Commodity Futures Trading Performance Using Neural Network Models versus ARIMA Models

CHRISPIN NTUNGO
MILTON BOYD

Neural networks trading returns are compared out-of-sample with traditional ARIMA returns for corn, silver, and deutsche mark. Results show that neural network and ARIMA models had positive returns, and at about the same levels. However, deutsche mark was less profitable and returns were not statistically different from zero, in contrast to corn and silver. © 1998 John Wiley & Sons, Inc. Jrl Fut Mark 18: 965–983, 1998

INTRODUCTION

Neural networks are computing structures modeled on the structure of the brain (Hecht-Nielsen, 1990). They are nonlinear methods for pattern recognition, classification, and prediction. Applications in finance have included mortgage risk assessment, economic prediction, risk rating of exchange-traded fixed-income investments, and portfolio selection and diversification.

The purpose of this article is to test further the application of the neural network as a commodity trading method. A traditional linear

*Correspondence author, Faculty of Agricultural and Food Sciences, Department of Agricultural Economics, The University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2.

- *Chrispin Ntungo is a former graduate student at the University of Manitoba in Winnipeg, Canada.*
- *Milton Boyd is a professor at the University of Manitoba in Winnipeg, Canada.*

The Journal of Futures Markets, Vol. 18, No. 8, 965–983 (1998)
© 1998 by John Wiley & Sons, Inc.
CCC 0270-7314/98/080965-19

ARIMA model is used as a benchmark for comparison with the neural network. This article first examines past research, then the structure of neural networks. This is followed by a discussion of forecast and trading procedures. The neural network trading results and an evaluation are presented and then compared with the ARIMA model. This is followed by a summary of the research results.

PAST RESEARCH

Earlier research by Halquist and Schmoll (1989) used a neural network model to predict trends in the S&P 500 index. They found that the model was able to predict trends 61% of the time. This was followed by Trippi and DeSieno (1992), who compared a neural network based trading strategy in S&P 500 Index futures with passive buy and hold strategy. They found that the neural network model strongly outperformed buy and hold strategy by as high as 228%, after inclusion of brokerage charges. Whereas neural networks are relatively new methods for commodity trading, some researchers have already tested neural network trading models for various markets. Grudnitski and Osburn (1993) found neural networks to be profitable for S&P 500 and gold, averaging annual returns of 17% and 16% per year.

Recent researchers argue that price generating processes for most assets exhibit nonlinear dynamics. Neftci (1991) argues that if price dynamics are nonlinear, technical analysis may be capturing information contained in higher-order moments of asset prices. Such information may not be captured by traditional linear models. Lukac and Brorsen (1990) show that twenty-two of the twenty-three technical trading systems simulations conducted in thirty different markets produce significant net returns. Also, Boyd and Brorsen (1991) showed that five technical trading systems tested on seven commodities yielded significant net returns. An examination of stock returns generated from some technical trading rules by Brock, Lakonishok, and LeBaron (1992) also appear to suggest that the return generating process of stocks is more complicated than that suggested by linear models. The preceding discussion suggests that the price adjustment process may be nonlinear. If the underlying commodity returns generating process is nonlinear, then a nonlinear technique such as a neural network may be useful for commodity trading.

STRUCTURE OF NEURAL NETWORKS

A neural network consists of a collection of input units and processing units, called neurons, that are arrayed in several layers. Each neuron at the input layer receives the data, multiplies them by a connection strength called a "weight" and sends them to the neurons in the next layer. The processing units immediately after the input layer are self-sufficient when processing the information they receive. Processing in each individual neuron takes place in parallel with other units according to a transfer function. If there is more than one layer, then the results from each neuron are sent to all neurons in the next layer. The relationship between each neuron and other neurons in the next layer is called "feed-forward." Neural networks are different from computers in the sense that they "learn" to solve problems. The network compares the results of its analysis with the desired output. The discrepancies between the desired output and that of network error are then minimized up to a tolerance level determined by the researchers.

