FIT5186 Assignment

Yuncheng Hua 29 May 2014

# Client's classification and clustering analysis based on neural networks in bank marketing campaigns

**Abstract**—In a banking institution, to distinguish clients who would subscribe banking products from all potential clients is important. If the clients who are willing to sign the contracts with bank can be detected correctly, the bank can sale the banking products efficiently. Neural network have been successfully employed in the banking sector for years, and it is suitable to classify this kind of clients for it is able to learn and recognize the features of clients' information. In our experiment, by designing and modifying the architecture of neural network, the accuracy of detecting a client who would subscribe the banking product is improved, therefore bank is able to sale more products efficiently.

## 1. INTRODUCTION

Commonly, in a banking institution, the marketing campaigns were based on phone calls. The bank workers make phone calls to clients to introduce the banking products, and if possible, help clients sign the products if they want. Provided the potential clients who would subscribe the contracts are able to be distinguished from those who would not, the bank workers can save a lot of workloads and resources to sale more products.

Suppose we have more than 2000 clients' personal information and the decisions they make to subscribe the banking products, the classification problem is to predict if the potential client will subscribe a banking product. The Nerual networks are capable of learning the features of data as well as the relationship between independent variables and dependent variables (Wong, Lai and Lam, 2000). In addition to that, since neural network makes no previous assumptions about the statistical distribution or properties of the data, it is more useful in practical situations (Smith & Gupta, 2000). Therefore in order to access if the product (bank term deposit) would be (or not) subscribed, neural networks are rational tools to classify the new clients as 'yes' or 'no' clients.

## 2. DATA SETS

The data set of clients' information in this experiment is derived from database: http://archive.ics.uci.edu/ml/datasets.html. The data is obtained from Paulo Cortez and Sérgio Moro, which was originally related with direct marketing campaigns of a Portuguese banking institution.

The number of instances of the data set is 2083. The clients' information is consisted of 16 attributes including age, job, marital status, education degree, credit status, balance, housing status, loan, the relation with the last contact, outcomes of the previous marketing campaign of clients, together with three attributes which represent the observed classification of each client as a 'yes' client (which means the client would subscribe the banking product) or a 'no' client (which means the client would not sign the product). Thus the clients' information is consisted of 19 attributes. Some kinds of the clients' features are numeric format, such as age and balance, others are categorical form, such as marital status and education degree.

In order to use the information to classify or cluster the clients, some kinds of categorical information features should be processed beforehand. For example, '0' should be used to represent that one client is not married yet, while '1' is used to describe that one client is married. Furthermore, the degree of education is measured on a standard scale of 1-4, which means that '1' represents 'unknown', '2' represents 'secondary', '3' represents 'primary' and '4' represents 'tertiary'. After processing the data, all information of clients is described in decimal numeral format; therefore the data are able to apply in SOFM to get the result.

Furthermore, by using 1-out-of-N method, the single column categorical data can be converted into numerical data contains several columns with binary variables. In this case, for example, we can use '0, 0' to represent a client is single, '0, 1' to represent a client is divorced and '1, 0' to represent a client is married.

## 3. EXPERIMENTS

### 3.1 Experiment 1

**Inputs & Outputs Definition**

In experiment 1, we use the numerical data and translated categorical data which is converted to numerical form without using 1-out-of-N method. So the number of attributes is still 16. We use all 16 features and 2 outputs. In outputs, '1,0' represents 'yes' client and '0, 1' represents 'no' client. In all data, 80% is used as training data and 20% is used as test data.

**Architecture and Parameters**

We choose the standard nets model, which means each layer connected to only the previous layer to make the classification.

In slab 1, since we have 16 features, slab 1 has 16 neurons. The scale function is linear [-1, 1], which is as default function.

In slab 2, the number of hidden neurons is 50, which is as default number. The activation function is logistic function, which is as default.

In Slab 3, since we have two outputs, the number of neurons is 2. The activation function is logistic function, which is default function.

In this experiment, the learning rate is set to 0.1, the momentum is 0.1 and the initial weight is 0.3.

