Trading Financial Markets With

Neural Networks and Genetic Algorithms

Neuroquant.com, May, 2001

"The **stock** market is a game. There aren't many rules. You keep score with money."

– an anonymous quote often falsely attributed to Bill Gates

Introduction

The reason people are so good at performing a wide range of tasks is that they have the ability to adapt to constantly changing conditions. The reason so many mechanical systems for trading the **stock** market are so bad at what they do is that they lack the ability to adapt to constantly changing conditions. The merger of biological-like adaptability with mechanical trading systems for the **stock** market by the use of artificial **neural** networks is the topic of this discussion, which will review technical analysis, **neural** networks, and their use together in financial trading.

Basics of technical analysis

Technical analysis is the process of analyzing a security's historical performance in an effort to determine probable future performance. There are two basic independent variables relating to the performance of a **stock**: price and volume. Technical analysis may involve the study of charts, indicators, or systems. A primary use of technical analysis is to time the market, that is, to generate signals as to when to buy and sell, or cover and short when trading both sides of the market. One of the most basic questions technical analysis attempts to answer is whether the market is going up or going down.

Figure 1- Some Basic Indicators: Price Chart With Moving Averages, Stochastics, and Directional Movement Index

A Few Basic Tools

Moving averages

Technical analysts have many tools at their disposal. One group of tools attempts to smooth price data to eliminate minor price variations which do not affect the trend under consideration. The simplest of these tools is the moving average, which averages price over a number of bars. For example, a 10-day simple moving average of the daily close is calculated by summing the closing price over the past ten days and dividing by ten. There are many variants of the moving average which weight the period unequally, such as the exponential moving average, which more heavily weights the more recent values, the triangular moving average, which weights the middle portion of the period more heavily, the volume weighted moving average, which weights bars with larger volume more heavily, and numerous variable moving averages, which alter weighting and period length according to market volatility and other factors.

The moving average represents a smoothed derivation of the price. By virtue of its method of calculation, a moving average lags the price. Notice in Figure 1 how both simple moving averages continue to trend upward after the price turns downward. The longer the period chosen, the greater the lag, and the smoother the resulting curve. Abrupt market moves are particularly hard for moving averages to accurately track.

The moving average may be used in a number of ways. The simplest use is to note its direction. Another method is to plot the price and moving average together, using crossovers of the price and moving average for buy and sell signals. For example, a buy signal would be generated when the price moved above the moving average line. Moving averages of different periods may be used, with signals generated when the moving average lines cross each other. A buy signal would be generated when a shorter period moving average line crossed above a longer period moving average line. The moving average convergence divergence (MACD) indicator uses a moving average of the difference between two moving averages to generate its signals.

Moving averages work well in trending markets, but subject the trader to whipsaws when markets are not trending. A whipsaw is the painful experience of buying just as a market turns down, and vice versa. Whipsaws are a product of the lag moving averages introduce. The selection of the period of the moving average is critical. If one is unfortunate to select a suboptimal moving average period, buys may be generated just as the market turns down and sells generated just as the market turns up. Trading capital will evaporate rather quickly.

Oscillators

In contrast to moving averages, which work well in trending markets, oscillators work well in trendless, range-bound markets. While moving averages are unbounded, and can range from zero to any value, oscillators are bounded, with values usually normalized to the range 0 to 100. One of the best-known oscillators is the stochastic oscillator, a neutrally named tool introduced by George Lane.

Stochastics are based on the premise that a value moving up tends to close near the top of its range, and vice versa. The stochastic value assesses how close the close was to the high for the period over a number of periods. If a **stock** were to close at its daily high for each period of the total number of periods considered, the stochastic value would be 100. There are several derivations (%K, %D), based on differing periods of evaluation and various smoothing operations of the basic indicator.

Stochastics are typically used to identify overbought and oversold conditions. For example, a value of over 80 may be considered overbought and a value of under 20 oversold. In typical systems, when the value crosses into overbought range and then out of this range, a sell signal is generated. A buy signal is generated when the stochastic value crosses into and then out of oversold territory.

Stochastics can be effective within a trading range, but perform poorly in a strongly trending market. There is nothing to prevent a **stock** from remaining overbought or oversold for a long period of time, and a buy or sell based solely on stochastics can remain of the wrong side of the market for a protracted period. As such, stochastics are rarely used as a sole indicator.

