An Application of Neural Network Based on Genetic Algorithm: Classification of Radar Return

Yue Jiang,

Computer Science and Information Technology Building, College of Engineering and Computer Science, Australian National University, Acton ACT, 2601 Canberra, Australia

u6311789@anu.edu.au

Abstract. Neural network is proved that it can be used in various range of problems and genetic algorithm (GA) becomes more and more popular these years. The goal of this paper is to prove that neural networks are suitable for the classification job and contributes to demonstrate the relationship between performance of neural network and the number of its hidden node. This paper will also research on how GA can influence neural networks. The method which is used in this paper is multilayer feedforward networks with back propagation algorithm. This experiment applies the radar return data set from Space Physics Group of The Johns Hopkins University Applied Physics Laboratory to train neural network. The conclusion of this paper is that neural network is suitable for the classification job with appropriate hidden nodes and the number of hidden nodes can influence the performance of neural network, but the relationship is not the positive correlation. In addition, this paper also shows GA can improve the performance of neural networks.

Keywords: neural network, classification, hidden nodes, genetic algorithm

1 Introduction

Neural network is proved that it can be used in various range of problems. It can learn from the training using a specific data set. Waugh & George[1] research on the data set of abalones and claim that the age of abalones can be predicted by using neural network. And they also claim that neural network with hidden nodes performs better than that doesn't have hidden nodes. Prediction is the problems that neural network suits, not only predicting the age, but also predicting the classification. It's one of the classic applications of neural network.

Genetic algorithm (GA) is inspired by the biological evolution and is popularly used in many different fields. Jahangiri & Farhad[2] claim that GA can be used to optimize the learning progress by implement several operators which is from the features of biological evolution. Therefore, improving neural network by using GA becomes a good assumption. Because GA can be helpful to filter the features that influence the performance of neural networks most.

In this paper, an experiment is conducted to prove neural network has a good capability of classification and can be improved based on GA. The data set used here is about ionosphere.

The data set is provided by Space Physics Group of The Johns Hopkins University Applied Physics Laboratory and it is collected by 17 antennas. 16 of the 17 antennas gain high frequency and the other one is low-frequency. The set is collected to study the ionosphere at the high altitude. The pulse from radar is transmitted to the ionosphere and then is received by the receiver of ground. By using the returns of the pulse, some physical problem can be solved, such as the determination of target velocity. And according to Sigillito, Wing, Hutton and Baker[3], from the received signal, the autocorrelation function (ACF) can be given to derive the target velocity. The detailed autocorrelation function can be found in the paper of Sigillito, Wing, Hutton and Baker. By simplifying the return signal, 17 pairs of return value can be demonstrated. In each pair, there are one value representing the real part of the complex ACF and the other one representing the imaginary part. Because the total number of the returns received in a year is too big, it is not possible to do the classification manually. Thus, the neural network can be used.

As referred above, the data set of radar returns has 17 pairs (34) of radar data as the input attributes and the one desire classification as the output class. The desire output varies between good and bad. In addition, this data set has 351 instances in total. 200 of the total instances are training data and 151 of them are test data. The reason why I choose this set as data set for my paper is that this set can be used to train the classification neural network. In addition, it has enough attributes and instances to obtain a clear conclusion.

This experiment mainly has two part and has three main goals. The first part is to achieve the first two goals and the last goal is achieved by the second part.

1. The first goal is to prove that neural network is suitable for classification job.
2. The experiments of Waugh & George[1] has proved that hidden nodes can help the improvement of neural network compared with perception, which is donated to a input-output structure without hidden nodes. But they did not discuss whether the addition of hidden node can improve the performance of the neural network. Thus, the second goal of this paper is to investigate the relationship between performance of neural network and the number of hidden nodes it has.
3. The third goal is to prove that GA is useful to improve the performance of neural networks.

This paper will use multilayer feedforward networks, which includes an input layer, a hidden layer and an output layer. And then this paper will use GA to improve the performance of multilayer feedforward networks. The detail of model design will be discussed in the following chapter. Since this data set was also used by Sigillito, Wing, Hutton and Baker[4] in their paper “classification of radar returns from the ionosphere using neural networks”, this paper will also do the comparison with the results they conducted. However, in Sigillito, Wing, Hutton and Baker’s paper, they did not mention genetic algorithm. Therefore, I choose another paper which mainly research neural networks and GA to compare the result about GA in this experiment.

In this paper, chapter two would be the detailed description of experiment method. Chapter three would be the display of the experiment’s result, the discussion about the result. Chapter four would be the conclusion and final chapter would be about the future work.

