Improvement of Neural Network in Radar Signal Classification

Zhou Shen

Research School of Computer Science,
The Australian National University
u6342291@anu.edu.au

Abstract. This paper focuses on applying classification on a radar signal data set. A feedforward neural network was used to classify different states of signal data. Genetic algorithm was used for feature selection to get an optimal feature set for classification, but the accuracy of result of feature selection did not have obvious improvement at last. Threshold modification was used to improve accuracy of result. However, since ReLU function was applied in neural network instead of sigmoid function, threshold modification did not help improve result. Confusion matrix and loss function were used for result evaluation. The result is 98.51% training accuracy and 96.64% testing accuracy, but it is worse than result of paper of Sigillito (1989) which has 100% training accuracy and 98% testing accuracy.

Keywords: Radar signal; Neural network; Classification; Genetic algorithm; Feature selection

1 Introduction

Radar signal is widely used in many fields, according to whitepaper of Altair, people need radar signal for navigation, meteorologists use radar signal to monitor precipitation and wind, police forces use radar guns to monitor vehicle speeds on the roads. We can say that the usage of radar signal has become a very important part of our daily life.

However, radar signal might be easily affected by a variety factors during its propagation. Thus, we need to classify those returned signal as whether it is useful for application or not.

Nevertheless, the classification which requires a great deal of calculations and time, thus we need an advanced method to help us classify it easily, and neural network method might be applicable to this problem.

1.1 The dataset

In this report, a dataset made by Vince (1989) about radar signal from ionosphere is used. Its task is to solve a classification problem which is about classifying radar data from ionosphere into good state or bad state.

It is made of 17 pulse numbers and one state describer. Each of the 17 pulse numbers is described by two attributions, corresponding to the values that returned by the function resulting from the complex electromagnetic signal. The state describer stores whether the state of radar signal is "good" or "bad", and if the state is "good", it means the signal is valid to analyze, meanwhile, if the state is "bad", it means the signal is invalid.

The radar signal dataset shows a typical binary classification problem which can be solved by neural network easily. Meanwhile, most of its attributions are continuous, which means they can be intuitive and easy to understand.

2 Method

In this section, multiple methods for constructing neural network as well as techniques that used to improve its performance will be shown.

2.1 Preprocessing

As is mentioned above, this dataset of radar signal from ionosphere has 35 attributions, including 34 numeric type pulse numbers which represents 17 pulse numbers and 1 string type state which represents the state describer.

Since the type of the last attribution is string which cannot be recognized by the neural network developed by pytorch. To make this dataset be practical in my neural network, I encoded the state attribution to numeric with 0 to represents "bad" state and 1 to represents "good" state.

Those 34 attributions has been normalized to the range [-1, 1] in the original dataset.

2.2 Implement of neural network

I used feedforward neural network which developed by pytorch. The neural network comprises one input layer, one hidden layers and one output layer, and it is a fully connected neural network.

I chose the first 34 attributions which represents pulse numbers as features of classification and the last attribution which represents state of radar signal is selected as the target.

Since there are 34 features, and 2 kinds of target, I use 34 input neurons, 5 hidden neurons and 2 output neurons at the beginning.

2.2.1 Number of hidden neurons

For activation function, according to Jordan(1995), "for binary (0/1) targets, the logistic function is an excellent choice", thus I chose sigmoid function at first.

According to Chio(2016), the number of hidden neurons does affect the model performance. Thus, I need to take it into consideration.

The plot is shown below.

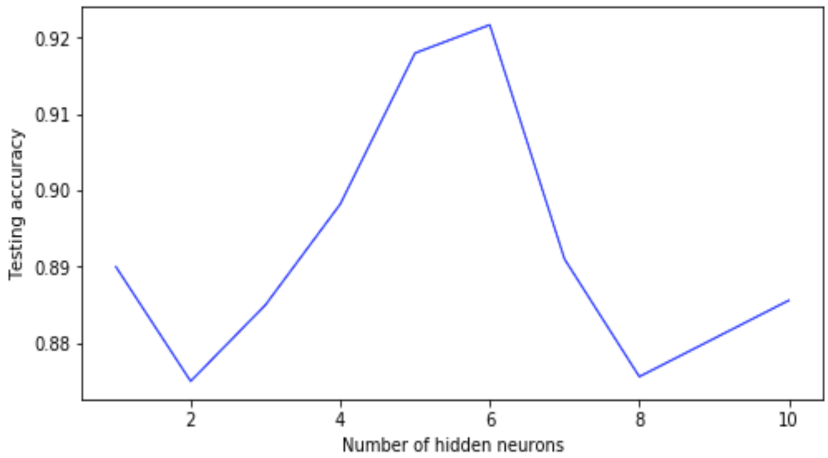


Fig. 1. The plot of Testing accuracy of different number of hidden nerons.

