A Binary Classification Method based on Genetic Algorithm and Network Reduction Techniques

Yuehan Zhao

74 Chandler Street, Belconnen ACT. U6417810@anu.edu.au

Abstract. Classification problem is one of the most important part in machine learning. Therefore, it is important to study how to use neural networks to solve classification problems. In this paper, I choose to classify the data set of Breast Cancer Wisconsin (Diagnostic). I use two methods for data classification and neural network optimization. First, I use genetic algorithm to do the feature selection. Genetic algorithm can be well applied to simplify data dimensions and reduce complexity. Secondly, I use Distinctiveness reduction on my neural network and compare my results with others. Experiments have verified the effectiveness of these two methods and have achieved satisfactory results.

Keywords: neural network, binary classification, genetic algorithm, network reduction techniques, feature selection

1 Introduction

In machine learning and statistics, classification problem is one of the most important part. Classification is based on the known data or observation belongs training set to identify which category a new observation should belong to. Classification is an instance of supervised learning. [1] Classification can be seen as two separate problems: binary classification and multiclass classification. [2] In this paper, only binary classification is mentioned. Binary classification only involves two classes. Binary classification can be used in many areas which include: Medical testing to identify if a patient has some disease or not; A "pass or fail" test method; Quality control, etc. There are many classification methods. Neural network is one of the most popular method to solve the binary classification problem.

Genetic algorithm is an adaptive global optimization probabilistic search method that evolved from the evolutionary law of the biosphere (survival of the fittest). It targets all individuals of a group and uses randomization techniques to efficiently search a coded parameter space. Strategies are taken to keep individuals with optimal fitness. Feature selection is one of the most important directions in machine learning. It targets at removing those irrelevant and redundant features. Genetic algorithm is useful in feature selection.[3]

Reduction technique in neural network is more and more important nowadays. Time complexity is an important factor when measure the performance of a network. However, many researchers use a neural network with a great number of hidden neurons, which may have good performance but take a lot of time. Therefore, how to reduce hidden neurons for a neural network is an important research direction.

1.1 Motivation

In some cases, it is the Contact Volume Editor that checks all the pdfs. In such cases, the authors are not involved in the checking phase.

My initial motivation of this study is to find out a proper way to solve a simple binary classification problem. Compared to multiclass classification, binary classification can be easier to understand and solve. Therefore, I think binary classification problem is easier for me to accomplish alone. In the given data set, Breast Cancer Wisconsin (Diagnostic) Data Set is the best choice for some reasons. First, a patient can only be diagnosed with either M(malignant) or B(benign). It will be a binary classification problem. Second, there is no missing data in the set and the data set is accurate. Third, there are 30 features in the whole data set and they are "computed from a digitized image of a fine needle aspirate of a breast mass" [4]. It's neither small or big dimensions and can be easily calculated and improved.

1.2 Model Design

In this study, I will use the most basic linear neural network to process the Breast Cancer Wisconsin (Diagnostic) Data Set. I generalize a simple three-layer feed-forward neural network and use back-propagation to calculate the loss and adjust the weights, and the activation function will be $y=(1-e^{-x})^{-1}$. Back-propagation is one of the most popular method used to train artificial neural networks. It is used in conjunction with optimization methods such as gradient descent. [5] There are 30 features in the Breast Cancer Wisconsin Data set, so I will choose genetic algorithm to do feature selection. Then, I will choose the distinctiveness reduction techniques to calculate the vector angle between hidden units, by this I can remove some undesired units without affecting the testing accuracy. [6]

1.3 Investigation

Before conducting this study, I made a number of investigations and finally selected the direction of the study. The first is the survey of feature selection. The process of selecting some of the most effective features from the original features to reduce the dimensionality of the data set is an important means to improve the performance of learning algorithms. Therefore, the selection and preprocessing of features become particularly important. I have read some papers including making rules for the student scores prediction. [7] But it might not be very easy for this data set. The features in Breast Cancer data set include radius, texture, perimeter and other parameters. It's difficult for me to judge the priority or threshold. Genetic algorithm is one of the most popular method to do feature selection, and it is simple and easy to implement. When choosing neural network optimization, I investigated several methods. As in the case of autoassociative topology, the weight value of the corresponding input or output is made equal, but this can be applied only to the same number of input and output, that is, suitable for image processing. [8] The database I choose is a csv file, so it is not suitable for such a method. At last, I chose to evaluate the similarity of hidden units to perform an algorithm to reduce the undesirable units.

