

FORECASTING FUTURES TRADING VOLUME USING NEURAL NETWORKS

By Iebeling Kaastra

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

Master of Science

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

© October, 1994



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-92208-7

Canada

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.

Economics, Agricultural

SUBJECT TERM

0503

SUBJECT CODE

U·M·I

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



Nom _____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.



SUJET

CODE DE SUJET

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS

Architecture	0729
Beaux-arts	0357
Bibliothéconomie	0399
Cinéma	0900
Communication verbale	0459
Communications	0708
Danse	0378
Histoire de l'art	0377
Journalisme	0391
Musique	0413
Sciences de l'information	0723
Théâtre	0465

ÉDUCATION

Généralités	515
Administration	0514
Art	0273
Collèges communautaires	0275
Commerce	0688
Économie domestique	0278
Éducation permanente	0516
Éducation préscolaire	0518
Éducation sanitaire	0680
Enseignement agricole	0517
Enseignement bilingue et multiculturel	0282
Enseignement industriel	0521
Enseignement primaire	0524
Enseignement professionnel	0747
Enseignement religieux	0527
Enseignement secondaire	0533
Enseignement spécial	0529
Enseignement supérieur	0745
Évaluation	0288
Finances	0277
Formation des enseignants	0530
Histoire de l'éducation	0520
Langues et littérature	0279

Lecture	0535
Mathématiques	0280
Musique	0522
Orientation et consultation	0519
Philosophie de l'éducation	0998
Physique	0523
Programmes d'études et enseignement	0727
Psychologie	0525
Sciences	0714
Sciences sociales	0534
Sociologie de l'éducation	0340
Technologie	0710

LANGUE, LITTÉRATURE ET LINGUISTIQUE

Langues	
Généralités	0679
Anciennes	0289
Linguistique	0290
Modernes	0291
Littérature	
Généralités	0401
Anciennes	0294
Comparée	0295
Médiévale	0297
Moderne	0298
Africaine	0316
Américaine	0591
Anglaise	0593
Asiatique	0305
Canadienne (Anglaise)	0352
Canadienne (Française)	0355
Germanique	0311
Latino-américaine	0312
Moyen-orientale	0315
Romane	0313
Slave et est-européenne	0314

PHILOSOPHIE, RELIGION ET THÉOLOGIE

Philosophie	0422
Religion	
Généralités	0318
Clergé	0319
Études bibliques	0321
Histoire des religions	0320
Philosophie de la religion	0322
Théologie	0469

SCIENCES SOCIALES

Anthropologie	
Archéologie	0324
Culturelle	0326
Physique	0327
Droit	0398
Économie	
Généralités	0501
Commerce-Affaires	0505
Économie agricole	0503
Économie du travail	0510
Finances	0508
Histoire	0509
Théorie	0511
Études américaines	0323
Études canadiennes	0385
Études féministes	0453
Folklore	0358
Géographie	0366
Gérontologie	0351
Gestion des affaires	
Généralités	0310
Administration	0454
Banques	0770
Comptabilité	0272
Marketing	0338
Histoire	
Histoire générale	0578

Ancienne	0579
Médiévale	0581
Moderne	0582
Histoire des noirs	0328
Africaine	0331
Canadienne	0334
États-Unis	0337
Européenne	0335
Moyen-orientale	0333
Latino-américaine	0336
Asie, Australie et Océanie	0332
Histoire des sciences	0585
Loisirs	0814
Planification urbaine et régionale	0999
Science politique	
Généralités	0615
Administration publique	0617
Droit et relations internationales	0616
Sociologie	
Généralités	0626
Aide et bien-être social	0630
Criminologie et établissements pénitentiaires	0627
Démographie	0938
Études de l'individu et de la famille	0628
Études des relations interethniques et des relations raciales	0631
Structure et développement social	0700
Théorie et méthodes	0344
Travail et relations industrielles	0629
Transports	0709
Travail social	0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES

Agriculture	
Généralités	0473
Agronomie	0285
Alimentation et technologie alimentaire	0359
Culture	0479
Élevage et alimentation	0475
Exploitation des pâturages	0777
Pathologie animale	0476
Pathologie végétale	0480
Physiologie végétale	0817
Sylviculture et taune	0478
Technologie du bois	0746
Biologie	
Généralités	0306
Anatomie	0287
Biologie (Statistiques)	0308
Biologie moléculaire	0307
Botanique	0309
Cellule	0379
Écologie	0329
Entomologie	0353
Génétique	0369
Limnologie	0793
Microbiologie	0410
Neurologie	0317
Océanographie	0416
Physiologie	0433
Radiation	0821
Science vétérinaire	0778
Zoologie	0472
Biophysique	
Généralités	0786
Médicale	0760

SCIENCES DE LA TERRE

Biogéochimie	0425
Géochimie	0996
Géodésie	0370
Géographie physique	0368

Géologie	0372
Géophysique	0373
Hydrologie	0388
Minéralogie	0411
Océanographie physique	0415
Paléobotanique	0345
Paléocéologie	0426
Paléontologie	0418
Paléozoologie	0985
Palynologie	0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT

Économie domestique	0386
Sciences de l'environnement	0768
Sciences de la santé	
Généralités	0566
Administration des hôpitaux	0769
Alimentation et nutrition	0570
Audiologie	0300
Chimiothérapie	0992
Dentisterie	0567
Développement humain	0758
Enseignement	0350
Immunologie	0982
Loisirs	0575
Médecine du travail et thérapie	0354
Médecine et chirurgie	0564
Obstétrique et gynécologie	0380
Ophtalmologie	0381
Orthophonie	0460
Pathologie	0571
Pharmacie	0572
Pharmacologie	0419
Physiothérapie	0382
Radiologie	0574
Santé mentale	0347
Santé publique	0573
Soins infirmiers	0569
Toxicologie	0383

SCIENCES PHYSIQUES

Sciences Pures

Chimie	
Généralités	0485
Biochimie	0487
Chimie agricole	0749
Chimie analytique	0486
Chimie minérale	0488
Chimie nucléaire	0738
Chimie organique	0490
Chimie pharmaceutique	0491
Physique	0494
Polymères	0495
Radiation	0754
Mathématiques	0405
Physique	
Généralités	0605
Acoustique	0986
Astronomie et astrophysique	0606
Électrique et électricité	0607
Fluides et plasma	0759
Météorologie	0608
Optique	0752
Particules (Physique nucléaire)	0798
Physique atomique	0748
Physique de l'état solide	0611
Physique moléculaire	0609
Physique nucléaire	0610
Radiation	0756
Statistiques	0463

Sciences Appliquées Et Technologie

Informatique	0984
Ingénierie	
Généralités	0537
Agricole	0539
Automobile	0540

Biomédicale	0541
Chaleur et thermodynamique	0348
Conditionnement (Emballage)	0549
Génie aérospatial	0538
Génie chimique	0542
Génie civil	0543
Génie électronique et électrique	0544
Génie industriel	0546
Génie mécanique	0548
Génie nucléaire	0552
Ingénierie des systèmes	0790
Mécanique navale	0547
Métallurgie	0743
Science des matériaux	0794
Technique du pétrole	0765
Technique minière	0551
Techniques sanitaires et municipales	0554
Technologie hydraulique	0545
Mécanique appliquée	0346
Géotechnologie	0428
Matériaux plastiques (Technologie)	0795
Recherche opérationnelle	0796
Textiles et tissus (Technologie)	0794

PSYCHOLOGIE

Généralités	0621
Personnalité	0625
Psychobiologie	0349
Psychologie clinique	0622
Psychologie du comportement	0384
Psychologie du développement	0620
Psychologie expérimentale	0623
Psychologie industrielle	0624
Psychologie physiologique	0989
Psychologie sociale	0451
Psychométrie	0632



FORECASTING FUTURES TRADING VOLUME USING NEURAL NETWORKS

BY

IEBELING KAASTRA

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

MASTER OF SCIENCE

© 1994

Permission has been granted to the LIBRARY OF THE UNIVERSITY OF MANITOBA to lend or sell copies of this thesis, to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film, and UNIVERSITY MICROFILMS to publish an abstract of this thesis.

The author reserves other publications rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's permission.

ABSTRACT

Artificial neural networks are universal and highly flexible function approximators first used in the fields of cognitive science and engineering. In recent years, neural network applications in finance for such tasks as pattern recognition, classification, and time series forecasting have dramatically increased. However, the large number of parameters that must be selected to develop a neural network forecasting model have meant that the design process still involves much trial and error.

Therefore, the objective of this study is to provide a practical introductory guide in the design of a neural network for forecasting economic time series data. An eight-step procedure to design a neural network forecasting model is explained. The design procedure is then illustrated by developing backpropagation neural network models to forecast monthly futures trading volume for barley, canola, flax, oats, rye, and wheat traded on the Winnipeg Commodity Exchange (WCE). Comparisons of forecasting accuracy are made by commodity and by forecast horizon.

The results indicate that the neural networks are able to forecast up to nine months ahead and outperform the naive model for all commodities except barley and rye. The neural network forecasts relative to the naive model currently in place to forecast futures trading volume do not deteriorate as the forecast horizon increases. The results suggest that improvements to the neural networks could be made by matching the moving average length of the inputs to the number of months forecasted ahead. Twenty three out of the 54 networks used three hidden neurons indicating that a small network can approximate the function quite well.

ACKNOWLEDGEMENTS

I would like to extend my greatest appreciation to my advisor Dr. Milton Boyd for introducing me to neural networks and for his continuing invaluable support and enthusiasm. I would also like to sincerely thank my other thesis committee members Drs. Merle Faminow and Russell Radford for their assistance. Giovani de Matos and Nowrouz Kohzadi also deserve recognition for their helpful discussions concerning neural networks. As well, I would like to express my gratitude to the Winnipeg Commodity Exchange for their information and funding support.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	v
LIST OF TABLES	vi
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. DESIGNING A NEURAL NETWORK FORECASTING MODEL	
Introduction	3
Backpropagation Neural Networks	5
Steps in Designing a Neural Network Forecasting Model	6
Variable Selection	7
Data Collection	8
Data Preprocessing	9
Training, Testing, and Validation Sets	12
Neural Network Paradigms	14
Evaluation Criteria	20
Neural Network Training	21
Implementation	27
Summary	28
CHAPTER 3. FORECASTING FUTURES TRADING VOLUME USING NEURAL NETWORKS	
Introduction	37
Backpropagation Neural Networks	38
Forecasting Evaluation Methods	40
Data	41
Variable Selection	42
Data Preprocessing	43
Neural Network Topology	44
Training and Forecasting Procedure	45
Evaluation and Comparison	46
Summary	48
CHAPTER 4. SUMMARY	56
Designing a Neural Network Forecasting Model	56
Forecasting Futures Trading Volume Using Neural Networks	57
Limitations and Suggestions for Future Research	58
REFERENCES	59

LIST OF FIGURES

Figure 2.1	A Feedforward Neural Network	31
Figure 2.2	Logarithmic Transformation of Monthly Wheat Futures Trading Volume at the Winnipeg Commodity Exchange, Measured in 20 Tonne Job Lots, 1977/78 - 1992/93	32
Figure 2.3	Walk-forward/Sliding Windows Testing Routine	33
Figure 2.4	Two Common Scaling Methods Applied to Daily Cash S&P 500 Closing Prices, 1962 - 1993	34
Figure 2.5	Possible Neural Network Training and Testing Set Errors	35
Figure 2.6	Simplified Graphical Representation of a Neural Network Error Surface	36
Figure 3.1	A Feedforward Neural Network	51
Figure 3.2	Futures Trading Volume at the Winnipeg Commodity Exchange, 1977/78 - 1992/93	53

LIST OF TABLES

Table 2.1	Common Parameters in Designing a Backpropagation Neural Network . .	30
Table 3.1	Results of Out-of-Sample Futures Trading Volume Forecasts by Neural Network Models, Winnipeg Commodity Exchange, 1991/92 - 1992/93 . .	52
Table 3.2	Number of Hidden Neurons Selected in Final Neural Network Forecasting Models Based on Testing Set Performance	55

CHAPTER 1

INTRODUCTION

Neural networks were originally developed in cognitive science and were later used in engineering for pattern recognition and classification. In recent years, neural networks have become increasingly popular in finance. As universal function approximators, they are powerful methods for pattern recognition, classification, and forecasting. Neural networks are nonparametric, less sensitive to error term assumptions, and they can tolerate noise, chaotic components, and heavy tails better than most other methods (Masters, 1993). Other advantages often cited are their greater adaptability, robustness, and fault tolerance.

However, neural networks have been criticized because of the "black box" nature of their solutions and the large number of parameters that must be experimentally selected to generate a good forecast. Common parameters that must be determined include the type of data scaling, learning rates, training tolerance, number of training iterations, number of hidden neurons and layers, among many others.

