NEUREAL NETWORKS VERSUS TIME SERIES MODELS FOR FORECASTING COMMODITY PRICES

By Nowrouz Kohzadi

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Agricultural Economics and Farm Management
University of Manitoba
Winnipeg, Manitoba

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-315-92298-2

Name ______________________________

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

______________________________ □□□□

SUBJECT TERM SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture	0729
Art History	0377
Cinema	0900
Dance	0378
Fine Arts	0357
Information Science	0723
Journalism	0391
Library Science	0399
Mass Communications	0708
Music	0413
Speech Communication	0459
Theater	0465

EDUCATION

General	0515
Administration	0514
Adult and Continuing	0516
Agricultural	0517
Art	0273
Bilingual and Multicultural	0282
Business	0688
Community College	0275
Curriculum and Instruction	0727
Early Childhood	0518
Elementary	0524
Finance	0277
Guidance and Counseling	0519
Health	0680
Higher	0745
History of	0520
Home Economics	0278
Industrial	0521
Language and Literature	0279
Mathematics	0280
Music	0522
Philosophy of	0998
Physical	0523
Psychology	0525
Reading	0535
Religious	0527
Sciences	0714
Secondary	0533
Social Sciences	0534
Sociology of	0340
Special	0529
Teacher Training	0530
Technology	0710
Tests and Measurements	0288
Vocational	0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language	
General	0679
Ancient	0289
Linguistics	0290
Modern	0291
Literature	
General	0401
Classical	0294
Comparative	0295
Medieval	0297
Modern	0298
African	0316
American	0591
Asian	0305
Canadian (English)	0352
Canadian (French)	0355
English	0593
Germanic	0311
Latin American	0312
Middle Eastern	0315
Romance	0313
Slavic and East European	0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy	0422
Religion	
General	0318
Biblical Studies	0321
Clergy	0319
History of	0320
Philosophy of	0322
Theology	0469

SOCIAL SCIENCES

American Studies	0323
Anthropology	
Archaeology	0324
Cultural	0326
Physical	0327
Business Administration	
General	0310
Accounting	0272
Banking	0770
Management	0454
Marketing	0338
Canadian Studies	0385
Economics	
General	0501
Agricultural	0503
Commerce-Business	0505
Finance	0508
History	0509
Labor	0510
Theory	0511
Folklore	0358
Geography	0366
Gerontology	0351
History	
General	0578
Ancient	0579
Medieval	0581
Modern	0582
Black	0328
African	0331
Asia, Australia and Oceania	0332
Canadian	0334
European	0335
Latin American	0336
Middle Eastern	0333
United States	0337
History of Science	0585
Law	0398
Political Science	
General	0615
International Law and Relations	0616
Public Administration	0617
Recreation	0814
Social Work	0452
Sociology	
General	0626
Criminology and Penology	0627
Demography	0938
Ethnic and Racial Studies	0631
Individual and Family Studies	0628
Industrial and Labor Relations	0629
Public and Social Welfare	0630
Social Structure and Development	0700
Theory and Methods	0344
Transportation	0709
Urban and Regional Planning	0999
Women's Studies	0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture	
General	0473
Agronomy	0285
Animal Culture and Nutrition	0475
Animal Pathology	0476
Food Science and Technology	0359
Forestry and Wildlife	0478
Plant Culture	0479
Plant Pathology	0480
Plant Physiology	0817
Range Management	0777
Wood Technology	0746
Biology	
General	0306
Anatomy	0287
Biostatistics	0308
Botany	0309
Cell	0379
Ecology	0329
Entomology	0353
Genetics	0369
Limnology	0793
Microbiology	0410
Molecular	0307
Neuroscience	0317
Oceanography	0416
Physiology	0433
Radiation	0821
Veterinary Science	0778
Zoology	0472
Biophysics	
General	0786
Medical	0760

EARTH SCIENCES

Biogeochemistry	0425
Geochemistry	0996
Geodesy	0370
Geology	0372
Geophysics	0373
Hydrology	0388
Mineralogy	0411
Paleobotany	0345
Paleoecology	0426
Paleontology	0418
Paleozoology	0985
Palynology	0427
Physical Geography	0368
Physical Oceanography	0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences	0768
Health Sciences	
General	0566
Audiology	0300
Chemotherapy	0992
Dentistry	0567
Education	0350
Hospital Management	0769
Human Development	0758
Immunology	0982
Medicine and Surgery	0564
Mental Health	0347
Nursing	0569
Nutrition	0570
Obstetrics and Gynecology	0380
Occupational Health and Therapy	0354
Ophthalmology	0381
Pathology	0571
Pharmacology	0419
Pharmacy	0572
Physical Therapy	0382
Public Health	0573
Radiology	0574
Recreation	0575
Speech Pathology	0460
Toxicology	0383
Home Economics	0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry	
General	0485
Agricultural	0749
Analytical	0486
Biochemistry	0487
Inorganic	0488
Nuclear	0738
Organic	0490
Pharmaceutical	0491
Physical	0494
Polymer	0495
Radiation	0754
Mathematics	0405
Physics	
General	0605
Acoustics	0986
Astronomy and Astrophysics	0606
Atmospheric Science	0608
Atomic	0748
Electronics and Electricity	0607
Elementary Particles and High Energy	0798
Fluid and Plasma	0759
Molecular	0609
Nuclear	0610
Optics	0752
Radiation	0756
Solid State	0611
Statistics	0463

Applied Sciences

Applied Mechanics	0346
Computer Science	0984
Engineering	
General	0537
Aerospace	0538
Agricultural	0539
Automotive	0540
Biomedical	0541
Chemical	0542
Civil	0543
Electronics and Electrical	0544
Heat and Thermodynamics	0348
Hydraulic	0545
Industrial	0546
Marine	0547
Materials Science	0794
Mechanical	0548
Metallurgy	0743
Mining	0551
Nuclear	0552
Packaging	0549
Petroleum	0765
Sanitary and Municipal	0554
System Science	0790
Geotechnology	0428
Operations Research	0796
Plastics Technology	0795
Textile Technology	0994

PSYCHOLOGY

General	0621
Behavioral	0384
Clinical	0622
Developmental	0620
Experimental	0623
Industrial	0624
Personality	0625
Physiological	0989
Psychobiology	0349
Psychometrics	0632
Social	0451

NEURAL NETWORKS VERSUS TIME SERIES MODELS FOR FORECASTING COMMODITY PRICES

BY

NOWROUZ KOHZADI

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

Neural networks were originally developed in cognitive science to study the function of the brain. They have been shown to be universal and highly flexible approximators which do not necessarily require restrictive error term assumptions needed for parametric approaches such as the Box-Jenkins ARIMA models. Neural networks have been widely used in engineering and computer science for classification and pattern recognition and began to become more popular around 1985 when the method of backpropagation was introduced by the PDP group.

The objective of this study was to compare traditional ARIMA time series models with neural networks to see which can produce the superior forecast. The properties and behavior of the data are first examined and it is tested for nonlinear and chaotic behavior, using monthly US cattle prices from 1973-1990. Grassberger and Procaccia, BDS, and Hurst exponent tests are used to test for nonlinear and chaotic behavior. The results showed no such behavior. Therefore, linear models such as ARIMA models appear to be suitable for modelling cattle prices. But neural networks may have an advantage over linear ARIMA models because they do not require differencing and they also are less sensitive to violations of error term assumptions.

Forecasts were examined for errors and for the correct turning point prediction, in order to evaluate buying and selling signals. Results of neural network models showed that they were able to successfully forecast prices out of sample with considerably more accuracy than ARIMA models.

ACKNOWLEDGEMENTS

I would like to extend my greatest appreciation to my advisor Dr. Milton Boyd, whose support and knowledge of price forecasting and neural networks made this possible. I would also like to sincerely thank Dr. Bahman Kermanshahi of Electrical Engineering for introducing me to the subject of neural networks and for his kind and valuable instructions during my neural network simulations. I would like to thank my other thesis committee members Drs. Daryl Kraft, Dana Hoag, and Harry Turtle for their assistance. As well, I would like to sincerely thank Dr. David Scuse of Computer Science and Dr. Howard Card of Electrical Engineering for their wonderful neural network theory courses I took from them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Expert opinion, econometric, and time series methods have traditionally been used for price forecasting. Time series techniques such as Box-Jenkins ARIMA have proven in many cases to be at least as effective as other econometric techniques in terms of forecasting power (Brandt and Bessler; Harris and Leuthold; Dorfman and McIntosh). In addition, a great advantage of time series models over econometric models is that they are highly cost efficient in terms of required data. They require fewer variables than econometric models, because they usually require only past lags of the dependent variable.

However, recent success of neural networks in engineering and physical science has opened a new area of inquiry for economics and price forecasting problems. Neural networks are computational structures which mimic the function of the human brain. They have been shown to have universal approximation power which can approximate all data generating processes. The objective of this study is therefore to compare the forecasting power of traditional time series models with neural network models. This study uses ARIMA time series models as a benchmark for forecasting US cattle prices and investigates whether neural networks can produce better forecasts than ARIMA time series models.

The organization of the study is as follows. Chapter two analyzes the properties of the data and tests the data series for chaotic behavior. This procedure is important because the behavior of the data and its properties have important implications for forecasting. Monthly US cattle prices are used and the period is from 1973-1990. A chaotic series can be defined as a series which appears random but in reality is not random. Chaotic series are a subset of nonlinear processes that cannot be modeled by linear time series or linear regression models. If prices are generated by nonlinear processes, then linear time series models such as ARIMA are no longer relevant. Therefore, it is necessary to determine whether ARIMA models are an appropriate choice for modelling US cattle prices. Since the presence of nonlinear behavior and chaos in time series usually is not picked up by the conventional tests and methods, Hurst (Mandelbrot), Grassberger and Procaccia, and Brock et al., have proposed tests which are used here. They can pick up chaotic and subtle nonlinear behavior which conventional tests fail to identify.

Since neural networks are nonlinear and flexible, they are likely to outperform ARIMA models if the data is nonlinear. However, even if the data is linear, neural networks may still have advantages over ARIMA models. This is because they do not strictly impose the differencing requirement on non-stationary data and they are also less sensitive to violations of traditional error term regression assumptions than ARIMA models.

The third chapter explains the structure of neural networks. Neural networks were originally developed in cognitive science and later were used in engineering for pattern recognition and classification. This chapter covers the history of neural networks, topology, transfer functions, learning algorithms and optimization, out of sample prediction, and the statistical aspects of neural networks in relation to regression analysis and applications to the economic forecasting problems.

Chapter four examines the out of sample forecasting power of ARIMA models with that of neural network models. The data are first examined for stationarity, and proper differencing is undertaken to estimate the ARIMA model. This same differenced data is used for training a fully connected single hidden layer neural network. At the forecasting stage, each forecasted value is used to perform the next step forecast without re-estimating the model or re-training the network. Forecasts are examined not only for forecast error, but also for correct turning point prediction and price direction so that buying and selling performance is taken into account.

In chapter five, the out of sample forecasting power of the ARIMA model is compared with the neural network model when both actual and differenced data are used as inputs for the neural network. The ARIMA models in this chapter still use the differenced data because it is necessary for these models to work with stationary data. However, differencing may cause loss of information inherent in the data, which may put the ARIMA models at a disadvantage. Since it is believed that actual data will capture the trend in the data and differenced data may capture the deviation around the trend, and since neural networks do not impose such a differencing restriction on the data, both actual and differenced data are used for the neural network model. Therefore, they may provide better forecasts than the ARIMA model. Two extra years of out of sample data are also added for this model in order to provide a more comprehensive test of out of sample performance.

CHAPTER 2

TESTING FOR CHAOS AND NONLINEAR DYNAMICS

Introduction

Developments in the study of chaotic time series reveal that conventional tests may not be able to pick up some of the nonrandom behavior. As a result, a seemingly random variable may not be truly random and inference based on this estimation can be misleading. Moreover, there are reasons to believe that some economic variables may be chaotic (Baumol and Benhabib). Therefore, before addressing the forecasting accuracy of prices from their past prices by the linear models, it is necessary to understand the behavior of the data and to make sure that the linear model is an appropriate method.

The objective of this chapter is therefore to test for the presence of chaos and nonlinear dynamics in prices. If the data is found to be linear, then traditional time series models are appropriate. However, if nonlinear dynamics are present, then ARIMA models are no longer suitable. In such cases, flexible functional form models such as neural network models should be used.

While cattle price behavior has been analyzed by numerous researchers, this is the first study to test for chaos and nonlinear dynamics in cattle prices. The theoretical background of chaotic time series are discussed followed by a discussion of the tests used to address the above problem. Results of the tests are discussed, followed by a conclusion.

Theoretical Background

Definition

Chaos may be defined as a nonlinear deterministic process that is nonrandom but looks random (Hsieh). Chaotic series are a subset of nonlinear processes which generate increasingly complex and irregular results (Larrain). The mapping $\mathbf{x}_t = f(\mathbf{x}_{t-1})$ is chaotic if there is a nonlinear feedback from the current realization of the process to its future behavior and if the system's coefficients pass their critical level (Savit). The following logistic map equation is a classic example of such a dynamic system:

$$\mathbf{x}_t = A\mathbf{x}_{t-1}(1 - \mathbf{x}_{t-1}) \tag{2.1}$$

where $\mathbf{x}_t$ is from the interval [0,1] and A = [0,4]. Once A exceeds 3.57, the process begins producing a rich variety of behavior which rarely repeat themselves in finite samples. In this case, if $\mathbf{x}_t$ is plotted against $\mathbf{x}_{t-1}$, visual inspection of the process cannot distinguish its behavior from a random process. In many cases the spectrum and the autocovariance function of the chaotic series are the same as white noise (Brock et al. 1992).

Properties

Although a chaotic series may look random, it has certain properties which makes it different from a random process. The first property is that increasing the embedding dimension of chaotic series does not change the dimension of their attractors. The embedding dimension is the dimension of a geometric space in which any scaler time series can be embedded. This dimension can vary from two to higher values. For example, equation 2.1 can be plotted in a two dimensional space such that one axis represents $\mathbf{x}_t$ and the other axis represents $\mathbf{x}_{t+1}$. The attractor of equation 2.1 is a one dimensional curve which is observed in that space. Now, if another dimension to the above space is added, e.g. for $\mathbf{x}_{t+2}$, and the series are drawn in this 3 dimension space, the attractor of equation 2.1 still remains one dimensional. In contrast, for random numbers, the dimension of the attractor will increase with increasing the embedding dimensions.

