Compare the performance of traditional gradient descent method and NEAT algorithm in the classification of rock lithology by generative neural network

Yousong Zeng

School of Computer Science & Engineering
Australia National University
U6660070@anu.edu.au

Abstract In petroleum and geological exploration, the distinction of rock porosity is crucial. Replacing expert systems with neural network models to distinguish porosity has become a future development trend. This study compares the performance of the models generated by the gradient descent method and the NEAT algorithm in related classification problems. The results show that the traditional gradient descent algorithm is more suitable for supervised learning projects such as rock porosity classification

Keywords: Core porosity, Neural network, Classification, Supervised Learning, NeuroEvolution of Augmenting Topologies, Evolutionary algorithm

1. Introduction

In petroleum exploration, understanding the form and distribution of rock porosity in the formation is the basis for successfully characterizing petroleum reservoirs. The traditional method is to use an expert system to evaluate the rock porosity. Although this method has higher reliability, it is less efficient and consumes human resources. As a powerful bionic modeling tool, neural network has been widely used in the field of machine learning and cognitive science with the continuous innovation of hardware technology. Similarly, in geological prospecting, neural networks are often used to replace some of the work of experts to save human resources. The advantage of a neural network is that it can reduce a lot of expensive expert time by learning from the example data fed to it(Harry.S & Tamás D, 1993).

The emergence of evolutionary algorithms provides new ideas for training neural networks. The emergence of evolutionary algorithms provides new ideas for training neural networks. Neuroevolution (NE) is the combination of evolutionary algorithms and neural networks. He no longer uses traditional gradient descent methods to train models, but uses natural selection of the survival of the fittest in genetic algorithms to screen out excellent models(K.O. Stanley & R Miikkulainen, 2002). NeuroEvolution of Augmenting Topologies (NEAT) is a kind of NE algorithm.

The purpose of this study is to classify rock porosity by training neural network models using gradient descent and NEAT methods, respectively, and compare the performance of the two methods.

2. Method

2.1 Pre-processing

This study used data to analyze the lithofacies in the strata and classify the demonstrated porosity. There are various forms of sedimentary rock characteristic heterogeneity in clastic rock reservoirs, such as porosity. Understanding the forms and spatial distribution of these heterogeneities is the basis for successfully characterizing petroleum reservoirs. From a geological point of view, the anatomy of reservoir heterogeneity requires two main pieces of information: the constituent lithofacies and their hydraulic properties and their internal structures (Tom. D, Dilip. T, Tao. L & Patrick, M, 2001). Due to the subjective analysis of the lithology that is mainly used for research, the data we used used six main lithologies, as follows:

Table 1: Characters and attributes used for porosity classification (Tom. D, Dilip. T, Tao. L, Patrick, M, 2001).

Characters and attributes used for porosity classification

Character (no. of attributes)	Descriptions	Attributes
Grain size (12)	The general dimensions (e.g. average diameter or volume). of the particles in a sediment or rock, or of the grains of a particular mineral that made up a sediment or rock.	Very Fine, Very Fine to Fine, Fine, Fine to Medium, Medium, Fine to Coarse, Medium to Fine, Medium to Coarse, Fine to Very Coarse, Coarse to Very Coarse, Very Fine with Coarse Quartz, Fine with Coarse Quartz.
Sorting (6)	The dynamic process by which sedimentary particles having some particular characteristic (e.g. similarity of size, shape, or specific gravity).	Well, Moderate to Well, Moderate to Poor, Moderate, Poor to Moderate, Poor.
Matrix (14)	The smaller or finer grained, continuous material enclosing, or filling the interstices between, the larger grains or particles of a sediment or sedimentary rock.	Argillaceous (Arg), Sideritic (Sid), Siliceous (Sil), Sid with Arg, Sid with Sil, Arg with Sil, Sil with Arg, Carbonaceous, Calcareous, Pyritic with Arg, etc.
Roundness (8)	The degree of abrasion of a clastic particle as shown by the sharpness of its edges and corners as the ratio of the average radius of curvature of the maximum inscribed sphere.	Subangular (subang), Angular (Ang) to Subang, Subang to Subrounded (subrndd), Subrndd to Ang, Subang, Subrndd, etc.
Bioturbation (6)	The churning and stirring of a sediment by organisms.	Abundant bioturbation (bioturb), Increase bioturb, Bioturb, Decrease bioturb, Minor bioturb, Trace of bioturb.
Lamina (10)	The thinnest or smallest recognisable unit layer of original deposition in a sediment or sedimentary rock	Irregular argular, Irregular Calcareous, Trace of Calcareous, Less Traces, Argillaceous, Calcareous, Irregular Silt, Thick, Irregular.

