Capital Market Applications of Neural Networks, Fuzzy Logic and Genetic Algorithms

Arnold F. Shapiro

Penn State University

Smeal College of Business, University Park, PA 16802, USA
Phone: 01-814-865-3961, Fax: 01-814-865-6284, E-mail: afs1@psu.edu

Abstract: The capital markets have numerous areas with potential applications for neural networks, fuzzy logic and genetic algorithms. Given this potential and the impetus on these technologies during the last decade, a number of studies have focused on capital market applications. This paper presents an overview of these studies. The specific purposes of the paper are twofold: first, to review the capital market applications of these technologies so as to document the unique characteristics of capital markets as an application area; and second, to document the extent to which these technologies, and hybrids thereof, have been employed.

Keywords: capital markets, applications, neural networks, fuzzy logic, genetic algorithms

Acknowledgments:

This work was supported in part by the Robert G. Schwartz Faculty Fellowship. The assistance of Asheesh Choudhary, Krishna D. Faldu, Jung Eun Kim, and Laura E. Campbell is gratefully acknowledged

1 Introduction

Neural networks (NNs) are used for learning and curve fitting, fuzzy logic (FL) is used to deal with imprecision and uncertainty, and genetic algorithms (GAs) are used for search and optimization. These technologies often are linked together because they are the most commonly used components of what Zadeh (1992) called soft computing (SC), which he envisioned as being "... modes of computing in which precision is traded for tractability, robustness and ease of implementation."

The capital markets have numerous areas with potential applications for these SC technologies. Given this potential and the impetus on these technologies during the last decade, a number of studies have focused on capital market applications and in many cases have demonstrated better performance than competing approaches. This paper presents an overview of these studies. The specific purposes of the paper are twofold: first, to review the capital market applications of these SC technologies so as to document the unique characteristics of capital markets as an application area; and second, to document the extent to which these technologies, and hybrids thereof, have been employed.

The paper has a separate section devoted to each of the capital market areas of market forecasting, trading rules, option pricing, bond ratings, and portfolio construction. Each section begins with a brief introduction and then SC studies in that application area are reviewed. The studies were drawn from a broad cross-section of the literature and are intended to show where each technology has made inroads into the capital market areas. However, since this paper is still in the development stage, only a representative sample of the literature has been included, so the study should be considered a work in progress. The paper ends with a prognosis for the SC technologies.

2 Neural Networks, Fuzzy Logic and Genetic Algorithms

It is assumed that readers are generally familiar with the basics of NNs, FL and GAs,[1] but they may not have conceptualized the overall processes associated with these technologies. This section presents an overview of these processes.

2.1 Neural Networks (NNs)

NNs, first explored by Rosenblatt (1959) and Widrow and Hoff (1960), are computational structures with learning and generalization capabilities. Conceptually, they employ a distributive technique to store knowledge acquired by learning with known samples and are used for pattern classification, prediction and analysis, and control and optimization. Operationally, they are software programs that emulate the biological structure of the human brain and its associated neural complex (Bishop, 1995).

The NN can be either supervised or unsupervised. The distinguishing feature of a supervised NN is that its input and output are known and its objective is to discover a relationship between the two. The distinguishing feature of an unsupervised NN is that only the input is known and the goal is to uncover patterns in the features of the input data. The remainder of this subsection is devoted to an overview of supervised and unsupervised NNs, as processes.

2.1.1 Supervised NNs

A sketch of the operation of a supervised NN is shown in Figure 1.

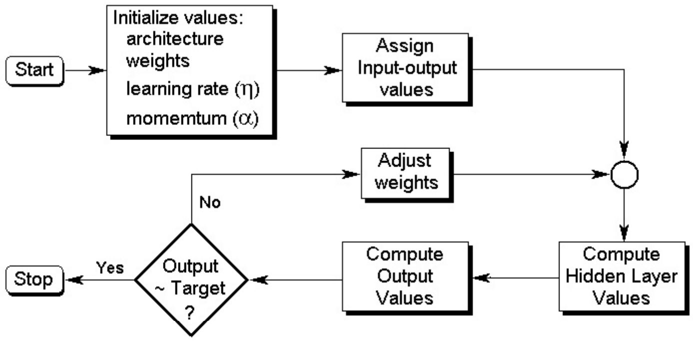

Figure 1: The Operation of a Supervised NN

Since supervised learning is involved, the system will attempt to match the input with a known target, such as stock prices or bond ratings. The process begins by assigning random weights to the connection between each set of neurons in the network. These weights represent the intensity of the connection between any two neurons. Given the weights, the intermediate values (in the hidden layer) and then the output of the system are computed. If the output is optimal, in the sense that it is sufficiently close to the target, the process is halted; if not, the weights are adjusted and the process is continued until an optimal solution is obtained or an alternate stopping rule is reached.

If the flow of information through the network is from the input to the output, it is known as a feed forward network. The NN is said to involve back-propagation if inadequacies in the output are fed back through the network so that the algorithm can be improved. We will refer to this network as a feedforward NN with backpropagation (FFNN with BP).

2.1.2 Unsupervised NNs

This section discusses one of the most common unsupervised NNs, the Kohonen network (Kohonen 1988), which often is referred to as a self-organizing feature map (SOFM). The purpose of the network is to emulate our understanding of how the brain uses spatial mappings to model complex data structures. Specifically, the learning algorithm develops a mapping from the input patterns to the output units that embodies the features of the input patterns.

In contrast to the supervised network, where the neurons are arranged in layers, in the Kohonen network they are arranged in a planar configuration and the inputs are connected to each unit in the network. The configuration is depicted in Figure 2.