2 Method

In this study, I use genetic algorithm to select features and use network reduction technique to prune the neural network.

2.1 Genetic Algorithm

The original dataset has 30 features, and some of them might not be important. So I want to select the most important features and to see whether the performance of the network can be improved. Genetic algorithm is one of the method that can help to remove the redundant or irrelevant features and it's easy to implement.

In machine learning, sometimes the main classification features are masked by redundant or irrelevant features. In order to eliminate such features, feature selection has become an important research direction. The feature selection process is to select some of the most effective and representative features from a set of features in order to achieve the purpose of reducing the feature space dimension.

Genetic algorithm is good for feature selection. Strategies are taken to keep individuals with optimal fitness. The selection is based on the fitness of the new individual. Genetic algorithm is based on the principle: the higher the fitness, the higher the chance of being selected. With low fitness, the chance of being selected is low. The initial data forms a relatively optimized group based on this selection process. After that, the selected individuals enter the crossover process to generate new individuals. The next step is mutation, which generates a new subset by mutation. Through this series of processes, a new generation of individuals is generated that is different from the original generation and is moving towards increasing overall fitness from generation to generation. Because the choice is to generate the next generation by selecting the better individuals, individuals with low fitness will gradually be eliminated. [9]

These are the steps in this paper.

Step1. Initialization:

A random method is used to generate N_0 initial strings $X_1^1, X_2^2 \cdots \cdots$ as an initial group, and each string is expressed in a binary code. Length is the number of features in the database. In this case, the length is 30. Each individual has a form like [0,1,1,1,0,0,$\cdots \cdots$ 0,1]. 0 represents that the feature is not selected. 1 represents that the feature is selected.

Step2. Calculate fitness:

According to the fitness function, I calculate each individual's fitness, and mark the individual with the highest fitness. In this case, I use logistic regression to get the accuracy of train data. Then the accuracy is used as the fitness function.

Step3. Selection:

The subset is selected from the parent group using the fitness ratio method. The probability that the child is selected is

$$P(x_k^i) = \frac{f\left(x_k^i\right)}{\sum_{i=1}^{N_0} f\left(x_k^T\right)} \qquad i=1,2,\cdots \cdots N_0$$

Step4. Crossover :

Selects two individuals from the subpopulation with the same probability and recombines these two individuals with the given probability P_c to generate two individuals and repeat the process.

Step5. Mutation:

The mutation operation randomly flips an individual bit according to the mutation rate P_m, generates a new individual, and repeats the process. Then merge the individual with the highest fitness value in step2 and finally form a new generation of groups. For example, before the mutation, the individual is [1,0,0,1,1,1,0]. Then after mutation, the individual might be [1,1,0,1,1,1,0]. Thus generating a new individual.

Step6.

If the program satisfies the termination condition, it stops the operation and outputs X as the approximate optimal individual. Otherwise, let k = k + 1, and go to step2.

After all these process, we'll get a best x_{k+1}^N with highest fitness. If x_{k+1}^N is [1,1,1,0,0,0], this means that the first, second and third feature is selected. The fourth, fifth and sixth feature is unselected.

Here is the pseudocode of genetic algorithm for feature selection.

```
Pc: probability of crossover
Pm: probability of mutation
M: population
G: generation

Initialize Pc, Pm, M,G and randomly generate the first group Pop.
    Calculate fitness of each individual in Pop F(i)
    Do{
          According to fitness ratio, choose two individuals from Pop.
          If (random (0,1)< Pc):
                Do crossover to two individuals
          If (random(0,1)<Pm):
                Do mutation to two individuals
    Add the two new individuals to the group new Pop
    Until meet G
```

2.2 Distinctiveness Reduction Method

The neural network with a great number of hidden units takes very long time to get results. Meanwhile, some of the hidden units might not be important. They might be replaced by others or can be removed. Therefore, I want to reduce the number of hidden neurons and improve the efficiency of the network. Distinctiveness reduction method is one of the method that can remove those undesirable units and might not influence the accuracy of the network.

