Using neural networks to predict whether a person belongs to high-income groups

Xuanming Gu

Research School of Computer Science,
Australian National University
Canberra, Australian
Xuanming.gu@anu.edu.au

Abstract. There are many statistical methods to predict people's income. Most of them perform well. However, using these methods requires a lot of statistical knowledge as well as expertise. I build a neural network model to help people predict income without using statistical knowledge. Neural networks require large amounts of time to train large data sets. I tackle this time-consuming problem by utilizing Least Trimmed Squares (LTS) and genetic algorithms for feature selection to speed up training. Although my results are slightly inferior (4% worse) to the state-of-art one based on the same dataset, my proposed methodology can significantly improve the training time of neural networks, specifically, with a 23.81% reduction of training.

Keywords: Feedforward neural network, Least Trimmed Squares, Feature Selection

1 Introduction

Neural networks, especially recent advanced techniques in deep learning, have dramatically improved the state-of-art performance in both classification and regression problems (Ac.els-cdn.com, 2018). Traditionally, the government relied on statistical knowledge to predict people's income. (Anon, 2018) Establishing a statistical model requires a lot of knowledge and spending a lot of time. Then people start using neural network to predict person's income. Using neural networks can avoid the establishment of complex statistical models. The accuracy of using neural networks can reach 84%. (Archive.ics.uci.edu). However training the neural network is time-consuming. Genetic algorithm can greatly accelerate training time. It will take 1.6 billion years to solve the complex TSP (24 cities) problem using traditional methods. Using genetic algorithm can solve TSP problem in a few seconds. (Nbviewer.jupyter.org, 2018) So that I will use genetic algorithm and neural network to build up an efficient prediction model.

Adult Data Sets are used to train and test neural networks. There are some reasons I pick Adult Data Set. At first Adult Data Set is based on bureau database (Archive.ics.uci.edu, 2018). These data have higher credibility. That means my research has meaning. Second, the Adult Data Set has more than 30000 valid data. Huge data set will help neural network training. I use five-layer feedforward neural network to implement this task. Because I used one-hot method to encode data. It creates too many features. Simple network model cannot achieve good results

Since Adult data sets are based on census. It is possible for people to misrepresent data for various reasons. Hence, there could be noise in the data set. I use Least Trimmed Squares to speed up training and reduce the impact of noise on training. Because Gedeon said If the test set is known to be clean, LTS maybe will lead better result (Slade P., Gedeon T.D.,1993). Predictive accuracy and time cost are main methods to analysis the performance of neural network. After data pre-processing by mention later method, we will get a data set containing 89 features. Excessive functionality can slow down training. Hence, I use genetic algorithm to do feature selection to speed up training.

2 Method

In this section, I will describe how to build, train, and evaluate my neural networks. My methodology includes four main steps. The first one is how to pre-process the data. In the first part I will also introduce how I use genetic algorithm to implement feature selection. The second is the structure of my neural network and explanations of important parameters. The third part is to introduce the evaluate function within the utilized genetic algorithms. The last part is how I improve my neural networks.

2.1 Data pre-processing

2.1.1 Basic data preprocessing

The original data set cannot be adopted directly. due to various problems with this data set, such as missing value. I performed a four-step preprocessing on the original data set. The first step is deleting missing data record. Normally we use mean values to interpolate missing values. But in this case, since Adult data set has over 30000 records. I choose to delete missing data record to improve the quality of the data set. The second step is to normalize the collected strings, such as deleting the extra spaces before each string. This is because extra spaces will result in the same type of data being identified as different features. For example, "lawyer" and " lawyer" will be identified as two different data without string normalization. The third step is processing object type data. I used label encoding to encode some sorted features, such as education (Preschool Significantly less than Master). And one-hot encoding has been used to encode some features which cannot be sorted. This can avoid that the values being "misinterpreted" by the algorithms (Moffitt, 2018). The last step is encoding people's income state. If people's annual income equal or greater than 50k then we encode as 1, otherwise we encode it as 0. After this four-step date pre-processing we extend the data's attributes from 14 to 89 dues to the effect of one-hot encoding. We pack those attributes and set them as input, people's income state as output. And we use those to train and test neural network.

