Forest Supra-Type Classification from Augmented Satellite Data: Combining Genetic Algorithm with Feedforward Neural Network and Casper Technique

Jiawen He
Research School of Computer Science, Australian National University, Canberra Australia
u6227591@anu.edu.au

Abstract. This paper investigates the use of genetic algorithm to reach optimal feature selection and hyper parameters for a neural network and Casper cascaded network upon the GIS dataset. The Geographic Information System data (GIS) describes augmented satellite data along with the forest supra-types. We want to obtain the supra-types automatically, instead of checking in person. Therefore, our task on this dataset is to develop a classifier to predict the forest supra-types accurately based on known information. Feedforward Neural Network and evolutionary algorithms have been typical bio-inspired computing techniques evolving in recent years. Casper has been proposed to be a strong derivation of cascaded neural network architecture. This paper describes research work which aims to compare performance of traditional neural network and Casper techniques on GIS data set. Additionally, genetic algorithm is utilized in selecting features and optimal hyper-parameters for both network techniques. Through this optimization, nearly best outcomes of both techniques are computed and analyzed in this paper. Results show that both feedforward neural network and Casper architecture give reasonable but limited prediction on forest supra-types, however, the former advantages over Casper with slight difference. This combination of genetic algorithm and neural network is also proved to be useful.

Keywords: Neural Networks; Casper; GIS dataset; Genetic Algorithm; Evolutionary Algorithms

1 Introduction

Geographical data, especially information with detailed terrain type has become intensely important for various applications and research work. Collection of data like forest supra-type can be very expensive and time-consuming. The condition of geographical types can change even in short time. Therefore, in order to save time and energy on more important analysis, we need to predict such information automatically from satellite data, as accurately as possible. We use a small set of data that are integrated from satellite imagery, soil maps and aerial photographs [1]. Among the spots explored, this dataset gives us sets of features and forest supra-type classifications. Our goal is to train a model that can classify the forest supra-type based on known geographical data automatically.

Related research by Milne, Gedeon and Skidmore has shown that there is no significant difference in performance between decision tree and neural network classifier, and they work slightly better than maximum likelihood classifier [1]. Hence in this paper, we mainly discuss the traditional neural network approach and an extended cascade neural network called Casper. We train the hyper-model (genetic algorithm) twice, one for finding the optimal feature selection and hyper-parameters on traditional neural network and the other on Casper technique. We evaluate the accuracy rate of prediction in both model and feasibility of genetic algorithm to find a best model in this way.

Casper is a feedforward neural network that adds a single neuron at a time during training. This cascading approach is inspired from the Cascor technique. Cascor has been shown very successful [2], but some shortages exist. One drawback is that the network can be very large due to weight freezing feature [3], while the other is the saturation problem led by use of correlation measure [4, 5]. In order to obtain better generalization, Casper was proposed. This technique utilizes Progressive RPROP optimization method and takes advantages of the weight freezing idea from Cascor. It has been shown to produce networks with fewer hidden neurons than Cascor, while also improving the generalization, especially with regression task [6].

On deciding the model, our question left is how to choose appropriate features and hyper parameters. We use genetic algorithm to find an optimal (or approximately optimal) solution. Genetic algorithm has been shown to be a very successful optimization method that can keep improving and jump out of local maxima. In both standard neural network and Casper approach, we train them with different parameters based on population in the genetic algorithm. The feature selection and hyper parameters settings are improved in genetic algorithm based on their testing accuracy rate. This also ensures we are comparing best possible outcomes from two techniques, regardless of bad feature selection or hyper parameters.

This paper gives a detailed explanation of the techniques, parameters and evaluation methodologies used in section 2. In section 3, the results of benchmarks are presented, analyzed and compared with related research work. Finally, in section 4, some conclusion and future work are summarized.

2 Method

The investigation of the prediction model can be divided into two identical parts, one for traditional neural network and the other for Casper. In each part, a genetic algorithm leads the whole training process and evaluates the accuracy level based on hundreds of runs of neural networks. The algorithm learns a better model settings and improve in each

generation. Details of data preprocessing, genetic algorithm, traditional neural network, Casper and evaluation to be described in this section.

