‘An Artificial Neural Networks Primer with Financial Applications Examples in Financial Distress Predictions and Foreign Exchange Hybrid Trading System ’

by

Dr Clarence N W Tan, PhD

Bachelor of Science in Electrical Engineering Computers (1986),
University of Southern California, Los Angeles, California, USA
Master of Science in Industrial and Systems Engineering (1989),
University of Southern California Los Angeles, California, USA
Masters of Business Administration (1989)
University of Southern California Los Angeles, California, USA
Graduate Diploma in Applied Finance and Investment (1996)
Securities Institute of Australia
Diploma in Technical Analysis (1996)
Australian Technical Analysts Association
Doctor of Philosophy
Bond University (1997)

URL: http://w3.to/ctan/

E-mail: ctan@computer.org

School of Information Technology, Bond University, Gold Coast, QLD 4229, Australia

Table of Contents

1. INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORKS 2

1.1 INTRODUCTION 2
1.2 ARTIFICIAL INTELLIGENCE 2
1.3 ARTIFICIAL INTELLIGENCE IN FINANCE 4
1.3.1 Expert System 4
1.3.2 Artificial Neural Networks in Finance 4
1.4 ARTIFICIAL NEURAL NETWORKS 5
1.5 APPLICATIONS OF ANNS 7
1.6 REFERENCES 10

2. AN ARTIFICIAL NEURAL NETWORKS' PRIMER 14

2.1 CHRONICLE OF ARTIFICIAL NEURAL NETWORKS DEVELOPMENT 14
2.2 BIOLOGICAL BACKGROUND 16
2.3 COMPARISON TO CONVENTIONAL COMPUTATIONAL TECHNIQUES 17
2.4 ANN STRENGTHS AND WEAKNESSES 19
2.5 BASIC STRUCTURE OF AN ANN 20
2.6 CONSTRUCTING THE ANN 21
2.7 A BRIEF DESCRIPTION OF THE ANN PARAMETERS 22
2.7.1 Learning Rate 22
2.7.2 Momentum 22
2.7.3 Input Noise 23
2.7.4 Training and Testing Tolerances 23
2.8 DETERMINING AN EVALUATION CRITERIA 23
2.9 REFERENCES 24

3. THE TECHNICAL AND STATISTICAL ASPECTS OF ARTIFICIAL NEURAL NETWORKS 27

3.1 ARTIFICIAL NEURAL NETWORK MODELS 27
3.2 NEURODYNAMICS 27
3.2.1 Inputs 27
3.2.2 Outputs 28
3.2.3 Transfer (Activation) Functions 28
3.2.4 Weighing Schemes and Learning Algorithms 30
3.3 NEURAL NETWORKS ARCHITECTURE 30
3.3.1 Types of interconnections between neurons 30
3.3.2 The Number of Hidden Neurons 31
3.3.3 The Number of Hidden Layers 32
3.3.4 The Perceptron 32
3.3.5 Linear Separability and the XOR Problem 34
3.3.6 The Multilayer Perceptron 37
3.4 LEARNING 39
3.4.1 Learning Algorithms 39
3.5 STATISTICAL ASPECTS OF ARTIFICIAL NEURAL NETWORKS 43
3.5.1 Comparison of ANNs to Statistical Analysis 43
3.5.2 ANNs and Statistical Terminology 43
3.5.3 Similarity of ANN Models to Statistical Models 44
3.5.4 ANNs vs. Statistics 45
3.5.5 Conclusion of ANNs and Statistics 46
3.6 REFERENCES 47

4. USING ARTIFICIAL NEURAL NETWORKS TO DEVELOP AN EARLY WARNING PREDICTOR FOR CREDIT UNION FINANCIAL DISTRESS 51

4.1 INTRODUCTION 51
4.2 EXISTING STUDIES: METHODOLOGICAL ISSUES 51
4.3 APPLICATIONS OF ANNS IN PREDICTING FINANCIAL DISTRESS 53
4.4 DATA AND TESTING METHODOLOGY 54
4.4.1 In-sample (Training) and Out-of Sample (Validation) Data Sets 54
4.4.2 Input (Independent) Variables 55
4.5 ANN TOPOLOGY AND PARAMETER SETTINGS 57
4.5.1 Learning Rate 58
4.5.2 Momentum 58
4.5.3 Input Noise 59
4.5.4 Training and Testing Tolerances 59
4.6 RESULTS 59
4.6.1 Training Set (In-sample) Discussion 60
4.6.2 Validation Set (Out-of-sample) Result Comparison 60
4.6.3 Validation Set (Out-of-sample) Evaluation 61
4.6.4 Result Summary of Type I and Type II Errors 70
4.7 ASSUMPTIONS AND LIMITATION OF METHODOLOGY 70
4.8 CONCLUSION 70
4.9 MANAGERIAL AND IMPLEMENTATION ISSUES 71
4.10 FUTURE RESEARCH 72
4.11 REFERENCES 74
4.12 APPENDIX A: THE 10 LARGEST PREDICTED CONDITIONAL PROBABILITIES. 77
4.13 APPENDIX B: IN-SAMPLE AND OUT-OF-SAMPLE RESULTS 78

5. APPLYING ARTIFICIAL NEURAL NETWORKS IN FINANCE: A FOREIGN EXCHANGE MARKET TRADING SYSTEM EXAMPLE WITH TRANSACTIONS COSTS .. 80

5.1 INTRODUCTION 80
5.2 LITERATURE REVIEW 81
5.2.1 Development of the Australian Dollar Exchange Rate Market 81
5.2.2 Studies of the Efficiency of the Australian Foreign Exchange Market 81
5.2.3 Literature Review on Trading Systems and ANNs in Foreign Exchange 82
5.3 FOUNDATIONS OF TRADING METHODOLOGY 83
5.4 DATA 84
5.4.1 ANN Data Sets: Training, Testing and Validation 84
5.4.2 AR Data Set: In-sample and Out-of-Sample 85
5.5 FINANCIAL TRADING SYSTEM STRUCTURE 86
1.1.1 The Rules Structure Underlying Approach 88
1.1.2 Calculating the Arbitrage Boundaries 89
1.1.3 Rules Structure 90
1.1.4 Filter Rules 90
1.6 ANN TOPOLOGY AND PARAMETER SETTINGS 91
1.6.1 Model Selection 91
1.7 PROBLEMS ENCOUNTERED IN THE RESEARCH 93
1.8 RESULTS 94
1.8.1 Perfect Foresight Benchmark Comparison 94
1.8.2 Performance Metrics 94
1.8.3 Profitability of the Models 96
1.8.4 PF Model’s Profitability Performance 97
1.8.5 AR Model’s Profitability Performance 97
1.8.6 ANNOAR Model’s Profitability Performance 98
1.8.7 ANNWAR Model’s Profitability Performance 98
1.8.8 Forecast Comparisons 110
5.9 ASSUMPTIONS AND LIMITATION OF METHODOLOGY 111
5.10 CONCLUSION 111
5.11 MANAGERIAL AND IMPLEMENTATION ISSUES 112
5.12 FUTURE RESEARCH 113

5.13 REFERENCES 114
5.14 APPENDIX C: INTRODUCTION TO FOREIGN EXCHANGE TRADING TECHNIQUES 118
5.14.1 Technical Analysis 118
5.14.2 Fundamental Analysis 119
5.14.3 ANNs and Trading Systems 119
5.14.4 Basic Structure of a Rule-based Financial Trading System 120
5.14.5 Selection of Indicators/Data Input to ANN 122

Chapter 1: Introduction to Artificial Intelligence and Artificial Neural Networks

“The beginning is the most important part of the work.”
Plato (c. 428-348 BC,) Republic

1. Introduction to Artificial Intelligence and Artificial Neural Networks

1.1 Introduction

There can be little doubt that the greatest challenge facing managers and researchers in the field of finance is the presence of uncertainty. Indeed risk, which arises from uncertainty, is fundamental to modern finance theory and, since its emergence as a separate discipline, much of the intellectual resources of the field have been devoted to risk analysis. The presence of risk, however, not only complicates decision financial making, it creates opportunities for reward for those who can analyze and manage risk effectively.

By and large, the evolution of commercial risk management technology has been characterized by computer technology lagging behind the theoretical advances of the field. As computers have become more powerful, they have permitted better testing and application of financial concepts. Large-scale implementation of Markowitz's seminal ideas on portfolio management, for example, was held up for almost twenty years until sufficient computational speed and capacity were developed. Similarly, despite the overwhelming need from a conceptual viewpoint, daily marking to market of investment portfolios has only become a feature of professional funds management in the past decade or so, following advances in computer hardware and software.

Recent years have seen a broadening of the array of computer technologies applied to finance. One of the most exciting of these in terms of the potential for analyzing risk is Artificial Intelligence (AI). One of the contemporary methods of AI, Artificial Neural Networks (ANNs), in combination with other techniques, has recently begun to gain prominence as a potential tool in solving a wide variety of complex tasks. ANN-based commercial applications have been successfully implemented in fields ranging from medical to space exploration.