The objective of this study is to provide a practical introductory guide in the design of a neural network for forecasting economic time series data. An eight-step procedure to design a neural network forecasting model is explained in chapter two, including a discussion of tradeoffs in parameter selection, some common pitfalls, and points of disagreement among practitioners. The design steps are: (1) variable selection, (2) data collection, (3) data preprocessing, (4) training, testing, and validation sets, (5) neural network paradigms, (6) evaluation criteria, (7) neural network training, and (8) implementation.

The design procedure is then illustrated in chapter three by developing backpropagation neural network models to forecast monthly futures trading volume on the Winnipeg Commodity Exchange (WCE). Transaction fees on futures trading volume are an important source of revenue for the WCE and forecasts are needed to assist in budgeting. Commodities examined include barley, canola, flax, oats, rye, and wheat.

The inputs to the neural networks include a three-period moving average of lagged trading volume, open interest, mean cash prices, futures price variability, and producer deliveries to licensed elevators in the prairies. The training set consists of 148 monthly observations from the crop years 1977/78 to 1990/91. Twenty testing observations were randomly selected from the training set. The out-of-sample set consists of 24 observations from the crop years 1991/92 and 1992/93.

Separate neural networks were designed and trained for each of the nine monthly forecasts. In this way, each neural network can specialize in its particular step ahead forecast. Three, five, seven, and nine hidden neurons were trained and evaluated on the testing set using seven sets of randomly selected starting weights.

The forecasting power of the neural networks are evaluated using root mean squared error (RMSE) and mean absolute percentage error (MAPE), and are then compared to the naive model using the Theil U statistic. Comparisons of forecasting accuracy are made by commodity and by forecast horizon.

Chapter four includes a summary of chapters two and three. The eight steps of designing a neural network are listed and data preprocessing, overfitting, and neural network training are explained from chapter two. Futures trading volume forecasts are summarized from chapter three including the out-of-sample results.

CHAPTER 2

DESIGNING A NEURAL NETWORK FORECASTING MODEL

Introduction

Neural networks have become increasingly popular in finance as financial services organizations have been the second largest sponsors of research in neural network applications (Trippi and Turban, 1993). Typical applications in finance include mortgage risk assessment, risk rating of exchange-traded fixed income investments, portfolio selection/diversification, simulation of market behaviour, index construction, identification of economic explanatory variables, and economic forecasting (Trippi and DeSieno, 1992).

Neural networks are universal function approximators that can map any nonlinear function (White, 1989). As such flexible function approximators, they are powerful methods for pattern recognition, classification, and forecasting. Neural networks are nonparametric, less sensitive to error term assumptions, and they can tolerate noise, chaotic components, and heavy tails better than most other methods (Masters, 1993). Other advantages include greater fault tolerance, robustness, and adaptability compared to expert systems due to the large number of interconnected processing elements that can be 'trained' to learn new patterns (Trippi and Turban, 1993; Lippman, 1987).

In practice, it appears that although many organizations have expressed interest in applying neural network technology, few have actually implemented them successfully (Egan, 1993). Those that have been successful have spent considerable resources experimenting and fine tuning neural networks for their particular applications as evidenced by the many commercial neural network software packages originally developed as in-house proprietary programs.

Neural networks have been criticized and their wide-spread successful application likely hindered because of the black box nature of their solutions, excessive training times, difficulty in obtaining and later replicating a stable solution, the danger of overfitting, tedious software, and the large number of parameters that must be experimentally selected to generate a good forecast. Table 2.1 lists the most common parameters that a researcher must choose when designing a neural network forecasting model. It excludes the many different proprietary features offered by neural network software vendors and ignores some more advanced topics. The large number of ways to organize a neural network account for its powerful function approximation capabilities. The cost of such flexibility in modelling time series data is that the researcher must select the right combination of parameters. As a result of the large number of parameters and the relatively recent introduction of neural networks to finance, deciding on the appropriate network paradigm still involves much trial and error.

Therefore, the objective of this chapter is to provide an overview of a step by step methodology to design a neural network for forecasting economic time series data. First, the architecture of a backpropagation (BP) neural network is briefly discussed. The BP network is used to illustrate the design steps since it is capable of solving a wide variety of problems and is the most common type of neural network in time series forecasting. This is followed by an explanation of an eight-step procedure for designing a neural network including a discussion of tradeoffs in parameter selection, some common pitfalls, and points of disagreement among practitioners. While there are few hard rules, the literature does contain numerous rules of thumb and practical advice that can assist beginners in designing a successful neural network forecasting model.

Backpropagation Neural Networks

Backpropagation (BP) neural networks consist of a collection of inputs and processing units known as either neurons, neurodes, or nodes (Figure 2.1). The neurons in each layer are fully interconnected by connection strengths called weights which, along with the network architecture, store the knowledge of a trained network. In addition to the processing neurons, there is a bias neuron connected to each processing unit in the hidden and output layers. The bias neuron has a value of positive one and serves a similar purpose as the intercept in regression models. The neurons and bias terms are arranged into layers; an input layer, one or more hidden layers, and an output layer. The number of hidden layers and neurons within each layer can vary depending on the size and nature of the data set.

Neural networks are similar to linear and non-linear least squares regression and can be viewed as an alternative statistical approach to solving the least squares problem (White, 1992). Both neural networks and conventional regression analysis attempt to minimize the sum of squared errors. The bias term is analogous to the intercept term in a regression equation. The number of input neurons is equal to the number of independent variables while the output neuron(s) represent the dependent variable(s). Linear regression models may be viewed as a feedforward neural network with no hidden layers and one output neuron with a linear transfer function. The weights connecting the input neurons to the single output neuron are analogous to the coefficients in a linear least squares regression. Networks with one hidden layer resemble nonlinear regression models. The weights represent regression curve parameters.

BP networks are a class of feedforward neural networks with supervised learning rules. Feedforward refers to the direction of information flow from the input to the output layer. Inputs are passed through the system once to determine the output. Supervised learning is the process of

comparing each of the network's forecasts with the known correct answer and adjusting the weights based on the resulting forecast error to minimize the error function.

The BP network is the most common multi-layer network estimated to be used in 80% of all applications (Caudill, 1992). Hornik et al. (1989) showed that the standard BP network using an arbitrary transfer function can approximate any measurable function in a very precise and satisfactory manner, if a sufficient number of hidden neurons are used. Hecht-Nielsen (1989) also demonstrated that a three-layer BP network can approximate any continuous mapping.

Steps in Designing a Neural Network Forecasting Model

A method of designing a neural network forecasting model into distinct steps is used here. The eight-step design methodology presented below draws on the steps outlined by Deboeck (1994), Masters (1993), Blum (1991), and Nelson and Illingworth (1991).

- | | |
|---|-----------------------------|
| 1. Variable Selection | 5. Neural Network Paradigms |
| 2. Data Collection | 6. Evaluation Criteria |
| 3. Data Preprocessing | 7. Neural Network Training |
| 4. Training, Testing, and Validation Sets | 8. Implementation |

The procedure is usually not a single-pass one, but may require visiting previous steps especially between training and variable selection.

Variable Selection

Success in designing a neural network depends on a clear understanding of the problem (Nelson and Illingworth, 1991). Knowing which input variables are important in the market being forecasted is critical. This is easier said than done because the very reason for relying on a neural network is for its powerful ability to detect complex nonlinear relationships among a number of different variables. However, economic theory can help in choosing variables which are likely important predictors. At this point in the design process, the concern is about the raw data from which a variety of indicators will be developed. These indicators form the actual inputs to the neural network.

The financial researcher interested in forecasting market prices must decide whether to use both technical and fundamental economic inputs from one or more markets. Technical inputs are defined as lagged values of the dependent variable or indicators calculated from the lagged values. Fundamental inputs are economic variables which are believed to influence the dependent variable. The simplest neural network model uses lagged values of the dependent variable(s) or its first difference as inputs. Such models have outperformed traditional Box-Jenkins models in price forecasting, although not in all studies. (Kohzadi, 1993; Sharda and Patil, 1992; Tang et al., 1990). A more popular approach is to calculate various technical indicators which are based only on past prices (and occasionally volume and/or open interest) of the market being forecasted (Deboeck, 1994). As an additional improvement, intermarket data can be used since the close link between all kinds of markets, both domestically and internationally, suggests that using technical inputs from a number of interrelated markets should improve forecasting performance. For example, intermarket data such as the Deutsche mark/yen and pound cross rates and interest rate differentials

could be used as neural network inputs when forecasting the D-mark. Fundamental information such as the current account balance, money supply, or wholesale price index may also be helpful.

The frequency of the data depends on the objectives of the researcher. A typical off-floor trader in the stock or commodity futures markets would most likely use daily data if designing a neural network as a component of an overall trading system. An investor with a longer term horizon may use weekly or monthly data as inputs to the neural network to formulate the best asset mix rather than using a passive buy and hold strategy. An economist forecasting the GDP, unemployment, or other broad economic indicators would likely use monthly or quarterly data.

Data Collection

The researcher must consider cost and availability when collecting data for the variables chosen in the previous step. Technical data is readily available from many vendors at a reasonable cost whereas fundamental information is more difficult to obtain. Time spent collecting data cannot be used for preprocessing, training, and evaluating network performance. The vendor should have a reputation of providing high quality data; however, all data should still be checked for errors by examining day to day changes, ranges, logical consistency (e.g. high greater than or equal to close, open less than or equal to low) and missing observations.

Missing observations which often exist, can be handled in a number of ways. All missing observations can be dropped or a second option is to assume that the missing observations remain the same by interpolating or averaging from nearby values. Dedicating an input neuron to the missing observations by coding it as a one if missing and zero otherwise is also often done.

When using fundamental data as an input in a neural network four issues must be kept in mind. First, the method of calculating the fundamental indicator should be consistent over the time series. Second, the data should not have been retroactively revised after initial publication as is commonly done in databases since the revised numbers are not available in actual forecasting. Third, the data must be appropriately lagged as an input in the neural network since fundamental information is not available as quickly as market quotations. Fourth, the researcher should be confident that the source will continue to publish the particular fundamental information or other identical sources are available.

Data Preprocessing

Data preprocessing refers to analyzing and transforming the input and output variables to minimize noise, highlight important relationships, detect trends, and tighten and flatten the distribution of the variable to assist the neural network in learning the relevant patterns. Since neural networks are pattern matchers, the representation of the data is critical in designing a successful network. The input and output variables for which the data was collected are rarely fed into the network in raw form. At the very least, the raw data must be scaled between the upper and lower bounds of the transfer functions (usually between zero and one or negative one and one).

Two of the most common data transformations in both traditional and neural network forecasting are first differencing and taking the natural log of a variable. First differencing, or using changes in a variable, can be used to remove a linear trend from the data. Logarithmic transformation is useful for data which can take on both small and large values and is

characterized by an extended right hand tail distribution. The histograms shown in Figure 2.2 illustrate the compressing effect of a logarithmic transformation on monthly futures trading volume for wheat. Logarithmic transformations also convert multiplicative or ratio relationships to additive which is believed to simplify and improve network training (Masters, 1993).

Another popular data transformation is to use ratios of input variables. Ratios highlight important relationships (e.g. hog/corn, financial statement ratios) while at the same time conserving degrees of freedom because fewer input neurons are required to code the independent variables.

Besides first differences, logs, and ratios, technical analysis can provide a neural network with a wealth of indicators including a variety of moving averages, oscillators, directional movement, and volatility filters. It is a good idea to use mix of different indicators to reduce variable redundancy and provide the network with the ability to adapt to changing market conditions through periodic retraining.

Smoothing both input and output data by using either simple or exponential moving averages is often employed. Empirical work on testing the efficient market hypothesis has found that prices exhibit time dependency or positive autocorrelation while price changes around a trend are somewhat random (Tomek and Querin, 1984). Therefore, attempting to predict price changes around the trend by using either unfiltered prices or price changes as inputs may prove to be difficult. Using moving averages to smooth the independent variables and forecasting trends may be a more promising approach.

Sampling or filtering of data refers to removing observations from the training and testing sets to create a more uniform distribution. The type of filtering employed should be consistent with the objectives of the researcher. For example, a histogram of price changes for a commodity would reveal many small changes from which a speculator cannot profit after deducting realistic execution costs. However, this dense region of the distribution will greatly impact the training of the neural network since small price changes account for the majority of the training facts. The network minimizes the sum of squared errors (or other error function) over all the training facts. By removing these small price changes, overall trading performance can be improved since the network specializes on the larger, potentially profitable price changes. It is possible for trading systems to be unprofitable even if the neural network predicted 85% of the turning points, as the turning points may be only small unimportant price changes (Deboeck, 1994). On the other hand, a floor trader holding positions overnight is likely interested in these small price changes. The researcher must be clear on what exactly the neural network is supposed to learn. Another advantage of filtering is a decrease in the number of training facts which allows testing of more input variables, random starting weights, or hidden neurons rather than training large data sets.