Neural networks are different according to their "learning laws incorporated in their transfer functions, the topology of their connections, and the weights assigned to their connection" (Hecht-Nielsen, 1990). One of the most widely used networks is the "back-propagation" (BP) network. This network has been popular due to its success in solving practical problems. Hornik et al. (1989) rigorously showed that the standard BP networks using an arbitrary transfer function are capable of approximating any measurable function in a very precise and satisfactory sense, provided that sufficient number of hidden units are used. Hecht-Nielsen (1989) has also shown that a three layer BP is capable of approximating any arbitrary continuous mapping. More specifics of neural network procedures and backpropagation, from Kaastra and Boyd (1995), can be found in the Neural Network Appendix.

PROCEDURE AND DATA

The neural network design procedure used is similar to Kaastra and Boyd (1995). Separate neural networks are designed for corn, silver and the deutsche mark. A weekly nearby futures price series is used for corn (1969–1995), silver (1972–1995), and the deutsche mark (1975–1995) from the vendor Technical Tools. Each Tuesday's closing price is used to construct the nearby weekly series. Contract months are rolled over approximately one month before expiration. A weekly price series rather than daily series is used in order to reduce computing time, given the highly time intensive nature of neural network computing.

The first five years of weekly data are initially used to estimate the model in-sample and make the first period out-of-sample forecast. For example, assuming a one step ahead forecast, observations 1 through 250 (five years) would be used to estimate the model and forecast price for observation 251. The next period, observations 2 through 251 would be used to forecast price for observation 252. A rolling five-year data period is constructed each period, for estimation and making forecasts for each period, by dropping the oldest observation and adding a new observation (known as a sliding window forecast procedure). After the first five years of in-sample estimation is subtracted from the full data set, this leaves the out-of-sample forecast period for corn (1974–1995), silver (1977–1995), and deutsche mark (1980–1995), to be used for trading.

ARIMA models are used as a benchmark for comparison with the neural network. An ARIMA may be estimated as an autoregressive (AR) if the moving average (MA) process is invertible. Therefore, AR models are the form of ARIMA estimated here because they are simple to estimate, and have well developed model selection criteria. Typical AR procedures are used with the AIC method (Akaike, 1981) to identify the lag length. Both ARIMA and neural network models are estimated over the same data periods for side by side comparison, with two lags for corn, fourteen lags for silver, and twenty lags for deutsche mark.

Trading Strategy and Rule

One major strategy employed by many futures traders is the use of the trend as an aid in making trading decisions. This behavior is based on the assumption that, once a trend starts, it will continue. Traders want to follow trends so they can take positions early in the trend and maintain that position as long as the trend continues. Traders may, however, change their position when they predict a change in the trend or market direction. Weekly prices are used, rather than daily, as this avoids the excessive neural network computation involved with a daily price model. In real time trading, it would save computation time as well, by requiring data input and forecast computation once per week for one step ahead forecasts.

The weekly price trading strategy used here is to buy (sell) on the open if a rising (falling) price trend is forecasted. If no change in the trend from the previous week is predicted the current position is maintained. The trading rule is summarized as:

If $P_{t-1} < P_t > P_{t+1}$ then *Sell - Go short.*
If $P_{t-1} > P_t < P_{t+1}$ then *Buy - Go long,*

where P_{t-1} is the previous week trading closing futures price, P_t is this week's trading closing futures price and P_{t+1} is next week's trading forecasted closing futures price. These trading rules reflect turning points in the price movement.

A stop loss tolerance level is established to allow for exits in case of losses, in case some trades have large losses. In this study, if losses on the trade are more than 5% of the entry price, then the trade is exited, and the trader waits for a new signal. Details on the trading model and computation of trading returns can be found in the Trading Model Appendix.

RESULTS

Trading Return Levels

Tables I through III show monthly trading returns and their descriptive statistics. Neural network and ARIMA results can be compared directly at 10 and 25 period ahead forecasts. ARIMA forecasts were also made for one, three, and five period ahead forecasts as they are relatively fast to estimate. However, these forecast periods were not made for the neural network due to the computing time intensity. For example, with 20 years of weekly data, this would require about 1000 estimations per commodity for the one step ahead forecast.

Results show that both neural network and ARIMA models had positive returns for all three commodities, though none of the trading models appeared to consistently have higher returns than the others. Table I corn monthly returns range from approximately 1% to 5% percent. Multiplying this by 12 gives annual returns ranging from approximately 12% to 60%. Of the seven cases, only the ARIMA 5 step ahead and ARIMA 25 step ahead model are not statistically different from zero. Table II silver monthly returns range from approximately 3% to 5%, and are higher on average than corn. They are statistically significant in all seven cases. Table III shows monthly deutsche mark returns, ranging from approximately 1% to 2%. These returns are considerably lower than those for corn and silver, and none are statistically significant.

