A NEURAL NETWORK APPROACH TO FUTURES TRADING

HONG PI
Comp. Science and Eng., Oregon Graduate Institute, P.O. Box 91000
Portland, Oregon 97291, U.S.A.
E-mail: pihong@cse.ogi.edu

and

THORSTEINN S. RÖGNVALDSSON
Comp. Science and Eng., Oregon Graduate Institute, P.O. Box 91000
Portland, Oregon 97291, U.S.A.
E-mail: denni@cse.ogi.edu

ABSTRACT

We present a futures trading system based on a nonstationary artificial neural network prediction model. The model is described and its design, as submitted to the first International Nonlinear Financial Forecasting Competition (INFFC),[a] is reported. The performance is demonstrated to be significantly better than some trivial trading strategies, like buy-and-hold and filter rules.

[a]INFFC is sponsored by the NEUROVE$T Journal and Genesis Financial Data Services

1. Introduction

Are there complex nonlinear dependencies in financial price series? Some (eg. academicians) say there are not and that financial price movements are a martingale. Thus excess returns, all costs and risks considered, are impossible to achieve in the long run. Others (eg. chartists) say there are, and that price series display definite patterns, with names like "head and shoulders", "double bottoms", and "island reversals" [1]. Furthermore, they are convinced these patterns can be exploited to generate excess profits, albeit that it takes "considerable experience" to use these patterns successfully.

In recent years there has been an upsurge of interest in new promising techniques for forecasting, made possible by the advent of fast and powerful computers. With methods like "Artificial Neural Networks" (**ANN**), a name no less enticing than those of trading patterns, one may try to extract nonlinear relations from the price series to develop profitable trading strategies. If this is truly possible, it would mean a challenge to the weak efficient market hypothesis, which is widely held to be true (by academicians).

We have constructed a trading system as part of the first International Nonlinear Financial Forecasting Competition (**INFFC**). The data set used is based on cotton futures and contains high frequency price registrations, which allows trading strategies utilizing both short and long term movements. Our trading system, which is based on nonstationary neural networks looking at short and long term trends, produces positive profits when transaction costs are considered

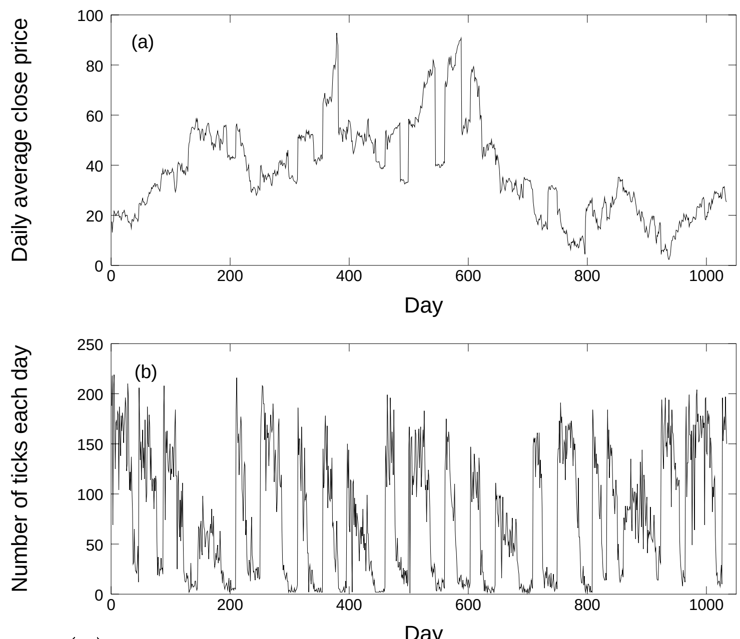

Figure 1: **(a)** Average closing price during each day versus day. **(b)** Total number of ticks each day versus day.

and outperforms simple filter rules and trivial buy/sell-and-hold strategies.

2. The Data

2.1. General Properties

The time series contains 80,000 entries, each with measurements of the six-variable tuple:

$$\{t, O(t), H(t), L(t), C(t), V(t)\} \tag{1}$$

where t is a time stamp in the 24hr format, $O(t)$, $H(t)$, $L(t)$, $C(t)$ are the open, high, low, and close price for that minute, respectively, and $V(t)$ is the total tick volume traded during that minute. These 80,000 entries correspond to 1034.5 trading days (the half day is the last day in the data set), which equals roughly four years of data.

