NCCC-134
APPLIED COMMODITY PRICE ANALYSIS, FORECASTING AND MARKET RISK MANAGEMENT

Cash Soybean Price Prediction with Neural Networks

by

Ken L. Claussen and Dr. J. William Uhrig

Suggested citation format:

Claussen, K. L., and Dr. J. W. Uhrig. 1994. "Cash Soybean Price Prediction with Neural Networks." Proceedings of the NCR-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management. Chicago, IL. [http://www.farmdoc.uiuc.edu/nccc134].

Cash Soybean Price Prediction With Neural Networks

Kent L. Claussen and Dr. J. William Uhrig*

Introduction

Interest in neural networks has grown remarkably in recent years. This effort has been characterized in a variety of ways: as the study of brain style computation, connectioninst architectures, parallel distributed-processing systems, neuromorphic computation, or artificial neural systems. The common theme to these efforts has been an interest in looking at the brain as a model of a parallel computational device very different from that of traditional serial computing.(Rumelhart A)

Neural networks have gained favor in several different areas over their traditional counterparts. Neural networks have been successfully applied to a wide range of applications ranging from optimal nuclear power plant operation(Guo), to predicting sunspot activity(Villarreal), to sonar pattern classification(Rumelhart B). In economics, neural networks have been applied to financial market forecasting, macro economic prediction, credit risk classification, exchange rate prediction and numerous other applications.(Trippi)

The last couple of years has seen a great deal of interest in using neural networks for commodity/stock market trading. Several private trading firms are claiming success with neural network based trading systems, but little research has been completed in the public sector regarding the use of neural networks in the commodity/stock markets.

Purpose

The research presented in this paper is designed to determine the ability of neural networks to predict the directional movements in the central Illinois cash soybean price. The predictive abilities of the neural network models will be tested for statistical significance versus a random walk price series in the 5, 10, 20 and 30 trading day horizons.

Scope

This paper is limited to forecasting the directional changes in the univariate central Illinois cash soybean price series 5 to 30 trading days in the future. Univariate time series techniques use past data to formulate their forecasts.(Snyder) This type of forecast is mainly used in short-term decision making.

Neural Networks-Biological Foundations

The human brain is a wonder of nature. For many tasks it is far superior to the most powerful super computers available today. The human brain is especially well suited for adaptive processing and pattern classification. Recent breakthroughs have allowed researchers to develop limited computer models of the human brain via artificial neural networks.

---

* Graduate Student and Professor, respectively, Purdue University, Dept. of Agricultural Economics, 1145 Krannert, West Lafayette, Indiana 47907, Ph:(317)-494-4308 Fax:(317)-494-9176

The neuron is the heart of basic structure within the human nervous system. The neuron, from a biological standpoint, is a processor of information and is divided into three basic components: the dendrites, soma, and the axon. Electronic pulses are collected on the dendrite from other synapses(to be defined later) or biological signal generating structures. The electronic pulses or signals are propagated along the dendrite to the neurons nucleus region or soma. The soma maps the incoming signal to a corresponding output signal on the axon. The axon propagates the signal to the synapse(s) connecting to subsequent neuron's dendrite or the signals destination.(Zudra) The synapse may amplify or impede the signal.

The human brain has 10^11 neurons or individual processing units acting in parallel, connected with approximately 10^4 synapses or interconnections per neuron.(Zudra) The human brain learns the correct mappings in the soma and adjustments in the synapse by trial and error.

Artificial Neural Networks Theory

The artificial neural network's neurons or *nodes* are very similar to their biological counter parts. The artificial neurons collect weighted inputs from previous neurons(as in the biological synapse and dendrite) or from data inputs. The biological soma is represented by a two step process. First, the weighted input signals are summed. Secondly, the cumulative signal is passed through a transfer function creating the signal to pass to subsequent neurons as represented by the axon. The collection of artificial synapses is represented by a weight matrix.

Typically the artificial neural networks are comprised of at most 10^4 neurons organized in a matrix fashion. The small number of neurons and corresponding synapses limit the artificial networks ability to retain information. Artificial neural networks in general are limited to learning a specific problem and can be cascaded or serially linked for the ability to deal with diverse and unique multiple criteria problems.(Trippi)

The neural networks used in this paper are similar to traditional regression based model as you present a set of variables and the model returns a prediction for the dependent variable. The network in the training or learning phase calculates the error associated with the prediction and adjusts the model accordingly.

The most common transfer function is the sigmoid. Researchers have found a great deal of success in using this nonlinear mapping over other transfer functions.

