

Currency Exchange Rate Forecasting with Neural Networks

Bona Patria Nasution¹ and Arvin Agah^{2†}

¹*Ernst & Young LLP, Kansas City, Missouri, U.S.A.*

²*Department of Electrical Engineering and Computer Science,
The University of Kansas, Lawrence, Kansas 66045 U.S.A.*

ABSTRACT

This paper presents the prediction of foreign currency exchange rates using artificial neural networks. Since neural networks can generalize from past experience, they represent a significant advancement over traditional trading systems, which require a knowledgeable expert to define trading rules to represent market dynamics. It is practically impossible to expect that one expert can devise trading rules that account for, and accurately reflect, volatile and rapidly changing market conditions. With neural networks, a trader may use the predictive information alone or with other available analytical tools to fit the trading style, risk propensity, and capitalization. Numerous factors affect the foreign exchange market, as they will be described in this paper. The neural network will help minimize these factors by simply giving an estimated exchange rate for a future day (given its previous knowledge gained from extensive training). Because the field of financial forecasting is too large, the scope in this paper is narrowed to the foreign exchange market, specifically the value of the *Japanese Yen* against the *United States Dollar*, two of the most important currencies in the foreign exchange market.

KEYWORDS

artificial neural networks, prediction, financial markets

[†]All correspondence should be addressed to Arvin Agah.
e-mail: agah@ukans.edu; tel: (785) 864-7752; fax: (785) 864-0387

[‡]Work performed while Bona Patria Nasution was at the University of Kansas

1. INTRODUCTION

The goal of the research presented in this paper is to study the prediction of foreign currency exchange rates using artificial neural networks. Neural networks, also called artificial neural systems, neural computers, naturally intelligent systems, or neural nets, are modeled after the structure and function of the human brain. Because they can generalize from past experience, namely, learn, neural networks represent a significant advancement over rule-based trading systems, which require a knowledgeable expert to define trading rules to represent market dynamics. It is practically impossible to expect that one expert can devise trading rules that account for, and accurately reflect, volatile and rapidly changing market conditions. Inflexible, rule-based systems are not dynamically adaptive, despite periodic re-optimizations for a system's indicators. Unlike conventional, rule-based technical trading systems that were popular in the 1980s, neural systems do not need predefined trading rules or 'optimization' of technical indicators to generate trading signals. With neural networks, a trader may use the predictive information either alone or with other available analytical tools to fit the trading style, risk propensity, and capitalization.

The idea of applying neural networks to financial forecasting is also very appealing, for neural networks and finance combine to make a relatively new field and exist together in an exclusively fast-paced, high priced, money-making world. Neural networks have been applied to financial decision making as recently as the late 1980s and early 1990s (Mendelsohn, 1993). Moreover, although complete packages have already been developed (Market Technologies Corporation, 1991), they are still relatively scarce and very expensive. In addition, few are targeted to the area of foreign exchange.

Numerous factors affect the foreign exchange market, as they will be described in this paper. To make things clear, the neural network will help minimize these factors by simply giving an estimated exchange rate for the following day (given its previous knowledge gained from extensive training). A trader may use the result of the system to make trading decisions because having an idea of where the market will be the following day will certainly be to his or her advantage. It must be emphasized that a trader should also use other information, which is outside the scope of this paper, in making critical

decisions. It should be noted that the results presented in this paper are of exploratory nature and are not intended as financial advice, and should not be used as such.

Because the field of financial forecasting is so large, the scope of this paper is narrowed to the foreign exchange market, specifically the value of the *Japanese Yen* against the *United States (U.S.) Dollar*. The Japanese Yen is chosen as the specific domain because Japan is one of the best current representatives of Asian economies, and the Japanese Yen is the second-most important currency in the foreign exchange market after the U.S. Dollar.

2. BACKGROUND

The necessary background is provided in this section, including the domain of application (foreign exchange market) and the presented methodology (artificial neural networks).

2.1 The Foreign Exchange Market

In theory, the price of a currency in a floating rate system is determined when supply equals demand. In practice, however, it is difficult to measure supply and demand as they relate to the minute-by-minute actions of prices in the real market. That there are transactions at a particular price indicates that buyers and sellers have met, and the market must be near equilibrium at that moment. The real challenges facing a trader are knowing when supply and demand are shifting and by how much prices are likely to change. Currencies fluctuate when traders and speculators react to the following factors:

- Real business
- News releases
- Economic statistics (fundamental analysis)
- Chart theory (technical analysis)

2.1.1 Real business. A trader's past experience and observations of certain market relationships also play a part in explaining the behavior of markets. Because of the dynamic nature of the economic system, however, these market conventions can change over time, and traders are often forced

to update their personal strategies. Knowing how markets relate to each other and being alert to changes in fundamental economic variables are critical to every trader's success. Dozens of variables differ in importance in today's markets. Traders must be canny and resourceful in choosing the market relationships that will guide them in making decisions. They must also judge how much weight to attach to the comments of various government officials and economists who speak about markets, often with the intention of moving or stabilizing prices.

The markets often overreact to informal comments by politicians who speak on subjects that are unrelated to their official functions. Whereas in other instances, although an individual's comments may have moved the markets at some point, on other occasions an important new announcement may go virtually unnoticed. In countries that are going through political or economic turmoil, these informal comments play a huge role in the currency exchange rate. Realizing that news announcements and working cause-and-effect relationships provide opportunities for speculation is vital for the unaccustomed trader. Traders evaluate all available information to detect reliable buy-and-sell signals to initiate positions in a currency in the hope of offsetting the transaction in a matter of minutes or hours and therefore, locking in a profit.

2.1.2 News releases. In the real currency market, the value of a currency is affected by news releases stemming from economic, as well as political, events and concerns. Dozens of news fragments are released during the trading day. Some have a general impact on the market, whereas others touch only a specific instrument. In most cases, however, the average news story affects nothing at all and only adds to the piles of uneventful information in the annals of financial history. Some news stories, however, whether economic or political, do create short-term currents and eddies in market sentiment, which traders exploit to break prices out of their established 'trading ranges'. On countless occasions, news stories have succeeded to move prices to levels that traders had failed to reach by their own power. In addition, news releases often take traders holding bad positions by surprise. If market reaction to the news threatens to make matters worse, losers are forced to liquidate their 'offside' positions. The excess supply or demand generated from a single news release can cause a significant move in prices.

Although many of the opinions, predictions, and news updates printed in the headlines contradict historical information, some of them inevitably turn out to be correct. For this reason, markets take them so seriously and sometimes overreact. Subjects covered under political news may not be as easy to classify, but their effects will be straightforward in most cases. Political news flashes can relate to (a) the management of the country's economy, (b) the global economy, (c) world politics, (d) foreign policies, and (e) defense policies. Certain current event news, such as natural disasters, may also affect various prices. As in the real market, a quick mind and strong stomach are the qualities that are necessary for success.

2.1.3 Economic statistics. The crucial economic concept affecting exchange rates is the business cycle, defined as the rate of growth of the gross national product (GNP). The GNP is the market value of all final goods and services produced within a given period of time. Fundamental analysis is the strict consideration of economic factors to determine the appropriate levels of exchange rates. Although many other factors, such as government intervention and speculation by large banks, come into play, fundamental analysis is a reliable predictor of the long-term business cycle in its currency's value. That being said, the task of forecasting the growth rate of GNP remains, and it is not an easy task. All economists rely on an array of statistics to estimate the level of growth of GNP, and their conclusions vary widely. A trader can consider a number of vital statistics (including GNP) in making his or her own assessment of the economy's performance. In addition to helping the trader establish the long-term trend in GNP, the release of these statistics is anxiously awaited by the business community and can have an immediate effect on the foreign exchange and financial markets.