1.2 Artificial Intelligence

AI has been described as software that behaves in some limited ways like a human being. The word *artificial* comes from the Latin root word *facere arte* which means "make something" thus AI translates loosely to *man made intelligence.* AI has been defined in many ways. Winston [1984] suggests one definition of AI as the study of ideas that enable computers to be intelligent. Rich and Knight [1991] define AI as the study of how to make computers do things which, at the moment, people do better.

The following are some more common definitions and/or descriptions of AI:

- AI is intelligent because it learns;
- AI transforms data into knowledge;
- AI is about intelligent problem solving;
- AI embodies the ability to adapt to the environment, to cope with incomplete or incorrect knowledge.

While artificial intelligence techniques have only recently been introduced in finance, they have a long history of application in other fields. Experience to date across a wide range of non-financial applications has been mixed. Patrick Winston, a leading AI researcher and the head of MIT's AI Laboratory conceded that the traditional AI methods such as search methods, predicate calculus, rule-based expert systems, game-playing, etc. have achieved little progress [Gallant 1994]. The problem domain that traditional AI methods seem to fail in is in the trivial and common sense-type of tasks that humans find easy such as recognizing faces and object, walking, etc.

Therefore, it was natural for AI researchers to turn to nature and the physical laws and processes for inspiration to find better solutions. As a result, many of the contemporary artificial intelligence tools developed in the natural sciences and engineering field have successfully found their way into the commercial world. These include wavelet transformations and finite impulse response filters (FIR) from the signal processing/electrical engineering field, genetic algorithms and artificial neural networks from the biological sciences, chaos theory and simulated annealing from the physical sciences. These revolutionary techniques fall under the AI field as they represent ideas that seem to emulate intelligence in their approach to solving commercial problems. All these AI tools have a common thread in that they attempt to solve problems such as the forecasting and explanation of financial markets data by applying physical laws and processes. Pal and Srimani [1996] state that these novel modes of computation are collectively known as *soft computing* as they have the unique characteristic of being able to exploit the tolerance imprecision and uncertainty in real world problems to achieve tractability, robustness, and low cost. They further state that soft computing are often used to find approximate solution to a precisely (or imprecisely) formulated problem. Huffman of Motorola states that "At Motorola, we call neural networks, fuzzy logic, genetic algorithms and their ilk *natural computing*" [1994].

These contemporary tools are often used in combination with one another as well as with more traditional AI methods such as expert systems in order to obtain better solutions. These new systems that combine one or more AI methods (which may include traditional methods) are known as 'hybrid systems'. An example of a hybrid system is the financial trading system described in Tan [1993] which combines an artificial neural network with a rule-based expert system. Lawrence [1994] preferred to use the term *computer intelligence* to describe expert systems and artificial neural networks as she felt it was less misleading and less controversial in defining the "intelligence" emulated by such systems.

1.3 Artificial Intelligence in Finance

1.3.1 Expert System

Financial analysis falls into the Expert Task Domain of AI as classified by Rich and Knight [1991]. Thus, it is not surprising that the most used AI methods in the financial field have been expert systems. An expert system is a program that is developed by a programmer, known as a knowledge engineer, who may have no domain knowledge of the task at hand with the help of a domain ‘expert’ who may not have any programming expertise. The system is developed by trying to capture the human expert’s knowledge into a set of programming rules that assist in decision making. Hence expert systems are often described as rule-based systems. Expert Systems have been used in medical diagnosis problems, fraud detection, prospecting and mineral detection, etc. The biggest limitation of expert systems is that they require full information about outcomes and therefore deal poorly with uncertainty.

1.3.2 Artificial Neural Networks in Finance

From the range of AI techniques, the one that deals best with uncertainty is the Artificial Neural Network (ANN). Dealing with uncertainty in finance primarily involves recognition of patterns in data and using these patterns to predict future events. Accurate prediction of economic events, such as interest rate changes and currency movements currently ranks as one of the most difficult exercises in finance; it also ranks as one of the most critical for financial survival. ANNs handle these problems better than other AI techniques because they deal well with large noisy data sets. Unlike expert systems, however, ANNs are not transparent, thus making them difficult to interpret.

According to Zahedi [1993], expert systems and Artificial Neural Networks offer qualitative methods for business and economic systems that traditional quantitative tools in statistics and econometrics cannot quantify due to the complexity in translating the systems into precise mathematical functions.

Medsker et al.[1996] listed the following financial analysis task of which prototype neural network-based decisions aids have been built:

- Credit authorization screening
- Mortgage risk assessment
- Project management and bidding strategy
- Financial and economic forecasting
- Risk rating of exchange-traded, fixed income investments.
- Detection of regularities in security price movements
- Prediction of default and bankruptcy

Hsieh [1993] stated the following potential corporate finance applications can be significantly improved with the adaptation to ANN technology:

- Financial Simulation
- Predicting Investor’s Behavior
- Evaluation
- Credit Approval
- Security and/or Asset Portfolio Management
- Pricing Initial Public Offerings
- Determining Optimal Capital Structure

Trippi and Turban [1996] noted in the preface of their book, that financial organizations are now second only to the US Department of Defense in the sponsorship of research in neural network applications.

1.4 Artificial Neural Networks

Artificial Neural Network (ANN) models were inspired by the biological sciences which study how the neuroanatomy of living animals have developed in solving problems. According to Nelson and Illingworth [1990], ANNs are also called:

- Parallel distributed processing models
- Connectivist/connectionism models
- Adaptive systems
- Self-organizing systems
- Neurocomputing
- Neuromorphic systems

ANNs consist of many interconnected processors known as neurons[1] that perform summing function. Information is stored in the weights on the connections. More detailed discussion on the technical aspects of ANNs is given in Chapter 2 and 3.

An ANN mimics the human brain’s biological neural network. The biological neural network is the mechanism through which a living organism’s nervous system functions, enabling complex tasks to be performed instinctively. The central processing unit of that nervous system is known as a "neuron". The human brain has around 10 to 100 billion neurons, each connected to many others by "synapses". The human brain has around 100 trillion synapses. These connections control the human body and its thought processes. In short, they attempt to replicate the learning processes of the human brain. The first ANN theories were expounded by researchers attempting to explain human behavior and the thinking process by modeling the human brain. To this day, many of the prominent researchers in the ANN field consist of researchers with background in psychology.

The four distinct areas of research in ANNs are:

[1] At the time of writing, there is still no standard terminology in the Connectionist field. The neuron has also been called the following in the Connectionist literature: processing elements, neurodes, processors, units, etc.

- Using ANNs to model the biological networks in order to gain understanding of the human brain and its functions. This area is of particular interest to psychologists and researchers in neuroanatomy;
- Using ANNs as an educational tool in order to gain understanding on how to solve complex tasks that traditional AI methodologies and computer algorithms have had difficulty in solving. Researchers in this area include computer scientists, engineers, etc., who are mainly interested in constructing better computer algorithms by studying the problem-solving process of an ANN;
- Using ANNs to solve real world-types of problems in various commercial applications. Many researchers in this area have backgrounds in areas other than those related to ANN. The attraction of using an ANN is the simplicity in using it as a tool and the reported ANN-based commercial application successes. There are many ANN software packages that are user-friendly enough for new users to start using without requiring them to have an in depth knowledge of the ANN algorithms. This is unlike conventional computer techniques which require a user to thoroughly understand the algorithm before writing program to apply it. In the case of ANNs, all a user needs to know is how to present the problem at hand in a form that an ANN can understand; and
- Improving ANN algorithms. Researchers in this field are interested in constructing better ANN algorithms that can 'learn' or model more efficiently, i.e. quicker training times and/or more accurate results.

Research efforts on ANNs are being conducted on a global basis. Nelson and Illingworth [1991] state that Jasper Lupo, the deputy director of the Tactical Technology Office of the Defense Advanced Research Projects Agency (DARPA), called the neural network technology "more important than the atom bomb" [Johnson and Schwartz 1988]. According to Nelson and Illingworth, DARPA originally earmarked US$390 million for an eight-year neural network program but even when the original funding was reduced to US$33 million over 17 months, there were still many applications for the research grants. More recently, Turban and Trippin [1996], state that following the five-year research program, the Department of Defense (D.O.D) is planning to spend an additional US$15 million in neural network research over the period 1995-2000. They further claim that the Japanese have embarked on a 10-year, US$20 million program to further develop neural network technology, mainly in the commercial arena.

Japan's main ANNs research is sponsored by its government under its post-fifth generation computer program called "The Human Frontiers". However, Japanese corporations are already developing products based on ANNs. Examples of Japanese corporations involvement with ANN technology are:

- Sharp Corporation’s optical character reading of printed Japanese program [Shandle 1993],
- Nippon Steel’s casting breakthrough prediction program [Shandle 1993],
- Hitachi’s ANN hardware system design [Shandle 1993],
- Ricoh’s experimental neurocomputer that runs without software and acquires all computing capabilities through learning [Dambrot 1992],
- Fujitsu’s ANNs-based mobile robot controller, and
- NEC Corporation’s neurocomputer [Nelson and Illingworth 1991], etc.

