary classification one in the following way. The original data, which contained 7 cover types, was first divided into 7 new datasets denoted by DSi-581, where 'i' ranged from 1 to 7 for different forest cover types. '581' in the above datasets indicate that these datasets were chosen from the original dataset which contained 581,012 instances. Then, for each of the 7 DSi-581 datasets, additional data sets were made by randomly sampling the original dataset with varying amounts of observations. These were DSi-30, DSi-60 and DSi-300 where numbers at the end indicate 30,000, 60,000 and 300,000 instances in them. Therefore, DS2-60 would contain 60,000 random samples and instances would be classified as positives (cover type 2) or negatives (not cover type 2). For the SVM and deep learning experiments described above, DS2-581 was chosen as the data set for training and testing purposes.

We would however use the complete dataset as described in [1] and try to improve our results using a multi-layer perceptron. 21-variable dataset would be used along with Relu as the activation function and aspect encoding as these gave us the best results in the previous study. The loss function used in the experiment is cross-entropy with simple backpropagation as the learning algorithm. The MLP trained is a fully connected neural network with Stochastic Gradient Descent as the optimizer. The backpropagation algorithm requires momentum and learning rate as two initialization parameters and these were fixed at 0.5 and 0.001 respectively. The optimum number of layers and neurons is found by investigating.

2 Method

2.1 Dataset Selection

The dataset used, is the same as used in the previous experiment and is available from the UCI website. Since the aim of this study is to improve the previous results, complete dataset was used for multiclass classification. It was divided into two categories following the approach set out by Jock A. [1]. In this method, we note that one of the cover types, Cottonwood/ Willow has just 2747 instances out of the total 581,012 and take that into account. Randomly selecting a dataset could mean too few of this cover type are selected for training or testing and thus our MLP would not be able to predict to a fair degree of accuracy. Therefore, random samples are taken from the entire dataset ensuring that in the training dataset there are equal number of cover types. This number is chosen to be 1648 which is 60 percent of the least amount of cover type data available i.e. Cottonwood [1]. Around 20% of the data that is left for Cottonwood i.e. 540, was chosen for each of the cover types as the testing set. The total in the testing set is thus 540 times 7 i.e. 3780. Figure 1 below shows the distribution of the cover types in our dataset.

