

Classification of Churn and non-Churn Customers for Telecommunication Companies

Tarik Rashid

*Computing Faculty/Research and Development Department
College of Computer Training (CCT)
102-103 Amiens Street, Dublin1, Ireland*

tarik@cct.ie

Abstract

Telecommunication is very important as it serves various processes, using of electronic systems to transmit messages via physical cables, telephones, or cell phones. The two main factors that affect the vitality of telecommunications are the rapid growth of modern technology and the market demand and its competition. These two factors in return create new technologies and products, which open a series of options and offers to customers, in order to satisfy their needs and requirements. However, one crucial problem that commercial companies in general and telecommunication companies in particular suffer from is a loss of valuable customers to competitors; this is called customer-churn prediction. In this paper the dynamic training technique is introduced. Dynamic training is used to improve the prediction of performance. This technique is based on two ANN network configurations to minimise the total error of the network to predict two different classes: namely churn and non-customers.

Keywords: Artificial Neural Network, Classification, Prediction, Dynamic Training, Telecommunication.

1. INTRODUCTION

The telecommunication industry is volatile and rapidly growing, in terms of the market dynamicity and competition. In return, it creates new technologies and products, which open a series of options and offers to customers in order to satisfy their needs and requirements [1, 2]. However, one crucial problem that commercial companies in general and telecommunication companies in particular suffer from is a loss of valuable customers to competitors; this is called customer-churn prediction. A customer who leaves a carrier in favor of competitor costs a carrier more than if it gained a new customer [1].

Therefore, "customer-churn prediction" can be seen as one of the most imperative problems that the telecommunication companies face in general. To tackle this problem one needs to understand the behavior of customers, and classify the churn and non-churn customers, so that the necessary decisions will be taken before the churn customers switch to a competitor. More precisely, the goal is to build up an adaptive and dynamic data-mining model in order to efficiently understand the system behavior and allow time to make the right decisions. This will also replace deficiencies of previous work and existing techniques, which are very expensive and time consuming, this problem is studied in the field of telephony with different techniques such as Hidden Markov Model [3], Gaussian and mixture and Bayesian networks [4], association rules [5] decision trees and neural networks [1].

In the last two decades, machine learning techniques [6] have been widely used in many different scientific fields.

Artificial Neural Network [7] is a very popular type of machine learning and it can be considered as another model that is based on modern mathematical concepts. Artificial neural computations are designed to carry out tasks such as pattern recognition, prediction and classification. The performance of this type of machine learning depends on the learning algorithm and the given application, the accuracy of the modeling and structure of each model. The most popular type of learning algorithm for the feed forward neural network is the back propagation algorithm.

The reason for the selection of the feed forward neural network with back propagation learning algorithm is mainly because the network is faster than some other types of network, such as a recurrent neural network. This network has a context layer which copies and stores the hidden neuron activations that can be fed along with the inputs back to the hidden neurons in an iterative manner [8]. On the one hand the context layer (memory) will add more accuracy to the network, than feed forward neural network. On the other hand the network will need more time to learn when it is fed with large training data sets and enormous input features. The feed forward neural network is used as a trade off technique to solve the customer churn and non churn prediction problem.

In the next section the architecture of the artificial neural networks is explained, and then the back propagation algorithm is outlined. Dynamic training is then introduced, after that simulation and results are presented, and finally the main points are concluded.

2. METHODS: NEURAL NETWORK ARCHITECTURE

A standard feed forward ANN architecture is used in this paper. This is a fully connected feed forward neural network also called Multi Layer Perceptron (MLP). The network has three layers input, hidden, and output as shown in Fig 1.

For supervised learning networks, there are several learning techniques that are widely used by researchers. The main three are: real time, back propagation, and back propagation through time, back propagation being what is used here [8, 9, 10] in this paper depending on the application.