From the plot shown above, we can find that the highest testing accuracy belongs to 6 hidden neurons, thus from this step, the number of hidden neurons is 6.

2.2.2 Activation function

Although Jordan chose sigmoid function to deal with binary targets, I think that the paper and its method might be out of date today. Therefore, I wanted to compare sigmoid function with other activation functions to find a better choice.

ReLU function is another commonly used activation function in neural network which is defined as

$$f(x) = \max(0, x),$$

and compared with sigmoid, it is faster, more biological inspired and has less chance of vanishing gradient. To find out whether it is also suitable for binary targets, I recorded it results to compare with sigmoid results.

The bar chart of comparison is shown below.

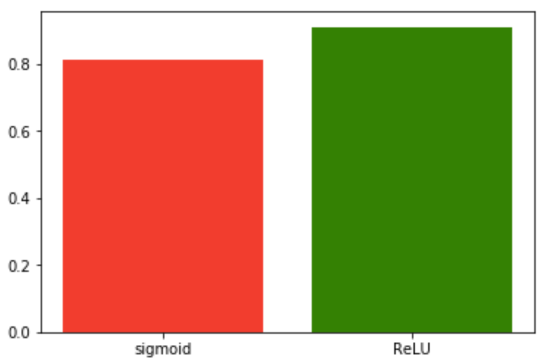


Fig. 2. The bar chart of testing accuracy of sigmoid function and ReLU function.

From this graph, we can find that ReLU works better in this neural network than sigmoid function.

However, according to Karpathy's CS231n course, ReLU units can be fragile and can be "dead" during training if learning rate is set too high.

Thus, a suitable learning rate is also necessary for my neural network.

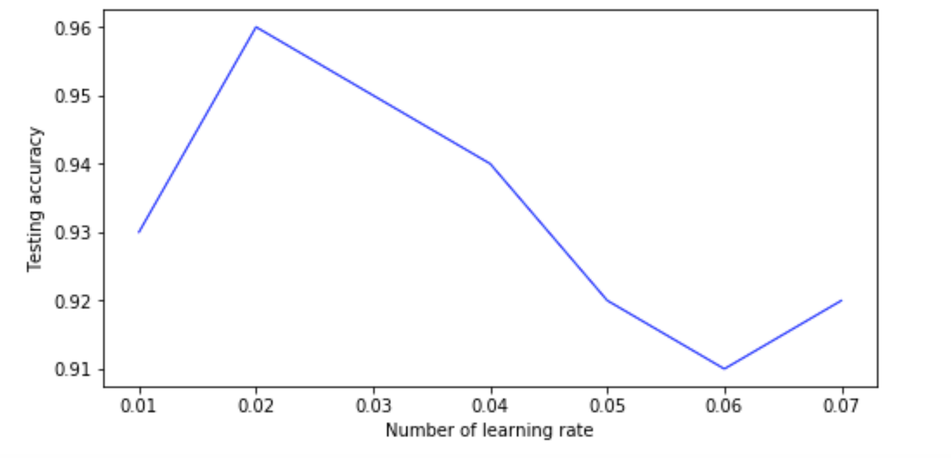


Fig. 3. The graph of testing accuracy of different value of learning rate.

The graph above shows that 0.02 is the best learning rate for this neural network, thus I will reset my learning rate to avoid weakness of the ReLU function.

2.3 Evolutionary algorithm

Evolutionary algorithm (EA) is heuristic, and it is useful for solving problems that might be hard to solve in polynomial time. According to Pranya (2016), “EAs have been widely used due to its many advantages over classical search and optimization techniques”, for example, it is simple to be used , its usable range is flexible and it has ability to solve problems without any human expertise.