Figure 1 shows the daily average closing price and the number of tick registrations each day. The trading activity, fig. 1b, shows a clear cyclic behaviour with a pattern of five activity peaks, which is repeated four times. The price series, fig. 1a, displays some remarkable drops in price, which last for 17-20 days before recovering to the price level before the drop. Downward level shifts occur when the trading activity is low and the recovery comes when the activity rises. This makes them quite predictable and one could imagine constructing a trading system based on these large level shifts. We believe, however, that these price changes mark transitions between contracts (or some artificially induced effect) and that it is impossible to make a profit from them in real life. We have therefore refrained from explicitely using these when building our system.

The trading activity, fig. 1b, displays at least one peculiarity. If one computes the time interval (in days) between successive peaks of the same height, one would get the number of trading days per year, which should be close to 250. However, something odd happens around day 825. The trading years are approximately 250 days long until that point, but become about 220 days long afterwards. This could be caused by one month of data having been left out or forgotten when the data was compiled.

The trading activity also shows a daily pattern. There are more trades being done during the first and the last minutes of the day than in the middle of the day. Furthermore, the trading activity during the initial 10-15 minutes of the day seems independent of the total number of trades made that day (i.e. some trades are independent of the season). Interday price changes in the data are generally more volatile than intraday price changes. The variance of interday price changes is almost an order of magnitude larger than the variance of intraday price changes. Price volatility is also generally higher during the initial 15 minutes of the trading day than during the remaining day.

3. Model Design

3.1. Design Considerations

To design a trading system for profit taking, one needs to construct a model that in one way or the other forecasts the price move, and then build a trading strategy that generates trading signals based on these forecasts. In the first part, the model construction, one must determine which input variables to include and what time horizons should be used for the price movement. For the second part, generating trading signals based on the model predictions, there are issues involving risk control. The transaction cost (dictated as 0.1% per transaction) would place an upper bound on how frequent one should trade (change positions). Should one clear positions overnight, or is it appropriate to hold positions overnight? Should one hold positions or go neutral when the system gives weak signals?

We choose to measure the risk vs. return trade-off with the

$$\textit{Sharpe ratio} = \frac{E[r] - R_F}{\sigma_r}, \tag{2}$$

where $E[r]$ is the expected return (after transaction costs have been accounted for), R_F is the *riskfree* rate of return (i.e. T-bill rate) and σ_r is the standard deviation of the return r.

The optimal trading frequency, in terms of the Sharpe ratio, depends on three things: How quickly the expected return grows with the length of the holding period, how far into the future the trader is able to forecast price movements, and the magnitude of the transaction costs. Obviously, the better the trader is able to forecast the price and the faster the expected return grows with holding time, the more frequent can the trades be.

The information needed to make the trading decision is of two sorts, long time and short time information. Trends are long time information, whereas changes in trends (turning points) are short time. Trend following should not generate excess returns if the market is at least nearly efficient and we can not expect to build a successful trading system by using only long term trend information. However, if the market exhibits some "inertia" and is slow in adjusting to new information, one may hope to gain excess return by quickly adjusting to short time price movements. Intuitively, we thus expect useful (i.e. hitherto unexploited) information to exist on the short time scale and not on the long time scale.

3.2. The Network Model

Our trading system consists of two components: a nonlinear model for predicting the direction of the market trend over certain time horizons, and a filter system for selecting the best indicators to generate trading signals.

Feedforward networks have been applied to predicting actual prices or price changes, with little or no success[2]. If accurate price predictions could be made, there is no doubt that one can easily build a winning trading system. However, if the accuracy is only slightly better than chance, a short term trading system based on these predictions would most likely fail in practice due to factors such as transaction cost and slippage cost (execution delays). Hence, rather than focusing on price itself, we choose to look for trends and changes in trends.