No particular reason can be easily identified as to why deutsche mark returns are lower than those for corn and silver, especially given that they are lower for both the ARIMA and neural network. Overall, these results of positive returns support the findings of Sweeny (1986), Boyd and Brorsen (1991), Lukac and Brorsen (1990), and Taylor (1994), who found positive trading returns using past prices. However, after removing the stops from the models, both models became less profitable, and some

TABLE I

Corn Futures Monthly Percent Returns for ARIMA and Neural Network Trading Models over Various Forecasting Steps Ahead (1974–1995)

MODEL	STATISTIC[b,c,d]		MONTHLY
ARIMA 1	Mean		4.01
	t-statistic		3.17
	Sharpe ratio		0.20
	Kurtosis		1.95
	Skewness		1.05
	Minimum		-40.42
	Maximum		75.81
	Draw down %	4.87	
ARIMA 3	Mean		4.06*
	t-statistic		2.78
	Sharpe ratio		0.20
	Kurtosis		19.90
	Skewness		3.01
	Minimum		-34.60
	Maximum		160.00
	Draw down %	4.37	
ARIMA 5	Mean		2.08
	t-statistic		1.18
	Sharpe ratio		0.09
	Kurtosis		20.51
	Skewness		3.30
	Minimum		-38.33
	Maximum		156.52
	Draw down %	4.12	
ARIMA 10	Mean		4.96*
	t-statistic		2.52
	Sharpe ratio		0.29
	Kurtosis		0.31
	Skewness		0.50
	Minimum		-37.96
	Maximum		56.52
	Draw down %	4.13	

(Continued)

TABLE I (Continued)

Corn Futures Monthly Percent Returns for ARIMA and Neural Network Trading Models over Various Forecasting Steps Ahead (1974–1995)[a]

MODEL	STATISTIC[b,c,d]		MONTHLY
ARIMA 25	Mean		1.08
	t-statistic		0.32
	Sharpe ratio		0.05
	Kurtosis		3.10
	Skewness		1.48
	Minimum		-24.18
	Maximum		70.61
	Draw down %	3.95	
NEURAL NET 10	Mean		2.67*
	t-statistic		2.07
	Sharpe ratio		0.13
	Kurtosis		7.77
	Skewness		1.46
	Minimum		-42.53
	Maximum		137.56
	Draw down %	4.87	
NEURAL NET 25	Mean		2.95*
	t-statistic		2.30
	Sharpe ratio		0.14
	Kurtosis		8.01
	Skewness		1.54
	Minimum		-42.53
	Maximum		137.56
	Draw down %	4.87	

[a]Results are out of sample.
[b]Commodity trading returns are calculated assuming 25% is invested in margins and 75% held back for potential margin calls.
[c]Draw down is the minimum of percentage changes in equity over the entire trading period. For example, if a trader started with $100,000 equity whose lowest point of equity at any time over the trading period was $60,000 then the draw-down would be $40,000 divided by $100,000, or 40%.
[d]Sharpe ratio is calculated by dividing mean return by standard deviation. The higher the Sharpe ratio the lower the risk of returns.
*Significant at 5%.

TABLE II

Silver Futures Monthly Percent Returns for ARIMA, and Neural Network Trading Models over Various Forecasting Steps (1977–1995)

MODEL	STATISTIC[b,c,d]		MONTHLY
ARIMA 1	Mean		5.07*
	t-statistic		5.26
	Sharpe ratio		0.37
	Kurtosis		2.13
	Skewness		1.11
	Minimum		-26.64
	Maximum		60.79
	Draw down %	20.06	
ARIMA 3	Mean		4.64*
	t-statistic		4.95
	Sharpe ratio		0.35
	Kurtosis		2.89
	Skewness		1.37
	Minimum		-17.98
	Maximum		60.78
	Draw down %	28.21	
ARIMA 5	Mean		5.22*
	t-statistic		6.01
	Sharpe ratio		0.43
	Kurtosis		1.51
	Skewness		0.89
	Minimum		-26.72
	Maximum		47.53
	Draw down %	28.21	
ARIMA 10	Mean		4.78*
	t-statistic		5.11
	Sharpe ratio		0.37
	Kurtosis		1.00
	Skewness		0.87
	Minimum		-26.72
	Maximum		47.53
	Draw down %	11.91	