Genetic algorithm is one kind of EA. According to Chih-Fong (2012), “In general, the genetic information (i.e., chromosome) is represented by a bit string (such as binary strings of 0s and 1s), and sets of bits encode the solution. Genetic operators are then applied to the individuals of the population for the next generation (i.e., a new population of individuals).” Crossover and mutation are two mostly used operators, which can create two offspring strings from two parent strings and can change the value of a single bit randomly respectively. Moreover, a fitness function is needed to judge the quality of each individual, and then those recorded quality values can be used to lead result towards an optimal solution.

2.3.1 Application

According to Jason (2014), “feature selection methods aid you in your mission to create an accurate predictive model. They help you by choosing features that will give you as good or better accuracy whilst requiring less data”. Choosing good features might help us get more accurate results, thus I thought about developing a method to help us get best features. As genetic algorithm can be used to get optimal solution, I thought it can be used for feature selection.

For this paper, I used a simple genetic algorithm with the following parameter settings:

- Population size: 50
- Number of generation: 10
- Probability of crossover: 0.5
- Probability of mutation: 0.001

2.3.2 Optimal feature subset

I ran genetic algorithm for several times, I got three different optimal choices, and the individuals (DNA) of optimal choice in different runs are shown below:

Table 1. Optimal choices.

No.	DNA
1	1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1
2	1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1
3	1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1

According to the optimal choices, first time, I dropped the 10th, 11th, 12th, 17th, 20th, 22th, 26th, 28th, 29th, 31th and 33th features. Second time, I dropped the 7th, 10th, 12th, 15th,16th, 23th, 24th, 25th, 29th, 31th and 32th features. Third time, I dropped the 6th, 10th, 15th, 16th, 20th, 21th, 26th, 27th, 29th, 30th and 33th features. The accuracy of them are shown below:

Table 2. Results of different try.

No.	Testing accuracy
1	93.28%
2	91.72%
3	94.96%
All features	96.64%

The table above shows that all of testing accuracy of those three choices are lower than the testing accuracy using all features. This result shows that feature selection might not be helpful to this data set.

From my point of view, all of features of my data set, which are all pulse numbers of radar signal, might contribute to classification of radar signal.

2.4 Evaluation method

In this research, I use the evaluation method used in the paper of Gedeon which uses confusion matrix to show both false negative and false positive as well as true negative and true positive, and training accuracy and testing accuracy are also used in this research to judge results.

2.5 Improvement

As Gedeon showed in his paper, "modifying the threshold in accordance with the results on the validation test set, can be used to modify the neural network result to minimize the false positive results while keeping correct results as high as possible". This technique might be helpful to improve my neural network.

Results are shown below:

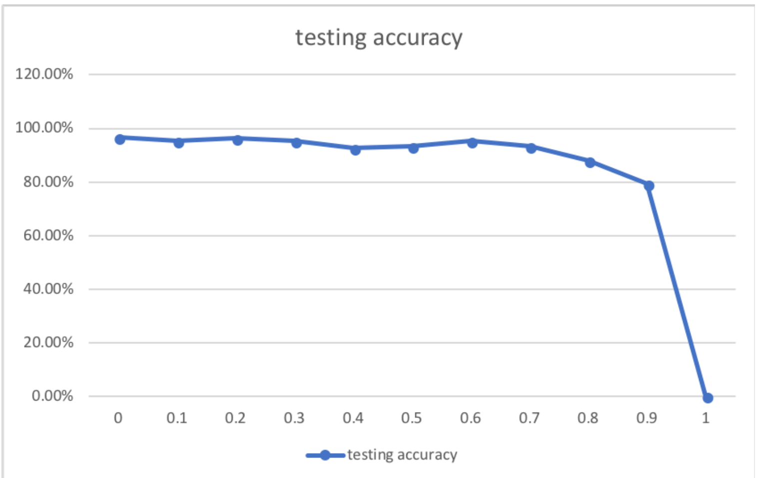


Fig. 5. Testing accuracy of different threshold value.

However, unlike what was imagined, the test accuracy of neural network decreases. It seems that the technique of modifying threshold did not work on our neural network.

According to Karpathy's CS231n course, the ReLU function is simply thresholded at zero. From my point of view, since the ReLU function has unchangeable threshold itself, setting a threshold before the function might be meaningless or even have negative effect on training.