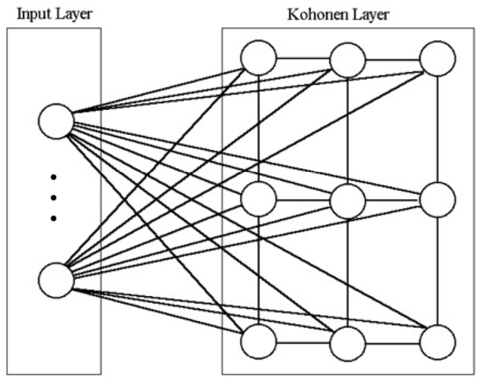

Figure 2: Two Dimensional Kohonen Network

As indicated, the Kohonen SOFM may be represented as a two-layered network consisting of a set of input units in the input layer and a set of output units arranged in a grid called a Kohonen layer. The input and output layers are totally interconnected and there is a weight associated with each link, which is a measure of the intensity of the link.

The sketch of the operation of a SOFM is shown in Figure 3.

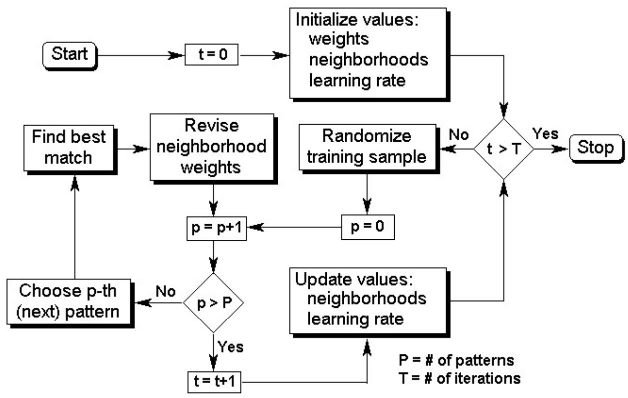

Figure 3: Operation of a Kohonen Network

The first step in the process is to initialize the parameters and organize the data. This entails setting the iteration index, t, to zero, the interconnecting weights to small positive random values, and the learning rate to a value smaller than but close to 1. Each unit has a neighborhood of units associated with it and empirical evidence suggests that the best approach is to have the neighborhoods fairly broad initially and then to have them decrease over time. Similarly, the learning rate is a decreasing function of time.

Each iteration begins by randomizing the training sample, which is composed of P patterns, each of which is represented by a numerical vector. For example, the patterns may be composed of stocks and/or market indexes and the input variables may be daily price and volume data. Until the number of patterns used (p) exceeds the number available ($p > P$), the patterns are presented to the units on the grid, each of which is assigned the Euclidean distance between its connecting weight to the input unit and the value of the input. This distance is given by $[\Sigma_j (x_j - w_{ij})^2]^{0.5}$, where w_{ij} is the connecting weight between the j-th input unit and the i-th unit on the grid and x_j is the input from unit j. The unit that is the best match to the pattern, the winning unit, is used to adjust the weights of the units in its neighborhood. For this reason the SOFM is often referred to as a competitive NN. The process continues until the number of iterations exceeds some predetermined value (T).

In the foregoing training process, the winning units in the Kohonen layer develop clusters of neighbors, which represent the class types found in the training patterns. As a result, patterns associated with each other in the input space will be mapped to output units that also are associated with each other. Since the class of each cluster is known, the network can be used to classify the inputs.

2.2 Fuzzy Logic (FL)

Fuzzy logic[2] (FL), which was formulated by Zadeh (1965), was developed as a response to the fact that most of the parameters we encounter in the real world are not precisely defined. As such, it gives a framework for approximate reasoning and allows qualitative knowledge about a problem to be translated into an executable set of rules. This reasoning and rule-based approach, which is known as a fuzzy inference system, is then used to respond to new inputs.

2.2.1 A Fuzzy Inference System (FIS)

The fuzzy inference system (FIS) is a popular methodology for implementing FL.[3] FISs are also known as fuzzy rule based systems, fuzzy expert systems, fuzzy models, fuzzy associative memories (FAM), or fuzzy logic controllers when used as controllers (Jang et al. 1997 p. 73). The essence of the system can be represented as shown in Figure 4.

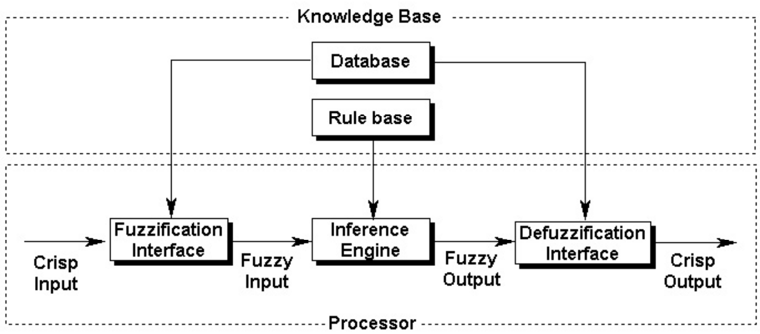

Figure 4: A Fuzzy Inference System (FIS)

As indicated in the figure, the FIS can be envisioned as involving a knowledge base and a processing stage. The knowledge base provides the membership functions (MFs) and fuzzy rules needed for the process. In the processing stage, numerical crisp variables are the input of the system.[4] These variables are passed through a fuzzification stage where they are transformed to linguistic variables, which become the fuzzy input for the inference engine. This fuzzy input is transformed by the rules of the inference engine to fuzzy output. The linguistic results are then changed by a defuzzification stage into numerical values that become the output of the system.

2.3 Genetic Algorithms (GAs)

Genetic algorithms[5] (GAs) were proposed by Holland (1975) as a way to perform a randomized global search in a solution space. In this space, a population of candidate solutions, each with an associated fitness value, is evaluated by a fitness function on the basis of their performance. Then, using genetic operations, the best candidates are used to evolve a new population that not only has more of the good solutions but better solutions as well.

This process, which can be described as an automated, intelligent approach to trial and error, based on principles of natural selection, is depicted in Figure 5.