For this purpose, we introduce a new time series $\{\beta_N(t),\ t = 1, 2, ...\}$, with $\beta_N(t)$ being the slope parameter of a linear regression line fit through the following set of points:

$$\{(t, C(t)), (t+1, C(t+1)), ..., (t+N-1, C(t+N-1))\}, \quad (3)$$

where $C(t)$ is the close price at time t. The β_N variable gives the average relative price change during the next N ticks. It ignores tick-to-tick fluctuations and thus provides a good measure for the local trend.

Our prediction model is a committee of three multilayer perceptrons that use information up to the last tick $(t-1)$ to predict the trend variable β at tick t. Each of the networks has 9 inputs, 4 tanh sigmoid hidden units, and 3 tanh sigmoid output units. The outputs the networks are trained to predict are

$$\beta_6(t),\ \beta_8(t),\ \beta_{10}(t)$$

that is, the slopes over the next 6, 8, and 10 ticks, respectively. The 9 inputs are

$$T(t-1),\ OC(t-1),\ HL(t-1),\ \beta_6(t-6),\ \beta_6(t-10),\ \beta_8(t-8),\ \beta_8(t-13),\ \beta_{10}(t-10),\ \beta_{10}(t-16),$$

where T is the elapsed time, in minutes, since the market opening (10:30 am), $OC(t) = [O(t) - C(t)]/C(t)$ is the relative change between the open and close price during the minute, and $HL(t) = [H(t) - L(t)]/H(t)$ is the relative price differential between high and low. To predict the slope $\beta_N(t)$, a time lag of at least N is maintained in using the past values of β_N as inputs, since $\beta_N(t-N+1)$ requires the price information at t (unavailable at time $t-1$). The positive variable HL is rescaled to fill the range $[0, 1]$, whereas all other variables are translated and rescaled to be in the range $[-1, 1]$.

The three networks are trained using stochastic backpropagation with different initial conditions and random sampling of training patterns. The outputs of the three networks are averaged over to obtain three overall outputs, a so-called committee. Each output node of the committee is then used to issue a trading signal according to the following rule

$$S(t) = \begin{cases} +1 & \text{if } \beta(t) > \gamma \\ -1 & \text{if } \beta(t) < \gamma \\ \text{hold} & \text{otherwise} \end{cases} \tag{4}$$

where γ represents some cutoff parameters, which may be different for different output nodes and are determined through the process of model selection and validation.

The final trading decision is made by taking a majority vote among the three committee nodes corresponding to the three future time horizons. That is, the trading signal is followed only if it is supported by at least two of the three committee output nodes.

4. Model Validation and Evaluation

The networks in the committee are trained using normal backpropagation with early stopping. The training involves a training set with N_{tr} exemplars and a validation set of N_{va} exemplars. The entire training/validation data window is rolled forward and the networks retrained to take advantage of new data and overcome the problem of nonstationarity in the data. Specifically, at time $t-1$, the training set consists of data entries in $[t - N_{tr}, t-1]$, and the validation set consists of entries in $[t - N_{tr} - N_{va}, t - N_{tr} - 1]$. Although the networks are trained to predict the actual values of the slopes, the model selection is based on choosing the network committee with the highest probability of matching the signs of the predicted slopes with the true signs over the combined training and validation set. The selected network committee is then tested in a simulated trading environment for the next N_{te} ticks in the time frame $[t, t + N_{te} - 1]$. When time reaches $t + N_{te} - 1$, it is reset to $t-1$, the entire validation-training data window is rolled forward and the networks retuned and revalidated, ready for being tested on the next N_{te} ticks.

For the results reported in this paper we have used $N_{tr} = 1000$, $N_{va} = 1000$, and $N_{te} = 250$. The networks are initially trained for 50 epochs. Every time 250 new data entries are accumulated, the validation-training data window is shifted forward by 250 and the networks are retuned for 5 epochs. The model testing is thus a true *out-of-sample* testing. Since the validation-training and testing data windows are rolled through the entire set of 80,000 ticks, the test results of our model represent an *out-of-sample* testing on 78,000 data entries (the first 2000 ticks are used for the initial network training).

