Using Neural Network Techniques to Predict Possibility of Default Payment on Credit Card

Yexiao Lin
College of Engineering & Computer Science
Australian National University
u6257745@anu.edu.au

Abstract: In this report, we focused on processing the monthly default payment data from Taiwan customers, which is collected from April to September in year 2005. Various techniques were implemented, such as preprocessing[1] the input data, changing to deeper neural network, applying dropout regularization[2], applying batch normalization[3], comparing with different activation functions, using RPROP[4] optimizer and applying genetic algorithm(GA)[5] to neural network. Those techniques were aimed at improveing training and testing acuracy of customer's default payment prediction, decreasing runtime cost and reducing overfitting as well.

Keywords: preprocessing[1], dropout regularization[2], RPROP[4], GA[5]

1 Introduction

1.1 Problem Background

With the tremendous increase of data today, a large amount of data provides different industries with quentities of opportunities, but at the same time, various types of information have brought out difficulties for training neural network in deep learning. In many cases, users should preprocess raw data for the convenience to calculate at first, with the input data for training and testing being tranfered to numerical value. Also, the meaningless data and misrepresenting data should be modified or omitted. On the other hand, to avoid overfitting situation, it is always necessary to normalize and dropout numerical input data as a preprocess method to improve the generalization ability and increase the precision of artificial neural network models.

To train large amount of data, simple fully connected neural networks has certain drawbacks such as the manual parameters selection, once improper parameters input, this training will come to failure. Other training method should be applied as an optimiser to automatically update weight and learning rates to adjusting trainning speed as well as improving the reliability of outcome.

1.2 Reasons for Choosing the Data Set

The following data is selected from UCI Machine Learning Repository, named as Default of Credit Card Clients Data Set (I-Cheng Yeh, 2016)[6], collected from April to September in 2005, which kept tracks of 30,000 customers' default payments in credit card. It consists of 30000 instances, which contains 30000 customers' relevant credit data, 23 input attributes, which stand for customers' relevant information such as given credit limit, gender, education, marital status, age, history deferred payment status, bill statesment and previous payment. The output attributes are binary, which represent 1 for default payment the next month, 0 otherwise.

Using this data set we need to predict their next's months default payment based on their history conducts. The motivation for trainining and testing this data is on the basis of risk management. The estimated outcome of neural network is responsible to evaluate customers' credit rating, determine their monthly payment limit in the future. The forcasting neural network is more valuable than other classification model dependent on its efficient to combine various types of personal factors to make prediction, which do not need large amount of time and cost input for human to complete.

1.3 Method Section

1. Preprocess technique[1]

Before training the neural network data, it is necesssary to decrypt data to transfer their value in the original data file to assure it becomes meaningful, so as to be convenient to process in the next steps' calucating. In the original data file, the information has decrypted by the dataset author to some extent. The resonable outcome has changed to binary numerical value as 1 represent they will have default payment next month and 0 otherwise. For convenience, we need to remove the first row in initial data file, which respresent the identified number of each column and also the second row, which explains the meaning of numerical value in each column such as the customer's given credit amount limit, gender, education, marital status, age, monthly payment status, amount of bill statement and amount of previous payment as well.

2. Normalization of input data

In part of normalizing input data, to reduce overfitting, we chose the z-score standarization method[7], which input data is normalized according to the mean and standard deviation of the original data (Kreyszig, 1979)[7]. After processing, the input data conforms to the standard normal distribution. Another nomalization method called rescaling using formula is x' = (x - min(x)) / (max(x) - min(x)), where x is the initial input value, x' is the processed value. After experimenting, we found that using z-score standarization[7] method generally would performances slighter higher accuracy than using rescaling method using the same parameters and input data. Therefore, we using z-score standarization[7] method as following experiment approaches of normalizing training and testing data.