Thus, this technique developed by Gedeon might not suitable for my neural network.

3 Results and Discussion

3.1 Results

At last, I use a neural network including one input layer with 34 input neurons, one hidden layer with 6 hidden neurons, as well as one output layer with 2 output layer.

The learning rate is 0.02, and the activation function used is ReLU.

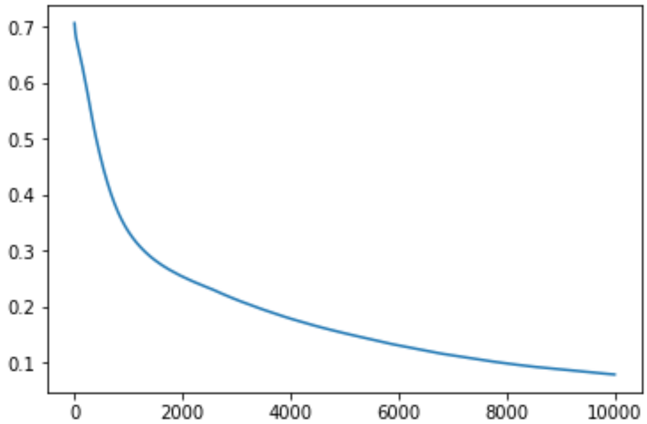
After using a variety of methods to improve performance of neural network, with 200 train data and 150 test data, I get the results of the classification task.

Table 3. Results for 2 different epochs.

Number of Epoches	Training accuracy	Testing accuracy
8000	97.86%	95.81%
10000	98.51%	96.64%

The best result is shown below, which has 98.51% training accuracy and 96.64% testing accuracy is.

```
Epoch [9951/10000] Loss: 0.0792  Accuracy: 98.51 %
```




```
Confusion matrix for training:

 99   1
  2  99
```

Fig. 6. The testing accuracy and the confusion matrixes for training.

```
Testing Accuracy: 96.64 %

Confusion matrix for testing:

 25    1
  4  119
```

Fig. 7. The testing accuracy and the confusion matrixes for testing.

3.2 Discussion

The 98.51% training accuracy shows that neural network learned well in this data set, and testing accuracy of 96.64% shows that most radar signal data were classified correctly with only 5 mistakes. It shows that neural network is proven to be suitable for radar signal classification.

The confusion matrix for training shows that there was one false positive and 4 false negative results, which means one “bad” signal was considered as “good” signal and 2 “good” signals were predicted as “bad” signals. The confusion matrix for testing shows that there was one false positive and 4 false negative results, which means one “bad” signal was considered as “good” signal and 4 “good” signals were predicted as “bad” signals.

In common research, both false negative and false positive results need to be reduced, however, this study is different. Since “bad” signals are useless, those true negative and false positive results will be dropped before being used in further analysis, and only false negative results will negatively affect further analysis of those radar signals. Therefore, we need to reduce number of false negative results to avoid negative effect on further analysis.

Above all, good accuracy means neural network can classify different states of most radar signal data successfully, but as mentioned above, future work is still needed to improve quality of the classification.

3.3 Comparison of another paper

Sigillito(1989) also used the same dataset in his paper, however, his result is better than mine since some different methods are taken in his neural network, and he achieved 100% accuracy on the training set and 98% accuracy on the testing set.

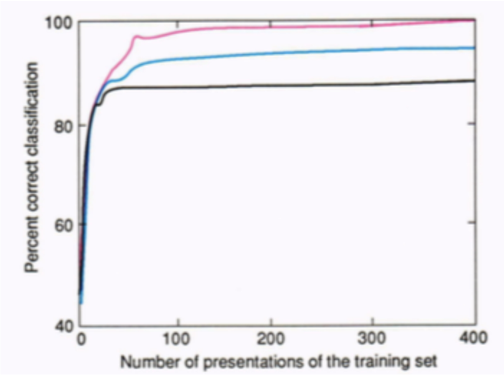


Fig. 8. Network learning curves for the linear perceptron (black), the nonlinear perceptron (blue), and multilayer feedforward network with sigmoid function and five hidden nodes (red). From paper of Sigillito(1989).