In practice, data preprocessing involves much trial and error. One method to select appropriate input variables is to test various combinations. For example, a 'top 20' list of variables consisting of a variety of technical indicators could be pretested ten at a time with each combination differing by two or three variables. Although computationally intensive, this procedure recognizes the likelihood that some variables may be excellent predictors only when in combination with other variables. Chaos theory and statistical tests cannot make such a determination. Also, the top 20 list can be modified over time as the researcher gains experience

on the type of preprocessing that works for his/her application. This approach is especially useful if the training set is small relative to the number of parameters (weights) which is likely the case if all 20 input variables are presented to the neural network at once.

Training, Testing, and Validation Sets

Common practice is to divide the time series into three distinct sets called the training, testing, and validation (out-of-sample) sets. The training set is the largest set and is used by the neural network to learn the patterns present in the data. The testing set, ranging in size from 10% to 30% of the training set, is used to evaluate the generalization ability of a supposedly trained network. The researcher would select the network(s) which perform best on the testing set. A final check on the performance of the trained network is made using the validation set. The size of the validation set chosen must strike a balance between obtaining a sufficient sample size to evaluate a trained network and having enough remaining observations for both training and testing. The validation set should consist of the most recent contiguous observations. Care must be taken not to use the validation set as a testing set by repeatedly performing a series of train-test-validation steps and adjusting the input variables based on the network's performance on the validation set.

The testing set can be either randomly selected from the training set or consist of a set of observations immediately following the training set. The advantage of randomly selecting testing facts is that the danger of using a testing set characterized by one type of market is largely avoided. For example, a small testing set may only consist of prices in a strong uptrend. The testing set will favour networks which specialize on strong uptrends at the expense of networks which generalize by performing well on both uptrends and downtrends. The advantage of using

the observations following the training set as testing facts is that these are the most recent observations (excluding the validation set) which may be more important than older data.

The randomly selected testing facts should not be replaced in the training set because this would bias the ability to evaluate generalization especially if the testing set is large relative to the training set (e.g. 30%). A deterministic method, such as selecting every n th observation as a testing fact, is also not recommended since it can result in cycles in the sampled data due solely to the sampling technique employed (Masters, 1993).

A more rigorous approach in evaluating neural networks is to use a walk-forward testing routine also known as either sliding or moving window testing. Popular in evaluating commodity trading systems, walk-forward testing involves dividing the data into a series of overlapping train-test-validation sets. Each set is moved forward through the time series as shown in Figure 2.3. Walk-forward testing attempts to simulate real-life trading and tests the robustness of the model through its frequent retraining on a large out-of-sample data set. In walk-forward testing, the size of the validation set drives the retraining frequency of the neural network. Frequent retraining is more time consuming, but allows the network to adapt more quickly to changing market conditions. The consistency or variation of the results in the out-of-sample sets is an important performance measure. For example, in the case of commodity trading systems, a neural network with poor forecasting resulting in excessive equity drawdowns in any out-of-sample period would not be implemented in order to avoid risk of ruin.

It is recommended that the training and testing sets be scaled together since the purpose of the testing set is to determine the ability of the network to generalize. However, by no means, should the validation set be scaled with either the training or testing sets since this biases the integrity of the validation set as a final and independent check on the neural network. In actual use, the researcher has no way of knowing the exact range of future values, but only has a reasonable estimate based on the range of the training and/or testing sets.

Neural Network Paradigms

There are an infinite number of ways to construct a neural network. Neurodynamics and architecture are two terms used to describe the way in which a neural network is organized. The combination of neurodynamics and architecture define the neural network's paradigm. Neurodynamics describe the properties of an individual neuron such as its transfer function and how the inputs are combined (Nelson and Illingworth, 1991). A neural network's architecture defines its structure including the number of neurons in each layer and the number and type of interconnections.

The number of input neurons is one of the easiest parameters to select once the independent variables have been preprocessed because each independent variable is represented by its input neuron. This section will address the selection of the number of hidden layers, hidden layer neurons, and output neurons, and the transfer functions.

Number of Hidden Layers

The hidden layer(s) provide the network with its ability to generalize. In theory, a neural network with one hidden layer with a sufficient number of hidden neurons is capable of approximating any continuous function. In practice, neural networks with one and occasionally two hidden layers are widely used and have performed very well. Increasing the number of hidden layers also increases computation time and the danger of overfitting which leads to poor out-of-sample forecasting performance. Overfitting occurs when a forecasting model has too few degrees of freedom. In other words, it has relatively few observations in relation to its parameters and therefore it is able to memorize individual points rather than learn the general patterns. In the case of neural networks, the number of weights, which is inexorably linked to the number of hidden layers and neurons, and the size of the training set (number of observations) determine the likelihood of overfitting (Baum and Haussler, 1989; Masters, 1993). The greater the number of weights relative to the size of the training set, the greater the ability of the network to memorize idiosyncrasies of individual observations. As a result, generalization for the validation set is lost and the model is of little use in actual forecasting.

Therefore, it is recommended that all neural networks should start with preferably one or at most two hidden layers. If a four-layer neural network (i.e. two hidden layers) proves unsatisfactory after having tested multiple hidden neurons using a reasonable number of randomly selected starting weights, then the researcher should modify the input variables a number of times before adding a third hidden layer. Both theory and virtually all empirical work to date suggests that networks with more than four layers will not improve the results.

Number of Hidden Neurons

Despite its importance, there is no ‘magic’ formula for selecting the optimum number of hidden neurons. Therefore, researchers fall back on experimentation. However, some rules of thumb have been advanced. A rough approximation can be obtained by the geometric pyramid rule proposed by Masters. For a three-layer network with n input neurons and m output neurons, the hidden layer would have $\sqrt{n \times m}$ neurons. The actual number of hidden neurons can still range from one-half to two times the geometric pyramid rule value depending on the complexity of the problem. Baily and Thompson (1989) suggest that the number of hidden neurons in a three-layer neural network should be 75% of the number of input neurons. Katz (1992) indicates that an optimal number of hidden neurons will generally be found between one-half to three times the number of input neurons. Ersoy (1990) proposes doubling the number of hidden neurons until the network’s performance on the testing set deteriorates. Klimasauskas (1991) suggests that there should be at least five times as many training facts as weights, which sets an upper limit on the number of input and hidden neurons.

It is important to note that the rules which calculate the number of hidden neurons as a multiple of the number of input neurons implicitly assume that the training set is at least twice as large as the number of weights and preferably four or more times as large. If this is not the case, then these rules of thumb can quickly lead to overfitted models since the number of hidden neurons is directly dependent on the number of input neurons (which in turn determine the number of weights). The solution is to either increase the size of the training set or, if this is not possible, to set an upper limit on the number of input neurons so that the number of weights is at least half of the number of training facts. Selection of input variables becomes even more critical in such

small networks since the luxury of the presenting the network with a large number of inputs and having it ignore the irrelevant ones has largely disappeared.

Selecting the 'best' number of hidden neurons involves experimentation. Three methods often used are the fixed, constructive, and destructive. In the fixed approach, a group of neural networks with different numbers of hidden neurons are trained and each is evaluated on the test set using a reasonable number of randomly selected starting weights. The increment in the number of hidden neurons may be one, two, or more depending on the computational resources available. Plotting the evaluation criterion (e.g. sum of squared errors) as a function of the number of hidden neurons for each neural network generally produces a bowl shaped error graph. The network with the least error found at the bottom of the bowl is selected. This approach is time consuming, but generally works very well.

The constructive and destructive approach involve changing the number of hidden neurons during training rather than creating separate networks each with a different number of hidden neurons, as in the fixed approach. Many commercial neural network software packages do not support the addition or removal of hidden neurons during training. The constructive approach involves adding hidden neurons until network performance starts deteriorating. The destructive approach is similar except that hidden neurons are removed during training.

Regardless of the method used to select the range of hidden neurons to be tested, the rule is to always select the network that performs best on the testing set with the least number of hidden neurons. When testing a range of hidden neurons it is important to keep all other parameters constant. Changing any parameter in effect creates a new neural network with a potentially different error surface which would needlessly complicate the selection of the optimum

number of hidden neurons.

Number of Output Neurons

Deciding on the number of output neurons is somewhat more straightforward since there are compelling reasons to always use only one output neuron. Neural networks with multiple outputs, especially if these outputs are widely spaced, will produce inferior results as compared to a network with a single output (Masters, 1993). A neural network trains by choosing weights such that the average error over all output neurons is minimized. For example, a neural network attempting to forecast one month ahead and six month ahead cattle futures prices will concentrate most of its effort on reducing the forecast with the largest error which is likely the six month forecast. As a result, a relatively large improvement in the one month forecast will not be made if it increases the absolute error of the six month forecasts by an amount greater than the absolute improvement of the one month forecast. The solution is to have the neural networks specialize by using separate networks for each forecast. Specialization also makes the trial and error design procedure somewhat simpler since the neural network is smaller and fewer parameters need to be changed to fine tune the final model.

Transfer Functions

Transfer functions are mathematical formulas that determine the output of a processing neuron. They are also referred to as transformation, squashing, activation, or threshold functions. The majority of current neural network models use the sigmoid (S-shaped) function, but others such as the hyperbolic tangent, step, ramping, arc tan, and linear have also been proposed. The

purpose of the transfer function is to prevent outputs from reaching very large values which can 'paralyze' a neural networks and thereby inhibit training.

Linear transfer functions are not useful for nonlinear mapping and classification. Levich and Thomas (1993) and Kao and Ma (1992) found that financial markets are nonlinear and have memory suggesting that nonlinear transfer functions are more appropriate. Transfer functions such as the sigmoid are commonly used for time series data because they are nonlinear and continuously differentiable which are desirable properties for network learning.

Klimasauskas (1994) states that if the network is to learn average behaviour a sigmoid transfer function should be used while if learning involves deviations from the average, the hyperbolic tangent function works best. The ramping and step functions are recommended for binary variables since the sigmoid transfer function approaches zero and one asymptotically. In a standard BP network, the input layer neurons typically use linear transfer functions while all other neurons use a sigmoid function.

Transfer functions such as the sigmoid require all inputs to be scaled zero and one. Linear and mean/standard deviation scaling are two of the most common methods used in neural networks. In linear scaling all observations are linearly scaled between the minimum and maximum values according to the following formula:

$$SV = TF_{\min} + (TF_{\max} - TF_{\min}) \times \frac{(D - D_{\min})}{(D_{\max} - D_{\min})} \quad (2.1)$$

where SV is the scaled value, TF_{\min} and TF_{\max} are the respective minimum and maximum values of the transfer function, D is the value of the observation, and D_{\min} and D_{\max} are the respective minimum and maximum values of all observations.

Simple linear scaling is susceptible to outliers because it does not change the uniformity of the distribution, but merely scales it into the appropriate range for the transfer function. In the S&P 500 data presented in Figure 2.4, linear scaling results in 98.6% of the training facts being contained within 10% of the neuron's activation range. Proper training is unlikely to take place with such a distribution. In mean and standard deviation scaling all values plus or minus x number of standard deviations from the mean are mapped to one and zero, respectively. All other values are linearly mapped between zero and one. This type of scaling creates a more uniform distribution and is more appropriate for data which has not been sampled in any way. Most neural network software programs will automatically scale all variables into the appropriate range. However, it is always a good idea to look at histograms of the scaled input and output variables.

Evaluation Criteria

The most common error function minimized in neural networks is the sum of squared errors. Other error functions offered by software vendors include least absolute deviations, least fourth powers, asymmetric least squares, and percentage differences. These error functions may not be the final evaluation criteria since other common forecasting evaluation methods such as the mean absolute percentage error (MAPE) are typically not minimized in neural networks.

In the case of commodity trading systems, the neural network forecasts would be converted into buy/sell signals according to a predetermined criterion. For example, all forecasts greater than 0.8 or 0.9 can be considered buy signals and all forecasts less than 0.2 or 0.1 as sell signals (Hamm et al., 1993). The buy/sell signals are then fed into a program to calculate some type of risk adjusted return and the networks with the best risk adjusted return (not the lowest testing set

error) would be selected. Low forecast errors and trading profits are not necessarily synonymous since a single large trade forecasted incorrectly by the neural network could have accounted for most of the trading system's profits.

Filtering the time series to remove many of the smaller price changes can largely prevent the situation where a neural network with high turning point forecasting accuracy remains unprofitable. Also, the value of any single trading system can only be established within the context of the user's portfolio of commodity systems. In this regard, neural networks may be especially useful if they behave more like counter trend systems as opposed to the more common trend following systems used by commodity funds.