Directional Movement Index

Moving average systems work well in trending markets, while stochastics work better in range bound markets. The directional movement index (DMI), introduced by Welles Wilder, helps to determine whether a market is trending or not. DMI has three components. ADX measures the strength of a trend, without specifying its direction. A value of 30 is a commonly used cutoff, above which a strong trend exists. DI+ and DI- specify the strength of the uptrend and downtrend respectively, and are used to specify the direction of the trend. In Figure 1, notice how sharp price movements are accompanied by increases in the ADX value and divergence in DI+ and DI-

Fourier Analysis

Moving averages, stochastics, and the directional movement index barely scratch the surface of the tools available to the technical analyst, and yet are enough to get started with basic technical analysis. One tool, that is mathematically more complex than the first three tools discussed, is Fourier analysis. Fourier analysis is a less traditional technical analysis tool borrowed from the realm of signal processing. It is based on the premise that a complex but periodic waveform, which the chart of a **stock** price may or may not be, can be decomposed into the sum of a number of sine waves of different frequency and amplitude. Likewise, the original waveform can be reconstructed by summing its component waveforms.

The power of Fourier analysis in our application lies in what can be done with the component waveforms before the original waveform is reconstructed. If one wishes to smooth a price chart by eliminating noise, one can simply omit component sine waves of a higher frequency than we are interested in. If we are trying to identify the dominant cycle length of the market, for example in trying to choose the proper length for a moving average, we can observe the frequency of component waveforms of higher amplitude.

Fourier analysis of **stock** charts suffers from several limitations. The algorithm used in most cases, the fast Fourier transform, is based on the assumption that the waveform is periodic, and that we are analyzing a full period. Neither of these assumptions may be true in a **stock** chart. Nonetheless, several mathematical workarounds are available, and this can be a powerful tool in our arsenal.

Systems and System Testing

As opposed to indicators, such as moving averages and stochastics, trading systems are more complete. They include rules on when to buy and sell, which may involve a combination of indicators. They may also include money management rules, which guide how much is risked on each trade. They may be simple, such as “risk 10% of capital on each trade”, or complex, such as the optimal-f system, which bases trade size on prior results.

One major advantage of mechanical trading systems, with firm rules that call for no trader input, is that they isolate the trader from market hysteria, whether positive or negative. On the other hand, discretionary systems, which call for the trader’s discretion, are subject to distortion based on the emotions of the trader. Such hysteria tends to be most misdirected at market extremes.

Another major advantage is that mechanical systems can be subjected to rigorous testing over different time periods and with different instruments. Testing using data which may have itself been used to develop the system is referred to as in-sample testing, while testing with data the system has never seen is referred to as forward-testing or out-of-sample testing. A system must perform well with both in-sample and out-of-sample data to be considered valid. Failure to adhere to this simple rule was responsible for much of the early disillusionment with **neural** networks among traders in the early 1980s.

Another advantage of mechanical trading systems is that they can be easily optimized. Optimization refers to the process of selecting parameters and values that maximize the yield, minimize drawdown (maximum excursion from equity high to equity low), or otherwise make the system perform as desired. The danger in optimization is that one ends up merely curve fitting the system to known data, rather than developing a robust system that can operate on data it hasn’t seen before

It is therefore crucial that a system not only be tested on data used to help optimize it, but also on data it has never seen before (forward testing.) This is typically done by holding out a portion of the available data to use for testing after optimization is complete. Once the system is verified to work on this previously unseen data, it may then be used to further optimize the system. This process is referred to as walk forward testing.

The value of technical analysis and mechanical systems trading

When I spoke at TCF in May of 2000, the concept of technical analysis and market timing was a tough sell. The Nasdaq has just completed a meteoric 100% rise over the past year, and there were few market timers who outperformed the broad market. As I write this paper in March of 2001, the Nasdaq has given back all of its 1999-2000 100% increase and more. Many market timers have avoided these losses. Interest in technical analysis and the techniques of market timing, which it underpins, seems to increase at times when “buy and hold” isn’t working very well.

Technical analysis and market timing have many skeptics and detractors, including most spokespeople for major brokerage firms. Typical arguments state that the techniques are too complicated, or are over optimized to fit particular market conditions. The following system serves to disprove this premise by counterexample.

