

CLASSIFYING FOREST SUPRA-TYPE FROM AUGMENTED SATELLITE DATA: COMPARING SIMPLE NEURAL NETWORK, BIMODAL AND CASPER TECHNIQUES

Jiawen He

Research School of Computer Science, Australian National University

Abstract

The Geographic Information System data was collected from an area in the Nullica State Forest on the south coast of New South Wales. Our task is to classify the forest supra-type among five options based on geographical features mainly collected by satellite.

We apply three kinds of neural networks, a two-layer neural network, another with the CasPer technique, and the third applying Bimodal outlier remover on basis of CasPer, to give a classification of Forest supra-type based on GIS dataset. The neural network result advantages over both techniques, with significantly higher correctness rate. A detailed analysis and comparison of performance on three methods are discussed in this paper.

Keywords

Neural Network; CasPer; Bimodal; GIS dataset

1 Data and Preprocessing

Geographical data, especially information with detailed terrain type has become intensely important for various applications and research work. With structured and known satellite data, more detailed information can be generated. The data was integrated from satellite imagery, soil maps and aerial photographs [1].

Among the spots explored, this dataset gives us 190 sets of features and forest supra-type classifications. In the raw data, there are 16 values for each plot, including aspect, sin and cos of aspect, altitude, topographic position, slope degree, geology descriptor, rainfall, temperature, and Landsat TM bands 1 to 7.

Some of the features have already been encoded by the geographers who provided the data. We have to decrypt the data and restructure the inputs and outputs. The preprocessing procedures performed on each feature are described as below:

- Aspect: It's originally represented in degrees which has a problem of continuity. To deal with this, the aspect is encoded as four inputs, in which each represents a major direction and spread adjacent directions accordingly [2].
- Sin and Cos of aspect: Redundant with Aspect. Not used in our classification.
- Altitude: Normalized to 0-1.
- Topographic position: Normalized to 0-1.
- Slope degree: Normalize to 0-1.
- Geology descriptor: Unknown meaning in values. Since the values appear to be categorical and has no particular distribution, we encode it as four inputs, distinguishing between the popular types and the rare ones [2].
- Rainfall: Normalize to 0-1.
- Temperature: Normalize to 0-1.
- Landsat bands 1-7: Normalize to 0-1.

The output categories are encoded by 4 units, in the purpose of avoiding sparse output vectors by equilateral coding. All units have some activation on each pattern, and the maximum distance between vectors is maintained. The final category can be computed through Euclidean distance between predicted values and the five different types [2].

As the number of samples is very small (only 190) in this dataset, 5 cross validation is used in order to avoid bad splitting of data.

2 Method

We perform three different method for cross validation. This is initially a classification problem, in which forest supertype is given as target, however, by equilateral encoding discussed above, it is modified into a regression problem.

2.1 Simple Neural Network

A 2-layer neural network regression model is performed on GIS dataset. The parameters are defined as follows:

- Input size: 20; hidden size: 12; output size: 4
- Number of epochs: 1000 (number 500 and 1500 have also been tried, in balancing learning and over-fitting, 1000 is set as a preferred choice.)
- Loss function: binary cross entropy. In PyTorch, we use BCEWithLogitsLoss, since this includes the activation function (Sigmoid) of last layer, and ensures numerical stability by combining the operations into one single layer.
- Optimizer: Adaptive Moment Estimation (Adam in PyTorch). This avoids the problem of getting stuck in local minima in stochastic gradient descent by adapting learning rate [3].

2.2 CasPer

In order to resolve the two problems in Cascade Correlation (CasCor) [4], one being large networks caused by weight freezing, the other being saturation problem led by use of correlation measure, the Cascade Architecture with Progressive RPROP (CasPer) technique was proposed [5]. CasPer algorithm is a very powerful training method that takes advantages of the weight freezing idea from CasCor, and has been shown to produce networks with fewer hidden neurons than CasCor, while also improving the resulting network generalization, especially with regression task [6].