3. RPROP optimizer

It is well known that back propagation (BP) is a supervised learning method to train neural network. But in face of large volumes of data, it may cost much time to train and make classification. Also, using BP we need to specify many parameters and hyper-parameters such as learning rates and weights in each iteration. In addition, in BP, the learning rate is a fixed η if users do not change it manually, to train a neural network using single fixed η, when learning neuron is far away from the output layer, its learning speed can be much slower. To resolve these problems, in this neural network, we choose RPROP technique to speed up the training (Riedmiller and Braun, 1993) [4]. It is an improved method compared to BP with merely applying the sign symbol for the gradient. It is faster than BP, the learning rates can be automatically fixed up according to changed value of each weight. RPROP algorithm[4] does not need user to change parameters each step, it avoides the situation when inappropriate learning rate may cause bad learning outcome. On the other hand, RPROP technique[4] also resolve the diffusion of gradient. In RPROP[4], the variation of weight ΔWi,j directly equals to ηi,j(t), so the gradient of loss function will not affect the changing value of weight, but merely decide the increase or descrease direction of weight. If the gradient of loss function is positive, we need to reduce the corresponding weight, it equals original value minus ηi,j(t), otherwise, if the gradient is negative, we need to make original weight plus ηi,j(t). The basic principle is as follows:

$$\Delta\omega_{i,j} = \begin{cases} -\eta_{i,j}(t), if g(t) > 0 \\ +\eta_{i,j}(t), if g(t) < 0 \\ 0, otherwise \end{cases}$$

Above description clarifies how the weight is updated in each step. Then we will explan the updating steps of learning rate ηi,j(t). At first, we should consider the signs in gradiant of t and (t-1) step. If at these two points their sign symbol is different, which means we have crossed the minimun value when at the point, the last upstate step across weights is too large. Then ηi,j(t) should be less than ηi,j(t-1) to slow down training to explore more precise minimun value. To achieve this, multiply ηi,j(t-1) by ηup, where ηup is a numerical value that less than 1 and greater than 0. Conversely, with the same signal of weight, which means that we have not reach the minimun value of loss function, then the learning rate will accordingly gets increased to speed up training neural network, to achieve that, multiply ηi,j(t-1) by a numerical value ηdown, where ηdown is greater than 1. The basic learning rate ηi,j(t) updating algorithm is as follows:

$$\eta_{i,j}(t) = \begin{cases} \eta^{up}\eta_{i,j}(t-1), g(t-1)g(t) > 0 \\ \eta^{down}\eta_{i,j}(t-1), g(t-1)g(t) < 0 \\ \eta_{i,j}(t-1), otherwise \end{cases}$$

Empirically, the default learning rate is always set as 0.1, with ηdown set as 1.2, and ηup set as 0.8 (Riedmiller and Braun, 1993) [4].

4. Genetic Algorithm

Genetic Algorithm[5] is a parallel stochastic search method based on simulation of natural genetic mechanism and biological evolution. The basic operation of genetic algorithm is divided by 3 steps. Firstly, select process, which refers to the probability of selecting individuals from the old group to the new group. The probability of individuals being selected is related to fitness value and fitness value of individuals. Secondly, crossover process, which refers to choose two individuals and testing the individual through the combination of two chromosomes. The crossover process involves selecting two chromosomes from a population and randomly selecting one or more for unknown exchange. The third process is mutation, which involves selecting an individual from a size of population and selecting a mutation from a chromosome point to produce a better individual.

2 Method

2.1 Preprocessing Data

At first, the preprocessing[1] method needed to be applied after the original default of credit card client.data[6] was downloaded. After transfering to a csv file, in order to simplifying the dataset, we made the copy of the original default of credit card client.csv file and renamed it as default of credit card clients process.csv. We noticed that the first row in the file which stands for the ID of each row was redundant. The second row which represents explanation of each column was also meaningless. Then we deleted first two rows and saved the default of credit card clients process.csv file.

We also checked the numerical value of each column to avoid the case they were out of range or misunderstood. The output attribute was set default payment of yes = 1 and no = 0, recored in column X24. The input attribute column was taged as X1 to X23, where X1 is numerical value, which stands for the amount of client's individual given credit and his family credit altogether. X2 represents gender, with 1 equals male and 2 equals female. X3 stands for education level of clients, with 1 representing graduate school, 2 representing university, 3 representing high school and other clients being classfied to 4. X4 stands for clients' individual marital status with 1 represents married, 2 represents single and 3 represents others. Column X5 represents client's age. Column X6 to X11 represents clients' history payment status, with -1 equals to pay duly, 1 equals to payment delay for one month, 2 equals to payment delay for two months and so on until the number 9 equals payment delay for nine months and above. Column X12 to X17 records amount of bill statement from September,2005 tracing back to April,2005. Column X18 to X23 recores the previous payment dollar from September traced back to April similarly.