The sigmoid transfer function is as follows:

$$F(net) = F(x) = \frac{1}{1 + e^{(-x/T)}} \qquad (1)$$

Where:

F(x): is the output of the neuron to the next layer

x: is the sum of the inputs from the previous layer times their corresponding connections weight

T: is the constant that determines the steepness of the transfer function (normally1).

The classical neural network contains three layers of neurons with two synaptic layers with sigmoid transfer functions. The first layer is called the input layer and just accepts the presented data without transformation. The second or hidden layer of neurons can consist of any number of nodes, typically one-half to two times the number of input neurons. The last layer of neurons is called the output layer, as it presents the networks results. The output layer, typically has one node for each variable being predicted.

During the training phases the network goes through a period of trial and error in creating the optimal connection(synapse) weights. The learning comes from adjusting the weights according to the error in the networks prediction. The adjustment of the weights is called backpropagation of the errors, for which the network receives its name, backpropagation network. The rate of change of the weights can be affected by the learning and momentum parameters.

Following are the eight steps for training a three layer back propagation neural network.(Zudra, Blum) All superscripts are for notation not exponentiation.

Where:

I and T are the matrixes of vectors of training inputs and target (correct) outputs to the neural network.

C is the number of training patterns in the training set.

H1 and O are the vectors of neuron activation levels for the neurons in the (first) hidden layer and output layer, respectively.

N, P1, and Q are the number of neurons in the Input, (first) hidden layer, and output layer, respectively.

W^{H1} and W^{O} are the weight vectors for the input to hidden layer and hidden layer to output layer. Superscripts notates receiving layer of the signal.

1. The weight matrix W^{O} and W^{H1} are initialized to small random values, typically between +1 and -1, (2) and (3). The weight adjustment vector for the period preceding the first is set to zero (4),(5) and other variable initialization (6).

$$W^{H1}{}_{p1} = Rand(+1,-1) \qquad \text{for p1=1,..,P1} \qquad (2)$$

$$W^{O}{}_{q} = Rand(+1,-1) \qquad \text{for q=1,..,Q} \qquad (3)$$

$$\Delta W^{O}{}_{0} = 0 \qquad (4)$$

$$\Delta W^{H1}{}_{0} = 0 \qquad (5)$$

$$c = z = 1 \qquad (6)$$

2. Set the tuple pointer, c, to the first tuple and load the data from the training matrixes into the appropriate vectors.

$$t_q = T_{c,q} \qquad \text{for q=1,..,Q} \qquad (7)$$

$$i_n = I_{c,n} \qquad \text{for n=1,..,N} \qquad (8)$$

3. The inputs are presented and the hidden layer(s) and output layers solution is calculated(9)(10).

$$H1_{p1} = F(i \cdot W^{H1}) \qquad \text{for p1=1..P1} \qquad (9)$$

$$Oq = F(H^{1} \cdot W^{O}) \qquad \text{for q=1..Q} \qquad (10)$$

4. The error adjustment vectors are computed for the hidden and output layers(11)(12).

$$\delta_q{}^{O} = O_q\,(1-O_q)(O_q - t_q) \qquad \text{for q=1..Q} \qquad (11)$$

$$\delta_{p1}{}^{H1} = H1_{p1}\,(1-H1_{p1})(W^{O}\delta^{O}) \qquad \text{for p1=1..P1} \qquad (12)$$

5. The weight adjustment vectors for period t are computed(13)(14).

$$\Delta W^{O}{}_{z} = \alpha H1\delta^{O} + \Theta\Delta W^{O}{}_{z-1} \quad (13)$$

$$\Delta W^{H1}{}_{z} = \alpha I\delta^{H1} + \Theta\Delta W^{H1}{}_{z-1} \quad (14)$$

Where:

$\propto$ = the learning rate.

θ = the momentum factor.

6. The weight vectors are adjusted according to the weight adjustment vectors(15)(16).

$$W^{O} = W^{O} + \Delta W^{O}{}_{t} \quad (15)$$

$$W^{H1} = W^{H1} + \Delta W^{H1}{}_{t} \quad (16)$$

7. Iterate to the next input/output pair in the training set. If last training pair continue(17)(18).

$$\text{If } (c < C) \quad (17)$$

$$c = c + 1 \quad (18)$$

go to step 2

8. Check for error tolerance, Mean Squared Error(19..22).

$$MSE_{z} = \frac{1}{C}\sum_{a=1}^{C}\left[\ \frac{1}{Q}\sum_{b=1}^{Q}(T_{a,b}\cdot O_{b})\cdot(T_{a,b}\cdot O_{b})\ \right] \quad (19)$$

if (MSE>error tolerance) (20)

$$c = 1 \quad (21)$$

$$z = z + 1 \quad (22)$$

goto step 2

Standard backpropagation can be improved in its speed by one to two orders of magnitude by including a momentum term. The use of momentum allows the network not only to respond to the local gradient, but also recent trends in the error surface. In essence this resolves the problem of the network getting trapped in a shallow local minimum in the error space.