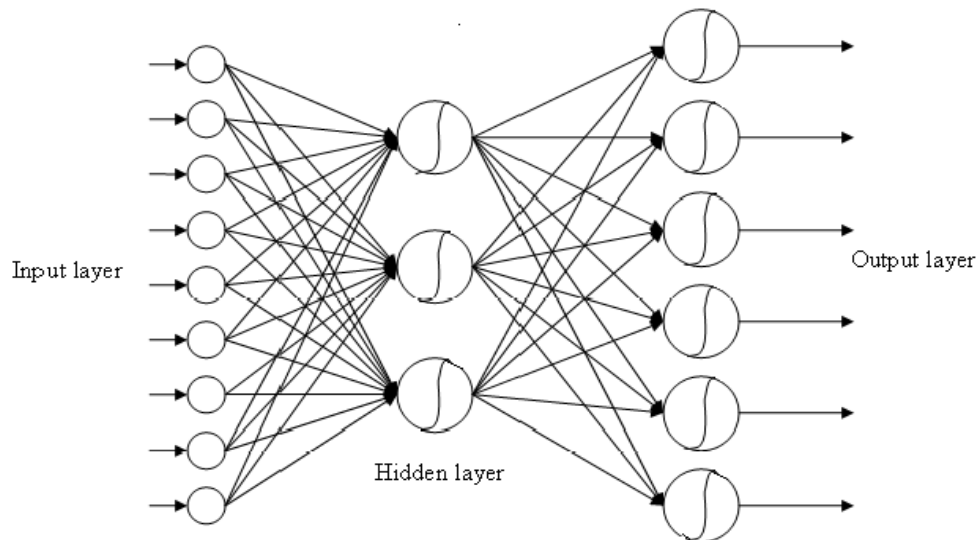


FIGURE 1: Standard feed forward neural network.

3. LEARNING ALGORITHM: BACK PROPAGATION

The back propagation (BP) algorithm is an example of supervised learning [9, 10]. It is based on the minimization of error by gradient descent. A new network is trained with BP. When a target output pattern exists, the actual output pattern is computed. The gradient descent acts to adjust each weight in the layers to reduce the error between the target and actual output patterns. The adjustment of the weights is collected for all patterns and finally the weights are updated.

The sigmoid function is used to compute the output neurons as in equation (1).

$$f(x) = \frac{1}{(1+e^{-x})} \quad (1)$$

Where x represents the net input, the derivative of activation function is

$$f'(x) = (1 - f(x))f(x) \quad (2)$$

The back propagation pass will find the difference between the target and actual output in the output layer

$$e_k = d_k - O_k \quad (3)$$

Where d_k and O_k are the desired and actual outputs for neuron k .

Backpropagation learning defines the sum of error

$$E = \sum_{\text{example}=1}^P \frac{1}{2} \sum_{k \in \text{outputs}} (d_k - O_k)^2 \quad (4)$$

For the output layer, the local gradient is calculated as follows:

$$\delta_k = e_k O_k (1 - O_k) \quad (5)$$

For the hidden layer, the local gradient is calculated as follows:

$$\delta_j = O_j(1 - O_j) \sum_k \delta_k w_{kj} \quad (6)$$

The network learning algorithm adjusts the weights by using delta rule [9, 10], by calculating the local gradients.

$$\Delta w_{jk}^{new} = \mu \Delta w_{jk}^{old} + \beta \delta O_k \quad (7)$$

$$w_{jk}^{new} = w_{jk}^{old} + \Delta w_{jk}^{new} \quad (8)$$

When new is the current iteration and old is the previous iteration. β is learning rate (0.009-0.9999), μ is momentum constant.

4. DYNAMIC TRAINING

Dynamic training is introduced and used to improve the prediction performance of the classifiers. This technique is based on two ANN network configurations. The first network is large and uses the whole training set. After the training is done, a random portion from the training set is taken as a testing set and presented to the network. The forecasting results of that portion are reorganized and used as input patterns with their original targets from the trainings and then used to train the second network; a smaller network. The termination of the learning phases is based on the specified threshold error. Then for testing, the data of the required predicted data is presented to the smaller network. The results of the larger network are reorganized in two inputs to be presented to the smaller network. Bear in mind that the larger network structure will have 124-40-2 (124 input neurons, 40 neurons hidden neurons, 2 output neurons). The smaller network configuration consists of 2-6-2.