2.1 Data preprocessing

Overall, there are 190 instances of data in the GIS data set. In the raw data, each instance has 16 features and last 5 labels as forest supra-types. The features include aspect, sine and cosine of aspect, altitude, topographic position, slope degree, geology descriptor, rainfall, temperature, and Landsat TM bands 1 to 7. The output labels are scrub, dry sclerophyll, wet/dry sclerophyll, wet sclerophyll and rainforest.

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. We ignore the sine and cosine values of aspect since they are redundant, and normalize the altitude, topographic position, slope degree, rainfall, temperature and all Landsat bands into range [0, 1]. The encoding of aspect, geology descriptor and output categories follow the principals as what Bustos and Gedeon described in [7]. Details of the encodings of these features are described in the tables below.

Table 1. Encoding of feature aspect. Since aspect indicates directions, the original data from 0 to 80 lose the information of relationship between one another. Hence, we use 4 new variables to encode this circular input feature.

Aspect	Direction	A1	A2	A3	A4
0	**Flat**	0	0	0	0
10	**N**	1	0.5	0	0.5
20	**NE**	1	1	0	0
30	**E**	0.5	1	0.5	0
40	**SE**	0	1	1	0
50	**S**	0	0.5	1	0.5
60	**SW**	0	0	1	1
70	**W**	0.5	0	0.5	1
80	**NW**	1	0	0	1

Table 2. Encoding of geology descriptor. We have no idea what this feature is about, but what we can do is to highlight the features based on statistics. The geology descriptor appears to be a nominal value and three of the types are particularly more common than the others. Bustos and Gedeon propose to encode them as another 4 inputs, distinguishing between the popular types and the rare ones.

Value	G1	G2	G3	G4
10	0.9	0.1	0.1	0.1
20	0.9	0.1	0.1	0.1
30	0.9	0.1	0.1	0.1
40	0.9	0.1	0.1	0.1
50	0.1	0.9	0.1	0.1
60	0.9	0.1	0.1	0.1
70	0.1	0.1	0.9	0.1
80	0.9	0.1	0.1	0.1
90	0.1	0.1	0.1	0.9

Table 3. Encoding of outputs (forest supra-types). The distribution of output is very sparse, which can lead to difficult learning. This is avoided by equilateral coding [8], so the 5 possibilities are represented by 4 units. The final category can be retrieved from calculating nearest neighbour.

Category	Unit1	Unit2	Unit3	Unit4
Scrub	0.1838	0.3174	0.3709	0.4
Dry sclerophyll	0.8162	0.3174	0.3709	0.4
Wet-dry sclerophyll	0.5	0.8651	0.3709	0.4
Wet sclerophyll	0.5	0.5	0.8872	0.4
Rain forest	0.5	0.5	0.5	0.9

After the preprocessing work is finished, there are 20 features in total, and 4 output labels. All values in the preprocessed data set are in range [0, 1]. In this way of encoding outputs, we transform this classification into a regression problem. In our whole training procedure, we perform an optimization in feature selection, so the input features are adjusted frequently in order to get a better result of prediction, usually less than 20 in the program.

2.2 Genetic algorithm (GA)

The genetic algorithm can be seen as 5 phases: initialization, fitness function, selection, crossover and mutation. It is an iteration process, and the last 4 stages are calculated for a large number of generations until it reaches an upper bound set by us. Basically, there is a fixed number of population representing hyper parameters to train the classifier and feature selection to feed into the classifier. After the population is initialized randomly, neural networks structured by the settings translated from each DNA in population are trained and tested. According to a probability calculated from test accuracy (to be described in section 2.5), some of the population survive while the others don't. The accuracy value evaluated from testing result are subtracted by the smallest one among population to form this probability of selection, so that the best models are more likely to be selected. After that, a crossover and mutation stage is performed and the next generation begins. The selection phase enables better model to be reserved and worse model to die out gradually. While the crossover and mutation phases give the population a possibility to jump into a different direction and out of local maxima. These phases form a big picture of this experiment, where the computation of fitness function involves feeding the features and hyper parameters selected into the traditional neural network and Casper network.