These vital statistics apply only to measures taken in the United States of America. Other countries may have their own definitions and measurements. Market experience has proven that price fluctuations resulting from releases of economic statistics are not always logical and orderly. Although sound fundamental information will usually affect markets' predictably within a reasonable time, such information will sometimes be ignored, and the ensuing confusion can result in unexpected price swings. Therefore, positioning that is based on fundamental market impact should be attempted with the utmost care to avoid being influenced by one particular point of view. More information can

be obtained from the United States Commerce Department (Federal Web Locator, 1998). The vital statistics include:

- *Auto Sales*: Auto sales represent roughly half of the durable goods components of monthly retail sales.
- *Capacity Utilization*: The rate of capacity utilization is the level of industrial production divided by an estimate of total industrial capacity.
- *Consumer Price Index*: The Consumer Price Index (CPI) represents the change in prices, from one month to the next, which affects the purchases of an average urban consumer.
- *Federal Deficit or Surplus*: Statistics on federal receipts and expenditures are consolidated into one important number, the federal budget surplus or deficit, which is released monthly.
- *Housing Starts*: Statistics on the housing units started is released monthly by the U.S. Commerce Department. A housing start represents the physical commencement of construction of a single-family dwelling.
- *Industrial Production*: The Industrial Production Index, released in the middle of each month, measures the output of utilities, mining, and manufacturing firms for the preceding month.
- *Trade Balance*: The Merchandise Trade Balance is released monthly by the U.S. Commerce Department (Federal Web Locator, 1998), and is essentially the U.S. dollar value of the difference between American exports and imports of tangible commodities, such as agricultural and industrial products.
- *Money Supply*: In the United States, Money Supply figures are calculated weekly by the Federal Reserve Bank. The most important component is called M1, and includes all cash in circulation and held by banks, travelers' checks, and all checking account measure, as closely as possible.
- *Personal Consumption Expenditure*: Personal Consumption Expenditure (PCE) is the result of Personal Income, and the mathematical difference between income and expenditure is Personal Saving.
- *Personal Income*: The monthly release of Personal Income is the best indicator for future trends in the demand for consumer goods and general economic activity. Personal Income incorporates all income in the form of wages and salaries, interest income, corporate dividends, rental income, and transfer payments (such as social security and farm subsidies).

- *Retail Sales*: Each month the U.S. Commerce Department calculates the value of retail sales for the previous month. Total sales are divided into two groups called 'durable goods' (which last longer than 3 years, as opposed to durables in PCE, which has a life span of more than 1 year) and 'non-durables' (including food, clothes, gasoline, etc.).
- *Unemployment*: The level of unemployment is probably the most scrutinized of all economic indicators. The level is defined as the percentage of civilian labor force that is unemployed but is actively seeking work.

2.1.4 Chart theory. Technical analysts study price movement itself, as opposed to external forces, which influence price movement. Technical analysis could be defined as the science of graphically recording the actual trading history of a stock, currency, commodity, or financial instrument, and extrapolating possible future trends on the basis of that information alone. Referring to historical data charts is useful in understanding the nature of exchange rates in general and their behavior under specific circumstances. This paper can be categorized as technical analysis.

2.2 Neural Networks

2.2.1 Artificial neural network. An *artificial neural network* is an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology (Fausett, 1994), based on the assumptions that:

1. Information processing occurs at many simple elements that are called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which, in a typical neural network, multiplies the signal that is transmitted.
4. Each neuron applies an activation function (usually non-linear) to its net input (sum of weighted input signals) to determine its output signal.

A *neuron* is a building block of a neural network. The neuron is very loosely based on the brain's nerve cell. Neurons receive inputs via weighted links from other neurons. The inputs are processed according to the neuron's activation function. Signals are then passed on to other neurons. There are

three different types of neurons within a neural network.

1. *Input* neurons receive encoded information from the external environment (sensory data).
2. *Output* neurons send signals out to the external environment in the form of encoded answers to the problem presented in the input (control signals).
3. *Hidden* neurons allow intermediate calculation between the input and the output neurons.

Typically, neurons within a layer do not connect to each other. Neurons of different layers communicate with one another by having specific mathematical weights (or connection strengths) assigned to their connections. A neural network with three layers is shown in Fig. 1. The task of designing the appropriate architecture for the neural systems has to be exact, with a number of different neural designs available. The type of neural system chosen to develop for financial forecasting applications is a feed-forward, *back-propagation* system. Learning is accomplished through a complex,

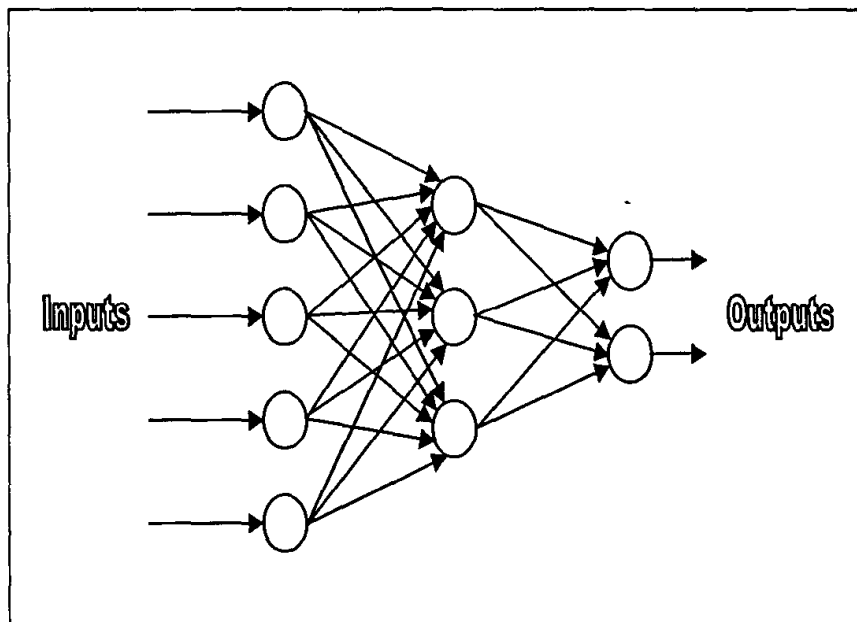


Fig. 1: An artificial neural network with neurons in three layers

mathematical, iterative process, whereby the neural system is ‘trained’ on the input data using statistical error analysis. During training, whenever the system’s projections are incorrect, the connecting weights between neurons are modified to minimize such errors during subsequent iterations. Each input/output pair of data is called a *fact*. The system learns by having these error signals propagate backwards through the neuronal layers to prevent the same error from occurring again each time that a fact is presented to the system during training, hence the name back-propagation. This iterative process will then be repeated until the system is successfully trained. Once this training is accomplished, providing its forecasted predictions should be a simple task for the system.

2.2.2 Applications. Neural networks are different from artificial intelligence systems in that they are trained to learn relationships in the data that they have been given. Just as a child learns the difference between a chair and a table by being shown examples, a neural network learns by being presented with a training set. Because of their complex, non-linear structures, neural networks can find relationships in data where humans or other techniques are unable to do so. Nevertheless, neural networks can only learn if the training set consists of good examples. The old saying “*garbage in—garbage out*” is also true for neural networks. Great care should be taken to present the correlated inputs, and as much prior knowledge as possible should be used to find relevant inputs. Care must also be taken so that the training set is representative because a neural network cannot learn from just a few examples. Therefore, the accuracy of a neural network depends on the quality and quantity of the training.

Distinguishing characteristics of neural networks include the following:

- Neural networks can find patterns in many data sets in which they are given.
- In financial applications, many neural network systems using non-correlated inputs and trading systems are currently yielding good results.
- Even simple neural network models will suffice for data that are not randomly distributed.
- Neural networks learn information by training rather than by programming.
- Neural networks store their knowledge in the interconnections between processing elements.