Europe’s ANNs research effort is called ESPIRIT II and is a five year project involving eight countries and several hundred worker-years of effort [Mehta 1988]. This has been supplemented by a new program announced by ESPIRIT in early 1989, known as the Application of Neural Networks for the Industry (ANNIE) [Newquist III, 1989]. Nelson and Illingworth [1991] states the following about ANNs research effort in individual European countries:

- Germany has a US$250 million 5 year program [Johnson 1989b];
- France, probably has the most active development with six neural-based microchip projects in Paris alone [Johnson 1988];
- Netherlands research has moved from independent research to government sponsored and coordinated research; and
- United Kingdom has a US$470 million project.

The UK Advisory Council for Science and Technology forecasted the market for neural network products in 1997 at US$1 billion which resulted in the UK Department of Trade and Industry (DTI) announcement of a Technology Transfer program that will invest 5.7 million pounds over the next three years to raise awareness of the benefits of neural networks to 6,000 UK companies [Milton 1993].

1.5 Applications of ANNs

Widrow, Rumelhart and Lehr [1993] argue that most ANN applications fall into the following three categories:

- Pattern classification,
- Prediction and financial analysis, and
- Control and Optimization.

In practice, their categorization is ambiguous since many financial and predictive applications involve pattern classification. A preferred classification that separates applications by method is the following:

- Classification
- Time Series and
- Optimization.

Classification problems involve either binary decisions or multiple-class identification in which observations are separated into categories according to specified characteristics. They typically use cross sectional data. Solving these problems entails 'learning' patterns in a data set and constructing a model that can recognize these patterns. Commercial artificial neural network applications of this nature include:

- Credit card fraud detection reportedly being used by Eurocard Nederland, Mellon Bank, First USA Bank, etc. [Bylinsky 1993];
- Optical character recognition (OCR) utilized by fax software such as Calera Recognition System's FaxGrabber and Caere Corporation's Anyfax OCR engine that is licensed to other products such as the popular WinFax Pro and FaxMaster [Widrow et al.1993];
- Cursive handwriting recognition being used by Lexicus[2] Corporation's Longhand program that runs on existing notepads such as NEC Versapad, Toshiba Dynapad etc. [Bylinsky 1993], and ;
- Cervical (Papanicolaou or 'Pap') smear screening system called Papnet[3] was developed by Neuromedical Systems Inc. and is currently being used by the US Food and Drug Administration to help cytotechnologists spot cancerous cells [Schwartz 1995, Dybowski et al.1995, Mango 1994, Boon and Kok 1995, Boon and Kok 1993, Rosenthal et al.1993];
- Petroleum exploration being used by Texaco and Arco to determine locations of underground oil and gas deposits [Widrow et al.1993]; and
- Detection of bombs in suitcases using a neural network approach called Thermal Neutron Analysis (TNA), or more commonly, SNOOPE, developed by Science Applications International Corporation (SAIC) [Nelson and Illingworth 1991, Johnson 1989, Doherty 1989 and Schwartz 1989].

In *time-series problems*, the ANN is required to build a forecasting model from the historical data set to predict future data points. Consequently, they require relatively sophisticated ANN techniques since the sequence of the input data in this type of problem is important in determining the relationship of one pattern of data to the next. This is known as the temporal effect, and more advance techniques such as finite impulse response (FIR) types of ANN and recurrent ANNs are being developed and explored to deal specifically with this type of problem.

Real world examples of time series problems using ANNs include:

[2] Motorola bought Lexicus in 1993 for an estimated US$7 million and the focus of Lexicus is now on developing Chinese writing recognition [Hitheesing 1996].

[3] The company has since listed in the US stock exchange (NASDAQ:PPNT) under the trading name of PAPNET of Ohio. The PAPNET diagnosis program has recently been made available in Australia.

- Foreign exchange trading systems: Citibank London [Penrose 1993, Economist 1992, Colin 1991, Colin 1992], HongKong Bank of Australia [Blue 1993];
- Portfolio selection and management: LBS Capital Management[4] (US$300m) [Bylinsky 1993] (US$600m) [Elgin 1994], Deere & Co. pension fund (US$100m) [Bylinsky 1993] (US$150m) [Elgin 1994], and Fidelity Disciplined Equity Fund [McGugan 1994];
- Forecasting weather patterns [Takita 1995];
- Speech recognition network being marketed by Asahi Chemical [Nelson and Illingworth 1991];
- Predicting/confirming myocardial infarction, a heart attack, from the output waves of an electrocardiogram (ECG) [Baxt 1995, Edenbrandt et al.1993, Hu et al.1993, Bortolan and Willems 1993, Devine et al., Baxt and Skora 1996]. Baxt and Skora reported in their study that the physicians had a diagnostic sensitivity and specificity for myocardial infarction of 73.3 and 81.1% respectively, while the artificial neural network had a diagnostic sensitivity and specificity of 96.0% and 96.0% respectively; and
- Identifying dementia from analysis of electrode-electroencephalogram (EEG) patterns [Baxt 1995, Kloppel 1994, Anderer et al.1994, Jando et al.1993, Bankman et al.1992]. Anderer et al. reported that the artificial neural network did better than both Z statistic and discriminant analysis [Baxt 1995].

Optimization problems involve finding solution for a set of very difficult problems known as Non-Polynomial (NP)-complete problems, Examples of problems of this type include the traveling salesman problem, job-scheduling in manufacturing and efficient routing problems involving vehicles or telecommunication. The ANNs used to solve such problems are conceptually different from the previous two categories (classification and time-series), in that they require unsupervised networks, whereby the ANN is not provided with any prior solutions and thus has to ‘learn’ by itself without the benefit of known patterns. Statistical methods that are equivalent to these type of ANNs fall into the clustering algorithms[5] category.

[4] LBS Capital Management Inc., is a Clearwater, Florida, firm that uses Artificial Neural Networks and Artificial Intelligence to invest US$600 million, half of which are pension assets. It has reported no loss year in stocks and bonds since the strategy was launched in 1986 and its mid-capped returns have ranged from 14.53% in 1993 to 95.60% in 1991, compared to the S & P 400 (sic?), which returned 13.95% and 50.10% respectively. [Elgin 1994].

[5] Cluster analysis basic objective is to discover the natural groupings of items (or variables) and clustering algorithms are used to search for good but not necessarily the best, groupings. They are widely used in understanding the complex nature of multivariate relationships (Johnson and Wichern 1988).

1.6 References

1. “Tilting at Chaos”, *The Economist*, p. 70, August 15, 1992.
2. Anderer P, et al., “Discrimination between demented patients and normals based on topographic EEG slow wave activity: comparison between z statistics, discriminant analysis and artificial neural network classifiers”, *Electroencephalogr Clin Neuropsychol*, No. 91 (2), pp. 108-17, 1994.
3. Bankman IN, et al., “Feature-based detection of the K-complex wave in the human electroencephalogram using neural networks”, *IEEE Trans Biomed Eng,*; No. 39, pp.1305-10, 1992.
4. Baxt, W.G. and Skora, J., “Prospective validation of artificial neural network trained to identify acute myocardial infarction”, *The Lancet*, v347 n8993, p12(4), Jan 6, 1996.
5. Baxt, W.G., “Application of Artificial Neural Networks to Clinical Medicine”, *The Lancet*, v346 n8983, p1135(4), Oct. 28, 1995.
6. Blue, T., “Computers Trade Places in Tomorrow’s World”, *The Australian*, August 21, 1993.
7. Boon ME, Kok LP, Beck S., “Histological validation of neural-network assisted cervical screening: comparison with the conventional approach”, *Cell Vision*, vol. 2, pp. 23-27, 1995.
8. Boon ME, Kok LP., “Neural network processing can provide means to catch errors that slip through human screening of smears”, *Diag Cytopathol*, No. 9, pp. 411-416. 1993.
9. Bortolan G, Willems JL., “Diagnostic ECG classification based on neural networks” *Journal of Electrocardiology*, No. 26, pp. 75-79, 1993.
10. Bylinsky, G., “Computers That Learn by Doing”, *Fortune*, pp. 96-102, September 6, 1993.
11. Colin, A, “Exchange Rate Forecasting at Citibank London*”, Proceedings, Neural Computing 1991*, London, 1991.
12. Colin, A. M., “Neural Networks and Genetic Algorithms for Exchange Rate Forecasting”, Proceedings of International Joint Conference on Neural Networks, Beijing, China, November 1-5, 1992, 1992.
13. Devine B, Macfarlane PW, “Detection of electrocardiographic `left ventricular strain’ using neural nets”, *Med Biol Eng Comput*; No. 31, pp. 343-48, 1993.
14. Doherty, R., “FAA Adds 40 Sniffers”, *Electronic Engineering Times*, issue 554, September 4, 1989.
15. Dybowski, R. and Gant, V., “Artificial neural networks in pathology and medical laboratories”, *The Lancet*, v346 n8984, p1203(5), Nov. 4, 1995.
16. Edenbrandt L, Devine B, Macfarlane PW., “Neural networks for classification of ECG ST-T segments”, *Journal of Electrocardiology*; No. 25, pp. 167-73, 1992.