The Four Percent Swing System

Marty Zweig popularized the percent swing system in a 1986 book entitled *Winning on Wallstreet*. The system logic is very simple. Originally based on weekly closes for the Value Line Composite index, the system buys after a four percent rise above a prior weekly low and sells after a four percent decline from a prior weekly high. The system can be followed on a chart without a computer. The system has been profitable during a 19-year backtest period from 1966 to 1985, prior to its publication, and has remained profitable since its publication.

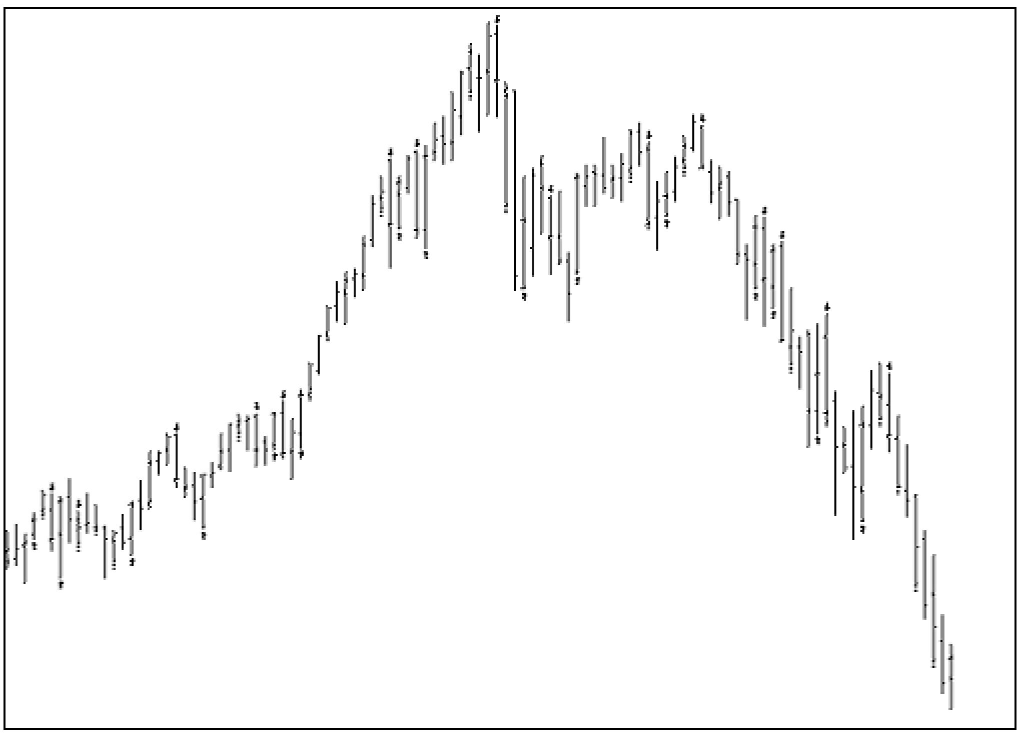

Figure 2 - 4% Swing System NASDAQ 100 Trust (QQQ) 1999-Mar 2001,Weekly Bars; Buys Indicated By Up Arrows Below Price Bars, Sells By Down Arrows Above Price Bars

The system meets criteria for traditional technical analysis systems of broad applicability and robustness. First, it works well across different instruments, such as stocks, market indices, and futures. Second, it works well across different time frames such as hour, day, and week bars. Finally, its parameters, the percent upswing necessary to trigger a buy and the percent downswing necessary to trigger a sell, are not particularly critical. The system works with a wide range of values, and is therefore robust. Although its gains are not spectacular, it has beaten market averages in both back tests and forward tests since its introduction.

The four percent swing system logic guarantees that a significant move in either direction will not be missed. While adequate to demonstrate that technical analysis has merits, the percent swing system also

demonstrates the weaknesses of simple non-adaptable systems with fixed parameters. Although its logic is good at capturing trending market moves, it does poorly in a range bound market. It subjects its adherents to whipsaws, in which signals are triggered just as a minor move is completed and the market reverses to the opposite direction. A more adaptable system might be able to recognize trending from trendless markets, and only fire signals when a trending market was detected. Alternately, parameters could be dynamically adjusted to better fit current market conditions.

The tradeoff for robustness and general applicability in the four percent swing system is somewhat lackluster performance. Although the system beats market averages over long periods of time, it suffers from poor performance during trendless market periods. When average market swings are less than 4%, the system doesn't trigger. When swings are between 4% and 8%, the system triggers and loses money with each trade. Although the 1999-2001 market showed many strong trends, the markets on average do not trend strongly. It's up to periods of strong trend to make up for the losses suffered in trendless markets.

