Different Neuron Networks Performance Improvement by Data Preprocessing, along with Network Reduction and Evolutionary Algorithm

Zhe Zhang
Research school of Engineering and Computer Science
Australian National University, ACT, Australia
U6128882@anu.edu.au

Abstract

The different neuron networks (NN) training performance can be dramatically improved at three stages, [1]analysis and development of the data in advance, [2]redundancy elimination and [3] corresponding developed learning algorithm. This paper indicates the ways to compare the results for different neuron networks by adopted methods using data set from UCI. Preprocessing actually pose a threat to information lossless. However, it can make the data more reasonable with proper explanation. Network reduction can lower its complexity to accelerate the efficiency and to remove the noise. Particularly, evolutionary algorithm can optimize the data with good performance. Furthermore, design of the weakly supervised neuron network based on more simple and limited units could be established to accelerate the quality and applicability scope.

Keywords: Data preprocessing, Classification, Neuron network, Evolutionary algorithm

1 Introduction

Neuron network training performance has been quickly raised continuously for decades of years. This paper aims to compare the results between different design of neuron networks, along with data preprocessing, including pruning and evolutionary process. Thus, the data sets chosen from UCI occupies different perspectives, from bank dataset to mnist. Actually, as for the data preprocessing, there are a lot of work from pioneers.

To begin with, it is claimed that the performance of the neuron network can be generally limited from the quality of the data by Azoff [1]. Thus, the first data set choosing from UCI is kind of classification style with many necessary and unnecessary attributes. When choosing the data set, the

data with more dichotomies are likely to be preferred. For example, the Soybean (Large) Data Set and Mushroom data set from UCI is a better choice, because it has more than 300 instances and 35 attributes. In detail, there are existing 19 classes, there are only 15 out of that which are common recognized. As for the attributes, it is a big quantity. Furthermore, some attributes can be useful for preprocessing, such as removing the irrelevant attributes, deleting the lines with missing values, giving reasonable defaults to the blank and stochastic prediction. Further, the C4.5 distribution-based classification, using reduced models, such as classification trees, and imputation of predictive values [2] can be helpful with the missing value in data set preprocessing and get the high training performance at a low cost. In the data set, firstly define the categorical values and numeric values. Then, considering the classification models, which is generally thought of SVM, Naïve Bayes, Perceptron, Logistic Regression, Decision Tree and the chosen Random Forest way. The random forest [3] is a classifier which contains a few decision trees. It is chosen because it can keep the relative accuracy even with a badly missing data values in data sets, and the process of the learning is fast. After that, encoding is a process to classify the binominal variables, ordinal variables and disorderly variables. The specific classification methods towards different variances are shown in methods part below. To optimize the training set, it is recommended that rank the variables by the influential weight. Especially, as for disorderly variables, dummy variables can be used to encode such variables, generally with the number of one less than the classification. Fortunately, the function is provided by the Pytorch. As for the data set finally in the tests below while considering all the factors above, the bank data set to predict whether the customers will have the deposit behavior.

When it comes to the preprocessing in numeric features, it is important to the machine learning algorithm. Before training, discretization of continuous features matters in escaping from confusion of the data. For example, the extremum values can pose a threat to the concentration on the misleading relationships. Discretization can avoid the overtopping from the exact models to some extent. So, the data sets chosen from UCI are also considered about this part. The further examples and discussion will be shown below in methods and results. Proper binning can consider quantile classification and standard variance. As for normalization, it is necessary the data into a reasonable range. For example, the units of different values can be contrasting, and it is necessary to regular all the data into the same unit and range to avoid useless influence and extremum value. It can also be done by standard scalar. As for reduction, manual analysis of the data set, especially for the attributes can dispose some of the attributes, such as date in a relative small range when training the class of the creatures, and the appearance of a person to predict the behavior of the deposit. Whatever, it can be deleted to avoid influence towards the results and to save the training time. After a period of processing and training, the methods of data persistence can be used to same time and to continue the training. Moreover, the data sets after preprocessing can be directly used in other researches in the same condition. A problem same with cascade correlation network is that it can possibly enhance the dimension of the network, and it can destroy the meaning of further training, even if it can be reduced. Because improper dimension reduction may lead to disaster of the training. The simple principal component analysis (PCA) can be both used in binary graphic and also the high dimension data sets if it is necessary. However, PCA can demonstrates some of the noise so that can eliminate it. That means the balance between and reduction of noise and losing information should be considered when using PCA [5]. Similarly, the recursions process both in cascade correlation and data persistence can dramatically accelerate the speed of the set training.