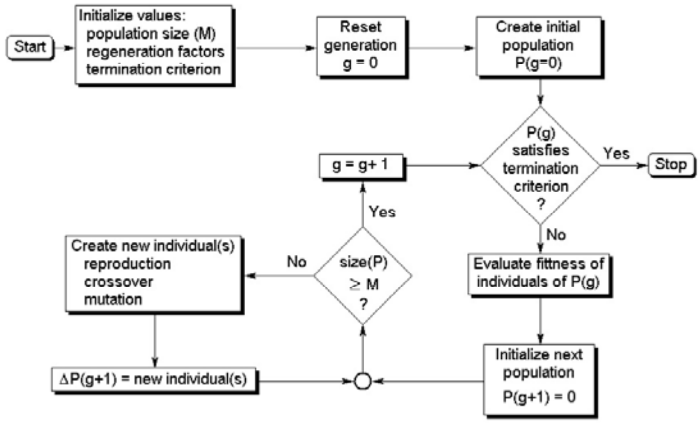

Figure 5: The GA Process

As indicated, the first step in the process is initialization, which involves choosing a population size (M), population regeneration factors, and a termination criterion. The next step is to randomly generate an initial population of solutions, P(g=0), where g is the generation. If this population satisfies the termination criterion, the process stops. Otherwise, the fitness of each individual in the population is evaluated and the best solutions are "bred" with each other to form a new population, P(g+1); the poorer solutions are discarded. If the new population does not satisfy the termination criterion, the process continues.

2.3.1 Population Regeneration Factors

There are three common ways to develop a new generation of solutions: reproduction, crossover, and mutation. Reproduction adds a copy of a fit individual to the next generation. Crossover emulates the process of creating children, and involves the creation of new individuals (children) from the two fit parents by a recombination of their genes (parameters). Under mutation, there is a small probability that some of the gene values in the population will be replaced with randomly generated values. This has the potential effect of introducing good gene values that may not have occurred in the initial population or which were eliminated during the iterations. In Figure 5, the process is repeated until the new generation has the same number of individuals (M) as the previous one.

3 Market Forecasting

Market forecasting involves projecting such things stock market indexes, like the Standard and Poor's (S&P) 500 stock index, Treasury bill rates, and net asset value of mutual funds. The role of SC in this case is to use quantitative inputs, like technical indices, and qualitative factors, like political effects, to automate stock market forecasting and trend analysis. This section provides an overview of representative SC studies in this area.

Apparently, White (1988) was the first to use NNs for market forecasting. He was curious as to whether NNs could be used to extract nonlinear regularities from economic time series, and thereby decode previously undetected regularities in asset price movements, such as fluctuations of common stock prices. The purpose of his paper was to illustrate how the search for such regularities using a feed-forward NN (FFNN) might proceed, using the case of IBM daily common stock returns as an example. White found that his training results were over-optimistic, being the result of over-fitting or of learning evanescent features. He concluded, "the present neural network is not a money machine."

Chiang et. al. (1996) used a FFNN with backpropagation (BP) to forecast the end-of-year net asset value (NAV) of mutual funds, where the latter was predicted using historical economic information. They compared those results with results obtained using traditional econometric techniques and concluded that NNs "significantly outperform regression models " when limited data is available.

Kuo et. al. (1996), recognized that qualitative factors, like political effects, always play a very important role in the stock market environment, and proposed an intelligent stock market forecasting system that incorporates both quantitative and qualitative factors. This was accomplished by integrating a NN and a fuzzy Delphi model (Bojadziev and Bojadziev, 1997 p. 71); the former was used for quantitative analysis and decision integration, while the later formed the basis of the qualitative model. They applied their system to the Taiwan stock market.

Kim and Chun (1998) used a refined probabilistic NN (PNN), called an arrayed probabilistic network (APN), to predict a stock market index. The essential feature of the APN was that it produces a graded forecast of multiple discrete values rather than a single bipolar output. As a part of their study, they use a "mistake chart," which benchmarks against a constant prediction, to compare FFNN with BP models with a PNN, APN, recurrent NN (RNN), and case based reasoning. They concluded that the APN tended to outperform recurrent and BP networks, but that case base reasoning tended to outperform all the networks.

Aiken and Bsat (1999) use a FFNN trained by a genetic algorithm (GA) to forecast three-month U.S. Treasury Bill rates. They conclude that an NN can be used to accurately predict these rates.

Edelman et. al. (1999) investigated the use of an identically structured and independently trained committee of NNs to identify arbitrage opportunities in the Australian All-Ordinaries Index. Trading decisions were made based on the unanimous consensus of the committee predictions and the Sharpe Index was used to assess out-of-sample trading performance. Empirical results showed that technical trading based on NN predictions outperformed the buy-and-hold strategy as well as "naive prediction". They concluded that the reliability of the network predictions and hence trading performance was dramatically enhanced by the use of trading thresholds and the committee approach.

Thammano (1999) used a neuro-fuzzy model to predict future values of Thailand's largest government-owned bank The inputs of the model were the closing prices for the current and prior three months, and the profitability ratios ROA, ROE and P/E. The output of the model was the stock prices for the following three months. He concluded that the neuro-fuzzy architecture was able to recognize the general characteristics of the stock market faster and more accurately than the basic backpropagation algorithm. Also, it could predict investment opportunities during the economic crisis when statistical approaches did not yield satisfactory results.

Trafalis (1999) used FFNNs with BP and the weekly changes in 14 indicators to forecast the change in the S&P 500 stock index during the subsequent week. In addition, a methodology for pre-processing of the data was devised, which involved differencing and normalizing the data, was successfully implemented. The text walked the reader though the NN process.

Tansel et. al. (1999) compared the ability of linear optimization, NNs, and GAs to model time series data using the criteria of modeling accuracy, convenience and computational time. They found that linear optimization methods gave the best estimates, although the GAs could provide the same values if the boundaries of the parameters and the resolution were selected appropriately, but that the NNs resulted in the worst estimations. However, they noted that non-linearity could be accommodated by both the GAs and the NNs and that the latter required minimal theoretical background.

Garliauskas (1999) investigated stock market time series forecasting using a NN computational algorithm linked with the kernel function approach and the recursive prediction error method. The main idea of NN learning by the kernel function is that the function stimulates to changes of the weights in order to achieve convergence of the target and forecast output functions. He concluded that financial times series forecasts by the NNs were superior to classical statistical and other methods.

