

Forecasting Stock Market Performance Using Hybrid Intelligent System

Xiaodan Wu¹, Ming Fung², and Andrew Flitman³

¹ jenny.wu@infotech.monash.edu.au

² mkfung@yahoo.com

³ andrew.flitman@infotech.monash.edu.au

School of Business Systems, Faculty of Information Technology,
Monash University, Victoria 3800 Australia

Abstract: Predicting the future has always been one of mankind's desires. In recent years, artificial intelligent techniques such as Neural Networks, Fuzzy Logic, and Genetic Algorithms have gained popularity for this kind of applications. Much research effort has been made to improve the prediction accuracy and computational efficiency. In this paper, a hybridized neural networks and fuzzy logic system, namely the *FeedForward NeuroFuzzy* (FFNF) model, is proposed to tackle a financial forecasting problem. It is found that, by breaking down a large problem into manageable "chunks", the proposed FFNF model yields better performance in terms of computational efficiency, prediction accuracy and generalization ability. It also overcomes the black art approach in conventional NNs by incorporating "transparency" into the system.

1. Introduction

Over the years, economy forecasters have been striving to predict various financial activities, such as exchange rates, interest rates and stock prices. Successful prediction of the "economic cycle" helps them to maximize their gains in the market place and/or to prevent from business failure. Owing to the nonlinear, dynamic and chaotic nature of economic environment, financial problems are usually difficult to analyze and model using conventional techniques. Those techniques rely heavily on assumptions that often mask constraints in the initial problem domain. As the result, the models derived tend to have poor quality in applications. As the alternative or supplementary approaches, artificial intelligent (AI) techniques, such as neural networks (NNs), fuzzy logic (FL) and genetic algorithms (GAs), have drawn serious attention from both industrial and business sectors in recent years. Many encouraging results have been achieved when applying these techniques to financial market [1,3, 5, 6].

Depending on the scenario being modeled, a particular technique or a combination of techniques can be used to achieve certain goals. Because each technique has its own strengths and limitations in terms of performance, reliability, missing functionality and biases, some problems cannot be solved or efficiently solved by using single technique. In these cases, a hybrid approach, which employs a combination of techniques, may provide rich opportunities to successfully implement the problem modeling.

This paper explores the application of AI techniques to forecast the monthly S&P500 closing price. A hybrid *FeedForward NeuroFuzzy* model is proposed. A basic 3-layer backpropagation NN is used as a benchmark to gauge the performance of the hybrid NeuroFuzzy model. The trend prediction accuracy was used to evaluate the performance of the models.

2. Experimental Data Selection and Preprocessing

Data selection, which usually requires knowledge of domain experts, is an important part in building a successful prediction model. Appropriate data is the essential ingredient to start with the modeling process. Obtained from “<http://www.economagic.com/popular.htm>”, the data used in this paper cover periods from January 1992 to April 2000. In light with the experts’ suggestions, the following variables are used as the inputs and output of the forecasting model:

x_1 : Consumer Price Index	x_2 : Total Industrial Production Index
x_3 : Leading Economic Index	x_4 : Bank Prime Loan Rate
x_5 : Federal Funds Rate	x_6 : Unemployment Rate
x_7 : S&P500 lag(1)	x_8 : S&P500 lag(2)
x_9 : S&P500 lag(3)	y : S&P500 closing prices (output)

As stock market prices can move up and down quite frequently, it makes more sense to predict the trend of the financial data series rather than the actual value itself. Since most stocks tend to move in the same direction as the S&P500, it would be desirable to know which direction this indicator heads.

Data preprocessing facilitates efficiency and effectiveness, the computational cost reduction, and better generalization ability. In this paper, S&P500 lags(1-3) were preprocessed to be the previous three months' closing values. These historical data are considered having significant impact on the predicted one.

3. Neural Networks Approaches

Owing to the complex and unstructured characteristics of economic forecasting, neural networks (NNs) may be the appropriate approach to solve the problem under consideration. Compared with conventional statistical methods, neural networks are more robust in terms of coping with noisy and incomplete data. They can interrelate both linear and non-linear data as well as the important influential factors that are not based on historical numerical data. They may also eliminate the constraints of unrealistic assumptions masked by conventional techniques. These make NNs more flexible and effective in the domain of financial forecasting.