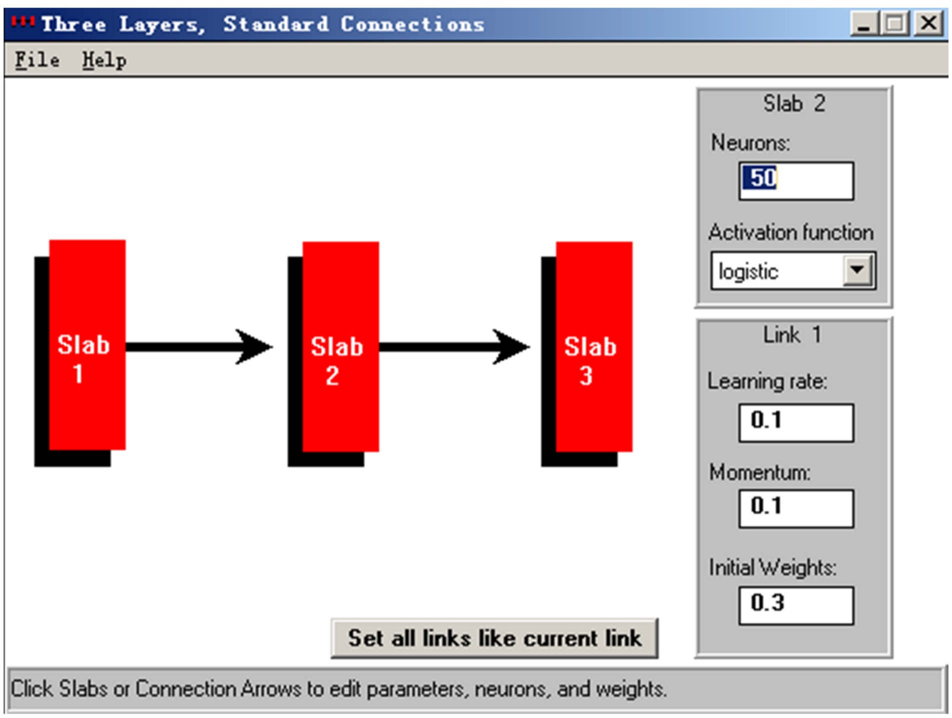


Figure 1 Architecture and Parameters using MFNN with 16-50-2

**Backpropagation Training Criteria**

Pattern selection is rotation, and weight updates is momentum. It also chooses Automatically Save Training on best test set. We choose learning will stop when average error < 0.03 in test set or in training set. Calibration interval is 200. Consider missing values to be error conditions.

