Landform Classification from Satellite Data: Application of Deep Neural Network

Longzhen Cui, u6693360@anu.edu.au

Abstract. Geographical data can provide many vital features which are necessary for generating a detailed map. However there still has a lot of features which are not included in the GIS data, and people have to predicate these value through certain methods. Nowadays the neural network (NN) has become one of the most common choice for such a machine learning task. In this paper, we used the NN approach to predicate the landform of a certain area based on the given GIS data. We firstly examined a previous research which used a fully connected neural network with a single hidden layer and found the result was not satisfactory enough. Then we replaced the structure with a deep neural network and found an optimal setting of hidden layers. The result shows that the accuracy of prediction is improved after applying multiple hidden layers.

Keywords: Neural Network, Deep Learning, Data Mining, Geographic Information System

1 Introduction

The bush fire occurred last year has greatly impact our lives. Many fire hazard forecast application were set up during that fire period. Such applications rely on detailed geographical information to make forecasts. However the accessible data usually would not contain the desired information directly. Thus we need to predict its value based on given information.

In this paper we firstly trained a neural network, which is suggested by previous research (Milne, 1995), to determine the landform based on the given GIS data. The data is pre-processed for our topic. The network is validated by the test set and the result is provided, and the result shows that it has much potential for improvements. We supposed this problem could be more complicated than a linear problem (Tien Bui et al., 2020). Thus we changed the original structure with a deep neural network. We found an optimal setting of hidden layers and test the model again, the result shows that the accuracy of prediction is improved.

2 Method

2.1 GIS data and Pre-processing

The geographical data we are using is from an area in the Nullica State Forest on the south coast of New South Wales. The site is around 20 by 10 km, is broken up into a grid of 179831 pixels, 30 by 30 m in size. The data is collected by satellites and has been pre-processed. The file contains 190 sampled plots with detailed information (Milne, 1995). The top line of the file explains the feature order of samples, followed by 190 samples. Each sample occupies a line with the information below:

Feature	Explanation	Feature	Explanation
PL AS SA CA	aspect	T1 T2 T3 T4 T5 T6 T7	Landsat TM bands
AL	altitude	SC DS WD WS RF	Landform: scrub,dry sclerophyll, wet-dry, sclerophyll, wet sclerophyll and rainforest. Value of 90 indicates the real type, and 10 for false.
TP	topographic position		
SL	slope		
GE	geology		
RA	rainfall		
TE	temperature		

The first column of features (PL) is removed in our program, as it is only a index and provide no useful information. The value of 5 potential landform is re-scaled for easier using: 1 means this type is the correct label and 0 means not. 190 samples are divided into two sets: 150 for the training set, the other 40 for the test set.

2.2 Fully Connected Neural Network model

A previous research (Milne, 1995) provided a fully connected neural network model which have 16 input features, one single hidden layer with 10 neurons and an output layer of length 5 gives the probability sample being the corresponding land type. There were some details the author did not mention, so we set these values with appropriate choices by ourselves.

The network use ReLU as activation function for all layers, which is given below:

$$f_{(x)} = \max(x, 0)$$

where x denotes the input value. The ReLU ensured that the output would never below than 0.

And softmax function is applied to the output layer to ensure the result in the desired format:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where z is the input vector, K is the number of classes and σ is the softmax.

The cross-entropy loss is selected for training and testing, as this fits classification problems:

$$\mathrm{L} = -\frac{1}{N}\sum_{n=1}^{N} T\ln(Y) + (1-T)\ln(1-Y)$$

where N represent the number of training samples, T and Y denotes the actual and predicted class of outputs, respectively.

Finally, we choose Adam algorithm as optimizer (Kingma, 2015).

2.3 Deep Neural Network model

A single-layer neural network can only be used to represent linearly separable functions. And a dataset with one or more convex margin would need more layers to successfully classify it.

