01 Jan 1991

A Commodity Trading Model Based on a Neural Network-Expert System Hybrid

K. Bergerson

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Recommended Citation

K. Bergerson and D. C. Wunsch, "A Commodity Trading Model Based on a Neural Network-Expert System Hybrid," *Proceedings of the International Joint Conference on Neural Networks, 1991., IJCNN-91-Seattle*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1991.
The definitive version is available at https://doi.org/10.1109/IJCNN.1991.155192

A Commodity Trading Model based on a Neural Network-Expert System Hybrid

Karl Bergerson
Donald C. Wunsch II

Neural Trading Company
8045 20th Ave N.W.
Seattle, WA 98117-4336

Learning capability is clearly a desirable quality in a software-based approach to commodity trading systems. A system that can learn can make use of the tremendous amount of data available[1], allowing semi-automatic generation of a good model. This capability becomes even more powerful in conjunction with proven techniques for profitable market analysis. For example, techniques such as rule-based futures trading with automated risk management can be considerably improved by learning-based approaches. Fortunately, the emergence of neural network technology has brought this capability into the mainstream of available software tools.[2] We demonstrate a system[3] that combines a neural network approach with an expert system to provide superior performance compared to either approach alone. *The key to this is to use each system for what it does well; pattern recognition by the neural network, and application of money management rules by the expert system.*

Neural networks apply to problems that have proven difficult if not impossible to solve by programming a computer with algorithms. Problems, such as predicting the market, fall into this category because solving them involves the recognition of patterns, even patterns that are vaguely defined, buried in noise, or are otherwise difficult to decompose into the neat steps of an algorithm.

It is not necessary to solve the mysteries of human intelligence to make a useful system. Even the smallest insects have pattern processing, learning, and other capabilities that elude even the most powerful supercomputers. It is of little surprize then, that some insights in the field of neural networks have led to impressive gains in the capabilities of software-based trading.

Like several others who have tried market prediction with neural networks, we have used the back-propogation network[4]. with some parameters selected experimentally that will be discussed below. However, we expended the majority of our efforts in providing good training data, which we believe sets our work apart. We wanted to control precisely what the network learned about market prediction, preferring that it only attempt to make a trade when the chances for profit were high. Therefore it was not desirable to train the network purely on historical data, expecting it to predict gains and losses under all possible conditions. Instead, we used a human expert to implicitly define patterns, using hindsight, that an intelligent system might have been able to use for an accurate prediction. Desired outputs were found by a combination of observing the behavior of technical indices that normally precede a certain kind of market behavior,

0-7803-0164-1/91/0000-0289$01.00©1991 IEEE

and by observing the actual market behavior in retrospect. Thus, the network learned to give signals based on data that looked favorable to a human expert, but tempered by the requirement that anything considered to be a "good example" must also be accompanied by a profitable history. The network also received a number of "bad examples", that is, examples where the indices looked good or borderline good, but were not borne out by historical data and therefore deserved an output indicating an unfavorable condition.

In contrast to merely pumping a neural network with massive amounts of historical data, our method was extremely labor intensive. Figures 1a, 1b, 1c & 1d show four technical indicators plotted against the daily Standard & Poors 500 Index (S&P 500). The boxes show that the human expert felt that the patterns were clear enough to indicate that a "sell" decision could be made, and the triangles are where the human expert chose a "buy" decision. These points, chosen manually, were given as training examples to the network, together with technical data from the recently preceding days. The process demanded many hours of expert time, and careful consideration of each potential pattern in the data. These figures cover the period of Sept 19, 1980 to Jan 2, 1981. The triangles and boxes were chosen manually by the expert, and took him about 3 hours to do this example. The total amount of data used for training covered approximately 9 years.

The selection of parameters for the neural architecture also involved some extra effort. Two issues that were addressed experimentally were the number of hidden units to use, and the amount of training to provide. The number of hidden units was determined by pruning. We began with a full layer of 54 hidden units, the same as the number of input units. After training had stabilized, we removed those units whose weights were smallest, and retrained. In this way we corrected for any possibility of overfitting the training data. The amount of training to provide was also monitored. We did this by comparing the error on the test data to the error on the training data. When these were the same, we did no further training. Note, however, that the test data we used for the performance figures that we show includes data that were never seen by the network, even in this implicit manner.

It is very important to note that this hybrid approach offers strong advantages over either rule-based or unaided neural network approaches. Rule-based approaches are lacking in the flexibility to easily deal with the recognition of poorly defined patterns. Unaided neural networks are better at pattern recognition (in a theoretical sense) than they are at doing things that are naturally well-handled by rules, such as risk management. It is possible to make a theoretically excellent market prediction system using neural networks alone, but it is the combination of this capability with a rule-based system that makes a useful real-world investment system. Our system uses a risk-management rule that governs where stop loss points are put to control losses when an incorrect prediction is made. Furthermore, these stops need to be increased when a trade goes well, so that one knows when to take profits. Also, certain extreme values of indicators are known to be a sign of extreme volatility in the market, making predictions more uncertain. These are best tracked by rules. The rule-based system thus has veto power over the neural networks' signals, but does not generate signals on its own. It is the synergy of the rule-based and neural system that permits the design of such an attractive reward to risk ratio trading model. The reward to risk ratio for our system's performance to date is shown in Table 1.

