ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 55-60

Neural versus NeuroFuzzy Systems for Credit Approval

Selwyn Piramuthu

Decision and Information Sciences
University of Florida
Gainesville, Florida 32611

Abstract. Credit approval decisions are important for the financial institutions involved due to the high level of risk associated with wrong decisions. The process of making credit approval decision is complex and unstructured. Neural networks are known to perform reasonably well compared to alternative methods for this problem. However, a problem with using neural networks for credit approval decision is that once a decision is made, it is extremely difficult to explain the rationale behind that decision. Researchers have developed methods using neural network to extract rules, which are then used to explain the *reasoning* behind a given neural network output. These rules do not capture the learned knowledge well enough. Neurofuzzy systems have been recently developed utilizing the beneficial properties of both fuzzy systems as well as neural networks. These neurofuzzy systems can be used to develop fuzzy rules *naturally*. In this study, we analyze the beneficial aspects of using both neurofuzzy systems as well as neural networks for credit approval decisions.

1. Introduction

Credit plays an important role in the lives of many people and in almost all industries that involve monetary investment in some form (Koch, 1995). Obtaining credit is inevitable for smooth and effective operation of industries. The value of credit depends on the need and urgency in obtaining the required credit. It is especially critical when the current worth of that credit is multifold when compared to what it would be worth in the future. In other words, if that credit is not obtained immediately, its worth decreases precipitously due to the loss of opportunities for which the credit was required in the first place. This is more pronounced in industries that operate in areas where there is a tremendous growth in technology and where, loosely speaking, today's technology would be outdated tomorrow. Credit is also essential for acquisition of capital-intensive investments that would be hard to obtain otherwise.

Credit approval decisions are inherently complex due to the various forms of risks involved. The numerous and varied risks in approving credit stem from

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 55-60

the many factors that can lead to the nonpayment of obligation when they come due (Reed et al., 1980; Haslem, 1985). Due to these complexities, the chances of making an error in credit approval decision is large even if done carefully. Given a choice, the party approving the credit is better off erring on the safe side resulting in approval of credit for most deserving clients and none of the risky ones, while possibly denying credit for some deserving clients. This is especially evident when the stakes involved in a wrong decision are high.

Although there is no fail-proof method currently available for credit approval, decisions involving approval of credit are done quite often that it requires extremely careful scrutiny. The fact that the risks associated with clients are inherently different depending on a multitude of factors, past experience can only be used as a guideline for future decisions. The payoff is usually high when a right credit approval decision is made. Any improvement in the methods that are currently being used would result in significant payoffs for the party involved in the credit approval decision.

We study the performance of neural networks and neurofuzzy systems, two recently developed methods under the rubric of artificial intelligence, for credit approval. Neural networks have been in existence for almost three decades. There has been an increased interest in using neural networks for various purposes over the last decade, resulting in a spate of applications in the financial domain. Credit approval certainly has its share in these recently developed methods. In fact, neural networks and related methods are being used quite commonly in credit evaluation decisions. On the other hand, neurofuzzy systems have been in existence only for a few years although both neural networks and fuzzy systems have been around for a few decades already. Neurofuzzy systems are still being studied for plausible applications. Both neural networks and neurofuzzy systems have their inherent advantages. In this paper, we study their performance using a credit approval data set.

This paper is organized as follows: Experimental results using both neural networks and neurofuzzy systems are given in section 2, and section 3 concludes the paper with a brief discussion of preliminary lessons learned in this study and future extentions of this study.

2. Experimental Results

We use the credit approval data that was used in Quinlan (1987) in this study. The data set was *cleaned* to remove examples with missing attribute values. This data is from a large bank. Each of the examples in this data corresponds to a credit card application, with 9 discrete and 6 real attributes. The discrete attributes have anywhere from 2 through 14 possible values. This is a binary classification data, corresponding to positive and negative decisions. There were 690 examples in this data set. We removed the incomplete examples and ended up with 653 examples of which 296 belong to positive class and 357 belong to negative class, where the classes correspond to whether or not credit was approved. This data set is also known to be noisy.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 55-60

Since we randomize the weights in both the neural network as well as the neurofuzzy system, we ran both these systems ten times and took the average and standard deviation of these values for further analyses. To study the generalizability property of both neural networks and neurofuzzy systems, we split the data into a training and a testing (holdout data) set. The training set consisted of 490 examples, of which 268 and 222 examples corresponded to positive and negative class values respectively. The testing set consisted of 163 examples, of which 89 and 74 examples corresponded to positive and negative class values respectively. The 10 training and testing samples used in the 10 runs of both neural networks and neurofuzzy systems were all randomly selected from the whole data set.