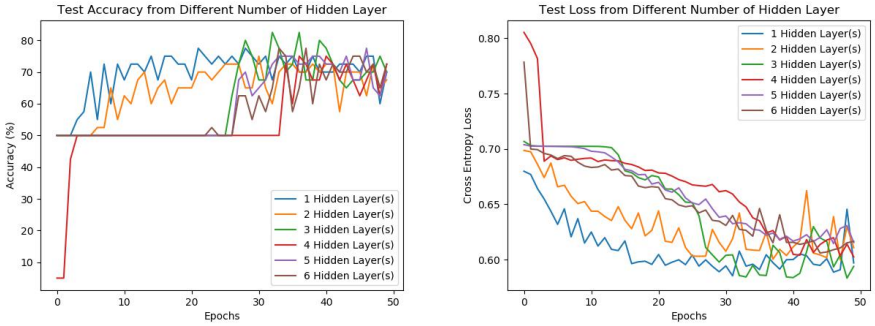
There are several methods to determine the optimal number of hidden layers and the number of neurons in each hidden layer (Stathakis, 2009), and we set our network through a trial-and-error approach on the given dataset.

Other settings of the original network suit the deep neural network well, so they remain unchanged.

3 Network training and Result

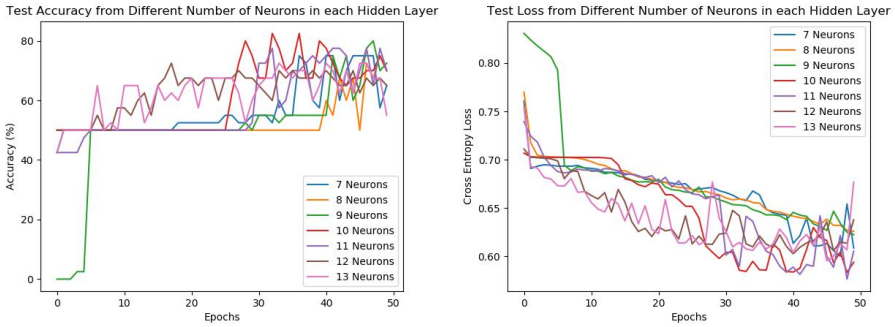
We use the cross entropy loss as loss function as we are working on a classification problem. The Adam optimizer is chosen for back propagation as it performs better among other in practice. We calculate accuracy and average test loss to show the performance of our network. We have to find out the most suitable number of hidden layers and neurons through experiments.

Firstly we would find the number of layers. We keep the number of neuron in the hidden layer and vary the number of hidden layers to see if the performance would be improved :



Two figures above shows the test accuracy and test loss obtained from different numbers of hidden layers. It could be observed that most performances become relative steady during the 20th loop to 30th loop. The final test accuracy is around 70%. A number of 3 hidden layers seems to have the best performance based on two figures.

Next, we set the number of hidden layer to 3, and change the number of neurons in each hidden layer to see the variation of the performance. We expect all hidden layers have same number of nodes:



The figures above illustrate the performance when using different number of neurons. It seems that the performance become optimal when using 10 or 11 nodes in hidden layers.

Through many repeated experiments we found the DNN can usually achieve an accuracy of 75% while the original network is always around 65%. We assumed that the performance of our model is improved

4 Discussion and Conclusion

The result appears to have great fluctuation between each experiments, which could be caused by the limited size of training set, as duplicated training on a small size dataset may lead to overfitting. We expect the performance become more steady on a larger dataset.

Additional layers enables the model to draw complex decision boundaries. But an excess of hidden layer and neurons would also lead to overfitting problem. A complicated structure would also consume too much computing resources. Thus, it is vital to carefully choose appropriate number of hidden layers and the number of neurons.

For the GIS dataset, we suppose some input features may not have strong relations between each other, hence some nodes inside hidden layers could be redundant and could be pruned. A genetic algorithm may suit this task.

References

1. Kingma DP, Ba J. (2015). Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego
2. Milne, L. K., Gedeon, T. D., & Skidmore, A. K. (1995). Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. training, 109(81), 0.
3. Stathakis, D. (2009). How many hidden layers and nodes?. International Journal Of Remote Sensing, 30(8), 2133-2147. doi: 10.1080/01431160802549278
4. Tien Bui, D., Hoang, N., Martínez-Álvarez, F., Ngo, P., Hoa, P., & Pham, T. et al. (2020). A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. Science Of The Total Environment, 701, 134413. doi: 10.1016/j.scitotenv.2019.134413