The cut-off parameter in eq. (4) is taken as

$$\gamma_i = 2.576\ \sigma(\beta_i), \tag{5}$$

where $\sigma(\beta_i)$ is the standard deviation of the i:th output node of the network committee, computed over the training set. If the predicted slope is normally distributed, the factor 2.576 would correspond to 1% in the tails of the distribution, which means that we trade only about 1% of the time.

Set	Trading strategy						
	B&H	S&H	F(5%)	F(15%)	F(25%)	F(35%)	Our system
1:5	0.444	-0.475	-0.370	-0.284	0.070	0.287	0.083 ± 0.137
2:5	-0.004	-0.015	-0.083	-0.864	-0.691	-0.076	-0.011± 0.058
3:5	-0.010	-0.002	-0.483	-0.994	-0.772	-0.772	0.277± 0.089
4:5	-0.184	0.177	-0.544	-0.642	-0.307	-0.187	0.322± 0.048
5:5	0.060	-0.069	-0.537	-0.114	0.000	-0.107	0.156± 0.108
Ave.	0.061	-0.077	-0.403	-0.580	-0.340	-0.171	0.165

Table 1: Sharpe ratios for buy/sell-and-hold strategies, filter rule trading (F), and our system. The Sharpe ratios are computed over the five different data subsets mentioned in the text. Numbers given within parentheses for the filter rule are the g percentage levels (see text).

Runs	I	II	III	IV	V
With Retrain	0.32(0.12)	0.35(0.14)	0.19(0.16)	0.13(0.14)	0.23(0.21)
No Retrain	-0.01(0.09)	0.19(0.11)	0.24(0.20)	-0.06(0.21)	0.32(0.22)

Table 2: Mean Sharpe ratios for various runs with and without retraining. The numbers in parenthesis are the standard deviations of the Sharpe ratios.

To evaluate our model, we compare its performance to that of some simple trading strategies, often mentioned in the literature on futures trading, as well as investigate the systems dependence on initial conditions and parameter values.

We divide the INFFC data set into five disjoint subsets, each roughly equivalent to ten 20-day periods over which we can compute a Sharpe ratio. We thus get an estimate of the variation in the Sharpe ratio between the different strategies. The three strategies we compare with are buy-and-hold, sell-and-hold, and the simple "g-percentage" filter introduced by Alexander[3] (see eg. Taylor[4] for a discussion). The latter has been claimed to generate considerable profits on trading with commodity futures even when transaction costs are considered. The filter rule is simple: if the trader is long and the price falls g percent below the highest price since the last position change, then go short. If the trader is short and the price rises g percent above the lowest price since the last position change, then go long. The initial position is based on g percentage deviations from the initial price. In our comparison, we use the average closing price each day as input to the filter rule, and not the tick-by-tick. The filter rule position is taken on the opening of each trading day, based on past average daily closing prices, including the previous day. As table 1 shows, the results for the filter trader are not overwhelming. It occasionally happens that the filter trading rule gives a positive Sharpe ratio for some percentage on one of the data subsets, but it never gives positive Sharpe ratios for the same percentage on two data subsets (a complete search was done for all percentage levels between 1 and 100).

For a long term evaluation of our system, we calculate the Sharpe ratios

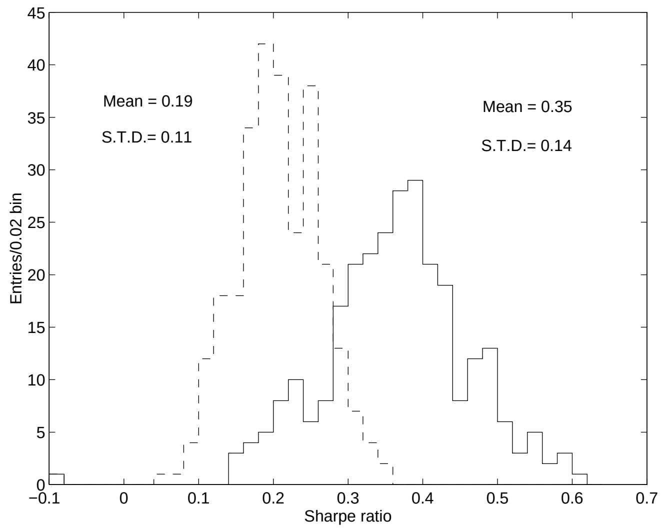

Figure 2: Distribution of Sharpe ratios for different starting points. The full and dashed lines correspond to the Run II with and without retraining, respectively (see text).