Neural Network Training

Training a neural network to learn patterns in the data involves iteratively presenting it with examples of the correct known answers. The objective of training is to find the set of weights between the neurons that determine the global minimum of the error function. Unless the model is overfitted, this set of weights should provide good generalization. The BP network uses a gradient descent training algorithm which adjusts the weights to move down the steepest slope of the error surface. Finding the global minimum is not guaranteed since the error surface can include many local minima in which the algorithm can become 'stuck'. A momentum term and five to ten random sets of starting weights can improve the chances of reaching a global minimum. This section will discuss when to stop training a neural network and the selection of learning rate and momentum values.

Number of Training Iterations

There are two schools of thought regarding the point at which training should be stopped. The first stresses the danger of getting trapped in a local minimum and the difficulty of reaching a global minimum. The researcher should only stop training until there is no improvement in the error function based on a reasonable number of randomly selected starting weights (Masters, 1993). The point at which the network does not improve is called convergence. The second view advocates a series of train-test interruptions (Deboeck, 1994; Mendelsohn, 1993). Training is stopped after a predetermined number of iterations and the network's ability to generalize on the test set is evaluated and training is resumed. Generalization is the idea that a model based on a sample of the data is suitable for forecasting the general population. The network for which the testing set error bottoms out is chosen since it is assumed to generalize best.

The criticism of the train-test procedure is that additional train-test interruptions could cause the error on the testing set to fall further before rising again or it could even fall asymptotically (see Figure 2.5). In other words, the researcher has no way of knowing if additional training could improve the generalization ability of the network especially since starting weights are randomized.

Both schools of thought agree that generalization on the validation set is the ultimate goal and both use testing sets to evaluate a large number of networks. The point at which the two approaches depart centres on the notion of overtraining versus overfitting. The convergence approach states that there is no such thing as overtraining only overfitting. Overfitting is simply a symptom of a network that has too many weights. The solution is to reduce the number of hidden neurons (or hidden layers if there is more than one) and/or increase the size of the training set. The train-test approach attempts to guard against overfitting by stopping training based on the

ability of the network to generalize.

The advantage of the convergence approach is that one can be more confident that the global minimum was reached. Replication is likely more difficult for the train-test approach given that starting weights are usually randomized and the mean correlation can fluctuate wildly as training proceeds. Another advantage is that the researcher has two less parameters to worry about; namely the point at which to stop training and the method to evaluate which of the train-test networks is optimal. An advantage of the train-test approach may be that networks with few degrees of freedom can be implemented with better generalization than convergence training which would result in overfitting. However, empirical work has not specifically addressed this issue. The train-test approach also requires less training time.

The objective of convergence training is to reach a global minimum. This requires training for a sufficient number of iterations using a reasonable number of randomly selected starting weights. Even then there is no guarantee with a BP network that a global minimum is reached since it may become trapped in a local minimum. In practice, computational resources are limited and tradeoffs arise. The researcher must juggle the number of input variable combinations to be trained, the interval of hidden neurons over which each network is to be tested, the number of randomly selected starting weights, and the maximum number of runs. For example, 50 input variable combinations tested over three different hidden neurons with five randomly selected starting points and a maximum number of runs of 4,000 result in 3,000,000 iterations (epochs). The same computational time is required for ten input variable combinations tested over six hidden neurons with ten randomly selected starting weights and 5,000 epochs.

One method to determine a reasonable value for the maximum number of runs is to plot the mean correlation, sum of squared errors, or other appropriate error measure for each iteration or at predetermined intervals up to the point where improvement is negligible (usually up to a maximum of 10,000 iterations). Each iteration can be easily plotted if the neural network software creates a statistics file or, if this is not the case, the mean correlation can be recorded at intervals of 100 or 200 from the computer monitor. After plotting the mean correlation for a number of randomly selected starting weights, the researcher can choose the maximum number of runs based on the point where the mean correlation stops increasing quickly and flattens.

The majority of studies that mention the number of training iterations report convergence from 85 to 5,000 iterations (Deboeck and Cader, 1994; Klaussen and Uhrig, 1994; Kohzadi, 1993). However, the range is very wide as 50,000 and 191,400 iterations (Klimasauskas, 1991; Odom and Sharda, 1991) and training times of 60 hours have also been reported (Hamm et al., 1993). Training is affected by many parameters such as the choice of learning rate and momentum values, proprietary improvements to the BP algorithm, among others, which differ between studies and so it is difficult to determine a general value for the maximum number of runs. Also, the numerical precision of the neural network software can affect training because the slope of the error derivative can become very small causing some neural network programs to move in the wrong direction due to round off errors which can quickly build up in the highly iterative training algorithm. It is recommended that researchers determine the number of iterations required to achieve negligible improvement for their particular problem and test as many randomly selected starting weights as computational constraints allow.

Learning Rate and Momentum

A BP network is trained using a gradient descent algorithm which follows the contours of the error surface by always moving down the steepest slope. The objective of training is to minimize the total squared errors, defined as follows (McClelland and Rumelhart, 1986):

$$E = \frac{1}{2} \sum_h^M E_h = \frac{1}{2} \sum_h^M \sum_i^N (t_{hi} - O_{hi})^2 \quad (2.2)$$

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

$$\delta_{hi} = (t_{hi} - O_{hi}) O_{hi} (I - O_{hi}) \quad (2.3)$$

$$\delta_{hi} = O_{hi} (I - O_{hi}) \sum_k^N \delta_{hk} w_{jk} \quad (2.4)$$

$$\Delta w_{ij}(n+1) = \varepsilon (\delta_{hi} O_{hj}) \quad (2.5)$$

where n is the presentation number, δ_{hi} is the error signal of neuron i for pattern h , and ε is the learning rate. The learning rate is a constant of proportionality which determines the size of the weight changes. The weight change of a neuron is proportional to the impact of the weight from that neuron on the error. The error signal for an output neuron and a hidden neuron are calculated by Eqs. (2.3) and (2.4), respectively.

As an analogy to the BP training algorithm, one can consider the problem of trying to throw a ball from point A to point C in Figure 2.6, although in reality the error surface is multidimensional and cannot be represented in a graphical format. The force on the ball is analogous to the learning rate. Applying too much force will cause the ball to overshoot its target and it may never return to point A or it can oscillate between points A and B. During training, a learning rate that is too high is revealed when the error function is changing wildly without showing a continued improvement. Too little force on the ball and it will be unable to escape from point A which is evident during training when there is very little or no improvement in the error function. A very small learning rate also requires more training time. In either case, the researcher must adjust the learning rate during training or 'brainwash' the network by randomizing all weights and changing the learning rate for the new run through the training set.

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

$$\Delta w_{ij}(n+1) = \epsilon (\delta_{hi} O_{hj}) + \alpha \Delta w_{ij}(n) \quad (2.6)$$

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

The momentum term suppresses side to side oscillations by filtering out high-frequency variations. Each new search direction is a weighted sum of the current and the previous gradients. Such a two-period moving average of gradients filters out rapid fluctuations in the learning rate. Momentum values that are too great will prevent the algorithm from following the twists and turns in weight space. McClelland and Rumelhart (1986) indicate that the momentum term is especially

useful in error spaces containing long ravines that are characterized by steep, high walls and a gently sloping floor. Without a momentum term, a very small learning rate would be required to move down the floor of the ravine which would require excessive training time. By dampening the oscillations between the ravine walls, the momentum term can allow a higher learning rate to be used.

Most neural network software programs provide default values for learning rate and momentum that typically work well. Initial learning rates used in previous work vary widely from 0.1 to 0.9 (Yoon and Swales, 1991; Salchenberger et. al., 1992). Common practice is to start training with a higher learning rate such as 0.7 and decrease as training proceeds. Many neural network programs will automatically decrease the learning rate and increase momentum values as convergence is reached.

Implementation

The implementation step is listed as the last one, but in fact requires careful consideration prior to collecting data. Data availability, evaluation criteria, and training times are all shaped by the environment in which the neural network will be deployed. Most neural network software vendors provide the means by which trained networks can be implemented either in the neural network program itself or as an executable file. If not, a trained network can be easily created in a spreadsheet by knowing its architecture, transfer functions, and weights. Care should be taken that all data transformations, scaling, and other parameters remain the same from testing to actual use.

An advantage of neural network's is their ability to adapt to changing market conditions through periodic retraining. Once deployed, a neural network's performance will degrade over time unless retraining takes place. However, even with periodic retraining, there is no guarantee that network performance can be maintained as the independent variables selected may have become less important.

It is recommended that the frequency of retraining for the deployed network should be the same as used during testing on the final model. However, when testing a large number of networks to obtain the final model, less frequent retraining is acceptable in order to keep training times reasonable. A good model should be robust with respect to retraining frequency and will usually improve as retraining takes place more often.

Summary

Neural networks are a type of artificial intelligence technology that mimic the human brain's powerful ability to recognize patterns. The application of neural networks in the field of financial economics in pattern recognition, classification, and forecasting is relatively new (Trippi and Turban, 1993). In theory, neural networks are capable of approximating any continuous function. Such flexibility provides a potentially powerful forecasting tool, while at the same time, the large number of parameters that must be selected complicates the design process. In practice, designing a neural network forecasting model involves much trial and error.

Therefore, the objective of this chapter was to provide a practical, non-technical introduction to designing a neural network forecasting model using economic time series. The design procedure was divided into eight steps; (1) variable selection, (2) data collection, (3) data preprocessing, (4) training, testing and validation sets, (5) neural network paradigms, (6) evaluation criteria, (7) neural network training, and (8) implementation. In each step, some rules of thumb, common mistakes, parameter selection, and points of disagreement among neural network developers were discussed.

The success of neural network applications for an individual researcher depends on three key factors. First, the researcher must have the time, patience and resources to experiment. The relatively recent application of neural networks in financial economics and the large number parameters means that only broad rules of thumb have been published and much experimentation and imagination is in order. Second, the neural network software must allow automated routines such as walk-forward testing, optimization of hidden neurons, and testing of input variable combinations; either through direct programming or the use of batch/script files. Many programs available are inadequate for serious development work and will quickly frustrate even the most determined researcher. Third, the researcher must maintain a good set of records that list all parameters for each network tested since any parameter listed in Table 2.1 may turn out to cause a significant change in neural network performance. In this way, a library of what is successful and what is not is built up.

Table 2.1 Common Parameters in Designing a Backpropagation Neural Network

Data Preprocessing

frequency of data - daily, weekly, monthly, quarterly

type of data - technical, fundamental

method of data sampling -

method of data scaling - maximum/minimum, mean/standard deviation

Training

learning rate per layer

momentum term

training tolerance

epoch size

learning rate limit

maximum number of runs

number of times to randomize weights

size of training, testing, and validation sets

Topology

number of input neurons

number of hidden layers

number of hidden neurons in each layer

number of output neurons

transfer function for each neuron

error function

Figure 2.1 A Feedforward Neural Network

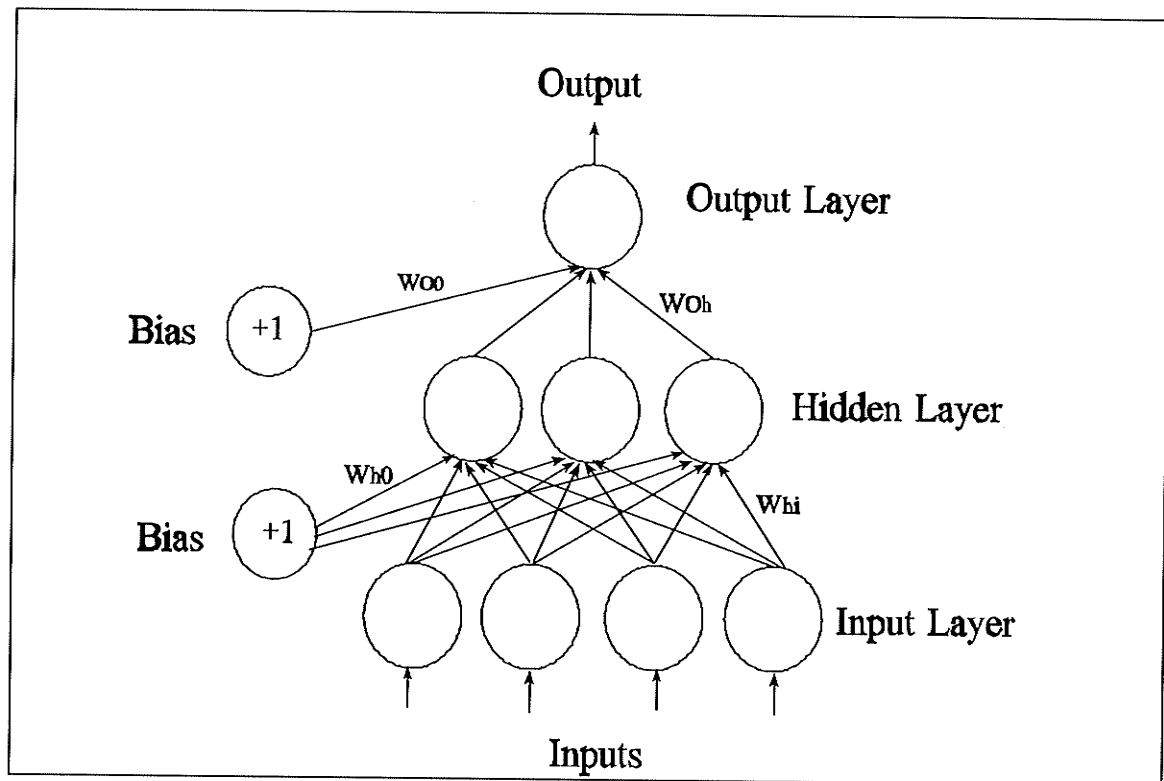


Figure 2.2 Logarithmic Transformation of Monthly Wheat Futures Trading Volume at the Winnipeg Commodity Exchange, Measured in 20 Tonne Job Lots, 1977/78 - 1992/93