The DNA representation of traditional neural network and Casper are slightly different due to their structures. In this paper, the translation process of DNA is going to be defined along with their techniques in section 2.3 and 2.4. To ensure consistent relationship between chromosome and mutated ones, gray encoding is used.

We use a population size of 10 and generation number of 20. Due to the large complexity in neural networks, the number has to be small to be feasible.

2.3 Traditional Neural Network

A back-propagation neural network with one hidden neuron is trained on the GIS data set. We use mean squared error as loss function, as this is compatible with regression tasks where outputs are in range [0, 1]. Optimization are done by adaptive moment estimation, a state of the art optimization technique, that can take several factors into account and require little tuning [9]. We use Sigmoid function as the activation function in both layers. (Fig 1)

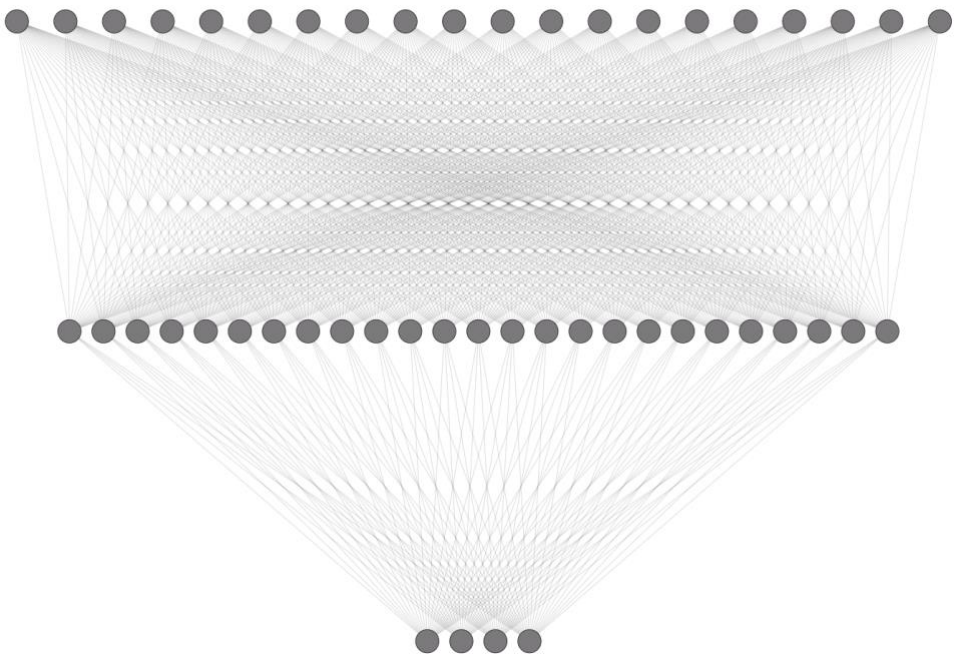

Fig. 1. An example traditional neural network with 20 input neurons, 25 hidden neurons, 4 output neurons. In our research, the number of neurons are not fixed and may be less or more than what is in this example graph.

The number of input neurons ranges from 1 to 20, depending on the feature selection settings, and it is fixed that there are 4 output neurons. Apart from feature selection, the hyper parameters that need to be determined are number of hidden neurons, number of epochs and learning rate. Their range, binary representation and conversion from binary representation to real values are described in table 4 below. The DNA size is 44 in traditional neural network.

Table 4. the range, number of bits, conversion from binary to real meaningful values of hyper parameters in neural network

Hidden size	[1,64]	6 bits	binary+1
Number of epochs	[1,512]	8 bits	binary+1
Learning rate	[0.0001,0.1024]	10 bits	(binary+1)/10000

2.4 Casper technique

The Cascade Architecture with Progressive RPROP (Casper) technique was proposed in order to overcome the drawbacks in Cascor technique [10]. In Casper technique, the neural network starts with only one input layer and one output layer, and new neurons are added one at a time, representing a new layer each time.