- Knowledge is not stored in specific memory locations but is distributed throughout the network.
- Neural networks generalize and extract the essence of the knowledge, making them relatively immune to distorted data.
- Neural networks recall knowledge in response to inputs similar to those used in training.
- Generalization allows neural networks to recognize patterns in new input data, even when the networks have never been presented with this data before.

2.3 Definition of Terms

The following terms are used throughout this paper:

- *Open*: The price at which a security opens the trading day. Generally, the opening price reflects the previous day's close, unless extraordinary news or demands to buy or sell have occurred before the market opens.
- *High*: The highest price that a security has traded at throughout the day.
- *Low*: The lowest price that a security has traded at throughout the day.
- *Close*: The price at which a security closes the trading day.
- *Change*: The change in price of a security from the previous day's closing price. For instance, $-1\frac{1}{8}$ means that the security has fallen \$1.12.
- *Volume*: The daily number of shares of a security that changes hands between a buyer and a seller.

3. RELATED WORK

There are a number of business application domains for neural networks, such as: buying-selling of stocks and options, risk management, portfolio management, commodity price forecasting, and foreign exchange rate prediction. In this section, the related work in the domain of financial decision-making is presented.

3.1 Foreign Currency Dealing

The foreign exchange market is very large. In 1989, the volume exchanged was estimated at about \$650 billion dollars per day. Of that amount, a very

small portion (less than 3%) covers actual commercial transactions; the rest is made up of speculative transactions between market makers (Lequarré, 1989). Currently, the most important currencies in the market are the U.S. Dollar (which acts as a reference currency), the Japanese Yen, the British Pound, the German Mark, and the Swiss Franc. Close followers in terms of importance are the French Franc, the Italian Lira, and the Dutch Gulden.

The process of exchanging currencies is called a deal. Deals made to exchange a currency against U.S. Dollars are called direct deals. Deals made to exchange any currency pair not including the dollar are called cross deals. Currencies are normally quoted against the U.S. Dollar. The large part (60%) of foreign exchange trading concerns spot deals, that is, deals that must be settled 1 or 2 days (depending on the currency) after the deal is done (Lequarré, 1989). An active market exists, however, in forward trading (deals that will only be settled at a given time in the future) and in currency swaps. There is also a very active market in derivative instruments (options), and foreign currency futures, which are traded in a centralized exchange.

Economic indicators, monetary parameters, and interest rates are the most important factors influencing exchange rates. Given the large amount of speculative dealing in the market, psychological factors also play a very large role in determining the exchange rate of a currency. Although knowing the exact rate of exchange in the future would be the most profitable, a rough estimation is often enough to make a reasonable profit on dealing operation. For speculative purposes, only trend turning points are actually important (Lequarré, 1989).

3.2 The Foreign Exchange Market Simulator

The Foreign Exchange Market Simulator by Zero Base (Logiceils Zero Base, Inc., 1989) is a simulator that helps in understanding how the foreign exchange market works. This simulator incorporates all the market aspects that are discussed in this paper. Using this software assists one in realizing how helpful it would be if a person could correctly forecast a currency rate for the next day. Initially this software was intended to be used to test the performance of the neural network. Unfortunately, this software generates its own data for each currency rate. This means that it does not use 'actual' data (data from the

actual foreign exchange market). Because the training of the neural network was done with real market data, its forecasts would be irrelevant to the numbers that are generated by the program in the Zero Base Software. Nevertheless, Zero Base Software can be used to study the foreign exchange market.

3.3 Santa Fe Time Series Prediction and Analysis

The data series described in this section is Data Set C of the Santa Fe Time Series Prediction and Analysis Competition, the 'tick-wise' bids for the exchange rate from Swiss francs to U.S. Dollars, as recorded by a currency trading group from August 1990 to April 1991 (Mozer, 1993). 'Tick-wise' means that the samples are produced at irregular intervals of time, as demanded by the traders. Each sample is thus indexed by a predetermined time of the day. There were six prediction tasks in the Competition, but the authors have focused on three: predicting the value of the series: 1, 15, and 60 minutes in the future. The training data set appears challenging because the average sampling rate is high relative to the rate of trends in the data. Although the nature of the data is that it is obviously noisy, smoothing is inappropriate because information may present in the high frequency changes.

The training data consisted of 10 contiguous spans of time with unspecified gaps between them, for a total of 114 days of data. Each day's data were treated as an independent sequence, i.e., it was assumed that no useful information was carried over from one day to the next. For each day, the data were processed to transform tick-wise samples to fixed-interval samples, one minute apart, operating on the assumption that the value of the series remained constant from the time of a sample until the time of the next sample. Performance on this series is measured in units of normalized mean squared error or NMSE. The normalization involved is to divide the mean squared error by the error that would have been obtained when assuming that the future value of the series is the same as the last observed value. This normalization term is the least-squares prediction when assuming a random walk model for the series. NMSE values less than 1.0 thus indicate that structure has been extracted from the series. It was reported that this approach yielded an NMSE of 0.859 for the 15-minute prediction and 0.964 for the 60-minute prediction.

Other research was performed in the Santa Fe Time Series Prediction and Analysis Competition (LeBaron, 1993). Some evidence was found for non-linearity both in the tick-by-tick data and on an hourly series. In addition, tests were run on the properties of the timing of the quotes in the data set. Some trading rules were set up to demonstrate the importance of considering both the economic objectives of trading profits and the transaction costs in a financial forecasting setting. One of the most challenging applications of modern time series forecasting approaches is the area of financial time series. These series show dependencies, which are much weaker and harder to detect than many other time series. The reason for this dependency is obvious. Market prices are not determined independent of trader behavior, and traders will not leave obvious patterns around to be converted into trading profits. Still, the question of whether new techniques may pull hidden structures out of these well-studied series is an interesting one. Many time series at high frequency have become available for many markets. LeBaron's work tests one such high-frequency series from the foreign exchange market. Such series have the potential to reveal the detailed dynamics that are going on at the actual level of trading, using information which is lost at the daily, weekly, and monthly horizons.

3.4 Utilization of Neural Networks

The stereotypical business executive (otherwise known as SBE) does not want to know how or why neural networks work; he or she just wants to make money (Bigus, 1996). The SBE may know what an average or percentage is, but he is morbidly afraid of 'statistics'. An SBE understands profit and loss but does not want to waste his or her time learning things involving complicated math, or even high-school algebra. It is also mentioned that for further information on the SBE, one must see the *Dilbert* comic strip. With an introduction to neural networks for the SBE, neural networks are explained at a practical, intuitive, and necessarily fundamental level without formulas, just what the SBE needs. Other attempts have been made to explain neural networks without formulas (Caudill & Butler, 1990).

Similar to the work presented in this paper, other researchers have tried to apply artificial neural networks to financial markets (Mendelsohn, 1991).

For the past decade, stock and futures traders have come to rely upon various types of technical analysis software programs to make trading decisions. In addition to applying artificial neural systems, it is envisioned that over the next several years, hybrid artificial intelligence-based systems, even combining expert systems, genetic algorithms, and neural networks, will be amalgamated for price forecasting in the financial markets. Before long, this emerging technology will result in a significantly broadened definition of technical analysis from what it is today.