The second property of chaotic series which makes it different from a random process is its sensitivity to the initial state. That is, a small error in measuring the initial state exponentially amplifies into the future realization of such series.

The third property, which is similar to the second, is that small changes in the parameter of the model dramatically change the behavior of the process. For example, in equation 2.1, once A takes on values greater than 3.57, the long run behavior of $\mathbf{x}_t$ becomes abrupt and totally different from its past.

Another important difference between random numbers and a chaotic series is that the routes of chaotic series to the chaos are the same. That is, chaotic systems evolve from having one equilibrium solution to 2, 4, 8, ..., and 2^n equilibrium solutions until they become chaotic. This phenomenon is called bifurcation and is independent of the type of mapping which generates such a series.

Implications and Previous Studies

One implication of chaos theory is that before drawing any inference upon a time series, it should be examined for hidden nonlinearity in the error term. The first property can be used to build tests of nonlinearity as will be discussed in the next section. Another implication of chaos theory is that if the process is not generated by a very complex model, it should have short term predictability but not with linear models (Hsieh). However, the initial state dependence from the second property indicates the importance of correct specification of the initial state of the process. The third property warns that a model that fits very well for past prices may not be able to perform well in the future if the price series is governed by chaos. This is because the process visits every point on its interval and it may take a long time before repeating itself. Therefore, care should be taken when attempting to forecast chaotic prices.

The study of chaotic time series began in natural science, physics and chemistry, and further attracted economists for studying economic variables (Willey). Empirical applications of nonlinear dynamics are mainly concentrated in macro economic analysis or capital markets (Barnett and Chen; Scheinkman and LeBaron). However, recently there have been a few studies trying to address nonlinear dynamics in the commodity markets (Blank; Chavas and Holt; Yang and Brorsen). Results of these studies are mixed, as Chavas and Holt found deterministic chaos in the dairy market, Blank found it in the soybean market, while Yang and Brorsen (1992) concluded that changes in cash prices of seven commodities they studied did not follow a low dimension deterministic chaos process. Yang and Brorsen (1993) also found no strong support for or against deterministic chaos in the futures markets for most of the above commodities.

Methods of Identifying Chaos

The underlying properties of chaotic time series suggest ways of identifying them from truly random variables. Tests of chaos include the Grassberger and Procaccia test, the BDS test, and the Hurst exponent test (Lorenz). This study will perform the above three tests for the case of US cattle prices.

Grassberger and Procaccia Method

Since chaotic processes eventually repeat themselves, one can start by plotting the map of $\mathbf{x}_{t+i}$ against $\mathbf{x}_t$ in a two dimensional space and examine whether there is any pattern in the data after a large number of iterations. If there is no pattern in the two dimensional space, the three dimensional space is tried. However, a problem with this method is that visual inspection of patterns in the data in a three dimensional spaces is difficult and for higher than three dimensional mappings, it is impossible. The Grassberger and Procaccia (GP) method overcomes this problem and identifies patterns in an m embedding dimension space. The steps in GP method are as follows:

First, filter the data by removing any autocorrelation. This is done by fitting a proper AR(p) model into the data and retrieving the residuals for the next step. Second, specify an embedding dimension m and a radius ε. Then specify the points in m-dimensional space such that

$$\mathbf{x}_t^1 = \mathbf{x}_t$$

$$\mathbf{x}_t^2 = (\mathbf{x}_{t-1}, \mathbf{x}_t)$$

.....

$$\mathbf{x}_t^m = (\mathbf{x}_{t-m+1}, \mathbf{x}_{t-m+2}, \ldots, \mathbf{x}_t)$$

Third, calculate the correlation integral $C_{m,T}(\varepsilon)$ by

$$C_{m,T}(\varepsilon) = (2/T_m(T_{m-1}))[\Sigma_{t<s} I(\| \mathbf{x}_t - \mathbf{x}^s_t \| <\varepsilon)] \qquad (2.2)$$

where T_m = T- m +1 and I is a binary variable equal to one if the difference between $\mathbf{x}_t$ and $\mathbf{x}_s$ is less than ε and equal zero otherwise. Fourth, start with setting m=2 and calculate the correlation integral $C_{m,T}(\varepsilon)$ for different values of ε. Then regress $\ln(C_{m,T}(\varepsilon))$ on $\ln(\varepsilon)$ to get the slope coefficient d_2 which is called the correlation dimension. Repeat the above calculation until the correlation dimension d_m for corresponding levels of embedding dimension m is calculated. Finally, examine the relationship between d_m and m by plotting d_m against m. If the correlation dimension increases with m but stabilizes beyond certain values of m, then it can be concluded that the data is chaotic. In contrast, if d_m continues to increase with increasing values of m, then the data is white noise.

BDS Test

The GP method has several limitations (Hsieh). First, it is a descriptive and not a statistical test. Second, it fails to distinguish some nonlinear models, such as ARCH, from chaotic models. Third, in a small sample size d_m it is biased downward showing that chaos is present, when it may not be. To correct the above problems, Brock et al. (1987) extended the idea of the correlation dimension to a statistical test known as the BDS test. This test can be considered as a statistical representation of the GP method. The idea behind the BDS test is that if a scalar time series is randomly sampled from an identically and independently distributed population (IID), then its correlation dimension must increase with increasing the embedding dimension. i.e.

$$C_{m,T}(\varepsilon) = C_{1,T}(\varepsilon)^m \tag{2.3}$$

Brock et al. showed that the difference between both sides of (2.3), standardized by the standard deviation of C_m, asymptotically has a normal distribution. i.e.

$$W_{m,T}(\varepsilon) = T^{1/2}[C_{m,T}(\varepsilon) - (C_{1,T}(\varepsilon))^m]/\sigma_{m,T}(\varepsilon) \sim N(0,1) \tag{2.4}$$

where T is the sample size, $C_{m,T}(\varepsilon)$ is the correlation integral, and $\sigma_{m,T}(\varepsilon)$ is the standard deviation of C_m . Note that the BDS test uses C_m instead of $\log C_m$ as in the GP method.

The BDS test is a test of null of IID against the alternatives. Therefore, rejection of IID does not automatically imply chaos. Hsieh suggests that in addition to chaos, there are two other candidates for the rejection of IID; structural changes and conditional heteroscedasticity.

To perform the test, Brock et al.(1992, p. 175) emphasize that the assumption of stationarity is crucial in the above test. Therefore, the steps to perform the above test are as follows:

1) Make the data stationary

2) Remove autocorrelation from the data.

3) Compute the correlation dimension for various m and ε.

4) Apply the test. If IID is rejected, then further test for presence of structural changes and conditional heteroscedasticity versus chaos.

The optimal performance of BDS test requires 500+ observations. However, Brock et al. (1992) pointed out that "even with 50 to 200 observations, BDS performs fairly well compared to the other tests."

Hurst Exponent Method

In the study of long-term reservoir storage, Hurst defined a scaled ranged statistics, R/S, which measures the cumulative departures of a scaler time series from its mean. The R/S statistic is computed by the following formula (Mandelbrot):

$$C_N = \sum_{t=1}^{N} (X_t - M_N) \tag{2.5}$$

where

C_N = cumulative deviation over N periods

X_t = observation in time t, t=1,...,N

M_N = average X_t over N periods

The range R is then defined as the difference between maximum and minimum value of C_N defined above. Hurst proposed to rescale the range by the standard deviation of X_t, S, and formulated the following relationship:

$$R/S = (a*N)^H \tag{2.6}$$

where a is a constant and H is the Hurst exponent. R/S is expected to increase with N.

The Hurst exponent has the advantage that it is a distribution free classifier which can identify random variables from non-random. Hurst and Feller independently showed that for a purely random series, H is equal to 0.5. On the other hand, if $H \neq 0.5$, then the series has memory and is not random.

The application of the Hurst exponent to economics includes Greene and Fielitz (1977), using security prices, Booth et al.(1982a, 1982b), using exchange rates and gold markets, Helms et al.(1984), using futures markets, and Peters (1991) using the stock market. The procedure for the calculation of Hurst exponent is as follow (Peters 1991): i) the series of residuals is divided into subseries of equally small ranges N and then the R/S of subseries are computed and are averaged by dividing by the number of subseries.

ii) the range of subseries N is increased and the corresponding average R/S are computed.

iii) the Hurst exponent, H, is estimated by the regression model:

$$\log(R/S) = \log(a) + H\log(N) + u \qquad (2.7)$$

where u is the error term.

Data

Monthly cash prices are used to avoid noise found in daily or weekly data. Prices in \$/100 lb of cattle (900-1100 lb) in Omaha are used to test for the presence of nonlinear dynamics and chaos. Data are obtained from the CRB Commodity Year Book, various issues, and cover the period 1973-1990.

Results

First, the autocorrelation function of the data is examined for the presence of non-stationarity. Figure 2.1 shows that the autocorrelation function of prices before differencing dies out slowly which is indicative of non-stationarity. Figure 2.2 shows the autocorrelation function of prices after the necessary first differencing. Figure 2.2 indicates that first differencing was appropriate to make the series stationary. To filter the data from serial dependency, the Box-Jenkins time series method was used. Results show that the following ARIMA is appropriate for the original data set to make residuals of the model both stationary and without autocorrelation.

$$\begin{array}{llllll}
\mathbf{p}_t = & 0.17 + & 0.22\,\mathbf{p}_{t-1} + & -0.22\,\mathbf{p}_{t-3} + & -0.10\,\mathbf{p}_{t-7} + & 0.23\,\mathbf{p}_{t-11} \\
\text{(t ratios)} & (0.95) & (3.51) & (-3.42) & (-1.45) & (3.51)
\end{array} \qquad (2.8)$$

Significant coefficients in the above model indicate that current price is correlated with past prices, and therefore, not random. While Ljung-Box test results (Table 2.1) indicate that residuals are white noise and that the underlying data generating process for this set of data is linear, chaos theory suggests further testing of the residuals for hidden nonlinearity in the data series is necessary.

Grassberger and Procaccia Test Results

Residuals from the above ARIMA model were used to calculate the correlation integral, $C_{m,T}(\varepsilon)$ and the correlation dimensions, d_m. Brock et al. (1992, p. 52) suggest that from a practical point of view, embedding dimensions m=5, as an upper bound, and ε ranging from 0.5 to 2 standard deviations of the data are reasonable choices for detecting chaos. The plot of $\log(C_{m,T})$ against the $\log(\varepsilon)$ for various levels of m and ε are shown in Figure 2.3. It can be seen that the relationships between the two variables are approximately linear. By regressing the $\log(C_{m,T}(\varepsilon))$ against $\log(\varepsilon)$, the correlation dimensions of the data for various levels of embedding dimension, m, were estimated (Table 2.2). Plot of d_m against m in Figure 2.4 shows that even for m>5, d_m is not stabilized. According to the GP criterion, this means that there is no hidden non-linearity or chaos in the data set.

BDS Test Results

Table 2.2 shows the results of the BDS test for embedding dimensions 2 to 5 and for ε ranging from 0.5 to 2 standard deviation of the data normalized between 0 and 1.

Results show that the BDS statistics are within the range of normal distribution. Therefore, the null of IID is not rejected. This is in support of the previous test that a linear model can best represent the data.

Hurst Exponent Test Results

To compute the Hurst exponent, residuals from the ARIMA model were first divided into non-overlapping subseries of length 6 to allow for enough variations for standard deviation estimates. The R/S statistic was calculated for each of these sub-samples and averaged over the sum to obtain the R/S statistics for N = 6. This procedure continued for N = 7 up to N = 102. As a result, 97 average R/S values corresponding to different length of subseries were obtained. The following equation is estimated using these data:

$$\begin{array}{lccccr} \log(R/S) & = & -\,0.097 & + & 0.56\,\log(N) & & (2.9) \\ \text{(t ratios)} & & (-4.31) & & (42.13) & R^2 = 0.94 & \end{array}$$

The above estimates show that the Hurst component for the data is 0.56, which is very close to that of a random series which would be 0.5. Although the above coefficient is statistically different from 0.5, Peters (1992) found similar value for a randomly scrambled data set. Therefore, the results of this test are again further evidence of the linear characteristics of US cattle prices.

Conclusion

Chaos theory suggests that seemingly random variables may come from nonlinear data generating processes. It also suggests that conventional tests and methods such as those of market efficiency may fail to identify such subtle nonlinearities. If this is the case, then inferences drawn upon linear models are no longer reliable. This study investigated the relevance of linear model for the case of monthly US cattle prices by examining the residuals of the fitted linear model for the hidden nonlinearities.

However, results of the three tests indicate that the dynamics of price behavior are linear and are not chaotic. Therefore, linear models appear to be acceptable models for monthly price forecasting in the cattle market. Finally, even though this study found the data to be linear, it follows that it is still reasonable to test whether nonlinear models such as neural networks are superior to linear time series models such as ARIMA, since neural networks are less sensitive to traditional regression error term assumptions.

Figure 2.1 Autocorrelation Function of Prices before Differencing

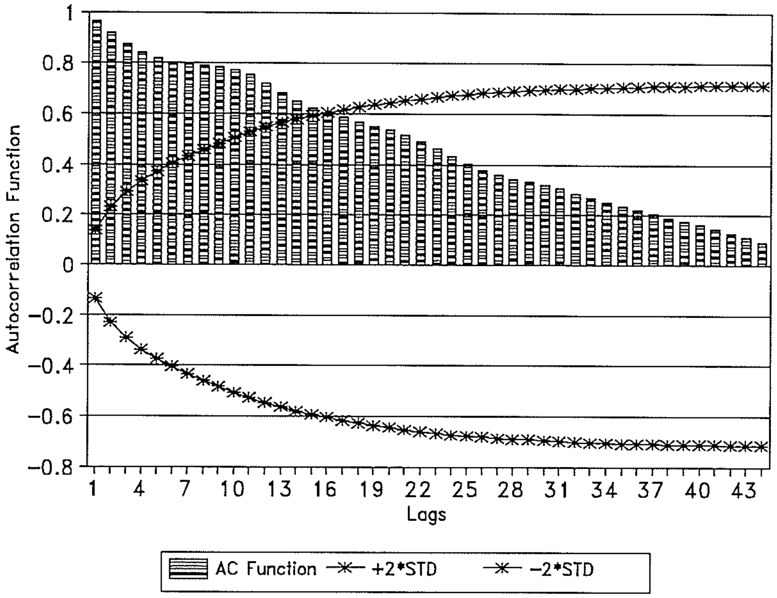


Figure 2.2 Autocorrelation Function of Prices after Differencing

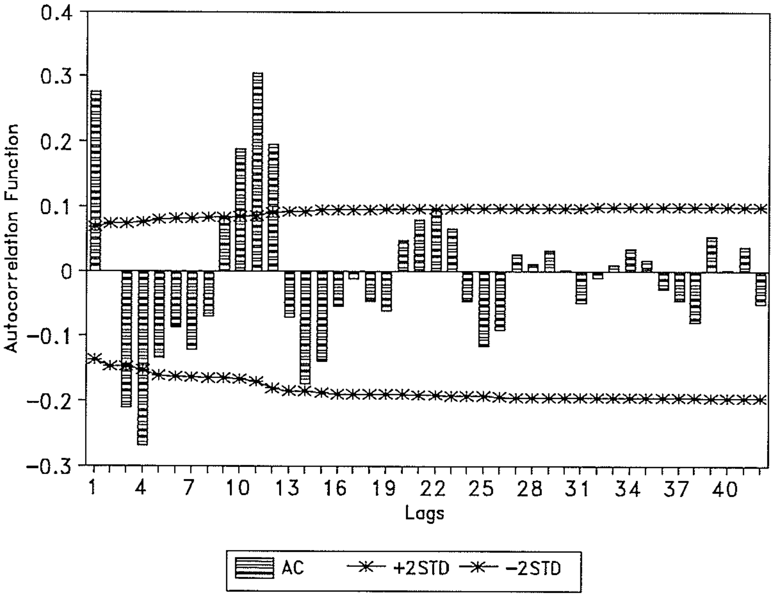


Table 2.1 Ljung-Box Test for the ARIMA Residual Autocorrelation.