A Casper architecture installs a new neuron when its RMS error has decreased by at least 1% compared to the previous RMS value. The number of epochs of each newly-installed neuron need to be at least 15 + P*N, where N is the number of currently installed neurons, and P is a parameter set by us. Here the p value is a part of the hyper parameters to be trained by genetic algorithm.

When a new neuron is installed, it adds connection to all the previous neurons and output neurons. The neurons in this topology is divided into three parts, each with own learning rate L1, L2 and L3. The first part includes all the weights connecting from previous neurons to the new neuron. The second part consists of all weights connecting from the new neuron to the output neurons. While the third part is for all the remaining weights, that is, all weights not related to the new neuron. The value of L1 is much larger than L2, and L3 is the smallest. We use the RPROP technique so L1, L2 and L3 are just initial learning rates. They are parts of the hyper parameters trained by genetic algorithm as well. The topology is depicted in figure 2.

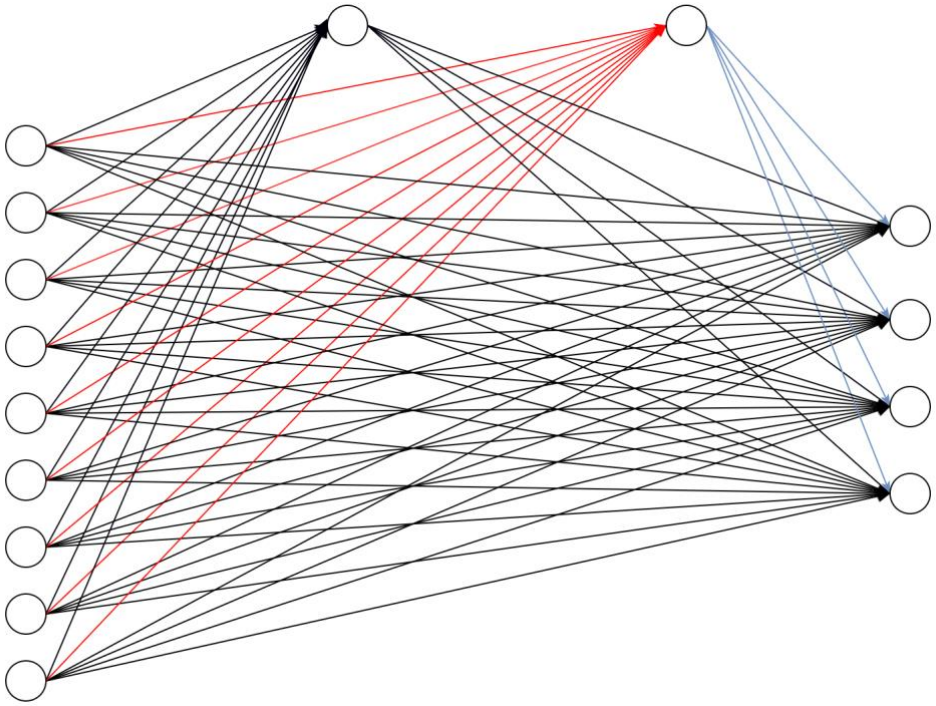

Fig. 2. An example Casper neural network with 9 input neurons and 4 output neurons. This shows when the second hidden neuron has just been inserted. Red weights have large initial learning rate L1, blue weights have smaller learning rate L2 and black weights have smallest learning rate L3. Note that the number of input neurons may change in our case.

To be consistent with traditional neural network described above, mean squared error loss function and Sigmoid activation function are still used.

In all, the number of total hidden neurons to be installed, p value and three initial learning rates are represented in the DNA binary representation as described in table 5 below. The DNA size is 42 in Casper neural network.