The weaknesses of the four percent swing system, mainly its lack of adaptability to changing market conditions, can be ameliorated in several ways. Rules could be constructed to attempt to identify whether the instrument was trending or not, and apply the four percent swing only during strong trends. During range-bound conditions, other trading systems could be used, or the trader could simply remain flat (out of the market.)

One problem is that with the introduction of new rules, new fixed parameters must be established, which require testing, optimization, and eventually generalization. It would be nice to have a way to decide what rules and parameters work best without exhaustive rote testing.

Basics of artificial **neural** networks

Before being able to discuss the application of artificial **neural** networks to trading systems, a brief introduction or review of **neural** networks will be undertaken.

Modern computers operate at speeds such that the speed of light and the time it takes for electrons to travel between portions of a microprocessor die are of great concern. The wavelength of the light used to etch chips is of concern because it affects the minimum thickness that a line can be etched. By contrast, biological **neural** networks work at a rather leisurely rate, where events are measured in milliseconds, and propagate their nerve impulses at speeds measured in meters per second.

Biological **neural** networks, such as the brain, make up for their lack of speed by sheer numbers. Of the approximately 1 trillion cells in the adult human body, 100 billion are neurons. Working in concert, they are able to effect feats of reasoning, pattern recognition, and deduction that far exceed capabilities of modern computers. These 100 billion neurons make almost 100 trillion connections among themselves. The human brain stores knowledge in the existence and strength of connections with other neurons, in ways that are only poorly understood.

Artificial **neural** networks (ANN) attempt to reproduce the abilities of biological **neural** networks by simulating neuronal architecture in either hardware of software. ANNs learn by manipulating the existence and strength of connections with other neurons. While there are an infinite number of ANN architectures possible, the feed forward back propagation ANN is one of the most frequently used architectures.

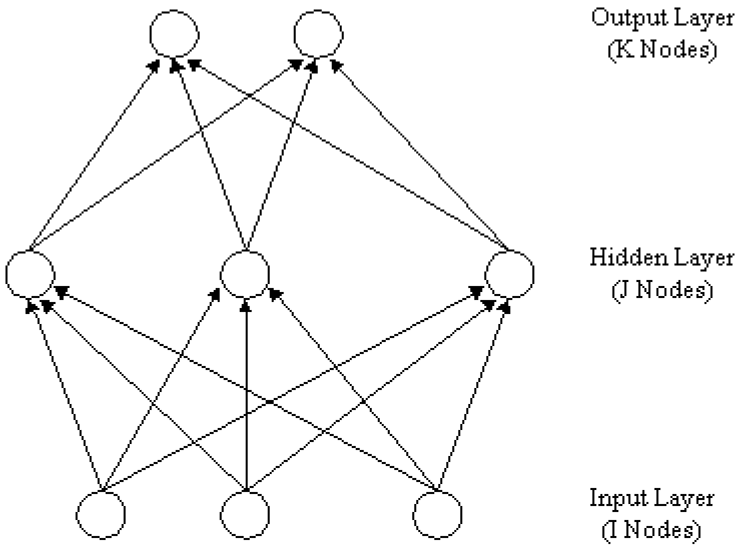

Figure 3 Schematic Diagram of a Feed Forward Back Propagation **Neural Network**

In the above diagram, I nodes represent inputs to the **neural network**. Every input node connects to every hidden node (fully connected **network**), with its contribution to each hidden node affected by its weight, with a unique weight given to each line. A node function calculates the output of each hidden node, which serves as an input to the output layer. Again, every hidden node connects to every output node and is influenced by a weight assigned to each connecting line. A node function calculates the final output value for each output node based on the weighted inputs from the hidden layer.

The number of input nodes is generally fixed by the number of inputs one selects, although the optimization process can trim this number. The number of hidden nodes is variable and can be altered during the training process. The number of output nodes is fixed by the design of the **network**. A typical **neural network** for financial trading has two or three outputs (buy/hold/sell).

The process of training a **neural network** to produce the desired output takes place in a number of iterations, or epochs, and ultimately is accomplished by adjusting the weights of connections between neurons. Initially, the weights are set to random values. The **network** is shown its inputs and outputs are generated. The difference between the actual and desired outputs in then propagated back through the **network** in an effort to adjust connection weights. The process is continued until the error is acceptable or no further improvement occurs. The reader is referred elsewhere for a more complete and mathematical discussion of this process (Reference: Zirilli).