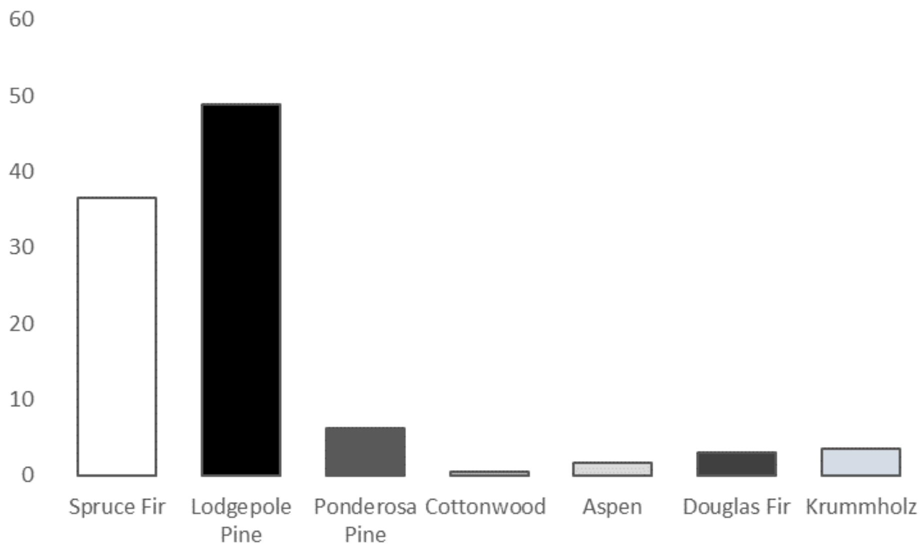

Figure 1. Distribution of forest cover types in the dataset

2.2 Multi-layer Perceptron Specification

The MLP was trained for 10,000 epochs with losses and accuracy printed out to the screen after every 1000 epochs. Since the aim was to find the optimal number of neurons in two and three layers, two models were made, each in a separate Python file. PyTorch provides a convenient way to build, train and test an MLP. A custom MLP with three layers is shown below:

```
class TwoLayerNet(torch.nn.module):
def __init__(self, n_input, n_hidden, n_output):
super(TwoLayerNet, self).__init__()
# Defining hidden layer output
Self.hidden1 = torch.nn.Linear(n_input, n_hidden)
self.hidden2 = torch.nn.Linear(n_hidden, n_hidden)
self.hidden3 = torch.nn.Linear(n_hidden, n_hidden)
# Defining output layer output
self.out = torch.nn.Linear(n_hidden, n_output)

def forward(self, x):
```

```
# Get hidden layer input
h1_input = self.hidden1(x)
h2_input = self.hidden2(h1_input)
h3_input = self.hidden3(h2_input)
# Define activation function for hidden layer
h1_output = F.Relu(h1_input)
h2_output = F.Relu(h2_input)
h3_output = F.Relu(h3_input)
# Get output layer output
Y_pred = self.out(h3_output)

return y_pred
```

As can be seen in the code there are three hidden layers. The 'i' in the hiddeni and hi_input indicates which of the three layer's we are referring to with 'i' ranging from 1 to 3. The lower the value of i, the nearer it is to the input. Each of the output of hidden layers is fed into the next hidden layer. Relu was used as the activation function for all the hidden layers as we found out that this it is the most-efficient for this dataset.

After every iteration, an error needs to be calculated to estimate how close to the actual results is the predicted result that our MLP produced. Mean-squared error is a common technique to improve neural network classifiers, but this study investigates the effects of cross-entropy. It has been shown that cross-entropy has significant, practical advantages over squared error [2]. The loss function can be defined as

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

It is this loss that is calculated at the output and propagated back into our neural network. The weights of the hidden layers are changed accordingly then.

2.3 Data Preprocessing and encoding

The training of the MLP starts with encoding and preprocessing of the data. Of the 54 variables, 4 variables excluded indicated which of the four areas the data belonged to. The information was presented using one-hot encoding and as it was a qualitative measure representing different regions it was not included to train the neural network. Elevation, slope, horizontal distance to hydrology, vertical distance to hydrology, horizontal distance to roadways, hill shade at 9am, hill shade at noon, hill shade at 3pm, horizontal distance to wildfire points and soil types were finally used. The soil types were originally represented using one-hot encoding too with 40 columns in the dataset. Each of the soil type had a different ELU code to distinguish it from others. However, each of the soil type could be categorized as belonging to one of the 8 geological zones and one of the 8 climatic zones using their ELU codes. The first of the 4 digit represents the climatic and the second digit the geological zone it belongs to. This means that 40 soil types could be categorized as 16 soil types but some of the 40 soil types had no data belonging to them. Therefore, less than 16 variables belonging to soil were input to the neural network. The forest cover types that needed to be identified were lodgepole pine (*Pinus contorta*), spruce/fir (*Picea engelmannii* and *Abies lasiocarpa*), ponderosa pine (*Pinus ponderosa*), Douglas-fir (*Psuedotsuga menziesii*), aspen (*Populus tremuloides*), cottonwood/willow (*Populus angustifolia, Populus deltoids, Salix bebianna, Salix amygdaloides*), and krummholz [1].

The Aspect was encoded using approach outlined by Tom [3] in "Decrypting GIS data". Any compass direction can be divided as if contributing fully, partially or none to one of the 4 major directions of North, South, East and West. All of the aspect degrees measurement are thus divided as if contributing 1, 0.5 or 0 to the major directions. Moreover, an aspect will not contribute to a single direction but to its adjacent directions as well though not as much. Therefore, we will end up with aspects values as inputs to four neurons representing different directions. A North would give 1 input to the North neuron but will also give 0.5 to East and West as these are adjacent directions and so on. Slope, and 8 other quantitative measures on the other hand were normalized between 0 and 1 by scaling values. Scaling the values of different independent input variables is highly required when their ranges are different [4].

2.4 Experimental Design

This time, one of the aims besides improving on the results of the ANN was to get important insights into the forest cover type data so that future data collected is more relevant and helps in training the neural network better. To do these two measures were performed. Firstly, most significant variables were found out using a decision tree classifier.

Secondly a pair plot was used to find out the distribution of the hill shades at various times with respect to the cover types.