2.1.2 Feature selection

I used genetic algorithms to implement feature selection. Here is the main process of genetic algorithm.

- Initialization population
- Fitness assignment
- Evaluate fitness value for each individual and pick up the best one
- Using the best one to process crossover and mutation
- Repeat until meet the terminate preconditions

Before I implement my genetic algorithm, I randomly sampled a utilized training data set from all the records. This is because the whole training data set has over 30000 records. If one insists using whole training data set, the time cost with genetic algorithm will be enormous. Because I use one-hot encoding, which implies that the sub-data-set need to include all the features. Otherwise the sub-data-set will have less features. The DNA chromosomes utilizes a binary representation. Specifically, 1 implies active and 0 implies inactive. As I mentioned before the first step is to create and initialize the individuals. The initialization of individuals usually is in random fashion. The second step is assigning the fitness to each individual. I trained the model with the training data then evaluate its selection error within the selected data. I adopted the accuracy for each individual as its fitness value in the genetic algorithm. The third step is selecting the best individuals from the population. In other words, we need to pick the individual which has higher fitness values to undertake crossover and mutation operations. Then we repeat this process until meeting the terminate preconditions. I set "the best one not changes in 3 gens" as terminate preconditions (Neuraldesigner.com, 2018)

2.2 Implementation

A five-layer feedforward neural network is used to implement this task. All connections are from units in one level to the subsequent one, with no lateral, backward or multilayer connections. The first layer is an input layer. The second layer to the fourth layer are hidden layers. The last layer is an output layer. Relu function is used as activation function between the input layer and first hidden layer, the first hidden layer and second hidden layer, the second hidden layer and third hidden layer. The size of input layer equal to the number of data's attributes. The first hidden layer consists of fewer units than input layer, the second hidden layer consists of fewer units than the first hidden layer, thus distinguish people's group deeply. That because Each layer categorizes the data once. As the number of layers increases, the classification will become less and less until has 2 class. The output layer only has two units and is used to determine whether people are high-income people.

Cross-Entropy Loss was used as my loss function. Because Cross-entropy loss can measure the similarity between two probability distributions. In addition, data loader function was also used in code. It would automatically divide training data. In other words, data will be trained more than once in one epoch. The size of sub-training data is important since each sub-training data set has 15000 records. They could guarantee neural network have enough data to train. The number of epochs was set as 200. The reason was I found that the predictive accuracy cannot be improved after training 200 epochs. Moreover, learning rate was set very small, large learning rate would cause the loss curve cannot drop smoothly.

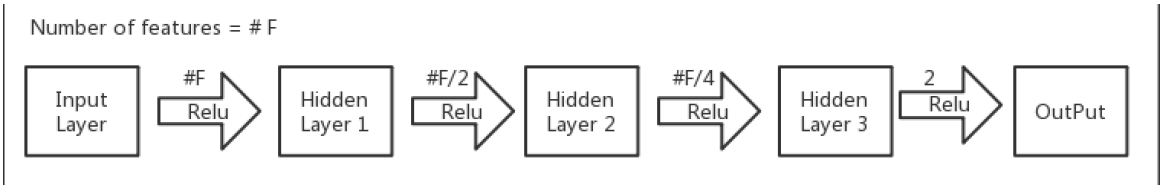


Fig.1. The framework of five-layer feedforward network

2.3 Evaluate Function

Predictive accuracy and time-cost of training were used as main functions to evaluate the performance of neural network in this article. Predictive accuracy was the most important indicator in this model. Because it directly determined whether this model had practical using value. The formula of predictive accuracy is the number of correctly prediction divided by total number of predictions.

