Use of the Genetic Algorithm and Distinctiveness of Hidden Units to Optimize the Characterization of Core Porosity

Guang Zhu

Research School of Computer Science,
Australian National University, Canberra, Australia
u6260778@anu.edu.au

Abstract

In many applications of neural networks, the main consideration is the training speed of the neural model. In practice, the most difficult decision to make is about the number of hidden units in each hidden layer, although a relatively large number of hidden units in a hidden layer will improve the accuracy of the training model, it would decrease the speed of the network. The goal of this study was to develop a feed-forward neural network that was trained by genetic algorithm to classify porosity into four groups using petrographical characteristics described in linguistic term [1]. This classification accuracy is higher than the previous work which utilized a feed-forward neural network with the distinctiveness network reduction techniques. This paper shows that the Genetic Algorithm combined with the network reduction techniques proposed by T.D. Gedeon [2] would guarantee a consist level of functionality and a smaller size of the network.

Keywords: Neural Network; Distinctiveness; Porosity; Genetic Algorithm; Backpropagation

1. Introduction

This paper presents an optimization approach using a genetic algorithm to classify porosity into four groups (0(Very Poor), 1(Poor), 2(Fair), 3(Good)). In this paper we will generally assume a feed-forward neural network of one input layer, one hidden layer and one output layer of processing units. This network was trained by a genetic algorithm that is a directed random search technique which is widely applied in optimization problem [3].

We first construct a feed-forward neural network for the classification of core porosity based on the linguistic description, we then apply the genetic algorithm for the optimization of this neural network model. One of the major decisions to make is about the number of hidden neurons in each hidden layer. If the number is too large, we could get a better performance on the training model, but the network model might overfit at last. When the number is too small, the neural network model might fail to capture all the patterns in the dataset [4].

The distinctiveness network reduction technique was used in this paper to get rid of the hidden unit that has similar functionality. The distinctiveness of hidden units is determined from the unit output activation vector over the pattern presentation set [2].

2. Data descriptions

The dataset used in this paper comes from a routine core analysis report on an oil well located in the North West Shelf, this dataset contains 226 core plug samples taken from a total of 54 m of cores obtained from three intervals [1]. The porosity and permeability values ranged from 2% to 22% and from 0.01 millidarcy to 5.9 darcies, respectively [1]. The objective of this study is to present how genetic algorithm and distinctiveness of vectors from hidden units can be utilized in optimizing the classification of linguistic descriptions of core samples into four different classes. The accuracy of this neural network will be tested on an unseen data set separated from the main dataset.

3. Data preprocess

Each attribute in the dataset represent a different property of the sedimentary rock, and they have different impact on the classification of the porosity. So, we need to preprocess the data, in this paper the input data was encoded into 7 numeric inputs by means of a linguistic encoding technique. As for the attributes of “Grain Size”, “Sorting”, “Matrix”, “Bioturbation”, “Lamina”, they could all be put in an increase or decrease order, thus we can allocate values evenly distributed from 0 to 1 to each attribute in each character. However, as for the character “Roundness” which represents the degree of abrasion of a clastic particle, two variables are required to encode this value [1].

The dataset adapted for this paper after preprocessing includes the following columns:

- **Grain Size**: The general dimensions of the particles in a sediment or rock.
- **Sorting**: The particular characteristic of sedimentary particles.
- **Matrix:** The finer grained filling the interstices between the larger grains of a sedimentary rock.
- **Sphericity&Roundness_1**: The degree of abrasion of a clastic particle (normalize into sine and cosine values).
- **Sphericity&Roundness_2**: The degree of abrasion of a clastic particle (normalize into sine and cosine values).
- **Bioturbation**: The churning and stirring of a sediment by organisms.
- **Laminae**: The smallest recognizable unit layer of original deposition in a sedimentary rock.

4. Implementation of Genetic Algorithm

Genetic Algorithm is a heuristic search and optimization approach inspired by natural evolution [5]. Genetic Algorithm could create multiple solutions to a given neural network and evolve them through a number of generations, each solution generated holds all hyperparameters that might help to enhance the performance of neural network model.

