Application of Fuzzy Neural Networks on Financial Problems

Martin Rast (MARTIN.RAST@SUNMAIL.LRZ-MUENCHEN.DE)
Ludwig-Maximilians-Universität, Institut für Mathematik,
Theresienstrasse 39/334, 80333 München, Germany

Today neural networks are used in a wide area of different applications. Before using the networks for real production or forecasting, they have to run through a validation process. This is classically done with backtesting or even sensitivity analysis. Nevertheless neural networks often seem to be just black boxes which do not allow an inside view, because one knows that neural networks do well in most cases but not how they *decide*.

The first impact using fuzzy neural networks came from looking for a solution to this dilemma, i.e. finding a way to analyze a trained neural network. Another advantage of them is that the user (or an human expert) can add some knowledge about the structure of the problem, which could speed up training or improve training on a pattern set from which these rules can not be easily extracted.

Fuzzy Neural Networks

Fuzzy Neural Networks combine Neural Networks with the ability of Fuzzy Logic to decribe a rule of reasoning in a *fuzzy* way. The idea is to model a rule set which is often given by human experts in a neural network. The usage of fuzzy rules instead of classical ones is quite straightforward in neural networks using well-known activation functions.

Converting Rules to Neural Topology

A problem to be solved by neural networks is normally given as pattern set. An expert can describe the functionality to be found as a (fuzzy) rule set consisting of rules like

```
if x is small then y is small
```

or similar using as well `big`, `near`, `very` etc. A combination is given with `and`, `or` and `not` and similar.

These rules lead to an equivalent topology of the network. The right side of all the rules is grouped by output variables and connected to an additional layer which is supposed to form the output layer of the fuzzy neural network.

Training of Fuzzy Neural Networks

The training of the network is quite straightforward. In a first approach the network can be trained using standard algorithms. But there are some implicit restrictions which a user might like to implement: first some meaning should be added to the rules if they specify sizes, i.e. every `small` or `big` etc. is expressed in the neural topology by an bias, and the bias for `small` should be smaller than that for `big`. This leads to restrictions on biases.

A second impact is that only a positive weight for the `then` implementes the right sense of the rule. A negative weight would implement an additional `not`. So one might to restrict some weights to be positive.

Nevertheless it might be interesting to train the network without restrictions to see if the training process leads to weights according to the implicit restrictions given above. If not it could be a hint that the rules do not apply to the pattern set.

Financial Applications

For the two major financial applications it is interesting to use fuzzy neural networks. The reasons outlined at the beginning apply to both of them.

Financial Forcasting

The forecasting of financial time series is used in many banks for trading and hedging purposes. In both cases, using technical indicators as well as fundamental, there is often a basic idea of a functionality in the pattern sets. This functionality can be retrieved by asking traders and economic analysts, respectively.

This can be transformed into a neural topology as described before. The trained network can be analyzed how it works. Networks resulting from this approach are more likely to be used in real trading because it is possible to see how they work, and which rules they use.

Customer Credit Rating

The same arguments hold in this case as well. Additionally using the credit customer database of a bank as pattern set leads to a inherent problem: Normally it does not contain data of customers which are obviously not relyable. Missing this subset of the customer database, a neural network cannot learn the implicit rules hidden in a such pattern set. But the rules for obviously bad debitors are often easy to formulate. Combining these rules with a standard network and using the output of both subnets improves the ability of neural networks for predicting bankruptcy.