The numerical value also need to be normalized to avoid overfitting also increase training model' generalization ability. Here we use z-score standarization method[7], the initial input data was minus by its mean then divided by its standard deviation.

Finally, we randomly splited the normalized data into 80% training set and 20% testing set.

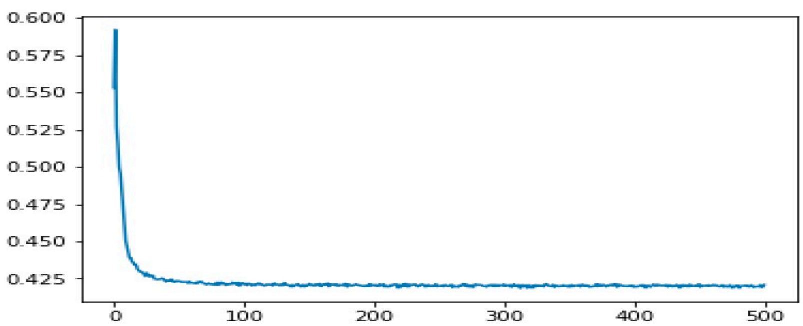
2.2 Define Multi Layer Neural Network

1. Adjust layer demension of neural network

Since the input attributes is 23, and output attributes have 2 results. We set input_neurons = 23, output_neurons = 2, num_epoch = 500, also use RPROP[4] as optimiser. Firstly, we used a single hidden neural network, set hidden neuron number as 60 but after training numbers of 500-epoch experiment, no matter what value of initial learning rate was, the accuracy for training and testing was all in the range of (0.74, 0.76). Therefore, we increased the hidden layer demension, use first hidden layer as Tanh function with 60 neurons, second layer as Sigmoid function with 40 neurons. The results of training accuracy get increased, but the testing accuracy stayed in the range of (0.74, 0.76). With the increase of hidden layer demision, the training accuracy kept increasing slightly, but the testing accuracy remained below 77%. After increasing hidden layers dimension number to 4 with 3 Tanh activation functions and last one with Sigmoid activation function, the training accuracy increased to 79.00%, testing accuracy still got 75.81%.

2. Apply dropout regularization

Since dropout[4] can be used to randomly frozen hidden neurons to reduce overfitting, we applied the dropout function[4] in torch.nn model to randomly dropout 20% hidden neurons after the first hidden layer, then trained 500 epochs again. The training accuracy finally changed to 78.39%, testing accuracy increased to 77.79%. After changing the dropout rates and change its position in hidden layers, the outcome was not better than the first dropout function training. The dropout regularzation made the decline of loss curve graph not smooth anymore but reduced the overfitting problem efficiently. The historical loss is now as follows:



3. Change activation function

There were many activation functions, then we choice 3 Tanh functions in 3 hidden layers to random number and random position of ReLU function and Sigmoid fundtion, but the outcome of training and testing accuracy was not increased but got slightly decreased conversely. Therefore, we kept the original 3 layer's Tanh function and last layer's Sigmoid function.

4. Apply Batch Normalization

Noticed that training and testing accuracy was difficult to get increased with different initial learning rate, to prevent the case of diffusion gradient, try to improve the training and testing accuracy, we applied batch normalization[3] to prevent diffusion gradient situation after the second and third hidden layers, with result of only slightly improved the accuracy of testing set by 0.07%. The structure of neural network was as follows:

```
input_neurons = n_features
output_neurons = 2
learning_rate = 0.11
num_epoch = 500

net = nn.Sequential(
        nn.Linear(input_neurons,60),
        nn.Tanh(),
        nn.Dropout(p=0.2),
        nn.Linear(60,40),
        nn.Tanh(),
        nn.BatchNorm1d(40),
        nn.Linear(40,30),
        nn.Tanh(),
        nn.BatchNorm1d(30),
        nn.Linear(30,10),
        nn.Sigmoid(),
        nn.Linear(10,2),
        )
loss_func = torch.nn.CrossEntropyLoss()
optimiser = torch.optim.Rprop(net.parameters(), lr=learning_rate)
```