A Genetic Algorithm used for the optimization of a neural network model in this paper includes the following procedures:

- Initialize a population of random networks
- Score each member of the population based on the fitness function
- Select and produce the best members of the population
- Mutate some members of the population randomly to find some better generations
- Breed new generations using cross-over operation
- Replace least-fit population with better generations

- Terminate and return the best solution when reaching the end of generation or have an accepted result.

Before we build the neural network mode, we need to classify the porosity value into four groups: Good (>15%); Fair (10-15%); Poor (5-10%) and Very Poor (<5%). We randomly chose 80% of the data set as the training data, and the remaining as the test data, the objective of this experiment is to train a neural network using the training data and the Genetic Algorithm, then test its performance on the unseen test data to see whether it can predict the right class of porosity.

In the experiment, we will use neural networks to perform supervised classification. A standard 7 input × 9 hidden × 4 output backpropagation neural network was used. The results of the network model built in this experiment is shown in Fig 1.

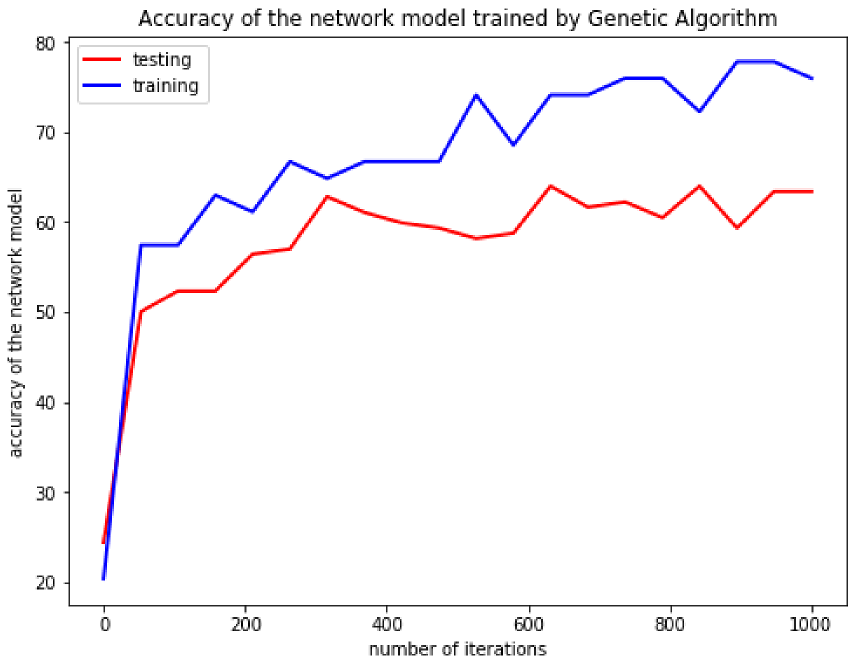

Fig 1 performance of the neural network model trained by Genetic Algorithm

As can be seen from Fig 1, the accuracy of this network model trained by Genetic Algorithm on the training data is around 78% which is slightly higher than we achieved in the previous work. In addition to this, the accuracy on the unseen test data is around 63% which is higher than 60% compared to our previous work. It is clear to see that the Genetic Algorithm contributes to the optimization of neural network model on the classification of core porosity.

As we have mentioned in our previous work, the accuracy on the training data is much higher than the accuracy on the test data. For a situation like this in a neural network model, over-fitting is expected to exist in the training model. There are various causes for the over-fitting in the neural network model, global parameters such as the network size, training time (e.g. based on validation tests) or the amount of weight decay are commonly used to control the bias/variance tradeoff [6]. In this paper we will focus on the network size which refers to the number of hidden units in the hidden layer. If over-fitting exists in network model, it is possible that the number of hidden units is too large. So, in the next phase of the experiment we will use some modern technique to reduce the size of the network, thus ideally will improve the network's performance on the test data set.

5. Implementation of Distinctiveness

There is no method to clearly identify and determine the exact number of hidden units in a neural network model [10]. This paper will discuss the appliance of distinctiveness of hidden units, the method of distinctiveness is applied by determining the similarity of all pairs of hidden units in the hidden layer. After comparing the similarity between each hidden unit, the neural network model will be pruned by removing the redundant hidden unit that has similar functionality compared with other hidden units [2]. As the result, the pruned network model does not need further training, since only the neurons with similar functionality were removed.