Following this, original dataset was preprocessed, encoded and divided into training and testing sets after which the MLP was trained. 10,000 epochs were used to train the neural network. It was found that the optimum number of hidden neurons was 90 when we varied the number of hidden neurons from 6 onwards for a single-layer ANN in previous work. A similar approach was used in our deep learning experiment where number of hidden neurons was increased from 50 onwards for both two-layer and three-layer MLPs. Each consecutive run added 10 neurons to each of the layers. No hidden neurons were added when further addition stopped bringing any significant improvements in the test results. The learning rate was fixed at 0.001 while momentum was fixed at 0.5.

Confusion matrix was used printed for the test data to see how well our deep-neural network performed [5]. A confusion matrix will tell us how accurately our MLP has classified different forest cover types. Using it we can see how many false positives or false negatives there are for a cover type.

3 Result and Discussion

A three-layered MLP was first trained and the results are shown in table 1 below.

Table 1. Results for three-layered MLP

Number of hidden neurons	Final accuracy (Training)	Accuracy (Testing)
50	52.12	54.50
60	53.44	54.02
80	53.70	53.86
10	55.72	56.85
110	57.31	57.49
120	56.60	57.96
130	57.13	57.96
140	58.21	58.65
150	58.08	58.57
160	58.73	58.02

Table 1 shows that our deep neural network not only improved training accuracy but also testing accuracy. The greatest accuracy was achieved with 140 number of hidden neurons in all the three layers. The classification accuracy achieved with this setting was 58.65%. Next, a two-layered MLP was trained and the results received recorded in the table below.

Table 2. Results for two-layered MLP

Number of hidden neurons	Final accuracy (Training)	Accuracy (Testing)
50	54.80	54.79
60	54.19	54.42
70	54.92	54.23
80	54.95	54.44
90	56.83	55.87
100	57.24	55.85
110	57.73	56.35
130	57.06	55.85
150	57.43	55.98

The table 2 shows that the optimum number of neurons is 110 in both layers for the two-layered MLP. The corresponding training and testing classification accuracy were 57.73% and 56.35% respectively. Table 1 and table 2 show an interesting property of the MLPs. In both cases we were able to achieve greater accuracy than a single-layer neural network. Also, high accuracy was achieved early in the model with fewer number of hidden neurons which remained as we increased the number of hidden neurons. Figure 2 shows the confusion matrix for the most accurate model trained so far. i.e. three-layered MLP with 14 hidden neurons in each of the three layers. Figure 3 shows the results from our previous work.

```
Confusion matrix for testing:

 349   107     0     0    31     7    46
 199   174    11     3   113    38     2
   0     1   196   145    17   181     0
   0     0    72   426     0    42     0
  45    73    49     6   326    41     0
   0    15    70   106    36   313     0
  97     8     1     0     4     0   430
[torch.FloatTensor of size 7x7]
```

Figure 3. Optimal solution for single feed-forward neural network

```
Confusion matrix for testing:

 249   174     0     0    47     7    63
 155   210     4     8   122    41     0
   1     2   108   234    15   180     0
   0     0    57   427     6    50     0
  49   103    17    48   271    45     7
   3    24    82   110    14   307     0
  66    31     0     4     3     0   436
[torch.FloatTensor of size 7x7]
```

Figure 4. Optimal solution for single-layer ANN

The confusion matrix for the MLP shows that accuracy has been particularly improved for the Spruce and Aspen cover type. Krumholz life before is still the least misclassified cover type, though there has been an increase in misclassification this time. There has however been a reduction in false Spruce classification as Krummholz. Although there is a high accuracy for right willow being identified by the neural network, we can see from the figure that there is still a tendency of the MLP to confuse classification between Willow, Douglas-Fir and Ponderose Pine. This is consistent with Jock A.'s result and could primarily be due to the fact these three types are found near to each other [1]. Another similarity that is present is that Krummholz is primarily misclassified largely as Spruce and after that Lodgepole pine. All of these cover types are present in high elevation areas which may be the reason why Neural Network confused in classifying them [1].