3.3.1 Differences

First, he used sigmoid function with modified threshold which did not work in my neural network. I believe that although I have tested that sigmoid function is weaker than ReLU function in my neural network, sigmoid function with modified threshold might work better.

Second, the number of epochs in my neuron network is 10000, but the value of loss has not been saturated, it still has space to increase. Although Sigillito did not tell us in his paper how many epochs had he made, his training accuracy has climbed to 100%, thus I think his number of epochs is bigger than mine.

3.3.2 Discussion

Although I mentioned some differences in section 3.3.1, I cannot compare different improving methods between my work and theirs, since they only modified several paraments of neural network and did not apply methods to improve it.

As they used an unimproved simple neural network, I think the reason why their result is better than mine is that they used a different evaluation method. However, they did not mention their evaluation method in their paper, thus I cannot give a clear conclusion about the reason why their result is better than mine.

4 Conclusion and Future Work

In this research, I have used radar signals from ionosphere as an example to classify different states of radar signals, and by using that dataset, I have demonstrated that neural network is suitable for classifying radar signals. I also used feature selection with genetic algorithm and result showed that feature selection might not be helpful for this data set. Furthermore, I have shown that the technique of modifying threshold might not suitable for a neural network using ReLU activation function.

The target of our classification is to find “good” signals which can be used in further analysis. As what was mentioned in section 3.2, negative false result need to be reduced in future work to improve classification quality. From my point of view, there are two methods to improve accuracy of “good” signals prediction.

First, we can improve total predicting accuracy to reduce number of both negative false and positive false results.

- Try better activation function like leaky_ReLu, which might be useful to improve total accuracy.
- Use more training data might be helpful, since the total number of instances of data set is around 300, which might be too small.
- Add more hidden layers.

Second, we can find way to only reduce negative false results even it will increase number of positive false results. Although Gedeon only used threshold to minimize number of false positive results, the study of Allision (2018) shows that suitable value of threshold can also lead to reduction of false negative results. Thus, we can try this method in future work.

Reference

1. Altair: Radar Application in Everyday Life. https://web.kamihq.com/web/viewer.html?source=extension_pdfhandler&file=https%3A%2F%2Fcdn2.hubspot.net%2Fhubfs%2F47251%2F2016_Germany%2F2016_FEKO%2FWhite_paper_Radar_every_day_use_en_web_A4.pdf.

2. Vince Sigillito: Ionosphere Data Set. https://archive.ics.uci.edu/ml/datasets/ionosphere.(1989)
3. Jordan, M. I. :Why the logistic function? A tutorial discussion on probabilities and neural networks, MIT Computational Cognitive Science Report 9503 (1995)
4. Karpathy, Andrej :Convolutional Neural Networks for Visual Recognition, http://cs231n.github.io/neural-networks-1/ (2018)
5. Sigillito V G, Wing S P, Hutton L V, et al. :Classification of radar returns from the ionosphere using neural networks, Johns Hopkins APL Technical Digest 10(3): 262-266 (1989)
6. Chio, Eric : Impact of number of hidden neurons to model performance, https://github.com/log0/digit_recognizer_2/blob/master/notebooks/impact_of_number_of_hidden_neurons_to_model_performance/Impact%20of%20number%20of%20hidden%20neurons%20to%20model%20performance.ipynb (2016)
7. L.K. Milne1, T.D. Gedeon1 and A.K. Skidmore : CLASSIFYING DRY SCLEROPHYLL FOREST FROM AUGMENTED SATELLITE DATA: COMPARING NEURAL NETWORK, DECISION TREE & MAXIMUM LIKELIHOOD
8. Chih-Fong Tsai, William Eberle and Chi-Yuan Chu: Genetic algorithms in feature and instance selection, Knowledge-Based Systems, Volume 39, Pages 240-247, ISSN 0950-7051.(2013)
9. Jason Brownlee: An Introduction to Feature Selection, Machine Learning Mastery. https://machinelearningmastery.com/an-introduction-to-feature-selection/ .(2014)
10. P.A. Vikhar: Evolutionary algorithms: A critical review and its future prospects, 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC). (2016)
11. Allison Ashenfelter: Case Study: Reducing False Negatives With Operating Thresholds. https://dzone.com/articles/case-study-reducing-false-negatives-with-operating. (2018)