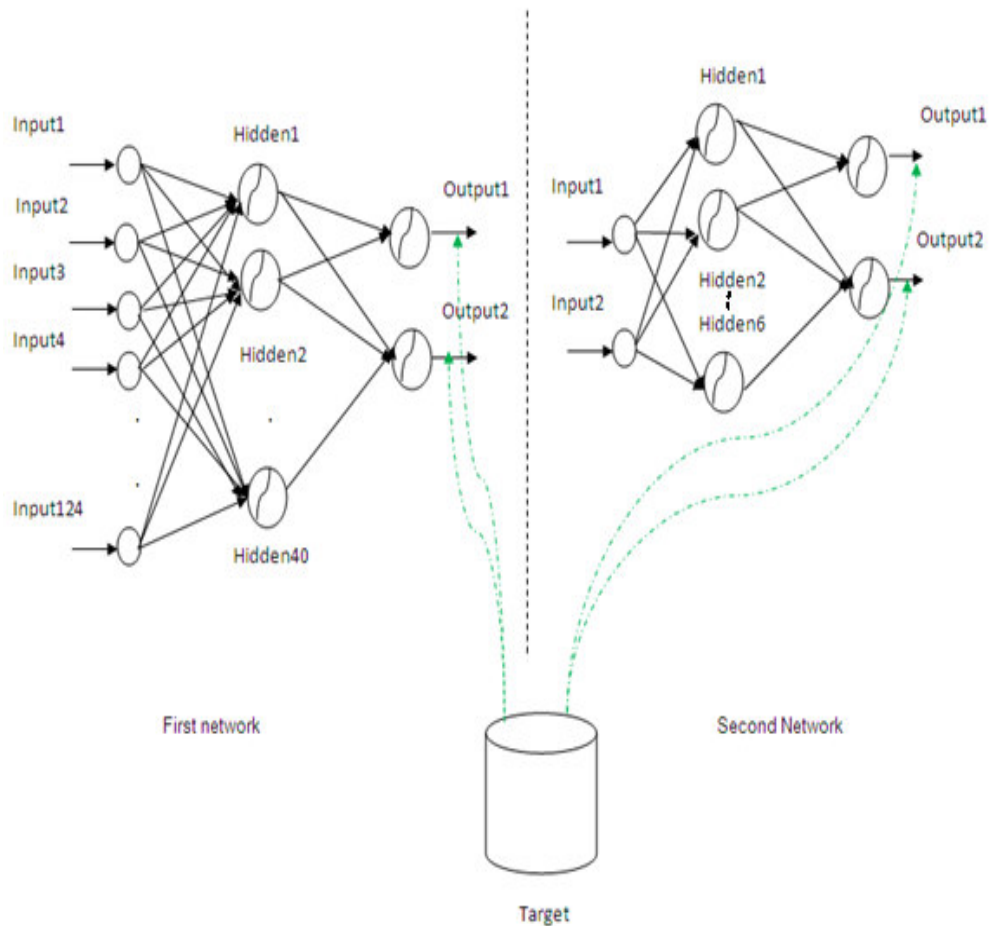


FIGURE 2: Figure displays dynamic training.

5. IMPLEMENTATION AND RESULTS

The prediction system can be processed as follows: obtain and analyze the historical data; pre-processing and normalizing information; choosing the training and testing set; choosing the type of network and its parameters; choosing a suitable learning algorithm; and finally implementation.

The prediction task mainly depends on the training and testing data sets. The size of data selected was 13,000 samples out of a total 1,500,000 customers as neural networks have the ability to learn and generalize.

The training and testing data sets were selected to perform the historical data. Given the nature of our generic selection for the training set, our system is in fact able to predict any random 1000 customers that are not trained and seen by the network (see Table 1).

Population: Number of customers	Size of the samples	Training set	Test set
1,500,000	13000	4000 customers	1000 customers

TABLE 1: Table displays the size of the historical data, training and testing patterns.

