After using genetic algorithm for feature selection, apply cascade-correlation learning architecture[1] to detect anger emotion veracity

Jifeng Li

Research School of Computer Science,
The Australian National University,
Canberra ACT 2601
u7101811@anu.edu.au

Abstract. After using genetic algorithm for feature selection, the Cascade-correlation neural network [1] algorithm is applied to the anger data set to explore the prediction accuracy of the network structure in detecting the authenticity of anger emotions. The anger dataset contains pupil changes of people facing real anger and fake anger. Experiments have found that feature selection can improve the performance of the Cascade-correlation neural network on this data set. And the influence of the number of hidden neurons in the cascaded correlation neural network on the accuracy is discussed. And compared with the backpropagation neural network in terms of performance. [2] It is found that the prediction accuracy of the back propagation neural network is higher.

Keywords: Cascade-correlation, Genetic algorithms, Detecting emotion veracity, Back-propagation neural network, anger , Feature selection

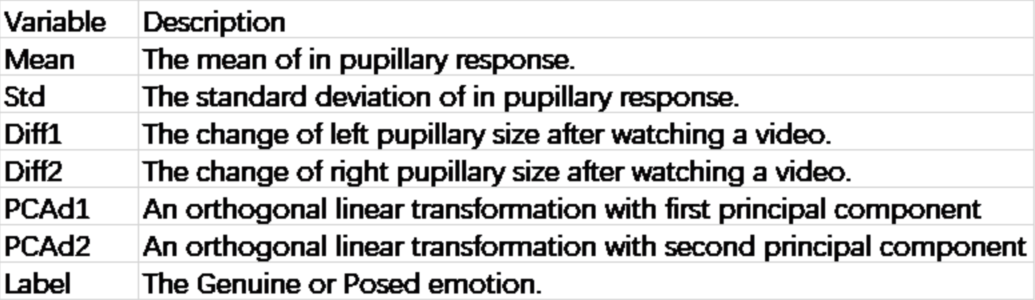
1 Introduction

There are a wide range of application scenarios for detecting the authenticity of emotions. Most people use backpropagation neural networks to train models that detect the authenticity of emotions. But the backpropagation neural network takes a lot of time when computing resources are scarce. I try to use genetic algorithm to select factors to reduce the data size. Use the Cascade-correlation algorithm to reduce the size of the network structure. The biggest advantage of cascade correlation is that it is faster and does not need to back-propagate the error signal through a network connection [1]. Perhaps the use of cascaded correlation algorithms can reduce the cost of computing resources while ensuring the accuracy of predictions. The cascade correlation first starts with a very small network, and then gradually increases the hidden neurons. This method of constructing a network structure makes it easier to find the most suitable minimum network structure. The purpose of this article is to use genetic algorithms to reduce the size of the data. Then use the cascaded correlation neural network to predict the results as accurately as possible through preprocessing, model design and parameter adjustment, and explore whether the method is worth using on the anger data set. The reason why I choose to explore the emotions of anger is because it is easier for people to recognize the emotions of anger than expressions of frustration and happiness [3]. This proves to a certain extent that recognizing anger is a higher priority for people.

2 Method

2.1 dataset select

The anger [2] data set I selected is mainly about pupil changes when people are angry and pretending to be angry. The data set is obtained in accordance with the principle of controlled trials, eliminating the interference of irrelevant variables. The data set has a total of 400 data, and each data has 7 main variables.



Variable	Description
Mean	The mean of in pupillary response.
Std	The standard deviation of in pupillary response.
Diff1	The change of left pupillary size after watching a video.
Diff2	The change of right pupillary size after watching a video.
PCAd1	An orthogonal linear transformation with first principal component
PCAd2	An orthogonal linear transformation with second principal component
Label	The Genuine or Posed emotion.

Table 1. Data description

2.2 dataset Preparation

Remove the useless Video tags from the data set. Then use 1 for Genuine in the Label tag and 0 for Posed. Then normalize the data except the label, using the zero-mean normalization method. Then the data set is divided into training set and test set according to the ratio of 8:2. Label is used as the dependent variable, and the others are used as independent variables.