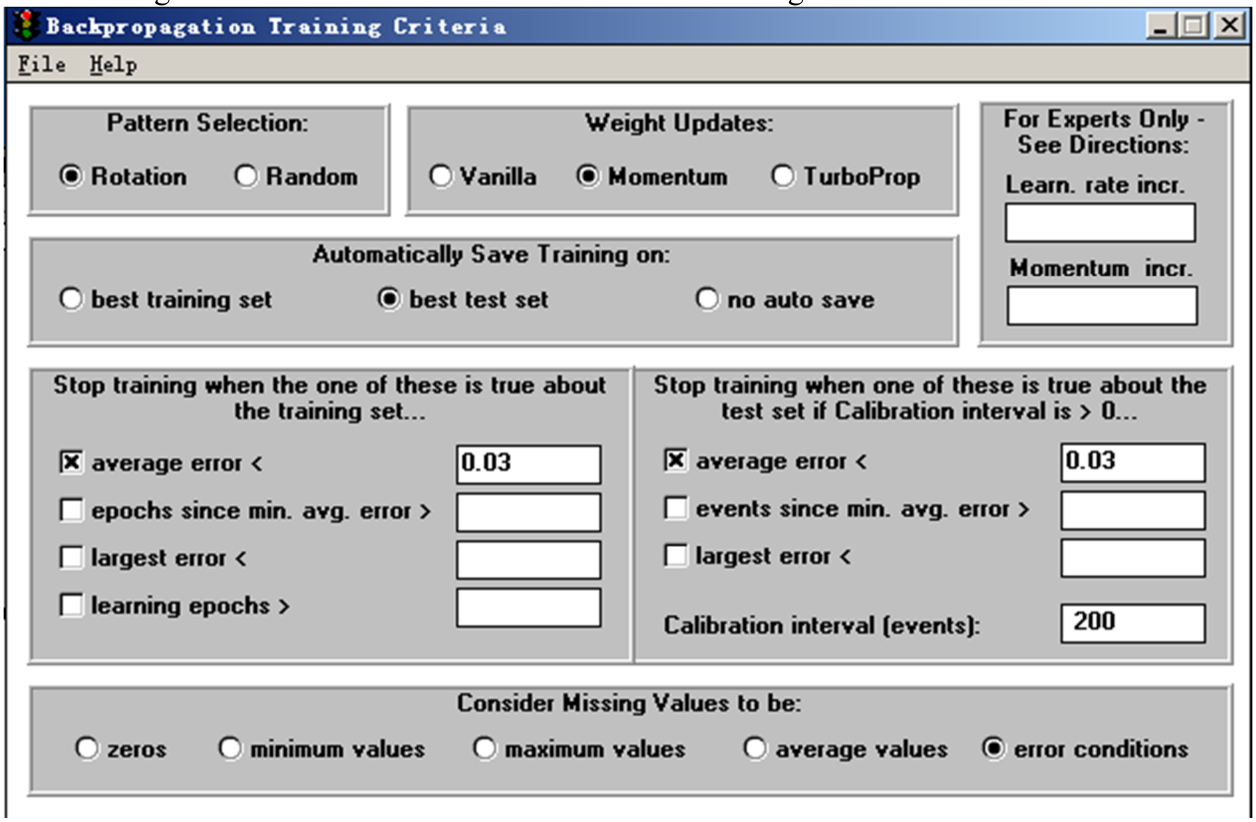

Figure 2 Backpropagation Training Criteria using MFNN with 16-50-2

**Learning**

After 11 minutes learning, since the average error of training set is smaller than 0.03, the learning is stopped.

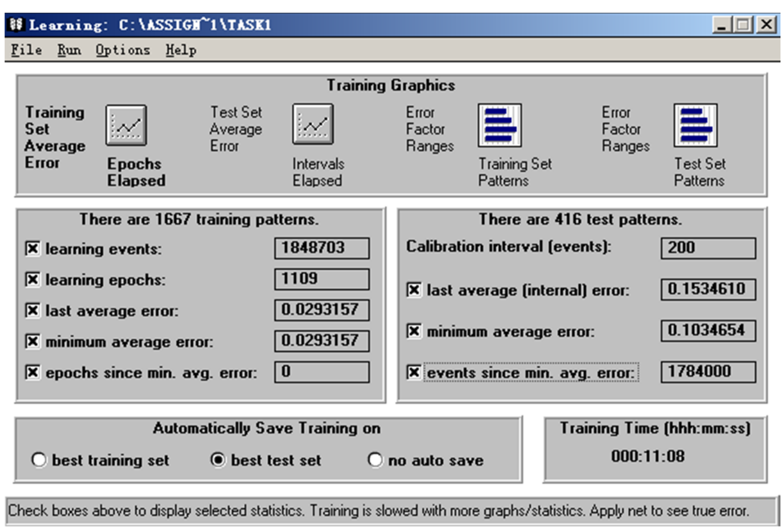

Figure 3 Learning result by using MFNN with 16-50-2

**Apply To File**

In others' study, normally R squared is used to be as the principal determinant of network quality (Flitman, 1997). In experiment 1, as figure 4 shows, the R squared is over 0.27.