Due to the concentration on the preprocessing, the partition of data set should be also raised in this part. Generally, dataset is separated into training set and test set. Actually, cross validation set is also established to evaluate the performance of different models. So that, the result from the best model quality can be tested on the real test set. In this model, Smote algorithm is adopted to do over sampling [6], and shuffle methods to do under sampling. Details will be described below in method part. Especially, to avoid the noise, only the numeric values should be kept instead of others.

After preprocessing part, there is work remaining to compare the performance between different kinds of neuron networks. In general, the long short-term memory recurrent neuron network(LSTM) deriving from recurrent neuron network(RNN) is more suitable for text recognition and prediction. Fully convolution neuron network (FCN) deriving from convolution neuron network (CNN) is more meaningful in pixel-wise image semantic segmentation and recognition. Moreover, the evolutionary algorithm can have a good performance on simple dataset pruning, particularly using multi-objective optimization on classification dataset. It

Furthermore, many works could be developed in detail to improve the performance score. As for preprocessing, better feature choosing, derived attribute [7], and deep explanation of attributes can enhance the efficiency and correctness. Also, it is better to adopt more reasonable imbalance of the dataset, and it is really influential. It can be dealt with cost-sensitive learning to reason the data more making sense. Moreover, cascade correlation network can be tried to accelerate the speed, and it is helpful to do the recursive training. A limited research is done in this paper whatever. However, more detailed tuning parameters in preprocessing counts more weight in this research.

2 Method

To begin with, it is grateful to follow the S. Moro, P. Cortez and P. Rita. [8] for the Bank Marketing Data Set from UCI mentioned above in [2.1]preprocessing and encoding part. There are four parts in the dataset. Actually, only the full examples data set and the ten percent of the examples data set as test are used. It aims to predict the success of the fixed deposits by binary results (1 or 0). It is followed by the comparison between different kinds of neuron networks. In the end, there is an experiment in evolutionary algorithm.

2.1 Data preprocessing and Network reduction

Although the data missing column shown in the CUI is unknown, after the reviewing, the numeric values has no missing while there is existing missing of the non-numeric, categorical values. There are three ways to deal with the missing values. Firstly, the method to remove the lines with missing values. Apparently, it may lead to losing of the valuable information. According to this, it is not likely to be chosen. Further, if the certain attribute accounts for few weights for the model, it can be assigned to the model. Actually, after the preprocessing, the remaining values are all influential to the training model. Thus, this is not a good way at all. At last, there is a better way to predict the default and unknown values. Using the complete data lines to predict the values, such as education, loan, housing and other categorical. There is one thing should be noticed is that transforming from classification variables to numerical values is both necessary useful in Pytorch and Sklearn.

In the implementation, the random forest is used to deal with it. Simply, it is made up with many classification and regression tree (CART) [3]. For each tree, it uses the sample training set from the full set with putting back. It means that all the traits are randomly gotten and put from the data set. Thus, the appearance of the sample can occur many times in a training set for the tree, and it can never appear. In detail, given training set S, test set T, feature dimension F, the number of CART t, the depth of the tree d, the feature number on each node f, terminal condition: the least samples number on nodes s, the least information gain on nodes m, for the (1-t)th tree, i = 1-t. From the training set S, get the same scale data set with put and get operations from S, and make it the root. On condition of the current node satisfy the terminal condition, set the current node as the leaf node. In classification, the prediction of the leaf node is the class with the maximum number, and the probability is the proportion in the current sample set. Then continuously train other nodes, if the current node does not reach the terminal condition, then select f features from F without putting back. Find the most effective one-dimension feature k and its threshold thr, the sample on the bode kth

dimension less than thr should be distributed into left node, and the others into right nodes. Repeat the steps above until all the nodes are trained or marked as leaf node. Finally, all the CART are trained.

After reviewing, to make the variables reasonable to be calculated, the attributes values should be numerical evaluation.

1. Binary variable encoding.

In the data set, the attributes such as housing and loan can be encoding as binary value, 0 and 1. It only means the yes and no. It is the most kind of simple variable to deal with.