There are a lot of important features that have been taken into consideration. These features are related to the customers of telecommunication in the historical data. The main features are the customer's contact data and details, customer behaviors and calls, customer's request for services, etc. The number of input features to the network was 124. And the number of the output features is 2. The input features are scaled down, normalized and transformed. The transformation involves manipulating the data input to create a single input to a neural network, whereas the normalization is a conversion performed on a single data input to scale the data into a suitable range for the neural network. Therefore there is no need to use binary code for the input data. Furthermore there isn't a strong trend in the data. All input data features are linearly scaled and within the range of all variables which are chosen (between 0 and 1).

The number of output features is 2. The output pattern is organized in binary code as 0 1 which represents churn customer and 1 0 represents non-churn customers (see Table 2).

Churn customer		Non-churn customer	
0	1	1	0

TABLE 2: displays output feature.

Two different network structures were used with different parameters for both feed forward networks. A generic model was selected to include all the data. The first network structure was consisted of 124-40-2 (124 input neurons, 40 hidden neurons, and 2 outputs), whereas the second network structure has two networks, as explained in section 5; the large network was 124-40-2 and the small was 2-6-2. The hidden layer neurons were selected based on trial and error and in tandem with each structure with each network (40 hidden neurons for the larger structure and 6 hidden neurons for the smaller network structure). Each network structure used relatively different network parameters. These parameters relied heavily on the size of training and testing sets. Learning rates and momentum were varied. The training cycles were also varied. The type of activation function was a logistic function for the hidden layer and linear for output layer. For the ANN structures patterns of training data were trained and presented to the network in cycles. After every cycle, the weight connection was modified and updated automatically. The processes were iterative. It is important to mention that a specific value of

tolerance should be declared to stop training. This threshold was chosen so that it ensured the model fitted to the training data, and it also did not guarantee good out-of-sample performance. The results with the first network with dynamic training technique was better than the standard technique, the matrix confusion and matrix rate [11] for both networks were shown. Table 3 displays classification for the predicted values for both churn and non churn classes against the actual target of the testing set. It also shows the matrix rate for prediction values for both churn and non-churn values against the actual values. Likewise Table 4 shows results for the standard network structure. As can be seen from Tables 3 and 4, clear misclassifications, in other words, 13 samples of churn class were misclassified and categorized as non-churn samples by the network as seen in Table 3. The likewise with Table 4, 16 samples of churn class were misclassified and categorized as non-churn class. We believe the reason behind this type of misclassification is the misrepresentation of our training and testing data; in other words, the imbalance of data sets caused this problem [11, 12, 13, 14]: as we have in our training set, the number of non-churn class is 3782, and churn class is only 218, and in the testing data set, the number of sample of non-churns 63, and the number of non-churn class is 937. The difference in the results as shown in Table 3 and 4 is small enough to be not essential. Nevertheless, these results for our relatively large sample of data are statistically significant.

matrix confusion		
	Actual	
Predicted	Churn	Non-Churn
Churn	50	13
Non-Churn	0.0	937
matrix rate		
	Actual	
Predicted	Non-churn	Churn
non-Churn	0.7936	0.2063
Churn	0.0	1.0

TABLE 3: Displays matrix confusion and matrix rate for the standard network with dynamic training.

matrix confusion		
	Actual	
Predicted	Churn	Non-Churn
Churn	47.0	16.0
Non-Churn	0.0	937.0

matrix rate		
	Actual	
Predicted	Non-churn	Churn
non-Churn	0.7460	0.2539
Churn	0.0	1.0

TABLE 4: Displays matrix confusion and matrix rate for the standard neural network.

6 CONCLUSION

This paper deals with the problem of classification of churn and non-churn customers in telecommunication companies. Telecommunication is very important as it provides various services of electronic systems to transmit messages through telecommunication devices. However, one crucial problem that commercial companies in general and telecommunication in particular suffer from is a loss of valuable customers to competitors; this is called customer-churn prediction. Machine learning techniques have been widely applied to solve various problems. These machines have been showing great results in many applications. Artificial neural network with the back propagation learning algorithm is used [7, 9, 10, 15, 16, 17]. Variant structures of neural network are discussed. The dynamic training technique is also introduced in this paper. It is used to improve the performance of prediction of the two classes, namely churn and non-churn customers. This technique is based on two ANN network configurations to minimize the total error of the network to predict two different types of customers. The artificial neural network with dynamic training performed better than just an artificial neural network alone. The difference in the results as shown in Table 3 and 4 is small enough to be not essential. However, these results for our relatively large sample of data are statistically significant.