It can be seen from the table that the six main attributes are discrete distributions rather than continuous. The data we obtained is also one-hot processed data, which is not conducive to our subsequent neural network training and interpretation, so I Pre-process the data to convert all attributes into numeric form**.**

For some attributes, there is an obvious sequence relationship, such as the attribute 'sorting', the category can obviously form a chain: 'Poor – Poor-moderate – Moderate-poor – Moderate – Moderate-well – Well-moderate – Well' It is very convenient to distribute it evenly to 0-1. For other attributes, such as "Roundness" in the data we used, the category of this attribute has no sequence relationship, so they cannot be normalized to the range of 0-1 in a simple order. I used the method proposed by Tom, Dilip, Tao and Patrick(2001). It is named Circular encoding, which is quantify the categories of the attribute into two-dimensional vectors, distribute the categories evenly in two-dimensional coordinates, and then encode and normalize them with sine and cosine Each category is represented as a vector in the range of 0-1.

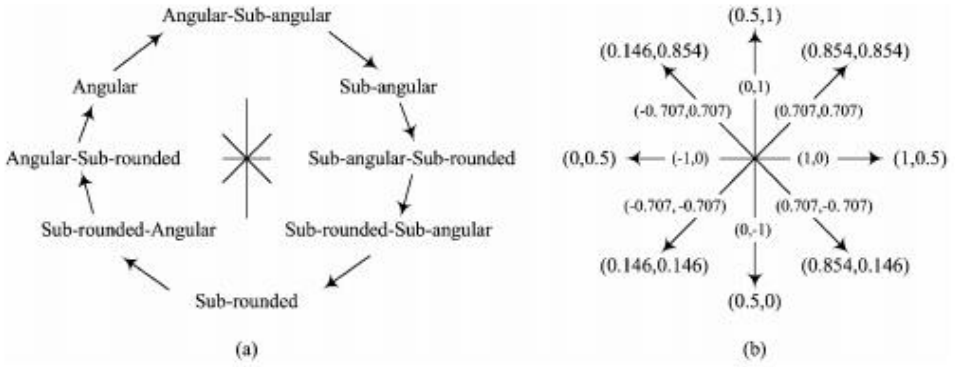

Fig 1. (a)circular encoding of roundness. (b)normalize to vector.(Tom. D, Dilip. T, Tao. L, Patrick, M, 2001)

Since the source data does not have a label, I also need to classify the data according to porosity. According to Tom, Dilip, Tao and Patrick(2001), the data should be divided into 4 classes, The basis for classification is: “Very poor” (porosity < 5%), “Poor” (5% < porosity < 10%), “Fair” (10% < porosity < 15%), “Good” (porosity>15%).

After re-encoding the data, extracting valid data, and removing redundant items, the final data has 168 rows and 9 columns, which will be the final training data set.