that it produces on data sets of size 140,000 minutes (approximately half the interpolated training data size), with the starting minute varying every 500 minutes across the training data set. The Sharpe ratios over these moving windows are averaged and the results for 5 independent runs are shown in table 2. For the runs I, II, and III, the networks are initially trained and validated over the ticks 1 to 2000, and then used for prediction over the entire data set. In run IV and V the networks were initially trained and validated over the ticks 10,001 to 12,000, and 43,001 to 45,000, respectively. The retrained network produces more consistent results than the not retrained network.

Fig. 2 shows the distribution of the Sharpe ratios for the moving data window in the run II. The Sharpe ratios are quite dependent on the starting point, but it is clear that the system produces positive Sharpe ratios more often than not. By averaging over the runs in table 2 we can deduce the average Sharpe ratio of 0.25 ± 0.16 to be a good estimate of the two year performance for our system.

Since the data has been preprocessed in a way unknown to us, it is sensible to ask whether our results (obtained from the preprocessed data) would remain valid when applied to the real data. However, comparing the data set to the price history of "Cotton No. 2 Futures" [5,6], we guess that the data has been preprocessed similar to

$$C = m\,(C_{true} - C_0), \tag{6}$$

where C is the close price in the data set, C_{true} is the true close price, and C_0 and m are positive constants. Due to this kind of rescaling, the returns obtained from our system would not correspond to reality. Since the Sharpe ratio is essentially the return normalized by the standard deviation of the return, it should not be affected much by the rescaling operation. For example, the trading signals from the run I will result in a Sharpe ratio of 0.26 for the entire 52-month data set, with an average "monthly excess return" of 10.7%. If we apply the trading signals to the "true" data by assuming $m = 4$ and $C_0 = 50$, the Sharpe ratio becomes 0.25 and the average monthly excess return changes to a more realistic 1.36% (with $R_F = 0.25\%$ considered, this means a gross yearly return of 19% for the system).

5. Conclusions

We have explored the feasibility of using artificial neural networks to construct a trend-sensitive trading system. Such a system, based on nonlinear prediction models and a filter trading strategy, seems able to pick up price movement directions and generate positive expected excess returns. The expected two year performance of the system is given by the mean Sharpe ratio 0.18, which represents an out-of-the-sample performance over nearly 80,000 minute records of the cotton futures price history.

Essential to our model design is how to properly define a "trend" variable. We find that the linear regression slope over a limited time horizon serves the purpose well.

The "rolling data windows" re-training and validation strategy we have adopted is advantageous in two aspects:

- It helps in overcoming the nonstationarity problem inherent in financial time series.
- It facilitates the efficient use of a large data set for both training and out-of-sample testing.

6. Acknowledgements

Both authors gratefully acknowledge funding provided by the Advanced Research Projects Agency and the Office of Naval Research under Grant ONR N000-14-92-J-4062. Dr. Rögnvaldsson acknowledges support from the Swedish Institute, and the Dept. of Theoretical Physics in Lund, Sweden, for the use of their computer equipment. We both thank Dr. Lizhong Wu and Steve Rehfuss for creative and stimulating input during the course of this work.

7. References

1. J. D. Schwager, *A Complete Guide to the Futures Market*, John Wiley & Sons, New York, 1984
2. A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Santa Fe Inst. Studies in the Sciences of Complexity vol. XV, Addison-Wesley, Reading MA, 1994
3. S. S. Alexander, "Price Movements in Speculative Markets: Trends or Random Walks", *Industrial Management Review* **2**, 1961, pp. 7-26
4. S. J. Taylor, *Modelling Financial Time Series*, John Wiley & Sons, Chichester 1986, pp. 196-224
5. N. Battley (ed.), *The World's Futures & Options Markets*, Probus Publishing Co., Cambridge, 1993, pp. 328-329

6. S. Buchanan *et al.* (eds.), *1991 CRB Commodity Year Book*, Commodity Research Bureau, Knight Ridder Financial Publishing 1991, pp. 63-72