3.5 Technical Analysis

Today's global markets demand new analytical tools for survival and profit as prevailing methods of analysis lose their luster. An emerging method of analysis, synergistic market analysis, can be applied to neural networks for financial forecasting (Mendelsohn, 1993). The selection and combination of various types of market information and transforming the data into a format that is appropriate for neural network training is an integral phase of such analysis. A neural network is only as good as the input data used to train it; therefore, data selection can be a demanding and intricate task. Many factors influence the choice of input data, including individual perspective on the markets. Technical analysis suggests the use of only single-market price data as inputs, whereas conversely, fundamental analysis concentrates solely on data inputs that reflect supply/demand and economic factors. In today's global environment, neither approach alone is sufficient for financial forecasting. Instead, synergistic market analysis combines both approaches with inter-market analysis within a quantitative framework using neural networks. This strategy overcomes the limitations of interpreting inter-market relationships through simple visual analysis of price charts and carries conceptualization of inter-market analysis to its logical conclusion. Once the most appropriate raw input data has been selected, the selection must be preprocessed; otherwise, the neural network will not produce accurate forecasts. Decisions that are made in this phase of development are critical to the overall performance of a network.

3.6 Other Techniques

Other examples of techniques for prediction include the following:

- *Nestor*: a neural network used for mortgage assessment, where the system is used for and reliable evaluation of mortgage applications (Collins et al., 1988).
- *VantagePoint*: a system for developing technical and inter-market analysis computer trading software for commodities futures trading in financial markets using neural networks, moving averages, price and trend forecasts (Market Technologies Corporation, 1998).
- *DataEngine*: an example of management applications that does sales forecasting in industry using neural networks (Bayer AG, 1998).
- *Predict*: a powerful and user-friendly package that is suited to such financial applications as modeling, trend recognition, and futures market trend prediction (Skye Advanced Technologies, 1997). Predict extends the Microsoft Excel environment by adding an extra menu item (Microsoft Corporation, 1997).

4. CURRENCY EXCHANGE DATA

The currency exchange market operates in a slightly unusual fashion. Offers to buy and sell, known as 'quotes', are posted electronically and appear on traders' screens around the world. When actually made, however, a trade is confirmed over the telephone, and no electronic record is posted in the quote sequence. This omission leaves only a partial picture of what is actually happening in the market. Offers can be observed in a detailed way at high frequency, but whether these were prices for which actual trades took place is not known. The latter is less of a problem for more centralized markets, such as the New York Stock Exchange. For this market, some series are available that include information both on quotes and on trades made. Nevertheless, these data sets may still miss some trades, which do not get executed on the floor of the exchange.

In this type of situation, it is therefore difficult to tell if a quote was 'serious', or if traders were just testing a new price range. In addition, a data set that only gives bid quotes from one bank suggests exercising some caution in interpreting results. For example, a trading rule that had been tested using this series may not have actually performed as well in the real world because it may

not have been able to buy or sell at some of the quotes. Obviously, an important phase of this type of research involves the gathering of data. Surprisingly large amounts of market data are available in this domain. Unfortunately, not all data is free, and even less seems to be useful. Two data sources, namely, TradeStar and Pacific Exchange Rate Service, are described in this section.

4.1 TradeStar

TradeStar (1998) is a company that provides information on currency rates against the U.S. Dollar. TradeStar provides rates for the following currencies: Australian Dollar (AUD), British Pound (GBP), Deutsche Mark (DEM), European Currency Unit (ECU), Hong Kong Dollar (HKD), Indian Rupee (INR), Indonesian Rupiah (IDR), Japanese Yen (JPY), Malaysian Ringgit (MYR), New Zealand Dollar (NZD), Philippines Peso (PHP), Singapore Dollar (SGD), South Korean Won (KRW), Taiwan Dollar (TWD), and Thai Baht (THB). The data are foreign exchange data, not stock market data; thus, they are a more stable source for the type of research presented in this paper. One problem with TradeStar's data is that the data are not updated on a regular basis, i.e., although they are built around a relational database, the updating is manual, meaning that the time of which the data is entered is not regular. Another shortcoming is that the only data available were from the previous 7 months, which is not sufficient to train and test a good neural network, considering also that the times are irregular.

4.2 PACIFIC Exchange Rate Service

PACIFIC is the acronym for Policy Analysis Computing and Information Facility in Commerce, which was developed at the University of British Columbia, Vancouver, Canada (Antweiler, 1997). PACIFIC is a project to provide computing and information services to researchers in the field of international trade and business. Additionally, PACIFIC provides computing services to members of the Policy Analysis Division at the Faculty of Commerce and Business Administration at the University of British Columbia. Initiated in December 1995, PACIFIC intends to provide information primarily through the World Wide Web.

PACIFIC's most popular service is the PACIFIC Exchange Rate Service, providing daily exchange rate updates, forecasts, and analyses for 65

countries and 3 commodities, and a database system that allows retrieval and plotting of historic exchange rates. This service provides access to current and historic daily exchange rates through an online database retrieval and plotting system. Also provided is a list of all the currencies of the world, with information on each country's exchange rate regime and ISO-4217 currency code, which is an internationally used code to represent currencies for trading purposes. Analysis and trend projections of the Canadian Dollar, the U.S. Dollar, and the Euro are available as well. This site is dedicated to the support of academic research in the area of exchange rate economics. As data can be retrieved for a particular interval of time, one has to specify which interval is going to be used as experimental data. In the foreign exchange market, forecasts can be done for three different lengths of time:

1. Long-term forecasting (3 years or longer).
2. Intermediate-term forecasting (1 to 3 years).
3. Short-term forecasting (less than 1 year).

5. NEURAL NETWORK IMPLEMENTATION

All the research implementation presented in this paper was done using MATLAB (MathWorks, 1998) because it contains a Neural Network Toolbox that is useful for training and testing a specific neural network. All the code written in MATLAB is included in the Appendix.

5.1 Experimental Setup of Data

The PACIFIC Exchange Rate was chosen to be used as the source of data for the experiments. Because intermediate-term forecasting is the most profitable and most widely used type of forecasting, the selected range is about 14 months of data, with the start date of July 1, 1997 and the end date of August 31, 1998. The retrieved data must be preprocessed into a format so it may be useable. The data were copied into a file with a text format that consists of a column containing the daily exchange rate of the Japanese Yen against the U.S. Dollar, with up to four significant digits of accuracy. Each row represents one day of data. The data are then stored in a data structure

(column vector) for the further processing that is necessary for the research. The first step is to load the input data into a column matrix called *forex_Yen*. Then the entire matrix, which is a (293×1) matrix, is multiplied by 100 so the values are not too small. These values are not normalized to preclude limiting the range of testing of the neural network. Thus, if provided with values outside the normalized range, the neural network can forecast properly. The last 50 days of data are saved for testing, and the rest are used for training the neural network.

For training the neural network, the data must be preprocessed further. It was determined to consider the past 5 days, one trading week of data, before making the sixth day prediction (to improve performance). Thus, another matrix of data is then generated to contain columns of 5 days of data. The reason for choosing the previous 5 days, as opposed to 10 days or more, is that this practice is commonly done in the real market, in which a trader chooses to consider the last trading week before making any financial decisions. The number of columns is 235, which is the result of taking the floor of 293 (number of original data) minus 50 (number of testing days) minus 5 (number of days to consider), resulting in a (5×235) matrix. The MATLAB code to do this is shown in Fig. A-8 (see Appendix). A target vector is then generated to contain the sixth day of actual exchange rate for training purposes (1×235) matrix). The target matrix setup is done in the portion of MATLAB as in Fig. A-9. Each pair, consisting of the 5 previous days and the 6th day value, is called one fact. An example of a fact would be:

[0.8734; 0.8811; 0.8783; 0.8887; 0.8870] --- [0.8850]

5.2 Methods

Getting the neural network to converge (successfully train) for all the given data is difficult because the number of training facts is too large. The larger the training set, the longer it takes to train the neural network. In addition, when needing a very high level of accuracy, training will take even longer. Thus, certain points in the data must be selected that represent the data as a whole to get the neural network to successfully converge. These points serve as a feature vector that represents sampling of the data that best

represents the overall data. The issue that must be addressed is the selection of suitable data points to get the best representation of the data as a whole. Two different approaches were used to address this question.