Chan et. al. (2000) investigated financial time series forecasting using a FFNN and daily trade data from the Shanghai Stock Exchange. To improve speed and convergence they used a conjugate gradient learning algorithm and used multiple linear regression (MLR) for the weight initialization. They conclude that the NN can model the time series satisfactorily and that their learning and initialization approaches lead to improved learning and lower computation costs.

Kim and Han (2000) used a NN modified by a GA to predict the stock price index. In this instance, the GA was used to reduce the complexity of the feature space, by optimizing the thresholds for feature discretization, and to optimize the connection weights between layers. Their goal was to use globally searched feature discretization to reduce the dimensionality of the feature space, eliminates irrelevant factors, and to mitigate the limitations of gradient descent. They concluded that the GA approach outperformed the conventional models.

Romahi and Shen (2000) developed an evolving rule based expert system for financial forecasting. Their approach was to merge FL and rule induction so as to develop a system with generalization capability and high comprehensibility. In this way the changing market dynamics are continuously taken into account as time progresses and the rulebase does not become outdated. They concluded that the methodology showed promise.

Abraham et. al. (2001) investigated hybridized SC techniques for automated stock market forecasting and trend analysis. They used principal component analysis to preprocess the input data, a NN for one-day-ahead stock forecasting and a neuro-fuzzy system for analyzing the trend of the predicted stock values. To demonstrate the proposed technique, they analyzed 24 months of stock data for the Nasdaq-100 main index as well as six of the companies listed therein. They concluded that the forecasting and trend prediction results using the proposed hybrid system were promising and warranted further research and analysis.

Cao and Tay (2001) used Support Vector Machines (SVMs) to study the S&P 500 daily price index. The generalization error with respect to the free parameters of SVMs were investigated and found to have little impact on the solution. They conclude that it is advantageous to apply SVMs to forecast the financial time series.

Hwarng (2001) investigated NN forecasting of time series with ARMA (p,q) structures. Using simulation and the performance of the Box-Jenkins model as a benchmark, it was concluded that FFNN with BP generally performed well and consistently for time series corresponding to ARMA(p,q) structures. Using the randomized complete block design of experiment, he concluded that overall, for most of the structures, FFNN with BP performed significantly better when a particular noise level was considered during network training

As a follow-up to Kuo et. al. (1996), Kuo et. al. (2001) developed a GA-based FNN (GFNN) to formulate the knowledge base of fuzzy inference rules, which can measure the qualitative effect (such as the political effect) in the stock market. The effect was further integrated with the technical indexes through the NN. Using the clarity of buying-selling points and buying-selling performance based on the Taiwan stock market to assess the proposed intelligent system, they conclude that a NN based on both quantitative (technical indexes) and qualitative factors is superior to one based only on quantitative factors.

4 Trading Rules

If one dollar were invested in 1926 in 1-month U.S. Treasury bills, it would have grown to $14 by December 1996. If that dollar had been invested in the S&P 500, it would have grown to $1,370 during that period. If the dollar had been invested with monthly switching to either Treasury bills or the S&P 500, whichever asset would perform the best during that month, it would have grown to over $2 billion dollars during that period.[6] Timing clearly is relevant and it is not surprising that trading rules have evolved that purport to optimize buy/sell timing decisions.

Of course, the extent to which timing is feasible is controversial. Sharp (1975) was skeptical that market timing could be profitable and Droms (1989) concluded that successful timing requires forecasting accuracy beyond the ability of most managers. Nonetheless, researchers continue to explore and enhance trading rules, driven, in large part, by the expanding technology. The goal of SC, as it pertains to trading rules, is to create a security trading decision support system, which, ideally, is fully automated and triggered by both quantitative and qualitative factors. This section provides an overview of representative SC studies in this area.

Kosaka et. al. (1991) demonstrated the effectiveness of applying FL and NNs to buy/sell timing detection and stock portfolio selection. They reported that in a test of their model's ability to follow price trends, it correctly identified 65% of all price turning points.

Wilson (1994) developed a fully automatic stock trading system that took in daily price and volume data on a list of 200 stocks and 10 market indexes and produced a set of risk-reward ranked alternate portfolios. The author implemented a five step procedure: a chaos-based modeling procedure was used to construct alternate price prediction models based on technical, adaptive, and statistical models; then, a SOFM was used to select the best model for each stock or index on a daily basis; then, a second SOFM was used to make a short-term gain-loss prediction for each model; then, a trade selection module combined these predictions to generate buy-sell-hold recommendations for the entire list of stocks on a daily basis; and finally, a portfolio management utility combined the trading recommendations to produce the risk-reward ranked portfolios. He concluded that the stock trading systems could produce better results than index funds and at the same time reduce risk.

Frick et. al. (1996) investigated price-based heuristic trading rules for buying and selling shares. Their methodology involved transforming the time series of share prices using a heuristic charting method that gave buy and sell signals and was based on price change and reversals. Based on a binary representation of those charts, they used GAs to generate trade strategies from the classification of different price formations. They used two different evaluation methods: one compared the return of a trading strategy with the corresponding riskless interest rate and the average stock market return; the other used its risk-adjusted expected return as a benchmark instead of the average stock market return. Their analysis of over one million intra-day stock prices from the Frankfurt Stock Exchange (FSE) showed the extent to which different price formations could be classified by their system and the nature of the rules, but left for future research an analysis of the performance of the resulting trading strategies.

Kassicieh et. al. (1997) examined the performance of GAs when used as a method for formulating market-timing trading rules. Their goal was to develop a monthly strategy for deciding whether to be fully invested in a broad based stock portfolio, the S&P 500, or a riskless investment, treasury bills. Following the methodology of Bauer (1994), their inputs were differenced time series of 10 economic indicators and the GA used the best three of these series to make the timing (switching) decision. They benchmarked against the dollar accumulation given a perfect timing strategy, and concluded that their runs produced excellent results.

As a follow-up study, Kassicieh et. al. (1998) used the same GA with different data transformation methods applied to economic data series. These methods were the singular value decomposition (SVD) and principal component NN with 3, 4, 5 and 10 nodes in the hidden layer. They found that the non standardized SVD of economic data yielded the highest terminal wealth for the time period examined.