2.2 Back-propagation neural network

A basic three-layer back-propagation neural network was established for classification, and gradient descent method was used to train the model. In this three-layer network, there are 8 input neurons (sorting, Bioturbation1, Bioturbation2, Grain size, Matrix, Roundness1, Roundness2, Lamina), which are the eight input attributes. Since there are four final classification labels, and the cross-entropy loss function is selected to calculate the loss of the model more accurately, the output neurons are determined to be four. The selection of the number of neurons in the hidden layer is relatively complicated. Domy, Risanuri, and Indah (2018) believe that increasing the number of hidden layer neurons will improve the accuracy of the model to a certain extent, but it will also reduce the generalization ability of the model, which may lead to the risk of overfitting. Therefore, For the model, a reasonable choice of the number of hidden layer neurons is crucial. Fanggang, Octavia, Chung and Jingwen (2016) pointed out that according to the kolmogorov-arnold theorem, any n-ary continuous function can be expressed by the sum of a set of continuous functions, and the number of summation functions does not exceed 2n + 1. This theorem just fits Our three-layer neural network, so we determine that the number of hidden neurons is 2n +1 = 17. In order to avoid falling into the local optimum and effectively adjust the gradient, the learning rate should be 0.01. The number of training epochs is 5000, which ensures that the network can fully learn from the training data. The activation function of the hidden layer is the Sigmoid function.

2.3 NeuroEvolution of Augmenting Topology(NEAT)

In traditional NE approaches, a topology is chosen for the evolving networks before the experiment begins. Usually, the network topology is a single hidden layer of neurons, with each hidden neuron connected to every network input and every network output. Evolution searches the space of connection weights of this fullyconnected topology by allowing high-performing networks to reproduce. The weight space is explored through the crossover of network weight vectors and through the mutation of single networks’ weights. Thus, the goal of fixed-topology NE is to optimize the connection weights that determine the functionality of a network (K.O. Stanley & R Miikkulainen, 2002). However, connection

weights are not the only aspect of neural networks that contribute to their behavior. Modifying the network structure has been shown effective as part of supervised training (Chen et al., 1993). NEAT is based on this theory. In the genetic algorithm, the carrier of the genetic information of the genome, the NEAT algorithm incorporates the topology and node parameters of the neural network into the genetic information, which means that the algorithm not only optimizes the parameters in the natural selection of the survival of the fittest, but also screens out the better. Head structure.

Table 2. Examples of node genomes in the NEAT

Node index	Node 1	Node 2	Node 3	Node 4	Node 5
Node Genes	Input node	Input node	Input node	Hidden node	Output Node

Table 3. Examples of connect genomes in the NEAT

Connect index	In : 1 Out : 4	In : 2 Out : 4	In : 3 Out : 4	In : 2 Out : 5	In : 5 Out : 4	In : 1 Out : 5
Connect Genes	Weight 0.7 Enabled	Weight -0.5 Disabled	Weight 0.5 Enabled	Weight 0.2 Enabled	Weight 0.4 Enabled	Weight 0.6 Enabled

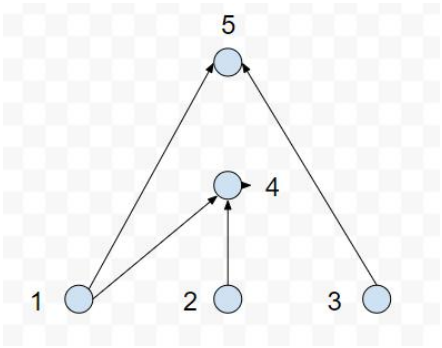

Fig 2. A simple neural network generated from the genomes in Table 2 and Table 3

Table 2 and Table 3 show the two genomes in the NEAT algorithm, which distinguishes Nodes and connects. The figure 2 is a genotype to phenotype mapping basis on the genomes in Table 2 and Table 3. A genotype is depicted that produces the shown phenotype. There are 3 input nodes, one hidden, and one output node, and seven connection definitions, one of which is recurrent. The second gene is disabled, so the connection that it specifies (between nodes 2 and 4) is not expressed in the phenotype.