5.2.1 The Nyquist theorem. One of the important rules of sampling is the *Nyquist Theorem*, which states that the highest frequency that can be accurately represented is one-half of the sampling rate (Jones & Flynn, 1993). The plot of the data in Fig. 2 shows that roughly every 50 days of data, a relative large change in behavior occurs.

Therefore, according to the Nyquist theorem, the feature vector consists of points at every 25 days of data. This approach sounds reasonable, and the network did converge with a relatively high accuracy. Nevertheless, there lies a fundamental problem with this approach: too much subjectivity is involved. To choose a feature vector to train the neural network, one cannot simply assume that trends of the data contain relatively large changes every so many days. There must be more objectivity in the feature selection.

5.2.2 Curve Fitting Approach. For this approach, a polynomial curve must be found that fits the data curve. The given data curve was fitted by the following polynomial:

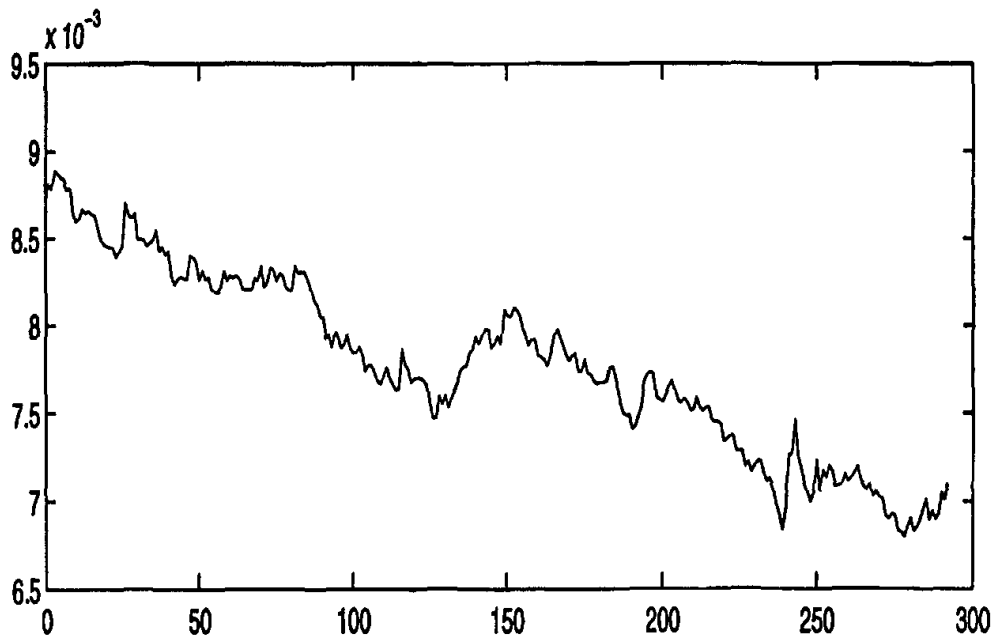


Fig. 2: Japanese Yen against the U.S. Dollar versus days.

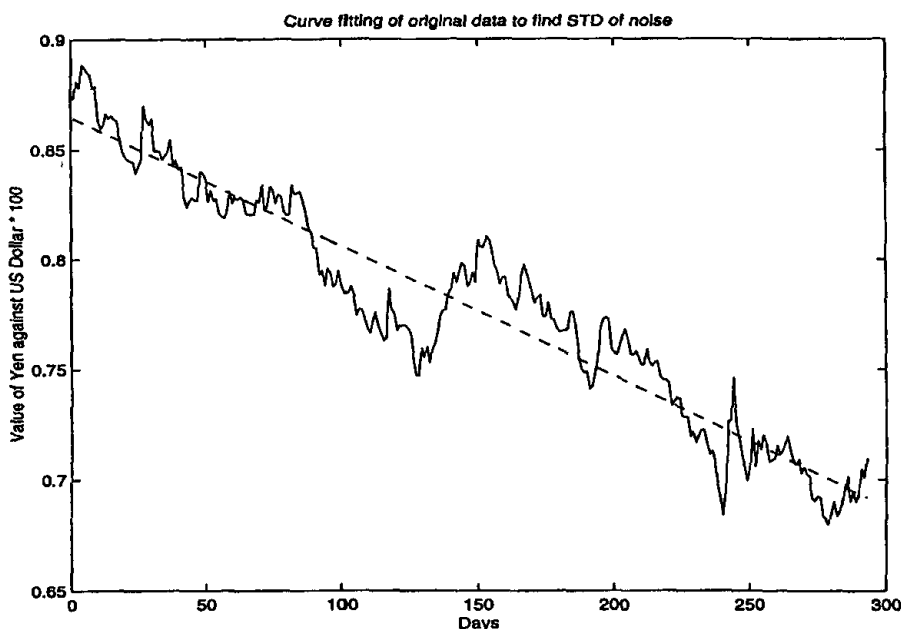


Fig. 3: Original data with the polynomial fit curve (data is represented using a solid line and the fit is presented using the dotted line).

$$y = -0.0006x + 0.8650$$

This is the best polynomial fit that can be found using MATLAB's *polyfit* function, i.e., a degree one polynomial is a good fit for the data, as illustrated in Fig. 3. Each day that the original data intersects with the fitting curve was selected. Note that the original data curve is either above or below the fitting curve. If the original data curve is on one side of the fitting curve for more than 5 days (one trading week) and then crosses to the other side of the fitting curve for more than another 5 days, then the point of intersection is considered as a point in the feature vector. The portion of the MATLAB code that does this is included in Fig. A-10. Setting up the feature vector, along with the target vector for the feature vector to train the neural network, is done using code shown in Fig. A-11. This approach contains the points that best reflect the overall behavior of the original data, thus this is the final approach used to train the neural network.

The following parameters were used:

- *Display Frequency*: set to 100, so it can be determined as to how close the network is to converging at every 100 iterations.
- *Max Epoch*: set to 50000, which is the total number of iterations before the training halts. If the neural network has not reached the desired error goal, then one can continue training for another 50000 iterations.
- *Error Goal*: set to 0.00001^2 times the size of data. The error goal is necessary to determine the neural networks' convergence, i.e., the neural network is said to have converged when the error goal has been met.
- *Learning Rate*: set to 0.0001. The system, however, has learning rate adaptability, and thus the learning rate will actually change and adapt throughout the training process until the best rate necessary for convergence has been found.
- *Learning Rate Increase*: set to 1.04. This determines the rate at which the learning rate may increase.
- *Learning Rate Decrease*: set to 0.6. This determines the rate at which the learning rate may decrease.
- *Momentum*: set to 0.4, to help the neural network converge by assisting in getting through local minimums and maximums that may prevent convergence.

5.5 Testing

After successfully training the neural network, we tested its performance. In MATLAB, a function called *simuff* will call the trained neural network in a feed forward manner and return the outputs of each layer. The last layer (output layer), which in this case will be the forecasted sixth day exchange rate, will be considered. It was initially tested on the feature vectors that were used to for training to verify that the neural network has been trained properly. The error should be very close to zero (if not equal) because these were the same values as those used to train the neural network. Figure A-13 includes the code used to do the initial test. Then the neural network was tested on the testing data, which are the data outside the range of the training data. This means that the neural network has not yet been exposed to the test data. The goal was to see how well the neural network performs. The test data consisted of the last 50 days of the original data. Because, however, the past 5

days of data are considered to forecast the sixth day, there were only 45 test days. A new matrix was generated for the test data, using the code in Fig. A-14. This code generates a (5×45) matrix in which each column is a vector of 5 previous days of exchange rates. These vectors are paired with their target vectors, which will be the actual sixth day of each column vector. This target vector is generated with the MATLAB code in Fig. A-15.