2. Ordered Classification Variable Encoding.

According to the data set, the attribute 'education' can be basically sorted by the influential weight, from 'illiterate', 'basic.4y' to 'university.degree' and 'professional.course'. Assign the sequential number from 1 to the values.

3. Disorder Classification Variable Encoding.

This is the most characteristic part to deal with. For the date attributes, it has the meaning and order in fact, however, it makes no sense towards the variable values in training as original. So, it is also necessary to make it more reasonable for the training algorithm. In preprocessing part, the month, dayofweek, the job, marital and contact are also needed to be encoded. Generally, it can be encoded by the dummy variables. For example, the 12 job classifications can be represented by 11 dummy variables, the dayofweek can be represented by six dummy variables, and the marital can be represented by 3 dummy variables as shown in figure 1:

Job	*V1*	*V2*	*V3*	*V4*	*V5*	*V6*	*V7*	*V8*	*V9*	*V10*	*V11*
Housemaid	0	0	0	0	0	0	0	0	0	0	0
Services	1	0	0	0	0	0	0	0	0	0	0
Blue-collar	0	1	0	0	0	0	0	0	0	0	0
Retired	0	0	1	0	0	0	0	0	0	0	0
Management	0	0	0	1	0	0	0	0	0	0	0
Unemployed	0	0	0	0	1	0	0	0	0	0	0
Technician	0	0	0	0	0	1	0	0	0	0	0
Self-employed	0	0	0	0	0	0	1	0	0	0	0
Unknown	0	0	0	0	0	0	0	1	0	0	0
Entrepreneur	0	0	0	0	0	0	0	0	1	0	0
Student	0	0	0	0	0	0	0	0	0	1	0

Figure 1

The extreme value in the dataset may mislead the result. For example, the maximum in attribute duration is above four thousand, however, the median is around two hundred. So, the discretization of continuous features is necessary. This can result in a more stable system by reduce the influence of the extreme value. The function qcut() in pandas can binning the data, and factorize() can be used to transform them into numeric values. Thus, regulating the contrast data values from huge gap and different units can be also helpful for preprocessing.

Due to the high cost of the prediction of the missing value especially for time consuming. It is a better choice to save the preprocessed data set locally. It can be used directly for training. So, in the function preprocessdata(), the preprocessed data is saved in the local file folder. It is really helpful. One thing should be mentioned is that the preprocessed data should be shuffled when return. As for the high dimension data set, principal component analysis could be used to abstract traits in low

dimension if necessary. In this experiment, 20 attributes are proper for the current dimension processing and training.

Network reduction can also show a good performance on the training process. As for the chosen data set, the attribute 'default' is actually the 'unknown' or 'no'. This kind of data does not make sense. The reasonable situation should be more certain values than default, especially for the binary values. This is only the value that should be reduced for contribution to a simpler network. This is kind of simple way in preprocessing. Furthermore, the principal component analysis is suitable for the dimension reduction. Different style of data set should be preprocessing with different kinds of algorithm [9], such as paired test, Wilcoxon signed ranks test and Mann Whitney u test.

2.2 Constructing Neuron Networks

In general, there are many kinds of neuron network. The test focuses on the convolution neuron network (CNN), and recurrent neuron network (RNN), particularly for their developed version, fully convolution network (FCN), and long short-term memory network (LSTM). Actually, CNN performs well on image recognition, and FCN replaces the fully connected layer with convolution layer for good image semantic segmentation [17]. Moreover, LSTM and nested LSTM performs excellent in text prediction with lasting memory [18]. In this part, the dataset chosen is the MNIST, which contains 60000 examples into two files separately for training and testing. It is one of the most popular handwritten digits recognition dataset from http://yann.lecun.com/exdb/mnist/. It includes the handwritten images of numbers from 0-9. So, it has ten possible targets. It is easy to establish a neuron network both by Pytorch and Tensorflow. Followed by the hidden layer before the softmax layer, two hidden layers are created. A number of weight and bias are created, and some noise is added to weight avoiding 0 gradient. Actually, as the ReLU is used, the initial bias should be a small value to avoid dead neurons. After that, loss function used by cross-entropy is defined including softmax. Evidence of an image aims to show the degree of its belonging to a classification. If it significantly differs from a certain classification, it should the negative. Otherwise, it should be positive. It can be calculated with weight W and bias B.