Figure 2 shows the results of a rule-based daily trading system that has been augmented by a neural network market predictor[3]. The neural network was trained on data from 1980 to 1988, and then ran with an initial investment of $10,000 on January 4, 1989 to January 25, 1991. The final account value was $76,034, which represents a growth of 660% over 25 months. The maximum drawdown was for the period from Sept 15, 1989 to Sept 25, 1989. During this period, the account went from a value of $32,954 to $32,187, a 2.3% loss. The program easily recovered from this in a single successful trade. The reason for this resilient property is a conservative risk management rule that limits the amount of losses that will be tolerated but allows maximal advantage of profit-making opportunities. It should be noted that these are theoretical gains, although we have been trading the system successfully with real dollars since August 1990.

Our point that learning can enhance the performance of trading systems is now clear. This enhancement is beyond that attainable with rule-based or neural network systems alone. The key issue is to move beyond mere theoretical prediction to profitability. As figure 2 makes clear, that move has been made. The key technical insight that led to this achievement is that the neural network can be used as a knowledge acquisition tool, and when that tool is used with some real world risk management expertise, the result is impressive. It is not a magic solution, in fact, it involved more hard work and more demands on the expert's time than traditional knowledge engineering approaches. The results, though, seem to justify the difficulty of the approach.

References:

1. Lowry for interday data, Technical Tools for intraday data
2. BrainMaker Professional from California Scientific Software, Grass Valley, CA
3. Neural$ from Neural Trading Company
4. Rumelhart, D. E., G. E. Hinton & J. R. Williams, "Learning Internal Representations by Error Propogation", in Parallel Distributed Processing, Vol. 1, Chapter 8, MIT Press, Cambridge 1986

Table 1

	LONG	SHORT	ALL
WIN RATIO %	58.33	91.67	75.00
REWARD/RISK	5.82	3.81	4.81
NUMBER OF TRADES	12.00	12.00	24.00
AVG GAIN PTS	6.10	4.06	5.08
MAX LOSS PTS	1.54	1.39	1.54
AVG LOSS PTS	0.45	0.12	0.28
AVG DURATION	9.42	2.17	5.79
MAX DRAWDOWN	1.04	0.90	1.04
AVG DRAWDOWN	0.32	0.16	0.24

Report dates: 890104 thru 910129, Slippage: 0.50 [$250]

Figure 1a [Solid line = S&P 500] Moving Average of Price - Normal

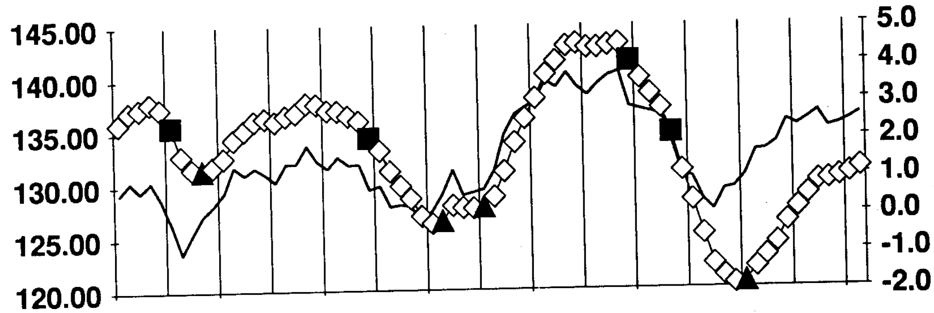

Figure 1b Moving Average of Price - Fast

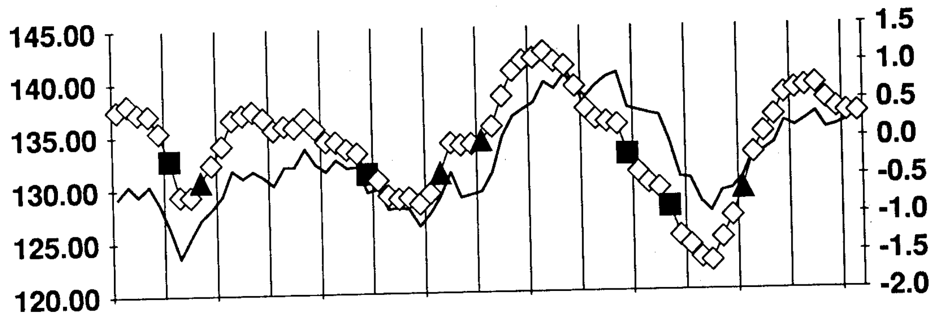

Figure 1c Moving Average of Advance Decline Line - Normal

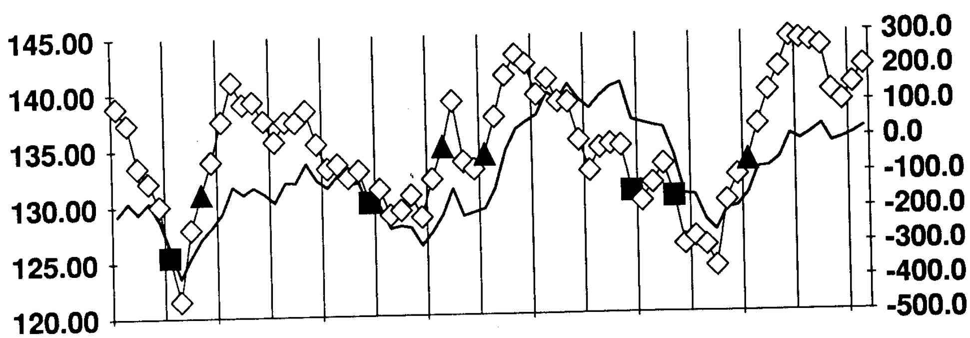

Figure 1d Moving Average of Advance Decline Line - Fast