TABLE II (Continued)

Silver Futures Monthly Percent Returns for ARIMA, and Neural Network Trading Models over Various Forecasting Steps (1977–1995)[a]

MODEL	STATISTICS[b,c]		MONTHLY
ARIMA 25	Mean		4.53*
	t-statistic		4.66
	Sharpe ratio		0.37
	Kurtosis		0.87
	Skewness		0.87
	Minimum		-19.24
	Maximum		45.50
	Draw down %	13.79	
NEURAL NET 10	Mean		4.03*
	t-statistic		3.79
	Sharpe ratio		0.28
	Kurtosis		5.07
	Skewness		1.58
	Minimum		-28.53
	Maximum		77.44
	Draw down %	13.79	
NEURAL NET 25	Mean		4.26*
	t-statistic		3.79
	Sharpe ratio		0.29
	Kurtosis		4.17
	Skewness		1.51
	Minimum		-28.53
	Maximum		77.4
	Draw down %	13.79	

[a]Results are out of sample.
[b]Commodity trading returns are calculated assuming 25% is invested in margins and 75% held back for potential margin calls.
[c]Draw down is the minimum of percentage changes in equity over the entire trading period. For example, if a trader started with $100,000 equity whose lowest point of equity at any time over the trading period was $60,000 then the draw-down would be $40,000 divided by $100,000, or 40%.
[d]Sharpe ratio is calculated by dividing mean return by standard deviation. The higher the Sharpe ratio the lower the risk of returns.
*Significant at 5%.

TABLE III

Deutsche Mark Futures Monthly Percent Returns for ARIMA and Neural Network Trading Models over Various Forecasting Steps (1980–1995)

MODEL	STATISTIC[b,c,d]		MONTHLY
ARIMA 1	Mean		0.64
	t-statistic		0.48
	Sharpe ratio		0.04
	Kurtosis		0.28
	Skewness		0.04
	Minimum		-50.83
	Maximum		45.59
	Draw down %	4.55	
ARIMA 3	Mean		1.71
	t-statistic		1.15
	Sharpe ratio		0.08
	Kurtosis		0.60
	Skewness		-0.35
	Minimum		-58.74
	Maximum		60.41
	Draw down %	5.02	
ARIMA 5	Mean		0.68
	t-statistic		0.41
	Sharpe ratio		0.03
	Kurtosis		1.85
	Skewness		0.03
	Minimum		-62.88
	Maximum		91.38
	Draw down %	5.02	
ARIMA 10	Mean		2.30
	t-statistic		1.36
	Sharpe ratio		0.10
	Kurtosis		1.38
	Skewness		0.06
	Minimum		-58.74
	Maximum		91.38
	Draw down %	5.02	

(Continued)

TABLE III (Continued)

Deutsche Mark Futures Monthly Percent Returns for ARIMA and Neural Network Trading Models over Various Forecasting Steps (1980–1995)[a]

MODEL	STATISTICS[b,c]		MONTHLY
ARIMA 25	Mean		2.58
	t-statistic		1.67
	Sharpe ratio		0.13
	Kurtosis		1.60
	Skewness		0.55
	Minimum		-39.62
	Maximum		84.66
	Drawdown %	4.87	
NEURAL NET 10	Mean		1.63
	t-statistic		1.10
	Sharpe ratio		0.08
	Kurtosis		1.02
	Skewness		0.46
	Minimum		-44.35
	Maximum		78.56
	Draw down %	5.03	
NEURAL NET 25	Mean		1.56
	t-statistic		1.03
	Sharpe ratio		0.07
	Kurtosis		0.99
	Skewness		0.47
	Minimum		-44.36
	Maximum		78.56
	Draw down %	5.03	

[a]Results are out of sample.
[b]Commodity trading returns are calculated assuming 25% is invested in margins and 75% held back for potential margin calls.
[c]Drawdown is the minimum of percentage changes in equity over the entire trading period. For example, if a trader started with $100,000 equity whose lowest point of equity at any time over the trading period was $60,000, then the drawdown would be $40,000 divided by $100,000, or 40%.
[d]Sharpe ratio is calculated by dividing mean return by standard deviation. The higher the Sharpe ratio the lower the risk of returns.
*Significant at 5%.

particular models became unprofitable, indicating that the models depend considerably on the stop. Since the ARIMA and neural network all had about the same levels of returns, it appears that the trading rule and stop loss method used may be more responsible for the positive trading returns rather than any of the two particular forecasting methods.