Distinctiveness reduction methods focus on undesirable hidden units, which include those units have opposite functionality and those units that are too similar. Mentioned by Gedeon and Harris, there should be four types of undesirable units. [6] First, unit which performs no function, including its weight linking to output are all zero or very "small". This unit is considered always off or always on. The second is a group of similar units. Their function might be the same. The third is the group of units producing no effect because their outputs are inverse. The fourth is group of units producing constant effects. Therefore, we can determine the number of retained units by calculating the distinctiveness. "The distinctiveness of hidden units is determined from the unit output activation vector over the pattern presentation set, Units with short activation vectors n pattern space are recognized as insignificant and can be removed."

In calculating similarity, I take the calculation of the angle between the vectors. Since all the outputs are after using activation function sigmoid, the output has been in range between 0 to 1.

Table 1. four hidden units output after activation

Hidden units	**output**			
1	**0**	1	1	0
2	**0.1**	0.5	0.9	0.1
3	**1**	0	0.9	0.1
4	**1**	0	0.3	1

In table 1, we can clearly get four units outputs, which is vector(0,1,1,0) for hidden unit 1, (0.1,0.5,0.9,0.1) for hidden unit 2, (1,0,0.9,0.1) for hidden unit 3 and (1,0,0.3,1) for hidden unit 4. We use the arccosine to represent the angles between the pair of vectors. However, the angles are from $0°$ to $90°$ because the original point is (0,0). In this case, we can only get the similarity between units, but will lose the opposite situation of two units. As mentioned above, we need to find out the group of units producing no effect, so we should set the original point to (0.5,0.5). Then the angle between two vectors will be from 0°to 180°.

After calculating the vectors, we can find the group of similar units. If the vector angle between two similar units is 0°, the two units are totally the same. However, it happens rarely, so we set a rule that if two angles are less than 15°, then we consider the two units perform similarly. In this case, we can preserve one of them and remove the other one. Also, we should add the weight and bias of the unit to the preserved unit. If the vector angle between two units are 180°, the two units produce no effects together. This might not usually happen. So we set that if the vector angles are greater than 165°, we consider that this pair of units producing no effects together, so we can remove both of the units.

Table 2. four hidden units output vector angles

pair	*vectors*
1,2	78.54
1,3	169.1
1,4	163
2,3	95
2,4	92
3,4	6.1

Table 2 shows the vector angles between each pair of hidden units from table 1. We can clearly observe that hidden unit 1 and 3 should be removed, because their angles are larger than 165°. The weight and bias of unit 4 (or unit 3) should be added to unit 3(or unit 4), then hidden unit 4 (or unit 3) should be removed.

After all of these process, the undesirable units can be removed, and produce new weight and bias for some units.

Here is the pseudocode of pruning method.

```
Pruning():
      Removal=[]
      Similar=[]
      If angle <15° and angle not in similar list:
                  These neurons are too similar, then add one of the two
            angles to similar list.
      If angle >165°  and angle not in removal list:
                  These neurons have opposite functionality, so add all of
            them to removal list
      For unit in similar list: add the weight and bias of second unit to the
      first one and remove the second one.
      For unit in removal list: remove the unit
```

2.3 Neural network

There are 569 cases in the Breast Cancer Wisconsin (Diagnostic) Data Set. I take 80% of the data as training set, and the remaining 20% dataset as the testing set. I assume a simple neural network with three layers. My neural network has following settings and parameters: Activation function**:** $y=(1-e^{-x})^{-1}$, Learning rate**:** 0.015, Hidden layers**:**10-500, Epochs: 3000, Optimizer: SGD.

The genetic algorithm for feature selection parameters will be: Feature size:30 (the same as the number of features in original dataset). Population size:50-500, Cross rate: 0.8, mutation rate: 0.003, generations:200.

Process of experiment:

1. Train the original network with train dataset.
2. Train the pruned network with train dataset.
3. Test the original network with test dataset.
4. Test the pruned network with test dataset.
5. Repeat step1-step4 for 5 times to get the mean accuracy.