2.3 cascade-correlation algorithms

1. The cascade-correlation neural network has only two layers of input and output at the beginning. Then perform weight training in a similar way to logistic regression. The residual of the predicted value and the true value is used as the loss function, and the training is stopped when the value of the loss function no longer drops or the number of gradient descents has been sufficient. Then store the weight of the output layer at this time and the predicted result. Of course, the loss function can be defined in different ways. The purpose of the loss function is to minimize the difference between the predicted and the target value

2. Training hidden nodes. Add a candidate node that only gets the input data and the output of the hidden layer before it as input. For example, if this is the second hidden node, then the first data it reads should be the initial first data plus the value generated by the first hidden node reading the first data. Randomly initialize the weights of candidate nodes for different dimensions of the input data. I denote the output of the hidden node as V. The residual error between the predicted result and the true result is denoted as E. We want to maximize the correlation between E and V as much as possible. It is believed that the more relevant E and V are, the more information the trained node contains. If the correlation is 1, then the predicted result only needs to subtract the value predicted by the hidden node to get 100% accuracy. In order to maximize the correlation, the loss function is defined:

$$S = o|p(Vp - V)(Eo{,}p - Eo)| \tag{1}$$

Among them, O represents the entire network, and P represents the first few pieces of data. The overline indicates the average value. Use the gradient descent method for the loss function. When the gradient descent amplitude is too low, or the number of gradient descents is enough, we stop training. And freeze the weight of the node, and pass the output to the next hidden neuron.

3. Keep training hidden neurons, that is, repeat step 2. Stop until the number of hidden neurons exceeds the prescribed number or the accuracy reaches the standard.

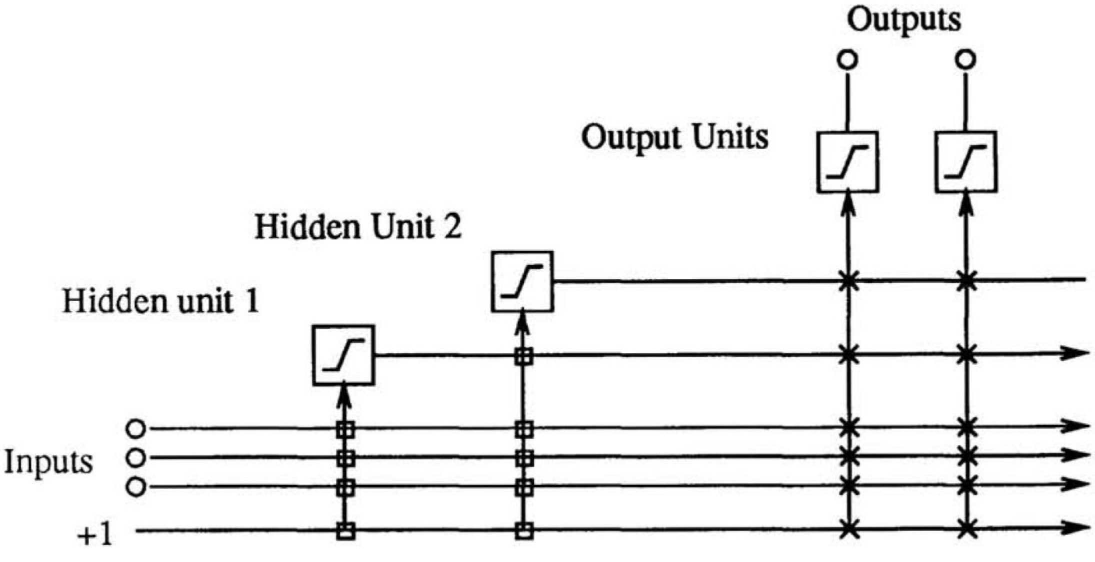


Fig. 1. Cascade-Correlation Learning Architecture

2.4 Genetic algorithm

The genetic algorithm was proposed by John Holland in 1975. The algorithm is designed with reference to the philosophical principle of survival of the fittest in nature. This is the main operation in the question we deal with:

1. Encode the feature to be selected. If the feature of the dimension is selected, the number of the dimension is 0; if the feature of the dimension is not selected, the number of the dimension is 1. For example, we have a total of 7 features, in order they are Mean, Std, Diff1, Diff2, PCAd1, PCAd2, Bias. If I select only the feature Mean and the feature Std as input, then the code of the individual is 1100000. Need to pay attention Yes, I added Bias in Cascade-correlation to the features that can be selected. Individual feature selection codes can be analogous to human DNA.

2. The second step is to initialize a population. In order to increase the diversity of the population, I set the number of the population to 1000. The features selected by each individual in the population are randomly generated.