To Lag	Chi Square	Degrees of Freedom	Probability
6	9.24	2	0.10
12	14.91	8	0.06
18	17.22	14	0.25
24	19.24	20	0.51
30	23.83	26	0.59

A probability value greater that 0.05 indicates that the estimated model is a reasonable representation of the data generating process for the ARIMA model.

Figure 2.3 Plot of Correlation Dimension versus log(ε)

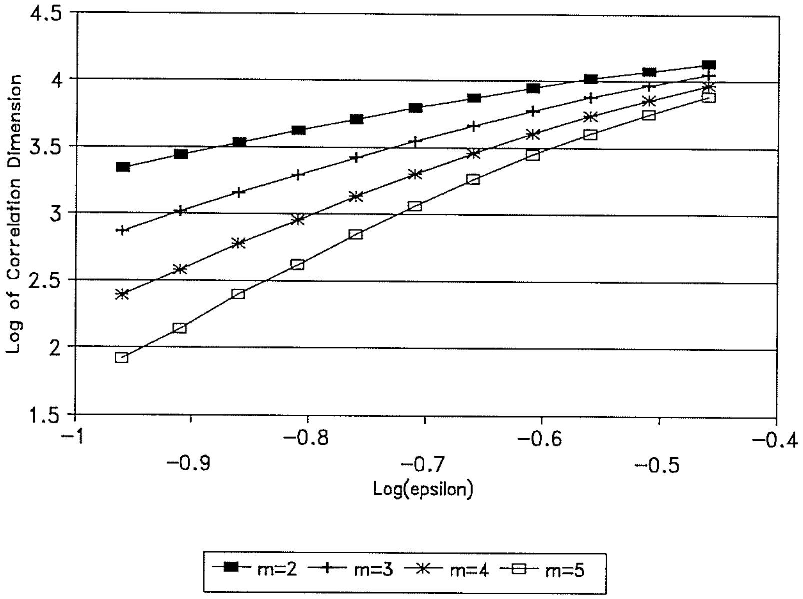


Figure 2.4 Plot of Correlation Dimension versus Embedding Dimension

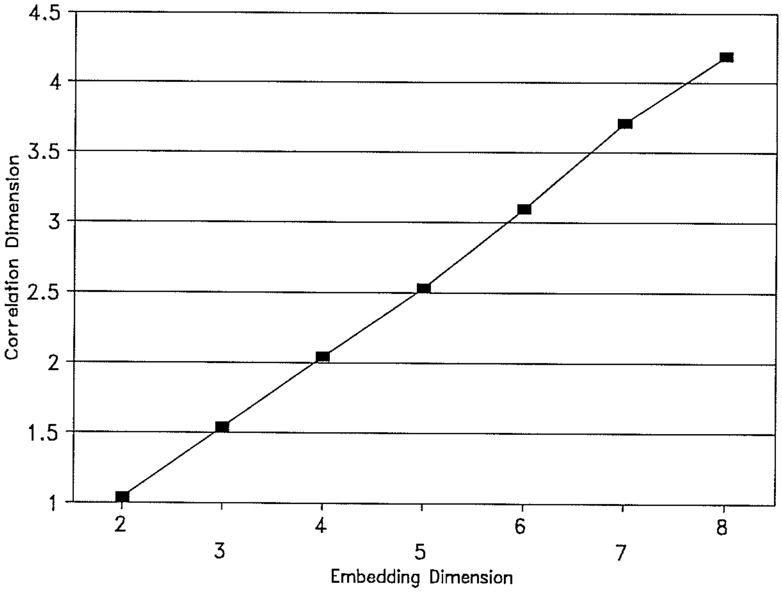


Table 2.2 BDS Statistics for Filtered Residuals of Linear Model, US Cattle Prices 1973-1990.

	ε|std				
m[a]	0.50	1.00	1.50	2.00	d_m[b]
2	0.064	0.242	0.306	0.261	1.040
3	0.025	0.260	0.489	0.503	1.540
4	0.11	0.200	0.561	0.661	2.040
5	0.007	0.147	0.563	0.747	2.530

[a] Embedding dimensions.

[b] Correlation dimensions. d_m was equal to 3.10, 3.71, and 4.19 for $m = 6$, 7, and 8, respectively.

This table shows the BDS statistics for dimensions 2 through 5 and for ε equal 0.5, 1, 1.5, and 2 standard deviations of the residuals from fitted linear model to the price series. Because none are significant, therefore, presence of nonlinear dynamics is rejected in favor of relevance of linear models to study these price series.

CHAPTER 3

NEURAL NETWORK THEORY

Introduction

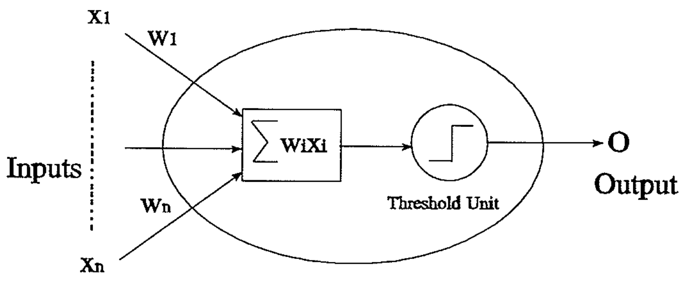
Artificial neural networks are computational structures based on the design of the human brain. The brain consists of 10^{11} processing units of different types called neurons. Neurons receive information and process it through a highly complex chemical process and send their outputs to other neurons. The information processing speed in the neurons are about one million times slower than current computer gates (Zurada). Nevertheless, the human brain is more efficient than the fastest computers in processing complex tasks such as visual images and speech recognitions because of parrallel processing with so many neurons. It appears that massive parallel processing in neural networks can perform certain tasks which conventional computers cannot perform.

The objectives of this chapter are to explain the structure of the neural networks, their capabilities, and their limitations in terms of applications to price analysis and forecasting. In this chapter, a brief history of artificial neural networks is explained followed by network topology and its various components. This is followed by discussion of neural network transfer functions, which have an important role in learning. Next, different types of learning methods are discussed followed by a section on generalization and out of sample forecasting ability of neural networks. Finally, statistical aspects of neural networks and how they are related to the conventional regression analysis are discussed in the last section.

History of Neural Networks

The first formal attempt to understand and to model the complex system of the human brain originated from the work of McCulloch and Pitts in 1943. They proposed a simple model of neurons that acts as binary threshold units (Figure 3.1).

Figure 3.1. McCulloch-Pitts Model of Brain Neurons



The inputs to this model are either zero or one. The incoming inputs in the above model are weighted and summed over the weights and are classified according to a step function. If the sum of weighted inputs is greater than or equal to some threshold level, the output would be one and zero otherwise. i.e.

$$O(t+1) = 1 \quad \text{if } \Sigma w_i x_i \geq \theta \tag{3.1}$$

and

$$O(t+1) = 0 \quad \text{if } \Sigma \mathbf{w}_i \mathbf{x}_i < \theta \tag{3.2}$$

where $\mathbf{x}_i$ is input, $\mathbf{w}_i$ is weight and θ is the threshold.

McCulloch and Pitts proved that in principal, a collection of these neurons connected by proper weights are capable of universal computations. "This means that it can perform any computation that an ordinary digital computer can, though not necessarily so rapidly or conveniently." (Hertz et al.) To address the process of arriving at the right weights in the above model, Hebb (1949) proposed a learning method now known as the "Hebbian learning rule." This rule essentially states that the weight between an input unit and its corresponding output unit should be strengthen (increase) if the two units are both firing, i.e. are on. In contrast, the weight should be decreased if either of them are off. Later, Rosenblatt (1958) used these ideas to build a trainable machine, the perceptron, which was capable of learning to classify certain tasks.

The great optimism created by the above and other works died in 1969 when Minsky and Papert proved that perceptron cannot classify nonlinear tasks. Neural networks did not receive new attention until around 1985 when the method of back-propagation (BP) was introduced by the PDP group (Rumelhart et al.). Rumelhart et al. showed that BP method can classify nonlinear tasks as well as linear ones. Since then, there has been an increase in research and applications in the area of artificial neural networks. However, research in economics, finance, and forecasting using neural networks is relatively new (Trippi 1992).

Topology of Neural Networks

Neural networks consist of a collection of inputs and processing units called neurons. The neurons are arrayed in interconnected layers known as the input layer, hidden layer(s), and the output layer, which represents the variable to be forecasted. These neurons are connected to each other through connection strengths called weights. Unlike the hidden or output layer neurons, input layer neurons only receive the information and do not process it. The system as a whole receives information, processes it through the neurons and then distributes its output. In addition to the processing neurons, there is a bias node connected to the processing units in the hidden and output layers. Input of the bias neuron is always +1 and its presence gives more flexibility to the neural network for classifying nonlinear problems.

The number of neurons in the input layer, as well as in the output layer are determined by the nature of the problem and the theory which addresses that problem. As an example, assume that the following function represents a dynamic demand function:

$$\mathbf{q}_t = f\ (\mathbf{y}_t, \mathbf{p}_{o,t}, \mathbf{p}_{s,t}, \mathbf{q}_{t-1}) \tag{3.3}$$

where $\mathbf{q}_t$ is demand at time t, $\mathbf{y}_t$ is income, $\mathbf{p}_{o,t}$ is own price, $\mathbf{p}_{s,t}$ is price of substitute, and $\mathbf{q}_{t-1}$ is demand one period lagged. To put this in a neural network context, 4 input neurons are needed to represent the inputs and one is needed for the output neuron.

As of now, there is no standard rule for the number of hidden neurons. Baily and Thompson suggests that the number of hidden neurons be at 75% of the input neurons. Figure 3.2 shows the topology of a feedforward neural network corresponding to the above model with 3 hidden neurons.

In figure 3.2, inputs are weighted and sent to the processing units in the next layer. At the processing stage, each neuron sums its weighted input, classifies it according to a transfer function, and sends its output to all hidden units in the next layer. Therefore, output of a processing neuron 'j' is given by

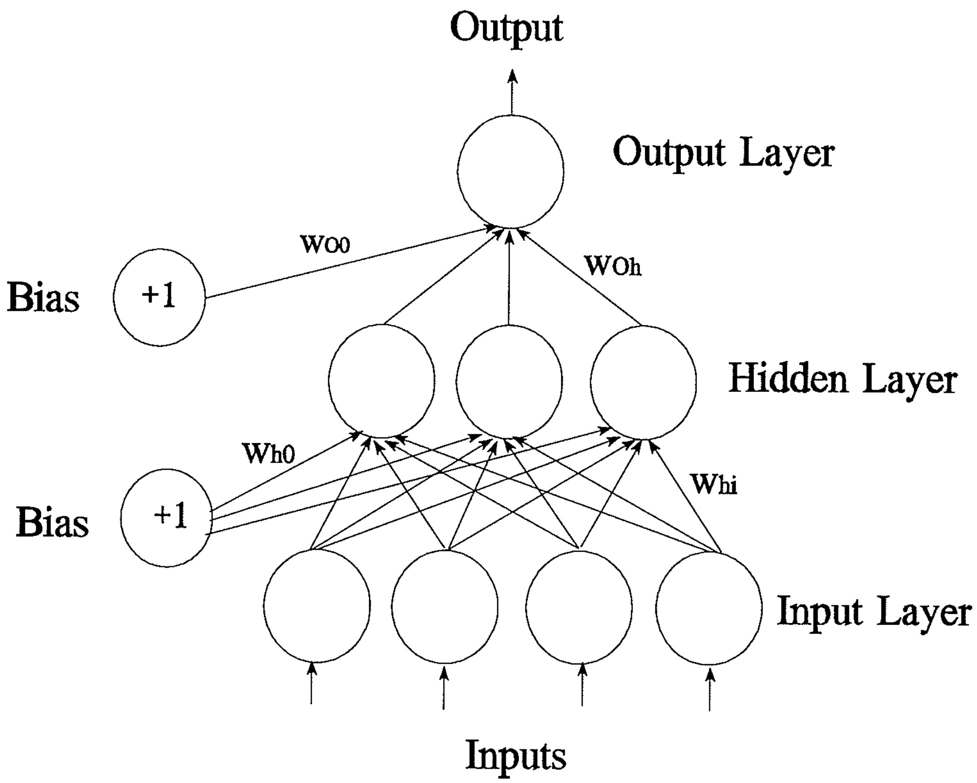
Output = f(Net) = f(sum of weighted input)

i.e.

$$\mathbf{o}_j = f(\Sigma \mathbf{w}_{ji}\mathbf{x}_i + \mathbf{w}_{j0}\beta_j) \qquad (3.4)$$

where $\mathbf{o}_j$ is the output, f is a transfer function, $\mathbf{w}_{ji}$ is weight between input i and neuron j, $\mathbf{x}_i$ is input i, and $\mathbf{w}_{j0}$ is weight between the bias neuron β_j and the processing neuron j.