In experiment data was split into three sub-sets for model training, test, and validation. This is to ensure that the trained network will not simply memorize the data in the training set but learn the relationships amongst the variables, i.e. having good generalization ability.

There are many issues involved in NNs development. These include choice of input variables, number of hidden neurons and their activation functions, network architecture and parameters, etc. Based on a trial and error approach, different architectures and parameters have been experimented in order to obtain an optimal configuration. The performance of this optimal network is then used to comparing with the proposed FFNF system.

4. A NeuroFuzzy Approach

Fuzzy logic (FL) provides a means of representing uncertainty and is good at reasoning with imprecise data. In a fuzzy system variables are defined in terms of fuzzy sets, and fuzzy rules are specified to tie the fuzzy input(s) to the output fuzzy set. Such rules derived also make it easy to maintain and update the existing knowledge in the problem domain. Bearing these benefits, FL also has limitations. In most applications, knowledge is embedded in the data itself and thus requiring significant effort to derive the accurate rules and the representative membership functions. Further, the rules derived cannot adapt automatically to changes. They have to be manually altered to cater for change in conditions.

NNs technique is commonly referred to as the "black box" approach. The causes of certain NNs' behaviours can neither be interpreted nor can it be changed manually. That is, the final state of the NN model cannot be interpreted explicitly. Contrast to NNs' implicit knowledge acquisition, FL facilitates transparency in terms of explicit rule presentation. From this sense, FL may be used to improve the effectiveness of a NNs model used for trading, for example, by incorporating knowledge about financial markets with clear explanation of how the trading recommendations are derived. It may also make the model verification and optimization easy and efficient [1].

The advantages from FL point of view is that NNs' learning ability can be utilized to adjust and fine-tune the fuzzy membership functions. This solves the difficult and resource-consuming process of deriving the rule base manually. The resultant optimal rule base turns out to be an advantage for NNs as it solves the "black box" problem.

In short, NNs learn from data sets while FL solutions are easy to verify and optimize. The combination of both techniques results in a hybrid NeuroFuzzy model. With the complementary advantages offered by both sides, they work together to deliver value.

There are many ways to integrate NNs and FL. In this paper, NN is adopted as the learning tool to evolve a set of fuzzy rules, based solely on the data itself. The inferred fuzzy rules can be used for forecasting the monthly S&P500 trend. The developed system would have the following characteristics:

- Vague inputs can be used
- More general boundaries can be described
- Automated fuzzy rule extraction
- Learning the rules through training of NNs

4.1. System Connections

The proposed NeuroFuzzy system is a FL system trained by a NN. A simplified structure of the system connections, which involves two inputs with two and three fuzzy sets respectively, is shown in Figure 1.

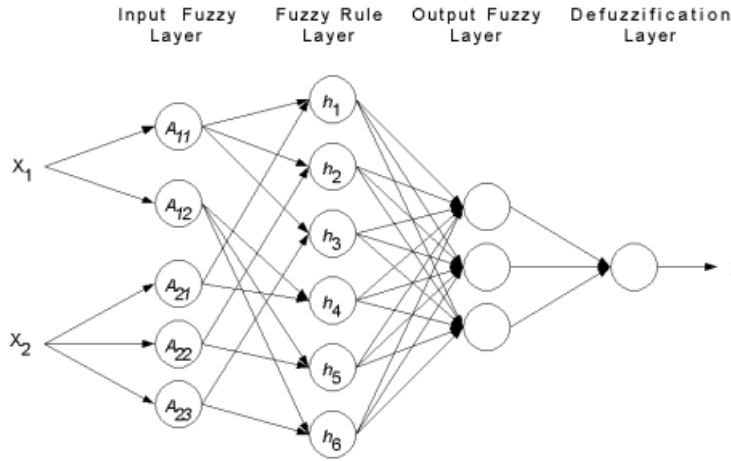


Fig. 1. Connections between different layers in the NeuroFuzzy module

Where:

- The *input fuzzy layer* represents linguistic sets in antecedent fuzzy membership functions. The encoding of the shapes, centres, and the widths of membership functions is done in this layer. The total number of fuzzy sets for all input variables determines the number of neurons as follow, where N_i is the number of fuzzy sets for input x_i , and m is the total number of input variables:

$$\sum_{i=1}^m N_i$$

- The *fuzzy rule layer* is built as fuzzy associative memories (FAM). A FAM is a fuzzy logic rule with an associated weight. Deboeck [2] has demonstrated how FAMs can be used to extract rules. With FAM, each rule is assigned a degree of support representing the individual importance of the rule. Rules themselves can be "fuzzy", meaning validity between 0 and 1. Each neuron in this layer represents a unique combination of input fuzzy sets. The links between this layer and the input fuzzy layer are designed in a way that all combinations among input fuzzy sets are represented as shown in table 1.

Table 1. Fuzzy rule table for x_1 and x_2

	x_2			
x_1		A_{21}	A_{22}	A_{23}
	A_{11}	R_1	R_2	R_3
	A_{12}	R_4	R_5	R_6

The "AND" association for each fuzzy set combination forms the antecedent relation established by these links, which build a relation between the input and output fuzzy membership functions and define the fuzzy rules. Each neuron in the module represents a fuzzy rule and the total number of neurons can be obtained by the product of all N_i s in the input fuzzy layer as:

$$\prod_{i=1}^m N_i$$

- The *output fuzzy layer* represents the consequent fuzzy membership functions for outputs. The number of fuzzy sets for output fuzzy variables determines the number of neurons in this layer. Each neuron in the fuzzy rule layer is connected to all neurons in this layer. The weights between fuzzy rule layer and output fuzzy layer are rule matrices, which can be converted into a FAM matrix.
- The *defuzzification layer* is for rule evaluation. The weight of the output link from the neuron represents the center of maximum of each output membership function of the consequent. The final output value is then calculated using the center of maximum method.

4.2. The FeedForward NeuroFuzzy Model

The system structure identifies the fuzzy logic inference flow from the input to the output variables. Fuzzification in the input interface translates analog inputs into fuzzy values. The fuzzy inference takes place in the rule blocks which contain the linguistic control rules. The outputs of the rule blocks are linguistic variables. Defuzzification in the output interface converts the fuzzy variables back into crisp values. Figure 2 shows the structure of the FFNF system including input interface, rule blocks and output interface, where the connection lines symbolize the data flow.

Based on the nature of the corresponding economic indicators, the original inputs of the FFNF system ($x_1 - x_9$) are categorized into three groups (Table 2) to form three sub-modules of the first tier of the rule blocks (see Figure 2). The second tier of the rule blocks has one processing module (the Final Module in Figure 2) designed to sum the rule evolving results from the first level modules. This layered structure helps to simplify the initial complex problem and to improve system efficiency. The size of the fuzzy rule base developed has been significantly reduced from a total of $3^{10} = 59,049$ rules (10 variables with 3 membership functions each) derived from a non-layered model to 324 rules (4 modules x 3^4 each) of the current structure.

Table 2. Grouping of input, intermediate and output variables

	Group 1 (indices)	Group 2 (rates)	Group 3 (historical data)	Final System
Inputs	x_1	x_4	x_7	y_1
	x_2	x_5	x_8	y_2
	x_3	x_6	x_9	y_3
Outputs	y_1	y_2	y_3	y

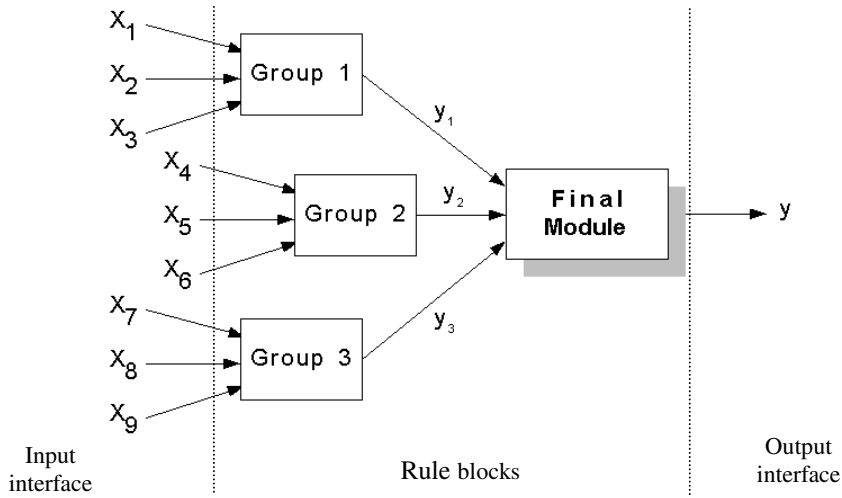


Fig. 2. Structure of the FFNF System.