Genetic Optimization

In this specific application, optimization refers to the selection of inputs and perhaps parameters of inputs that best help to describe to the **neural network** the output desired. Optimization could be implemented as a brute-force series of nested loops, trying all possible values for selection of inputs and their parameters. However, given the length of time it takes to calculate a single **neural network**, this approach is usually time-prohibitive.

Genetic algorithms (GA) are a more elegant technique, frequently used in combination with **neural** networks. Like **neural** networks, genetic optimization is based on a natural model, namely the process of exchange and recombination of genetic material occurring during sexual reproduction. GA construct genes and chromosomes based on possible solutions to a problem, and through a process of recombination, mutation, and fitness testing, attempt to arrive at a more optimal solution. The reader is referred to other sources for more detail (Reference: Zirilli).

Putting It All Together

The following examples use equity stocks and exchange traded funds, although the techniques are generally applicable to mutual funds, and with some caveats, futures and equity options. The techniques presented use daily or weekly data and generate trades in the intermediate timeframe, between days and weeks in length. The inputs to each **network** are discussed, along with the trading results. These techniques also trade both sides on the market, long and short. If one has confidence in buy and sell signals, there is no reason to limit trading to **B**uying **L**ong **O**nly **W**ithout **S**horting (the bold letters summarize my view of this artificial trading limitation.) It makes as much sense as only trading on odd days or odd months. It either case, you are bound to leave money on the table that you are otherwise entitled to.

Example 1 – A Simple Moving Average System

The first example will use only moving averages as inputs to the **neural** networks. Buy and sell side decisions use a single identical rule each. The rule is: trigger on SMA1>AMA>SMA2. SMA1 and SMA2 represent simple moving averages of unspecified length. AMA represents an adaptive moving average, which is calculated in advance by a separate **neural network**, and is used as a smoothed proxy for the price of the instrument under study. A genetic optimizer selects the length of each moving average for SMA1 and SMA2. A different value may be selected for each side of the market (buy and sell.) No exit rules are specified; the system is always in the market, on either the long or short side, and the exit signal for a trade is a trade triggered in the opposite direction. Weekly bars were employed, although the system works well with daily bars as well.

Two stocks with considerably different performance were studied. The first, Human Genome Sciences (HGSI), was a remarkably volatile performer, paralleling the performance of the NASDAQ over the past several years, but magnifying its gains and losses. Over the 2.5-year training and optimization period, the system performed well, as we would expect. Seven trades are triggered, four on the long side, and three on the short side. All trades in this period are profitable. The **stock** itself appreciated 230% annualized during the training period, corresponding to the huge Nasdaq run-up during this period. The system increased this gain to 326% annualized, with most of the gain coming from long trades. However, note that these in-sample testing figures are largely irrelevant, as they employ data that one would not have in real-time trading. They serve only to show the system has the potential to produce profits in out-of-sample testing.

Out-of-sample testing refers to testing on data the system has not trained or optimized on. The system parameters are frozen, and the system is then presented data is hasn't seen before. This represents a truer measure of the system's performance in real-time trading. During the one-year period ending 3/23/01, the system gained 45.8%, while the **stock** lost 11.7%. Three of four trades were profitable, with a gross profit/loss ratio of 4.43. Clearly the use of this system for this **stock** would have improved real-time results over buy-and-hold.