Next, the *simuff* function is called to test the performance of the neural network on the test data, as shown in Fig. A-16. The percentage error is calculated by dividing the difference between the forecasted output and the actual data by the actual data. If the performance of the neural network is not satisfactory at this point, the training can be continued until the achievement of an acceptable level of performance. To continue training for another 50000 iterations, we use the MATLAB's *trainbpx* function while providing it with the new updated weights from the previous training session. The code to perform this function is shown in Fig. A-17. The results of the neural network experiments are presented in the next section (see Fig. 4).

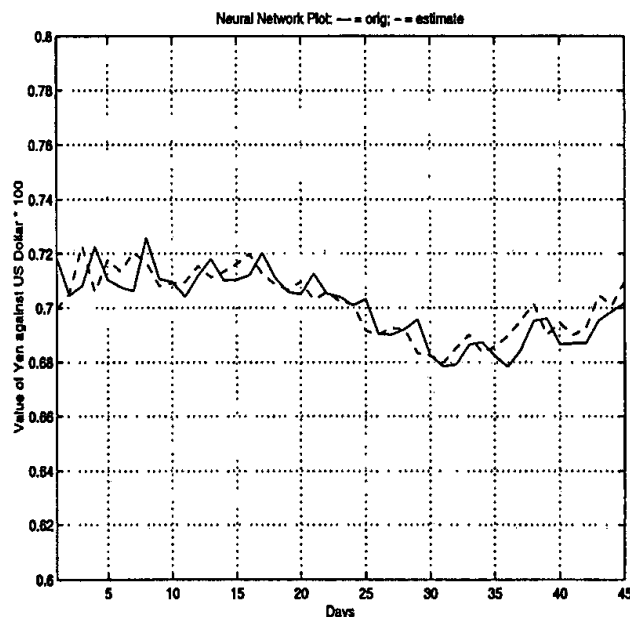


Fig. 4: Plot of neural network results, where the dashed line represents the original data and the solid line represents the forecast values.

TABLE 1

The effects of iteration numbers on average percentage error.

Number of iterations for training	Average percentage error
50,000	2.731006
100,000	1.141966
150,000	0.966556
200,000	0.926451

6. EXPERIMENTAL RESULTS

The neural network produced an average percentage error of 0.843308 at the end of 250,000 iterations. A plot of both the original data and the estimated forecasts of the neural network are shown in Fig. 4. Table 1 shows that the neural network produced better results as the training was continued. The results are also shown in Fig. 5.

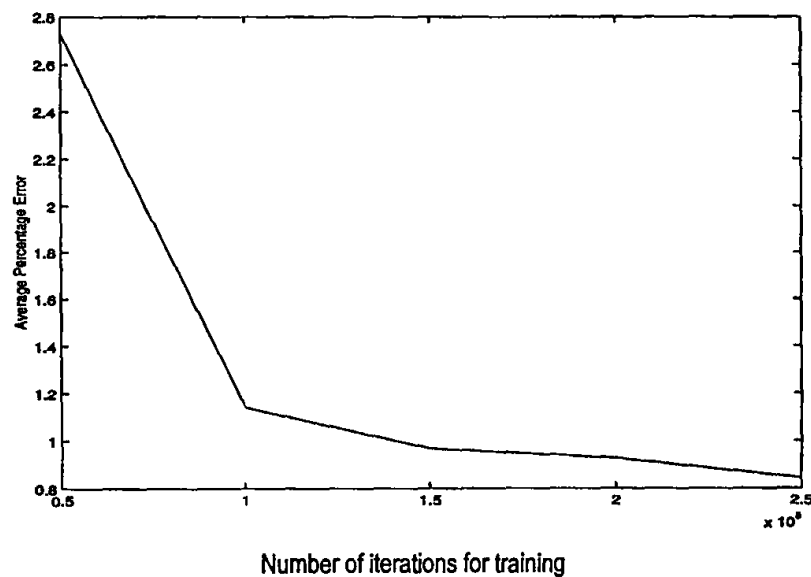


Fig. 5: Plot of neural network training.

As can be seen from Table 1 and Fig. 5, the average percentage error is decreasing. The percentages are relatively small, as the typical expected percentage error is 10%.

5.3 Neural Network Structure

Because the last five days are considered as an input vector to forecast the sixth day of the input data, the neural network initially takes five inputs. The input layer consists of five neurons, and the hidden layer consists of three neurons. As the forecasting is done for one day of output, the output layer contains only one neuron in the output layer. This setup of 5-3-1 neurons for the input layer, hidden layer, and output layer was not randomly selected. Generally, there are a couple of rules to follow:

After experiments with other setups such as 8-4-1, 10-5-1, 10-10-1, and 100-50-1, the 5-3-1 setup proved to be just as good, i.e., the rate of convergence per number of iterations for each setup was not significantly different. The time of convergence, however, increased with the increase in the number of neurons. The goal was to find a setup that allows the neural network to converge at relatively high level of accuracy because the value of the Japanese Yen is very small in comparison with that of the U.S. Dollar. This situation means that any setup should suffice, as long as the level of accuracy is achieved. The 5-3-1 setup converged at a high level of accuracy, and because the total number of neurons in this setup is only nine, it was relatively quicker than those with more neurons at converging.

5.4 Network Training

MATLAB has a built-in Neural Network Toolbox containing tools that enable users to both train and test a neural network, given the proper parameters. To train the network, we used the MATLAB' *trainbpx* function. Initially the connection weights (strengths) between each layer of neurons are set randomly. Then, when the routine is called, the network will begin the training phase by adjusting the connection weights between each neuronal layer. When the error

TABLE 2

Comparison of the three methodologies used for forecasting.

Neural network 250,000 iterations	Linear prediction using mean	Linear prediction using median
0.843308	0.862795	0.925424

goal is met, with the current weights, the network is said to have converged. Figure A-12 (Appendix) includes the code that performs these steps.

Experiments were also performed using a linear prediction, where the mean and median of the past 5 considered days were used as the forecasts for the following sixth day in the series. These turned out to be worthy results. When using the mean of the previous 5 days, the forecasts yielded an average percentage error of 0.862795, and when using the median, the forecasts yielded an average percentage error of 0.925424. Although good, the results were less accurate than those of the neural network that had been trained with 250,000 iterations, which yielded an average percentage error of 0.843308. Table 2 shows the results of the three methods used for forecasting. The comparative results are plotted in Fig. 6.

7. CONCLUSION

Research in the area of neural networks in the domain of the foreign currency exchange market is still relatively new. Further work remains to be done in this area of research. This paper presented one approach to using artificial neural networks to predict foreign currency exchange rates. Potential extensions of this work are described in this section.