2 Methods

2.1 Model design

The model that I choose for this experiment is multilayer feedforward network and multilayer feedforward network based on genetic algorithm.

a) Multilayer feedforward networks

Multilayer feedforward networks that this paper uses are consisted by three parts. They are input layer, hidden layer and output layer. Input layer is an interface to the input data and transfers the data to hidden layer without computations. The hidden layer is to process the data from input layer with a specific activation function and then transfer it to the output layer. The hidden layer is not connected directly with the outside but connected with input and output layer only. In addition, the nodes in the hidden layer can be changed to achieve the experiment goals. After hidden layer, the final layer of this network is output layer and the output of output layer is the predicted class value. Every connection between different layers has a weight and the transferred data is changed by the weight. By updating the weight, network can learn and improve the performance of classification.

Usually, the networks use the back propagation algorithm to update the weight of connections and in this way the accuracy of the output can be improved. The algorithm is started with the comparison between actual output and the desire output and move backwards layer by layer. The back propagation algorithm includes two different phases. The first phase is called propagation, which is mainly do the propagation from the output value to the input value in every layer and calculate the error. The second phase is the update of every weight after doing the propagation to receive a better performance. In the back propagation algorithm, the quality and speed of training process can be influenced greatly by the learning rate. Therefore, determining an appropriate learning rate is very important.

The reason why I choose the multilayer feedforward network is that this kind of network is that it has a great performance in prediction jobs according to the previous paper. And the number of hidden neurons in the hidden layer is easy to change. By changing this number and compare the results, the goal of this experiment can be reached.

b) Genetic algorithm

Genetic algorithm is an optimization technique which is inspired by biological process, mainly about the evolution principles. The popularity of genetic algorithm is increasing in recent years. The objective of genetic algorithm contains several operations, such as initialize population, crossover, mutation and selection, which are also inspired by evolution principles. Genetic algorithm often stared with initialization of population of individuals, which is also can be seen as candidate solutions. And each individual is represented as a chromosome. In the meanwhile, evaluate function or fitness function has to be determined. The fitness function is to select better solution based on fitness value. If the result after fitness function does not satisfy the criterion of termination, the population will experience crossover, mutation and selection and produce a new population until the criterion is met. The process of genetic algorithm is as figure 1.

Fig. 1. Genetic algorithm progress

2.2 Implementation

At the first part of this experiment, I conduct a neural network to achieve the first and the second goals. The data set contains 34 different attributes in the radar data. Therefore, neural network sets 34 input nodes in the input layer. Because there are 2 different desire outputs, the network has 2 output nodes in the output layer. As for the output value, the network use 0 to refer to good returns and 1 to refer to bad returns. As for the hidden layer, I choose 1,5, 10 and 15 hidden neurons as sample. By comparing the result of these samples, the conclusion will appear. In addition, the network use loss function to describe the loss during back propagation process.

The total number of the data set is 351. Therefore, I choose the first 200 instances from the data set in the training process and the last 151 instances in the test process. The reason why I divided the data set like this is to be consistent with the paper of Sigillito, Wing, Hutton and Baker. In addition, the value of input data is between -1 and 1, so I choose 0.03 as the learning rate. And I also choose the total number of epoch as 20000 to present the result more conveniently.

The result is presented by accuracy of the prediction. Accuracy is used to evaluate the performance of this network. The formula of accuracy is as below:

$$\text{Accuracy } (\%) = \text{correct prediction number / total output number.} \qquad \textbf{(1)}$$

The performance of networks is evaluated by final accuracy in both training process and test process to prove that neural network can be used in the classification jobs. Besides, by changing the number of hidden node in the hidden layer, the performance changes of the network can be observed.

At the second part of this experiment, I introduce GA to the neural network above. I use GA to select features to improve the performance of this network. I initialize the 20 individuals as the initial population. The individual of the population that represent candidate solution is a 0-1 string whose length is same as the number of input, which is 34 in this experiment. Each bit of individual represents whether this feature would be used or not, where 1 represents yes and 0 represents no. And then I use these selected features of training data set to train the network. The fitness value is represented by test accuracy of the trained network. And then the population will do crossover, mutation and selection and combine the previous population to generate a new population. The end condition is when the number of individuals in the population is over 1000. The reason why I set 1000 as an end condition is that I do not want the experiment to be very time consuming. And for the same reason, I change the epoch number of training from 20000 to 200.