Table 5. the range, number of bits, conversion from binary to real meaningful values of hyper parameters in Casper

Hidden neurons	[1,16]	4 bits	binary+1
P value	[1,8]	3 bits	binary+1
Learning rate 1	[0.1,0.8]	3 bits	(binary+1)/10
Learning rate 2	[0.002,0.006]	2 bits	(binary+1)*0.002
Learning rate 3	[0.0001,0.1024]	10 bits	(binary+1)/10000

2.5 Evaluation

The evaluation method we use in this research is very simple. Initially, we use both mean squared error and accuracy rate defined in formula 1. Before we calculate the accuracy rate, some operations are needed to convert numerical results to categories using nearest neighbor approach. Based on several training and testing, we find that they are very much related with one another, in a way of negative correlation. The formula is defined below:

accuracy rate = number of correct classification results / number of instances tested (or trained) (1)

We compare the accuracy rate on training and testing set to check overfitting. But in most cases, we mainly look at the accuracy rate on test data, as this reflects most on the generalization quality of the classifier.

More importantly, we use 5-fold cross validation here. The reason is that there are only 190 patterns in the GIS data set, which is normally insufficient for training model. Random separation may lead to bad distribution on either training or testing data set. Therefore, we split the data into 5 proportions, use only one from the five parts for testing and train the classifier model through 80% of the data set each time. After the 5 iterations of training and testing, the average test accuracy value is taken to evaluate the overall performance of the given model.

3 Results and Discussion

3.1 outcome and analysis

During each run, we take the average among the 5-fold cross validation processes. Detailed data throughout the genetic algorithm training process are described in figure 3 and figure 4. For each generation, the point indicates the best performance on testing dataset among the population.

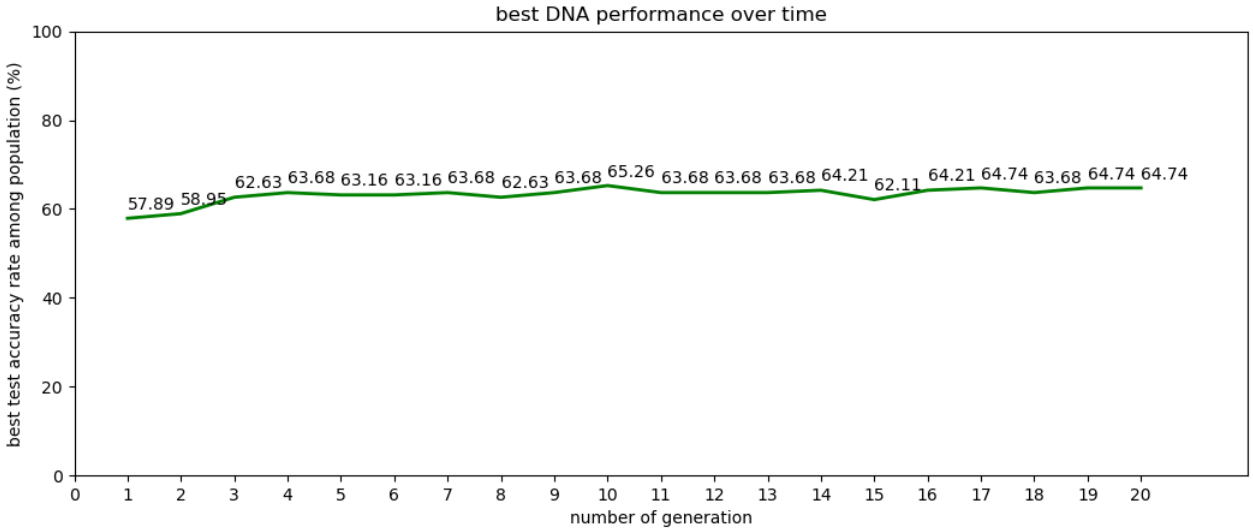

Fig. 3. Best accuracy rate among population in each generation (traditional neural network)