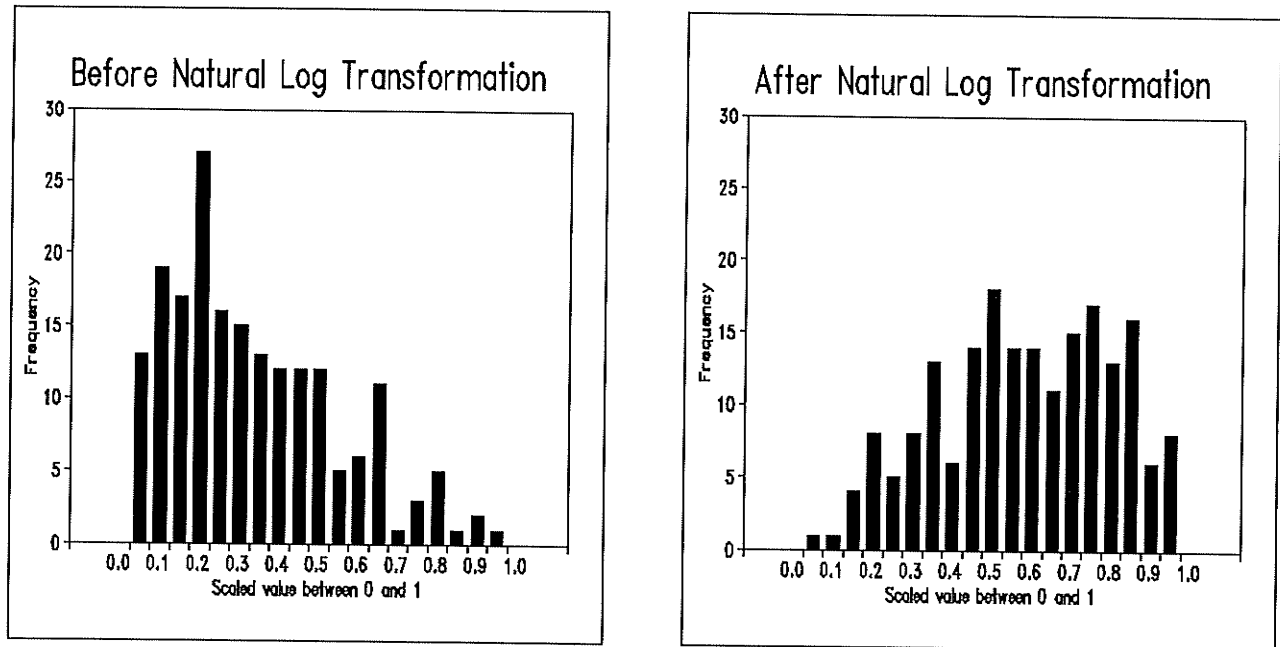


Figure 2.3 Walk-forward/Sliding Windows Testing Routine

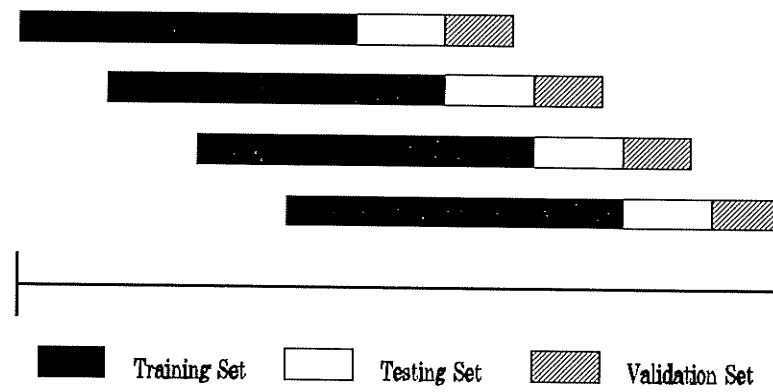


Figure 2.4 Two Common Scaling Methods Applied to Daily Cash S&P 500 Closing Prices, 1962 - 1993

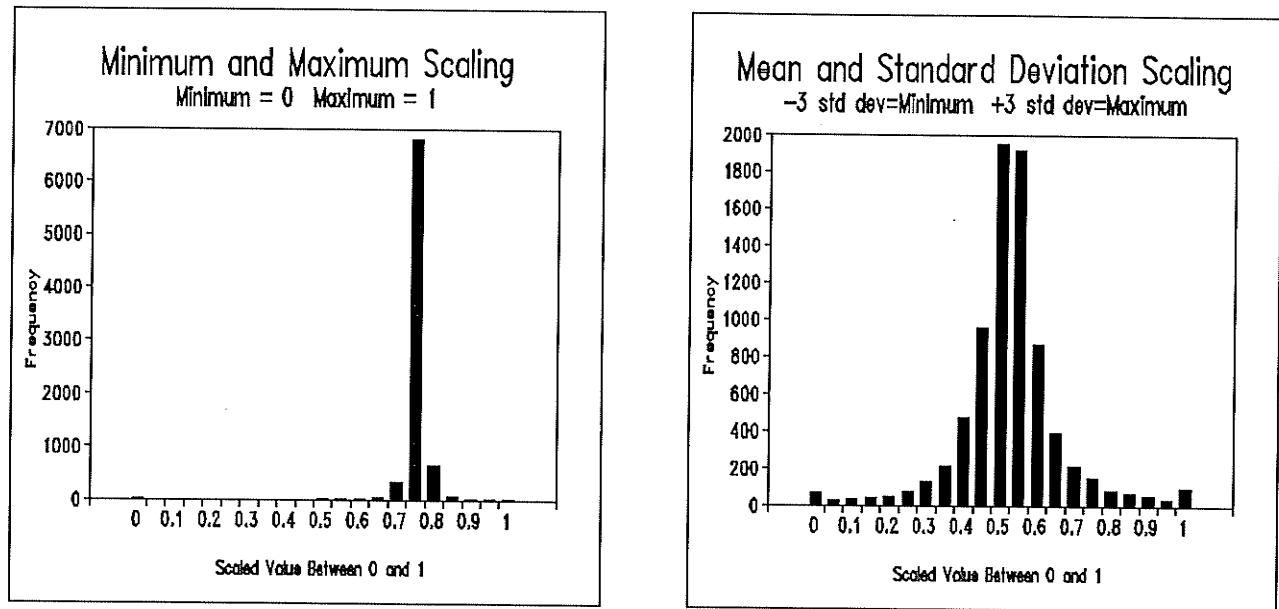


Figure 2.5 Possible Neural Network Training and Testing Set Errors

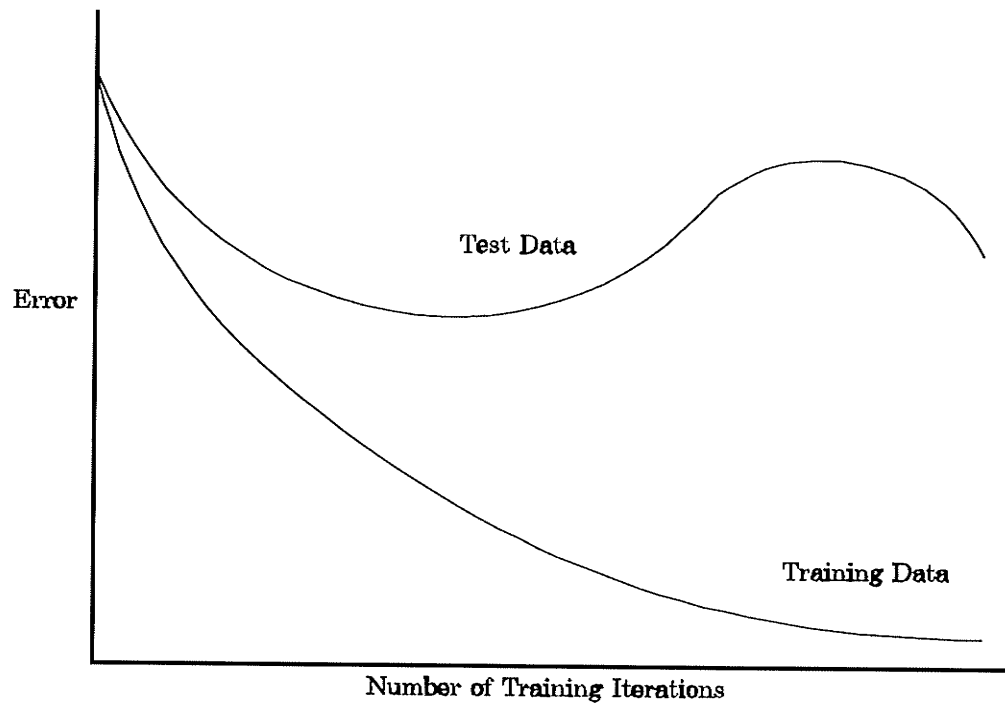
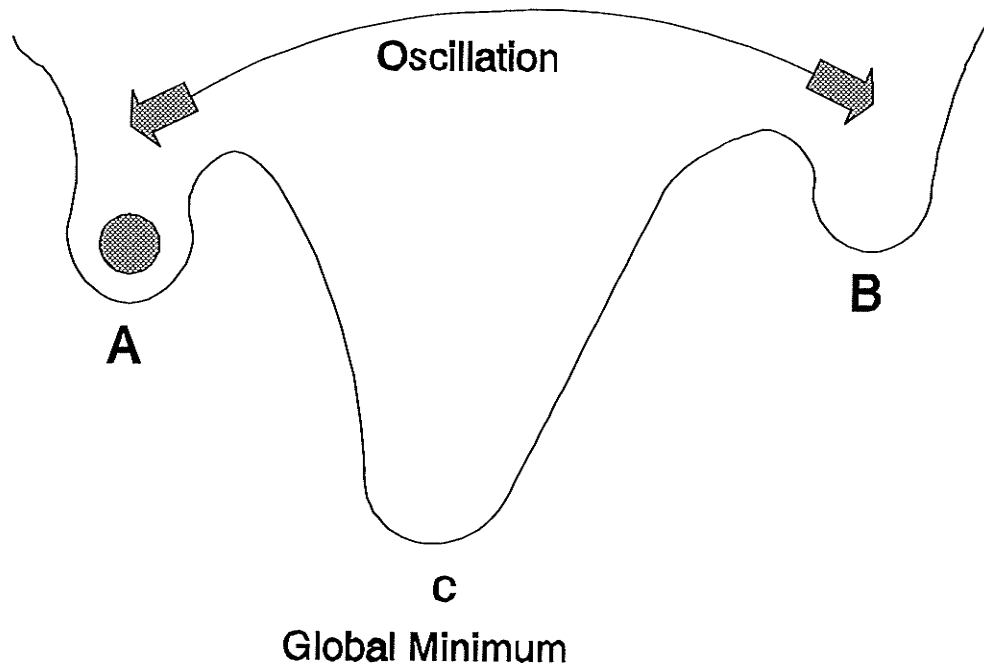


Figure 2.6 Simplified Graphical Representation of a Neural Network Error Surface



CHAPTER 3

FORECASTING FUTURES TRADING VOLUME USING NEURAL NETWORKS

Introduction

Neural networks are a type of artificial intelligence technology that mimic the human brain's powerful ability to recognize patterns. Typical applications in finance include mortgage risk assessment, risk rating of exchange-traded fixed income investments, portfolio selection/diversification, simulation of market behaviour, index construction, identification of economic explanatory variables, and economic forecasting (Trippi and DeSieno, 1992). As universal function approximators, neural networks have the ability to handle nonlinear and chaotic time series data better than traditional forecasting models (Masters, 1993).