Allen and Karjalainen (1999) used a GA to learn technical trading rules for the S&P 500 index using daily prices from 1928 to 1995. However, after transaction costs, the rules did not earn consistent excess returns over a simple buy-and-hold strategy in the out-of-sample test periods. The rules were able to identify periods to be in the index when daily returns were positive and volatility was low and out of the index when the reverse was true, but these latter results could largely be explained by low-order serial correlation in stock index returns.

Fernandez-Rodriguez et. al. (1999) investigated the profitability of a simple technical trading rule based on NNs applied to the General Index of the Madrid Stock Market. They found that, in the absence of trading costs, the technical trading rule is always superior to a buy-and-hold strategy for both "bear" and "stable" markets but that the reverse holds during a "bull" market.

Baba et. al. (2000) used NNs and GAs to construct an intelligent decision support system (DSS) for analyzing the Tokyo Stock Exchange Prices Indexes (TOPIX). The essential feature of their DSS was that it projected the high and low TOPIX values four weeks into the future and suggested buy and sell decisions based on the average projected value and the then-current value of the TOPIX. To this end, they construct an (8, 15, 2) FFNN using a hybrid weight-training algorithm that combines a modified BP method with a random optimization method. Initially, the buy-sell decision was on an all-or-nothing basis; subsequently, using the GAs, an algorithm was developed for buying or selling just a portion of the shares. They conclude that NNs and GAs could be powerful tools for dealing with the TOPIX.

5 Option Pricing

This section provides an overview of the use of SC technologies for pricing options. As expected, the Black-Scholes option pricing model was a benchmark for many of the SC solutions. On the one hand, the issue was the extent to which the SC out-of-sample performance could duplicate the Black-Scholes result; on the other hand, the issue was the extent to which the SC model could outperform the Black-Scholes model. Another line of inquiry was related to methods for estimating the volatility[7] of options, and involved a comparison of implied volatility, historical volatility, and a SC-derived volatility. Other topics addressed included common option features used for SC modeling and specific types of options that have been modeled. This section gives an overview of these SC applications.

Malliaris and Salchenberger (1994) compared the estimated volatility of daily S&P 100 Index stock market options using implied volatility, historical volatility, and a volatility based on a FFNN (13-9-1) with BP. They used the following 13 features: change in closing price, days to expiration, change in open put volume, the sum of the at-the-money strike price and market price of the option for both calls and puts for the current trading period and the next trading period, daily closing volatility for current period, daily closing volatility for next trading period, and four lagged volatility variables. They concluded that the NN was far superior to the historical method.

Chen and Lee (1997) illustrated how GAs, as an alternative to NNs, could be used for option pricing. To this end, they tested the ability of GAs to determine the price of European call options, assuming the exact price could be determined using Black-Scholes option pricing theory. They conclude that the results were encouraging.

Anders et. al. (1998) used statistical inference techniques to build NN models to explain the prices of call options written on the German stock index DAX. Some insight into the pricing process of the option market was obtained by testing for the explanatory power of several NN inputs. Their results indicated that statistical specification strategies lead to parsimonious NNs with superior out-of-sample performance when compared to the Black-Scholes model. They further validated their results by providing plausible hedge parameters.

Gottschling et. al. (1999) discussed a novel way to price a European call option using a proposed new family of density functions and the flexible structure of NNs. The density functions were based upon the logarithm of the inverse Box-Cox transform[8]. Essentially, they viewed the activation function of a NN as a univariate pdf and constructed their family of probability density functions, which have the property of closed form integrability, as the output of a single hidden layer NN. Then, observing that the price of a European call option could be expressed in terms of an integral of the cumulative distribution function of risk neutralized asset returns, they derived a closed form expression from which the free parameters could then be estimated.

Yao et. al. (2000) use a FFNN with BP to forecast the option prices of the Nikkei 225 index futures. Different data groupings affected the accuracy of the results and they concluded that the data should be partitioned according to moneyness (the quotient of stock prices to strike prices). Their results suggested that for volatile markets a NN option-pricing model outperforms the traditional Black-Scholes model, while the Black-Scholes model is appropriate for pricing at-the-money options.

Amilon (2001) examined whether a FFNN with BP could be used to find a call option pricing formula that corresponded better to market prices and the properties of the underlying asset than the Black-Scholes formula. The goal was to model a mapping of some input variable onto the observed option prices and to benchmark against the Black-Scholes model using historical and implicit volatility estimates. He found that, although the NNs were superior in the sense that they outperform the benchmarks both in pricing and hedging performances, the results often were insignificant at the 5% level.

6 Bond Ratings

Bond ratings are subjective opinions on the ability to service interest and debt by economic entities such as industrial and financial companies, municipals, and public utilities. They are published by major bond rating agencies, like Moody's and Standard & Poor's, who guard their exact determinants. Several attempts have been made to model these bond ratings, using methods such as linear regression and multiple discriminant analysis, and in recent years SC has been applied to the problem. This section provides an overview of three such SC studies.

Surkan and Ying (1991) investigated the feasibility of bond rating formulas derived through simplifying a trained FFNN with BP. Under their method, features are systematically eliminated, based on the magnitude of the weights of the hidden layer and subject to error tolerance constraints, until all that remains is a simple, minimal network. The network weights then provide information for the construction of a mathematical formula. In their example, the result of refining the network model was a reduction from the seven features provided in the original financial data to only the two that contribute most to bond rating estimates. The derived formula was found to generalize very well.

Although not specifically addressing bond ratings, one example of how FL could be applied to bonds was provided by Hosler (1992 p. 15), who showed how MFs could be used to describe the risk of call of a security. She noted that randomness is associated with the behavior of market interest rates and that fuzziness arises from the subjective opinion of the investor. The function could be altered to reflect the desirability of the security based upon the call risk.

Daniels and Kamp (1999) applied NNs to bond rating, with a special emphasis on the flexibility of the NNs and their validity, especially when the number of observations is small. Their aim was to establish a general network construction procedure and, to that end, they discussed how techniques such as cross-validation and monotonicity analysis[9] can be effectively combined to optimize the NN. A special class of monotonic NNs and a corresponding training algorithm were developed.