6. Select the main features according to genetic algorithm
7. Train the original network with train dataset.
8. Train the pruned network with train dataset.
9. Test the original network with test dataset.
10. Test the pruned network with test dataset.
11. Repeat step7-step10 for 5 times to get the mean accuracy.

3 Result and Discussion

3.1 Result

Table 3. Performance of network reduction technique

Hidden units	*Testing Accuracy (%)*	*Number of units can be reduced*	*Testing accuracy*
10	95.93	6	95.12
20	95.64	11	94.26
40	94.02	26	94.07
60	95.80	25	94.97
100	96.50	65	95.98
200	95.33	130	95.26
500	96.69	290	96.88

As shown above, it is obvious that after reducing the undesirable hidden units, the size of the neural network has been reduced. Meanwhile, the accuracy of the testing set is not affected. The accuracy of pruned network is close to the original network(∓2%). If the number of hidden units are increased, the performance is better than before.

Distinctiveness reduction method focuses on the undesired hidden neurons. Therefore, after remove these undesired hidden neurons, the performance of the network might not be affected that much.

Case1. Pruned network performs worse than the original network.

This might because the distinctiveness reduction method removes some units that are not that similar or some groups of units produce little effects together. Therefore, after removing those units, the accuracy is lower than before.

Case2. Pruned network performs better than the original network

This might because many hidden units have undesired functionalities. Those units might have undesired weights and bias thus influencing the performance of the neural network. After removing these hidden units, the performance of the network is increased.

Case3. Pruned network performs the same as the original network

In testing neural network with 100 hidden neurons, I find that the pruned network has almost the same accuracy with the original network. This might because the undesired neurons have extreme similar or opposite effects, so remove them might not affect the performance of network at all.

Meanwhile, the as the table shows, the size of the hidden neurons can be reduced almost half of the original size. This might because in a simple neural network, many hidden units behave similarly or large group of hidden units produce no effects together. Consequently, the bigger the network is, the more hidden units can be reduced. Thus, reducing half size of the hidden units can significantly increase the efficiency of a neural network. In this case, it is necessary for a designer to find a suitable size of hidden units to have a better performance network.

Table 4. Performance of neural network after using genetic algorithm (hidden units:200)

Pop Size	*Unselected Feature*	*Original Testing Accuracy (%)*	*Pruned Testing accuracy(%)*
50	6,8,11,17,19,24,26	95.19	96.12
50	5,6,10,11,13,15,17,19,20,21,22,23,24	91.00	90.87
100	5,6,8,11,12,15,16,17,18	96.05	96.03
100	4,5,6,8,13,15,18,19,26	95.46	96.26
200	4,6,8,12,13,15,17,18,19,26,27,30	96.46	96.02
200	6,7,9,11,12,15,17,18,19,23,26,30	96.49	96.49
500	6,7,9,11,13,15,18,19,26,30	96.41	96.81
500	6,7,9,11,12,13,15,17,18,19,26,30	96.43	96.58

There are 30 features in the original data set. As shown above, after using genetic algorithm to choose features, almost in all cases the accuracy is better than before. (original testing accuracy 95.33%, pruned network accuracy 95.26% with 200 hidden units). Meanwhile, the number of features is reduced thus reducing the dimensions of inputs.

There are four cases in my study.

Case1. Population size is 50

There are two experiments with the same population size as 50. The unselected features are quite different from each other. And the accuracy in second experiment is quite low. I believe this is because the population size is too small. The program will end after testing 50 subsets. Therefore, the result largely depends on the initial pop, which is generated randomly. In this case, the result is not advisable.

Case2. Population size is 100

There are two experiments with the same population size as 100. There are some similar unselected features in two experiments, and the accuracy is quite similar. This might because these unselected features have similar influence in the neural network.

Case3. Population size is 200

There are two experiments with the same population size as 200. The similarity of unselected features and the accuracy between two experiments is more similar than before. We can conclude that some of unselected features are in these lists. The accuracy of the network is improved because some of the redundant and irrelevant features are removed.

Case4. Population size is 500

There are two experiments with the same population size as 500. There are only two different unselected features, which is feature 12 and feature 17. Compare the first test with 500 population size to the second test with 200 population size, there are only one different unselected feature, which is feature 12. The accuracy is approximate in case 3 and case 4. We can conclude that these are the redundant and irrelevant features in the original data set.