17. Edenbrandt L, Heden B, Pahlm O, “Neural networks for analysis of ECG complexes” *Journal of Electrocardiology*, No. 26, p.74, 1993.

18. Elgin, P. E., “Pioneers Try ‘Neural Networks’ to Pick Stocks” (Trends: Investments & Benefits), *Computer Cashflow Magazine*, v15 n8, p5(2), July 1994.

19. Gallant, S. I., *Neural Network Learning and Expert Systems*, MIT Press, Cambridge, USA, pp. 4-6, 1993.

20. Hitheesing, N., “The Mother of Development”, *Forbes*, v157 n2, p88(2), Jan 22, 1996.

21. Hortnik, K., Stinchcombe, M., White, H., “Multilayer Feedforward Networks are Universal Approximators”, *Neural Networks*, Vol. 2, pp. 359-366, 1989.

22. Hsieh, C., “Some Potential Applications of Artificial Neural Systems in Financial Management”, *Journal of Systems Management*, v.44 n4, p12(4), April 1993.

23. Hu YH, et al., “Applications of artificial neural networks for ECG signal detection and classification”. *Journal of Electrocardiology*, No. 26, pp.66-73, 1993.

24. Huffman. J., “Natural computing is in your future”, *Appliance Manufacturer*, v42 n2, p10(1), Feb. 1994.

25. Jando G, et al., “Pattern recognition of the electroencephalogram by artificial neural networks”, *Electoencephalogr Clin Neurophysiol*, No. 86, pp. 100-09, 1993.

26. Johnson, R. A., and Wichern, D. D., “Clustering”, *Applied Multivariate Statistical Analysis Second Edition*, Prentice-Hall International, pp. 543-589, 1988.

27. Johnson, R. C. and T. J. Schwartz, “DARPA Backs Neural Nets”, *Electronic Engineering Times*, issue 498, p1, 96, August 8, 1988.

28. Johnson, R. C., “DARPA Neural Awards Stress Practical Use”, *Electronic Engineering Times*, issue 558, p. 22, October 2, 1989b.

29. Johnson, R. C., “French Research: Vive le Neuron” and “nEuro ‘88 Abuzz with nEuropean Activity”, *Electronic Engineering Times*, issue 492, p. 57, June 27, 1988.

30. Johnson, R. C., “Neural Nose to Sniff Out Explosives at JFK Airport”, *Electronic Engineering Times*, issue 536, May 1, 1989a.

31. Kloppel B., “Application of neural networks for EEG analysis. Consideration and first results”, *Neuropsychobiology*, No. 29, pp. 39-46, 1994.

32. Lawrence, J., *Introduction to Neural Networks: Design, Theory, and Applications 6th edition*, edited by Luedeking, S., ISBN 1-883157-00-5, California Scientific Software, California, USA, July 1994.

33. Mango LJ., “Computer-assisted cervical cancer screening using neural networks”, *Cancer Letter*, vol. 77, pp. 155-62, 1994.

34. McGugan, I., “The Machines from MENSA”, *Canadian Business*, v.67 n3., p. 61(3), March 1994.

35. Medsker, L., Turban, E. and R. Trippi, “Neural Network Fundamentals for Financial Analysts”, *Neural Networks in Finance and Investing edited by Trippi and Turban*, Irwin, USA, Chapter. 1, pp. 329-365, ISBN 1-55738-919-6, 1996.

36. Mehta, A., “Nations Unite for Electronic Brain”, *Computer Weekly*, issue 1148, January 11, 1988.

37. Milton, R., “Neural Niches”, *Computing*, p. 30(2), Sept. 23, 1993.

38. Nelson, M. M. & Illingworth, W. T., *A Practical Guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., USA, 1991.

39. Newquist III, H. P., “Parlez-Vous Intelligence Artificielle?”, *AI Expert*, vol. 4, no. 9, p. 60, September 1989.

40. Pal, S. K. and Srimani, P. K., “Neurocomputing: Motivation, Models, and Hybridization”, *Computer*, ISSN 0018-9162, Vol. 29 No. 3, IEEE Computer Society, NY, USA, pp. 24-28, March 1996.

41. Penrose, P., “Star Dealer who works in the dark”, *The London Times*, p. 28, Feb. 26, 1993.

42. Rich, E. & Knight, K., *Artificial Intelligence, Second Edition*, McGraw Hill, pp. 4-6, 1991.

43. Rosenthal DL, Mango LJ, Acosta DA and Peters RD., “"Negative" pap smears preceding carcinoma of the cervix: rescreening with the PAPNET system.”, *American Journal of Clinical Pathology*, No. 100, pp. 331, 1993.

44. Schwartz, T. J., “IJCN ‘89”, *IEEE Expert*, vol. 4 no. 3, pp. 77-78, Fall 1989.

45. Schwartz, T., “Applications on Parade”, *Electronic Design*, v43 n16, p68(1), August 7, 1995.

46. Shandle, J., “Neural Networks are Ready for Prime Time”, *Electronic Design*, v.41 n.4, p51(6), Feb. 18, 1993.

47. Takita, H., “Pennies from Heaven: selling accurate weather predictions*”, Today (Japan),* v63 n7, p14(3), July 1995.

48. Trippi and Turban, *Neural Networks in Finance and Investing 2n. Edition*, Irwin, USA, ISBN 1-55738-919-6, 1996.

49. Widrow, B., Rumelhart, D. E., Lehr, M. A., *Neural Networks: Applications in Industry, Business and Science*, Journal A, vol. 35, No. 2, pp. 17-27, July 1994.

50. Winston, P. , *Artificial Intelligence, Third Edition*, Addison-Wesley, 1992.

51. Zahedi, F., *Intelligent Systems for Business: Expert Systems with Neural Networks*, Wadsworth Publishing Company, Belmont, USA, pp. 10-11, 1993.

Chapter 2: An Artificial Neural Networks' Primer

"There is no expedient to which a man will not go to avoid the real labor of thinking"

Thomas A. Edison (1847-1931), Posted on signs in the Edison laboratories

2. An Artificial Neural Networks' Primer

2.1 Chronicle of Artificial Neural Networks Development

According to Nelson and Illingworth [1991], the earliest attempt to understand the human brain goes back centuries. They cite information given by Fischler and Firschein [1987] who refer to the work of Hippocrates, and the less familiar Edward Smith Papyrus; a treatise written around 3000 BC that described the location of certain sensory and motor control areas in the brain. For the most part of history, since the days of ancient Greek philosophers such as Plato and Aristotle, the study of the brain has been limited to the philosophical question of whether the mind and the body are one. As Rich and Knight state in the beginning of their book, *Artificial Intelligence*, "Philosophy has always been the study of those branches of knowledge that were so poorly understood that they had not yet become separate disciplines in their own right". This was certainly true with modern brain theory and the eventual development of Artificial Neural Networks (ANNs). Technology to enable the study of the workings of the brain was not available until the late nineteenth century. Since then, ANNs have had a very rocky climb to fame. There are four distinct periods of their development to their current status. Eberhart and Dobbins [1990] classified them in the following order:

• 1890-1969	The Age of Camelot
• 1969-1982	The Dark Age (Depression Age)
• 1982-1986	The Renaissance
• 1986-Current	The Age of Neoconnectionism

The first period began in the late nineteenth century with the advent of modern science and the pursuit for better understanding of the workings of the brain. As technology improved, psychologists and biologist were able to start hypothesizing on *how* rather than *why* the human brain functions. Most ANNs literature places the beginning of the ANNs and modern brain theory era with the publication of a text by William James[6] entitled "Psychology (Briefer Course)" [James 1890]. The text contained many insights into brain activities and was the precursor of many of the current theories.

It was some fifty years later before the next major breakthrough came in 1943, when McCulloch and Pitts presented their first model of a biological neuron [McCulloch and Pitts 1943]. They developed theorems related to models of neuronal systems based on the knowledge of the biological structure at the time. Their models could solve any finite logical expressions, and, since James, they were the first authors who proposed a massively parallel neural model. However, their models could not "learn" as they used only fixed weights. Donald Hebb [1949], an eminent psychologist, added to this knowledge with his hypothesis of how the neurons communicated and stored knowledge in the brain structure. This hypothesis became known as *Hebbian Learning Rule* and enabled the eventual development of learning rules for the McCulloch-Pitts neural models.