7 Portfolio Construction

Portfolio construction is that part of the investment process that involves the determination of which assets to invest in and the proportion of funds to invest in each of the assets. At a minimum, effective portfolio optimization involves simultaneously maximizing the portfolio return and minimizing the portfolio risk, subject to various constraints, but it also can involve such things as maximizing the possibility of reaching higher returns. This section provides an overview of some of the SC studies in this area.

Lowe (1994) demonstrated the use of NNs in two types of capital market problems: effective portfolio optimization and short-term prediction of multiple equities. Assuming the existence of a market portfolio, his first goal was to find an approximating portfolio that minimized the "risk," defined in terms of the mean squared error between the market portfolio and the approximating portfolio, subject to constraints, which he transformed into an analytic cost function, and resolved using an analog NN. He viewed short-term equities prediction as a problem in nonlinear multichannel time series forecasting, which can be addressed by a FFNN and resolved using a radial basis function. The network then was used to predict the one stock in the approximating portfolio that would gain the most in the next investment period.

Wendt (1995) used a GA to build a portfolio efficient frontier. The underlying data consisted of 250 scenarios of annual returns for eight asset classes. To evaluate the GA process, the final GA output was compared to the efficient frontier created by a sophisticated nonlinear optimizer. After about 50 cycles, the GA found portfolios very close to the efficient frontier generated by the nonlinear optimizer.

Guo and Huang (1996) used a possibilistic linear programming method for optimal asset allocation based on simultaneously maximizing the portfolio return, minimizing the portfolio risk and maximizing the possibility of reaching higher returns. This was analogous to maximizing mean return, minimizing variance and maximizing skewness for a random rate of return.

The authors conceptualized the possibility distribution of the imprecise rate of return of the i-th asset of the portfolio as the fuzzy number $\tilde{r}_i = (r_i^p, r_i^m, r_i^o)$, where r_i^p, r_i^m, r_i^o were the most pessimistic value, the most possible value, and the most optimistic value for the rate of return, respectively. Then, taking the weighted averages of these values, they defined the imprecise rate of return for the entire portfolio as $\tilde{r} = (r^p, r^m, r^o)$, the portfolio risk as $(r^m - r^p)$, and the portfolio skewness as $(r^o - r^m)$. The authors then showed in a step-by-step fashion how the portfolio could be optimized using Zimmermann's (1978) fuzzy programming method. The authors conclude that their algorithm provided maximal flexibility for decision makers to effectively balance the portfolio's return and risk.

Jackson (1997) applied a GA to the problem of asset allocation, first using the traditional mean variance approach and then using a direct utility maximization method for a step utility function. As a benchmark, he compared the performance of GAs with Newton's method of optimization. In the first case, he assumed the fund was maximizing the expected utility of wealth, which lead to a quadratic objective function. He found that the portfolio compositions were similar for both the Newton method and the GA, but that the GA took considerably longer to optimize the portfolio. In the second case, where the fund had a step utility function, Newton's method was very unstable, as a gradient-based method has difficulty with steps in the utility function, and produced different results for different starting values. In contrast, the GA was more robust to discontinuities in the search space, and not as sensitive to the starting values.

8 Comment

The purpose of this article has been to provide the reader with an overview of where NNs, FL and GAs have been implemented in the capital markets and to document the manner in which these SC technologies were employed. Based on these studies, there is ample evidence that SC have made inroads into many facets of the capital markets. As we improve our understanding of the strengths and weaknesses of the SC technologies and improve the manner by which we leverage their best features, it seems inevitable that SC will become one of our important tools for analyzing capital markets.

References

Abraham, A., B. Nath and P. K. Mahanti. (2001) "Hybrid Intelligent Systems for Stock Market Analysis," Computational Science, Springer-Verlag Germany, Vassil N. Alexandrov et. al. (Eds.), ISBN 3-540-42233-1, San Francisco, USA, pp. 337-345.

Aiken, M. and M. Bsat. (1999) "Forecasting Market Trends with Neural Networks." Information Systems Management 16 (4), 42-48.

Alemdar, N. M. and S. Özyildirim. (1998) A genetic game of trade, growth and externalities." Journal of Economic Dynamics and Control 22, 811-832.

Allen, F. and R. Karjalainen. (1999) "Using genetic algorithms to find technical trading rules," Journal of Financial Economics 51: 245–271

Amilon, H. (2001) "A Neural Network Versus Black-Scholes: A Comparison of Pricing and Hedging Performances." Department of Economics, Lund University, Lund, Sweden.

Anders, U., O. Korn and C. Schmitt. (1998) "Improving the Pricing of Options: A Neural Network Approach." Journal of Forecasting 17 (5-6), 369-388.

Baba, N., N. Inoue and H. Asakawa. (2000) "Utilization of Neural Networks & GAs for Constructing Reliable Decision Support Systems to Deal Stocks." Working paper of Information Science, Osaka-Kyoiku University, Asahiga-Oka, 4-698-1, Kashiwara City, Osaka Prefecture, 582-8582, Japan.

Bezdek, J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

Bishop, C. M. (1995) Neural *Networks for Pattern Recognition*, Clarendon Press.

Bojadziev, G. and Bojadziev, M. (1997) *Fuzzy Logic for Business, Finance and Management*. World Scientific, Singapore.

Brockett, P. L., Xia, X., and Derrig, R. A. (1998) "Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud," *J Risk and Insurance* 65(2), pp. 245-274.

Cao, L. and F. E. H. Tay. (2001) "Financial Forecasting Using Support Vector Machines," Neural Comput & Applic (2001) 10: 184–192.

Chan, M-C, C-C Wong, and C-C Lam. (2000) "Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization," Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

Chen, S.-H. and W.-C. Lee. (1997) "Option Pricing with Genetic Algorithms: A Second Report." IEEE. 21-25.

Chiang, W.-C., T. L. Urban and G. W. Baldridge. (1996) "A Neural Network Approach to Mutual Fund Net Asset Value Forecasting." Omega, Int. J. Mgmt Sci. 24 (2), 205-215.

