Comparing Pattern Removal Algorithm with Genetic Algorithm in Classification Problem

Jingjing Shi
Research School of Computer Science
Australian National University
U5792801@anu.edu.au

Abstract. With the artificial neural network developing, the data set becomes more and more important since it determines the network's accuracy. If there are too many noisy patterns or useless features, they may affect the network to have a slow learning time or make the network over-fitting by escalating training time. I here use ionosphere data set to achieve binary classification problem by implementing noisy patterns removing approach, genetic algorithm and combining them in neural network and comparing these results. Meanwhile, using genetic algorithm could provide the best testing accuracy and the combination of two algorithms can accelerate the convergence. In addition, this paper compares with the other paper which used the same data set.

Keywords: erroneous, neural network, remove, binary, genetic algorithm, feature selection, classification

1 Introduction

1.1 Background

With continued research and the technology developing, the neural network has a great breakthrough in signal processing, such as Lippmann described several approaches to signal processing in an introduction to computing with neural network [1], Recchione and Russo [2] used feed-forward neural network system to detect and characterize sonar signals with the characteristic spectrogram textures and so on. Obviously, using neural network to deal with signal processing is a topical subject in the world now. On the other hand, I am interested in neural network and signal processing as well. So, for this paper, I choose ionosphere dataset in UCL [3] and mainly focus on binary classification problem in order to discriminate "good" or "bad" radar returns by using several deep learning methods in the artificial neural network. The reason I chose classification problem is that it is close to our life, everywhere has classification problem, such as discriminate cat or dog and garbage collection (recycling or garbage).

1.2 Problem Define

When doing investigation, Sigillito, Wing, Hutton and Baker in classification of radar returns from the ionosphere using neural networks also used this data set. [4] They got a breathtaking result which the multilayer feed forward networks outperformed the single-layer networks, which achieves up to 98% accuracy on the testing set. However, they did not think about "erroneous" problem. This problem is inevitable since every data set could exist some incorrect data no matter how accuracy the data set is. It could cause serious result if ignoring this. On the one side, these inaccurate patterns may affect the network to have a slow learning time of the majority of patterns in order to learn these few erroneous patterns. On the other side, they seem to make the network over-fitting by escalating training time. [5] In order to reduce the influence by these erroneous patterns, many approaches are identified. Joines and White (1992) discovered and tested Least Median Squares(LMS) and Least Trimmed Squares(LTS) in improving generalization by using robust cost function [6]. Gedeon and Bowden found heuristic pattern reduction [7], bimodal distribution removal is discovered by Slade and Gedeon [5].

The word, "erroneous" inspire me, and push me to think about this problem from different ways. The approaches I list above are all based on removing patterns, in other words, removing rows. How about removing features(columns)? Do they provide some breathtaking results? Or What about combing removing rows and columns together, does this will give us some breathtaking result as well or will cause over fitting?

1.3 Investigation Outline

Based on this data set, the first part of this paper will discuss the process of pre-processing dataset. Then, a detailed artificial neural network will be explained precisely and provide a new way to describe the performance of neural network (evaluating testing accuracy). Applying removing pattern algorithm (bimodal distribution removal) and Genetic Algorithm (feature selections) in the neural network respectively and comparing their results. Besides, combing these two algorithms together and try to find some breathtaking results or limitations. Also, this paper will compare the results with other research paper which used the same data set. In the end, a summary will be concluded for findings and future work.

2 Method

2.1 Data pre – processing

This data set came from Dermatology Data set, which in UCL Machine Learning Repository. [3] The original data set has 34 attributes and 351 instances. The past usage is to discriminate "good" from "bad" radar returns from the ionosphere [4], so the values in last column are "good" and "bad". In order to fit classification problem, I use "1" to represent "good", and "0" means "bad". The process is done manually in ionsphere.csv. In addition, the content in the second column are all "0", I deleted it in the program since it influenced the column's normalization but did not influence the result.

2.1 Neural Network

For bimodal distributional removal algorithm, the network here I used a feed-forward neural network of three layers which comprise an input layer, a hidden layer and an output layer. All connections from one layer to the subsequent one are linearly, with no lateral, backward or multilayer connections. The number of inputs is based on the features in the data set, and the number of outputs is in terms of the target numbers in the data set.

2.1.1 Hidden Neuron

There is no rule to consider how many hidden neurons needed in the neural network. But in general, the optimal number is between the output neurons' number and the input neuron's number. In addition, the number of hidden layers depends on the size of data set. If the data set is enormous, multilayer is needed. On the contrary, one or two is enough. The reason is simple since there is information loss from one hidden layer to the subsequent one.