4.3. NeuroFuzzy Learning Algorithm

Considering a simple case of two input variables (\$x_1, x_2\$) and one output variable (\$y\$) for demonstration purpose, the following mathematics forms the basis of the learning algorithm [4].

$$x_1 : A_{1i} (i \in 1, 2, \dots, r)$$

$$x_2 : A_{2j} (j \in 1, 2, \dots, k)$$

where : \$r\$ means the number of fuzzy partitions for \$x_1, k\$ means for \$x_2\$

Gaussian type membership functions :

$$A_{1i}(x_1) \propto \exp(-((x_1 - a_{1i})^2 / 2\sigma_{1i}^2))$$

$$A_{2j}(x_2) \propto \exp(-((x_2 - a_{2j})^2 / 2\sigma_{2j}^2))$$

Where :

\$a_{1i}\$ and \$\sigma_{1i}\$ are respectively the centre and width of \$A_{1i}\$, assuming that the rule base is defined based on all the above combinations of \$A_{1i}\$ and \$A_{2j}\$ (\$i \in 1, \dots, r, j \in 1, \dots, k\$) [4] :

- Rule 1 : \$A_{11}, A_{21} \propto R_1\$
- ...
- Rule \$k\$: \$A_{11}, A_{2k} \propto R_k\$
- ...
- Rule \$2k\$: \$A_{12}, A_{2k} \propto R_{2k}\$
- ...
- Rule \$(i \in 1)k \in j\$: \$A_{1i}, A_{2j} \propto R_{(i \in 1)k \in j}\$
- ...
- Rule \$rk\$: \$A_{1r}, A_{2k} \propto R_{rk}\$

The above fuzzy inference rules can be expressed in the form of rule table below:

Table 3. Fuzzy rule table for A_{1i} and A_{2j}

$x_2 \backslash x_1$	A_{21}	A_{22}	...	A_{2j}	...	A_{2k}
A_{11}	R_1	R_2	...	R_j	...	R_k
A_{12}	R_{k+1}	R_{k+2}	...	R_{k+j}	...	R_{2k}
...
A_{1i}	$R_{(i-1)k+j}$
...
A_{1r}	$R_{(r-1)k+1}$	R_k

Based on the above, a fuzzy inference y can be obtained using product-sum-gravity fuzzy reasoning method. The intersection (agreement) of A_{1i} and A_{2j} ($i = 1 \dots r; j = 1 \dots k$) at (x_1, x_2) is calculated as:

$$h_{(i-1)(k-j)} = A_{1i}(x_1) A_{2j}(x_2)$$

5. Experiment Results and Discussion

For the purpose of performance comparison of the two models (NNs vs. FFNF), the trend accuracy was calculated as shown below:

$$\text{Correct Trend} = \frac{100}{n} \sum_{i=1}^n \alpha_i; \quad \text{where } \alpha_i = \begin{cases} 1 & \text{if } (x_{t-1} \leq x_t)(\hat{x}_{t-1} \leq x_t) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

where n, x, \hat{x} are the number of observations, the actual and predicted values respectively.

A correct trend of 100% means that the model can perfectly predict the directional changes, while a value of greater than 50% means that the model performs better than tossing a coin. The evaluation was performed based on the proposed FFNF model and the optimal NN configuration in Table 4. The results of the two models are given in Table 5.

For the NN model, the best result (in bold font) was obtained using a 3-layer architecture trained with the standard backpropagation (BP) algorithm. The transfer function was linear over the range [1, -1]. The tanh function was adopted as the activation function in the hidden layer. Both the learning rate and the momentum were at 0.1, and the initial weight was set at 0.3 (see Table 4).