One reason for positive returns from ARIMA and neural network models, in contrast to the traditional market efficiency theory of zero profits, may be disequilibrium in markets. Disequilibrium theory argues that markets do not adjust instantaneously to information shocks because of market frictions, such as the cost of acquiring new information. However, over a longer time period, markets may slowly adjust to new information. Price trends may be the result of the adjustment process over time, caused by the information shocks, which in turn makes trading returns profitable (Beja and Goldman, 1980; Nawrocki, 1984).

For all cases, the higher steps ahead, e.g., 25 versus 10 versus 5, for which the forecast is made, does not appear to produce lower trading returns. This is in contrast to what may be expected. This may be because that while forecast error amount deteriorates with higher steps ahead, the forecast direction or up/down turning points may not (e.g., whether it gives a buy or sell signal). Given these results, it would appear that there is no particular advantage for trading from forecasting a few periods ahead, rather than 10 or 25 periods ahead, for these particular models.

Risk and Variability of Returns

Risk adjusted trading returns are given in Tables I to III by the Sharpe ratio. The Sharpe ratio used here adjusts returns for risk, by dividing the mean by the standard deviation on a per trade basis. Results show Sharpe ratios for corn ranged from 0.05 to 0.29. Silver was higher, ranging from 0.33 to0 .43. deutsche mark showed lowest return to risk, ranging from 0.03 to 0.13, which is relatively undesirable for traders, compared to those from corn and silver. Overall, the range of the Sharpe ratios for both the neural network and ARIMA models are comparable to those by Lukac and Brorsen (1990).

Draw down percentage is another measure of risk for trading returns. The type of draw down measured here is the maximum percentage of original trading capital (equity) lost at any point during trading. For example, if original trading capital is $100,000, and losing trades cause trading capital to fall by a maximum of $15,000 over the trading period, then the draw down is 15%. In practical terms, it describes the risk of how close the trader comes to losing all their initial capital at any point.

Table I shows draw down for corn ranging from about 4% to 5%. This indicates a relatively safe model to trade, as the most of the original capital the model loses at any point is 4% to 5%. Table II shows draw down for silver ranging from 13% to 28%. While this level is higher than that of corn, it is roughly consistent with that of trend following systems used by large commodity funds, and results of Lukac and Brorsen (1990). Table III shows draw down for deutsche mark 4% to 5%, which is relatively low and desirable for traders. The stop loss rule may have contributed to these relatively low draw downs.

Kurtosis statistics (Tables I to III), or fourth moments, show all returns are leptokurtic, meaning positive kurtosis coefficients. In other words, the returns have high central peaks with fat tails, as opposed to a normal distribution, which has a bell shape. This means that the overall returns are generated with a number of extreme positive and negative returns concentrated in the tails, similar to findings of Lukac and Brorsen (1990). Positive skewness or third moments are also found in all cases (except for the three step ahead ARIMA for deutsche mark), differing from the normal bell shaped distribution. This means in general that the distributions are also weighted by positive returns in the right tails.

Neural Network Performance

Some past research suggests that neural networks should perform comparably or more accurately than ARIMA models. However, in reality this may not always be the case, for a number of reasons, due to possible limitations of neural networks. One possibility is that the software error minimization algorithm may not always reach the minimum error. This could be due to the particular software used, and/or the subjective judgment required for the various parameters and estimation procedures in training, such as preprocessing and transfer functions. In other words, two researchers using identical neural network software and data can produce different results.

It is often argued that neural networks may be theoretically expected to perform better than the traditional linear ARIMA models if data are nonlinear. However, the data used in this study may only have limited nonlinear behavior, and testing data for nonlinear characteristics was beyond the scope of the study. Alternatively, from a practical standpoint, the neural network may not have been able to pick up possible nonlinearities due to possible neural network limitations in error minimization, mentioned above.