$$\mathbf{evidence}_i = \sum_j W_{i,j} x_j + b_i$$

Sigmoid function is generally used to normalize the input in range (0,1). However, it actually has apparent weakness of saturation. It means that in the two sides, the derivation comes to 0. So, it can dramatically reduce the updating efficiency of weight W and bias B. To solve this problem, cross-entropy is introduced to overcome the slow weight updating in cost function. 'a' is the real output of the neuron.

$$C = -\frac{1}{n} \sum_x [y \ln a + (1-y) \ln(1-a)]$$

Softmax is used to show the probability of the likelihood of the image belonging to classifications instead of the absolute value 0,1 (true, false). Certainly, all the possibilities are summed to 1.
After that, there are many strategies to be implemented in training, such as regular SGD, ADAM, Momentum and ADAGRAD. Actually, SGD has many developed version, such as batch gradient descent and mini-batch gradient descent. Basically, stochastic gradient descent (SGD) is an efficiency way to calculate the gradient of the cost function. Comparing to batch gradient descent (BGD), SGD updates the gradient as soon as dealing with a data instead of the whole training set. Thus, the processing speed of the BGD is much slower than SGD. However, while using the convex cost function, BGD can absolutely find the global optimum and local optimum for concave function. The SGD may be stuck in a saddle. To utilize the advantage of the two strategies, mini-batch gradient

descent (MBGD) is a trade-off way, $\Theta=\Theta-\alpha\cdot\nabla\Theta J(\Theta;x(i:i+n),y(i:i+n))$. For each batch in a iteration, only a batch with a fixed size is calculated for gradient. One of the limit for the methods above is the continuous training may result in overfitting, because the learning rate is fixed. To solve it, ADAGRAD and ADAM are the good choice, because they are all automatically adjusting the learning rate. In particular, when dealing with sparse data, ADAGRAD can use the large learning rate for the data with low frequency, and in contrast, small learning rate for the frequent data. Moreover, ADAM makes the learning rate in a stable field.

Furthermore, it is helpful to adjust the batch size and learning rate for comparison. Actually, parameter adjustment is important in training process.

2.3 Evolutionary Algorithm

Evolutionary algorithm (EA) has good performance on global optimum. In this article, it is thought adopt it to the pre-training. I think it is a good way to optimize the dataset after preprocessing and encoding. It is known that genetic algorithm (GA) allows that the data can be optimized with crossover and mutation to evolve into a better population [19] [20]. It can be represented by two ways, binary and real number, such as float encoding. Within the maximum iteration times, set a population to evaluate the fitness value indicating adaptability for individuals. Actually, there is a little difference between EA and GA. In GA, choosing the good individuals is followed by crossover for these excellent individuals. In contrast, EA allows that every individuals can experience the crossover process, however, the children performing under requirement level are going to be killed. The proper individuals will be pushed to crossover pool. The explicit strategy should be adjusted by the actual problem. For example, if the target is to get the smallest one, then the data with small gene should be attached to high probability. After the cross over process, there are mutation process for individuals without direction. In this process, it is implemented by a mutation probability which is set in advance. It can help with diversity and it may result in jumping out of the local optimum. The process terminate after the fixed cycles. Actually, the tool box almost contains all the methods, and the experiments in this article concentrate on the changing of the tool in the toolbox, such as ‘mate’, ‘population’, ‘select’, ‘mutate’ and the probability for the crossover to get relative excellent performance according to the DEAP documentation. In detail, the pre-training concentrates on the feature selection. It is helpful to the data preprocessing by optimize the dataset. Selecting more relevant data and abandoning redundant features. In this way, the model can performance well instead of misleading by the fixed learning rate on redundant and flaw data.

3 Results and Discussion

This part aims to compare the training results from original to the preprocessed data. Furthermore, the result is comparing to the result in another paper using the same data set from Moro, Cortez and Rita [8]. The precision recall (BP) will be shown for comparing result here. Actually, when comparing, the implementation of preprocessing behavior limited. That means, much more work should be done further. Moreover, the performance between neuron networks will be compared, and further discussion for improvement is explored. As for the limited EA experiments, the result will also be discussed here.

3.1 Data Set Preprocessing and Evaluation

The data set division with a cross validation instead of only training set and test set has a good performance. The training set accounts for sixty percent of the full data, and the twenty percent for cross validation set and test set respectively. In this process, the best model can be chosen from the

cross validation set. And the result using the best model can be tested in the test set. This is really reasonable way to do so.