These experiments results show that if the population size is small, the result of genetic algorithm largely depends on the first population, which is generated randomly. Setting a reasonable population size is important for genetic algorithm. The fitness function also influences a lot. In this experiment, I use logistic regression to calculate the accuracy as the fitness. If the size of population is too small (case1, case2), the program will not provide the pop with highest fitness. Therefore, the accuracy of feature selection is low in this case. After selecting a reasonable size of population, the irrelevant and redundant features can be removed. The result becomes credible and genetic algorithm will improve the performance of the network.

In all, the distinctiveness reduction function can largely decrease the number of hidden units without affecting the original accuracy. Using genetic algorithm to select features can slightly increase the accuracy of neural network and reduce the dimensions of inputs.

3.2 Comparison

The accuracy of using Ant-Miner is 96.04±2.80 with 3000 epochs and no optimizer. [10] The testing accuracy of my approach is around 95.50 with 3000 epochs and momentum optimizer. In the Wisconsin Breast Cancer data set, the difference between my result and that of Ant-Miner's is quite small.

In summary, the authors of Ant-Miner algorithm have discovered some simple and accurate rules for the Wisconsin breast cancer data set. In my case, I don't set rules for the attributes. By contrast, I use genetic algorithm to remove irrelevant and redundant features of original data set. Using of network reduction techniques reduces the number of hidden units and maintain a high accuracy.

4 Conclusion and Future Work

This work has offered a combination of feature selection using genetic algorithm and network reduction techniques. Genetic algorithm helps to find the main features and reduce the dimensions of the original Wisconsin Breast cancer data set. The network reduction techniques calculate the vector angles between hidden units, thus deciding the undesirable units and remove them. In this paper, I have demonstrated the process of using genetic algorithm, how to calculate the vector angles and remove the undesirable units.

I have compared the performance of this paper and the ant-miner paper in Wisconsin Breast cancer data set. In all, the testing accuracy is quite similar.

There're still a lot of work to do for me. In feature selection part, I only compared the influence of population size. There are many other parameters can be tested. For example, the size of generation and the probability of mutation. Meanwhile, the fitness function is very important. There might be other choice of fitness function in this data set. I can find out more fitness functions and compare their performance.

There are other research directions. In distinctiveness reduction method, the angle threshold is 15° and 165°. This might change according to different type of neural network or different activation function. So I can have more experiments on choosing different thresholds.

References

1. Alpaydin, Ethem (2010). *Introduction to Machine Learning*. MIT Press. p. 9. ISBN 978-0-262-01243-0.
2. Har-Peled, S., Roth, D., Zimak, D. (2003) "Constraint Classification for Multiclass Classification and Ranking." In: Becker, B., Thrun, S., Obermayer, K. (Eds) *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, MIT Press. ISBN 0-262-02550-7
3. Goldberg, D. E. (1989). Genetic algorithm in search.
4. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
5. Cun, Y. L., Boser, B., Denker, J. S., Howard, R. E., Habbard, W., & Jackel, L. D., et al. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems, 2*(2), 396--404.
6. Gedeon, T. D., Harris, D. (1991). Network Reduction Techniques. *Lecture Proceedings International Conference on Neural Networks Methodologies and Applications* (Vol.1, pp.119-126).AMSE
7. Gedeon, T. D., & Turner, S. (2002). Explaining student grades predicted by a neural network. *International Joint Conference on Neural Networks, 1993. IJCNN '93-Nagoya* (Vol.1, pp.609-612 vol.1). IEEE.
8. Gedeon, T. D. (1998). Stochastic bidirectional training. *, 2*, 1968-1971 vol.2.
9. Zhang, P., Varma, B., & Kumar, K. (2003). Neural vs. statistical classifier in conjunction with genetic algorithm feature selection in digital mammography. *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on* (Vol.2, pp.1206-1213 Vol.2). IEEE.
10. Rafael S. Parpinelli and Heitor S. Lopes and Alex Alves Freitas. An Ant Colony Based System for Data Mining: Applications to Medical Data. CEFET-PR, CPGEI Av. Sete de Setembro, 3165.