Time expenditure is also an important indicator. Training neural networks was time-consuming. By adopting Rapid training of neural networks, it effectively reduced costs and also, train more times in one-unit time, hence, the predictive results were improved significantly. We use python built-in function, time (), to calculate time cost. (All time cost calculate running is MacBook Pro, Retina,13-inch, Mid 2014 with 2.6GHz Intel Core i5 Processor and 8GB 1600 MHz DDR3 memory)

Changes in loss over epoch were also important assessment method. Using graphs to more intuitively observe changes in loss helps us analyze models. Pyplot library is used to draw the loss curve. In each epoch, the program will collect the value of loss once and store in a list. After training, Pyplot will draw the graph of loss.

2.4 Improvements

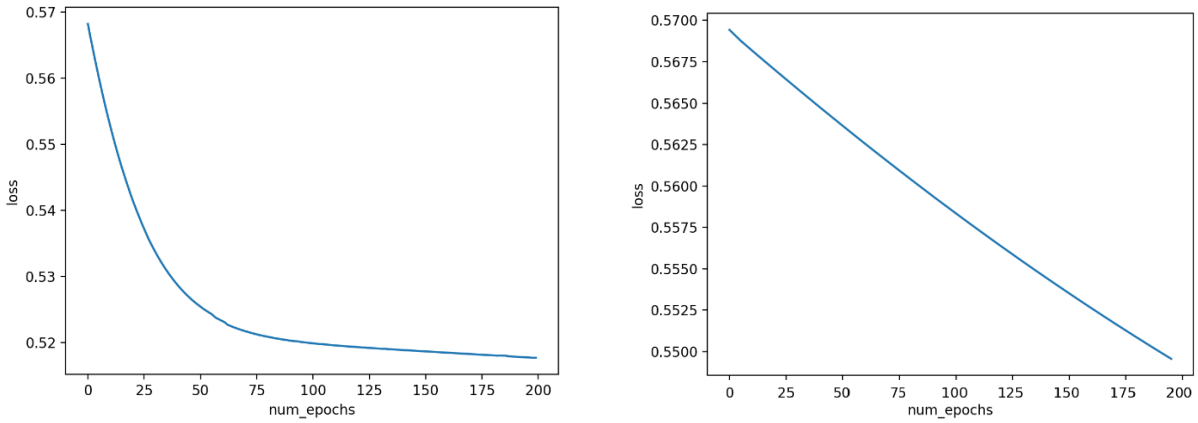
I use Least Trimmed Squares (LTS) to improve my neural network. Because the test set is clean, LTS method can work well in this case. This method aims to accelerate the convergence of training. The basic premise is to minimize only a portion of the mean square error in the training set. In each fifth epoch, all patterns in the training set are arranged in ascending order according to the mean square error. The pattern associated with the lowest mean square error is used to train the network for the next 5 epochs. Then repeat the process and select a new subset of the full training set as the training patterns. This approach may lead to better generalization because the outliers in the training set will never be used to train the network because they will produce large mean square error. As a result, weights will never be adjusted to fit these outliers. (Slade P., Gedeon T.D. 1993).

3 Results and Discussion

I used Least Trimmed Squares (LTS) to improve my neural network. From the figure below, we can see LST version's loss curve nearly linear function's. In other words, LTS method speeds up convergence of training. That is because LTS only uses best loss to training next data. I also calculated the time they spent training 200 epochs, and the accuracy of the predictions. The basic version cost 42.73 seconds and has 78% predictive accuracy. The LST version costs 34.38 seconds and has same predictive accuracy with basic version. The predictive accuracy did not improve maybe because of the data set has little noise and size of patterns are too large. So that the outliers still exist in each pattern. They are still used in training. The predictive accuracy of basic version is 78%. The Feature selection improves the accuracy and speed up the training.