Figure 3.2 A Feedforward Neural Network



Transfer Functions

Transfer functions, also known as activation, threshold, or squashing functions, are functions which determine the output of processing neurons. Transfer functions are either linear or nonlinear. In linear transfer functions, output is simply a linear multiple of the inputs. As a result, they are not useful for nonlinear mapping and classification.

Figures 3.3-3.6 show four types of popular transfer functions including step functions, ramping functions, sigmoid functions, and tanh function. The first two transfer functions are useful for inputs and outputs which have binary values, or, more generally, are discrete. For example, they can be used in discrete choice models in which the dependent variable is not continuous.

In case of continuous inputs and outputs models, sigmoid and tanh functions are common approaches. Because of their differentiability, they exhibit many desirable properties for training and minimization of the error in the network. The output of a processing unit with sigmoid function is given by

$$f(net) = f(x,w) = \frac{1}{1 + \exp(-net)} \tag{3.5}$$

where **x** and **w** are as defined above and *net* is the sum of weighted inputs.

Similarly, the output of a processing neuron with tanh transfer function is given by

$$\tanh(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{3.6}$$

Figure 3.3 Step Transfer Function

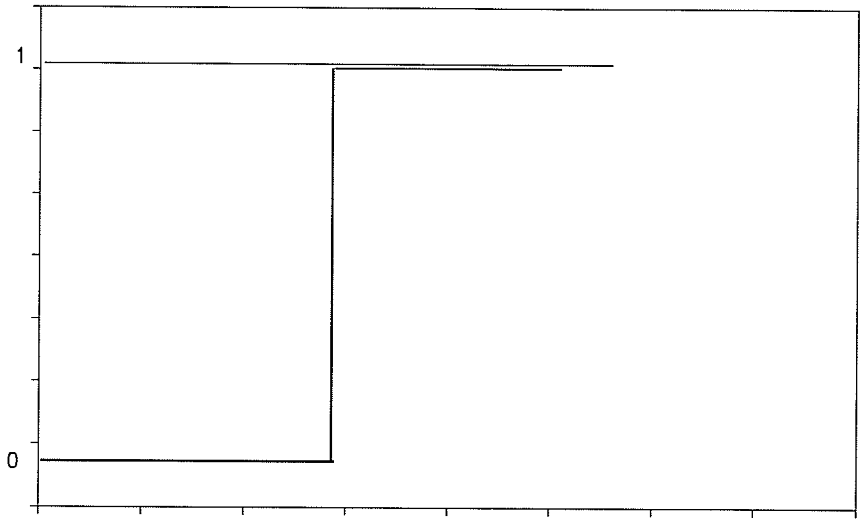


Figure 3.4 Ramping Transfer Function

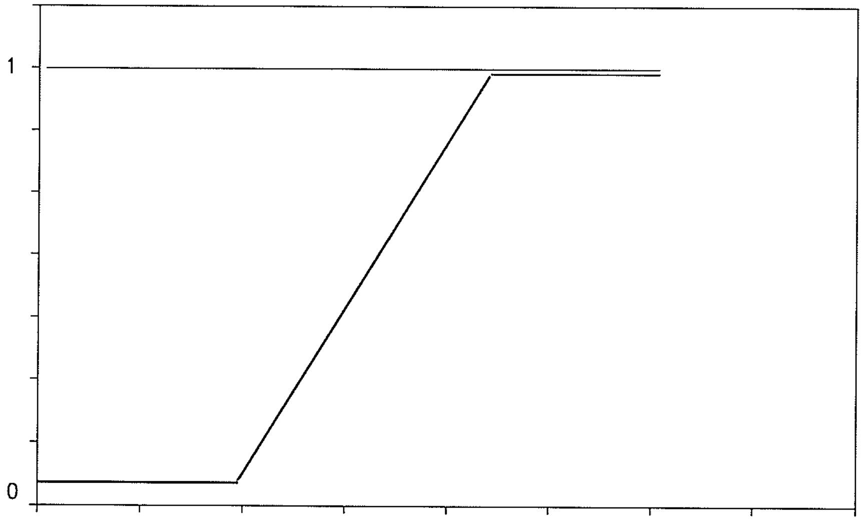


Figure 3.5 Sigmoid Transfer Function

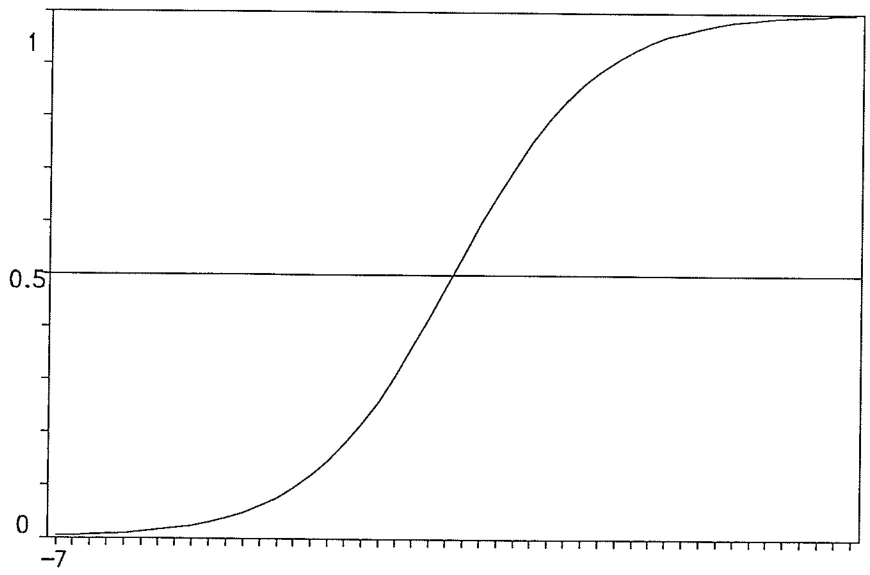
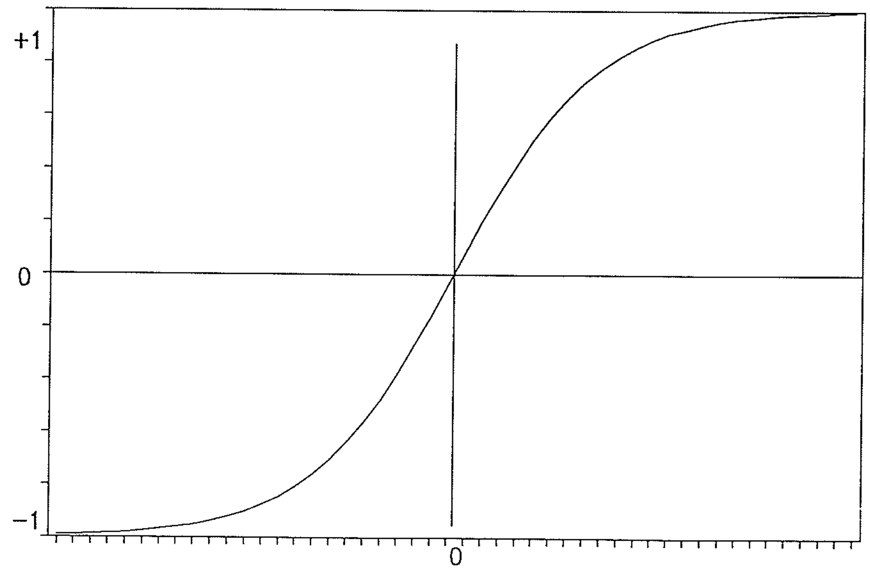


Figure 3.6 Tanh Transfer Function



Neural Network Learning

Artificial neural networks are different from conventional programs in the sense that they learn to solve problems. Learning in neural networks takes place by adjusting weights such that the created set of weights can map inputs to the output(s). There are two ways of adjusting the weights, supervised and unsupervised learning. In supervised learning, the network is presented with both input patterns and the desired output patterns. Therefore, the network can compare its results with the desired outputs and to minimize the error. In unsupervised learning, the output patterns are not defined and the network tries to classify the input patterns according to the features inherent in the inputs. Supervised neural networks are the most common for time series prediction.

Neural networks can also be classified according to the direction of flow of information from the input layer to the output layer. This flow is either feedforward, or, feedforward and feedbackward, which is called bidirectional or recurrent. In bidirectional networks the states of neurons are simultaneously determined. On the other hand, in feedforward networks, inputs are fed into the network and in passing through the system, the output is determined.

Backpropagation(BP) networks are a class of feedforward neural networks with supervised learning rules which have been successfully used for pattern recognition and classification problems. BP calculates the error signals from the last layer by backpropagating them along the path of steepest decent in the network. There are several ways of adjusting the weights based on the above calculated network errors. Standard BP employs an optimization method called the gradient decent method to map the input patterns to the output patterns. A more advanced method, conjugate gradient method, extends the original BP method by searching along all directions for minimizing the errors. Both methods are explained in the following subsections.

Standard Backpropagation Method (Gradient Descent)

Define the jth neuron's squared error as (Rumelhart et al.)

$$E = \frac{1}{2}(d_j - o_j)^2 \tag{3.7}$$

where d_j is the desired output of unit j and o_j is output produced by the network. The weight change for any neuron should be proportional to the impact of the weight from that neuron on the error. i.e.

$$\Delta w_{ji} = -\alpha \frac{\partial E}{\partial w_{ji}} \tag{3.8}$$

where α is the learning rate which determines the magnititue of weight changes and

$$-\frac{\partial E}{\partial w_{ji}} = -\frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} \tag{3.9}$$

Similarly, the contribution of weighted sum of inputs to the output error from any neuron, denoted by δ_j, is given by

$$\delta_j = -\frac{\partial E}{\partial o_J}\frac{\partial o_J}{\partial net_j} = -\frac{\partial E}{\partial o_j}f'_j(net_j) \tag{3.10}$$

and

$$\frac{\partial E}{\partial o_j} = -(t_J - o_j) \tag{3.11}$$

therefore

$$\delta_j = (t_j - o_j)f'_j(net_j) \tag{3.12}$$

For the hidden unit h connected to the neuron k, the δ_h is given by

$$\frac{\partial E}{\partial o_j} = \sum_k \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial o_j} = \sum_k \delta_k w_{kj} \tag{3.13}$$

The above relationships gives a recursive procedure for weight adjustments in the network. Practically, in multi-layer networks with continuous transfer functions and one hidden layer the above method is translated into the following steps (Zurada):

Step 1. Weights are initialized at small random values to avoid making symmetric changes.

Step 2. Training step starts by computing the output, *f(net)*.

Step 3. The difference between *f(net)* and the desired output, error, is calculated:

$$E = 1/2\Sigma(d_k - o_k)^2 \tag{3.14}$$

where E is the error, d_k is the vector of desired outputs, and o_k is the network's outputs.

Step 4. Error signal vectors δ_o and δ_h of hidden layer and output layer are calculated:

$$\delta_{ok} = 1/2(d_k - o_k)(1 - o^2_k)o_k \tag{3.15}$$

$$\delta_{hi} = 1/2h_i(1 - h_i)\Sigma\delta_{ok}w_{ki} \tag{3.16}$$

Step 5. Output layer weights are adjusted.

Step 6. Hidden layer weights are adjusted.

Step 7. If E is less than or equal to the tolerance level, the training cycle is completed. If E is greater than the specified level then a new training cycle is initiated by going to the step 2.

The learning speed in BP can be increased without leading to oscillation. This is achieved by modifying the BP learning rule to include a momentum term (McClelland and Rumelhart). The rule is

$$\Delta W_{ij}(n+1) = \varepsilon(\delta_{pi}a_{pi}) + \alpha W_{ij}(n) \tag{3.17}$$

where subscript n denotes the presentation number, ε is the learning rate (the speed by which weights are changed), δp_i is error, p_i is the output of transfer function, and α is momentum term which determines how past weight changes should affect the current weight changes.

Conjugate Gradient Method

Training a network can be described as moving down an error surface which takes place by weight adjustments during the learning phase. The magnitude of weight adjustments in the standard BP depends on the learning rate and momentum factors selected by the researchers. Selecting a high learning rate may cause the network to jump from one side of the error surface to the other side and never reach the minimum point. On the other hand, a low learning rate slows down the training and may cause the network to be trapped in nearest minima. The conjugate gradient method used in this study is an improvement over gradient descent method in the sense that it explores the minimization of the network error in all possible directions and guarantees the convergence to the local minima. Moreover, it does not require the researcher to set the learning rate and momentum factors.

The conjugate gradient method (CG) constructs a set of n directions which are all conjugate to each other along the minimization direction **u** such that minimization subdirections u_i's are not interfering. The Fletcher-Reeves version of CG method is as follows (Krose and Smagt). Consider the function $f(x)$ to be minimized as approximated by the Taylor series expansion:

$$f(x) = f(P) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \, \partial x_j} x_i x_j \tag{3.18}$$

or

$$f(\mathbf{x}) \approx (1/2)\, \mathbf{x}'\, A\, \mathbf{x} - b'\, \mathbf{x} + c \tag{3.19}$$

where ' denotes transpose and $b \equiv -\nabla f|_p$ and A is a symmetric positive definite Hessian matrix of the second partial derivatives of $f(\mathbf{x})$ at p. A change in $\mathbf{x}$ results in the change in the gradient of $f(\mathbf{x})$ as

$$\delta(\nabla f(\mathbf{x})) = A\ (\delta\mathbf{x}) \tag{3.20}$$

When $f(\mathbf{x})$ is to be minimized along a direction $\mathbf{u}_i$, the perpendicularities of the gradients ensure that moving along $\mathbf{u}_{i+1}$ does not impair minimization along $\mathbf{u}_i$. In such case $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ are said to be conjugate. This is true when the following equation holds:

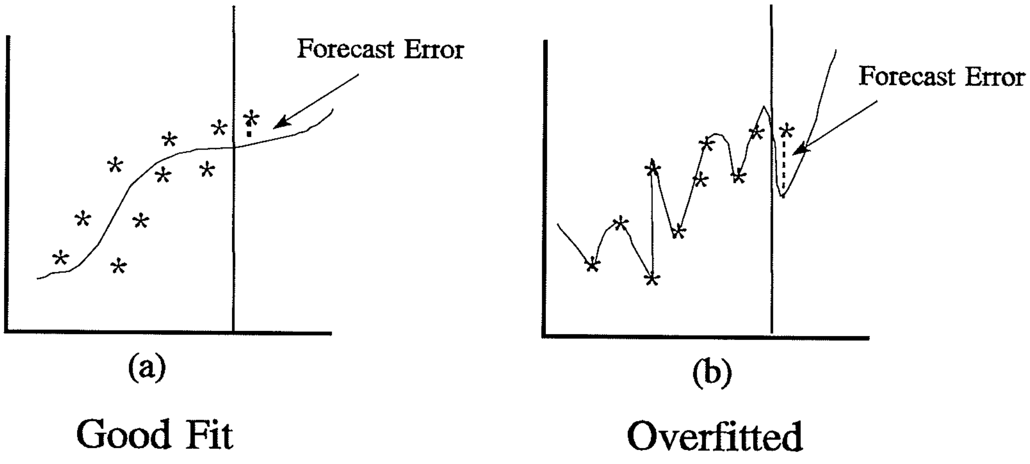
$$\mathbf{u}_i = \mathbf{u}_i'(\mathbf{g}_{i+1} - \mathbf{g}_{i+2}) = \mathbf{u}'_i\delta(\nabla f(\mathbf{x})) = \mathbf{u}'_i A \mathbf{u}_{i+1} \tag{3.21}$$

Starting the minimization at any point results in new points until all possibilities are exhausted and the minimization is completed.