The function that I used in crossover operation is "one point" function, which randomly select one position to cut the parent strings and copy different areas to children string. In mutate operation, I set the mutate possibility with 0.1, so every bit in children generated by crossover operation has 0.1 of possibility to mutate. In selection operation, I select individuals from present population based on the fitness value. And fitness values are the accuracies of each generation after training.

The performance of the network is represented by the mean accuracy and max accuracy of each generation. By comparing the accuracy of neural network which does not introduce GA, the result can be observed.

3 Result and discussion

3.1 Result

a) Part 1

After the implementation of experiments, I obtain some results. The result of training neural networks with training data set can be described in the table 1 bellowed. Different column represents different hidden nodes in the hidden layer, including 1, 5, 10, 15 respectively. Every row of the table represents the accuracy after every 1000 epoch.

table 1 Accuracy with different number of hidden layers during training process (%)

hidden node(s)	One	Five	Ten	Fifteen
Epoch [1000/20000]	50.50	49.50	68.00	67.00
Epoch [2000/20000]	50.50	47.00	72.00	70.50
Epoch [3000/20000]	54.00	69.50	72.00	71.50
Epoch [4000/20000]	68.50	73.50	72.50	70.50
Epoch [5000/20000]	73.00	75.00	73.50	72.00
Epoch [6000/20000]	73.50	75.00	75.00	73.00
Epoch [7000/20000]	73.50	75.00	75.50	73.50
Epoch [8000/20000]	74.00	75.50	76.00	74.00
Epoch [9000/20000]	74.00	76.50	76.00	74.50
Epoch [10000/20000]	74.00	77.00	76.00	76.50
Epoch [11000/20000]	75.00	77.00	76.50	77.00
Epoch [12000/20000]	75.50	77.00	77.00	77.00
Epoch [13000/20000]	75.50	77.00	77.50	77.00
Epoch [14000/20000]	75.50	77.50	77.50	78.00
Epoch [15000/20000]	76.00	76.50	78.00	78.00
Epoch [16000/20000]	76.50	77.00	78.50	78.00
Epoch [17000/20000]	76.50	78.00	79.50	78.50
Epoch [18000/20000]	77.00	79.00	79.50	78.50
Epoch [19000/20000]	77.50	79.50	79.50	78.50
Epoch [20000/20000]	78.00	79.50	80.50	79.00

Using the diagram to demonstrate the change of accuracy as the increase of epoch is more intuitive. The figure 1 below displays the same data as table 1. In the figure 1, the abscissa axis represents the epoch from 0 to 20000, and the ordinate axis refer to the present accuracy after every 1000 epoch.

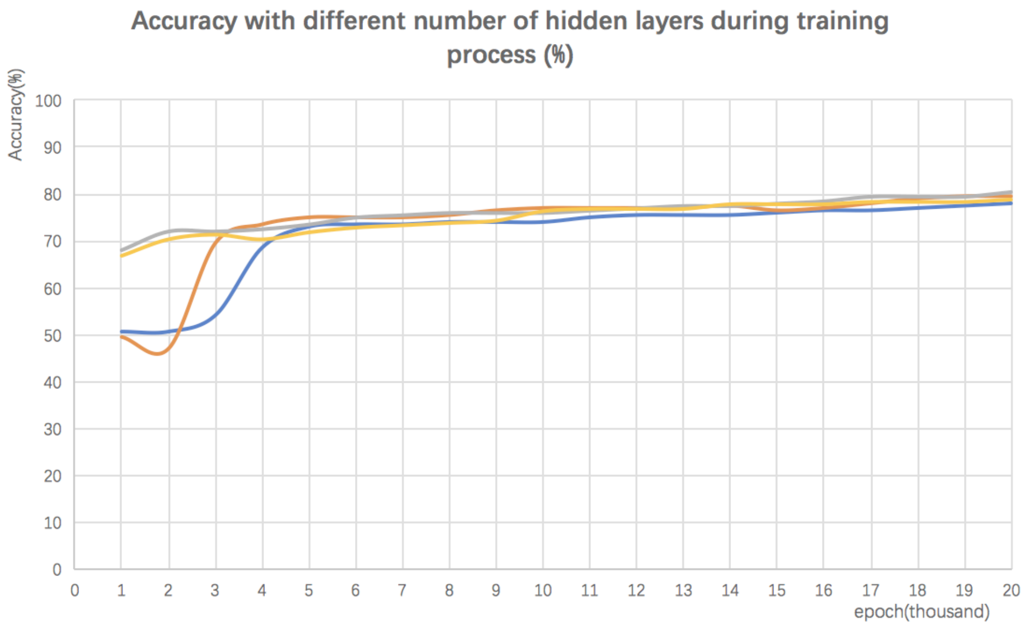

Fig. 2. Learning curves of neural networks with 1 hidden node (blue), 5 hidden nodes (orange), 10 hidden nodes (grey) and 15 hidden nodes (yellow)