2.1.2 Sigmoid Function

In this neural network, sigmoid function is used in hidden layer as activation function, which it is $\sigma(z) = \frac{1}{1+e^{-z}}$.
Here are some advantages:

1. the most important one is that sigmoid function squashes the input to the range 0 and 1. This is suitable for classification problem and predict the probability.
2. The derivative of sigmoid function is simple and less computation.

2.1.3 Stochastic Gradient Descent

Stochastic gradient descent is to optimize the model and minimize the loss function. Comparing with other gradient descent methods, this is the most efficient one since it tries to use one data point to get an approximate gradient rather than an exact gradient. That saves time a lot.

2.1.4 Confusion Matrix

In order to describe the performance of the neural network, the confusion matrix is added. The matrix size only depends on the target numbers, for example, if the target number is 2, the matrix size is 2x2, if the target number is 3, the matrix size is 3x3 and so on. In the matrix, row represents the actual number and predicted number reflects in column. If the number only appears on diagonal, that means the network has high accuracy, and every object can be discriminated. In contrast, if the number appears in other position, apparently, the object can not be discriminated correctly.

2.1.5 Definition of Error

On the original paper, the writes did not have a precise definition of error. They only say "… the errors for all patterns in the training set were…" [5]. Based on the understanding of the neural network, I assume the error here is to use the Cross-Entropy function to calculate each pattern's loss between the predicted number and target number.

2.2 BDR Algorithm

1. In order to use the data set in the neural network, the first thing I need to do is to pre – processing the data set. Because the target of the original data set is either 'b' or 'g' and the problem which I deal with is the classification problem, the target need to convert into the binary ('b' == 0, 'g'==1).
2. Randomly split data into training set (60%) and testing set (40%)
3. Split training data into input (before the last column) and target (the last column)
4. The input patterns passes through the neural network which I built before
5. The cross entropy function to calculate each pattern's loss between the predicted number and its associated target number, and store them into a list

6. Calculate the list's mean, $\overline{\delta}_{ts}$ and variance, v_{ts}
7. If the error in the list is larger than $\overline{\delta}_{ts}$, take it out, and store it into a sub set
8. Calculate the subset's mean, $\overline{\delta}_{ss}$ and standard derivative, $\overline{\sigma}_{ss}$. The error in this subset are potential erroneous patterns since the mean is dominated by the large error.
9. If the error in the subset > $\overline{\delta}_{ss}$+ a* $\overline{\sigma}_{ss}$ (0<a<1) (5), remove its relative patterns, otherwise, continue to training
10. When v_{ts} is smaller than a constant number, stop removing (Actually, this number is relevant to the data set size and the its content)

2.3 Genetic algorithm

The core concept of genetic algorithm is based on Darwinian evolutionary theory, which is "survival of the fittest". The meaning of these four words is that the individuals with the best characteristics are more likely to survive and reproduce in natural selection. [8] If we use this concept in computer science, there are separated into six main parts, the flow diagram below represents these relations.

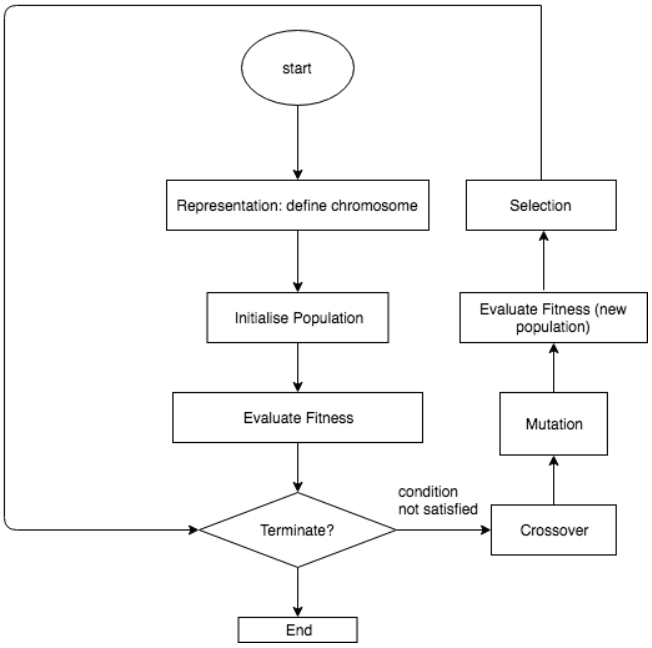

Figure 1: the flow diagram represents the main steps of genetic algorithm.

I use feature selection in genetic algorithm to train the neural network. Next, I will explain each part in detail.

2.3.1 Representation:

Representation is used here to define the chromosome. Chromosome represents the characteristic of individuals with long string information. In biology, it expresses the genome and every single variable in the long string represents gene. In this dataset, I set 34 attributes as chromosome.