For the neural network the learning rates were set at 0.19, since we found this to be a good number from past experience. These networks were allowed to run until the total sum of squares of error terms (tss) reached 0.04, or until the number of iterations (epochs) reached 2000, or if minimum error is not decremented for more than 10 epochs. None of these neural networks converged, and were run until 2000 epochs were reached. The number of input units were set at 15, corresponding to the number of input attributes. The number of output unit was set at 1, corresponding to the binary output class values. The number of hidden units in a hidden layer were set at 8. We chose this as the average of input and output units, since there is no available method to analytically and/or experimentally determine the optimal number of hidden units.

In the neurofuzzy system, the learning rates were set at 0.015, again from past experience in dealing with neurofuzzy systems. These systems were allowed to run until the total sum of squares of error terms (tss) reached 0.04, or the number of iterations (epochs) reached 2000, or if minimum error is not decremented for more than 10 epochs. Weighted sum was used as the aggregate function, for the units in the output layer in the neurofuzzy system. All these neurofuzzy systems converged before 50 epochs, based on non-decrement of minimum error for more than 10 epochs. So, we set the stopping criterion at 50 epochs. These networks had 15 input units corresponding to the 15 input attributes and an output unit corresponding to the binary output class values. By trial and error, we set the number of hidden units to be 250, based on the maximum initial number of rules that were created in these systems. Although there could have been more than 250 rules that were created at any given epoch as learning proceeded, we set the maximum at 250. Hence, only the 250 best rules were used at any given epoch.

The results using credit approval data in neural networks and neurofuzzy systems described above are given in Tables 1 and 2 respectively.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 55-60

#	net configuration	epochs	time (secs.)	classification training (%)	classification testing (%)
1	15-8-1	2000	2642	95.9183	97.546
2	15-8-1	2000	2754	95.3061	87.116
3	15-8-1	2000	2576	96.3265	79.754
4	15-8-1	2000	3135	95.7142	93.251
5	15-8-1	2000	2648	95.5102	80.981
6	15-8-1	2000	2926	94.6938	85.889
7	15-8-1	2000	2886	95.7142	79.141
8	15-8-1	2000	2994	94.6938	76.687
9	15-8-1	2000	2880	96.1224	82.822
10	15-8-1	2000	2424	95.9183	72.392
average			2786 (204.80)	95.59 (0.53)	83.56 (7.22)

Table 1: Results using Neural Networks for Credit Approval

The net configuration entry in Tables 1 and 2 correspond to the network architecture. Here, 15-8-1 represents 15 units in the input layer, 8 units in a hidden layer and 1 output unit. The values for time in the table corresponds to the real time taken for the neural networks to run 2000 epochs. The classification accuracies for both the training as well as the testing (holdout) examples are given in Tables 1 and 2 as percentage correctly classified. The last line in these tables have average as well as standard deviation[1] values from 10 runs of neural network and neurofuzzy system.

As can be seen from Table 1, the classification results using training examples were better than those using testing examples. This is expected since the data is noisy, and it is hard for any method to learn to generalize noisy examples. The classification results using the training examples were more or less consistent, based on the standard deviation values for the training examples being small.

Table 2 provides the results using neurofuzzy system using the same data as in Table 1. Each of the rows in both Tables 1 and 2 correspond to the same set of training and testing example sets respectively. The time taken by the neurofuzzy system was less than that taken by the neural network. It should be noted that neural networks were run using a 66MHz IBM-compatible PC whereas the neural networks were run in a SUN-4 machine. In spite of the difference in machines, the neurofuzzy systems were faster using a slower machine. This could be because of the fewer number of epochs, although the neurofuzzy systems were slower on a per epoch basis due to the large number of hidden units. If we use connection updates[2] as a measure to compare the methods, we still would have the neurofuzzy system performing in lesser number of connection updates compared to the neural networks. The neurofuzzy systems

[1] The standard deviation values are given in parentheses in Table 1.