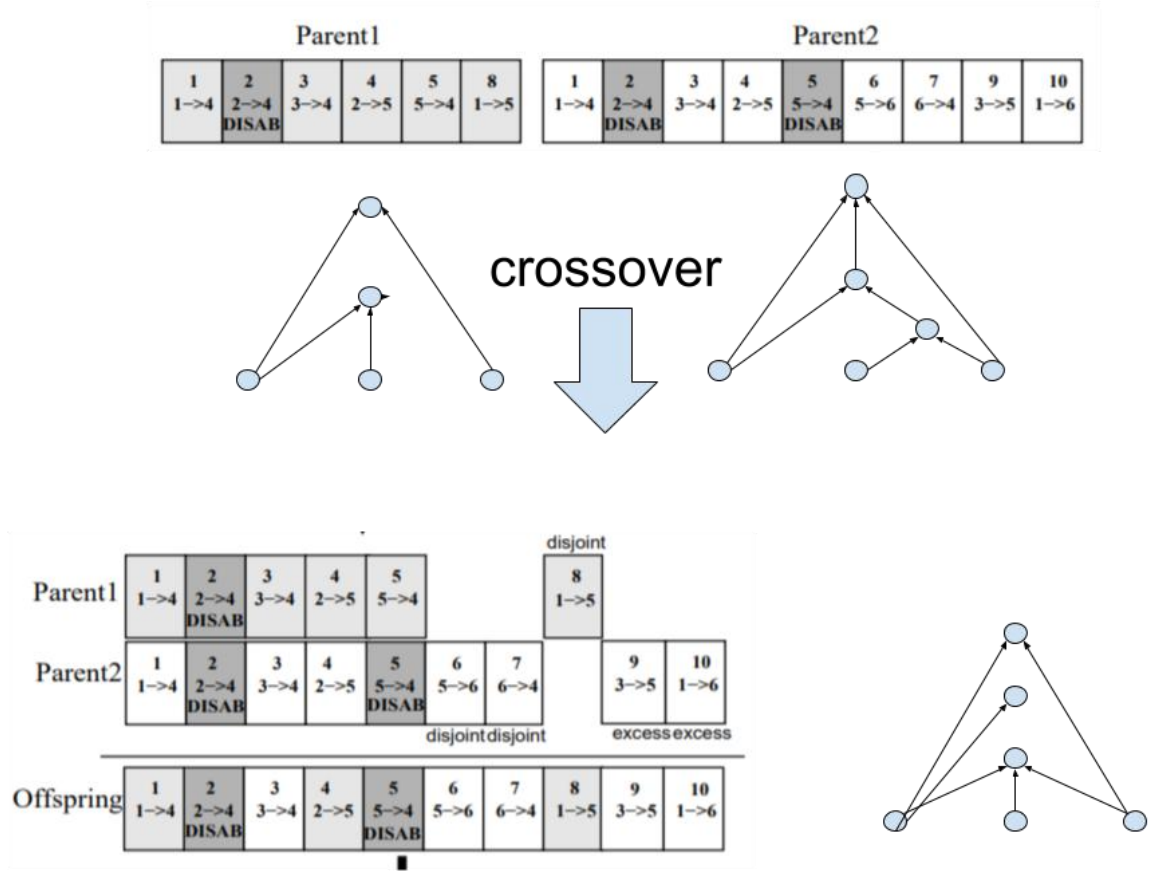

Fig 3. In the NEAT algorithm, the example of parent generates the child through crossover

In figure 3, Although Parent 1 and Parent 2 look different, their node numbers tell us which genes match up with which. Even without any topological analysis, a new structure that combines the overlapping parts of the two parents as well as their different parts can be created. Matching genes are inherited randomly, whereas disjoint genes (those that do not match in the middle) and excess genes (those that do not match in the end) are inherited from the more fit parent. In this case, equal fitnesses are assumed, so the disjoint and excess genes are also inherited randomly. The disabled genes may become enabled again in future generations: there's a preset chance that an inherited gene is disabled if it is disabled in either parent.

In this experiment, the population of each generation of the NEAT algorithm is set to 200, and the generations is set to 5000. Since we are studying classification problems, I used cross entropy to calculate fitness. Eq.1 shows the principle of cross entropy calculation.

$$L = \frac{1}{N}\sum_{i} L_i = \frac{1}{N}\sum_{i} -\sum_{c=1}^{M} y_{ic}\log(p_{ic}) \tag{1}$$

The final result is trained according to the hyperparameters set above.

3. Result

3.1 Performance of BP Neural Network

After training with 5000 epochs, a bp neural network model that is finally used to classify porosity is trained.