One thing should be recognized is that the number of negative data is dramatically larger than the positive. However, in preprocessing, balancing the influence directly with more classification cannot that make sense. So it is really encouraged to review the target task, predicting the binary result of whether the customer will buy the fixed deposit. In this part, synthetic minority oversampling technique (Smote) algorithm is helpful to do the over sampling and randomly discarding to do under sampling. The basic idea of the Smote algorithm is that for each sample x in minority classes, the Euclidean distance is used as the standard to calculate the distance to all the samples in the minority class, and its k-nearest neighbors are obtained. Then a sampling ration is set according to the sample unbalance ratio to determine the sampling ration N. For each minority sample x, several samples are randomly selected from its k-nearest neighbors to construct a new sample. For the data of this experiment, in order to prevent the noise of the newly generated data from being too large, only new numerical variables are actually newly generated in the new sample, and other variables are consistent with the original sample. It actually generates a good effect to eliminate the unbalance. And it make the minority classes more effective. Nevertheless, it does not improve the information details, so it may lead to overfitting. For the randomly discarding data from majority class can pose a threat to the missing of valuable information. In conclusion, the balance should be noticed, and more reasonable methods should be raised here.

For the preprocessing evaluation, firstly, training accuracy is about 88%. However, one factor should be considered to change the strategy. Because the negative values in the data set account for the dramatic majority part. That means predicting the customer cannot purchase the fixed deposit product works well. However, it does not make sense. Because the training target is to find the customer group who are more likely to purchase the fixed deposit. That means, the standard should be set as positive value. After the parameters adjustment, the suitable parameter in random forest model performs well. In detail, over sampling the positive data with about 8 times and under sampling the negative data with 1 times performs well when threshold is around 0.5, which means separate the samples with predicted probabilities comparing to 0.5. The result from the original data set is around 91.0 percent, while after the preprocessing, the result comes to 92.6 percent in average. Nevertheless, the improvement is relative limited. This means that the preprocessing done is not that reasonable enough to predict the analysis the customers behavior. It is reason that the cascade correlation is mentioned in this paper. The improved model may make the data more reasonable after faster and deep learning. And the reduction of ‘default’ cannot be responsible for the result after test. Another part can improve the performance, is to find the better way to solve the unbalance in the data set between the negative values and positive values. It is influential when comparing the tests.

Comparing with relative paper using the same data set mentioned above, this is a relative fail case for preprocessing. Because the result is less than the results in other paper. The work should be continuously done further especially for more specific and efficient preprocessing work. However, it is really improved by the preprocessing even though it is a little development. The limited improvement may result from the not precise parameter adjustment and misleading way to choose the meaning traits. That means in preprocessing, giving the reasonable numeric values to different attributes matters in training process. Proper preprocessing can be helpful to accelerate the speed and enhance the precision. Another way should be involved in is the more specific parameter adjustment in more sensible models. Nevertheless, the score improvement should balance well with the data in advance and process.

3.2 Neuron Networks Comparison

Firstly, for a same network, parameter adjustment can be a useful factor for training process. In this article, two elements are experimented, learning rate α, and batch size m. In the result, very high

accuracy dramatically over 1, though it is meaningless, can lead to exponent increase of the loss since it can result in misleading. However, in MNIST dataset, it has never occurred. Generally, high learning rate may converge to local optimum, and low learning rate may lead to close to linear. However, proper learning rate can have a rapid convergence and lead to global optimum. Actually, it is difficult to fix a proper learning rate. However, small learning rate may lead to a slow convergence and large learning rate can result in significant fluctuation which can obstacle the convergence process. Moreover, fixed learning rate after a long period of training, it may lead to overfitting. The efficiency ways include drop out, and adjusting learning rate. As mentioned above, learning rate can be automatically changed after fixed m times iteration, or followed by reaching a threshold for cost function. To solve this problem, the ADAM and ADAGRADE actually performs better because of proper learning rate changing strategies. The summary result for learning rate is shown in figure 2.