Software in Java language is implemented and used to compute the confusion and rate matrices. The results are presented. As can be seen from our results, both networks showed clear misclassifications. We believe the reason behind this type of misclassification is the misrepresentation of our training and testing data. In other words, the imbalanced training and testing data sets caused this problem. Therefore, further research work should be carried out in order to tackle the misrepresentation of the historical data and to improve the dynamic training technique.

7. REFERENCES

1. Mozer M.C., Dodier R., Colagrosso M.D., Guerra-Salcedo C., "Wolniewicz R., *Prodding the ROC Curve: Constrained Optimization of Classifier Performance Advances*" in Neural Information Processing Systems 14, MIT Press, 2002.
2. Cedric Archaux, H. Laanya, A. Martin and A. Khenchaf. "An SVM based Churn Detector in Prepaid Mobile Telephony", In IEEE. 2004.
3. Hollmen J., "User Profiling and Classification for Fraud Detection". PhD Theses doctorate, University of Helsinki, 2000.
4. Taniguchi M., Haft M., Hollmen J., Tresp V. "Fraud detection in communications networks using neural and probabilistic methods", ICCASP, Vol2, 1998, pp. 1241-1244.

5. Rosset S., Murad U., Neumann E., Idan Y., Pinkas G., "*Discovery of fraud rules for telecommunications-challenges and solutions*", Proceedings ACM SIGKDD, 1999
6. H. Van Khuu, H.-KieLee, and J.-Liang Tsai. "*Machine learning with neural networks and support vector machines*", 2005.
7. K. Anil and J. Mao. "*Artificial neural networks: A tutorial*". IEEE ComputerSociety, 29 (3), 1996, 31 - 44.
8. T. Rashid and M-T.Kechadi, "*Effective Neural Network Approach for Energy Load Forecasting*". International Conference on Computational Intelligence, Calgary, Canada, 2005.
9. P. J. Werbos. "*Backpropagation through time: What it does and how to do it*". In Proceedings of the IEEE, volume 78, 1990, pp. 1550–1560.
10. M. Boden. "*A guide to recurrent neural networks and back propagation*". The DALLAS project. Report from the NUTEK-supported project AIS-8, SICS. Holst: Application of data analysis with learning systems, 2001.
11. M . Hay, "The derivation of global estimates from a confusion matrix", International Journal of Remote Sensing, 1366-5901, Volume 9, Issue 8, 1988, pp. 1395 - 1398.
12. Zhi-Hua Zhou and Xu-Ying Liu, "*On Multi-Class-Cost-Sensitive Learning*", The American Association for Artificial Intelligence. 2006.
13. L. Breiman, J. H. Friedman (1998), R. A. Olshen and C. J. Stone, "*Classification and Recognition Trees*", Wadsworth International Group, 1998, Belmont, CA.
14. U. Knoll, G. Nakhaeilzadeh, and B. Tausend, (1994), "*Cost-sensitive pruning of decision trees*", in Pro, ECML 1994.
15. Shanthi Dhanushkodi, G.Sahoo , Saravanan Nallaperumal "Designing an Artificial Neural Network Model for the Prediction of Thrombo-embolic Stroke" International Journal of Biometrics and Bioinformatics (IJBB), Volume 3, Issue 1, pp: 10-18, 2009.
16. Chien-Wen Cho, Wen-Hung Chao, You-Yin Chen "A linear-discriminant-analysis-based approach to enhance the performance of fuzzy c-means clustering in spike sorting with low-SNR data" International Journal of Biometrics and Bioinformatics (IJBB) Volume 1, Issue 1, pp 1-13, 2007.
17. Aloysius George "Multi-Modal Biometrics Human Verification using LDA and DFB" International Journal of Biometrics and Bioinformatics (IJBB) Volume 2, Issue 4, pp :1-10, 2008.