5. Adjust hyper-parameters

Finally, we changed the learning rate, dropout rate, ratio of test set and train set, two η values in RPROP[4] optimisers many times and finally determined when learning rate = 0.11, dropout rate = 0.2, ratio of training set changed to 0.9, two η values in RPROP kept original as 1.2 and 0.8 would get the best testing accuracy 80.92 %. Some recoded data examples are shown below:

Ratio of Training set	Learning Rate	Dropout Rate	Testing Accuracy	Training Accuracy
0.9	0.11	0.2	80.92%	80.25%
0.9	0.12	0.2	80.07%	80.10%
0.9	0.1	0.3	79.72%	80.26%
0.8	0.09	0.2	79.57%	80.50%
0.8	0.11	0.2	78.95%	80.44%
0.9	0.05	0.2	78.98%	80.52%

2.3 Compare Results after Applying Genetic Algorithm

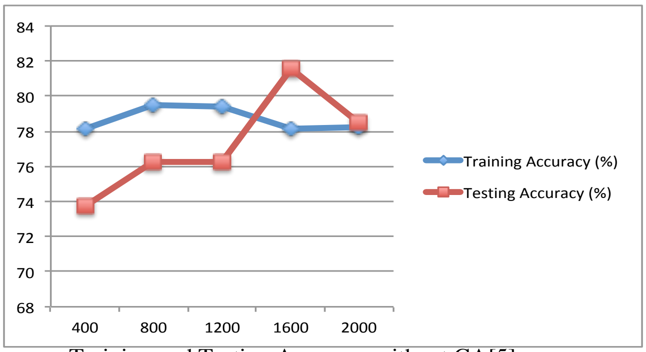
Next step, we used Genetic Algorithm[5] adding to a single layer neural network, then compared their runtime cost and training and testing accuracy with different extracted number of instances.

1. Preprocess data

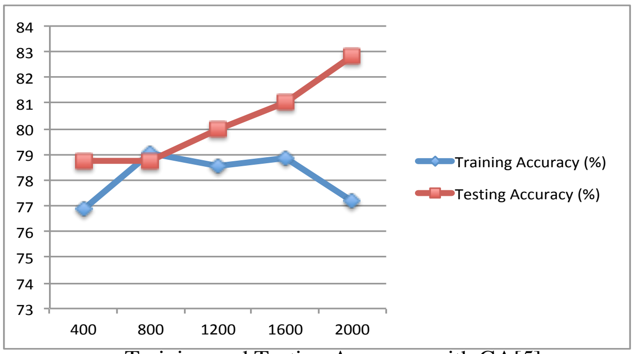
Realized that the multi-layer neural network experiments cost much time to train and test. Every time we adjusted hyper-parameters, we needed to experiment with around 70 seconds to run 500 epochs. Therefore, the preprocessing technique[1] was applied again. Considered that the possibility of consumers' defalut payment mostly depend on their bill statements and past payment status, we simply extracted the first 2000 consumers' data in Default of Credit Card Clients Data Set[6] and mainly focused on the column X12 to X17 which stands for the consumer's amount of bill statement in the past 6 months and column X18 to X23, which represents the data about the consumer's amount of previous payment, we used these data to calculate the average of every consumer's past six months' bill statement amount and average previous payment amount. Then we treated 2 numerical values as input features, with X24 column recorded that if the next month's defalut payment would occur as output label to generate a new csv file. Then used z-score standarization[7] method to normalize input data as before.