b) Part 2

The mean accuracy and max accuracy of each generation after introducing GA is as table 2 shows. The mean accuracy and max accuracy of generation 0 represents the result pf initial neural network without GA. Other results of generation 1-7 demonstrates different mean accuracy and max accuracy after introducing GA.

table 2 Mean accuracy and max accuracy of each generation (%)

Generation	Mean accuracy	Max accuracy
0 (without GA)	66.23	66.23
1	84.23	94.03
2	84.57	94.60
3	85.83	94.63
4	85.82	94.71
5	86.32	94.83
6	86.68	95.36
7	86.80	95.45

The figure 3 below demonstrates the same data with generation 1-7 in table 2. The blue line represents mean accuracy of each generation and the orange one represents max accuracy. From figure 3, the change of accuracies can be observed more clearly.

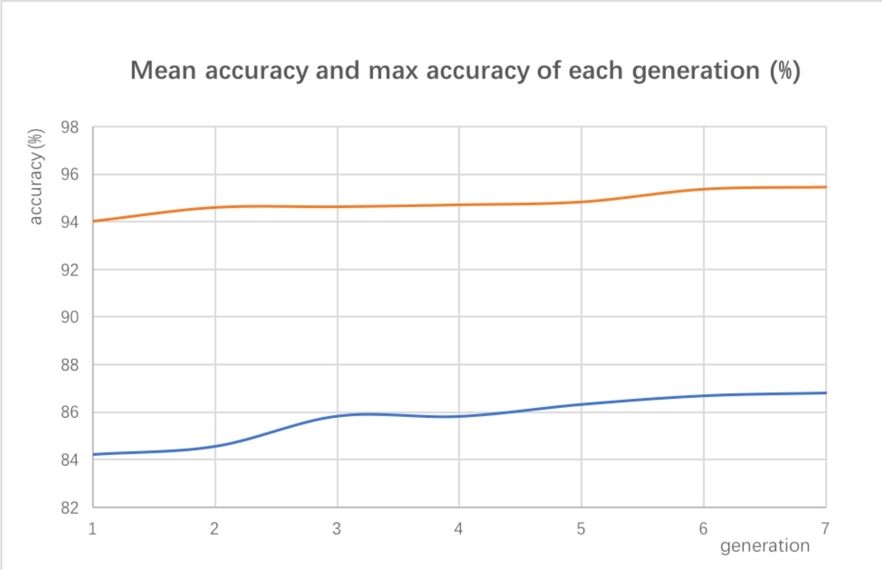

Fig. 3. Mean accuracy (blue) and max accuracy (orange) of each generation

3.2 Discussion

At the part 1 of this experiment, the first step is to train the neural network with the training data set. The neural networks are set to have 1, 5, 10 and 15 hidden neurons respectively and the result of this experiments reflect performance of the networks. Table 1 and figure 1 demonstrate the result. When the neural network is set to have 1 hidden neuron (as the blue line showing in the figure 1), the accuracy ended with 78.00% after 20000 epochs of training. And the neural network with 5 hidden neurons (as the orange line showing in the figure 1) shows the accuracy of 79.50% after training. In addition, the accuracy of neural network with 10 hidden nodes stops in 80.50% and 79.00% of total radar return is correctly predicted by the 15-hidden-node network.

The second step of this experiment is to pass the test data set to test the neural network. The test data set contains 27 bad radar returns and 124 good returns which is much more than bad returns. When using this data set to test the neural networks, the 1-hidden-node network has 64.90% of accuracy and the network with 3 hidden nodes show the accuracy of 80.79%. The highest accuracy shows in the 10-hidden-node network, which reaches 86.09% and 15-hidden-node network has 83.44% of accuracy, which is less than 10-hidden-node neural network.

From these results we can see that in the training process the final accuracy of the four sample all stopped around 80 and also have the trend of increase. And the test process result of the last 3 sample are more superiority, which all reaches 80% of accuracy. Therefore, the neural network is suitable for the classification job with appropriate hidden nodes.