For pruning it is necessary to identify hidden neurons with similar functionality [8]. As mentioned by T.D. Gedeon [2], the recognition of similarity of pairs of vectors in done by the calculation of the angle between them in patter space. If the angle between two hidden units is less than 15°, then these two hidden units are considered too similar and one of them is removed. If the angular separation of two hidden units is over 165°, these two hidden units are considered complementary, and both should be removed.

Since there are 9 hidden units in the neural network model we built using Genetic Algorithm, we need to calculate the vector angles for these 9 hidden units, the results are shown below in Fig 2.

From the results it is clear to see that the vector angle between the 8th and the 9th hidden unit is 8.54° which is less than 15°, and vector angle between other hidden units and these last two hidden units are always the same, for example the vector angel between the first hidden unit and the 8th hidden unit is 31.14° which is equal to the vector angle between the first hidden unit and the 9th hidden unit. We can confirm that the last two hidden units have the similar functionality, thus one of them will be removed to accelerate the speed of the network model. In addition to this, there is no angle between hidden units that is greater than 165°, so none of the 9 hidden units are considered complementary.

The next step is to use the result of the distinctiveness of hidden units to prune the neural network. Since the 8th hidden unit has similar functionality as the 9th hidden unit, we chose to remove the 9th hidden unit. The network after pruning is now a standard 7 input × 8 hidden × 4 output backpropagation neural network.

Pairs of units	Vector angle	Pairs of units	Vector angle	Pairs of units	Vector angle
12	143.04	27	63.51	48	142.86
13	55.84	28	139.27	49	142.86
14	150.43	29	139.27	56	74.38
15	25.55	34	116.58	57	120.98
16	71.70	35	49.84	58	29.73
17	140.93	36	97.81	59	29.73
18	31.14	37	111.35	67	98.79
19	31.14	38	54.77	68	76.87
23	100.35	39	54.77	69	76.87
24	40.48	45	141.99	78	137.52
25	147.98	46	84.95	79	137.52
26	116.11	47	45.55	89	8.54

Fig 2 vector angles between hidden units

6. Comparison and Discussion

From Fig 3 we can see that the accuracy of the pruned neural network on the training data is around 81%, and the accuracy on the test data is around 65% compared to the result of the unpruned model which is 63%. In addition to this, the time taken for the training of the unpruned network model is 460 seconds, while the pruned network model takes 435 seconds, there are enough evidence to show that removing the redundant hidden unit can improve the performance and the training speed of the neural network model on the testing set.