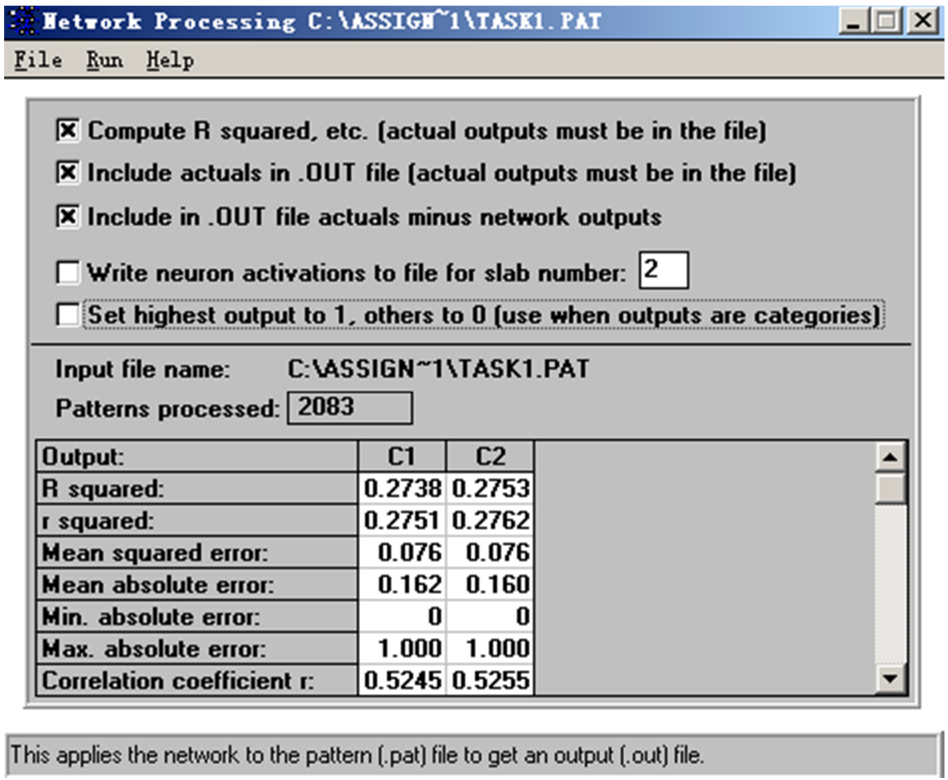


Figure 4 Output statistic by using MFNN with 16-50-2

**Result Analysis**

After the output data is exported and processed by using Excel, the classification results can be derived. The classification accuracy result is as form 1 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 63 | 186 | 25.30% |
| actually no | 41 | 1793 | 97.76% |
| column accuracy | 60.58% | 90.60% | |

Form 1 Classification accuracy using MFNN with 16-50-2

From the results, it can be concluded that the neural network was able to classify correctly 63 of the 249 known 'yes' clients who would subscribe the banking products, and 1793 of the 1834 known 'no' clients who would not sign contracts with bank. The trained neural network is capable of distinguishing a 'no' client 97.76% of the time, however recognizing a 'yes' client 25.30% of the time. When the neural network classifies a client as a 'yes' client, it is correct 60.58% of the time, meanwhile is correct 90.60% of the time when it classifies a 'no' client. It seemed difficult for the trained neural network to find a 'yes' client from all potential clients. For the bank wants more clients to subscribe the products, it is more important to classify a yes client from all clients. So we should modify the architecture of neural network to improve the accuracy of detecting a yes client.

### 3.2 Experiment 2

#### Inputs & Outputs Definition

In experiment 2, we use the translated categorical data by using 1-out-of-N method. So the number of attributes is increased to 30.

#### Architecture and Parameters

We still choose the standard nets model.

In slab 1, since we have 30 features, slab 1 has 30 neurons. The scale function is linear [-1, 1].

In slab 2, the number of hidden neurons is 57, which is as default number. The activation function is logistic function, which is as default.

In Slab 3, the number of neurons is 2. The activation function is logistic function, which is default function.

In this experiment, the learning rate is set to 0.1, the momentum is 0.1 and the initial weight is 0.3 as experiment1.

Backpropagation training criteria and learning process are as those of experiment1. After applying to file, the R squared is 0.2875, which is bigger than the R squared in experiment 1.

#### Result Analysis

After the output data is exported and processed by using Excel, the classification results can be derived. The classification accuracy result is as form 2 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 80 | 169 | 32.13% |
| actually no | 35 | 1799 | 98.09% |
| column accuracy | 69.57% | 85.87% | |

Form 2 Classification accuracy using MFNN with 30-57-2

From form 2, it can be concluded that the percentage of yes clients that are detected has been improved, which means using 1-out-of-N method will improve the performance of neural network to detect a yes client.

### 3.3 Experiment 3

#### Inputs & Outputs Definition

In experiment 2, we analyze the attributes that contribute the least to classify a yes client. The contribution of each feature can be derived in contribution factor of experiment 2, which is as figure 5 shows.