Daniels, H and B. Kamp. (1999) "Applications of MLP networks to bond rating and house pricing." Neural Computing and Applications 8 (3), 226-234.

Datz, J. O. and D. L. McCormick. (2000). *The Encyclopedia of Trading Strategies*, New York: McGraw-Hill.

Droms, W. G. (1989) "Market Timing as an Investment Policy," Financial Analysts Journal, January/February, pp. 73-77.

Edelman, D., P. Davy and Y. L. Chung. (1999) "Using Neural Network Prediction to achieve excess returns in the Australian All-Ordinaries Index". In: Queensland Financial Conference, Sept 30th & Oct 1st, Queensland University of Technology

Falas, T., A. Charitou and C. Charalambous. (1994) "The Application of Artificial Neural Networks in the Prediction of Earnings." IEEE International Conference on Neural Networks 6, 3629-3633. 27 June-2 July.

Farmer, J. D. and A. W. Lo. (1999) "Frontiers of finance: Evolution and efficient markets," Proc. Natl. Acad. Sci. USA Vol. 96, pp. 9991–9992.

Fernandez-Rodriguez, F., C. González-Martel and S. Sosvilla-Rivero. (1999) "On the Profitability of Technical Trading Rules Based on Artificial Neural Networks: evidence from the Madrid Stock Market." Working Paper.

Francis, L. (2001), "Neural Networks Demystified," *Casualty Actuarial Society Forum*, Winter, pp. 253-319.

Frick, A., R. Herrmann, M. Kreidler, A. Narr and D. Seese. (1996) "Genetic-Based Trading Rules-A New Tool to Beat the Market With? – First Empirical Results." AFIR Vol. 2, 997-1017.

Garliauskas, A. (1999) "Neural Network Chaos and Computational Algorithms of Forecast in Finance." Proceedings of the IEEE SMC Conference on Systems, Man, and Cybernetics 2, 638-643. 12-15 October.

Gottschling, A., C. Haefke, and H. White. (1999) "Closed Form Integration of Artificial Neural Networks with Some Applications to Finance." Working Paper

Guo, L. and Z. Huang. (1996) "A Possibilistic Linear Programming Method for Asset Allocation," *Journal of Actuarial Practice*, 2(1):67-90.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.

Hosler, V. R. 1992. "The Application of Fuzzy Sets to Group Health Underwriting," ARCH 2, pp. 1-63.

Hwarng, H. B. (2001) "Insights into Neural-Network Forecasting of Time Series Corresponding to ARMA (p,q) Structures." International Journal of Management Science, Omega, 29, 273-289.

Jackson, A. (1997) "Genetic Algorithms for Use in Financial Problems." AFIR Vol 2, 481-503.

Jang, J-S.R., Sun, C-T., and Mizutani, E. (1997) *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice Hall, Upper Saddle River, NJ.

Kassicieh, S. K., T. L. Paez and G. Vora. (1997) "Investment Decisions Using Genetic Algorithms." Proceedings of the Thirtieth Hawaii International Conference on System Sciences 5, 484-490. 7-10 January.

Kassicieh, S. K., T. L. Paez and G. Vora. (1998) "Data Transformation Methods for Genetic-Algorithm-Based Investment Decisions." Proceedings of the Thirty-First Hawaii International Conference on System Sciences 5, 122-127. 6-9 January.

Kim, S. H. and S. H. Chun. (1998) "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index." *International Journal of Forecasting*. 14, 323-337.

Kim, K-J. and I. Han. (2000) "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index." Published by Elsevier Science, Ltd., Expert Systems with Applications, 19, 125-132.

Kohonen, T. (1988) "Self-Organizing Feature Maps," *Self-Organizing and Associative Memory*, 2nd ed. Spring-Verlag, Berlin, Heidelberg, Germany.

Kosaka, M., H. Mizuno, T. Sasaki, R. Someya and N. Hamada. (1991) "Applications of fuzzy logic/neural network to securities trading decision support system." Proceedings of the IEEE conference on Systems, Man, and Cybernetics 3: 1913-1918. 13-16 October.

Kuo, R. J., C. H. Chen and Y. C. Hwang. (2001) "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network," Fuzzy Sets and Systems: 118 (1) (2001) pp. 21-45.

Kuo, R. J., L. C. Lee and C. F. Lee. (1996) "Integration of Artificial Neutal Networks and Fuzzy Delphi for Stock Market Forecasting." IEEE, June, 1073-1078.

Lowe, D. (1994) "Novel Exploitation of Neural Network Methods in Financial Markets." IEEE International Conference on Neural Networks 6, 3623-3628. 27 June-2 July.

Malliaris, M. and L. Salchenberger. (1994) "Do-ahead replaces run-time: a neural network forecasts options volatility." Proceedings of the Tenth Conference on Artificial Intelligence for Applications, 480-481. 1-4 March.

Neely, C. J. and P. A.Weller. (2002) " Predicting Exchange Rate Volatility: Genetic Programming Versus GARCH and RiskMetrics," The Federal Reserve Bank of St. Louis 84(3): 43-54.

Rast, M. (1997) "Forecasting Financial Time Series with Fuzzy Neural Networks." IEEE International Conference Intelligent Processing Systems 1, 432–434. 28-31 October.

Refenes, A_P. 1995. *Neural Networks in the Capital Markets*. New York: John Wiley & Sons.

Refenes, A.-P. N., Y. Abu-Mostafa, J. Moody and A. Weigend. (1995) Neural Networks in Financial Engineering, London: World Scientific Publisjhing Company.

Resta, M. (2000a) "ATA: The Artificial Technical Analyst Building Intra-Day Market Strategies." *IEEE*, 729-732.

Resta, M. (2000c) "Towards an Artificial Technical Analysis?" *IEEE Transactions on Neural Networks*, Vol. XX, No. Y, 100-105.

Resta, M. (2000d) "Towards an Artificial Technical Analysis of Financial Markets." *IEEE Transactions on Neural Networks*, Vol. XX, No. Y, 117-122.