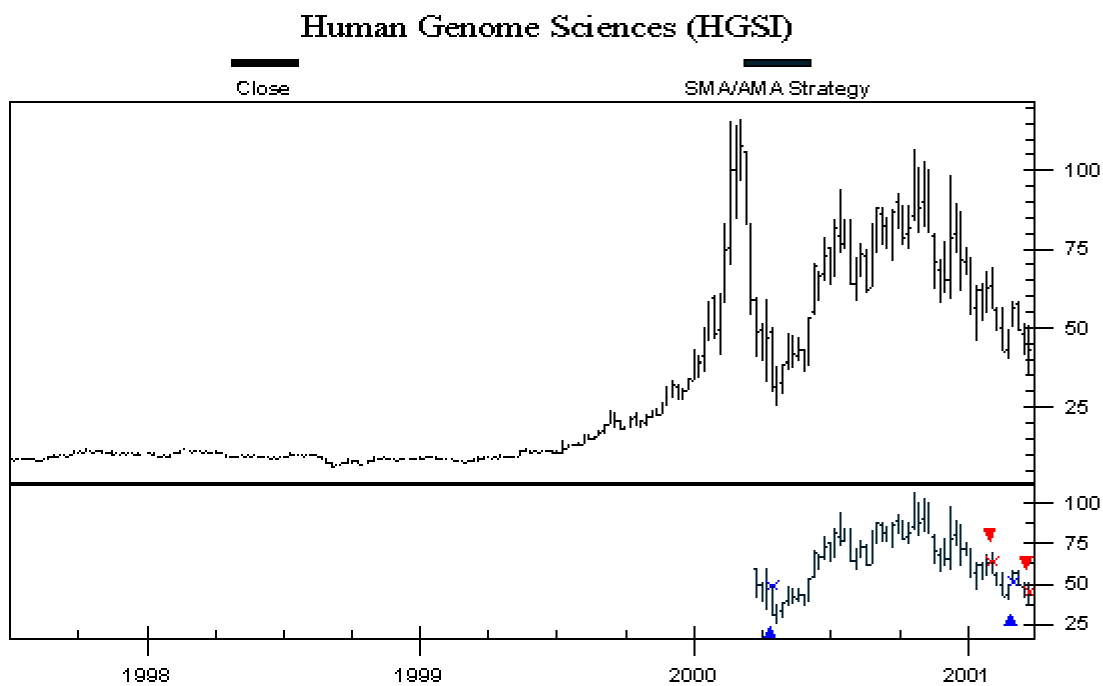

Figure 4 HGSI traded by SMA/AMA System; bottom graph shows out-of-sample testing, with buys indicated by up arrows and sells by down arrows

In contrast to the breathtaking volatility of HGSI, Boise Cascade (BCC) showed far less variability during the past several years. During the testing and optimization phase, the **stock** declined 7.3% on an annual basis, while the SMA/AMA system trading BCC showed a 78.5% annual return, with six winning and one losing trade. During the one year out-of-sample testing period ending 3/23/01, the **stock** lost 7.3% on an annualized basis, while the SMA/AMA system gained 37.7% annualized, with four winning and two losing trades. Furthermore, the gross profit/loss ratio was 7.63. Drawdowns were minimal. This would be a particular easy system to trade, with only 6 trades over a year, each triggered on Friday's close and executed on Monday's open, amid a backdrop of limited volatility.

How were the two stocks, HGSI and BCC, chosen for analysis? If a large number of stocks were tested and only the stellar performers chosen for display, the results are not valid, as this is a form of in-sample testing. In this case, I have followed these stocks for years, and knew one to be a volatile performer, while the other was rather staid. They are the only two stocks I tested with the SMA/AMA system, and the results are presented as they came off the system.

Example 2 – The Three Cross System

In this example, buy and sell signals are generated with three inputs each to an optimized **neural network**. The three signals are all crossover indicators. In the first, two moving averages of different lengths generate a signal when they cross each other on a chart. In the second, two stochastic indicators, %K and %D, generate signals when they cross. In the third signal, two commodity channel indices (CCI) of different time periods generate signals when they cross. The commodity channel index provides a measurement of the deviation of the current mean price from the average mean price relative to the deviation of all the previous n mean prices from the same mean price average. As opposed to the first example, which used an identical rule for the short and long side, this system uses complimentary rules for each side of the market.

The system performed well when tested with BCC and HGSI. BCC returned 55.1% annualized during a period when the price of the **stock** declined 2.5%. 80% of trades were profitable. HGSI returned 757% annualized during a period when the **stock** increased 175% annualized. However, the majority of the profit was concentrated in just a few trades making it somewhat unsuitable for implementation.