A technical trading model is also tested for brief comparison, since these models are used by many large commodity fund traders. The dual moving average model used here has been one of the most popular used by large commodity funds. It generates a buy signal if the 20 day moving average is equal or greater than the 50 day moving average. If the 20 day moving average is below the 50 day moving average, it generates a sell signal, so is always in the market. In order to remain consistent with earlier results, a 5% stop is also used. Results for mean monthly returns show 2.9% for corn, 2.7% for deutsche mark, and 7.2% for silver. These returns are similar to the neural network and ARIMA models for corn, but higher for silver and deutsche mark. The main reason for the higher silver returns is likely that the dual moving average system was able to quickly pick up the large up and down movement around 1980 in silver.

SUMMARY

The objectives of this study were to test the application of neural networks for commodity futures trading, which led to a number of results. Both the neural network and ARIMA model showed positive returns and at about the same levels. Because none of the particular forecasting models appeared to consistently produce higher returns than others, the trading rule used with each of the forecasts may have contributed more to producing positive returns than the particular forecasting method. However, returns were higher for corn and silver than deutsche mark, and they were not statistically different from zero for the deutsche mark.

One explanation for the positive returns may be disequilibrium in the various markets, which produce profitable price trends. Risk adjusted returns from the Sharpe ratio are comparable with those from traditional trend following trading systems. Draw down, the maximum percentage of trading capital lost from the original level over the trading period, also compares favorably with trend following systems. Since large commodity funds use have often used trend following systems, a dual moving average trend following system was also tested for a brief comparison with the ARIMA and neural network models. It was found to be generally more profitable for the deutsche mark and silver than was the ARIMA and neural network. One advantage of this model is that it uses only two parameters, yet is still capable of picking up nonlinear trends.

While the neural network would be theoretically expected to perform better than the ARIMA model if data is nonlinear, it did not, and this may be due to a number of reasons. This could be because the particular neural network software may have been unable to always reach the minimum error, however, this could also be due to particular procedures used by the researchers. Because neural networks require some subjective judgements for various parameters and estimation procedures, no two researchers will necessarily produce the same results. As well, the data may not necessarily be highly nonlinear, and in such cases, a simpler linear model may be expected to perform as well.

In summary, this particular research should not be used to make a general statement regarding the use of neural networks for trading, but is, rather, simply one more piece of evidence as to their capabilities and limitations. Further research should include a larger sample of commodities, since the three commodities tested here represesent only a small number of the 30 or 40 commodities that are fairly widely traded on futures markets, and are possibly an unrepresentative sample. Also, part of the time period of the data used here included parts of the 1970s and 1980s, which may be unrepresentative, as they are often recognized as very profitable trading periods. In contrast, markets may have become more efficient during the 1990s, with more large commodity funds impacting markets and less inflation, resulting in more choppy markets and less well defined tradeable trends. Considerably more research is needed on the usefulness of neural networks for trading in order to generalize results, given that they are still a relatively new and untested application for trading.

NEURAL NETWORK APPENDIX

Backpropagation Method

Standard BP employs an optimization method called gradient descent algorithm, which follows the contours of the error surface by always moving down the steepest slope. The objective of training is to minimize the total squared errors, defined as follows (McClelland and Rumelhart, 1986):

$$E = \frac{1}{2}\sum_{h}^{M} E_h = \frac{1}{2}\sum_{h}^{M}\sum_{i}^{N}(t_{hi} - O_{hi})^2 \tag{1}$$

where E is the total error of all patterns, E_h represents the error on pattern h, the index h ranges over the set of input patterns, and i refers to the ith output neuron. The variable t_{hi} is the desired output for the ith output neuron when the hth pattern is presented, and O_{hi} is the actual output of the ith output neuron when pattern h is presented. The learning rule to adjust the weight between neuron i and j is defined as:

$$\delta_{hi} = (t_{hi} - O_{hi})\, O_{hi}\, (1 - O_{hi}) \tag{2}$$

$$\delta_{hi} = O_{hi}\, (1 - O_{hi}) \sum_{k}^{N} \delta_{hk}\, w_{jk} \tag{3}$$

$$\Delta w_{ij}(n + 1) = \varepsilon(\delta_{hi}\, O_{hj}) \tag{4}$$

where n is the presentation number, δ_{hi} is the error signal of neuron i for pattern h, and is the learning rate. The learning rate is a constant of proportionality which determines the size of the weight changes. The weight change of a neuron is proportional to the impact of the weight from that neuron on the error. The error signal for an output neuron and a hidden neuron are calculated by Eqs. (2) and (3), respectively.