2.3.2 Initialize Population

The first step of genetic algorithm is to initialize population. The initialization is randomly and making sure each individual has some characteristic. For achieving feature selection, I use "0" or "1" to represent each attribute's status: "1" represents the selected and "0" represents not selected. These binary numbers generate randomly and form a long string. In this data set, there are 50 chromosomes, so the program generates 50 binary strings, and each of them has 34 binary numbers initially. Also, all of these are different in order to maintain the features diversity and help to convergence quickly in the future to find the best solution. On the other hand, this process can be regarded as encoding.

2.3.3 Evaluate Fitness

The fitness function is the core thing in genetic algorithm, and we need to find it carefully since it not only influences the selection probability, crossover and mutation but also determines the convergence speed and whether or not the best solution can be found. I use the neural network testing accuracy as fitness function because our goal is to discriminate the good radar returns from the ionosphere as much as possible.

2.3.4 Terminate

This terminology is easy to understand. If the population it found satisfies some conditions, the evolution stop, otherwise it will undergo crossover and mutation to generate new population which more fit the fitness function.

2.3.5 Genetic Operators

In order to generate a second generation population of solutions from parents, a combination of genetic operators is used, they are crossover and mutation.

2.3.5.1 Crossover

Crossover is a process that taking more than one parent solution and producing a child solution from them [9]. In this genetic algorithm, the parent breeds with new individual who is randomly generated to get the new child.

2.3.5.2 Mutation

Mutation is used to maintain genetic diversity from one population to next. Each gene has some probability to mutate. In this neural network, the mutation algorithm is "0" to "1" or "1" to "0".

2.3.6 Select

Selection used here is to select individual genomes from population in order to later breeding. The high testing accuracy of the genome has, the high probability it will be chosen. In addition, this process verifies Darwinian evolutionary theory, survival of the fittest.

3 Results and Discussion

Here I use ionosphere data set to realize the binary classification problem which helping to discriminate "good" or "bad" radar returns based on the existing data set. The patterns in data do not exist in common with others, and it is randomly split into training set (60%) and testing set (40%).

3.1 Comparing with other modules

This data is used to train and test the neural network several times under three different model, the first one is the basic network without any methods or algorithm, the second one is to use the pattern removing method, and the third one is to use the genetic algorithm.

	Model One: without any algorithm	Model Two: using pattern removing method(BDR)	Model Three: genetic algorithm (10 generations)
Training Accuracy (epochs 1000)	63%-70%	65%-85%	70%- 80%
Testing Accuracy (epochs 1000)	~65%	~65%	68%- 80%
Training Accuracy (epochs 1500)	63%-70%	65%-85%	70%- 84%
Testing Accuracy (epochs 1500)	~65%	~66%	70%- 84%

Table 1: demonstrates the different models' training accuracy and testing accuracy under different epoch number by using the same neural network.

Expectedly, the second and third models have better results compared with the first one. Having a comparison with first and second models, it is clear to show that training accuracy of model two is between 65% and 85%, higher than the basic neural network. However, the testing accuracy is stable and similar to the first one, or the difference is very small. The third model which used feature selection algorithm has the best results that the training and testing accuracy are on the same range whatever the epochs' number is 1000 or 1500 and its entire probability is higher than the model one.

The reason caused the second module has high accuracy is very straightforward since it removes the erroneous patterns during the training. The erroneous would prolong the learning time and influence the training accuracy. As for the board range, the reason is that according to the pattern removing mechanism, the pattern's error depends on the loss function in the neural network. And the neurons' weight is the major factor to dominate the loss function. The initial value for each

neuron's weight is different in every training, and the network can get different erroneous patterns based on it. Therefore, the accuracy has a board range.

As to the second model has similar testing accuracy with the first model, there are two reasons: the first one is that the removing approach only happened in training set. It could have some probabilities that there are some erroneous patterns in the testing set which they are not removed. The second reason is that maybe there are too many patterns removed and make the network over-fitting and has high variance.

The reason that caused the third model has the same range of testing accuracy and training accuracy is that it deletes un-selected features not only from training data set but also from testing data set. So, it used the same features in both data sets rather than like BDR algorithm that the erroneous patterns are only deleted in training data set. In addition, feature selection can help the neural network to remove the useless features and reduce the dimension in order to improve the performance and save training time. So, its accuracy is higher than others integrally.

After observing this table, I was a little surprised that the training or testing accuracy are not influenced by the number of epochs. In general, if set epochs' number too large or too small, the convergence will be effected, such as the neural network stops too early or too much training caused over fitting. However, when I increase 50% of epochs, the result is still similar to the previous one.