This chapter presents a neural network application in time series forecasting. Specifically, the objective of this chapter is to design neural networks for forecasting monthly futures trading volume at the Winnipeg Commodity Exchange (WCE). Commodities examined include barley, canola, flax, oats, rye, and wheat. Separate neural networks are designed for each monthly forecast up to nine months ahead. The forecasting power of the neural networks are evaluated using root mean squared error and mean absolute percentage error, and are then compared to the naive model using the Theil U statistic.

There are no formal models in place to forecast futures trading volume at the WCE. Current practice is to base trading volume forecasts on recent levels and keep any growth projections conservative. Transaction fees on futures trading volume accounted for 21% of the WCE's total annual operating budget in the fiscal year 1992/93. Canola accounted for 63% of all transaction fees over the same period.

The volatility of futures trading volume makes forecasting and hence budgeting difficult. Monthly mean absolute percent changes over the past 16 years are large ranging from 20% to 35% depending on the commodity with standard deviations ranging from a low of 28% for canola and a high of 55% for rye. This is the first study to attempt to forecast futures trading volume on out-of-sample data.

This chapter is divided into four main sections. The first section briefly describes the structure and training of the popular backpropagation neural network which is used in this study. This is followed by a discussion of the forecasting evaluation methods and the data. The third section covers the design of the neural network forecasting models in more detail. It is arranged into four parts: variable selection, data preprocessing, neural network topology, and the training and forecasting procedure. The last section consists of an evaluation and comparison of the neural network models followed by a brief summary.

Backpropagation Neural Networks

Backpropagation (BP) neural networks consist of a collection of inputs and processing units generally known as either neurons, neurodes, or nodes (see Figure 3.1). The neurons in each layer are fully interconnected by connection strengths called weights which, along with the network architecture, store the knowledge of a trained network. In addition to the processing neurons, there is a bias neuron connected to each processing unit in the hidden and output layers. The bias neuron having a value of positive one is analogous to the intercept term in a regression equation and provides the network with more flexibility in classifying nonlinear problems. The neurons and bias terms are arranged into layers; an input layer, one or more hidden layers, and an output layer.

BP networks are a class of feedforward neural networks with supervised learning rules. Feedforward refers to the direction of information flow from the input to the output layer. Inputs are passed through the system once to determine the output. Supervised learning is the process of comparing each of the network's forecasts with the known correct answer and adjusting the weights based on the resulting forecast error. The number of input neurons is equal to the number of independent variables in the model while the output neuron(s) represent the dependent variable(s). The number of hidden layers and neurons within each layer can vary depending on the size and nature of the training set.

The BP training algorithm consists of a two-step process. First, inputs are passed forward to the first (or only) hidden layer. For each hidden layer neuron a weighted sum of the inputs is calculated by multiplying each input neuron's value by its respective weight. The weighted sum is modified by a transfer function which then passes the output forward to the next hidden layer or the output layer. A weighted sum is calculated and passed through its transfer function for each hidden layer and output layer neuron. In the second step, an error signal is calculated by comparing the forecasts at the output layer with the known correct answer. This error signal is 'backpropagated' to the hidden layer(s) and the input layer with the weights being adjusted proportionally to each neuron's contribution to the forecast error. Standard BP employs an optimization method called gradient descent to minimize the sum of squared errors of the output neuron(s).

Forecasting Evaluation Methods

Three criteria are used to evaluate and compare the forecasting power of the neural networks by commodity and by forecast horizon. The first is the mean absolute percentage error (MAPE). It is a measure of the average absolute error for each point forecast made by the neural networks. Dividing each absolute error by its actual observation allows comparisons of forecasting accuracy to be made among the six commodities. MAPE is given by

$$MAPE = \frac{|(P_t - A_t) \div A_t|}{N} \quad (3.1)$$

$$t = t+1, t+2, \dots, N$$

where P and A are the predicted and actual values, respectively.

The second criterion is the root mean squared error (RMSE) which measures the overall performance of a model. It is a commonly used measure of forecast error and goodness-of-fit. The formula for RMSE is

$$RMSE = \sqrt{\frac{\sum (P_t - A_t)^2}{N}} \quad (3.2)$$

The third criterion is the naive forecast. The simplest naive forecast is $P_t = A_{t-1}$. This forecasting standard is implicit in the coefficients of inequality (U_2) proposed by Theil (1966), defined as follows:

$$U_2 = \frac{\sqrt{\frac{\sum (P_t - A_t)^2}{N}}}{\sqrt{\frac{\sum A_t^2}{N}}} \quad (3.3)$$

where $P_i = P_t - A_{t-1}$ and $A_i = A_t - A_{t-1}$. The larger the value of U_2 , the worse the model's forecasting ability. Perfect forecasts are implied when U_2 is equal to zero while U_2 equal to one is equivalent to the forecasting ability of a naive model which only used last period's actual value of the dependent variable. The neural network's forecasting power is worse than the naive model if U_2 is greater than one.

Data

Total monthly transactions for barley, canola, flax, oats, rye, and wheat measured in 20 tonne job lots are used to train and evaluate the neural networks.¹ Each commodity's total trading volume is the sum of the transactions of all the futures delivery months traded during that month. Data for the crop years 1977/78 to 1987/88 are obtained from various issues of the Winnipeg Commodity Exchange (WCE) Statistical Annual. The internal records of the WCE provided total trading volume data from 1988/89 to 1992/93 due to the change in reporting in 1988/89. Open interest is the daily average open interest of all futures delivery months as reported in the WCE Statistical Annuals. Monthly mean cash price in dollars per metric tonne is the daily average close of cash transactions at the WCE obtained from various issues of the Grains Industry Handbook published by the Canada Grains Council. The grade of the cash commodity chosen is based on the grade deliverable at par on the respective futures contract. The futures price series consists of the nearby futures closing prices obtained from the data vendor Technical Tools. Total producer deliveries to licensed elevators in Alberta, Saskatchewan, and Manitoba were obtained from the Canadian Grain Commission's Visible Supplies and Disposition publication.

Variable Selection

Four monthly technical independent variables were chosen as likely predictors of futures trading volume. These include lagged trading volume, open interest, futures price variability, and mean cash prices. The fundamental variable consisted of producer deliveries to licensed elevators in the prairies. Each of these variables is "preprocessed" into a technical indicator before being fed into the neural network as an input.

Open interest reflects the average daily outstanding long or short positions during the month and can be interpreted as a measure of the expected future positions of hedgers since intra-day speculators do not affect open interest. Martell and Wolf (1987) find a statistically positive correlation between monthly open interest and futures trading volume for five metals traded on the COMEX. This is not surprising considering that all other things being equal, volume will always increase whenever open interest changes. Futures trading volume and one-month lagged open interest for all six commodities on the WCE are positively correlated and together could be useful predictors in a neural network forecasting model.

Price volatility has been shown to be positively related to futures trading volume. Karpoff (1987) in a survey paper finds that 18 out of 19 studies investigating the relationship between absolute price change and volume report a positive correlation. Price volatility and volume traded should be positively related because of the joint dependence on a common underlying variable which can be interpreted as the rate of information flow to the market (Najand and Yung, 1991). Price volatility is measured using the standard deviation of daily nearby futures prices. Hedgers would likely increase trading in response to price volatility in order to protect against potentially large cash market losses. Speculators may also be attracted to the futures market as the opportunities for profit are increased with increasing price volatility.

Mean cash prices are included as inputs since short hedgers may believe that as mean cash prices are at historic lows that there is little need to hedge. Conversely, long hedgers may be reluctant to hedge as prices are near their highs.

Producer deliveries are likely a good proxy for hedging demand by grain handling companies and exporters. Hedging demand is very important at the WCE since it is largely a commercial market where hedging volume accounts for the majority of total trading volume.

Data Preprocessing

The design of a reliable neural network forecasting model requires considerable experimentation. The selection of proper inputs, as in any forecasting model, is likely of greatest importance. The transformation of raw data to minimize noise, highlight important relationships, detect trends, and tighten and flatten the distribution of the variable to assist the neural network in learning the relevant patterns is called preprocessing.

Preliminary evaluation of the training and testing sets indicated that higher correlations could be obtained by presenting the neural network with a three-period arithmetic moving average of the inputs. The moving averages smooth the data to minimize noise, detect trends, and may assist in capturing some seasonality. Other less successful preprocessing attempted on the training and testing sets include the use of logarithmic transformations on all variables and the use of a seven-period moving average. The three-period moving average was applied to all inputs for each of the six commodities.

All variables were scaled between zero and one using mean/standard deviation scaling. Observations plus or minus two standard deviations from the mean were mapped to one and zero, respectively. This type of scaling results in a more uniform distribution and helps to suppress the

effect of outliers. Given the small sample size and the scaling method used, no outliers were removed from the training and testing sets or sampling was performed.

All inputs, except for producer deliveries, are lagged from one to ten months depending on the number of periods the network is forecasting ahead. Lagging producer deliveries an additional month accounts for the fact that the information is not publicly released until the middle or end of the following month.

Neural Network Topology

The small training set of 148 observations limited the size of the neural network that could be expected to produce good out-of-sample results by avoiding an overfitted model. Overfitting occurs when a model has too few degrees of freedom relative to its parameters. In the case of neural networks, the number of weights relative to the size of the training set determine the likelihood of overfitting. The number of input neurons is fixed at five for all neural networks which should balance the need for a sufficient set of independent variables while at the same time limiting the number of weights in the neural network.

The number of hidden neurons in the single hidden layer are selected experimentally based on the testing set performance of each neural network. Three, five, seven, and nine hidden neurons were evaluated resulting in a total of 24, 36, 48, and 60 weights, respectively, including bias terms and output neurons. Networks with more than nine hidden neurons were not evaluated because overfitting is very likely to occur as the number of training facts relative to the number of weights approaches two to one. Increments of two for the hidden neurons allow a reasonable range to be tested while keeping computational times feasible.

One neural network was designed for each of the nine monthly forecasts. The output neuron used in each neural network represented the trading volume for one to nine months ahead. For example, one neural network forecasted next month's trading volume while another one specialized in forecasting six months ahead. In all, nine neural networks were trained for each commodity with each differing only by the number of step ahead forecasts.

A three-layer fully interconnected backpropagation neural network is used in this study. Transfer functions for the input neurons are linear while all others are sigmoid. Each neural network is trained to minimize the sum of squared errors of all training facts.

Training and Forecasting Procedure

The multidimensional error surface present in multi-layer neural networks often contains many local minima in which the training algorithm can become "stuck". Local minima are believed to be especially troublesome for networks with a small number of hidden neurons, as is the case here. (Masters, 1993). Training each neural network multiple times using randomly selected starting weights is one common method to increase the likelihood of not becoming trapped in a local minimum. In this study, each neural network is trained seven times for 5,000 iterations using randomly selected starting weights which should provide for a reasonable opportunity of reaching a global minimum.

The network with the highest mean correlation on the testing set out of the seven random sets of starting weights is saved by the neural network. The testing set was randomly selected from the first 168 observations. Once selected, the 20 testing observations were removed from the training set leaving 148 training facts. Randomly selecting testing observations is used to insure that the testing set contains both volume decreases and increases of varying magnitudes. In this

way, better out-of-sample performance is possible.

The neural network with the lowest MAPE on the testing set for each commodity is used to forecast futures trading volume on the validation set. The validation set consists of the 24 observations from the crop years 1991/92 and 92/93. In situations where the forecasting performance on the testing set is similar between two networks, the network with the smallest number of hidden neurons will always be selected.

The nine separate forecasts for each of the six commodities, tested over four different hidden neurons, with seven sets of random starting weights resulted in a total of 1,512 networks that required training. Training time for all six commodities was approximately 110 hours on a 486-DX2 66 Mhz. computer using the neural network software NTRAIN from Scientific Consultant Services. Extensive use of its batch/script file capability was made in order to automate the training and testing procedure.

Evaluation and Comparison

Table 3.1 presents the Theil U, RMSE and MAPE statistics for the six commodities examined over a nine month forecast horizon. Overall, the results are mixed with forecasting performance on some commodities much better than others.

The Theil U statistic compares the neural network forecasting power relative to a naive model which only uses last period's trading volume as its independent variable. The neural networks generally outperformed the naive model, as 36 of the 54 Theil U values in Table 3.1 are less than one. However, notable exceptions are barley and rye for which the Theil U statistic is greater than one in 17 out of 18 cases. One possible explanation for the poor performance of both barley and rye is the large variability during the training and testing period as compared to very

little fluctuations for all 24 out-of-sample observations. For example, barley trading volume during the training period regularly ranged from 20,000 to over 100,000 contracts per month (see Figure 3.2). During the validation period, trading volume ranged from 9,000 to 25,000 contracts and was in a general decline. Outliers were not removed and this may account for forecasts that are consistently much too high. The same argument can be made for rye.