This period peaked in 1958 when Frank Rosenblatt published his landmark paper [Rosenblatt 1958] that defined a neural network structure called the *perceptron*. Rosenblatt was inspired by the way the eye functioned and built his *perceptron* model based on it. He

[6] According to Eberhart and Dobbins [1990], James was considered by many to be the greatest American.

incorporated learning based on the Hebbian Learning Rule into the McCulloch-Pitts neural model. The tasks that he used the perceptron to solve were identifying simple pattern recognition problems such as differentiating sets of geometric patterns and alphabets. The Artificial Intelligence community was excited with the initial success of the perceptron and expectations were generally very high with the perception[7] of the perceptron being the panacea for all the known computer problems of that time. Bernard Widrow and Marcian Hoff contributed to this optimism when they published a paper [Widrow and Hoff 1960] on ANNs from the engineering perspective and introduced a single neuron model called ADALINE that became the first ANN to be used in a commercial application. It has been used since then as an adaptive filter for telecommunication to cancel out echoes on phone lines. The ADALINE used a learning algorithm that became known as the delta rule[8]. It involves using an error reduction method known as gradient descent or steepest descent.

However, in 1969, Marvin Minsky and Samuel Papert, two well renown researchers in the Artificial Intelligence field, published a book entitled 'Perceptron' [Minsky and Papert 1969], criticizing the perceptron model, concluding that it (and ANNs as a whole) could not solve any real problems of interest. They proved that the perceptron model, being a simple linear model with no hidden layers, could only solve a class of problems known as *linearly separable* problems. One example of a non-linearly separable problem that they proved the perceptron model was incapable of solving is the now infamous exclusive-or[9] and its generalization, the parity detection problem. Rosenblatt did consider multilayer perceptron models but at that time, a learning algorithm to train such models was not available.

This critique, coupled with the death of Rosenblatt in a boat accident in 1971 [Masters 1993], cast doubt on the minds of research sponsors and researchers alike on the viability of developing practical applications from Artificial Neural Networks. Funds for ANNs research dried up, and many researchers went on to pursue other more conventional Artificial Intelligence technology. In the prologue of the recent reprint of 'Perceptron', Minsky and Papert [1988, pp. vii-xv][10] justified their criticism of the perceptron model and pessimism of the ANNs field at that time by claiming that the redirection of research was "no arbitrary diversion but a necessary interlude". They felt that more time was needed to develop adequate ideas about the representation of knowledge before the field could progress further. They further claimed that the result of this diversion of resources brought about many new and powerful ideas in symbolic AI such as relational databases, frames and production systems which in turned, benefited many other research areas in psychology, brain science, and applied expert systems. They hailed the 1970s as a golden age of a new field of research into the representation of knowledge. Ironically, this signaled the end of the second period of ANN development and the beginning of the Dark Ages for ANNs research.

[7] Pardon the pun!

[8] This algorithm is also known as the Widrow-Hoff or Least Mean Squares method. An extension of this algorithm is used today in the back-propagation algorithm.

[9] The exclusive-or (XOR) problem and linear separability issue is discussed in more detail in Chapter 3.

[10] Interestingly, the reprint of the 'Perceptron' was dedicated by the authors to the memory of Frank Rosenblatt.

However, pockets of researchers such as David Rumelhart at UC San Diego (now at Stanford University), Stephen Grossberg at Boston University, Teuvo Kohonen in Finland and Kunihiko Fukushima in Japan, persisted with their research into Artificial Neural Networks. Their work came into fruition in the early 1980s, an era that many deemed as the Renaissance period of ANNs. John Hopfield of the California Institute of Technology, a prominent scientist, presented a paper [Hopfield 1984] at the Academy of Science on applying ANNs to the infamous 'traveling salesman problem'. It was his ability to describe his work from the point of a scientist coupled with his credibility, that heralded the gradual re-acceptance of ANNs. Interest grew from researchers from a multitude of fields, ranging from biologists to bankers, and engineers to psychologists. This era culminated with the publication of the first of the three volume, now famous reference text on ANNs, 'Parallel Data Processing' by Rumelhart et al. [1986b]. The authors had proposed the 'back-propagation' learning algorithm in an earlier publication [1986a] that was popularized by the text. The back-propagation algorithm overcame some of the pitfalls of the perceptron model that were pointed out by Minsky and Papert by allowing multi-layer perceptron models to learn. According to Ripley [1993], the back-propagation algorithm was originally discovered by Bryson and Ho [1969] and Werbos [1974] but did not gain prominence until it was rediscovered and popularized by Rumelhart et al. According to Eberhart and Dobbins [1990], it is hard to overstate the effect the Parallel Data Processing (PDP) books had on neural network research and development. They attribute the success of the books in one sentence: "The books presented everything practical there was to know about neural networks in 1986 in an understandable, usable and interesting way; in fact, 1986 seemed to mark the point at which a 'critical mass' of neural network information became available".

The current era begins where the PDP books left off and has been called the Age of Neoconnectionism by Cowan and Sharp [1988]. In this era, there has being a growing number of commercial ANN applications as well as continued prolific research interest from a wide field of disciplines in ANNs, as evident by the number of publications and conferences on ANNs. Sejnowski and Rosenburg's [1987] success on their NETtalk ANN-based speech generation program that teaches itself to read out aloud and subsequent work by Martin [1990] on an ANN-based handwriting recognition to recognize zip codes for the US Post Office, spurred on the prominence of ANNs as a potential application tool for handling difficult tasks. The significant improvements in computer technology as well as the rapid reduction in the cost of high powered computers have resulted in making the development of ANNs applications a universally attractive and affordable option

2.2 Biological Background

ANNs were inspired by the biological sciences, particularly the neurological sciences, as discussed in the section on the chronicle of their development. However, ANNs resemblance to their biological counterparts are limited to some borrowed concepts from the biological networks, mainly for their architecture. They are still far from resembling the workings of the simplest biological networks, due to the enormous complexity of the biological networks.

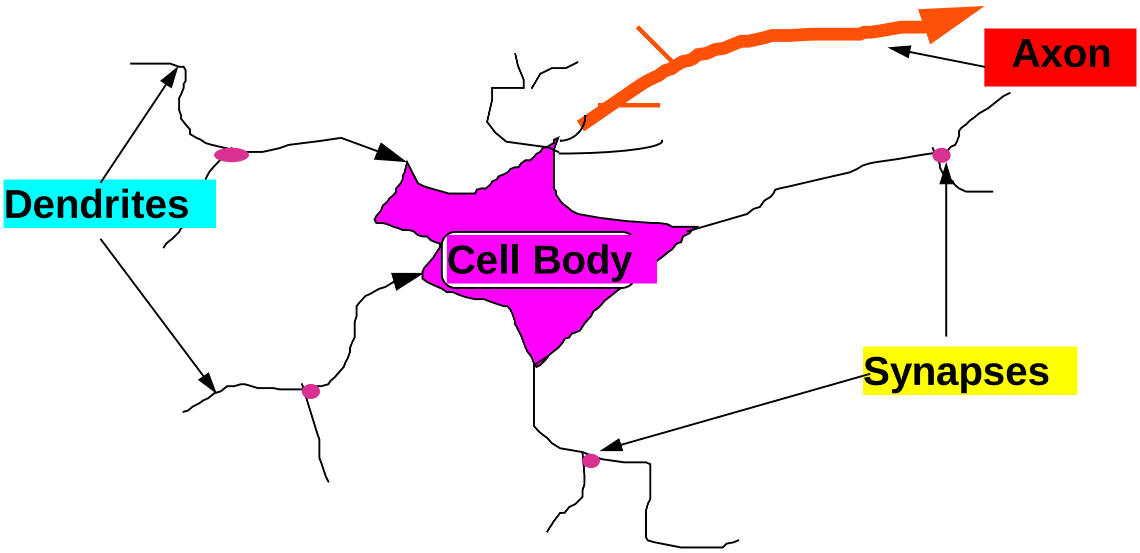
The cells found in the human brain and nervous system are known as *neurons*. Information or signals are transmitted out unidirectionally through connections between neurons known as *axons*. Information is received by a neuron through its *dendrites*. The human brain consist of around 100 billion neurons and over 10^{14} synapses. Neurons communicate with each other through *synapses* which are gaps or junctions between the connections. The transmitting side of the synapses release neurotransmitters which are paired to the neuroreceptors on the receiving side of the synapses. Learning is usually done by adjusting existing synapses, though some learning and memory functions are carried out by creating new synapses. In the human brain, neurons are organized in clusters and only several thousands or hundred of thousands participate in any given task. Figure 2-1 shows a sample neurobiological structure of a neuron and its connections.

The axon of a neuron is the output path of a neuron that branches out through axon collaterals which in turn connect to the dendrites or input paths of neurons through a junction or a gap known as the synapse. It is through these synapses that most learning is carried out by either exciting or inhibiting their associated neuron activity. However, not all neurons are adaptive or plastic. Synapses contain neurotransmitters that are released according to the incoming signals. The synapses excite or inhibit their associated neuron activity depending on the neurotransmitters released. A biological neuron will add up all the activating signals and subtract all the inhibiting signals from all of its synapses. It will only send out a signal to its axon if the difference is higher than its threshold of activation.