2. Apply Genetic Algorithm

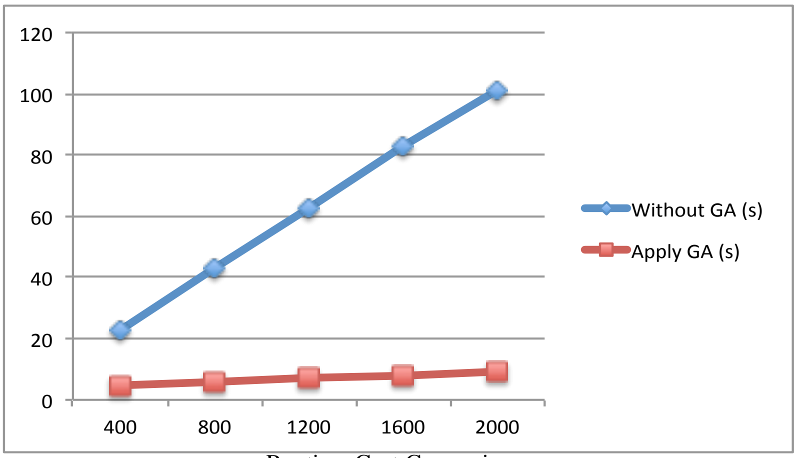
Firstly, we set a model of single layer hidden neural network, set the input size plus bias number as 3, hidden layer neurons' number as 8. Then we initialized population size as 50. In evolution process, we set crossover possibility as 0.7 and 0.2, mutation possibility as 0.2. When crossover happened, the two parent argument's random position index would be swapped. Finally, we set the max evolving interations as 1000 and compared the training runtime cost and accuracy results with a same demension single layer network with 1000 epoch numbers. We changed the number of instances from 400 to 2000 accordingly and the results were shown below:



Training and Testing Accuracy without GA[5]



Training and Testing Accuracy with GA[5]



Runtime Cost Comparison

The results show that the training and testing accuracy of both 2 implement methods' is around 80%, in single layer neural network, when instance size was smaller than 1400, the training accuracy showed slightly higher than testing accuracy, but with the instance number continued to increase, the overfitting problem was disappeared. However, after applying GA[5] approach, results show that no overfitting problem ocurred when instance size was smaller than 1400. Also, from the third curve graph we found the GA[5] approach largely decreased the runtime cost, when instance number increased, normal neural network approach's runtime cost got linear growth, conversely, the runtime of GA approach almost remained the same. Compared to simple fully connected neural network, adding GA[5] approach help to avoid overfitting problem and save a large amount of runtime cost.

3 Results and Discussion

To training neural network using Default of Credit Card Clients Data Set[6], we used various techniques such as preprocessing the input data[1], using deeper neural network, applying dropout regularization[2], using different activation functions in hidden layers, applying batch normalization[3], using efficient RPROP[4] algorithm as an optimiser and applying GA[5] to neural network. The optimal outcome of testing and training accuracy could not be improved greater than 83% at the same time. One assumption is that consumers' next months' default payment possibility will be affected by other unforeseen situation or emergencies, so the results cannot be fully predicted correctly by their past data.

In the literature, the relevant paper explores and introduces 6 different data mining techniques that can be applied to predict the accuracy of this data set[8] but does not processing with training neural network and does not have present any outcomes. In addition, no other relevant literature using this UCI credit client data set was found.

4 Conclusion and Future Work

The combination of techniques has been applied in order to efficiently speed up training multi-layer networks, reduced overfitting, avoided diffusion gradient and improved training and tesing accuracy. After experiments, results show that applying dropout[2], normalization[3][7] and GA approach[5] can effectively avoid overfitting problem, using Rprop[4] optimizer and GA[5] approach, adjusting hyper-parameters can help us get higher accuracy and speed up traing input dataset.

For the future, we will explore more efficient techique to train this dataset, such as applying convolutionary neural network to improve the accuracy of prediction of default payment of credit card clients.

References

1. Bustos, R., Gedeon, T.: Decrypting neural network data: A GIS case study. In: Artificial Neural Nets and Genetic Algorithms, pp. 231-234. Springer, 1995
2. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15, 1929-1958 (2014)
3. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
4. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: Neural Networks, 1993., IEEE International Conference on, pp. 586-591. IEEE, (1993)
5. Mitchell, M.: An introduction to genetic algorithms. MIT press (1998)
6. Yeh, I.-C.: Default of credit card clients data set. Department of Information Management, Chung Hua University, Taiwan (2016) https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#
7. Kreyszig, E.: Advanced Engineering Mathematics. Wiley, New York (1979)
8. Yeh, I.-C., Lien, C.-h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications 36, 2473-2480 (2009)