One method to increase the learning rate and thereby speed up training time without leading to oscillation is to include a momentum term in the backpropagation learning rule. The momentum term determines how past weight changes affect current weight changes. The modified BP training rule is defined as follows:

$$\Delta w_{ij}\, (n + 1) = \varepsilon(\delta_{hi}\, O_{hj}) + \alpha \Delta w_{ij}\, (n) \tag{5}$$

where α is the momentum term, and all other terms as previously defined.

The momentum term suppresses side to side oscillations by filtering out high-frequency variations. Each new search direction is a weighted sum of the current and the previous gradients. Such a two-period moving average of gradients filters out rapid fluctuations in the learning rate. Momentum values that are too great will prevent the algorithm from following the twists and turns in weight space while learning rates that are too small will require an inordinate amount of time to converge to a solution.

Neural Network Design, Evaluation, and Implementation

A three layered feedforward backpropagation neural network is developed for each commodity. Several neural network architectures are constructed and evaluated. The number of weights interconnecting each of the neurons in each of the layers determines a large part of the neural network's learning ability. A network with too few weights will be unable to learn the data patterns, whereas a network with too many weights may overfit the data. Overfitting occurs when a model has too few degrees of freedom, which means too few observations relative to its weights (parameters). It may also occur when data is pretested, and not analyzed out-of-sample. In order to avoid overfitting, a large five year sample is used for estimation, with a minimal number of hidden layers, and out-of-sample results are generated with no pretesting.

The neural networks are evaluated based on the test results after 500 runs. The evaluation criteria involves looking the average error, the root mean square error and the coefficient of determination (or R squared). The neural network is required to produce minimum average and root mean square errors, and maximum R squared. The architecture that meets this criteria becomes the architecture of the neural network to be trained for forecasting. Training the neural networks is automated using built in automation features of Brainmaker Professional version 3.0. For all the neural networks an automated sliding window training technique is employed (Kaastra and Boyd, 1995). Data preprocessing is done by using the automated preprocessing feature provided by Brainmaker Professional version 3.0 (California Scientific Software, 1993). The data are scaled based on maximum and minimum values. The neural network model is estimated on approximately 260 observations (five years) of weekly data in-sample. Neural network training uses 90 percent of the 260 observations and the remaining 10% are used for neural network testing.

Initial weights are chosen at random at the beginning of training. The learning rate is initially set at 3, but is allowed to change automatically depending on the magnitude of errors the neural network encounters and the speed at which the network is learning. Since the errors must decrease in size as the neural network trains, the learning rate is automated to change from 3 down to 0. The learning rate takes any number between 0 and 3. The training tolerance is initially set at 10%. Training is automated such that when 95% of the training observations have errors equal to training tolerance or below, the training tolerance is cut in half until it reaches 2%.

Testing is done while training at every 80 runs. The neural network is programmed to stop training when 90% of the test observations have errors less than or equal to 2%, or when the neural network reaches 4000 runs. Then the neural network is saved and is ready to be used for forecasting. The neural network output is used on out-of-sample data it has not seen before and makes a forecast. Forecasting is done for ten and twenty-five steps ahead ex post. Upon reading the observations the neural network makes a forecast of the output corresponding to each observation. The output is saved, and later this output is used with a trading rule to produce a buy or sell trading signal.

TRADING MODEL APPENDIX

Trading Model and Assumptions

The trading model is a computer program that simulates the trading of the technical trading system used in the present research. The model keeps track of weekly price movements, sell and buy transactions as well as weekly percentage returns. This information is saved for each commodity traded. The trading model assumes, first, no pyramiding of profits or positions is allowed. Second, the nearby futures contract is traded. Third, any draw down in equity is replaced with additional capital. Returns are calculated by subtracting transaction costs, which are made up of two parts. First, a $100 transaction fee per trade, which includes commission costs and skid error or slippage (Mandelbrot, 1963). In addition, a transaction cost of $100 per rollover is also subtracted. A transaction fee of $100 is used in order to make historical comparisons with other studies which used older data. A more recent and realistic transaction fee estimate would likely be $25 to $50.