Generalization and Out of Sample Prediction

The power of neural networks lies in their ability to generalize the regularities in the training set to unseen patterns. Generalization is the idea that a model based upon a sample of a population should be suitable for forecasting the general population. For time series data, a sample or part of the data is typically used to fit the model. If procedures are correct, the model can then be used to forecast the general population which is called 'out of sample prediction' and is the data in the future to be forecasted. Network training can be described as finding a matrix of weights through sample training that when applied to unseen patterns out of sample in the population, will produce the desired outputs. The generalization ability of multi-layer feedforward networks comes from the hidden units in these networks. Whether this potential can be realized depends on the quality of data and parameters of the network selected at the training phase. Too small a number of hidden units or too large a tolerance rate prevents the network from properly learning the patterns. On the other hand, too many hidden units or too small a tolerance rate at the training phase result in the network memorizing patterns rather than learning them for generalization. Memorizing patterns is similar to overfitting a curve to a set of points. Figure 3.7 depicts examples of a good fit and an overfitted curve to a set of data.

Figure 3.7 Out of Sample Forecasts by Appropriate versus Overfitted Models



The above figure shows that although all points in (b) are captured by the curve, it lost the power of capturing the trend in the data. As a result, it will produce a higher error in predicting point a point out of sample, compared with (a).

Statistical Aspects of Neural Networks

In most real world problems the true relationships between variables are not known and so attempts are made to find models which best fit the data. Supervised neural networks are similar to conventional regression analysis in the sense that both try to find the best fit for the function Y = f(X), by minimizing the model's error given by

$$E = Y - f(X) \tag{3.22}$$

Linear regression models may be viewed as a feedforward network with no hidden layers and linear transfer functions in the output neurons. Removing the hidden layer in the figure 3.2 will create a network that corresponds to a linear regression model with four explanatory variables and one output. The output of this network is given by:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \tag{3.23}$$

where x_i are inputs, β_i (i=1,..,4) are inputs to output weights, and β_0 is the bias term's weight. Since the bias always has a value of 1, therefore, its weight has the same role as the constant term in a regression model. Similarly, networks with hidden layers resemble nonlinear regression models and for one hidden layer are given by

$$y = G(\sum_{j=0}^{h} \beta_j G(\sum_{i=0}^{m} \gamma_{ji} x_i)) \equiv f(x,\phi) \tag{3.24}$$

where G is the transfer functions in the hidden and the output layers in the neural network, h is the number of neurons in the hidden layer, m is the number of input units, β_j are the connection weights between output neurons to hidden neurons, γ_{ji} are the connection weights between hidden neurons and input neurons, and ϕ denotes all connection weights in the network.

A difference between neural networks and regression models is that in the latter case, certain assumptions regarding the distribution of error terms must hold. Moreover, in regression models, whether linear or nonlinear, it is implicitly assumed that inputs are mapped into the outputs via the selected functional forms. But in the neural networks, these assumptions are not required. Several studies (Hetch-Nielsen; Hornik et al.; and Stinchcombe and White) show that feedforward networks with at least one hidden layer belong to a class of flexible functional forms which do not make any assumption about the distribution of variables concerned or the underlying functional form in the data generating process. They showed that the above networks with a large enough number of hidden units, are accurate universal approximators to any function of **x**.

A disadvantage of neural networks in comparison with regression models is their lack of explanation. Regression analysis can identify the contribution of each individual input in determining the output and also can give some measures of confidence about the estimated coefficients. On the other hand, currently there is no theoretical or practical way of accurately interpreting the weights in neural networks. Relatively lower value weights do not necessarily mean that the corresponding inputs have lower contribution to the output and vice versa (Masters).

Aside from assigning important judgements to the inputs with low or high connection weights, inputs with insignificant contribution for determining the output must eventually have weights which tend to zero. But, how close to zero "is insignificant", is a matter of statistical examination. This examination has important implications for optimal network architecture especially when the network or the training patterns are large and complex. For example, speed of training can be increased by removing insignificant inputs or hidden units whose weights are not significantly different from zero. The above problem can be addressed by defining the goal of neural network training as finding a set of optimal weights which minimize a performance measure λ over an entire weight space (White 1992). i.e.,

$$\min \hat{\lambda}(w) \equiv n^{-1}\sum_{t=1}^{n}(y_t - f(x_t,w))^2/2 \qquad (3.25)$$

where n is the training sample size, y_t is the desired output, f is the network output, and the summation denotes the loss function from the error minimization problem give by:

$$\mathbf{e}(\mathbf{y}, \mathbf{o}) = (\mathbf{y} - \mathbf{o})^2 \qquad (3.26)$$

where **y** is the desired output vector and **o** is vector of the network's output.

White(1992) showed that the solution to 3.25 has a limiting multivariate normal distribution of

$$\sqrt{n}(w_n - w^*) \sim N(0, C^*) \qquad (3.27)$$

where w_n is the network's weights vector, w^* is the true optimal weights, and C^* is the asymptotic covariance matrix of the network's weights after training. From equation 3.27 it follows that

$$n w_n' S' (S \hat{C}_n S')^{-1} S w_n \;\rightarrow\; \chi_h^2 \tag{3.28}$$

where S is a h x i matrix of random numbers; h is the number of hidden units and i corresponds to the number of input of hidden neurons under pruning. Equation 3.28 states that under the null hypothesis of irrelevant input neurons or hidden neurons, the above scaler has a χ^2 distribution with h degrees of freedom.

Conclusion

Neural networks are powerful methods which possess flexible functional form properties. Therefore, they are useful models for forecasting purposes especially when the underlying data generating processes are unknown. However, the appropriate design of the network's architecture and learning rules are crucial for obtaining satisfactory results. Despite their powerful generalization and out of sample properties, neural network design and architecture suffers from a lack of detailed explanation and comprehensive theory. Therefore, the final results are somewhat subjective and involve some trial and error. Also, for complex and large problems, the computation costs are an important factor. However, statistical theory can offer some suggestions for designing optimal network architecture and reducing the above costs.

CHAPTER 4

A COMPARISON OF NEURAL NETWORK AND TIME SERIES MODELS FOR FORECASTING COMMODITY PRICES: MODEL I

Introduction

Price forecasting is an integral part of commodity trading and price analysis. Quantitative accuracy with small errors along with turning point forecasting power are both important in evaluating forecasting models. Numerous studies have found that univariate time series, such as Box-Jenkins ARIMA models, are as accurate as larger econometric models such as vector autoregressive models [Bessler and Brandt; Dorfman and McIntosh; and Harris and Leuthold]. Recent developments in the study of neural networks show that feedforward neural networks are nonlinear universal mapping structures that can approximate any arbitrary function (Cybenko; Hecht-Nielsen; Hornik et al.). Therefore, such a flexible model may be superior to ARIMA models especially for nonlinear time series data. But even if data is linear as was found in chapter two, ARIMA models may be at a disadvantage to neural network models since they require restrictive regression assumptions regarding the error term, including the zero mean and constant variance and the assumption of being independent of explanatory variables, in order to be valid for modelling the price behavior. Neural network models are less sensitive to violations of these assumptions.

The objective of this study is to examine whether neural network models can outperform traditional ARIMA models in forecasting commodity prices. Specifically, a neural network is used to forecast US cattle prices. We then compare the results with the ARIMA model as a bench mark. The remainder of the chapter is organized as follows. The traditional univariate time series approach to forecasting is described followed by the neural network architecture and analysis. Evaluation methods for comparing the two forecasting approaches and data and forecast procedures are then discussed, with results obtained from the ARIMA and neural network forecasts. In conclusion, overall evaluation and comparison of two techniques is presented.

ARIMA Time Series Model

Dorfman and McIntosh suggest that "structural econometrics may not be superior to time series techniques even when the structural modellers are given the elusive true model." Therefore, a common approach to forecasting is the Box-Jenkins time series approach which is used here for comparison with neural networks. It has attracted researchers because it is a parsimonious approach which can represent both stationary and non-stationary stochastic processes (Harvey). It is used here to build an Autoregressive Integrated Moving Average model (ARIMA) which adequately represents the data generating process. The basic Box-Jenkins model has the following form:

$$y_t = \alpha_0 + \Sigma\alpha_i y_{t-i} + \Sigma\beta_j e_{t-j} \quad i=1,..p \text{ and } j=0,1,....,q \tag{4.1}$$

where y_t is a stationary stochastic process with non-zero mean , α_0 is constant term, and e_t is a white noise disturbance term. The second and third terms in the right hand side of equation 4.1 are autoregressive and moving average parts of the model. Equation 4.1 is denoted by ARIMA(p,d,q) in which "d" stands for the number of differencing performed on y_t before estimating the above model. Box-Jenkins methods involves the following four-step iterative cycle:

(i) model identification,

(ii) model estimation,

(iii) diagnostic checking, and

(iv) forecasting with the final model

Forecasting with the estimated model is based on the assumption that the estimated model will hold in the horizon for which the forecasts are made. The AR part of the model indicates that the future values of y_t are weighted averages of the current and past realizations. Similarly, the MA part of the model shows how current and past random shocks will affect the future values of y_t.

The Neural Network Approach

Neural networks, or parallel distributed processing, are computational structures modeled on the gross structure of the brain (Hecht-Nielsen 1988). Neural networks are powerful methods for pattern recognition, classification, and prediction. As far its application to economics is concerned, they have been primarily used to address financial economics problems. Typical applications in finance have included mortgage risk assessment, economic prediction, risk rating of exchange-traded fixed-income investments, portfolio selection/diversification, simulation of market behavior, index construction, and identification of explanatory economic factors (Trippi and DeSienco). For example, the

US government in 1989 "embarked on a five-year, multi-million dollar program for neural network research, but financial services organizations have been the principal sponsors of research in neural network applications." (Trippi and DeSieno).

There are a number of studies in which, along with conventional methods, neural networks are used to address financial economic problems (c.f., Schoneburg; Kamijo and Tanigawa; Kimoto and Asakawa; Stephens et al; Odom and Sharda; Surkan and Singleton; Hoptroff et al; Tam and Kiang; Trippi and DeSieno). For instance, Surkan and Singleton found that the neural network model outperforms the "multivariate discriminate analysis" (MDA) for bond rating. In their study, the neural network model provided 88% correct classification compared with, at most, 56.6% by the MDA method. Odom and Sharda set up both MDA and neural network models for predicting bankruptcy for various companies listed in the Wall Street Journal. They also found that neural networks were over 20% more accurate that the MDA.

Halquist and Schmoll set up 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. Trippi and DeSieno 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%, even after inclusion of brokerage charges.

Evaluation Methods

Three criteria will be used to make comparison between the prediction power of time series model and the neural network model. The first is mean squared error, MSE, which measures the overall performance of a model. The formula for MSE is

$$MSE = \frac{1}{T}\Sigma(P_t - A_t)^2 \tag{4.2}$$

Where P_t is the predicted value for time t, A_t is the actual value at time t, and T is the number of predictions.

The second criterion is the absolute mean error, AME. It is a measure of average error for each point forecast made by the two methods. AME is given by

$$AME = (1/T)\Sigma|P_t - A_t| \tag{4.3}$$

While MSE and AME are good measures of deviation of predicted values from the actual values, they do not say much about the power of models in predicting the turning points. For many traders and analysts the market direction and turning points are as important as the value forecast itself. "In these markets, money can be made simply by knowing the direction in which the series will move" (McIntosh & Dorfman). A correct turning point forecast requires:

$$\text{sign}(P_t - A_{t-1}) = \text{sign}(A_t - A_{t-1}) \tag{4.4}$$

Ability of a model to forecast the turning points can be measured by a third method developed by Cumby and Modest which is a version of Merton's test. Merton's test is as follows:

define a forecast variable F_t and an actual direction variable A_t such that

$$A_t = 1 \text{ if } \Delta A_t > 0 \text{ and } A_t = 0 \text{ if } \Delta A_t \leq 0 \tag{4.5}$$

$$F_t = 1 \text{ if } \Delta P_t > 0 \text{ and } F_t = 0 \text{ if } \Delta P_t \leq 0 \tag{4.6}$$

where ΔA_t is the amount of change in actual variable between time t-1 and t and ΔP_t is the amount of change in the forecasting variable for the same period.

The probability matrix for the forecasted direction of changes in the actual value conditional upon the direction of changes in the forecasting variable F_t is

$$P_1 = \mathrm{Prob}[F_t = 0 | A_t = 0] \tag{4.7}$$

$$1 - P_1 = \mathrm{Prob}[F_t = 1 | A_t = 0] \tag{4.8}$$

$$P_2 = \mathrm{Prob}[F_t = 1 | A_t = 1] \tag{4.9}$$

$$1 - P_2 = \mathrm{Prob}[F_t = 0 | A_t = 1] \tag{4.10}$$

In other words, (4.7) and (4.9) are the probability that the forecasted direction have actually occurred and (4.8) and (4.10) are probabilities of wrong forecasts.

By assuming that the magnitude of changes in F_t and A_t are independent, Merton(1981) showed that a necessary and sufficient condition of market timing ability is that

$$P_1(t) + P_2(t) > 1 \tag{4.11}$$

i.e. the forecaster on average has to be right in more than half of the time the forecasts are made. So the null hypothesis to be tested is

$$H_0 \ : \quad P_1 + P_2 - 1 \leq 0$$

vs

$$H_1 \ : \quad P_1 + P_2 - 1 > 0$$

Cumby and Modest showed that the above hypothesis can be tested through the regression equation:

$$X_t = \alpha_0 + \alpha_1 A_t + \varepsilon_t \tag{4.12}$$

where:

X_t is the change in actual price from previous period at time t

A_t is the realized price direction variable defined in (4.5)

ε_t is the error term,

$\alpha_1 = P_1 + P_2 - 1$,

and an α_1 significantly different from zero is needed to prove the forecasting ability.