In PyTorch implementation, new linear weights are created each time a new hidden neuron is added, and appended to a big list which includes all weights Linear instances. New RProp optimizers are created and associated with new parameters as new hidden neurons are installed. Whether or not to install a new neuron is based on the decrease rate of RMS values.

Hyper parameters in this method are set generally following the standard values. Tests have been made with 12 and 14 hidden neurons.

2.3 CasPer & Bimodal

The GIS is a very small dataset, with only 190 pieces of data, in which outliers may affect training model significantly. Bimodal distribution removal technique is a robust method to remove outliers, thus improving generalization performance [7]. After a number of epochs on training, some data are recognized as outliers and removed in this technique.

The implementation structure is basically the same as the CasPer technique discussed above, with an addition of loss value analysis and outlier removal each time a new hidden neuron is added. In our case (GIS), the variance threshold and halting condition are set at relatively low values, since the MSE values are small.

3 Results and Discussion

During each run, we take the average correctness rate among each five cross-validation process, the best result over 100 runs are displayed in the table following:

	loss value on training data	correctness on training data	loss value on testing data	correctness on testing data
Vanilla Neural Network	0.0075	93.1579%	0.0371	63.1579%
CasPer (12 neurons)	0.0215	69.0789%	0.0277	65.2632%
CasPer (14 neurons)	0.0215	70.9211%	0.0270	65.2632%
CasPer & Bimodal (12 neurons)	0.0215	68.6842%	0.0270	63.6842%
CasPer & Bimodal (14 neurons)	0.0214	69.8684%	0.0284	64.2105%

Table 3.1

From the performance data in table 3.1, it is obvious that CasPer technique performs better than Vanilla Neural Network on testing data. The correctness on training data drops, however, indicating that CasPer is better at generalisation and learning the general pattern from given knowledge. We can also see that adding more hidden neurons has slight impact on generalisation, but not significantly.

After Bimodal technique is imported into CasPer, the performance is slightly worse, by removing outliers from dataset itself. Possible reason is that the program has removed too much ‘outliers’, and those features that would be eventually learnt in later epochs are removed in early stage.