The learning rate is a critical factor in assuring trainability in a neural model. Too large of a learning rate will lead to unstable learning where as too small a rate will lead to increased training time.(Demuth)

The excellent performance of a artificial neural network is not surprising when one considers the solid theoretical foundation on which the three layer feed-forward network rests. For example, any continuous function defined over a compact subset of $\Re^{n}$ can be approximated to arbitrary accuracy given sufficient hidden neurons.(Masters)

Artificial neural networks are unique in the sense that they incorporate the transfer function, topology of the network and connection weights into an optimal model. They also offer the ability to generalize by sensibly interpolating input patterns that are new to the network.(Demuth, Zudra)

Data

The data used in this study represents the Central Illinois cash soybean price. The data stream starts in January, 1974 and ends December 31, 1993. The primary data stream was purchased from Technical Tools (Technical Tools) with the supplementary data for bad or

missing observations coming from Dial Data (Dial Data). The data was used in its raw form without the use of any data transformations.

Theoretical Input Selection, Data Preprocessing and Procedures

The preprocessing of the inputs is essential to the development of an efficient neural network. Studies have shown data transformations can have a substantial effect on performance of a neural network.(Klimasauskas) Typically, over seventy-five percent of the time spent developing a neural network is focused on input data selection and transformation.(Trippi)

The literature available on data selection mainly touts the neural network's ability to sort through irrelevant data and extract only the meaningful relationships. For example, neural networks excel at taking data presented to them and detecting what data is relevant.(Blum) This is accomplished by downward adjusting the connection strengths of the irrelevant nodes while upward adjusting the relevant nodes. This in effect is the creation of a self-pruning neural network. The beauty of this style of network is it never needs to know what exact factors make the network solve, but just what works.(Blum) The problem in an approach of this nature is most of the computational expense will be used for data feature extraction and not for optimal network weight adjustment.

Neural networks must be presented during training with examples of all classes of the problem that are desired during testing or recognition. In order for the network to have adequate performance the data presented must be evenly spaced over the spectrum of input ranges. Studies have shown that training a neural network on data points representing the boundaries in the input hyperspace improves the neural networks predictive ability.(Davis)

It is normally necessary to trim the upper and lower spectrum of the training sample to alleviate the clustering of the values over a small portion of the input range.(Klimasauskas) The trimming improves the neural network's predictive ability by increased dispersion of the training sample through the nonlinear mapping function.

The presentation of the fundamental characteristics of any time series is essential in creating a trainable neural network. The training set can be analyzed at various points to determine the optimal lagging of input data. Standard cycle analysis can be used to detect patterns or short-term cycles in long term trending periods. Cycle analysis maps price data from a time series based domain to a frequency based domain. The results of the cycle analysis can be used to optimally develop lagged inputs to the neural network.

Classification of higher dimensia hyperspaces is one of the recurring problems in pattern classification. Procedures, such as neural network training, which are analytically or computationally manageable in lower dimension can become completely impractical and inefficient in a input space of higher dimensia. Various techniques have been developed for reducing the dimensionality of the feature or input space in the hope of obtaining a more manageable problem.(Duda) The lower dimensia training sets typically train faster and require less computational expense.

A well represented problem will exhibit trainable properties within the first two hundred and fifty epochs. If a solution is not attained by one thousand epochs a reformulating or transformation of the input space is needed.

Evaluation Procedure

The direction of the price move in many cases can be as important as the absolute price level. Mertorn's test is used to check for market direction forecastability.(Bodie)

$$\text{if } \Delta A_t > 0 \text{ then } A_t = 1 \text{ if } \Delta A_t \le 0 \text{ then } A_t = 0 \quad (23)$$

$$\text{if } \Delta F_t > 0 \text{ then } F_t = 1 \text{ if } \Delta F_t \le 0 \text{ then } F_t = 0 \quad (24)$$

where:

ΔA_t is the change in the actual price level from period t to the forecast horizon.

ΔF_t is the directional price change(+1 or -1) form period t to the forecast horizon.