If we see Table 1 closely, we will find that the model three' range in training accuracy is slightly narrow down comparing with model two. Generally, if the training accuracy does not have vibrational change and the range is as small as possible, it means that the neural network has good convergence and the best solution can be found. Although in genetic algorithm, there is no guarantee to find an optimal solution on finite time, and several factors affect the convergence, such as the way of initializing population, the fitness function and so on, but it is slightly stable comparing with model two which used removing pattern algorithm since removing pattern algorithm is based on neurons' weight, the major determinant to determine the convergence in neural network.

3.2 Combing with two algorithms

After comparing three models respectively, I combined removing pattern algorithm with genetic algorithm in one neural network. However, the result was not the same as I imagined.

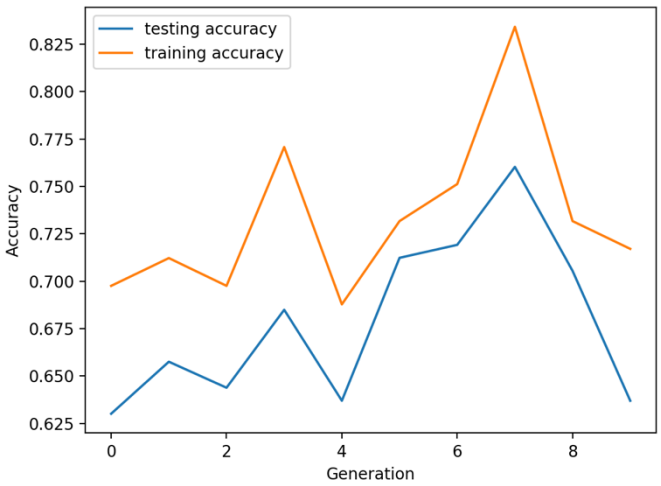

Figure 2: demonstrate the result by applying removing pattern algorithm and feature selection algorithm in one neural network under 10 generations. The orange line represents the training accuracy and the blue line is the testing accuracy.

From Figure 2, it shows that the range of accuracy fluctuation is small no matter on testing accuracy or training accuracy. This is unusual and surprised me. Before trying this method, I thought this range would be bigger since these two methods can not provide a sufficient way to find the best solution and convergence network. However, it is opposite. One thing that is expected is that the testing accuracy is lower than the model three and higher than the model two. This is because only feature selection algorithm can improve the testing performance.

3.3 Comparing with other neural network

In classification of radar returns from the ionosphere using neural networks, Vincent, Simon, Larrie and Kile concluded that the accuracy of multilayer feed – forward network (MLFN) outperformed the single layer networks, which 100 % accuracy on the training set and 98 % on the testing set [4]. They referred to network with hidden neuron as MLFN. I don't think their method and mine have comparability since that paper published in 1989, the neural network did not mature during that time, and people had less knowledge of that area.

4 Conclusion and Future Work

Overall, in this paper, three models with different algorithms were compared mutually and found pattern removing algorithm and feature selection algorithm could improve the training accuracy respectively and the feature selection algorithm could improve the testing accuracy as well. In addition, combing two algorithms could help to accelerate the convergence, but the testing accuracy was less than satisfactory. There are still some aspects which need to future discuss and research, such as is there any sufficient way to find the best solution by using genetic algorithm since the current technic for this algorithm does not support to find the best solution in finite time or how to use pattern removing algorithm in testing data set.

Reference:

1. Lippmann, R.D., (1987 Apr), 'An Introduction to Computing with Neural Nets,' *IEEE ASSP Mag*., 4-22

2. Recchione, Michael C., Russo, Anthony P. (1996), 'Feedforward neural network systems for the detection and characterization of sonar signals with characteristic spectrogram textures'

3. UCI Machine Learning Repository: Ionosphere Data Set. In: Archive.ics.uci.edu. http://archive.ics.uci.edu/ml/datasets/Ionosphere Accessed 30 May 2018

4. Sigillito, V. G., Wing, S. P., Hutton, L. V. & Baker, K. B., (1989), 'Classification of radar returns from the ionosphere using neural networks' Johns Hopkins APL Tech. Digest,10, pp.262-266

5. Slade, P., & Gedeon, T.D. (1993) June, 'Bimodal distribution removal'. In *International Workshop on Artificial Neural Networks* (pp.239-254). Springer, Berlin, Heidelberg.

6. Joines, M &White, M (1992), 'Improving generalization by using robust cost function,' *IJCNN*, Baltimore, vol.3, pp.911-918

7. Gedeon, T.D., & Bowden, T.G. (1992), 'Heuristic pattern reduction'. In *International Joint Conference on Neural Networks,* Vol. 2, pp.449-453

8. Gedeon T.D (2018) Introduction to evolutionary computer.

9. (2017) Crossover (genetic algorithm) In: En.wikipedia.org. https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm) Accessed 30 May 2018