The forecasting performance of the neural networks relative to the naive model remains strong throughout the nine-month forecast horizon and does not appear to deteriorate as the forecast horizon increases. One possible explanation for this phenomenon is that the naive model is deteriorating while the neural network's nearby forecasts are not as good. The naive model is expected to perform well for the one-step ahead forecast as last period's value is likely a very good indicator of next period's trading volume. The neural networks by using a moving average may lose out on the one-step ahead forecast, but gain accuracy for the longer term forecasts. This suggests that a possible improvement for the neural networks may be to include last period's trading volume as an input for the one-step ahead forecast to allow the network to, at least, emulate the naive model. For the longer term forecasts, the length of the moving average can match the number of periods forecasted ahead. For example, a seven month ahead forecast would use a seven or eight-month moving average.

The MAPE vary widely by commodity and by forecast horizon. Canola and wheat exhibit the lowest MAPE ranging from 13% to 25% while barley and rye show the highest values. The strong forecasting performance for canola and wheat is likely attributable to the relatively low mean absolute percentage changes from month to month and the consistency in volume variability from the training/testing sets to the validation set. The RMSE follows a similar pattern as MAPE although there are instances in which the two statistics do not move in the same direction.

In terms of the network topology, the number of hidden neurons also vary by commodity and by forecast horizon. Out of the 54 networks listed in Table 3.2, 23, 15, and 16 used three, five, and seven hidden neurons, respectively. Nine hidden neurons were not able to perform well on the testing set which is likely indicative of some overfitting during training. The strong showing of networks with three hidden neurons indicates that a small neural network can approximate the function quite well. However, a range of hidden neurons should still be tested as the network topology varied considerably by commodity and by forecast horizon.

Overall, the results are encouraging for the WCE in terms of its financial implications for two main reasons. First, the relatively strong forecasting performance by the neural networks for canola and wheat is good news because these two commodities combined accounted for over 76% of total transaction fees during 1991/92 and 92/93. On the other hand, the poor forecasting performance for rye and barley is less of a problem because together these two commodities accounted for slightly over 9% of total transaction fees during the two-year validation period. Second, the results are encouraging because the forecasts do not appear to deteriorate as the forecast horizon increases relative to the naive model which is currently in place.

Summary

This study used the popular backpropagation neural network to forecast futures trading volume for six commodities traded on the Winnipeg Commodity Exchange. The forecasts can assist the WCE in budgeting its transaction fees. Separate neural forecasting network models were created for barley, canola, flax, oats, rye, and wheat. Trading volume was forecasted up to nine months ahead with a single neural network specializing in each step ahead forecast. The five independent variables used as inputs to the neural networks include three-period moving averages

of lagged trading volume, open interest, mean cash prices, futures price variability, and producer grain deliveries. Three, five, seven, and nine hidden neurons were trained and evaluated on the testing set.

The results indicate that the neural networks are able to forecast up to nine months ahead and outperform the naive model for all commodities except barley and rye. The neural network forecasts relative to the naive model currently in place to forecast futures trading volume at the WCE do not deteriorate as the forecast horizon increases. The results suggest that improvements to the neural networks could be made by matching the moving average length of the inputs to the number of months forecasted ahead. For example, a seven month ahead forecast would use a seven or eight-month moving average.

In terms of network topology, neural networks with three hidden neurons were selected for 23 out of the 54 networks. Networks with five and seven hidden neurons were selected almost equally often while no network with nine hidden neurons performed well on the testing set.

Endnotes

1. Prior to the 1988/89 crop year volume of trade published in the Winnipeg Commodity Exchange Statistical Annual included the following transactions: all broker allocations to principals, give-ups, error corrections and exchanges of futures for physicals. These transactions are no longer included in the volume figures. Since transaction fees received by the WCE are based on total transactions, which include both long and short trades, it is more appropriate for the neural networks to be forecasting total transactions for budgeting purposes rather than the recently revised trading volume figures.

Figure 3.1 A Feedforward Neural Network

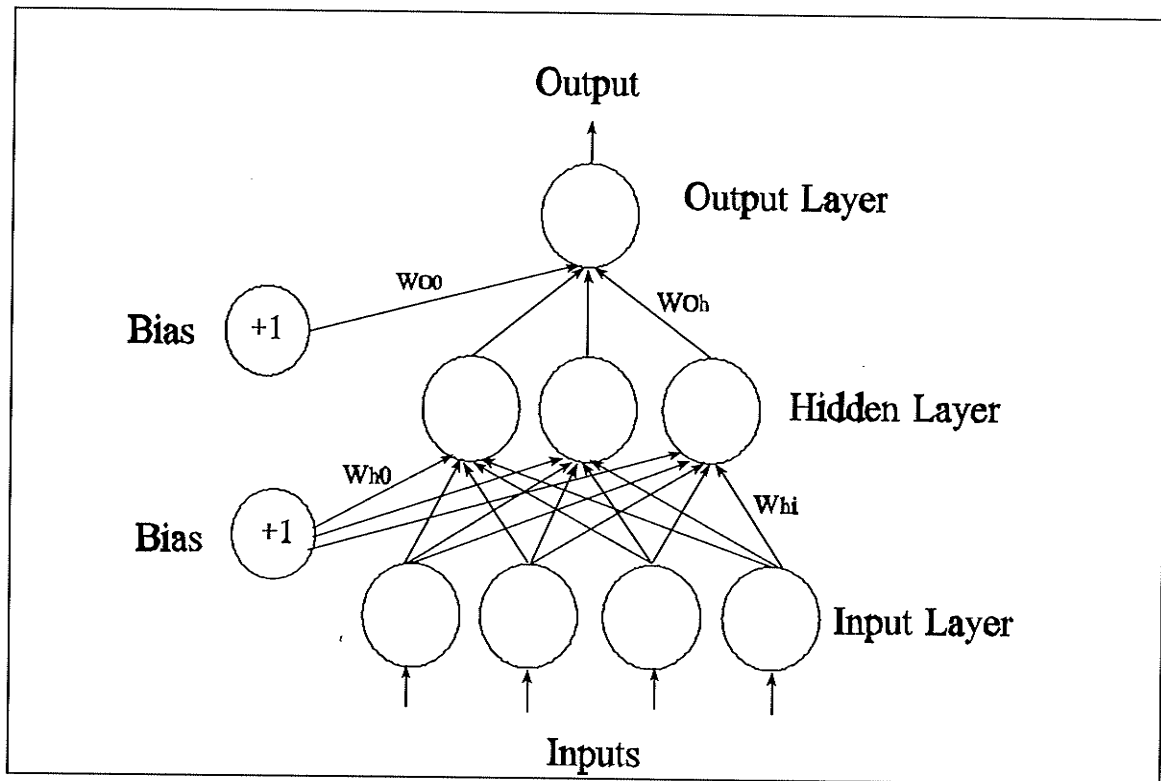


Table 3.1 Results of Out-of-Sample Futures Trading Volume Forecasts by Neural Network Models, Winnipeg Commodity Exchange, 1991/92 - 1992/93

Forecast Horizon	<u>RMSE^a</u>						<u>MAPE^b</u>						<u>Theil U</u>					
	Barley	Canola	Flax	Oats	Rye	Wheat	Barley	Canola	Flax	Oats	Rye	Wheat	Barley	Canola	Flax	Oats	Rye	Wheat
1 Month	5,891	23,135	7,947	4,098	2,908	8,107	37.4	13.1	34.3	53.3	52.4	19.0	1.09	0.82	0.89	1.17	1.13	0.85
2 Months	5,879	31,246	8,324	4,782	2,918	8,214	36.9	18.9	33.7	58.0	58.5	19.3	1.18	0.98	0.96	0.98	1.14	0.86
3 Months	9,188	24,611	7,190	3,864	3,896	9,031	65.6	15.4	30.9	52.5	155.4	18.6	1.82	0.81	0.88	0.73	1.55	0.94
4 Months	6,796	32,487	6,455	3,317	2,949	6,603	42.8	22.5	29.5	49.1	98.5	18.0	1.30	0.88	0.69	0.65	1.02	0.66
5 Months	6,728	27,471	7,712	4,101	3,137	7,043	43.5	17.2	35.0	44.3	124.9	19.0	1.46	0.89	1.08	0.94	1.37	0.67
6 Months	9,213	31,241	11,328	3,119	3,648	7,553	58.5	20.9	51.2	47.2	70.3	18.4	1.45	0.83	1.23	0.69	1.36	0.73
7 Months	13,065	25,076	9,807	4,146	3,124	9,403	89.2	15.7	44.9	58.9	91.9	25.0	2.85	0.64	1.17	0.89	1.20	1.15
8 Months	16,374	30,361	7,110	5,213	2,579	8,255	113.3	19.3	33.5	70.3	56.1	20.3	2.87	0.90	0.85	0.95	0.83	0.89
9 Months	15,111	27,933	9,248	5,010	3,016	9,303	107.5	19.4	42.8	67.1	119.2	24.7	2.39	0.78	1.10	0.83	1.13	0.76

^aRoot Mean Squared Error.

^bMean Absolute Percentage Error.

Figure 3.2 Futures Trading Volume at the Winnipeg Commodity Exchange, 1977/78 - 1992/93

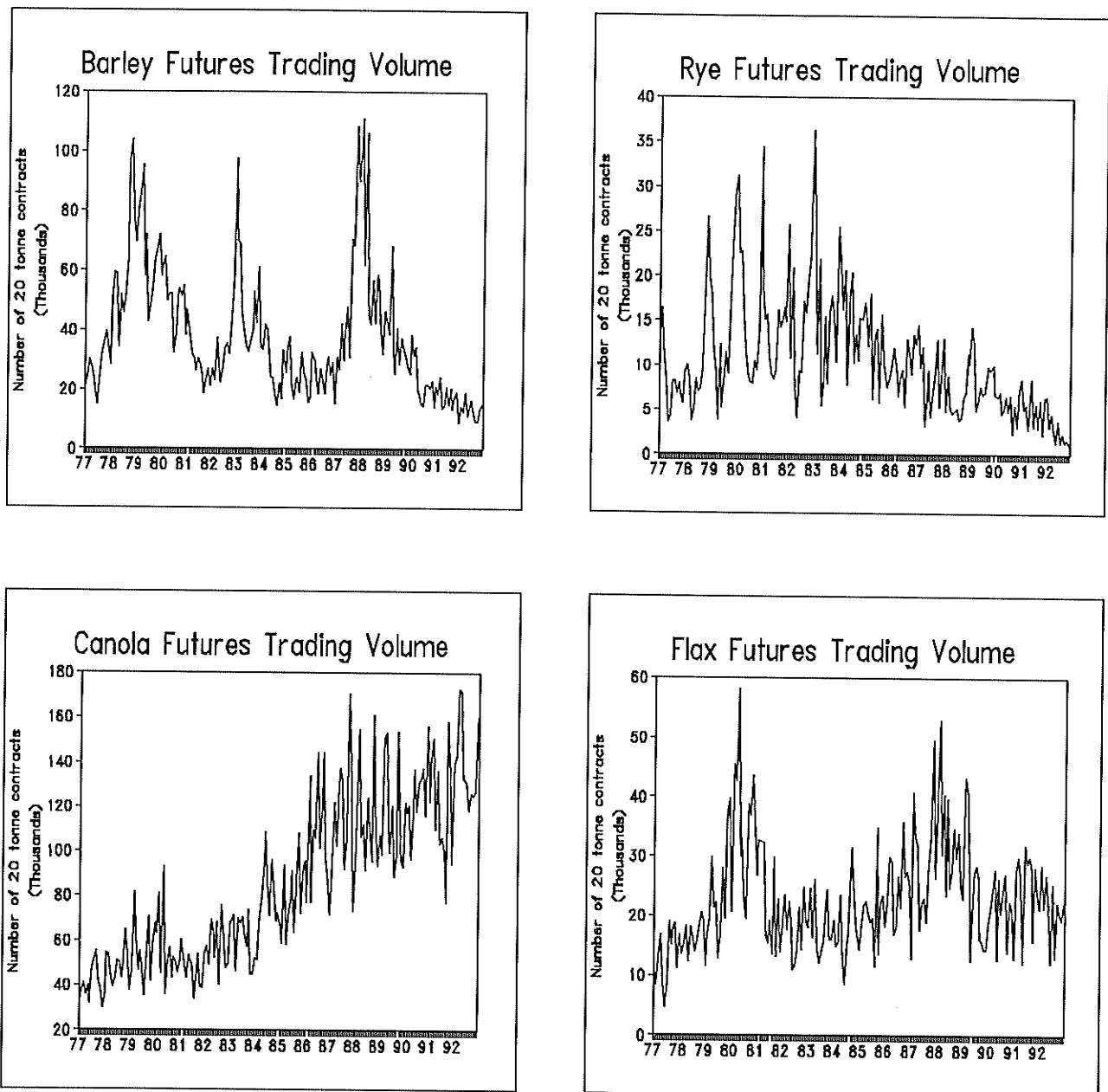


Figure 3.2 (Continued) Futures Trading Volume at the Winnipeg Commodity Exchange,
1977/78 - 1992/93