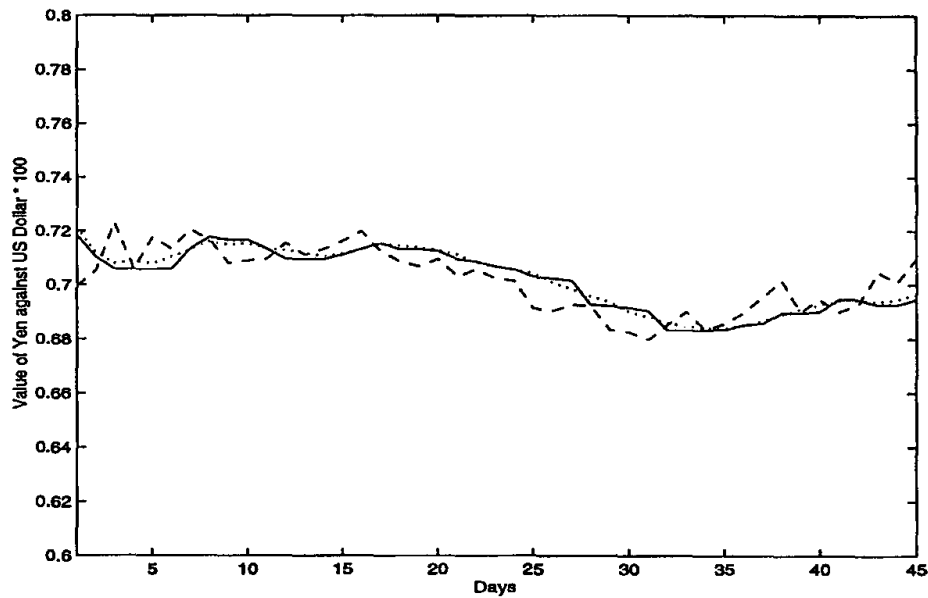


Fig. 6: Comparison plot of mean results and median results. ---, original data; ..., forecast curve of the mean technique; —, forecast curve of the median technique.

7.1 Complex Neural Networks

One possible direction for future work would be to create a system using neural networks that could actually tell a trader to either buy or sell a certain currency given a certain time period and amount to be traded. As previously mentioned, there are many factors that affect the foreign exchange market. Research presented in this paper relies on the technical data aspect of forecasting. However, a system can contain multiple neural networks, where each neural network represents one factor. The results can then be the input to a higher level neural network that can make the decision on whether to buy or sell. Figure 7 illustrates the idea of how this may be implemented.

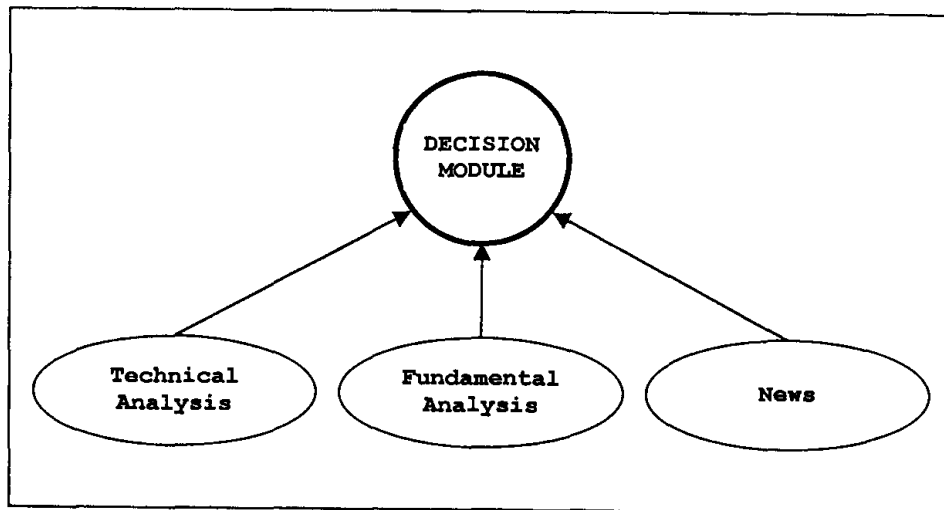


Fig. 7: Multiple stage neural network.

7.2 Other Neuron Definitions

New types of neurons have been designed based on identification maps and superposition of waves (Resconi & Pessa, 1998) Attempts are being made to determine the feasibility of this new neuron and learning algorithm through different application domains. This new approach has proven to be very robust in certain prediction domains. Possible future work would be to test the feasibility of this new type of neuron in the domain of financial forecasting and business decision making. Implementation and testing of this system could then be compared to a back-propagation neural network system.

7.3 Other Potential Future Approaches

Synergistic market analysis is a new method of analysis that can accurately forecast prices and predict trend direction changes (Mendelsohn, 1995). This method accomplishes the forecasting by using neural networks to find the market synergy in today's global interrelated markets via quantifying hidden patterns and

relationships in technical, fundamental, and intermarket data that current, single-market methods of analysis cannot handle.

Other soft computing technologies such as genetic algorithms are useful in implementing synergistic analysis (GA Archives, 1998). These algorithms, which mimic the characteristics associated with evolution, have proven to be robust search mechanisms. They are well suited to optimization problems, such as optimizing neural network training parameters. The same technology incorporated into genetic algorithms has also been used in classifier systems and genetic programming. Classifier systems perform a type of machine learning that generates rules from examples. Genetic programming goes even further by automatically generating a program from a set of primitive constructs. Even *fuzzy logic* (Rich & Knight, 1991) and *chaos* (Mendelsohn, 1995) are finding applications in financial forecasting. No doubt, other technologies will continue to emerge. It should be understood that none of these tools, individually, would provide the whole answer. Instead, each tool has its own role to play in putting together the pieces of the puzzle needed to implement a synergistic approach to today's globalized markets.

REFERENCES

- Antweiler, W. (1997). *Pacific exchange rate service retrieval interface*.
<http://pacific.commerce.ubc.ca/xr/>.
- Bayer AG. (1998). Sales forecasting in industry using neural networks.
Management Applications – Forecasting, March 1998.
- Bigus, J.P. (1996). *Data Mining with Neural Networks: Solving Business Problems – from Application Development to Decision Support*. McGraw-Hill.
- Caudill, M. and Butler, C. (1990). *Naturally Intelligent Systems*. MIT Press, Cambridge.
- Collins, E., Ghosh, S., and Scofield, C.L. (1988). An Application of Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgments. *IEEE International Conference on Neural Networks*, San Diego,

- CA, Vol. II, 459-466.
- Fausett, L. (1994). *Fundamentals of Neural Networks—Architectures, Algorithms, and Applications*. Prentice-Hall, Inc.
- Federal Web Locator. (1998). The federal web locator. <http://www.vcilp.org/Fed-Agency/fedweb.exec.html#doc>, November 1998.
- GA Archives. (1998). The Genetic Algorithms Archive. <http://www.aic.nrl.navy.mil/galist/>, November 1998.
- Jones, J.L. and Flynn, A.M. (1993). *Mobile robots: inspirations to implementation*. A.K. Peters, Wellesley, Massachusetts.
- LeBaron, B. (1993). Nonlinear diagnostic and simple trading rules for high-frequency foreign exchange rates. *Times Series Prediction: Forecasting the Future and Understanding the Past*, (Eds.) Weigend, A.S. and Gershenfeld, N.A., SFI Studies in the Science of Complexity, Vol. XV, Addison-Wesley, 1993, 457-474.
- Lequarré, J.Y. (1993). Foreign Currency Dealing: A Brief Introduction. *Times Series Prediction: Forecasting the Future and Understanding the Past*, (Eds.) Weigend, A.S. and Gershenfeld, N.A., SFI Studies in the Science of Complexity, Vol. XV, Addison-Wesley, 1993, 131-137.
- Logiciels Zero Base, Inc. (1989). The Foreign Exchange Market Simulator.
- Market Technologies Corporation. (1991). *VantagePoint – Intermarket Analysis Software*. http://www.profitaker.com/vantagepoint_software.htm.
- Market Technologies Corporation. (1998). http://profitaker.com/technical_mendelsohn.htm.
- MathWorks. (May 1998). *Getting Started with MATLAB*. <http://www.mathworks.com/products>.
- Mendelsohn, L.B. (1991). Applying artificial neural systems to the financial markets, *Wall Street Micro Investor*, November/December 1991.
- Mendelsohn, L.B. (1993). Preprocessing Data for Neural Networks. *Technical Analysis of Stocks and Commodities*, October 1993.
- Mendelsohn, L.B. (1995). Synergistic market analysis: combining technical, fundamental, and intermarket analysis using artificial intelligence. *Virtual Trading*, Probus Publishing Company, Chicago, Illinois.