Version	Num epochs	# Features	Correct	Total Test	Accuracy	Time cost
basic	200	89	23527	30162	78 %	42.7358
LST version	200	89	23523	30162	77.98%	34.3754
Feature selection	200	58	23828	30162	79%	32.3112

Table 1. Basic test data



Basic version LTS version

Fig.2. A line chart that illustrates the tendency of the loss in Basic & LST version

The results were slightly inferior than Ron Kohavi's paper based on the same dataset. Ron Konhavi use NBTree algorithm to implement this task. "NBTree is similar to the classical recursive partitioning schemes, except that the leaf nodes created ate Naïve-Bayes categorizer instead of nodes predicting a single class." (R. Kohavi1996) NBTree algorithm amazingly improves accuracy to 85. Decision-tree and Naïve-Bayes rule was used in NBTree algorithm. Conditional probabilities for each attribute value given the label will be requires by classifiers. Decision-trees will approximate a reasonable function as the database grows (R. Kohavi1996). The reason why NBTree algorithm lead to better result maybe is Naïve-Bayes rule gives better weight for each feature. And Naïve-Bayes classifiers create more hidden layer to classify people.

The Feature selection improves the accuracy and speed up the training. Speeding up the training because the network only need process less features. It reduces the calculations. Slightly improved accuracy may be because of the genetic algorithm eliminates some of the noise.

4 Conclusion

In conclusion, five-layer feed-forward neural network model was used in this article to help people predict whether a person belong to high-income groups. I pick Adult Data Set as main research data set. Because at first Adult Data Set based on bureau database. The data has meaning. Second, the Adult Data Set has more than 30000 valid data. Huge data set will help neural network training. I used Least Trimmed Squares and genetic algorithms to speed up training and reduce the impact of noise on training. LST can effectively speed up the training process. However, it failed to improve prediction accuracy. It may because of data set itself or the failure to split the data set sufficiently.

5 Future work

Experiments show that a five-layer feed-forward neural network can predict people's income level. However, good accuracy cannot be achieved. In the future we can use more complex network models such as CNN to training to network. Genetic algorithms are effective, and in the future, we can try to update the algorithms of the selection and fitness components to enhance genetic algorithms.

Reference:

Archive.ics.uci.edu. (2018). UCI Machine Learning Repository: Adult Data Set. [online] Available at: http://archive.ics.uci.edu/ml/datasets/Adult [Accessed 29 Apr. 2018].

Ac.els-cdn.com. (2018). [online] Available at: https://ac.els-cdn.com/S000437020200190X/1-s2.0-S000437020200190X-main.pdf?_tid=fc2769dc-822f-4eaa-8f13-fffaa7f3da89&acdnat=1527570097_3ca6759f6b1642655f62291853fef1c6 [Accessed 29 May 2018].

Anon, (2018). [online] Available at: https://www.tandfonline.com/doi/full/10.1080/2330443X.2017.1317223 [Accessed 29 May 2018].

Moffitt, C. (2018). Guide to Encoding Categorical Values in Python - Practical Business Python. [online] Pbpython.com. Available at: http://pbpython.com/categorical-encoding.html [Accessed 29 Apr. 2018].

Nbviewer.jupyter.org. (2018). Jupyter Notebook Viewer. [online] Available at: http://nbviewer.jupyter.org/github/lmarti/evolutionary-computation-course/blob/master/AEC.03%20-%20Solving%20the%20TSP%20with%20GAs.ipynb [Accessed 29 May 2018].

R. Kohavi(1996), "Scaling Up the Accuracy of Naive-Bayes Classi ers: a Decision-Tree Hybrid Accuracy Scale-Up: the Learning", Data Min. Vis. no. Utgo 1988, pp. 1-6, 1996.

Slade P., Gedeon T.D. (1993) Bimodal distribution removal. In: Mira J., Cabestany J., Prieto A. (eds) New Trends in Neural Computation. IWANN 1993. Lecture Notes in Computer Science, vol 686. Springer, Berlin, Heidelberg

Neuraldesigner.com. (2018). Genetic algorithms for feature selection in Data Analytics | Neural Designer. [online] Available at: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection [Accessed 29 May 2018].