Data and Forecast Procedure

Monthly commodity prices ($/cwt) of US cattle (900-1100 lb) traded in Omaha are used to test the prediction power of the two approaches. Data are obtained from the CRB Commodity Year Book, various issues, and cover the period 1973-1987. Monthly data from 1973 through 1986 are used to estimate the time series model. The estimated coefficients are then used to forecast cattle prices out of sample and twelve steps ahead without updating. The forecasted values are then compared with the actual prices for 1987. The same monthly data, 1973-1986, are also used for training and testing the neural network and to predict monthly cattle prices out of sample in 1987.

ARIMA Time Series Results

Identification and estimation results

Results of the identification step suggest that the ARIMA model below can best represent the price behavior for the period of study. The maximum likelihood estimate of the model produced:

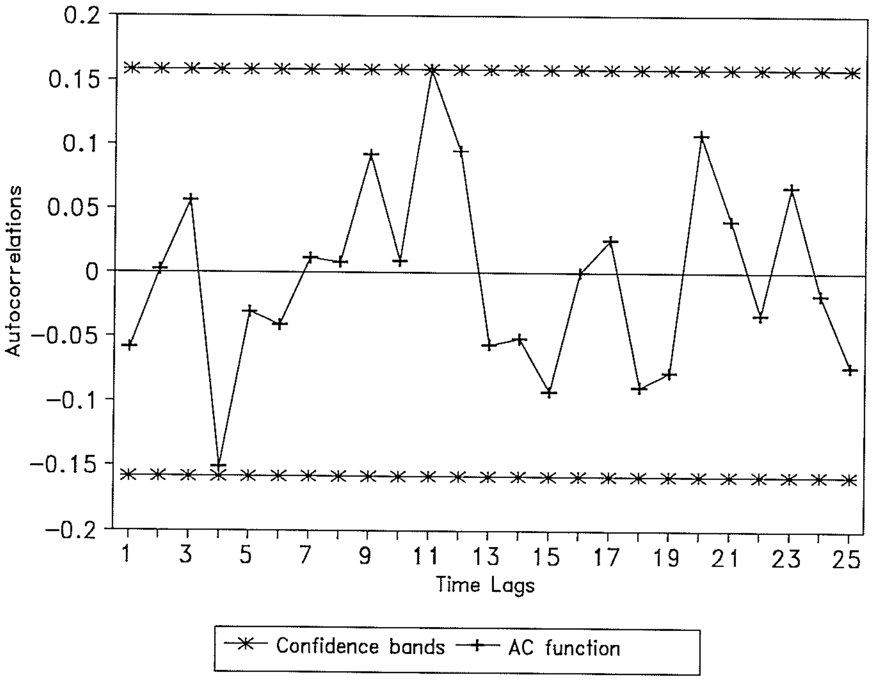
$$y_t = \underset{(0.59)}{0.09644} + \underset{(3.49)}{0.25228y_{t-1}} - \underset{(-3.61)}{0.27087y_{t-3}} - \underset{(-2.48)}{0.18760y_{t-7}} \tag{4.13}$$

T statistics in parentheses show that all coefficients are significant other than the constant term. However, since the mean is not subtracted from differenced data, therefore, the constant term is kept in the model for the forecasting step.

Diagnostic checking

Sample autocorrelation plots of estimated residuals fall within the two standard error bands (figure 4.1) and suggest remaining serial dependence is minimal.

Figure 4.1 Plot of Residuals Autocorrelation after Differencing



This indicates a white noise error term in the estimated model and proper modelling procedure in that all information has been extracted from the error terms. Ljung-Box test statistics reported in Table 4.1 show that all estimated probabilities are greater than 10%. Therefore, equation 4.13 can be considered as an acceptable representation of data generating process for the ARIMA model.

Forecasting results

Results of time series forecasts using equation 4.13 are shown in Table 4.2. Results show mean squared errors of 36.79 for the ARIMA model. Absolute mean errors indicated that forecasted prices by ARIMA were as much as 5.33 different from the actual prices. Results also show a 8.1% forecasting error.

Neural Network Results

Network architecture

A multi-layer feedforward neural network with one hidden layer was set up (figure 4.2). To make the comparison with the time series models, twelve lags of the data series were assumed to be sufficient as inputs to the network to forecast prices. At the training stage, various numbers of neurons in the hidden layer were examined. The best in sample results were produced by nine neurons in the hidden layer. The output layer had one neuron which was set up to output the prices. With the above specifications, it took 2666 iterations to train the network.

Forecasting with the trained network

To do forecasting out of sample, 12 months of 1986 prices were fed into the trained network as inputs to forecast the first month of 1987 out of sample. To forecast the second month of 1987, the forecasted price for the first month replaced one of previous inputs, the first one, and a new forecast was obtained without training the network. i.e. at the forecasting stage no training took place and no prices from actual 1987 data, other than the network's own forecasts were fed into the network. This process of forecasting and recursively substituting continued until all 12 months out of sample forecasts of 1987 prices were obtained.

Results of the neural network forecasts are shown in Table 4.2. Results show that the MSE was 7.99, which is over 450% lower than 36.79 for the ARIMA.

Evaluation and Comparison

Quantitative evaluation

In terms of quantitative forecasts, Table 4.2 results clearly show that the ARIMA model is outperformed by neural network. The absolute mean error for the ARIMA is 5.33 while this same error measure is much lower at 1.84 for the neural network forecasts. In percentage terms, the last two columns of Table 4.2 show that neural network errors were three times lower than ARIMA.

Turning point evaluations

Plots of the two forecasts along with the actual prices are shown in Figure 4.3. A graph of the actual prices shows that in 1987 there were four turning points in months 1, 5, 8, and 10. Figure 4.3 shows that only one of the four turning point, month 5, was predicted by ARIMA. On the other hand, the neural network was able to predict almost all of them. However, the neural network made a mistake by predicting one additional turning point between months 6 and 8 which did not materialize.

The formal statistical test of turning points for both models is performed by estimating equation 4.12 above and results, (after adjusting for autocorrelation), are shown in Table 4.3. The t ratio of slope coefficient, α_1, for the ARIMA model shows that it is not statistically different from zero. This implies that for the period of 1987 the ARIMA model had extremely limited turning point forecasting power. On the other hand, for the neural network predictions, α_1 is highly significant and different from zero. This evidence supports the turning point forecasting power of neural network in addition to accurate price level forecasts.

Conclusion

The traditional view in economics is that market prices are random and that past prices cannot be used as a guide for the price behavior in the future. This study used a neural network approach by examining commodity prices using US cattle prices. Results show successful price forecasting from past prices that could not be captured by the traditional time series models. They indicate that feedforward neural network models have the ability to both identify and forecast the time series examined here with quantitative and qualitative accuracy. As well, the neural network forecasts were considerably more accurate than those of the traditional ARIMA models, which was used as a bench mark. Finally, the neural network results conform to the theoretical proofs that a feedforward neural network with only one hidden layer can precisely and satisfactorily approximate any continuous function.

Figure 4.2 Neural Network Price Forecasting Model

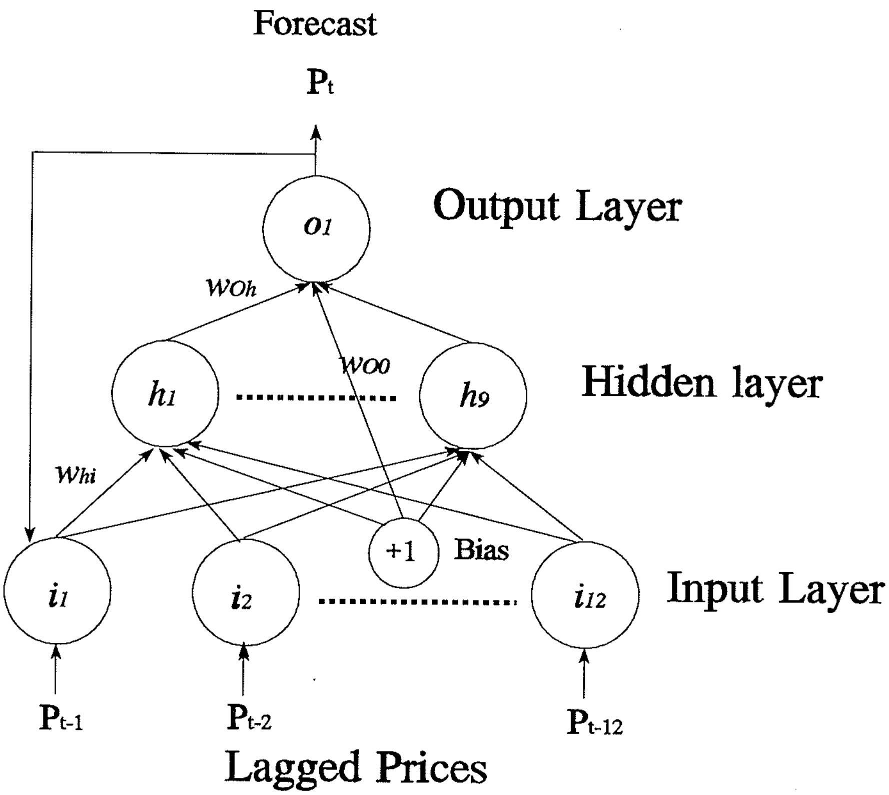


Table 4.1 Ljung-Box Test for the ARIMA Residual Autocorrelation, 1974-1986.

To Lag	Chi Square	Degrees of Freedom	Probability
6	5.47	3	0.14
12	12.92	9	0.17
18	17.10	15	0.31
24	21.90	21	0.41
30	23.89	27	0.64

A probability value greater that 0.05 indicates that the estimated model is a reasonable representation of the data generating process for the ARIMA model.

Table 4.2 Results of US Monthly Cattle Price Forecasts by the ARIMA and Neural Network, 1987.

Month	Actual	Forecasts		Sq. Errors		Abs. Errors		% Errors	
		ARIMA	NN*	ARIMA	NN	ARIMA	NN	ARIMA	NN
1	58.79	59.7	59.6	0.90	0.66	0.95	0.81	1.6%	1.4%
2	61.02	58.6	59.9	6.05	1.37	2.46	1.17	4.0%	1.9%
3	61.58	58.7	61.4	8.29	0.04	2.88	0.2	4.7%	0.3%
4	66.3	58.8	62	56.25	18.75	7.5	4.33	11.3%	6.5%
5	70.66	59.2	64	131.10	44.89	11.45	6.7	16.2%	9.5%
6	68.83	59.1	63.6	95.65	26.94	9.78	5.19	14.2%	7.5%
7	65.8	59.4	65.6	40.70	0.03	6.38	0.18	9.7%	0.3%
8	64.5	59.5	64.7	24.70	0.03	4.97	0.16	7.7%	0.2%
9	64.81	59.9	65.3	23.72	0.23	4.87	0.48	7.5%	0.7%
10	64.81	60	63.9	22.75	0.85	4.77	0.92	7.4%	1.4%
11	64.2	60.1	63.5	16.56	0.44	4.07	0.66	6.3%	1.0%
12	63.93	60.1	62.6	14.82	1.72	3.85	1.31	6.0%	2.0%
Mean				36.79	7.99	5.33	1.84	8.1%	2.7%

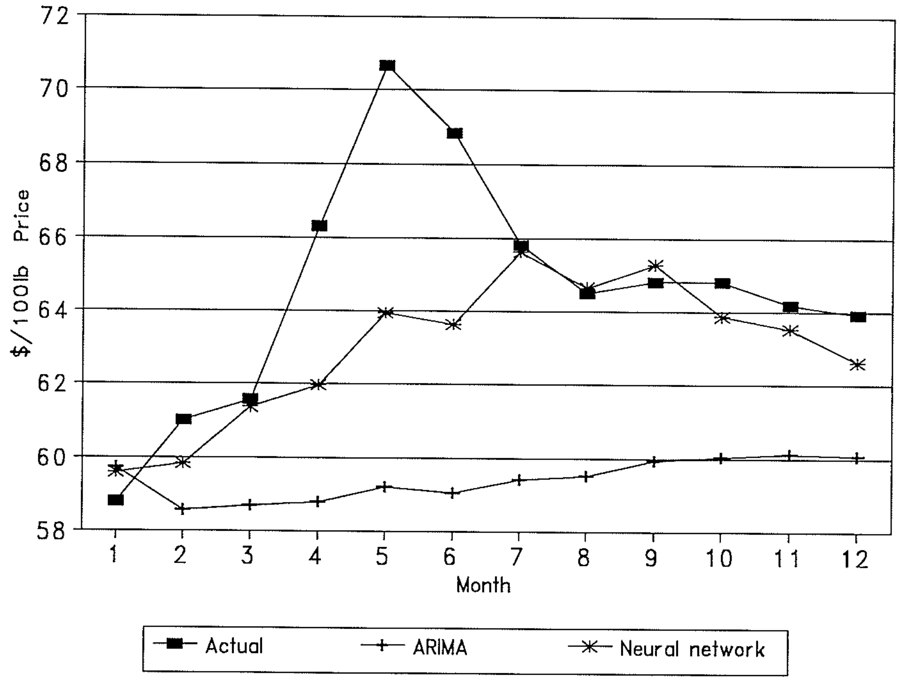
*Neural Network

Table 4.3 Results of Merton's Test of Turning Point Forecasting Power for the ARIMA and Neural Network, 1987.

	α_0	α_1	R^2
ARIMA Model (t ratio)	-0.23 (-0.39)	0.85 (0.62)	0.03
Neural Network (t ratio)	-0.84* (-2.93)	2.37* (2.23)	0.28

*Significant at 5 percent level; $X_t = \alpha_0 + \alpha_1 A_t + \varepsilon_t$ is turning point test equation.

Figure 4.3 Price Forecasts by the ARIMA and Neural Network, 1987.



CHAPTER 5

A COMPARISON OF NEURAL NETWORK AND TIME SERIES MODELS FOR FORECASTING COMMODITY PRICES: MODEL II

Introduction

Regression models are developed on the assumption that variables concerned are stationary (Judge et. al.). Violation of this assumption means that some of the desirable properties of ordinary least squares are no longer guaranteed. On the other hand, most economic variables are not stationary (Dolado et al. 1990) and differencing has been a popular way to make time series stationary. However, Engle and Granger (1987) point out that differencing may cause loss of some information contained in the level prices which may prove useful in explaining the behavior of the price series.