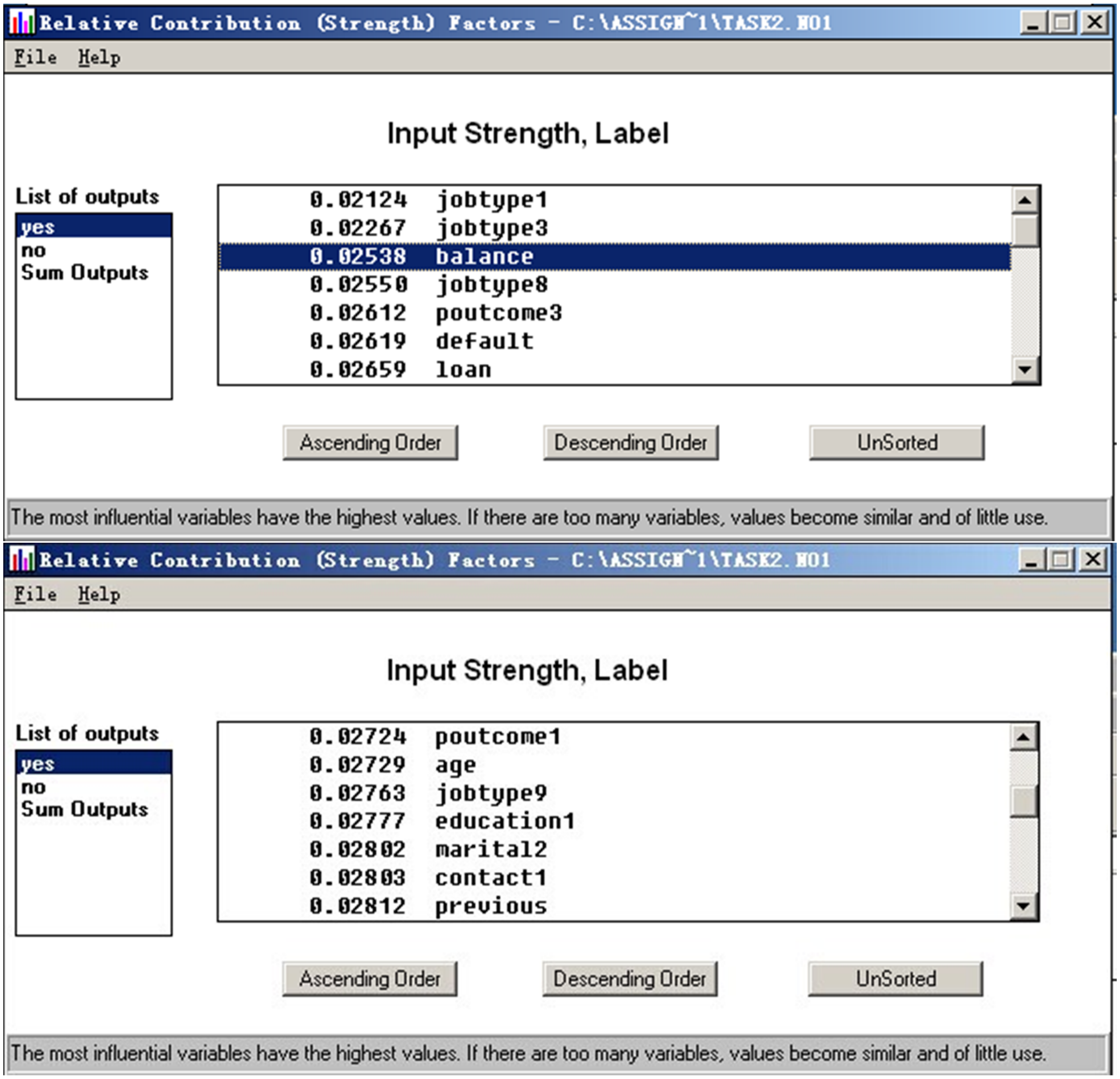

Figure 5 Contribution factors to detect a yes client by using MFNN with 30-57-2

From the figure of experiment 2, contributions of jobtype1/3/8/9, balance, poutcome1/3, default, loan, age, education1, marital2, contact1 and previous are the features that have the least contributions. Since jobtype, poutcome, education, marital and contact are unified features, we should consider all unified features. Thus, the features balance, default, loan, age and previous are removed from the input. Thus the number of input features is 25.

**Architecture and Parameters**

Compared with experiment 2, the parameters that need to be changed are number of neurons in slab 1 (from 30 to 25), the number of hidden neurons in slab 2 (from 57 to 54).

Other parameters are same as those of experiment2. After applying to file, the R squared is 0.2801, which is almost the same as the R squared in experiment 2.

**Result Analysis**

After the output data is exported and processed by using Excel, the classification results can be derived. The classification accuracy result is as form 3 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 81 | 168 | 32.53% |
| actually no | 29 | 1805 | 98.42% |
| column accuracy | 73.64% | 91.49% | |

Form 3 Classification accuracy using MFNN with 25-54-2

Comparing form 3 and form 2, it can be concluded that the percentage of yes clients that are detected has been improved slightly, which means removing some attributes which contribute the least to classifying a yes client will slightly improve the performance of neural network.