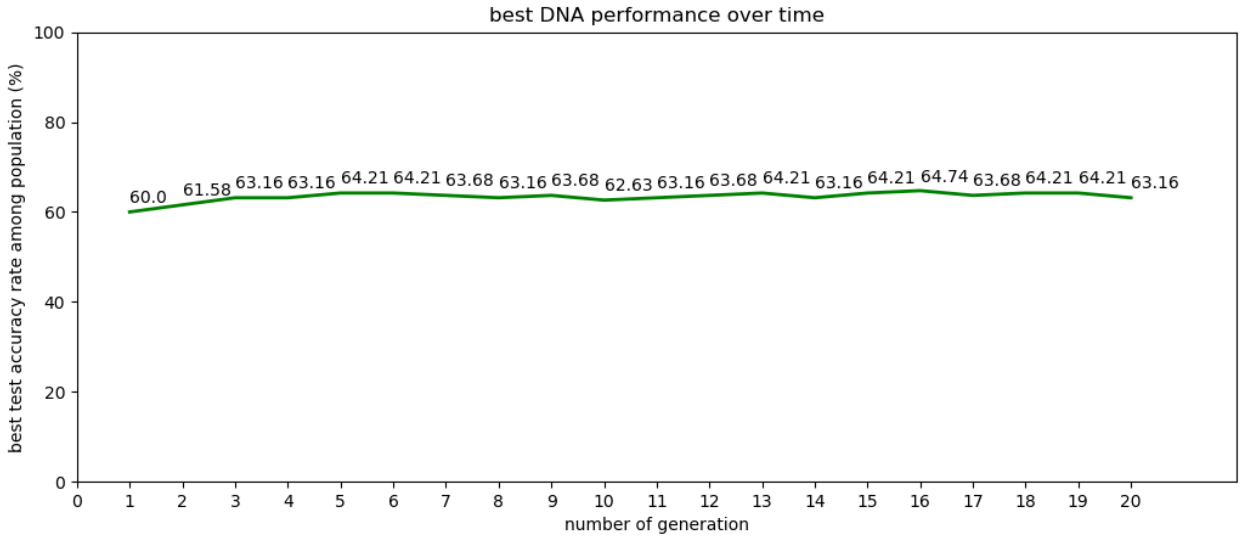

Fig. 4. Best accuracy rate among population in each generation (Casper neural network)

As we can observe, the resulting accuracy rate on traditional neural network is around 65% at highest, while the Casper's best performance is around 62%. It is obvious that traditional fixed size neural network performs better than the cascaded neural network Casper in predicting geographic forest supra-types.

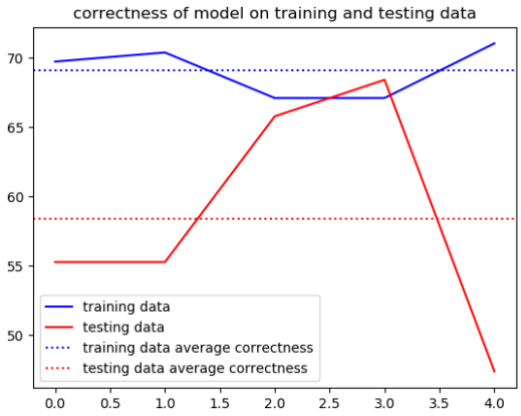

Fig. 5. A typical case of performance in each run of 5-fold cross validation

From figure 5, we know that the correctness of training and testing data in each cross-validation process vary a lot. This means that the data divided in each round are not even at all. It is hard to avoid such situations when only a small data set is available. This may be another reason of bad performance.

We also did some investigations into the nearly optimal feature selection and hyper parameters. The best feature selections in several runs generally have a pattern in it. The common thing is that altitude, topographic position and rainfall data are always in the selected features, while slope degree and topographic position are almost always not. We cannot find any regularity in the 7 Landsat levels and 4 aspects units, but we can observe that it is usually the case that more than half of them are present in feature selection. The optimal number of hidden neurons is 12 in traditional neural network. Surprisingly, only 2 hidden neurons in Casper technique can produce an accuracy rate as good as more than 62%. This strongly indicates that Casper is indeed a very productive technique that can generalize with an extremely small network.