A_t and F_t are the actual and forecasted binary values associated with the directional prediction in period t

The probability matrix for the forecasted change in actual value A_t conditional upon the sign of the changes in the forecast variable F_t are:

$$P_i = \text{Prob}(F_t = 0 | A_t = 0) \quad (25)$$

$$P_d = \text{Prob}(F_t = 1 | A_t = 1) \quad (26)$$

$$1 - P_i = \text{Prob}(F_t = 1 | A_t = 0) \quad (27)$$

$$1 - P_d = \text{Prob}(F_t = 0 | A_t = 1) \quad (28)$$

Equations (25) and (26) are the probability of correctly classifying an increase and decrease, respectively. (27) and (28) are the probabilities of an incorrect forecast.

The null hypotheses to determine if the forecasts are significantly different from a random walk price series is:

$$H_0: \; P_i + P_d - 1 \le 0 \quad (29)$$

vs.

$$H_1: \; P_i + P_d - 1 > 0 \quad (30)$$

The above hypotheses can be modeled via a regression equation.(Cumby)

$$F_t = \alpha_0 + \alpha_1 A_t + \varepsilon_t \quad (31)$$

where:

F_t is the forecasted binary value in (24) for period t.

α_0 is a constant in the regression.

$\alpha_1 = P_i + P_d - 1$

A_t is the actual binary value in (23) for period t.

ε_t is the error term for period t.

Given the regression equation in (31) α_1 must be significantly different from zero in order to prove forecastability.

Procedures

The preprocessing of the data was started by subsetting the original data into training and testing samples. The out of sample testing data started in January 1991 and continued through December 1992. The testing data was comprised of the data from January 1974 to December 1989. The required data points were pulled from 1973 for startup overhead. The data ranging from January 1990 to December 1990 was omitted in order to complete an out of sample test

elevating the possibility of any lagging indicators entering the training set and enough startup overhead for the testing data set.

The original testing data set was analyzed using cycle based analysis in 2 trending periods in the testing data set. The data points used for the analyses were then removed from the training set. The analysis indicated 10 moving average inputs ranging 2 to 58 days were optimal. In addition a starndard exponential moving average was added to the input set.

The completed training set of 11 inputs was then analyzed and the tuples with price changes two standard deviations from the mean were dropped. The training data dropped were considered to be outliers. Outliers typically slow training of a neural network. In reality there are sudden shocks to the price series such as dry weather in the summer causing huge price rallies.

The data was then mapped to a lower dimensia by standard discriminate analysis.

The transformed training data was not trimmed to present only boundary cases.

The resulting training data was presented to the neural network and trained for price changes 5,10,20, and 30 trading days in the future. The network consisted of a single layer with 16 neurons, 1.5 times the initial dimensia of the training sample.

The networks converged in a range of 85 to 483 epochs. Training was completed on a 486-66 running 32 bit version of MS-Windows 3.11(Microsoft). Preprocessing of the data was completed in custom programs developed in Borland C++4.0(Borland) and Excel 5.0(Microsoft). The neural network software used was Brainmaker 3.0 Professional(California Scientific Software) and custom neural network software in Borland C++ 4.0 for early data exploration and Matlab w/Neural Network toolbox(Mathworks) for final data analysis.

Results

Table 1 presents the results of the of Merton's test.

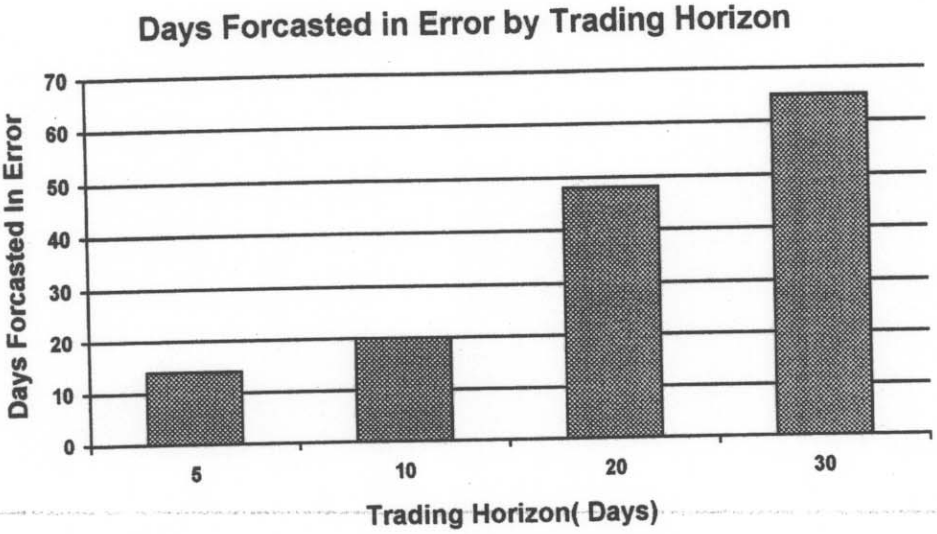
Results of Merton's Test

Forecast Length (Trading Days)	α_0	α_1	R^2
5	.0177	.9280	.8709
	(1.93)	(58.1)*	
10	.0400	.9210	.8400
	(3.55)*	(52.2)*	
20	.1090	.8160	.655
	(6.37)*	(30.9)*	
30	.1123	.7404	.5496
	(5.83)*	(24.8)*	

(t values in parentheses)

* Significant at 5 percent level.