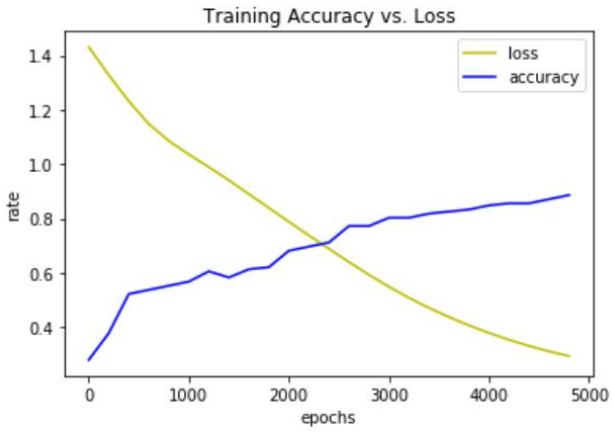

Fig. 4. Trends in training accuracy and loss during training

It can be seen that loss is gradually decreasing and accuracy is gradually increasing, which shows that the model gradually converges during training. It can be seen that although the final loss is still gradually declining, the accuracy almost does not rise. In order to enhance the generalization ability of the model and reduce the possibility of over fitting, the training should be stopped at this time.

Table 4. Confusion matrix of training set

Actual \ Predicted	VP	PR	FR	GD	Total	Accuracy
VP	27	0	0	0	27	100%
PR	5	31	1	0	37	83.8%
FR	0	2	18	6	26	69.2%
GD	0	0	1	41	42	97.6%
Total Accuracy						88.6%

Table 5. Confusion matrix of testing set

Actual \ Predicted	VP	PR	FR	GD	Total	Accuracy
VP	8	3	0	0	11	72.7%
PR	3	2	1	0	6	33.3%
FR	0	2	7	2	11	63.6%
GD	0	2	1	6	9	66.7%
Total Accuracy						62.16%

Table 4 and Table 5 show the confusion matrix of the model for the training set and test set, respectively. It can be seen that the accuracy of the training set reaches 88.7%, while the accuracy of the test set is only 62.7%, and there is a

relatively large gap between the two. Although the accuracy of the test set fluctuates in the range of 55% -70% due to the uncertainty of the neural network, there is always a relatively large deviation from the results of the test set.

3.2 Performance of NEAT

After a population of 200 generations in a population of 200 generations, the neural network generated by the NEAT algorithm was finally obtained. The following is a schematic diagram of the neural network model.

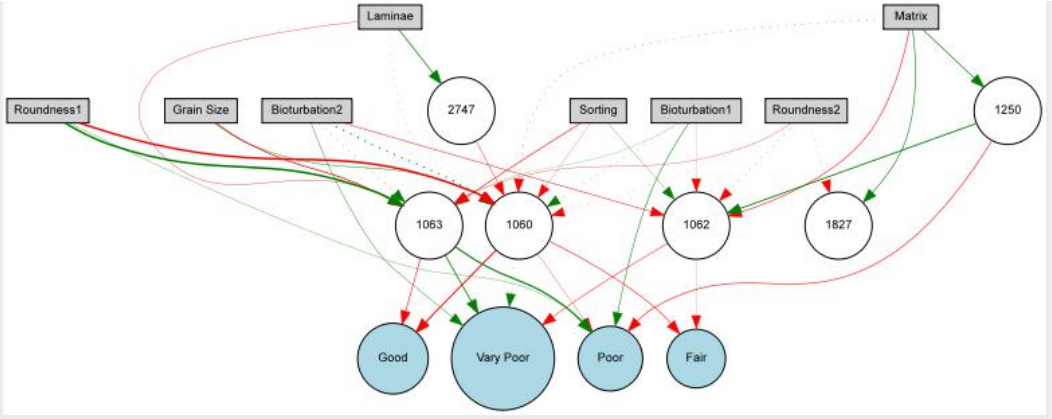

Fig 5. Schematic diagram of neural network generated by NEAT algorithm. (In the net, if it is a solid line, it is Enable, if it is a dashed line, it is Disable; the red line indicates the weight weight <= 0, the green indicates the weight> 0, and the thickness of the line is related to the size.)