From the figures above, we also know that the genetic algorithm doesn't always improve the performance in each generation. Instead, it follows a pattern of growing gradually and sometimes drops a little bit due to crossover and mutation. As explained in previous sections, we have to use small hyper parameters for genetic algorithm because of the large computation complexity. This can be a huge problem when even larger data set is required. Another possible reason why the genetic algorithm is not able to climb too much is that we already give them small ranges for each value. For example, we range the number of learning rate in [0.0001, 0.1024], then however it generates, it is not going to vary much. Therefore, in the random initialization, it is highly possible that a good classifier setting is generated. This is unavoidable because we specify the range based on experience, where we believe that anything out of range will do no good to the performance. If we want to expand the range, we probably have to increase the generation numbers, but it is not feasible again because of the complexity. The principals and behavior of genetic algorithm, however, reassure us that the outcome should be somewhere near the optimal result.

3.2 comparison with previous researches

In comparison with techniques applied on the same dataset, the neural network described in [7] 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:

Table 6. training and testing accuracy rate from 3 different techniques described reference paper [1]

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

From Table 6, 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. It is highly possibly the case that an accuracy level of around 65% is a limit that this small data set can provide. The result from Table 6 is based on the classification encoding of output, which is shown to be possibly fitting better on our problem.

4 Conclusion and Future Work

This research investigated performance of two techniques, namely traditional neural network and Casper, on the prediction of forest supra-type based on a small GIS data set. A comparison between neural network and Casper is discussed based on the experimental results. To find optimal feature selection and hyper parameters settings, the genetic

algorithm is combined in our program. Some advantages and shortcomings of this kind of combination are also analysed in this paper.

In terms of comparison between traditional neural network and Casper, the former performs better than Casper. Casper is said to be better in generalisation of regression problem, and our task is modified to become a regression problem. However, it still seems to have bottleneck in learning better in such problems. Overall, the results obtained from both techniques are unsatisfactory. The splitting of data may have some problem as well.

For genetic algorithm, we value it for the gradual improvement it makes, and its capability to jump out of local maxima, which generally ensure that it can reach optimal or nearly optimal. However, such a combination may still not be a good idea. Because of the large complexity required to get a satisfying result, it is not feasible in real world to train a neural network hundreds or even thousands of times only to get an optimal settings of feature selection and hyper parameters. The range of the translated DNA values are very small, so it can usually get a relatively good result even during initialization and get stuck in later generations. Therefore, it would be even more feasible to just randomly initialize tens of settings and pick the best one.

There are a lot for us to experiment on in the future. 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. This may be done by adding some dummy data, or more dividing techniques are to be investigated. Besides, instead of genetic algorithm, we need a more efficient and feasible technique to obtain optimal hyper parameters and feature selection. Some research can also be done to find out in which case is Casper better than traditional neural network, and what exactly is Casper not good at.

References

1. Milne, LK & Gedeon, Tom & Skidmore, AK. Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. (1995)
2. Fahlman, S.E., and Lebiere, C. The cascade-correlation learning architecture. In Advances in Neural Information Processing II, Touretzky, Ed. San Mateo, CA: Morgan Kauffman, 1990, pp. 524-532. (1990)
3. Kwok, T., and Yeung, D. Experimental Analysis of Input Weight Freezing in Constructive Neural Networks. In Proc. 1993 IEEE Int. Conf. Neural Networks. pp. 511-516. (1993)
4. Hwang, J., You, S., Lay, S., and Jou, I. The Cascade-Correlation Learning: A Projection Pursuit Learning Perspective. IEEE Trans. Neural Networks vol. 7, no. 2. pp. 278-289. (1996)
5. Adams, A., and Waugh, S. Function Evaluation and the Cascade-Correlation Architecture. In Proc. 1995 IEEE Int. Conf. Neural Networks. pp. 942-946. (1995)
6. Treadgold, N.K. & Gedeon, T.D. Extending CasPer: A Regression Survey. Int. Conf. On Neural Information Processing. (1997)
7. Bustos, RA and Gedeon, T.D. Decrypting Neural Network Data: A GIS Case Study. (1995)
8. Masters, T, Practical Neural Network Recipes in C, Academic Press, Boston, (1993)
9. Diederik P. Kingma & Jimmy Ba. Adam: A Method for Stochastic Optimization (2014)
10. Treadgold, N.K. & Gedeon, T.D. A Cascade Network Algorithm Employing Progressive RPROP. (1997)