A neural network does not necessarily require this assumption of stationarity, so does not suffer from this restriction. Therefore, the objective of this section is to compare the ARIMA model with a neural network model which uses the same set of data, but uses both actual and differenced data as inputs. Since the neural network estimation still includes the trend, no price information is lost by differencing, yet an input variable of differences is also included to capture changes from the trend.

Data and Procedure

Monthly cash prices ($/cwt) of US cattle (900-1100 lb) traded in Omaha are used to test the prediction power of the two approaches. Data are obtained from the CRB

Commodity Year Book, various issues, and cover the period 1974-1990. Monthly data are used to estimate the three time series models from 1974 through 1987, 1988, and 1989 respectively. The estimated coefficients from each of the above three models are then used to forecast cattle prices out of sample and twelve steps ahead without updating. The forecasted values from the models fitted over 1974-1987, 1974-1988, and 1974-1989 are then compared with the out of sample actual prices for 1988, 1989, and 1990 respectively.

ARIMA Time series Results

Identification and Estimation Results

Results of the identification step suggest that the ARIMA models below can best represent the price behavior over the various periods. The maximum likelihood estimate of the three models are:

1974-1987 model

$$\begin{array}{lccccr} y_t = & 0.12646 & +\,0.25788y_{t-1} & -\,0.26293y_{t-3} & -\,0.17033y_{t-7} & (5.1) \\ & (0.79) & (3.69) & (-3.63) & (-2.34) & \end{array}$$

1974-1988 model

$$\begin{array}{lcccccr} y_t = & 0.14611 & +\,0.21051y_{t-1} & -\,0.21857y_{t-3} & -\,0.21488y_{t-4} & -\,0.211y_{t-7} & (5.2) \\ & (1.18) & (3.09) & (-3.10) & (-3.01) & (-2.97) & \end{array}$$

1974-1989 model

$$\begin{array}{lccccr} y_t = & 0.1572 & +\,0.26187y_{t-1} & -\,0.26205y_{t-3} & -\,0.16774y_{t-7} & (5.3) \\ & (1.06) & (3.99) & (-3.84) & (-2.44) & \end{array}$$

T statistics in parentheses show that all coefficients are significant other than the constant terms. However, since the means are not subtracted from differenced data, therefore, the constant terms are kept in the models for the forecasting steps.

Diagnostic Checking

Plots of autocorrelation of estimated residuals were inside the two standard error bands. This indicates a white noise error term in the estimated model and proper modelling procedure in that, by ARIMA's standards, all information has been extracted from the error terms. Ljung-Box test statistics reported in Table 5.1 show that all estimated probabilities are greater than 1%. Therefore, equation 5.1 to 5.3 can be considered as the acceptable representation of data generating processes for the ARIMA models.

Forecasting Results

Results of Box-Jenkins forecasts using equation 5.1 to 5.3 are shown in Table 5.2. Results show root mean squared errors of 4.39 for 1988, 2.10 for 1989, and 2.30 for 1990 forecasts for the ARIMA model. Absolute mean errors indicated that forecasted prices by ARIMA were on average, \$3.56, \$1.65, and \$1.77 for 1988, 1989 and 1990.

Neural Network Results

A multi-layer back-propagation neural network with one hidden layer was set up (Figure 5.1). To make the comparison with the time series models, twelve lags of both actual and differenced prices were assumed to be sufficient as inputs to the network. The hidden layer was set up with 18 neurons, 9 for actual prices and 9 for differenced prices. This is 75% of the number of neurons in the input layer following Baily and Thompson's suggestion. The output layer had two neurons which were set up to output the current cattle prices in level and difference formats. The same neural network specification were used for each model over the three years.

For forecasting, a moving window of twelve month lagged prices was created. For the first step the weight matrix was applied to the twelve month cattle prices in 1987 to obtain a price forecast for the first month of 1988. Without the network weights being changed, the forecasted prices then became part of inputs to the network and the last prices in the previous input series were dropped. The above process of forecasting and back substitution continued until all twelve forecasts were made. Results of neural network are shown in Table 5.2. Results show root mean squared errors of 1.99 for 1988, 2.98 for 1989, and 1.17 for 1990. Absolute mean errors indicate that forecasted prices by neural network were on average, $1.53, $2.17, and $0.98 for 1988, 1989 and 1990 respectively.

Evaluation and Comparison

Quantitative Evaluation

In terms of quantitative forecasts, Table 5.2 results show that in 1988 and 1990 the neural network outperformed the ARIMA model. In terms of forecast error, in the above period, the neural network outperformed the ARIMA model by 97% to 233%. In terms of error variance, the neural network forecasts' variances were 4 to 5 times lower than ARIMA's forecasts. In 1989, however, ARIMA forecasts errors were on average $0.52 lower than that of the neural network and also had lower forecast error variance.

Turning Point Evaluations

The formal statistical test of turning points for both models is performed by estimating equation 4.12 in the previous chapter and results are shown in Table 5.3. The t ratios of slope coefficients, α_1, show that the ARIMA model did not have significant turning point forecasting power. The negative signs in 1988 and 1989 also imply that the ARIMA models were even giving wrong signals about the turning points in the above periods. In contrast, for the neural network predictions the α_1 was 0.80 in both 1988 and 1989 and significantly different from zero. The value of slope parameter shows that the neural network was correct 80% of the time. In 1990, forecasts of neither the ARIMA nor the neural network had significant turning point forecasting power. This may be expected if prices were truly stochastic in 1990.

Conclusion

The purpose of this chapter was to examine the performance of the neural network when both actual and differenced prices are fed into the model and compare it to traditional Box-Jenkins ARIMA methods as a benchmark. The advantage of this model may be that the price levels pick up the trend in the data while the price differences capture the prices changes around the trend. In contrast, ARIMA models may lose valuable price information when data must be differenced and trends removed. Results of this chapter show that price information may not be fully captured by the traditional time series models. Results also indicate that feedforward neural network models have the ability to both identify and forecast time series examined here with considerable quantitative and qualitative accuracy. The neural network generally outperformed the ARIMA model in both forecast error and turning point prediction. This supports the theoretical proofs by Hornik and also Hecht-Nielsen that a feedforward neural network with only one hidden layer can approximate any continuous function.

Table 5.1 Ljung-Box Test Results for Autocorrelation of Residuals From The Estimated ARIMA Models.

		1974-1987		1974-1988		1974-1989	
To Lag	Degrees of Freedom	Chi Square	Prob.	Chi Square	Prob.	Chi Square	Prob.
6	3	6.08	0.11	0.35	0.84	7.73	0.05
12	9	14.21	0.12	9.25	0.32	18.73	0.03
18	15	18.24	0.25	13.86	0.46	22.57	0.09
24	21	21.44	0.43	16.27	0.70	24.99	0.25
30	24	24.14	0.62	19.91	0.80	28.79	0.37

A probability value greater that 0.05 indicates that the estimated model is a reasonable representation of date generating process.

Figure 5.1 Topology of the Neural Network Price Forecasting Model

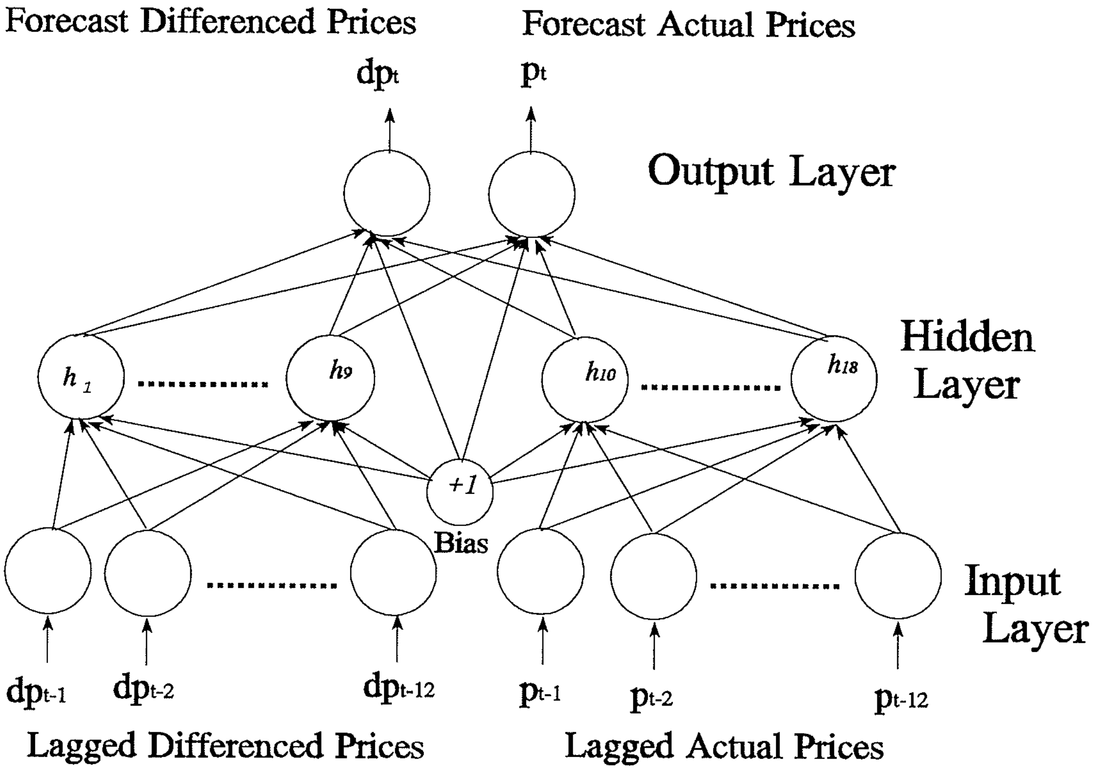


Table 5.2 Results of US Monthly Cattle Price Forecasts by the ARIMA and Neural Network, 1988-1990.

	1988		1989		1990	
	Forecast error[a]		Forecast error		Forecast error	
Month	ARIMA	NN	ARIMA	NN	ARIMA	NN
1	0.68	-0.16	0.17	-0.75	0.48	-0.61
2	3.06	1.15	-0.14	-1.70	0.47	-1.54
3	5.60	2.27	3.05	0.53	2.63	0.38
4	6.62	2.22	2.99	0.40	3.65	2.39
5	9.11	4.92	2.78	1.11	1.80	1.80
6	4.48	1.28	0.21	-0.04	-0.03	0.81
7	-0.31	-2.61	-0.84	-0.69	-0.85	-0.18
8	0.67	-1.29	-0.73	-1.33	1.01	1.23
9	1.29	-0.82	-3.64	-5.60	0.34	-0.56
10	2.72	0.28	-2.75	-6.03	1.65	-0.82
11	3.58	0.63	-0.26	-4.69	3.79	-0.43
12	4.54	0.70	2.18	-3.12	4.55	-1.04
RMSE[b]	4.39	1.99	2.10	2.98	2.30	1.17
MAE[c]	3.56	1.53	1.65	2.17	1.77	0.98
Var. of errors	6.62	1.62	1.71	4.21	2.16	0.39

[a] Mean prices for 1988, 1989, and 1990 were $69.54, $72.52, and $77.40 (per cwt) respectively.

[b] Root Mean Squared Errors.

[c] Mean of Absolute Errors.

Table 5.3 Results of Merton's Test of Turning Point Forecasting Power for the ARIMA and Neural Network, 1988-1990.

Year	Model	α_0	α_0	R^2
1988	ARIMA	1.00^{**} (3.54)	-0.20 (-0.07)	0.04
	Neural Network	0.0	0.80^{*} (2.58)	0.40
1989	ARIMA	0.40 (1.97)	-0.46 (-1.72)	0.23
	Neural Network	0.20 (1.58)	0.80^{**} (4.83)	0.70
1990	ARIMA	0.40 (1.74)	0.31 (1.05)	0.10
	Neural Network	0.40 (1.74)	0.31 (1.05)	0.10

t values in parentheses.
*Significant at 5 percent level. **Significant at 1 percent level; $X_t = \alpha_0 + \alpha_1 A_t + \varepsilon_t$ is turning point test equation.

CHAPTER 6

SUMMARY

The objective of this study was to test whether neural network models can produce more accurate out of sample forecasts than the bench mark Box-Jenkins ARIMA time series models. Data used were US monthly cattle prices. Neural networks are computational structures which were developed in cognitive science to model the function of the brain. Hecht-Nielsen, Cybenko, and Hornik et al. showed that single hidden layer feedforward neural networks are universal approximators which can approximate all data generating processes. They have been widely used in engineering and computer science for problems such as pattern recognition and classification tasks and started to become more popular around 1985 when the method of back-propagation (BP) was introduced by the PDP group (Rumelhart et al.). They showed that the BP method can classify nonlinear tasks as well as linear ones. Since then, there has been an increase in research and applications in the area of neural networks. However, research in economics, finance, and forecasting using neural networks is relatively new.

Chaos and Nonlinear Tests

Chapter two tested for the presence of chaos and nonlinear dynamics in cattle prices. The properties and behavior of the data were analyzed and tested for chaotic behavior using monthly US cattle prices from 1973-1990. This time period was used since it is believed that structural change occurred in the market after the early 1970's. Such tests are important, since the behavior of the data must be understood before appropriate forecasting models can be constructed. A chaotic series can be defined as a series which appears random but in reality is not random. Chaotic series are a subset of nonlinear processes that cannot be modeled by linear time series or linear regression models. If prices are generated by nonlinear processes, then linear models such as ARIMA are no longer suitable. Therefore, it is necessary to determine whether ARIMA models were an appropriate choice for modelling US cattle prices.

Since the presence of nonlinear and chaos in time series usually are not picked up by the conventional tests and methods, Hurst (Mandelbrot), Grassberger and Procaccia, and Brock et al. have proposed tests which were used here. These tests can pick up chaotic and subtle nonlinear behavior which conventional tests fail to identify. In such cases, nonlinear models such as neural networks would be expected to perform better than linear models such as ARIMA. However, the above chaos tests indicated that the dynamics of price behavior was linear and was not chaotic. Therefore, linear ARIMA models appear to be acceptable models for monthly price analysis and forecasting. However, even with linear data, neural networks may have advantages over time series models, which are mentioned below.