Table 4. The optimal configuration obtained using a standard 3-layer BP NN

Input Variables	Hidden Neurons	Learning Rate	Momentum	Initial Weights	R ²		
					Training	Test	Pattern
6	22	0.6	0.9	0.6	0.9817	0.9836	0.9830
9	13	0.1	0.1	0.3	0.9908	0.9884	0.9905
12	30	0.7	0.8	0.3	0.9775	0.9454	0.9836

There are constraints that may limit the scope of the selection of input variables. Subjective opinions may also have significant influence on how successful the prediction model will be. The contribution factor of a variable can be considered as a rough measure for evaluating the significance of the variable, relative to the other variables in a same network, in predicting the output. In our experiment, the consumer price index (CPI) is the most significant input variable of the prediction model.

Based on initial values, the NeuroFuzzy model tunes rule weights and membership function definitions so that the system converges to the behaviour represented by the data sets. Changes in the weights during learning can be interpreted as changes in the membership functions and fuzzy rule.

Despite its flexibility in system tuning, the model requires substantial effort in optimization so as to arrive at the optimal combination of member functions, fuzzy rules, and fuzzy inference and defuzzification methods. If no rules were generated with a reasonable level of degree of support (DoS) for certain input variables, it is reasonable to assume that they either do not contribute or are correlating to another variable for which rules are generated. Examples of the rule bases evolved for the modules are shown in Appendix.

In the experiment, the original inputs are categorized into three groups to form three sub-modules of the first tier of the rule blocks (see Figure 2 above). The rules evolved from these sub-modules are expressed in a simple, clear-explained format. They would not represent the overall system behavior directly but can, however, provide valuable assistance in understanding the complex initial problem domain.

The second tier of the rule blocks has one processing "Final Module" designed to sum the rule evolving results from the first level modules. The inputs of it are the defuzzified crisp values of the three sub-modules in the previous stage. The rule base evolved from this Final Module cannot be used in a stand-alone manner. Instead, by feeding all the inputs through three rule bases derived in the first tier, the final output in the second tier is meaningful and therefore is applicable for prediction purpose.

At this stage, the system cannot operate in parallel as the results of the first tier modules have to be computed before feeding into the Final Module. However the intermediate outputs of the first tier are valuable and useful. They facilitate decision-making assistance depending on expert knowledge or subjective opinions. The final output when using all four modules as a single rule block provides high prediction accuracy. The layered structure helps simplify the initial complex problem and to improve system efficiency.

As shown in Table 5, the FFNF system is superior to the standard 3-layer BPNN in the financial forecasting problem under consideration. By breaking down a large, complex problem into manageable smaller modules, the FFNF design yields better performance in terms of computational efficiency (i.e. a significant reduction in the rule base), prediction accuracy and good generalization ability.

Table 5. Models performance

	3-Layer BPNN	FFNF
Correct Trend	96%	99%

6. Conclusions

Financial prediction in an unstructured, nonlinear and dynamic environment is a sophisticated task. It is worthy of considering the hybrid artificial intelligence approach. This paper proposes a *Feedforward NeruoFuzzy* (FFNF) system for financial application. The system integrates the fuzzy rule-based techniques and the NNs technique to accurately predict the monthly S&P500 trend. Experimental results showed that the proposed system has better performance than the BPNN model. By highlighting the advantages and overcoming the limitations of individual AI techniques, a hybrid approach can facilitate the development of more reliable and effective intelligent systems to model expert thinking and to support the decision-making processes. It has diverse potential and is a promising direction for both research and application development in the years to come.