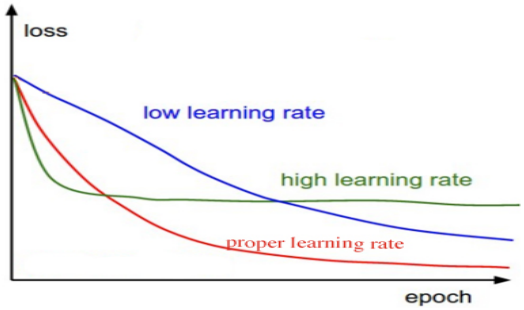


Figure 2

As for batch size, it can influence the amplitude below. That means the small mini-batch size can reduce the data utilization. Large mini-batch size can reduce the updating speed of weight and bias.

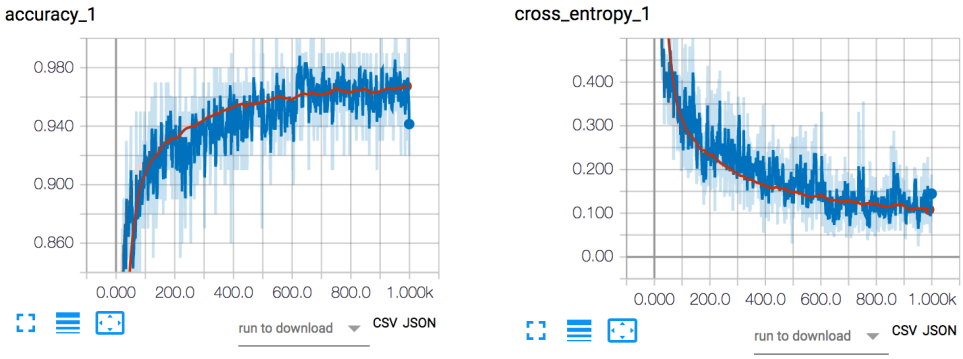


Figure 3

For the comparison of CNN and RNN, FCN deriving from fully connected forward network experiences a resizing process from getting small to retrieving to the original size to predict every pixel classification explicitly. In up-sampling, increasing steps perform better than one step with coordinate assistance from traits in different steps. However, FCN has a low distinguishing ability from different depth layer traits. This may lead to the waste of high dimension traits. LSTM deriving from RNN solve the problem of gradient disappearance in time axis. It performs higher accuracy than ANN and CNN in this experiment actually. It can be imaged that LSTM or nested LSTM can have a more reasonable and accurate result because of the lasting memory particularly for natural language processing.

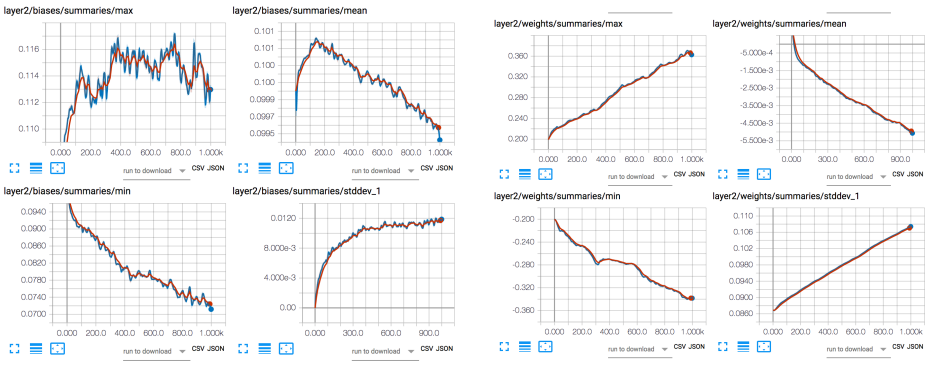


Figure 4

3.3 Evolutionary Algorithm

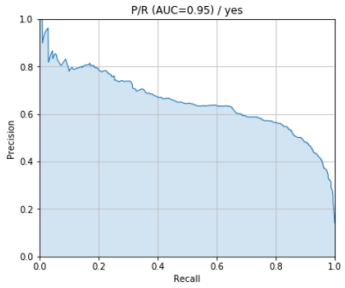
The genetic algorithm for feature selection has a little developed performance for the experiment. In the result, it seems like the pruning of the data, and the beginning performance correspondingly reduces. In the representation of binary form, the attributes which are on behalf of 0, are deleted by the training process. Accordingly, the optimized attributes are used to adopted for training in neuron networks. However, it really improve the efficiency of the training process, even though the saved calculation is occupied by the GA process. One confusion thing is that the one target GA process is easy to implement. In contrast, the dataset generally contains many targets. In conclusion, EA can actually improve the training performance by pre-training to some extent, and it allows robust potential parallelism. However, it should make more sense on the risk of selecting, particularly when training complex dataset. The choosing of the parameters are all depends on the experience. Actually, it costs much time in experiment. It may come from the deficiency of the real time feedback. Another problem occurs randomly. The result possibly convergence to a local optimum too early, and the reduction of the stride may loss the opportunity to get the global optimum.