Data analysis was also done as part of identifying how variables were correlated to each other and what were the most contributing variables. The figure below shows the relative significance of the 15 most significant variables.

```
1. Elevation      :     0.33419569645787883
2. R_Roadways     :     0.10575807271001152
3. Soil_7         :     0.10268226340670074
4. R_Fire_Points  :     0.10174208000276595
5. Hillshade_9am  :     0.07417995038329211
6. R_Hydrology    :     0.06193174424027837
7. Hillshade_Noon :     0.044742424909211105
8. Z_Hydrology    :     0.04344315697189546
9. Hillshade_3pm  :     0.035619205027961495
10. Soil_1          :   0.025359584340087563
11. Slope           :   0.02439415644694587
12. Aspect3         :   0.012384332085139123
13. Soil_8          :   0.008170382020042478
14. Soil_3          :   0.007729061724825724
15. Soil_11         :   0.005613998792989447
Contribution of 15 most important features: 0.9879461095200257
```

Figure 4. 15 most significant variables

The figure 5 shows that elevation plays a crucial role in deciding the cover type. Distance to roadways is another important variable which is interesting to note. We can see that all the hill shades form a good proportion of the 15 most significant variables. Therefore, to further investigate the effect of hill shades, pair plots were plotted using Python's seaborn library. The results are shown in the figure below.

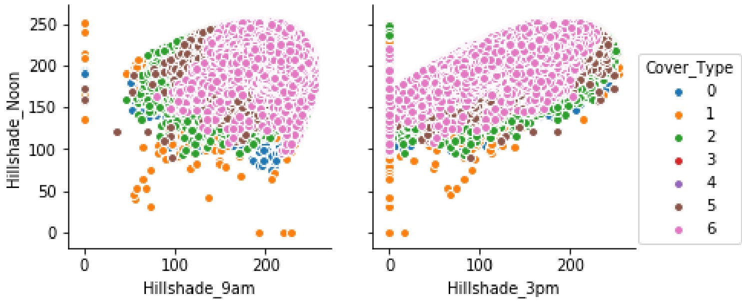

Figure 5. Elevation for different forest cover types

The results show that cover type 7 gets the most amount of hill shade at all the 3 times. Followed by that is cover type 3 though cover type 5 can be seen near it as well. From this we conclude that in the future more quantitative measures of the amount of hill shade received would improve the training of the neural network. For instance, the average amount of sunlight received by an area would be more descriptive.

4 Conclusions and future work

We trained an MLP that was able to predict forest cover types to an even higher accuracy than our previously trained ANN. Relu as an activation function was chosen along with cross-entropy from the previous study as they gave us the best result. Highest accuracy was achieved with a three-layered Perceptron with 145 hidden neurons in each layer. An astounding accuracy of 58% was achieved for the test set. Finally, a data analysis was done where we found out the 15 most important variables that contributed to 0.98% of the information using Gini-Index. From the most significant variables, hillshades were observed and their data analyzed using pairplots.

Also, important improvements could still be made to the neural network to improve the classification accuracy. We could prune the network in each epoch to take out the neurons that do not contribute to the results. Also, improved quantitative measures of important variables such as sunlight received by trees in a certain area could help us train the network better.

References

[1] J. Blackard and D. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables", Computers and Electronics in Agriculture, vol. 24, no. 3, pp. 131-151, 1999.

[2] D. Kline and V. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers", Neural Computing and Applications, vol. 14, no. 4, pp. 310-318, 2005.

[3] R. Bustos and T. Gedeon, "Decrypting Neural Network Data: A GIS case study", Artificial Neural Nets and Genetic Algorithms, pp. 231-234, 1995.

[4] C. Marzban and G. Stumpf, "A Neural Network for Tornado Prediction Based on Doppler Radar-Derived Attributes", Journal of Applied Meteorology, vol. 35, no. 5, pp. 617-626, 1996.

[5] P. Hardin and J. Shumway, "Statistical Significance and Normalized Confusion Matrices", Photogrammetric engineering and remote sensing, vol. 63, no. 6, pp. 735-739, 1997.

[6] L. Milne, T. Gedeon and A. Skidmore, "Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood.", Proceedings Australian Conference on Neural Networks, pp. 160-163, 1995.

[7] B. Guan and G. Gertner, "Modeling individual tree survival probability with a random optimization procedure: an artificial neural network approach.", AI Applications, 1995.

[8] M. Trebar and N. Steele, "Application of distributed SVM architectures in classifying forest data cover types", Computers and Electronics in Agriculture, vol. 63, no. 2, pp. 119-130, 2008.

[9] C. Gulcehre, K. Cho, R. Pascanu and Y. Bengio, "Learned-Norm Pooling for Deep Feedforward and Recurrent Neural Networks", Joint European Conference on Machine Learning and Knowledge Discovery in Databases., pp. 530-546, 2014.