## 3.4 Experiment 4

In experiment 4, we choose another hidden layer's activation function in slab 2. In others' study, by using the Gaussian function as the hidden layer activation function, the experiment would gain the optimal configurations (Hoffman, Mielens, Omari, Rommel, Jiang and McCulloch, 2013). Therefore, in my experiment, I changed the activation function of hidden layer to Gaussian. Except this, all other parameters remain unchanged. The architecture and parameters using in experiment 4 is as figure 6 shows.

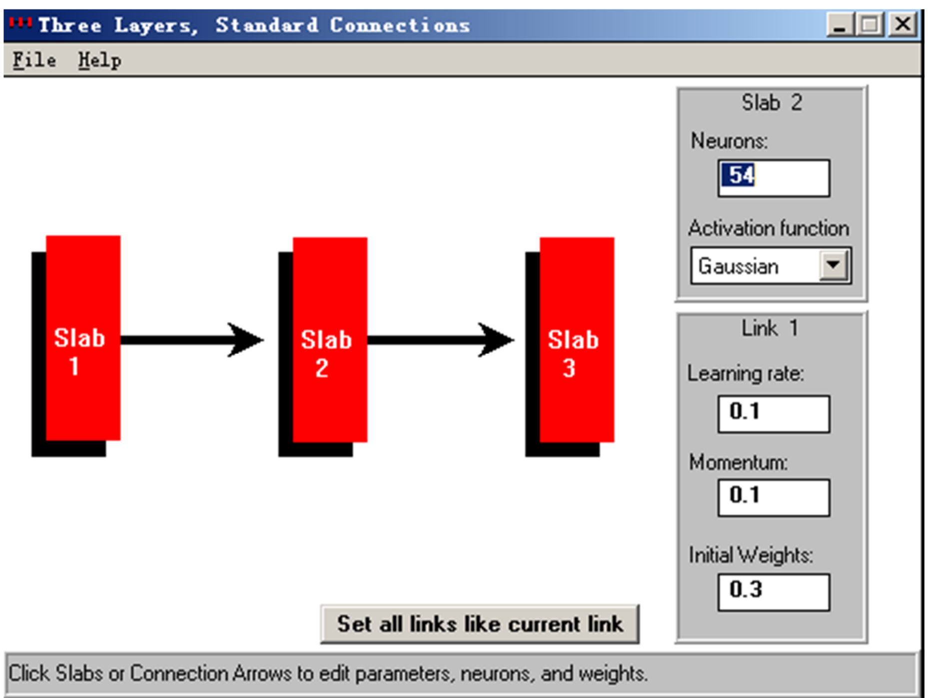


Figure 6 Architecture and Parameters using MFNN with 25-54-2 and Gaussian function

**Result Analysis**

After the output data is exported and processed by using Excel, the classification results can be derived. The classification accuracy result is as form 4 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 115 | 134 | 46.18% |
| actually no | 40 | 1794 | 97.82% |
| column accuracy | 74.19% | 93.05% | |

Form 4 Classification accuracy using MFNN with 25-54-2 and Gaussian function

Comparing form 4 and form 3, it can be concluded that the percentage of yes clients that are detected has been improved a lot, which means using the Gaussian function as activation function in hidden layer will improve the performance of neural network to detect a yes client.

**3.5 Experiment 5**

In experiment 5, we change the value of momentum from 0.1 to 0.4, and initial weights from 0.1 to 0.4. Other parameters are same as those of experiment 4. The classification accuracy result is as form 5 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 117 | 132 | 46.99% |
| actually no | 30 | 1804 | 98.36% |
| column accuracy | 79.59% | 93.18% | |

Form 5 Classification accuracy using MFNN with 25-54-2 and Gaussian function

Comparing form 5 and form 4, it can be concluded that the percentage of yes clients that are detected has been improved slightly, which means changing the values of momentum and initial weights will influence the performance of neural network at some extent.

Then in experiment 5_2, we change learning rate to see the influence on classifying results. The performance of the neural network is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the neural network takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the neural network moves across the performance surface.

Firstly, we set the learning rate to 0.3, and other parameters remain unchanged. After 35 minutes learning, the minimum average error of training set is converged to 0.10, which is too bigger than 0.03. Thus I changed the learning rate to 0.2. After 45 minutes learning, the learning result is like figure 7 shows. The minimum average error of training set is converged to 0.075, which is still too bigger than 0.03. So there two learning rates are both not appropriate, the learning rate 0.1 is better.