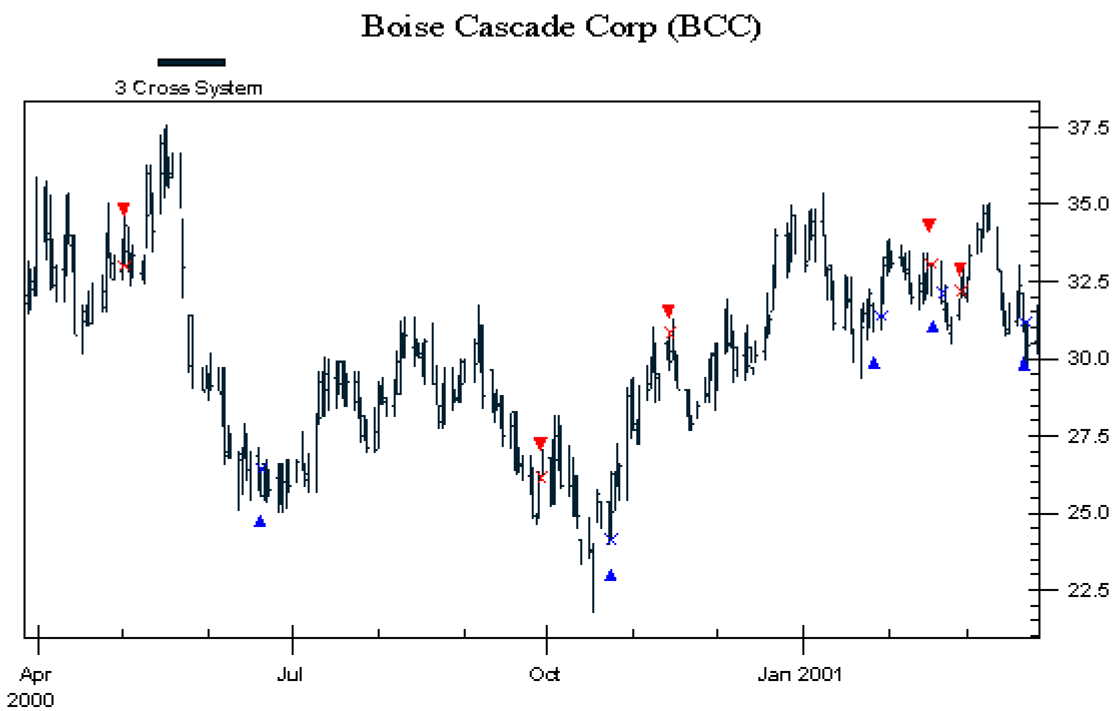

Figure 5 Boise Cascade (BCC) timed with the Three Cross System

Example 3 – Opening a black-box system

A final example uses **neural** networks to analyze the workings of a commercial **stock**, mutual fund, and futures system that sells for in excess of $6,000 for a permanent license. The system, which I will not specifically identify, it heavily advertised and said to be applicable to all financial instruments in all time frames. It is a black-box system, in that the underlying rules are not disclosed.

In observing sample charts from this system, the signals generated looked vaguely familiar. Although the system generates discrete buy, neutral, or sell signals without numeric values, the chart showed some characteristics of a stochastic oscillator.

One approach might be to disassemble the software and reverse engineer the algorithm. This approach was not used for this study. In the first place, I don't own the software. I only had a number of charts posted on an Internet mailing list and web site of the system in action. Second, although I don't know specifically in this case, most software licenses prohibit this. Finally, such an effort is simply beyond my technical ability.

Turning to tasks within my ability, I looked at the ability of **neural** networks with genetic optimization to fit a set of rules to a desired output. In this case, rather than assigning the **stock** price some time in the future as the desired output, the value of the indicator under study was assigned. Then, a number of different indicators, including stochastic oscillators, were offered as potential inputs to the **network**.

The system selected a single input as adequate to describe the **network** output, in this case the indicator under study. With some minor additional manipulation, the two rules best describing the system are:

BUY when 9 period stochastic %K is greater than 65

SELL when 9 period stochastic %K is less than 35

That's it. An example of this system vs. indicator X is shown below. It should be pointed out that this derived system has not been extensively tested over a wide range of instruments and time frames to assure its performance. The commercial system also includes stops, which have not been duplicated here. Given the simplicity of the underlying system, it is likely the stop generation algorithm is similarly simple in design. It is hard to imagine what the commercial vendor is actually charging for, except perhaps extensive marketing efforts.