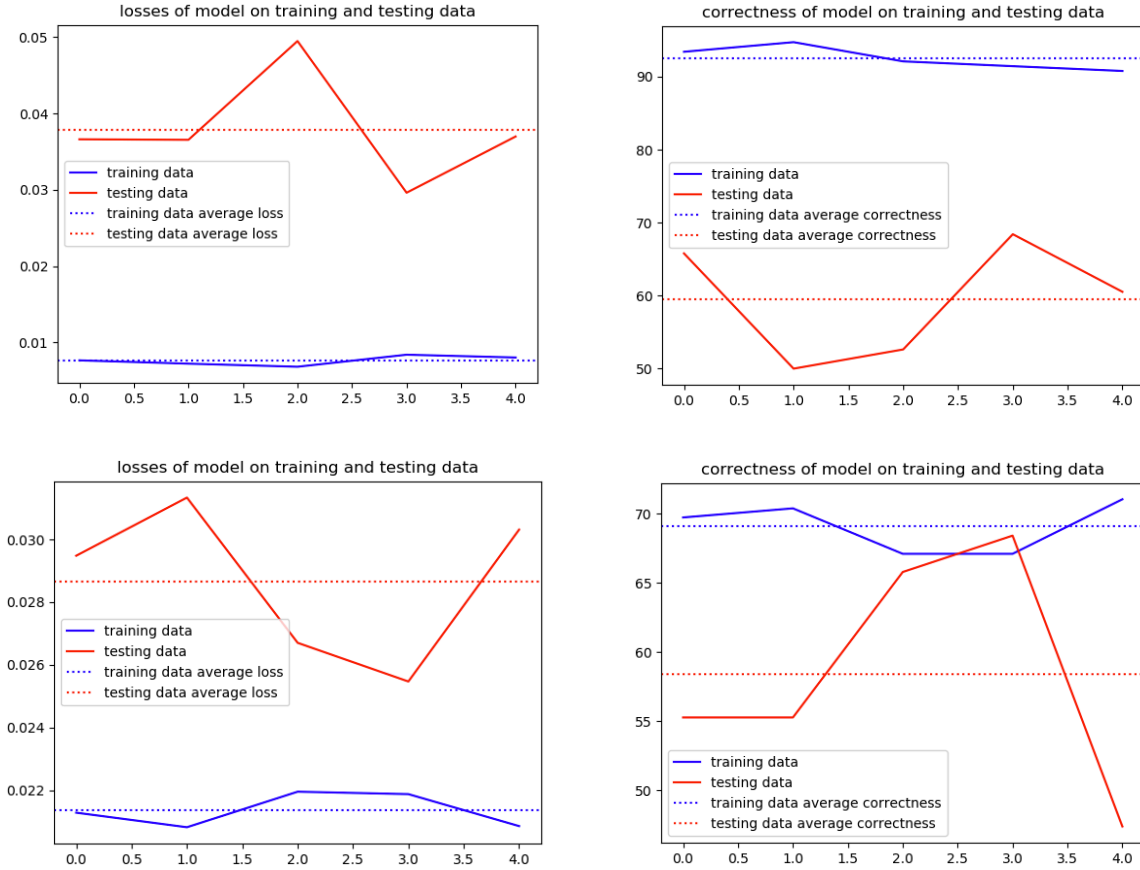


Figure 3.1

The plots in Figure 3.1 depicts the performance on each cross validation set in a typical run, with the top two graph as simple neural network, and the bottom ones from CasPer algorithm. The scale level of the problem is clearly shown. We can find some reasons of bad performance from the above graph: obviously, the distribution across five validation sets are not even, leading to great changes of losses and correctness in different round.

In comparison with techniques applied on the same dataset, the neural network described in [2] has lower performance than all of our models. However, the techniques used in [1] outperform our implementation in some ways. The following table shows their result:

Technique	Training correctness	Testing correctness
Decision tree	57.4%	65.7%
Maximum likelihood	65.3%	60%
Neural network	52.6%	65.7%

Table 3.2

From Table 3.2, the performance is sometimes slightly better than ours. However, they are at the same scale, and are not significantly different in the context of such small dataset. The result from Table 3.2 is based on the classification encoding of output, which is shown to be possibly fitting better on our problem.

4 Conclusion and Future Work

Experiments conducted have shown that both CasPer and Bimodal techniques are useful in improving the learning process. However, no significant progress has been observed. There might be a few reasons, including bad distribution of training and testing set, not being able to choose the right hyper parameters, not having enough data and the dataset not fitting well enough on model. Though CasPer overcomes the generalisation problem on regression model, it may still perform better on classification problems.

There is a lot for use to experiment on. The most important issue is to adopt new method of splitting training and testing data. This ensures that a whole picture of features is to be learned by models. Then, after double checking the correctness of my implementation, much more time will be spent on adjusting parameters, such as the threshold of variance in Bimodal technique. Apart from that, more models and techniques are to be tested on GIS dataset, and more settings of hyper parameters can be tried. More importantly, our learning model may be limited by the availability of data. More data are to be collected for better training and prediction.

Reference:

1. Milne, L. K., Gedeon, T. D., & Skidmore, A. K. Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. (1995)
2. Bustos, RA and Gedeon, T.D. Decrypting Neural Network Data: A GIS Case Study. (1995)
3. Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. (2014)
4. Fahlman, S.E., & Lebiere, C. The Cascade-Correlation Learning Architecture. In Advances in Neuron Information Processing II, Touretzky, Ed. San Mateo, CA: Morgan Kauffman, pp.524-532. (1990)
5. Treadgold, N.K. & Gedeon, T.D. A Cascade Network Algorithm Employing Progressive RPROP. (1997)
6. Treadgold, N.K. & Gedeon, T.D. Extending CasPer: A Regression Survey. Int. Conf. On Neural Information Processing. (1997)
7. Slade, P. & Gedeon, T.D. Bimodal Distribution Removal. (1993)