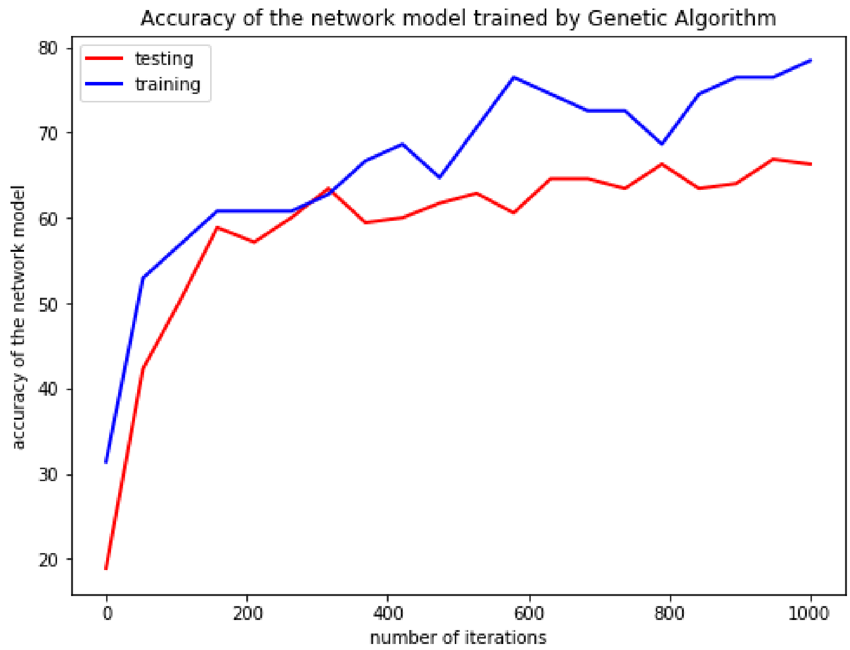

Fig 3 performance of the pruned neural network

Compare to our previous work, the Genetic Algorithm can improve the performance of our neural network model, however the Genetic Algorithm only improves the accuracy by around 5%, there are several reasons for this. First is that the dataset we had used is relatively small, the dataset only contains 226 samples and since we had choose to use 80% of the dataset for training and the remaining 20% for testing, there are only around 45 samples that our neural network can be tested on which is small, thus the effect of Genetic Algorithm can not be fully presented in this experiment. Second is that during the preprocessing of dataset, we dropout some features to simplify the neural network model, those features that we dropped may have influence on the performance of the Genetic Algorithm, thus may reduce the effect of Genetic Algorithm on the optimization of neural network model.

From T. Gedeon et al [1] experiments on the same data set, the overall accuracy on the blind test on Set #1 is 60.0%, and the overall accuracy on the blind test on Set #2 is 62.8%. The average accuracy on the T.D. Gedeon's neural network model is 61.4% which is close to the accuracy of the pruned neural network model in this paper. We can say that the experiment in this paper is a successful simulation of the experiments of T. Gedeon et al [1].

7. Conclusion

The two-layer feed-forward neural network with 8 hidden units is proved to be acceptable in classifying the core porosity using the related features. In this paper we have demonstrated that using the Genetic Algorithm we can improve the performance and the training speed of the neural network model. In addition to this, we had found that to a certain extent, the distinctives network reduction technique will help to remove the hidden units with similar functionality without sacrificing too much of the neural network's classification ability [4].

During the experiments in this paper, we found that the use of preprocessing and input encoding of the original data improve the performance of neural network. The next stage in our work will include using all features in the dataset to train the neural network model and optimize it using the Genetic Algorithm; apply the distinctiveness network reduction technique in other neural networks, for example image compression. Since the computation complexity of determining the distinctiveness between hidden units is $O(n^2)$ (n refers to the number of hidden units in a neural network model), a more efficient algorithm to determine the distinctiveness between hidden units would also be considered as part of the future project.

References

1. Gedeon, T.D., Tamhane, D., Lin, T., Wong, P., 2001. "*Use of linguistic petrographical descriptions to characterise core porosity: contrasting approaches.*" Journal of Petroleum Science Engineering 31, 193–199.
2. Gedeon, TD, & Harris, D "*Progressive Image Compression*," Proceedings International Joint Conference on Neural Networks, vol. 4, pp. 403-407, Baltimore, 1992.
3. F. H. F. Leung, H. K. Lam, S. H. Ling and P. K. S. Tam, "*Tuning of the structure and parameters of a neural network using an improved genetic algorithm*,"

in *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79-88, Jan. 2003, doi: 10.1109/TNN.2002.804317.

4. Zirui, Josephine, "*Classification of Humans' Perception of Manipulated and Unmanipulated Digital Images Using Feedforward Neural Network with Network Reduction Technique*", 2019
5. Benjamin Lynch, "*Optimizing with genetic Algorithms in Neural Networks* ", 2006. [Online]. Available: https://www.msi.umn.edu/sites/default/files/OptimizingWithGA.pdf. [Accessed: 30 May 2020].
6. S. Lawrence and C. L. Giles, "*Overfitting and neural networks: conjugate gradient and backpropagation*," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 2000, pp. 114-119 vol.1, doi: 10.1109/IJCNN.2000.857823.
7. A. Abraham, "*Analysis of Genetic Algorithms and Distinctiveness in Back Propagation Neural Network on Diabetic Retinopathy classification*," 2018.
8. T. D. Gedeon, "*Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour*," Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, 1995, pp. 26-29, doi: 10.1109/ANNES.1995.499431.
9. S. Huang and Y. Huang, "*Bounds on number of hidden neurons of multilayer perceptrons in classification and recognition*", IEEE International Symposium on Circuits and Systems, 1991.
10. M. Li, T. Zhang, Y. Chen and A. Smola, "*Efficient mini-batch training for stochastic optimization*", Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14, 2014.
11. L. Smith, "*Cyclical Learning Rates for Training Neural Networks*", 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.