References

1. Altrock, C.: Fuzzy Logic and NeuroFuzzy Applications in Business and Finance. Prentice PTR (1997)
2. Deboeck, G.J.: Why Use Fuzzy Modeling. In: Deboeck, G.J. (ed.): Trading on the Edge. John Wiley and Sons, New York (1994) 191-206
3. Goonatilake, S. and Khebbal S.: Intelligent Hybrid Systems: Issues, Classifications and Future Directions. In Goonatilake, S. and Khebbal S.(eds.): Intelligent Hybrid Systems. John Wiley and Sons, U.K. (1995) 1-19
4. Shi, Y., Mizumoto, M., Yubazaki, N. and Otani, M.: A Method of Generating Fuzzy Rules Based on the Neuro-fuzzy Learning Algorithm. Journal of Japan Society for Fuzzy Theory and Systems. Vol. 8, No.4 (1996) 695-705
5. Tano, S.: Fuzzy Logic for Financial Trading. In Goonatilake, S. and Treleaven, P. (eds.): Intelligent Systems for Finance and Business. John Wiley and Sons, Chichester (1995) 209-224
6. Yen, J.: Fuzzy Logic-A Modern Perspective. IEEE Transactions on Knowledge and Data Engineering. Vol. 11, No. 1 (1999) 153-165

Appendix: Examples of the Rule Base Evolved

Table A1. Rules of the rule block "Group 1"

IF			THEN	
x_1	x_2	x_3	DoS (weights)	y_1
low	low	low	1.00	low
medium	low	low	0.86	low
medium	medium	low	0.99	low
medium	high	medium	0.66	high
high	medium	medium	0.64	low
high	high	medium	0.90	high

- Rule 1: IF ($x_1 = \text{low}$ and $x_2 = \text{low}$ and $x_3 = \text{low}$) OR ($x_1 = \text{medium}$ and $x_2 = \text{low}$ and $x_3 = \text{low}$) OR ($x_1 = \text{medium}$ and $x_2 = \text{medium}$ and $x_3 = \text{low}$) OR ($x_1 = \text{high}$ and $x_2 = \text{medium}$ and $x_3 = \text{medium}$), THEN $y = \text{low}$
- Rule 2: IF ($x_1 = \text{medium}$ and $x_2 = \text{high}$ and $x_3 = \text{medium}$) OR ($x_1 = \text{high}$ and $x_2 = \text{high}$ and $x_3 = \text{medium}$), THEN $y = \text{high}$

Table A2. Rules of the rule block "Group 2"

IF			THEN	
x_4	x_5	x_6	DoS (weights)	y_2
low	low	medium	0.98	low
low	low	high	0.88	low
medium	medium	low	0.98	high
medium	medium	medium	0.76	low
high	medium	low	0.59	medium
high	high	low	0.55	high
high	high	medium	0.67	low

- Rule 1: IF ($x_1 = \text{low}$ and $x_2 = \text{low}$ and $x_3 = \text{medium}$) OR ($x_1 = \text{low}$ and $x_2 = \text{low}$ and $x_3 = \text{high}$) OR ($x_1 = \text{medium}$ and $x_2 = \text{medium}$ and $x_3 = \text{medium}$) OR ($x_1 = \text{high}$ and $x_2 = \text{high}$ and $x_3 = \text{medium}$), THEN $y = \text{low}$
- Rule 2: IF ($x_1 = \text{high}$ and $x_2 = \text{medium}$ and $x_3 = \text{low}$), THEN $y = \text{medium}$
- Rule 3: IF ($x_1 = \text{medium}$ and $x_2 = \text{medium}$ and $x_3 = \text{low}$) OR ($x_1 = \text{high}$ and $x_2 = \text{high}$ and $x_3 = \text{low}$), THEN $y = \text{high}$

Table A3. Rules of the rule block "Final Module"

IF			THEN	
y_1	y_2	y_3	DoS (weights)	y
low	low	low	0.65	decrease
low	medium	low	0.72	decrease
medium	medium	low	0.59	decrease
high	medium	medium	0.82	increase
high	high	medium	0.98	increase

- Rule 1: IF ($x_1 = \text{low}$ and $x_2 = \text{low}$ and $x_3 = \text{low}$) OR ($x_1 = \text{low}$ and $x_2 = \text{medium}$ and $x_3 = \text{low}$) OR ($x_1 = \text{medium}$ and $x_2 = \text{medium}$ and $x_3 = \text{low}$), THEN $y = \text{decrease}$
- Rule 2: IF ($x_1 = \text{high}$ and $x_2 = \text{medium}$ and $x_3 = \text{medium}$) OR ($x_1 = \text{high}$ and $x_2 = \text{high}$ and $x_3 = \text{medium}$), THEN $y = \text{increase}$