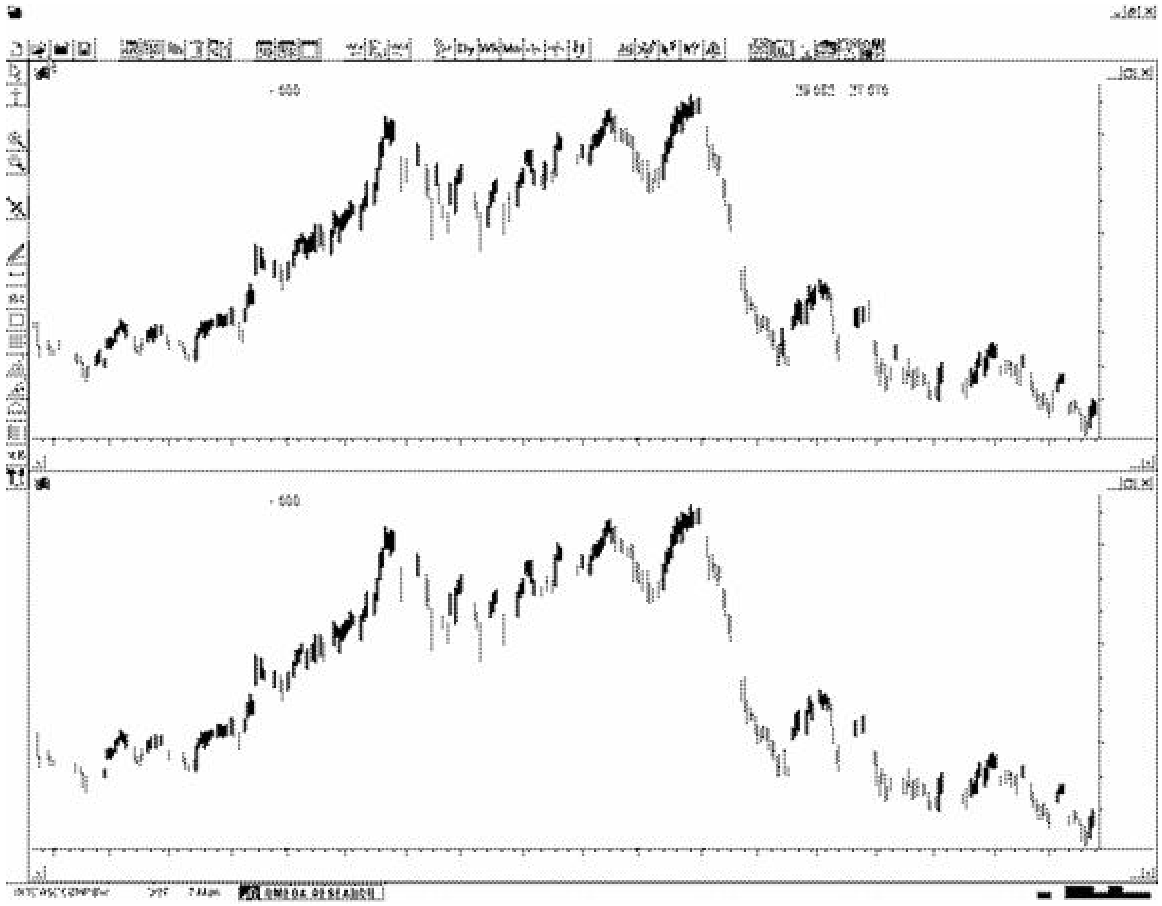

Figure 6 System X above vs. the derived system below - in each graph, buys are indicated by thin bars, and sells by broad bars – note that only a few bars differ

Conclusion

Timing the market increases profit potential by avoiding the majority of price action opposite in direction to positions held. Mechanical trading systems offer a number of advantages over discretionary systems, including but not limited to, the exclusion of emotion from trading decisions and the ability to extensively backtest potential trading systems. **Neural** networks with genetic optimization offer a tool that can produce mechanical trading systems that are adaptable to changing market conditions. Successful implementation of such a system requires an understanding of the strengths and limitations of artificial **neural** networks, as well as extensive testing.

Author's note

By necessity of the length limitation for this discussion, many of the topics here have been presented in an extremely abbreviated fashion. The reader is referred to many excellent monographs on this and related topics available in references below and on numerous websites. A search with the words "**stock neural network**" will get you started. I would be interested to know what comes of your research

References

Acheli, Steven B. *Technical Analysis from A to Z.* **http://www.equis.com/free/taaz/**

Arnold, Curtis M. Timing The Market. Weiss Research, Inc., 1993.

Fosback, Norman G. ***Stock*** *Market Logic*. Dearborn Financial Publishing, Inc., 1991.

Zirilli, Joseph S. *Financial Prediction Using* ***Neural*** *Networks*. International Thomson Computer Press, 1997.

Zweig, Martin. *Winning On Wall Street*. Warner Books, 1986.