- Microsoft Corporation. (1997). *Getting Results with Microsoft Office 97*.
- Mozer, M.C. (1993). Neural net architectures for temporal sequence processing. *Times Series Prediction: Forecasting the Future and Understanding the Past*, (Eds.) Weigend, A.S. and Gershenfeld, N.A., SFI Studies in the Science of Complexity, Vol. XV, Addison-Wesley, 1993, 243-264.
- Resconi, G. and Pessa, E. (1998). The Morphogenetic Neuron as a hierarchical neural network. *EMCSR'98*, Vienna.
- Rich, E. and Knight, K. (1991). *Artificial Intelligence*, Second Edition, McGraw Hill, Inc., New York.
- Skye Advanced Technologies. (1997). Introduction to Neural Networks. <http://skye.co.za/ie/predict.htm>, November 1997.
- TradeStar. (1998). Currency Rates. <http://asiatradedirectory.com/currencies.cfm>, May 1998.

APPENDIX

All the code in this section is written in MATLAB (MathWorks, 1998), using the Neural Network Toolbox.

```
% CREATE NEW INPUT MATRIX OF SIZE (MAX-1,Num) FOR TRAINING
K = 0;
for I = 1:(MAX-Num),
    count = Num;
    % Num IS LENGTH OF NEW ROWS
    for J = 1:Num,
        NewMat(I,J) = forex_Yen(K+count,1);
        count = count - 1;
    end
    K = K + 1;
end
NewMat = NewMat';
```

Fig. A-8: New matrix setup.

```

% GET TARGET MATRIX
Target = zeros(1,MAX-Num);
for I = 1 : (MAX-Num),
    Target(1,I) = forex_Yen(I+Num,1);
End

```

Fig. A-9: Target matrix setup.

```

% GATHER POINTS FOR FEATURE VECTOR BY FINDING POINTS OF
% INTERSECTION WHERE STAYS ON 1 SIDE FOR MORE THAN FIVE DAYS
% FIRST FIND ALL POINTS OF INTERSECTION OF TRAINING CURVE
TempSave = [];
origflag = 0;
for i = 1 : MAX,
    if (y(1,i) < y2(1,i)) flag = 1;
    else flag = 0;
    end
    if (flag != origflag)
        TempSave = cat(1,TempSave,i);
    end
    origflag = flag;
end
% NOW JUST KEEP THE ONES THAT LAST FOR MORE THAN FIVE DAYS
Points = [];
Points = cat(1,Points,TempSave(1,1));
S = size(TempSave);
for i = 1:(S(1,1)-1),
    if((TempSave(i+1,1) - TempSave(i,1)) > 5)
        Points = cat(1,Points,TempSave(i+1,1));
    end
end
end

```

Fig. A-10: Feature vector extraction.

```

% INITIALIZE VARIABLES
count = 1;
S = size(Points);
% INITIALIZE MATRICES
FVectors = [];
FTarget = [];

for I = 1 : S(1,1),
    FVectors = cat(2,FVectors,NewMat(:, Points(I,1)));
    FTarget = cat(2,FTarget,Target(1,Points(I,1)));
count = count + 1;
end

```

Fig. A-11: Feature vector setup.

```

% NUMBER OF INPUTS IS Num (Number of days to consider)
R = Num;
% GET NUMBER OF NEURONS FOR EACH LAYER
S1 = input('Enter number of neurons in input layer: ');
S2 = input('Enter number of neurons in hidden layer: ');
% NUMBER OF NEURONS IN OUTPUT LAYER IS 1
S3 = 1;

% SET RANDOM WEIGHTS AND BIASES
[W1, B1] = rands(S1, R);
[W2, B2] = rands(S2, S1);
[W3, B3] = rands(S3, S2);

% GET TRAINING PARAMETERS
disp_freq = 100;
max_epoch = 50000;

% GET SIZE OF Fvector & FTarget
Size = floor((MAX-Num)/25);
err_goal = power(0.00001,2) * Size;

% GET LEARNING RATE
lr = input('Enter the learning rate (eq. 0.42, 0.6, 0.8): ');

```



```

% GET LEARNING RATE INCREASE
lri = input('Enter the learning rate increase (eq. 1.04): ');
');
% GET LEARNING RATE DECREASE
lrd = input('Enter the learning rate decrease (eq. 0.6): ');
% GET MOMENTUM
mom = input('Enter the momentum constant (eq. 0.4): ');
% GET MAXIMUM ERROR RATIO
mer = input('Enter the maximum error ratio (eq. 1.04): ');
% START TRAINING
TP = [disp_freq max_epoch err_goal lr lri lrd mom mer];
[W1,B1,W2,B2,W3,B3,epochs,TR] =
trainbpx(W1,B1,'logsig',W2,B2,'logsig',
W3, B3,'logsig', FVectors, FTarget, TP);

```

Fig. A-12; Neural network training.

```

% INITIALIZE ERROR MATRICES
Error = [];
PercentErr = [];
Forecast = [];

% count from Yen.m, -1 because it was updated
for I = 1 : (count-1),
    % EXTRACT EACH COLUMN OF TESTMAT
    TestInput = FVectors(:,I); [O1,O2,O3] =
        simuff(TestInput,W1,B1,'logsig',W2,B2,'logsig',
            W3,B3,'logsig');
    % CALCULATE AVERAGE ERROR
    Forecast = cat(1, Forecast, O3(1,1));
    Error = cat(1, Error, abs(O3(1,1) - FTarget(1,I)));
    PercentErr = cat(1, PercentErr,
        (abs((O3(1,1) - FTarget(1,I)))
            / TestTarget(1,I)) * 100);
end

```

Fig. A-13: Evaluation of the neural network's performance.

```

% GET NUMBER OF TESTING DAYS : 50
Temp = size(forex_Yen);
End = Temp(1,1);
Start = Temp(1,1) - TestDays;

% INITIALIZE TestMat AGAIN
TestMat = [];

% CREATE NEW INPUT MATRIX FOR TESTING
K = 0;
for I = 1 : (TestDays - Num),
    count = Num;
    % Num IS LENGTH OF NEW ROWS
    for J = 1:Num,
        TestMat(I,J) = forex_Yen(Start+K+count,1);
        count = count.- 1;
    end
    K = K + 1;
end
TestMat = TestMat';

```

Fig. A-14: New test matrix setup.

```

% GET ACTUAL VALUES
Actual = ones(1,TestDays-Num);
for I = 1 : (TestDays - Num),
    Actual(1,I) = forex_Yen(Start+I+Num,1);
End

```

Fig. A-15: New target matrix setup.

```

% INITIALIZE ERROR MATRICES
Error = [];
PercentErr = [];
Forecast = [];

for I = 1 : (TestDays - Num),
    % EXTRACT EACH COLUMN OF TESTMAT
    TestInput = TestMat(:,I); [O1,O2,O3] =
        simuff(TestInput,W1,B1,'logsig',W2,B2,'logsig',
            W3,B3,'logsig');
    % CALCULATE AVERAGE ERROR
    Forecast = cat(1, Forecast, O3(1,1));
    Error = cat(1, Error, abs(O3(1,1) - Actual(1,I)));
    PercentErr = cat(1, PercentErr,
(abs((O3(1,1) - Actual(1,I))
        / Actual(1,I)) * 100));
end

```

Fig. A-16: Calculation of average percentage error for neural network.

```

[W1,B1,W2,B2,W3,B3,epochs,TR] =... trainbpx(W1,B1,'logsig',
W2,B2,'logsig',
W3,B3,'logsig',FVectors,FTarget,TP);

```

Fig. A-17: Continuation of training