The processing in the biological brain is highly parallel and is also very fault tolerant. The fault tolerance characteristic is a result of the neural pathways being very redundant and information being spread throughout synapses in the brain. This wide distribution of information also allows the neural pathways to deal well with noisy data.

A biological neuron is so complex that current super computers cannot even model a single neuron. Researchers have therefore simplified neuron models in designing ANNs.

Figure 2-1
A typical biological neuron



2.3 Comparison to Conventional Computational Techniques

ANNs differ from conventional computational techniques in that the system builder of an ANN is not required to write programs, hence, there is no necessity for system builder to know a priori the necessary rules or models that are required to perform the desired task. Instead, a system builder trains an ANN to 'learn' from previous samples of data in much the same way that a teacher would teach a child to recognize shapes, colors, alphabets, etc. The ANN builds an internal representation of the data and by doing so 'creates' an internal model that can be used with new data that it has not seen before.

Existing computers process information in a serial fashion while ANNs process information in parallel. This is why even though a human brain neuron transfers information in the milliseconds (10^{-3}) range while current computer logic gates operate in the nanosecond (10^{-9}) range, about a million times faster, a human brain can still process a pattern recognition task much faster and more efficiently than the fastest currently available computer. The brain has approximately 10^{11} neurons and each of these neurons acts as a simple processor that processes data concurrently; i.e. in parallel.

Tasks such as walking and cycling seem to be easy to humans once they have learned them and certainly not much thought is needed to perform these tasks once they are learnt. However, writing a conventional computer program to allow a robot to perform these tasks is very complex. This is due to the enormous quantity of data that must be processed in order to cope with the constantly changing surrounding environment. These changes require frequent computation and dynamic real time processing. A human child learns these tasks by trial and error. For example, in learning to walk, a child gets up, staggers and falls, and keeps repeating the actions over and over until he/she has learned to walk. The child effectively 'models' the walking task in the human brain through constant adjustments of the synaptic strengths or weights until a stable model is achieved.

Humans (and neural networks) are very good at pattern recognition tasks. This explains why one can usually guess a tune from just hearing a few bars of it or how a letter carrier can read a wide variety of handwritten address without much difficulty. In fact, people tend to always associate their senses with their experiences. For example, in the 'Wheel of Fortune' game show, the contestants and viewers are usually able to guess a phrase correctly from only a few visible letters in a phrase. The eyes tend to look at the whole phrase, leaving the brains to fill in the missing letters in the phrase and associate it with a known phrase. Now, if we were to process this information sequentially like a serial computer; i.e., look at one visible character at a time; and try to work out the phrase, it would be very difficult. This suggests that pattern recognition tasks are easier to perform by looking at a whole pattern (which is more akin to neural network's parallel processing) rather than in sequential manner (as in a conventional computer's serial processing).

In contrast, tasks that involve many numerical computations are still done faster by computers because most numerical computations can be reduced to binary representations that allow fast serial processing. Most of today's ANN programs are being simulated by serial computers, which is why speed is still a major issue for ANNs, specifically the training time. There are a growing number of ANN hardware[11] available in the market today including personal computer-based ones like the Intel's Ni1000 and the Electronically Trainable Artificial Neural Network (ETANN), the IBM's ZISC/ISA Accelerator for PC and the Brainmaker Professional CNAPS™ Accelerator System. These ANN hardware process information in parallel, but the costs and the learning curves

[11] See Lindsey and Lindblad [1994, 1995] and Lindsey et. al.'s [1996] for a comprehensive listing of commercial ANN hardware.

required to use them are still quite prohibitive. Most researchers are of the view that in the near future, a special ANN chip will be sitting next to the more familiar CPU chip in personal computers, performing pattern recognition tasks such as voice and optical character recognition.

2.4 ANN Strengths and Weaknesses

ANNs are easy to construct and deal very well with large amounts of noisy data. They are especially suited to solving nonlinear problems. They work well for problems where domain experts may be unavailable or where there are no known rules. ANNs are also adaptive in nature. This makes them particularly useful in fields such as finance where the environment is potentially volatile and dynamic.

They are also very tolerant of noisy and incomplete data sets. Their robustness in storing and processing data, earned them some applications in space exploration by NASA, where fault tolerant types of equipment are required. This flexibility derives from the fact that information is duplicated many times over in the many complex and intricate network connections in ANNs, just like in the human brain. This feature of ANNs is, in contrast to the serial computer[12] where if one piece of information is lost, the entire information set may be corrupted.

The training process of an ANN itself is relatively simple. The pre-processing of the data, however, including the data selection and representation to the ANN and the post-processing of the outputs (required for interpretation of the output and performance evaluation) require a significant amount of work[13]. However, constructing a problem with ANNs is still perceived to be easier than modeling with conventional statistical methods. There are many statisticians who argue that ANNs are nothing more than special cases of statistical models, and thus the rigid restrictions that apply to those models must also be applied to ANNs as well. However, there are probably more successful novel applications using ANNs than conventional statistical tools. The prolific number of ANNs applications in a relatively short time could be explained by the universal appeal of the relatively easy methodology in setting up an ANN to solve a problem. The restrictions imposed by many equivalent statistical models is probably less appealing to many researchers without a strong statistical background. ANN software packages are also relatively easier to use than the typical statistical packages. Researchers can successfully use ANNs software packages without requiring full understanding of the learning algorithms. This makes them more accessible to a wider variety of researchers. ANN researchers are more likely to learn from experience rather than be guided by statistical rules in constructing a model and thus they may be implicitly aware of the statistical restrictions of their ANN models.

[12] Serial computers are also called Von Neumann computers in computer literature.

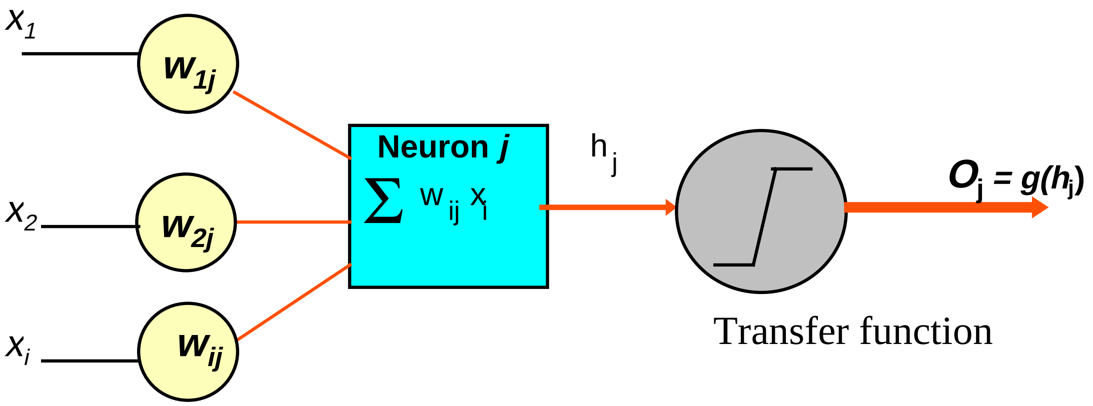
[13] The old adage of garbage in, garbage out holds especially true for ANN modeling. A well-known case in which an ANN learned the incorrect model involved the identification of a person's sex from a picture of his/her face. The ANN application was trained to identify a person as either male or female by being shown various pictures of different persons' faces. At first, researchers thought that the ANN had learnt to differentiate the face of a male from a female by identifying the visual features of a person's face. However it was later discovered that the pictures used as input data showed all the male persons' heads nearer to the edge of the top end of the pictures, presumably due to a bias of taller males in the data than females. The ANN model had therefore learned to differentiate the sex of a person by the distance his/her head is from the top edge of a picture rather than by identifying his/her visual facial features.

The major weakness of ANNs is their lack of explanation for the models that they create. Research is currently being conducted to unravel the complex network structures that are created by ANN. Even though ANNs are easy to construct, finding a good ANN structure, as well as the pre-processing and post processing of the data, is a very time consuming processes. Ripley [1993] states 'the design and learning for feed-forward networks are hard'. He further quoted research by Judd [1990] and Blum and River [1992] that showed this problem to be NP-complete[14].

2.5 Basic Structure of an ANN

The basic structure of an ANN consists of artificial *neurons*[15] (similar to biological neurons in the human brain) that are grouped into *layers*[16]. The most common ANN structure consists of an input layer, one or more hidden layers and an output layer. A modified simple model of an artificial neuron is shown in Figure 2-2.

Figure 2-2
An Artificial Neuron



In the human brain, neurons communicate by sending signals to each other through complex connections. ANNs are based on the same principle in an attempt to simulate the learning process of the human brain by using complex algorithms. Every connection has a weight attached which may have either a positive or a negative value associated with it. Positive weights activate the neuron while negative weights inhibit it. Figure 1 shows a network structure with *inputs* (x_1, x_2, ...x_i) being connected to neuron *j* with weights (w_{1j}, w_{2j},...w_{ij}) on each connection. The neuron sums all the signals it receives, with each signal being multiplied by its associated weights on the connection.