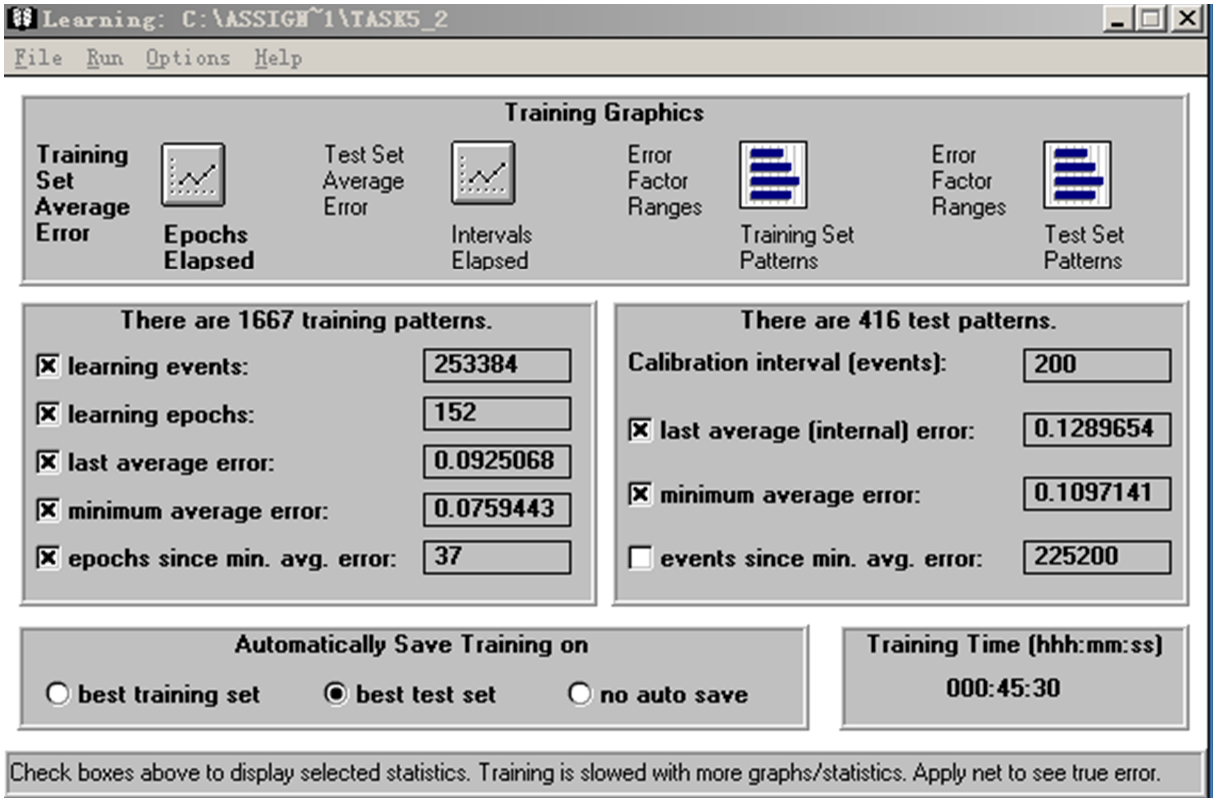


Figure 7 Learning result of using MFNN with learning rate 0.2

### 3.6 Experiment 6

In this experiment, we change the number of hidden neurons in slab 2. Set momentum, initial weights and learning rate as 0.4, 0.4 and 0.1. Use Gaussian function as the hidden layer activation function. Some basic heuristics to determine the numbers of hidden neurons have been found (Flitman, 1997).

Three examples are:

Number of hidden neurons = 1/2 (Inputs + Outputs) + Sqrt (Number of Patterns)

Number of hidden neurons= 2*square root (number of inputs or defining characteristics + the number of outputs or classifying characteristics) rounded down to the nearest integer.

Number of hidden neurons < = (Number of Patterns * Error Tolerance)/ (inputs + outputs)

Here, number of inputs is 25, number of outputs is 2, number of patterns is 2083 and error tolerance is 50%. So the number of hidden neurons computed by first function is 59, which is almost same as the default number of hidden neurons. The number of hidden neurons computed by second function is 10, and is 38 by the third function.

In experiment 6, we change number of hidden neurons to 10. The classification accuracy result is as form 6 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 32 | 217 | 12.85% |
| actually no | 6 | 1804 | 99.67% |
| column accuracy | 84.21% | 89.26% | |

Form 6 Classification accuracy using MFNN with 25-10-2 and Gaussian function

It can be concluded that, though the percentage of no clients that are detected has been improved, the percentage of yes clients has decreased a lot. The second function to compute number of hidden neurons is not suitable to the neuron network.