[2] Connection Updates are measured by the product of the number of connections in the network times the number of epochs.

converged after fewer epochs whereas the neural networks did not converge even after 2000 epochs.

#	net configuration	epochs	time (secs.)	classification training (%)	classification testing (%)
1	15-250-1	50	2358	90.6122	77.914
2	15-250-1	50	2393	89.7959	87.730
3	15-250-1	50	2414	92.4489	74.846
4	15-250-1	50	2306	92.8571	74.233
5	15-250-1	50	2406	91.6326	76.073
6	15-250-1	50	2393	89.7959	87.730
7	15-250-1	50	2415	92.2448	77.300
8	15-250-1	50	2247	92.8571	74.233
9	15-250-1	50	2416	91.8367	76.073
10	15-250-1	50	2372	93.2653	73.006
average			2372 (52.69)	91.74 (1.20)	77.91 (5.10)

Table 2: Results using Neurofuzzy Systems for Credit Approval

In terms of classification accuracies, both the training as well as the testing classification performance of the neurofuzzy systems were worse than those using the neural networks. This could be attributed to the approximations of both the inputs as well as the output made by the neurofuzzy system, as it fuzzified the inputs and defuzzified the outputs. The neural networks did not use any such approximations. The classification accuracies using neurofuzzy system is also influenced by the overlap in the way the range of values of a given attribute is split into its various categories (e.g., range of values for small, medium, and large). Again, these are pitfalls associated with the mechanisms used for both fuzzification and defuzzification of input and outpt data respectively. Currently, there is no optimal means to fuzzify/defuzzify data. These problems could be alleviated as progress is made in this area.

3. Discussion

We studied the performance of neural networks and neurofuzzy systems using credit approval data. Although the results provided in this paper are only preliminary, we believe that this is a good first step in understanding the dynamics of using neural networks versus neurofuzzy systems for credit approval decisions.

This study illustrates the classic tradeoff between classification performance results and understandability of results obtained. The learning results obtained using neurofuzzy systems are understandable by any user since they are in IF-THEN rule form. Any decision made by these neurofuzzy systems can be analyzed using these rules. Thus these systems have an in-built reasoning mechanism. Whereas, in a neural network, all an user can do is to take the output given by the neural network as the most appropriate output without any explicit reasoning. While evaluating credit of a customer, in real-world situations, the credit risk evaluator needs to specify why a certain credit approval/denial decision was made. Under these circumstances, it is hard to use a neural network for this purpose.

Neural networks performed better than neurofuzzy systems in terms of classification accuracy, for both training as well as testing data. This result is not surprising given the various approximations that are made while dealing with fuzzification/defuzzification and also the approximations that are made in fuzzy arithmetic, while learning the rules in the neurofuzzy system.

Hence, from this study, we understand that neural networks are probably better for credit approval decisions only if we are not interested in how it came to a particular conclusion. On the other hand, although the classification performance of neurofuzzy systems are not good at the moment, the benefits associated with generating a set of IF-THEN rules need to be evaluated in light of the fact that the user probably needs to know the line of reasoning behind a decision. Since neurofuzzy systems have been in existence only for the past few years, the various algorithms that are used for fuzzification/defuzzification as well as fuzzy arithmetic itself should improve over time. Neurofuzzy systems will become attractive to users involved in credit approval decisions when better neurofuzzy systems are developed. On the other hand, neural networks are bound to be attractive when methods of extracting rules from weights, accurately, are developed.

References

[1] Haslem, J. A. *Commercial Bank Management*, Reston: Virginia, 1985.

[2] Koch, T. W. *Bank Management*, 3rd edition, Dryden, 1995.

[3] Quinlan, J. R. Simplifying Decision Trees. *International Journal of Man-Machine Studies*, 27, pp. 221-234, 1987.

[4] Reed, E. W., R. V. Cotter, E. K. Gill, and R. K. Smith. *Commercial Banking*, 2nd edition, Prentice-Hall:N.J., 1980.

[5] Rondeau, L., R. Ruelas, E. Levrat, and M. Lamotte. Fuzzy Rules Learning Method. *Proceedings of 3rd European Congress on Intelligent Techniques and Soft Computing*, pp. 515-519, 1995.