α_0, α_1 coeficients in equation 31.



The results show that neural networks are able to forecast the market changes in all of the tested horizons in the future. The results could be very helpful to farmers as they make decisions on the sale of cash soybeans. The network would be able to tell in the short-term (i.e. 5 to 30 trading days) whether they should sell now or hold for a higher price. The results could also be used in deciding when to price deferred pricing contracts.

The above graph shows as the forecast horizon increases the possibility of misclassification increases at close to an exponential rate.

The network was able to predict the price changes 97%, 95%, 90% and 86% of the time in the 5, 10, 20, and 30 trading day horizons respectively. Accuracy levels of this nature would indicate the network is very well suited for this style of activity. The testing sample did not include any years with a large summer rally.

Conclusion

The purpose of this study was to evaluate the ability of neural networks to predict directional movements in the cash soybean market. The long-standing view in economics states that prices follow a random walk and past prices or prices fluctuations are not indicative of future prices changes. The market fluctuations therefore are believed to be chaotic time series. The typical statistical tests applied to time series are of a linear nature and therefore are not able to capture the higher order relationships embedded within the price series. Neural networks are able to extract the nonlinear mappings from the input to output space, as it is a completely data driven model.(Owens) The results of this study suggest that artificial neural networks are very well suited for predicting directional changes in the cash soybean price series.

.Blum, Adam, *Neural Networks in C++*, New York :John Wiley and Sons, 1992.

Bodie, Zvi; Kane, Alex; Marcus, Alan J., *Investments*, Homewood,IL:Irwin, 1992.

Borland Corporate Headquarters, 100 Borland Way, Scotts Valley, CA 95066-3249

California Scientific Software, Nevada City, CA 95959

Cumby, R.E. and Modest, D.M., "Testing for Market Timing Ability: A Framework for Forcast Evaluation," *Journal of Financial Economics*, 19:(1987)169-189

Davis, Daniel T. and Hwang, Jenq-Neng, "Attentional Focus Training by Bounndary Region Data Selection," Information Processing Laboratory, Dept. of Elecetrical Engineering, University of Washington, Seattle, WA.

Demuth, Howard and Beale, Mark *Neural Network ToolBox for Use with MATLAB-Users Guide*, The Math Works Inc, Natick, MA. 1992.

Dial Data, Brooklyn, NY

Duda, Richard O. and Hart, Peter E., *Pattern Classification and Scene Analysis*, New York:John Wiley and Sons, 1973.

Guo, Zhichao and Uhrig, Robert E., "TVA Plant Preformance Study with Neural Networks," Nuclear Engineering Dept., University of Tennessee, Knoxville, TN.

Klimasauskas, Casimir C., "Making a Difference with Data Transformation," *Advanced Technology for Developers*, June 1993

Masters, Timothy, *Pratical Neural Network Recipies in C++*, New York:Academic Press Inc., 1993.

MathWorks Inc., 24 Prime Park Way, Natick, Mass. 01760-1500.

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

Owens, Aaron J. "Artificial Neural Networks for Process Modeling and Process Control," DuPont Co., Wilimington, DE, 1993.

Rumelhart, David E.; Widrow, Benard; Lehr, Michael A., "The Basic Ideaas in Neural Networks," *Communications of the ACM*, March,1994A.

Rumelhart, David E.; Widrow, Benard; Lehr, Michael A., "Neural Networks: Applications in Industry, Business, and Science,"*Communications of the ACM*, March, 1994B

Snyder, John; Swear, Jason; Richardson, Michelle; Pattie, Doug, "Developing Neural Networks To Forecast Agricultural Commodity Prices," *Hawaii Internation Conference on System Sciences-25*,Kauai, Hawaii, Jan 8-10, 1992.

Technical Tools, Los Altos, CA.

Trippi, Robert R. and Turban Efraim, *Neural Networks in Finance and Investing*, Chicago:Probus, 1992.

Villarreal, James and Basses, Paul "Sunspot Prediction Using Neural Networks," NASA Lyndon B Johnson Space Center, Huston, TX.

Zudra, Jacek M., *Introduction to Artificial Neural Systems*, New York:West Publishing Company, 1992.