It can be seen from the figure that the neural network finally formed by the NEAT algorithm has 8 input nodes, 6 hidden nodes, and 4 output neurons. Although there is no very neat structure, the model is the optimal model after natural selection.

Table 6. confusion matrix of Training set

Actual \ Prediction	VP	PR	FR	GD	Total	Accuracy
VP	19	10	1	2	32	59.4%
PR	7	19	5	8	39	48.7%
FR	4	4	6	11	25	24%
GD	0	0	2	42	44	95.5%
Total Accuracy						61.4%

Table 7. Confusion matrix of Testing set

Actual \ Prediction	VP	PR	FR	GD	Total	Accuracy
VP	3	2	1	0	6	50%
PR	1	2	1	0	4	50%
FR	1	3	2	6	12	25%
GD	0	0	2	5	7	71.4%
Total Accuracy						41.4%

Table 4 and Table 5 show the confusion matrix of the model for the training set and test set, respectively. Table 4 and Table 5 show the confusion matrix of the model for the training set and test set, respectively. Due to the high randomness of the neural network generated by the NEAT algorithm, the accuracy of the training set fluctuates in the range of 52% to 65% during multiple debugging, and the accuracy of the test set fluctuates in the range of 38% to 43%

4. Discussion

Through the analysis of the confusion matrix of BP neural network, I found that the model has a lower accuracy when distinguishing the poor class, and it is easier to classify the samples of the Poor class as the very poor class. The reason for this result may be that the rule when distinguishing data is to forcibly classify the porosity in the rock, which results in samples with very similar porosity but just being divided into two different categories, which is somewhat misleading to the model. For the problem that there is a certain difference in accuracy between the training set and the test set of the BP neural network, after research, it may be because the training set has relatively few samples. In order to allow the neural network to extract features more fully and learn more fully, I am in Increased the number of epochs during training, resulting in over-fitting.

When using the NEAT algorithm to generate a neural network model, because the natural evolution of the population is a long process, I spent a lot of time on training the model.

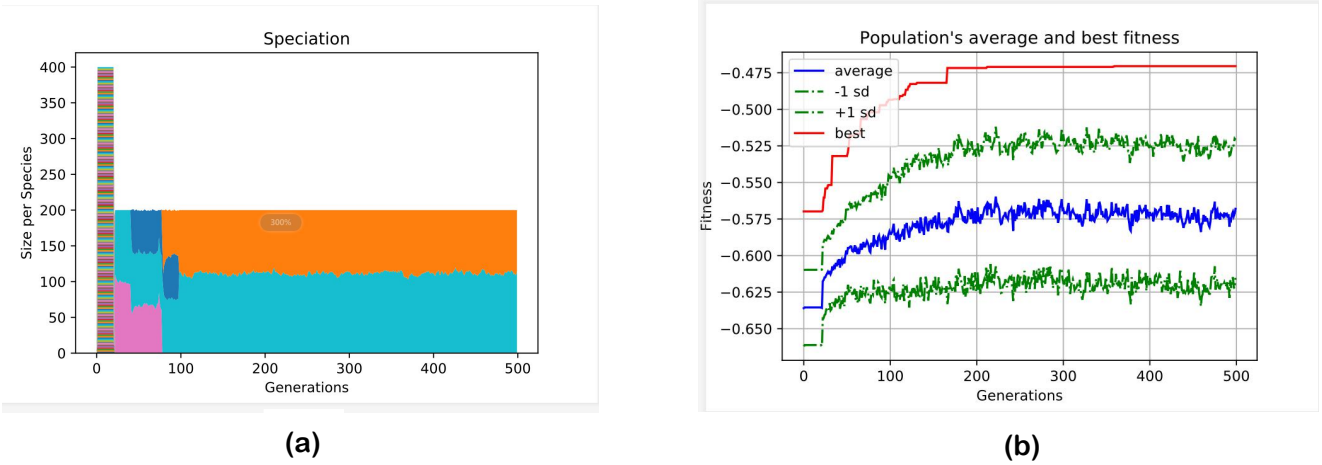

Fig 6. (a)Trend of species evolution in model training (Each color represents a species).