In experiment 6_2, we change the number of hidden neurons to 38. The classification accuracy result is as form 7 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 82 | 167 | 32.93% |
| actually no | 24 | 1810 | 98.69% |
| column accuracy | 77.36% | 91.55% | |

Form 7 Classification accuracy using MFNN with 25-38-2 and Gaussian function

Compared with form 5, it can be concluded that, the percentage of yes clients has decreased a lot, which means the third is not suitable to the neuron network. So the first function to compute number of hidden neurons has most satisfied result.

### 3.7 Experiment 7

In experiment 7, set momentum, initial weights, learning rate and number of hidden neurons as 0.4, 0.4, 0.1 and 54. Use Gaussian function as the hidden layer activation function. Change the output format from vector format to a single output. If the output value is 2, which means the client would sign the contract with bank, while 1 means the client would not subscribe the product. Here we introduced a new parameter which is threshold value. The threshold value is set to distinguish the clients who would sign contracts from those who would not. For instance, if we set the threshold value as 1.5, which means if the output value is bigger than 1.5, the client is likely to sign the contract, otherwise the client is probably not to sign the contract.

Because we want to distinguish more clients that would sign the contract, which means we want to decrease the number of the clients who are wrongly assessed as having no wish to sign the contracts, but de facto are willing to sign the contracts. So if we classify applicants as those who would sign contract if the output value is greater than 1.25, then the network is more likely to distinguish clients as yes clients.

The classification accuracy result by using threshold value 1.25 is as form 8 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 167 | 82 | 67.05% |
| actually no | 117 | 1717 | 93.62% |
| column accuracy | 58.80% | 95.44% | |

Form 8 Classification accuracy using threshold value 1.25

It can be concluded that the percentage of yes clients that are detected has been improved a lot. Furthermore, if we change the threshold value to 1.10, the result is as form 9 shows.

| | classified yes | classified no | row accuracy |
|---|---|---|---|
| actually yes | 203 | 46 | 81.53% |
| actually no | 339 | 1495 | 81.52% |
| column accuracy | 37.45% | 97.01% | |

Form 9 Classification accuracy using threshold value 1.10

From the form, it can be found that though the neural network was able to classify correctly 203 of the 249 known ‘yes’ clients, 339 clients had been misclassified as yes clients, which is not satisfied. If too many de facto no clients are misclassified as yes clients, the bank may waste many resources on communicating with these no clients.

## 4. LIMITATION

The number of instances used to train the neural network is relatively small. Provided there are more instances, the neural network is able to gain better classification result.

In addition, some parameters such as learning rate and initial weights remain to be modified to enable the neural network to gain better performance. More researches need to be conducted to derive the optimal value of these parameters.

## 5. CONCLUSION

In this experiment, we use neural network to solve the classification problem in bank marketing campaigns. By using neural network, bank is able to classify the clients who are likely to subscribe the banking products from the potential clients. The data come from the clients’ personal information. The data need to be translated into numerical form, therefore is able to be applied to neural network. After learning over 2000 examples of existing bank clients, the neural network can accomplish the task of distinguishing ‘yes’ clients (who are willing to sign the contracts with bank) from ‘no’ clients (who would not subscribe the banking products). By changing different parameters and modifying the architecture of the neural network, it can be found that the accuracy of detecting a ‘yes’ client is improved. Some researches remain to be conducted in future to modify neural network to gain better classification results.

**REFERENCES**

Smith, K.A, & Gupta, J.N.D. (2000). Neural networks in business: techniques and applications for the operations researcher. *Computers and Operations Research,* 27 (11), 1023-1044. DOI: 10.1016/S0305-0548(99)00141-0

Flitman, A.M. (1997). Towards analyzing student failures: Neural networks compared with regression analysis and multiple discriminant analysis. *Computers and Operations Research,* 24 (4), 367-377.

Wong, B.K. Lai, V.S., & Lam, J. (2000). A bibliography of neural network business applications research: 1994–1998. *Computers and Operations Research,* 27 (11), 1045-1076.

Hoffman, M. R. Mielens, J. D. Omari, T. I. Rommel, N. Jiang, J. J., & McCulloch, T. M. (2013). Artificial neural network classification of pharyngeal high-resolution manometry with impedance data. *The Laryngoscope,* 123 (3), 713-720. DOI: 10.1002/lary.23655