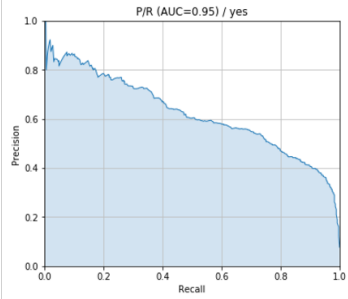
4 Conclusion and Future Work

There is much work to do for preprocessing in the future. In result above, the accuracy cannot reach above 95 percent, even though the this certainly improve the performance of training. However, preprocessing contributes to the training result, especially for the accuracy. In this model and data set, it originally has a relative high score, so the improvement is not that apparent. It can be improved in many aspects. The derived attribute may be helpful to do the work, such as combine the loan and marital to predict the remaining expenditure pressure. It make the values more reasonable to predict. Further, the way to solve the unbalance between negative and positive data should be raised more reasonable methods. However, the parameter adjustment is also one of the most necessary step to enhance the network performance. Apparently, adopting cascade correlation network and other models to improve the performance may be more directly helpful. In conclusion, the preprocessing can make the neuron network more reasonable and fast. As for the different kinds of neuron networks, proper design should be adopted to suitable dataset and usage. For example, the Segnet performance well on image semantic segmentation, while the LSTM has a good prediction on natural language processing. The EA actually has effect on data for pre-training, but it should both trade off the risk of abandoning. The future work should focus on the weakly supervised training. In conclusion, the data can be more useful with proper dealing in ANN training.

5 Appendix

The PR curve on cross validation set and test set.





References

1. Azoff, E. M. (1994). *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc..
2. Saar-Tsechansky, M., & Provost, F. (2007). Handling missing values when applying classification models. *Journal of machine learning research*, 8(Jul), 1623-1657.
3. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.
4. Suits, D. B. (1957). Use of dummy variables in regression equations. *Journal of the American Statistical Association*, 52(280), 548-551.
5. Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3), 37-52.
6. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
7. Lin, T. Y. (2002). Attribute (feature) completion-the theory of attributes from data mining prospect. In *Data Mining, 2002. ICDM 2003. Proceedings.* 2002 IEEE International Conference on (pp. 282-289). IEEE.
8. S. Moro, P. Cortez and P. Rita. (2014). A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31.
9. Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research.* 7(Jan), pp.1-30.
10. Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In *Advances in neural information processing systems* (pp. 524-532).
11. Bustos, R. A., & Gedeon, T. D. (1995). Decrypting Neural Network Data: A GIS Case Study. In *Artificial Neural Nets and Genetic Algorithms* (pp. 231-234). Springer, Vienna.
12. Milne, L. K., Gedeon, T. D., & Skidmore, A. K. (1995). Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. In *Proceedings Australian Conference on Neural Networks* (pp. 160-163).
13. Hagiwara, M. (1990, June). Novel backpropagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference* on (pp. 625-630). IEEE.
14. Zheng, Z. (1993). *A benchmark for classifier learning*. Basser Department of Computer Science, University of Sydney.
15. Khoo, S.,& Gedeon, T. (2008, November). Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks. In *International Conference on Neural Information Processing* (pp. 236-243). Springer, Berlin, Heidelberg.
16.Treadgold, N. K., & Gedeon, T. D. (1998, May). Exploring architecture variations in constructive cascade networks. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence*. The 1998 IEEE International Joint Conference on (Vol. 1, pp. 343-348). IEEE.
17. Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 22*(10), 1533-1545. Retrieved on 8th May 2018, from doi:10.1109/TASLP.2014.2339736
18. Sundermeyer, M., Ney, H., & Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23*(3), 517-529. Retrieved on 8th May, 2018 from doi:10.1109/TASLP.2015.2400218
19. Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.
20. Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, *3*(2), 95-99.