Start-up Equity, Margins, and Trading Returns

Margins percentages vary by commodity and are assumed to be 5% for corn, 10% for silver, and 3% for the deutsche mark. These are consistent with historical average percentage margin levels (Brorsen and Irwin, 1987). Commodity trading returns are calculated using the total investment, of which 25% initial investment is invested in margins and the 75% held back for potential margin calls. This investment strategy is consistent with the capital management practices of large commodity funds (Lukac, Brorsen, and Irwin, 1988). The percent returns are then calculated on a monthly basis.

BIBLIOGRAPHY

Akaike, H. (1981): "Likelihood of a Model and Information Criteria," *Journal of Econometrics*, 16:3–14.

Beja, A., and Goldman, M. B. (1980): "On the Dynamic Behaviour of Prices in Disequilibrium," *Journal of Finance*, 34:235–247.

Boyd, M. S., and Brorsen, B. W. (1991): "Factors Related to Futures Market Disequilibrium," *Canadian Journal of Agricultural Economics*, 39:769–778.

Brock, W., Lakonishok, J., and LeBaron, B. (1992): "Simple Technical Trading Rules and the Stochastic Properties of Trading Returns," *Journal of Finance*, 47:1731–1764.

Brorsen, B. W., and Irwin, S. H. (1987): "Futures Funds and Price Volatility," *Review of Futures Markets*, 6:118–135.

California Scientific Software. Brainmaker, Nevada City, CA 95959.

Grudnitski, G., and Osburn, L. (1993): "Forecasting S & P and Gold Futures Prices: An Application of Neural Networks," *The Journal of Futures Markets*, 13:631–638.

Halquist, C., and Schmoll, G. (1989): "Neural Networks: A Trading Perspective," *Technical Analysis of Stocks and Commodities*, November:48–54.

Hecht-Nielsen, Robert. (1990): *Neurocomputing*. Reading, MA: Addison-Wesley.

Hornik, K., Stinchcombe, M., and White, H. (1989): "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, 2:359–366.

Kaastra, I., and Boyd, M. S. (1995): "Forecasting Futures Trading Volume Using Neural Networks," *The Journal of Futures Markets*, 15:953–970.

Lee, J. K., Trippi, R. R., Chu, S., and Kim, H. (1990): "K-Folio: Integrating the Markowitz Model with a Knowledge-Based System," *Journal of Portfolio Management*, 16:89–93.

Lukac, L., Brorsen, B. W., and Irwin, S. (1988): "A Test of Futures Market Disequilibrium Using Twelve Different Types of Technical Trading Systems," *Applied Economics*, 20:623–639.

Lukac, L. P., and Brorsen, B. W. (1990): "A Comprehensive Test of Futures Market Disequilibrium," *Financial Review*, 25:593–622.

McClelland, J. L., Rumelhart, D.E., and the PDP Group (1986): *Parallel Distributed Processing. Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA: MIT Press.

Mandelbrot, B. (1963): "The Variation of Certain Speculative Prices,"*Journal of Business*, 36:394–419.

Nawrocki, D. (1984): "Adaptive Trading Rules and Dynamic Market Disequilibrium," *Applied Economics*, 16:1–14.

Neftci, S. N. (1991): "Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of 'Technical Analysis,' " *Journal of Business*, 64:549–571.

Sweeny, R. J. (1986): "Beating the Foreign Exchange Market," *Journal of Finance*, 41:549–571.

Taylor, S. J. (1994): "Trading Futures Using a Channel Rule: A Study of the Predictive Power of Technical Analysis with Currency Examples," *The Journal of Futures Markets*, 14:215–235.

Technical Tools Data Vendor. Technical Tools, 980 Federal Highway, Suite 202, Boca Raton, FL 33432.

Trippi, R. (1990): "Intelligent Systems for Investment Decision Making," In Fabozzi, F. (ed.), *Managing Institutional Assets*. New York: Harper & Row.

Trippi, R., and DeSieno, D. "Trading Equity Index Futures with a Neural Network," *Journal of Portfolio Management*, 19(1992):27–33.

Copyright of Journal of Futures Markets is the property of John Wiley & Sons, Inc. / Business and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.