(b) Trend of overall fitness of the population

Fig 6 shows the evolution of various populations during the model training process. At the beginning of the training, the model automatically generated 200 species, each species has two individuals, which avoids that some populations have only a single individual, so that the offspring cannot be reproduced by genetic algorithm, and ensures that the genetic algorithm can run smoothly. After 20 generations have been propagated, the algorithm screens the population. If certain populations are stagnated, they are removed. This is the embodiment of natural selection in the algorithm, which ensures that our model will eventually converge. After the 20th generation, the species diversity dropped sharply, indicating that the population with higher fitness was retained, and the population with lower fitness was eliminated, and the average utilization and optimal fitness of the population did not increase to the end, represent the convergence of the model. It can be found from the picture that new species have been mutated and old species have been eliminated during the reproduction process. In addition, after many times training to compare all models, I found that the structure

of the neural network generated each time is very different. This is also the advantage of the NEAT algorithm, so that we will not be limited by the neural network. The structure can search for the optimal solution in a larger solution space.

By analyzing the confusion matrix of the model generated by the NEAT algorithm, it is found that the model is very inaccurate for the classification of the FR class, and the accuracy rate is only about 25%. There may be two reasons, one is that the characteristics of the FR data itself are not obvious, and the second is due to the characteristics of the NEAT algorithm. When calculating fitness, the overall fitness of the population is considered, which results in better overall results but partial classification The model with poor performance survives. This can be modified by increasing the fitness calculation method to increase the penalty for the local poor performance population.

From the above results, whether it is in the accuracy of the training set or the accuracy of the test set. The BP neural network trained by gradient descent method is superior to the neural network trained by NEAT algorithm. Moreover, the difference between the two types of development is also reflected in the efficiency of the algorithm. We performed 5000 epochs when training the BP neural network, and it took about 5 seconds to complete the model training. When using the NEAT algorithm for 5000 generation iterations, it took more than 100 minutes to complete the training. This shows that for projects that use rock equivalent data in rocks to classify rock porosity, the gradient down method is more suitable than the NEAT algorithm.

5. Conclusion

The BP neural network implemented by the gradient descent method performs better than the model generated by the NEAT algorithm in the classification of rock porosity. The advantage of BP neural network lies in its years of experience and relatively high efficiency in the field of supervised learning. The advantage of the NEAT algorithm is the flexible and changeable structure. This advantage is not fully reflected in the field of supervised learning, and it is more suitable for the NEAT method in the field of reinforcement learning with greater uncertainty.

Reference

[1] Chen, D., Giles, C. L., Sun, G. Z., Chen, H. H., Lee, Y. C., & Goudreau, M. W. (1993, March). Constructive learning of recurrent neural networks. In IEEE International Conference on Neural Networks (pp. 1196-1201). IEEE.

[2] Féraud, R., & Clérot, F. (2002). A methodology to explain neural network classification. *Neural Networks, 15*(2), 237-246. doi:10.1016/S0893-6080(01)00127-7

[3] Gedeon, T. D., Tamhane, D., Lin, T., & Wong, P. M. (2001). Use of linguistic petrographical descriptions to characterise core porosity: Contrasting approaches. *Journal of Petroleum Science and Engineering, 31*(2), 193-199. doi:10.1016/S0920-4105(01)00130-9

[4] Kristomo, D., Hidayat, R., & Soesanti, I. (2018). Syllables sound signal classification using multi-layer perceptron in varying number of hidden-layer and hidden-neuron. *MATEC Web of Conferences, 154*, 3015. doi:10.1051/matecconf/201815403015

[5] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. Evolutionary computation, 10(2), 99-127.

[6] Wang, F., Dobre, O. A., Chan, C., & Zhang, J. (2016). Fold-based kolmogorov-smirnov modulation classifier. *IEEE Signal Processing Letters, 23*(7), 1003-1007. doi:10.1109/LSP.2016.2572666