Neural Network Theory

Chapter three explained the structure of neural networks and covered the history of neural networks, the topology, transfer functions, learning algorithms and optimization, out of sample prediction, and the statistical aspects of neural networks in relation to regression analysis and economic forecasting problems. Although chaos tests indicated that the data are linear, neural networks may still have advantages over ARIMA models.

This is because they do not strictly impose the differencing requirement on non-stationary data and they are also less sensitive to violations of traditional regression assumptions than ARIMA models.

However, it was argued that the performance of neural networks for price analysis depends on the structure of the particular models and the learning rules employed. While the statistical aspects of time series and regression models have been investigated by statisticians, neuroscientists have been less concerned with the statistical aspects of neural networks. Therefore, this study also discussed the role of statistics in designing computationally efficient neural network models and also reducing their learning time.

Neural Network Results

Chapter four examined the out of sample forecasting power of ARIMA models with that of neural network models. The data are first examined for stationarity and after proper differencing, are used to estimate the time series model. This same differenced data was used for training a fully connected single hidden layer neural network. At the forecasting stage, each forecasted value was used to perform the next step forecast without re-estimating the model or re-training the network. Forecasts were examined for forecast error by using mean squared errors and mean absolute errors criteria. They were also examined for correct turning point prediction and price direction by using the Cumby-Modest version of Merton's test so that buying and selling performance is taken into account. Results showed that this model was able to successfully forecast prices out of sample and was about four times more accurate than the ARIMA model. This study also showed that neural networks were also able to capture a significant number of turning points in contrast to the traditional ARIMA models.

Chapter five used both actual price levels and price changes for the neural network models, in contrast to chapter four which used only actual data. However, since time series models required differencing and differencing may cause loss of information inherent in the data, this may be one disadvantage of ARIMA models compared to neural networks. Since it is believed that actual data will capture the trend in the data and differenced data may capture the deviation around the trend, the neural network model used here with actual data and differenced data does not suffer from possible loss of price information due to differencing. Two extra years of out of sample data were added for this chapter in order to provide an even more comprehensive test of out of sample performance. Results showed that the above neural networks on average could forecast price movements with lower mean square errors and lower error variance than ARIMA methods. The models were also able to capture a significant numbers of turning points in contrast to the ARIMA time series models. The turning point test results indicate that neural networks have promising potential for giving buy and sell signals in commodity trading where the knowledge of the turning points is crucial. This is consistent with Hornik et al. and others who have shown theoretically that feedforward neural networks are universal approximators which can approximate all data generating processes.

Limitations and Suggestions for Future Research

This study showed that neural networks are superior to linear ARIMA time series models for commodity price forecasting using monthly cattle prices for the time period analysed. However, neural networks should be applied to other commodity prices in order to comprehensively establish their effectiveness and accuracy. While this study used monthly data to address forecasting in commodity markets in order to avoid possible noise in daily or weekly data, neural networks should examine this shorter term data also to see how well it can be forecasted, even though theory suggests that neural networks are reasonably noise tolerant. Finally, neural networks are complex and still at the beginning stage of theoretical developments, so much of what is known about the design of the networks architecture is somewhat limited. Consequently, results produced by neural networks are often quite subjective due to the trial and error in design, and therefore dependent on the skill of the user. Therefore, further research is also needed in the theory of neural networks.

REFERENCES

Bachelier, L. "Theory of Speculation" in P. Cootner, ed., The Random Character of Stock market Prices, Cambridge, MIT Press, 1964.

Bailey, David L. and Donna M. Thompson. "Developing Neural Network Applications" AI Expert, (September)(1990):34-41.

Barnett, W. and P. Chen. "The Aggregation-Theoretic Monetary Aggregates are Chaotic and Have Strange Attractors: An Econometric Application of Mathematical Chaos" In Dynamic Econometric Modelling (W. Barnett, E. Berndt, and H. White, eds.), Cambridge University Press, 1988.

Baumol, W.J. and J. Benhabib. "Chaos: Significance, Mechanism, and Economic Applications" Journal of Economic Perspectives, 3(1989):77-105.

Bessler, D.A. and J.A. Brandt. "Forecasting Livestock Prices with Individual and Composite Methods" Applied Economics, 13(1981):513-22.

Blank, Steven C. "Chaos in Futures Markets? A Nonlinear Dynamic Analysis" The Journal of Futures Markets, 11(1991):711-728.

Booth, G.G., F.R. Kaen, and P.E. Koveos. "Persistent Dependence in Gold Prices" Journal of Financial Research, (1982):85-93.

Booth, G.G., F.R. Kaen, and P.E. Koveos. "R/S Analysis of Foreign exchange Rates Under Two International Monetary Regimes" Journal of Monetary Economics, 10(1982):407-415.

Bowerman, Bruce L. and Richard T. O'Connell. Time Series Forecasting, 2nd ed., Duxbury Press, Boston, 1987.

Box, G.E.P. and G.M. Jenkins. Time Series Analysis: Forecasting and Control, Revised Edition, Holden-Day, San Francisco, 1976.

Brandt, J.A. and D.A. Bessler. "Forecasting With Vector Autoregressions Versus A Univariate ARIMA Process: an Empirical Example with US Hog Prices" North Central Journal of Agricultural Economics, 6(1984):29-36.

Brock, William A., David A. Hsieh, and Blake LeBaron. Nonlinear Dynamics, chaos, and Instability: Statistical Theory and Economic Evidence, The MIT Press, 1992.

Brock, William A., W. Dechert, and J. Scheinkman. "A Test for Independence Based on the Correlation Dimension" Working Paper, University of Wisconsin at Madison, University of Houston and University of Chicago, 1987.

Chakraborty, Kanad, K. Mehrotra, C.K. Mohan, and S. Ranka. "Forecasting the Behavior of Multivariate Time Series Using Neural Networks" Neural Networks, 5(1992):961-970.

Chavas, Jean-Paul and Matthew T. Holt. "Market Instability and Nonlinear Dynamics" American Journal of Agricultural Economics, 75(1993):113-120.

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

Cybenko, G. "Approximation by Superpositions of a Sigmoidal Function" Mathematics of Control, Signals and Systems, 2(1989):303-14.

Dolado, Juan J., Tim Jenkinson, and Simon Sosvilla-Rivera. "Cointegration and Unit Roots", Journal of Economic Surveys, 4(1990):247-273.

Dorfman, Jeffrey H. and Christopher S. McIntosh. "Results of A Price Forecasting Competition" American Journal of Agricultural Economics, 72(1990):804-808.

Engle, R.F. and C.W.J. Granger. "Co-Integration and Error Correction: Representation, Estimation and Testing" Econometrica, 55(1987):251-276.

Fama, Eugene F. "The Bahavior of Stock Market Prices" Journal of Business, 39(1965):226-241.

Fama, Eugene F. "Efficient Capital Markets: a Review of Theory and Empirical Work", Journal of Finance, 25(1970):383-417.

Fama, Eugene F. "Efficient Capital Markets: II" Journal of Finance, 46(1991):1575-1617.

Farmer, Doyne J. and John J. Sidorowich. "Predicting Chaotic Time Series" Physical Review Letters, 59(1987):845-848.

Feller, W. "The Asymptotic Distribution of the Range of Sums of Independent Random Variables" Annals of Mathematics and Statistics, 22(1951):427-435.

Gill, J.M. and L.M. Albisu. "Composite Forecasting Methods: An Application to Spanish Maize Prices" Journal of Agricultural Economics, 44(1993):264-271.

Goodwin, Barry K. "Forecasting Cattle Prices in the Presence of Structural Change" Southern Journal of Agricultural Economics, December(1992):11-22.

Grassberger, P. and I. Procaccia. "Measuring the Strangeness of Strange Attractors" Physica, 9D(1983): 30-31.

Greene, M.T. and B.D. Fielitz. "Long Term Dependence in Common Stock Returns" Journal of Financial Economics, 4(1977):339-349.

Harris K.S. and R.M. Leuthold. "A Comparison of Alternative Forecasting Techniques for Livestock Prices: A Case Study" North Central Journal of Agricultural Economics, 7(1985):40-50.

Harris, K.S. and R.M. Leuthold. "A Comparison of Alternative Forecasting Techniques for Livestock Prices: A Case Study" North Central Journal of Agricultural Economics, 7(1985):40-50.

Harvey, A.C. The Econometric Analysis of Time Series, The MIT Press, 1990.

Hecht-Nielsen, Robert. "Neurocomputing: Picking the Human Brain" IEEE Spectrum, March 1988.

Hecht-Nielsen, Robert. Neurocomputing, Addison-Wesley, Menlo Park, CA, 1989.

Helms, B.P., F.R. Kaen, and R.E. Rosenman. "Memory in Commodity Futures Contracts", Journal of Futures Markets, 4(1984):559-567.

Hertz, John., Anders Krogh, and Richard G. Palmer. Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Company, 1990.

Hornik, Kurt., Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks are Universal Approximators" Neural Networks, 2(1989):359-66.

Hsieh, David A. "Chaos and Nonlinear Dynamics: Application to Financial Markets" Journal of Finance, 46(1991):1839-1877.

Hurst, H.E. "Long-Term Storage Capacity of Reservoirs" Transactions of American Society of Civil Engineering, 116(1951):770-799.

Judge, G. George, R.C. Hill, W.E. Griffiths, H. Lutkepohl, and T. Lee. Introduction to the Theory and Practice of Econometrics, 2nd ed., New York, John Wiley & Sons, 1988.

Kamijo, Ken-ichi and Tetsuji Tanigawa. "Stock Price Pattern Recognition: A Recurrent Neural Network Approach" IEEE International Joint Conference on Neural Networks, 1(1990):215-221.

Kermanshahi, Bahman. "Artificial Neural Network for Forecasting Daily Loads of a Canadian Electric Company" Proceedings of the Second International Forum on Application of Neural Networks to Power Systems, Yokohama, Japan, April 1993.

Kimoto, Takashi and Kazuo Asakawa. "Stock Market Prediction Systems With Modular Neural Networks" IEEE International Joint Conference on Neural Networks, 1(1990):1-6.

Krose, B.J.A. and P.P. van der Smagt. An Introduction to Neural Networks, 5th ed, University of Amsterdam, 1993.

cLapedes, A. and R. Farber. "Nonlinear Signal Processing Using Neural Networks" Proceedings of IEEE Conference on Neural Information Processing System - Neural and Synthetic, 1987.

Larrain, Maurice. "Testing Chaos and Nonlinearities in T-Bill Rates" Financial Analysts Journal, 47(1991):51-62.

Lee, T., Halbert White, and C.W.J. Granger. "Testing for Neglected Nonlinearity in Time Series Models" Journal of Econometrics, 56(1993):269-290.

Lorenz, H. W. "Nonlinear Dynamical Economics and Chaotic Motions" in Lecture Notes in Economics and Mathematical Systems, New York, Springer-Verlag, 1989.

Mandelbrot, B. The Fractal Geometry of Nature, New York, W. H. Freeman, 1982.

Masters, Thimothy. Practical Neural Network Recipes in C++, Academic Press, New York, 1993.

McClelland, J. L. and D. E. Rumelhart. Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercise, Cambridge, MIT Press, 1988.

McIntosh, S.C. and J.H. Dorfman. "Qualitative Forecast Evaluation: A Comparison of Two Performance Measures" American Journal of Agricultural Economics, 74(1992):209-214.

Merton, R.C. "On Market Timing and Investment Performance: An Equilibrium Theory of Value for Market Forecasts" Journal of Business, 54(1981):363-406.

Naik, Gopal and R. M. Leuthold. "A Note on Qualitative Forecast Evaluation" American Journal of Agricultural Economics, 68(1986):721-26.

Odom, Marcus D. and Ramesh Sharda. "A Neural Network Model For Bankruptcy Prediction" IEEE International Joint Conference on Neural Networks, 2(1991):163-68.

Peters, Edgar, E. Chaos and Order in the Capital Markets, John Wiley & Sons, 1992.

Peters, Edgar, E. "A Chaotic Attractor for the S&P 500" Financial Analysts Journal, 47(1991):55-81.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. Numerical Recipes in C, Cambridge University Press, 1988.

Rumelhart, D., J. McClelland, and the PDP Group. "Parallel Distributed Processing" Explorations in the Microstructure of Cognition, Vol. 1: Foundation, Cambridge, MIT Press, 1986.

Savit, Robert. "When Random is Not Random: An Introduction to Chaos in Market

Prices" The Journal of Futures Markets, 8(1988):271-289.

Scheinkman, J. and B. LeBaron. "Nonlinear Dynamics and Stock Returns" Journal of Business, 62(1989):311-337.

Schonebug, E. "Stock Price Prediction Using Neural Networks: A Project Report" Neurocomputing, 2(1990):17-27.

Stein, Roger. "Preprocessing Data for Neural Networks" AI Expert, (March)(1993):32-7.

Stephens, Kenneth R., William R. Hutchison, Sharon S. Hormby, and Thomas M. Bell. "Dynamic Resource Allocation Using Adaptive Networks" Neurocomputing, 2(1990)9:16.

Surkan A.J. and J.C. Singleton. "Neural Networks For Bond Rating Improved by Multiple Hidden Layers" IEEE International Joint Conference on Neural Networks, 2(1991):157-162.

Tam, Kar Yan and Melody Y. Kiang. "Managerial Application of Neural Networks: The Case of Bank Failure Predictions" Management Science, 38(1992):926-47.

Trippi, Robert R. and Duane DeSieno. "Trading Equity Index Futures With a Neural Network" Journal of Portfolio Management, 19(1)(1992):27-33.

White, H. Artificial Neural Networks Approximation and Learning Theory, Blackwell, Cambridge, 1992.

White, H. "On Learning the Derivatives of an Unknown Mapping With Multilayer Feedforward Networks" Neural Networks, 5(1992):129-138.

White, H. "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns" IEEE International Joint Conference on Neural Networks, 2(1988):451-58.

Willey, Thomas. "Testing For Nonlinear Dependence in Daily Stock Indices" Journal of Economics and Business, 42(1992):63-74.

Working, H. "A Random Difference Series for Use in the Analysis of Time Series" Journal of the American Statistical Association, 29(1934):11-24.

Yang, Seung-Ryong and B. Wade Brorsen. "Nonlinear Dynamics of Daily Cash Prices" American Journal of Agricultural Economics, 74(1992):706-715.

Yang, Seung-Ryong and B. Wade Brorsen. "Nonlinear Dynamics of Daily Future Prices: Conditional Heteroscedasticity or Chaos?" The Journal of Futures Markets, 13(1993):175-191.

Zurada, Jacek M. Introduction to Artificial Neural Systems, West Publishing Company, New York, 1992.