In addition, from these results we can find that the performance of the neural network changed as the increase of the hidden nodes number. In the training process, as for the first three sample, the accuracy is improving with the increase of the hidden nodes from 78% to 80.5%. But the accuracy dropped back to 79% when the neural network has 15 hidden nodes. From the result, we can see that the performance of neural network does not has a positive correlation with the number of hidden nodes, because we can find that the 10-hidden-node neural network with a higher accuracy obviously perform better than 15-hidden node network.

The result of experiments conducted by Sigillito, Wing, Hutton and Baker in their paper can also can come into similar conclusion. In their paper, Sigillito, Wing, Hutton and Baker conclude that the neural network suit the classification job of radar returns because the accuracy of 5-hidden-node multilayer feedforward network can reach nearly 100%. In addition, from the figure 3 given by their paper, we can see that the performance of 15-hidden-node neural network is worse than 10-hidden-node network, even worse than 5-hidden-node network.

The reason why the increase of hidden neurons cannot always improve the performance of neural network may be two. Firstly, the increase of hidden neurons makes the computing more complicated, which results in a worse performance. The second reason may be too many hidden neurons can make the model easier to overfit.

At the second part of this experiment, we can see that neural network which is not based on GA has only accuracy of 66.23%. However, neural network based on GA performs much better. The accuracies are all over 80%. The mean accuracy of last generation reaches 86.8% and the max accuracy even reaches 95.45%. In addition, we can see that as the algorithm goes, the mean accuracy and the max accuracy both increase, which means GA can improve the performance of neural network. Because I did not set a very big number of population as end condition, I suppose that the accuracy would continue increase after more generations.

Boutorh & Guessoum[4] also made the same conclusion in their paper. They use GA to improve the neural network which is used to deal with the problem of SNP selection and classification. They claim that neural network with GA shows a higher performance where the accuracy can reach almost 100%.

4 Conclusion

Through the experiment, I find that the radar return data set can be used to train the neural network. With appropriate hidden nodes in the hidden layer, the accuracy can reach a satisfied value. Therefore, neural networks are reliable to do the classification job and they can save much time compared with doing this job manually. The first goal about proving the reliability of neural network is reached.

As for the second goal, by passing the training data set I have found that the neural network with 15 hidden nodes does not perform better than that with 10 hidden nodes. Therefore, the increasing number of hidden node cannot certainly improve the performance of neural network. The result of test data set can also prove this conclusion. Sometimes, too many hidden nodes are a barrier to train a network. Therefore, finding an appropriate number of hidden nodes can improve the performance of neural networks. The reason may lie in that too many hidden neural would make the computing complicated and increase the possibility of overfitting.

The experiment also proves that genetic algorithm can help to improve the performance of neural network. Because the neural network with GA shows better mean accuracy and max accuracy. In addition, as GA goes, these accuracies increase.

5 Future work

Although the goals of the experiment are met, there are still many extended works to do because this experiment exists many shortcomings.

The first problem is that the sample of my experiment is too small. I only test 4 different networks to describe the relationship between performance of neural network and the number of its hidden neurons. Therefore, I need conduct more experiment on different kind of neural network.

The second problem is that I haven't find an appropriate number of hidden nodes for this experiment. This will be the next step that I require to reach.

Another problem is that considering the time consuming, both epoch number of neural network and number of end population are set too small. Therefore, the result is not so convincing. I would like to continue to research on genetic algorithm's performance in the future.

References

1. Waugh, S.G., George, S.: Extending and Benchmarking Cascade-Correlation: Extensions to the Cascade-Correlation Architecture and Benchmarking of Feed-forward Supervised Artificial Neural Networks. Thesis (Ph.D.)-University of Tasmania. 1, 1--207 (1995)
2. Jahangiri, M., Razaghian, F.: Fault detection in analogue circuits using hybrid evolutionary algorithm and neural network. Analog Integrated Circuits and Signal Processing. 80, 551--556 (2014)
3. Sigillito, V.G., Wing, S.P., Hutton, L.V., Baker, K.B.: Classification of Radar Returns from the Ionosphere Using Neural Networks. Johns Hopkins APL Technical Digest. 10, 262--266 (1989)
4. Boutorh, A., Guessoum, A.: Complex diseases SNP selection and classification by hybrid Association Rule Mining and Artificial Neural Network—based Evolutionary Algorithms. Engineering Applications of Artificial Intelligence. 51, 58--70 (2016)
5. Schwenk, H., Bengio, Y.: Boosting Neural Networks. Neural Computation. 12, 1869– 1887(2000)