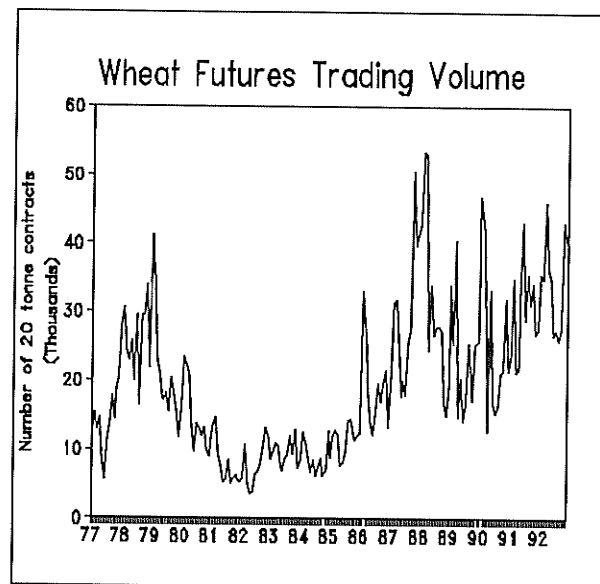
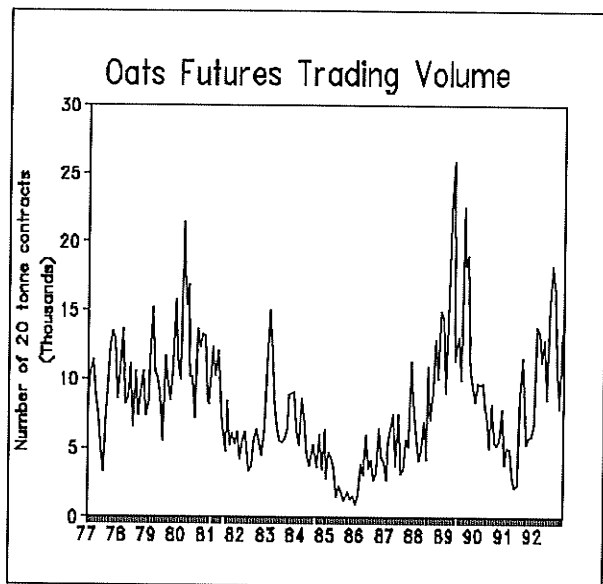


Table 3.2 Number of Hidden Neurons Selected in Final Neural Network Forecasting Models Based on Testing Set Performance

Forecast Horizon	Barley	Canola	Flax	Oats	Rye	Wheat
1 Month	7	7	3	3	7	7
2 Months	3	7	3	3	7	7
3 Months	5	3	3	5	5	5
4 Months	7	3	3	3	5	7
5 Months	7	3	5	3	7	7
6 Months	3	3	3	3	7	7
7 Months	3	5	5	3	5	5
8 Months	3	5	3	5	5	7
9 Months	3	3	5	3	5	7

CHAPTER 4

SUMMARY

The objective of this study was to provide an introductory guide to forecasting economic time series data using neural networks. An eight-step procedure for designing a neural network forecasting model was presented including a discussion of tradeoffs in parameter selection, some common pitfalls, and points of disagreement among practitioners. The design procedure was then illustrated by developing neural networks to forecast monthly futures trading volume on the Winnipeg Commodity Exchange (WCE). The forecasts are important for the WCE in its budgeting as transaction fees are a significant source of revenue.

Designing a Neural Network Forecasting Model

Chapter two describes an eight-step procedure for designing a neural network forecasting model. The design steps are: (1) variable selection, (2) data collection, (3) data preprocessing, (4) training, testing, and validation sets, (5) neural network paradigms, (6) evaluation criteria, (7) neural network training, and (8) implementation. Among the many topics discussed, the most important ones are data preprocessing, overfitting, and neural network training.

Data preprocessing is the transformation of the input and output variables to minimize noise, highlight important relationships, detect trends, and tighten and flatten the distribution of the variable to assist the neural network in learning the relevant patterns. There are many ways to present data to a neural network such as first differencing, logarithmic transformations, and a variety of technical indicators. As a result, preprocessing usually involves much trial and error on the training and testing sets.

Overfitting occurs when a forecasting model has too few degrees of freedom. In other words, it has relatively few observations in relation to its parameters and therefore it is able to memorize individual points rather than learn the general patterns. In the case of neural networks, the number of weights, which is inexorably linked to the number of hidden layers and neurons, and the size of the training set (number of observations) determine the likelihood of overfitting (Baum and Haussler, 1989; Masters, 1993). The solution is to increase the size of the training set and/or decrease the number of hidden neurons or layers.

Neural network training involves iteratively presenting the network with examples of the correct known answers. The objective of training is to find the set of weights between the neurons that determine the global minimum of the error function. Unless the model is overfitted, this set of weights should provide good generalization. The backpropagation neural network uses a gradient descent training algorithm which adjusts the weights to move down the steepest slope of the error surface. Finding the global minimum is not guaranteed since the error surface can include many local minima in which the algorithm can become 'stuck'. A momentum term and five to ten random sets of starting weights can improve the chances of reaching a global minimum.

Forecasting Futures Trading Volume Using Neural Networks

Chapter three presents backpropagation neural network models to forecast monthly futures trading volume on the Winnipeg Commodity Exchange (WCE). Commodities examined include barley, canola, flax, oats, rye, and wheat. The inputs to the neural networks consist of a three-period moving average of lagged trading volume, open interest, mean cash prices, nearby futures price variability, and producer deliveries to licensed elevators in the prairies. Separate neural networks were designed and trained for each of the nine monthly forecasts.

The forecasting power of the neural networks were evaluated using root mean squared error (RMSE) and mean absolute percentage error (MAPE) and compared to the naive model using the Theil U statistic. Results indicate that the neural networks are able to forecast up to nine months ahead and outperform the naive model for all commodities except barley and rye. The neural network forecasts relative to the naive model currently in place to forecast futures trading volume at the WCE do not deteriorate as the forecast horizon increases. Neural networks with three hidden neurons were used for 23 out of the 54 networks. Networks with five and seven hidden neurons performed almost equally well while no network with nine hidden neurons was selected.

Limitations and Suggestions for Future Research

This study showed that neural networks are able to forecast futures trading volume up to nine months ahead for four commodities at the Winnipeg Commodity Exchange (WCE). The study could be expanded to other commodities traded on other exchanges. The preprocessing consisted of a simple three-period moving average. The results suggest that improvements to the neural networks could be made by matching the moving average length of the inputs to the number of months forecasted ahead. In addition, the backpropagation training algorithm cannot guarantee convergence. Therefore, it is possible that some of the 54 neural networks tested on the validation set may not have trained sufficiently. Another limitation of this study is the small data set which impacts the training, testing, and validation sets. The small training set of 148 observations provides few examples for the neural network to learn the patterns. The randomly selected testing set of only 20 observations may not have been sufficient to adequately test all types of possible patterns. Finally, the validation set of only 24 observations presented problems for barley and rye as apparently the patterns learned in the training set were different in the validation set.

REFERENCES

- Baily, D. and D.M. Thompson. "Developing Neural Network Applications." AI Expert, September (1990):33-41.
- Baum, E.B. and D. Haussler. "What Size Net Gives Valid Generalization?" Neural Computation, 6(1989):151-160.
- Blum, A. Neural Networks in C++: An Object-Oriented Framework for Building Connectionist Systems, John Wiley & Sons Inc., New York, 1992.
- Caudill, M. "The View from Now." AI Expert, June(1992):24-31.
- Deboeck, Guido J., Editor. Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, John Wiley & Sons Inc., New York, 1994.
- Deboeck, Guido J. and Masud Cader. "Trading U.S. Treasury Notes with a Portfolio of Neural Net Models." in Deboeck, Guido J., Editor. Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, John Wiley & Sons Inc., New York, (1994):102-122.
- Egan, Jack. "Artificially Intelligent Investing." U.S. News & World Report, 56(March 15, 1993):73
- Ersoy, O. Tutorial at Hawaii International Conference on Systems Sciences, January 1990.
- Hamm, Lonnie, B. Wade Brorsen and Ramesh Sharda. "Futures Trading With a Neural Network." NCR-134 Conference on Applied Commodity Analysis, Price Forecasting, and Market Risk Management Proceedings, Chicago, (1993):286-296.
- Hecht-Nielsen, Robert. Neurocomputing, Addison Wesley Publishing Company, Menlo Park, CA, 1989.
- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks are Universal Approximators". Neural Networks, 2(1989):359-366.
- Kao, G.W. and C.K. Ma. "Memories, Heteroscedasticity and Prices Limit in Currency Futures Markets." Journal of Futures Markets, 12(1992):672-692.
- Karpoff, J.M. "The Relation between Price Changes and Trading Volume: A Survey." Journal of Financial Quantitative Analysis, 22(1987):109-126.
- Katz, Jeffrey Owen. "Developing Neural Network Forecasters For Trading." Technical Analysis of Stocks and Commodities, April(1992):58-70.

Klaussen, Kent L. and J. William Uhrig. "Cash Soybean Price Prediction with Neural Networks." NCR-134 Conference on Applied Commodity Analysis, Price Forecasting, and Market Risk Management Proceedings, Chicago, (1994):56-65.

Klimasauskas, Casimir C. "Applying Neural Networks." in Trippi, Robert R. and Efraim Turban, Editors. Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance, Probus Publishing Company, Chicago, (1993):64-65.

Kohzadi, N. "Neural Networks Versus Time Series Models for Forecasting Commodity Prices" Ph.D. Thesis. Department of Agricultural Economics and Farm Management, University of Manitoba, Canada, 1993.

Levich, R.M. and L.R. Thomas. "The Significance of Technical Trading Rule Profits in the Foreign Exchange Market: A Bootstrap Approach." in Strategic Currency Investing - Trading and Hedging in the Foreign Exchange Market, Probus Publishing Company, Chicago, (1993):336-365.

Lippman, R.P. "An Introduction to Computing with Neural Nets." IEEE ASSP Magazine, April(1987):4-22.

Martell, T.F. and A.S. Wolf. "Determinants of Trading Volume in Futures Markets". Journal of Futures Markets, 7(1987):233-244.

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

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

Mendelsohn, Lou. "Training Neural Networks." Technical Analysis of Stocks and Commodities, November (1993):40-48.

Najand, Mohammad and Kenneth Yung. "A GARCH Examination of the Relationship between Volume and Price Variability in Futures Markets." Journal of Futures Markets, 11(1991):613-621.

Nelson, Marilyn McCord and W.T. Illingworth. A Practical to Guide to Neural Nets, Addison Wesley Publishing Company, Reading, Massachusetts, 1991.

Odom, Marcus D. and Ramesh Sharda. "A Neural Network Model For Bankruptcy Prediction." Proceedings of the IEEE International Conference on Neural Networks, San Diego, (1992):II163-II168.

Salchenberger, Linda M., E. Mine Cinar and Nicholas A. Lash. "Neural Networks: A New Tool For Predicting Thrift Failures." Decision Sciences, 23(July/August 1992):899-916.

Scientific Consultant Services, Inc. Ntrain V1.02. 20 Stagecoach Road, Seledon, New York, 11784.

Sharda, Ramesh and Rajendra B. Patil. "A Connectionist Approach to Time Series Prediction: An Empirical Test." in Deboeck, Guido J., Editor, Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, John Wiley & Sons Inc., New York, (1994):451-464.

Tang, Z., C. de Almedia and P.A. Fishwick. "Time Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology". International Workshop on Neural Networks, Auburn, Alabama, January 1990.

Theil, H. Applied Economic Forecasting, Rand McNally & Company, New York, 1966.

Tomek, William G. and Scott F. Querin, "Random Processes in Prices and Technical Analysis." Journal of Futures Markets, 4(1984):15-23.

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

Trippi, Robert R. and Efraim Turban, Editors. Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance, Probus Publishing Company, Chicago, 1993.

White, Halbert. "Learning in Neural Networks: A Statistical Perspective." Neural Computation, 4(1989):425-464.

White, Halbert, A.R. Gallant, K. Hornik, M. Stinchcombe, and J. Wooldridge. Artificial Neural Networks: Approximation and Learning Theory, Blackwell Publishers, Cambridge, Massachusetts, 1992.

Yoon, Youngohc and George Swales. "Predicting Stock Price Performance: A Neural Network Approach." Proceedings of the 24th Annual International Conference of Systems Sciences, Chicago, (1991):156-162.