3. The third step is to calculate the ability of each individual to adapt to the environment. That is, the fitness of each individual. For each different individual, I take the dimension selected by the individual as input, and then use the Cascade-correlation network for training. Use 0 to 15 hidden units for training respectively, and take the highest accuracy rate as the individual's accuracy rate. And the accuracy of the individual is regarded as fitness.

4. Selecting. We need to ensure that the number of populations remains unchanged after reproduction. I will select the parents to reproduce the next generation according to the probability, and the number of them will remain the same. The proportion of the fitness of each individual to the total fitness of all individuals is used as the probability of the individual being selected.

5. Perform a crossover. Use the selected population. Traverse the population according to the cross rate probability of 0.8. In other words, each individual has a probability of 0.8 to exchange its genes with other individuals. What is exchanged is that the first dimension is random.

6. Perform mutation. Every individual after the crossover has a very small probability of genetic mutation. Here I set the probability of mutation in each dimension of each individual's gene to 0.002.

7. The above are the operations needed to reproduce one generation. After completion, re-enter step 3 until the result is satisfactory. Here I set a total of 200 generations.

2.5 About hyperparameter settings

I used Adam gradient descent for training hidden neurons and final output weights. I set the learning rate of each gradient descent to 0.01, and then set the L2 regularization, and the weight decay is 0.003. And the loss function used to train the output neuron is BCE With Logits Loss.

3 Result and Discussion

3.1 Result

In the original 7-dimensional input, the genetic algorithm finally selected only one dimension. Only one dimension of PCAd1 is used as input. I will compare the result of using all dimensions as input and the result of using only PCAd1 as input. I try to gradually increase the number of hidden neurons. Then count the accuracy of the training set and the test set when the network has a different number of hidden neurons. A total of five statistics are collected, and the average value is taken as the final result.

When all dimensions are used as input, the accuracy rate is greater than 73% at the highest point, and the accuracy rate reaches its peak when the network structure uses 7 hidden neurons. The accuracy rate obtained is shown in the figure below:

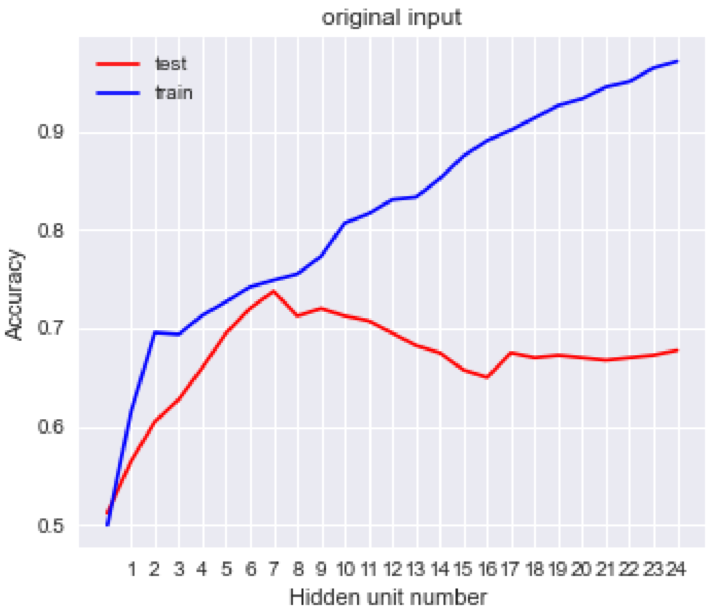


Fig. 2. Accuracy with original input

After comparison, the accuracy of using only one dimension of PCAd1 can reach up to 81%, which reaches a peak when the network structure uses 10-13 hidden neurons. The accuracy rate obtained is shown in the figure below:

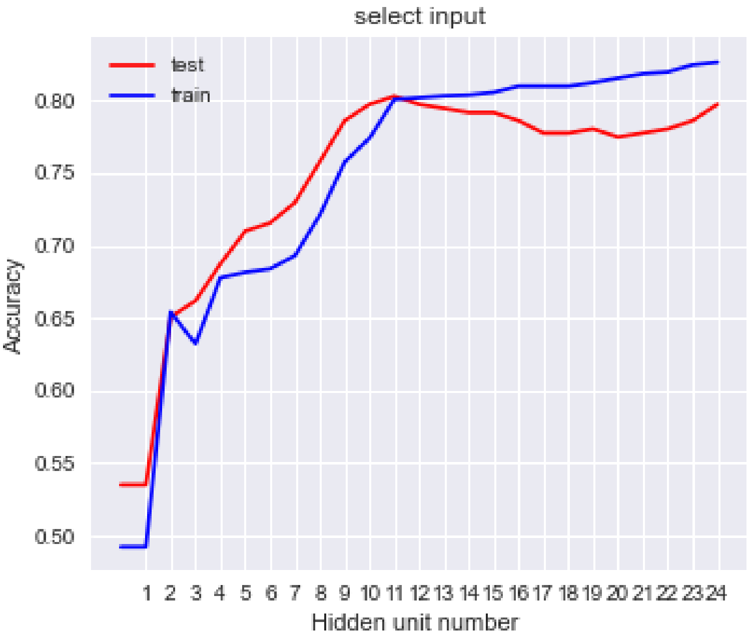


Fig. 3. Accuracy with hidden input

3.2 Discussion

Some researchers have achieved 95% accuracy on this data set through a back-propagation neural network. The structure it uses is the classifier used in [4].

When selecting data of all dimensions as input, the average highest accuracy rate of the Cascade-correlation model is about 74%. It has to be said that the accuracy gap between the two is still very large. It may be that although cascade-correlation simplifies the structure and can reduce unnecessary neuron overhead, at the same time, the highest degree of fit to the data set has also declined.Looking at the network with different numbers of hidden neurons, we found that for the training set, the greater the number of hidden neurons, the higher the accuracy. But for the test set, the peak accuracy occurs when the network contains 5-7 hidden neurons. Afterwards, it may be due to over-fitting, and the accuracy will decrease instead.

Then compare the results when only taking the PCAd1 dimension as the input. At this time, the highest accuracy rate of the network model is about 82%. The accuracy rate at this time is much higher than the accuracy rate when all dimensions are used as input. This shows that the use of genetic algorithms for feature selection is effective. It is worth noting that after the genetic algorithm is used to reduce the dimensionality, the accuracy of train and the accuracy of test are very close. It can be said that the genetic algorithm has reduced the phenomenon of over-fitting very well. But at the same time, the number of hidden neurons at the peak accuracy rate also increased. The reason for this phenomenon is worth leaving for future work to be studied.

4 Conclusion and future work

Reading this article can help everyone better understand genetic algorithms and cascaded neural networks. Using genetic algorithms to reduce data dimensionality can not only reduce the data size in some application scenarios, but even reduce the risk of overfitting. The cascade correlation neural network usually only needs a very simple structure to train, and the training time is very short. However, it is not easy to find the appropriate Cascade-correlation neural network parameters, and it is prone to overfitting. When computing power is sufficient, backpropagation neural networks seem to be more advantageous.

The highest accuracy I achieved was only 82%, but this does not mean that the potential of the Cascade-correlation model is limited to this. We can try different hyperparameter tuning methods, regularization methods and optimization methods to explore the maximum possibility of this model. Secondly, genetic algorithms with different parameters can also be considered to select feature factors. Not only that, but it can also be optimized by changing the structure of hidden neurons [5]. Or use the caper method to experiment.

In the future, you can also consider using the Cascade-correlation model combined with genetic algorithms for feature selection on different data sets. Because the Cascade-correlation model consumes less computing resources, it can calculate the individual fitness in the genetic algorithm faster. The combination of the two may be able to help reduce the loss of computing resources.

5 Reference

[1] Fahlman, Scott E., and Christian Lebiere. *The cascade-correlation learning architecture*. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1990.

[2] Chen, Lu, et al. "Are you really angry? Detecting emotion veracity as a proposed tool for interaction." *Proceedings of the 29th Australian Conference on Computer-Human Interaction*. 2017.

[3] M. Mather, M.R. Knight, Angry faces get noticed quickly: threat detection is not impaired among older adults, *J. Gerontol. Ser. B: Psychol. Sci. Soc. Sci.*, 61(2006), 54-57. DOI: https://doi.org/10.1093/geronb/61.1.P54

[4] Md Zakir Hossain and Tom Gedeon. 2017. Classifying Posed and Real Smilesfrom Observers' Peripheral Physiology. *11th International Conference on Pervasive Computing Technologies for Healthcare* (PervasiveHealth '17), EAI Conference Series, Barcelona, Spain, 1-4. Retrieved from http://bit.ly/2gkllCZ

[5] Khoo, Suisin, and Tom Gedeon. "Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks." International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, 2008.