Romahi, Y. and Q. Shen. (2000) "Dynamic Financial Forecasting with Automatically Induced Fuzzy Associations." *IEEE*, 493-498.

Rosenblatt, F. (1959), "Two theorems of statistical separability in the perceptron," *Mechanization of Thought Processes, Symposium held at the National Physical Laboratory*, HM Stationary Office, pp. 421-456.

Shapiro, A.F. (2000) "A Hitchhiker's Guide to the Techniques of Adaptive Nonlinear Models," *Insurance: Mathematics and Economics* 26, pp. 119-132.

Shapiro, A. F., J. S. Pflumm and T. A. DeFilippo. (1999) "The Inner Workings of Neural Networks and Genetic Algorithms," *ARCH,* 1999 Vol. 1, 415-426.

Sharpe, W. F. (1975) "Likely Gains from Market Timing," Financial Analysts Journal, March/April, pp. 60-69.

Surkan, A. J. and X. Ying. (1991) "Bond Rating Formulas Derived Through Simplifying a Trained Neural Network." IEEE International Joint Conference on Neural Networks 2, 1566-1570. 18-21 November.

Tan, C. N. W. (1997) "An Artificial Neural Networks Primer with Financial Applications Examples in Financial Distress Predictions and Foreign Exchange Hybrid Trading System."

Tansel, I. N., S. Y. Yang, g. Venkataraman, A. Sasirathsiri, W. Y. Bao and N. Mahendrakar. (1999) "Modeling Time Series Data by Using Neural Netwroks and Genetic Algorithms." In in Dagli, C. H., A. L. Buczak, J. Ghosh, M. J. Embrechts, and O. Ersoy (Eds.) *Smart Engineering System Design: neural networks, fuzzy logic, evolutionary programming, data mining, and complex systems*. Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE '99). New York: ASME Press, 1055-1060.

Thammano, A. (1999) "Neuro-fuzzy Model for Stock Market Prediction," in Dagli, C. H., A. L. Buczak, J. Ghosh, M. J. Embrechts, and O. Ersoy (Eds.) Smart Engineering System Design: neural networks, fuzzy logic, evolutionary programming, data mining, and complex systems. Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE '99). New York: ASME Press, pp. 587-591.

Thomas, B. (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press.

Trafalis, T. B. (1999) "Artificial Neural Networks Applied to Financial Forecasting," in Dagli, C. H., A. L. Buczak, J. Ghosh, M. J. Embrechts, and O. Ersoy (Eds.) *Smart Engineering Systems: Neural Networks, Fuzzy Logic, Data Mining, and Evolutionary Programming*, Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE'99). New York: ASME Press, 1049-1054.

Tsaih, R. (1999) "Sensitivity Analysis, Neural Networks, and the Finance." *IEEE* International Joint Conference on Neural Networks 6, 3830-3835.

Wendt, R. Q. (1995) "Build Your own GA Efficient Frontier." Risks and Rewards, December: 1, (24) 4-5.

White, H. (1988) "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns" in Proceedings of the Second Annual IEEE Conference on Neural Networks, II: 451-458

Widrow, B. and Hoff, M.E. (1960), "Adaptive switching circuits" *IRE Western Electric Show and Convention Record, Part 4*, August, pp. 96-104.

Wilson, C. L. (1994) "Self-Organizing Neural Network System for Trading Common Stocks." IEEE International Conference on Neural Networks 6, 3651-3654. 27 June-2 July.

Yao, J., Y. Li, C. L. Tan. (2000) "Option Price Forecasting Using Neural Networks." Omega, The International Journal of Management Science, 28 (2000) 455-466.

Zadeh, L.A. (1965) "Fuzzy Sets," *Information and Control*, Vol. 8, pp.338-353.

Zadeh, L. A. (1992) Foreword of the Proceedings of the Second International Conference on Fuzzy Logic and Neural Networks, xiii-xiv, Iizuka, Japan.[10]

Endnotes

[1] Readers not familiar with neural networks, fuzzy logic or genetic algorithms can find a simple overview of the technical details of these technologies in Shapiro (2000). A working knowledge of these technologies can be obtained by reading Francis (2001) and Brockett et al. (1998) for neural networks, Ostaszewski (1993) for fuzzy logic, and Shapiro et al. (1999) and Wendt (1995) for genetic algorithms.

[2] Following Zadeh (1994, p. 192), in this article the term fuzzy logic is used in the broad sense where it is essentially synonymous with fuzzy set theory.

[3] All the FL studies reviewed in this article use some form of the FIS, so that is all that is reviewed in this section. It is important to mention, however, that FISs cannot adapt or learn because they are unable to extract knowledge from existing data and, where this is an issue, a fuzzy clustering method such as the fuzzy c-means algorithm (Bezdek 1981) is often used. The essence of the c-means algorithm is that it produces reasonable centers for clusters of data, in the sense that the centers capture the essential feature of the cluster, and then groups data vectors around cluster centers that are reasonably close to them. The net result is that the within clusters variances are minimized and the between clusters variances are maximized.

[4] The numerical input can be crisp or fuzzy. In this latter event, the input does not have to be fuzzified.

[5] GAs are a subset of the broader category of Evolutionary Computing (EC), which is comprised of the evolutionary optimization methods that work by simulating evolution on a computer. The three main subcategories of EC are Genetic Algorithms (GAs), Evolutionary Programming (EP) and Evolution Strategy (ES) (Thomas (1996). GAs are the most commonly used.

[6] Kassicieh et. al. (1998, p. 122) and Farmer and Lo (1999, p. 9991).

[7] The two main methods for estimating volatility are the historical volatility and the implied volatility. Historical volatility is the annualized standard deviation of historical rates of daily return and is estimate from a sample of past prices of the underlying asset. Implied volatility is obtained by solving the Black-Scholes option pricing model for the volatility that yields the observed call price, and is the standard method of estimating volatility at the moment of trading.

[8] The Box-Cox transformation is useful when the relationship between the variables lacks normality or has a non-constant variance.

[9] Monotonicity analysis measures the extent to which the relationship between the output of the NN and each input variable is monotonic.