[14] NP (Non-Polynomial)-complete problems as mentioned in Chapter 1, are a set of very difficult problems.

[15] There is no standardization of terminology in the artificial neural network field. However, the Institute of Electrical and Electronic Engineers currently have a committee looking into it. Other terminology that has been used to describe the artificial neuron include processing elements, nodes, neurodes, units, etc.

[16] In some ANN literature the layers are also called *slabs*.

This output (h_j) is then passed through a transfer (activation) function, *g(h),* that is normally non-linear to give the final output O_j. The most commonly used function is the sigmoid (logistic function) because of its easily differentiable properties[17], which is very convenient when the back-propagation algorithm is applied. The whole process is discussed in more detail in chapter 3.

The back-propagation ANN is a feed-forward neural network structure that takes the input to the network and multiplies it by the weights on the connections between neurons or nodes; summing their products before passing it through a threshold function to produce an output. The back-propagation algorithm works by minimizing the error between the output and the target (actual) by propagating the error back into the network. The weights on each of the connections between the neurons are changed according to the size of the initial error. The input data are then fed forward again, producing a new output and error. The process is reiterated until an acceptable minimized error is obtained. Each of the neurons uses a transfer function[18] and is fully connected to nodes on the next layer. Once the error reaches an acceptable value, the training is halted. The resulting model is a function that is an internal representation of the output in terms of the inputs at that point. A more detailed discussion of the back-propagation algorithm is given in chapter 3.

2.6 Constructing the ANN

Setting up an ANN is essentially a six step procedure.

Firstly, the data to be used need to be defined and presented to the ANN as a pattern of input data with the desired outcome or target.

Secondly, the data are categorized to be either in the training set or validation (also called test and out-of-sample) set. The ANN only uses the training set in its learning process in developing the model. The validation set is used to test the model for its predictive ability and when to stop the training of the ANN.

Thirdly, the ANN structure is defined by selecting the number of hidden layers to be constructed and the number of neurons for each hidden layer.

Fourthly, all the ANN parameters are set before starting the training process. The ANN parameters are discussed briefly in the next section and in more detail in chapter 3.

Next, the training process is started. The training process involves the computation of the output from the input data and the weights. The backpropagation algorithm is used to

[17] The sigmoid (logistic) function is defined as $O_{pj} = \dfrac{1}{1+e^{-net_{pj}}}$. In the ANN context, O_{pj} is the output of a neuron j given an input pattern p and net_{pj} is the total input to the ANN. The derivative of the output function to the total input is required to update the weights in the back-propagation algorithm. Thus we have: $\dfrac{\partial O_{pj}}{\partial net_{pj}} = O_{pj}(1-O_{pj})$, a trivial derivation. For a more detailed discussion on the back-propagation algorithm, see Chapter 3.

[18] A sigmoid function like the logistic function is most common transfer function in ANNs. Transfer functions are discussed in more detail in Chapter 3.

'train' the ANN by adjusting its weights to minimize the difference between the current ANN output and the desired output.

Finally, an evaluation process has to be conducted to determine if the ANN has 'learned' to solve the task at hand. This evaluation process may involve periodically halting the training process and testing its performance until an acceptable result is obtained. When an acceptable result is obtained, the ANN is then deemed to have been trained and ready to be used.

As there are no fixed rules in determining the ANN structure or its parameter values, a large number of ANNs may have to be constructed with different structures and parameters before determining an acceptable model. The trial and error process can be tedious and the experience of the ANN user in constructing the networks is invaluable in the search for a good model.

Determining when the training process needs to be halted is of vital importance in obtaining a good model. If an ANN is overtrained, a curve-fitting problem may occur whereby the ANN starts to fit itself to the training set instead of creating a generalized model. This typically results in poor predictions of the test and validation data set. On the other hand, if the ANN is not trained for long enough, it may settle at a local minimum, rather than the global minimum solution. This typically generates a sub-optimal model. By performing periodic testing of the ANN on the test set and recording both the results of the training and test data set results, the number of iterations that produce the best model can be obtained. All that is needed is to reset the ANN and train the network up to that number of iterations.

2.7 A Brief Description of the ANN Parameters

This section gives a brief introductory non-technical description of the ANN parameters. The mathematical descriptions of the parameters and learning process are discussed in more detail in chapter 3.

2.7.1 Learning Rate

The learning rate determines the amount of correction term that is applied to adjust the neuron weights during training. Small values of the learning rate increase learning time but tend to decrease the chance of overshooting the optimal solution. At the same time, they increase the likelihood of becoming stuck at local minima. Large values of the learning rate may train the network faster, but may result in no learning occurring at all. The adaptive learning rate varies according to the amount of error being generated. The larger the error, the smaller the values and vice-versa. Therefore, if the ANN is heading towards the optimal solution it will accelerate. Correspondingly, it will decelerate when it is heading away from the optimal solution.

2.7.2 Momentum

The momentum value determines how much of the previous corrective term should be remembered and carried on in the current training. The larger the momentum value, the more emphasis is placed on the current correction term and the less on previous terms. It serves as a smoothing process that 'brakes' the learning process from heading in an undesirable direction.

2.7.3 Input Noise

Random noise is used to perturb the error surface of the neural net to jolt it out of local minima. It also helps the ANN to generalize and avoid curve fitting.

2.7.4 Training and Testing Tolerances

The training tolerance is the amount of accuracy that the network is required to achieve during its learning stage on the training data set. The testing tolerance is the accuracy that will determine the predictive result of the ANN on the test data set.

2.8 Determining an Evaluation Criteria

It is not always easy to determine proper evaluation criteria in designing an ANN model to solve a particular problem. In designing an ANN to solve a particular problem, special attention needs to be taken in determining the evaluation criteria. This can be done by careful analysis of the problem at hand, the main objective of the whole process and the ANN role in the process.

For example, in designing an ANN to perform the task of designing a trading system for the foreign exchange market, there are many ways to evaluate the ANN model. The most obvious is to determine the forecast accuracy in terms of forecast error. However, in this particular problem, the accuracy of the forecast is not as important as the ability of the ANN model to generate profit in the trading system. Thus the evaluation criteria in this case is the profit made in trading the out of sample data period.

In the task of designing an early warning predictor of credit unions in distress in chapter 4, the evaluation criteria is based on the number of Type I errors committed, i.e., the number of credit unions actually in distress that were predicted to be not in distress. The ANN forecast was a number between zero and one with zero indicating no distress and 1 being in distress. However, in this case, in developing the evaluation criteria, an acceptable cut-off value has to be determined in differentiating distress from non-distress. The obvious choice is to use 0.5 but on further analysis, a value of 0.1 is determined to be a better value for this task.

2.9 References

1. Blum, A. L. and Rivers, R.L., "Training a 3-node Neural Network is NP-complete", *Neural Networks 5*, pp. 117-127, 1992.
2. Bryson, A. E., Ho, Y, -C., *Applied Optimal Control*, Blaisdell, 1969.
3. Cowan, J. D. and Sharp, D. H., "Neural Nets and Artificial Intelligence", *Daedalus*, 117(1), pp. 85-121, 1988.
4. Fischler and Firschein, *Intelligence: The Eye, the Brain, and the Computer*, Reading, MA, Addison-Wesley, p. 23, April 1987.
5. Hebb, D. O., *The Organization of Behavior*, John Wiley, New York, 1949.
6. James, W., *Psychology (Briefer Course)*, Holt, New York, 1890.
7. Judd, J. S., Neural Network Design and Complexity of Learning, MIT Press, USA, 1990.
8. Lindsey, C. S. and. Lindblad, T., "Review of Hardware Neural Networks: A User's Perspective", *Proceedings of ELBA94.*, 1994.
9. Lindsey, C. S. and. Lindblad, T., "Survey of Neural Network Hardware", *Proceedings of SPIE95*, 1995.
10. Lindsey, C. S., Denby, B. and Lindblad, T., June 11, 1996, *Neural Network Hardware*, [Online], Artificial Neural Networks in High Energy Physics, Available: http://www1.cern.ch/NeuralNets/nnwInHepHard.html, [1996, August 30].
11. Masters, T., *Practical Neural Network Recipes in C++*, Academic Press Inc., San Diego, CA., USA, ISBN: 0-12-479040-2, p.6, 1993.
12. McCartor, H., "Back Propagation Implementation on the Adaptive Solutions CNAPS Neurocomputer", Advances in Neural Information Processing Systems 3, USA, 1991.
13. McCulloch, W. S. and Pitts, W., "A Logical Calculus of Ideas Immanent in Nervous Activity:, *Bulletin of Mathematical Biophysics*, pp. 5:115-33, 1943.
14. Minsky, M. and Papert, S. A., *Perceptrons*, MIT Press, Cambridge, MA, USA,1969.
15. Minsky, M. and Papert, S. A., *Perceptrons. Expanded Edition*, MIT Press, Cambridge, MA, USA, ISBN: 0-262-63111-3, 1988.
16. Nelson, M. M. and Illingworth, W. T., *A Practical Guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., ISBN: 0-201-52376-0/0-201-56309-6, USA, 1991.
17. *Neural Computing: NeuralWorks Professional II/Plus and NeuralWorks Explorer*, NeuralWare Inc. Technical Publishing group, Pittsburgh, PA, USA, 1991.
18. Ripley, B. D., "Statistical Aspects of Neural Networks", Networks and Chaos: Statistical and Probabilistic Aspects edited by Barndoff-Nielsen, O. E., Jensen, J.L. and Kendall, W.S., Chapman and Hall, London, United Kingdom, 1993.
19. Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain", *Psychological Review*, 65:pp.386-408, 1958.
20. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Back-Propagating Errors", *Nature*, No. 323: pp.533-536, 1986a.

21. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing: Explorations in the microstructure of Cognition edited by Rumelhart, McClelland and the PDP Research Groups Vol.1*, pp. 216-271, MIT Press, Cambridge Mass., USA, ISBN: 0-262-18120-7, 1986b.

22. Sejnowski, T. J. and Rosenburg, C. R., "Parallel Networks that Learn to Pronounce English Text, *Complex Systems*, No. 1, pp. 145-168, 1987.

23. Shih, Y., *Neuralyst™ User's Guide*, Cheshire Engineering Corporation, USA, p. 21, 1994.

24. Werbos, P., *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. thesis, Harvard University*, 1974.

25. Widrow, B. and Hoff, M. D., "Adaptive Switching Circuits*", 1960 IRE WESCON Convention Record*, Part 4, pp. 96-104, 1960.

Chapter 3: The Technical and Statistical Aspects of Artificial Neural Networks

"The real problem is not whether machines think but whether men do."
B. F. Skinner, Contingencies of Reinforcement, 1969

"There are two kind of statistics, the kind you look up and the kind you make up."
Rex Stout (1886-1975), Death of a Doxy, 1966

3. The Technical and Statistical Aspects of Artificial Neural Networks

3.1 Artificial Neural Network Models

According to Nelson and Illingworth [1990], there are infinitely many ways to organize a neural network although perhaps only two dozen models are in common usage. A neural network organization can be described in terms of its *neurodynamics* and *architecture*. Neurodynamics refer to the properties of an individual artificial neuron that consist of the following:

- combination of input(s);
- production of output(s);
- type of transfer (activation) functions; and
- weighting schemes, i.e. weight initialization and weight learning algorithms.

These properties can also be applied to the whole network on a system basis.

Network architecture (also sometimes referred to as *network topology*) defines the network structure and includes the following basic characteristics:

- types of interconnections among artificial neurons (henceforth referred to as just neurons[19]);
- number of neurons and
- number of layers

3.2 Neurodynamics

3.2.1 Inputs

The input layer of an ANN typically functions as a buffer for the inputs, transferring the data to the next layer. Preprocessing the inputs may be required as ANNs deal only with numeric data. This may involve scaling the input data and converting or encoding the input data to a numerical form that can be used by the ANN. For example, in an ANN real estate price simulator application described in a paper by Haynes and Tan [1993], some qualitative data pertaining to the availability of certain features of a residential property used a binary representation. For example, features like the availability of a swimming pool, a granny flat and a waterfront location, were represented with a binary value of ‘1’, indicating the availability of the feature, or ‘0’ if it was not. Similarly, a character or an image to be presented to an ANN can be converted into binary values of zeroes and ones. For example, the character ‘T’ can be represented as shown in Figure 3-1.

[19] As mentioned earlier, they are also called processing elements, neurodes, nodes, units, etc.

Figure 3-1
The binary representation for the letter 'T'

1111111

000**1**000

000**1**000

000**1**000

3.2.2 Outputs

The output layer of an ANN functions in a similar fashion to the input layer except that it transfers the information from the network to the outside world. Post-processing of the output data is often required to convert the information to a comprehensible and usable form outside the network. The post-processing may be as simple as just a scaling of the outputs ranging to more elaborate processing as in hybrid systems.

For example, in chapter 4 of this book, on the prediction of financial distress in credit unions, the post-processing is relatively simple. It only requires the continuous output values from the ANN to be converted into a binary form of '1' (indicating a credit union in distress) or '0' (indicating a credit union is not in distress). However, in the foreign exchange trading system application in chapter 5, the post-processing of the network output is more complex. The ANN output is the predicted exchange rate but the trading system output requires a trading signal to be generated from the ANN output. Thus, the ANN output has to go through a set of rules to produce the trading signal of either a 'Buy' or 'Sell' or 'Do Nothing'.

3.2.3 Transfer (Activation) Functions

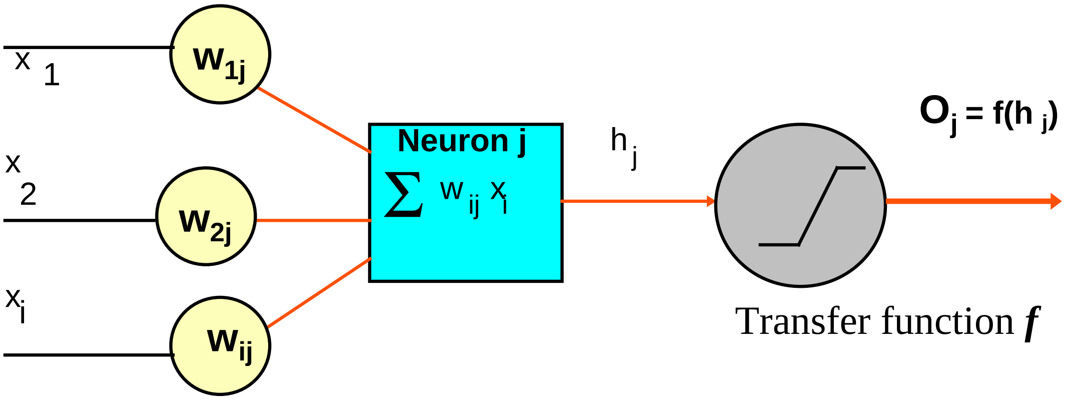
The transfer or activation function is a function that determines the output from a summation of the weighted inputs of a neuron. The transfer functions for neurons in the hidden layer are often nonlinear and they provide the nonlinearities for the network.

For the example in Figure 3-2, the output of neuron j, after the summation of its weighted inputs from neuron 1 to i has been mapped by the transfer function f can be shown as:

$$O_j = f_j\left(\sum_i w_{ij} x_i\right)$$

Equation 3-1

Figure 3-2
Diagram of the Neurodynamics of Neuron j



A transfer function maps any real numbers into a domain normally bounded by 0 to 1 or -1 to 1. Bounded activation functions are often called *squashing* functions [Sarle 1994]. Early ANN models, like the perceptron used, a simple threshold function (also known as a step-function, hard-limiting activation or Heaviside function):

threshold: $f(x) = 0$ if $x < 0$, 1 otherwise.

The most common transfer functions used in current ANN models are the sigmoid (S-shaped) functions. Masters [1993] loosely defined a sigmoid function as a 'continuous, real-valued function whose domain is the reals, whose derivative is always positive, and whose range is bounded'. Examples of sigmoid functions are:

logistic: $f(x) = \dfrac{1}{1+e^{-x}}$

hyperbolic tangent: $f(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

The logistic function remains the most commonly applied in ANN models due to the ease of computing its derivative:

$$f'(x) = f(x)(1 - f(x))$$

The output, $O_{j,}$ of the neuron x_j of the earlier example in Figure 3-2 if the function f is a logistic function becomes:

$$O_j = \frac{1}{1+e^{-\sum_i w_{ij}x_i - \theta_j}}$$

Equation 3-2

where θ_j is the threshold on unit *j*.

If the function f is a threshold function instead, the output, O_j will be:

$$O_j = \begin{cases} 1, \sum_i w_{ij} x_i > \theta_j \\ 0, else \end{cases}$$

Equation 3-3

However, Kalman and Kwasny [1992] argue that the hyperbolic tangent function is the ideal transfer function. According to Masters [1993], the shape of the function has little effect on a network although it can have a significant impact on the training speed. Other common transfer functions include:

linear or identity:	$f(x) = x$	Normally used in the input and/or output layer.
Gaussian:	$f(x) = e^{-x^2/2}$	

Sigmoid functions can never reach their theoretical limit values and it is futile to try and train an ANN to achieve these extreme values. Values that are close to the limits should be considered as having reaching those values. For example, in a logistic function where the limits are 0 to 1, a neuron should be considered to be fully activated at values around 0.9 and turned off at around 0.1. This is another reason why ANNs cannot do numerical computation as well or as accurate as simple serial computers; i.e. a calculator. Thus ANNs is not a suitable tool for balancing check books!

3.2.4 Weighing Schemes and Learning Algorithms

The initial weights of an ANN are often selected randomly or by an algorithm. The learning algorithm determines how the weights are changed, normally depending on the size of the error in the network output to the desired output. The objective of the learning algorithm is to minimize this error to an acceptable value. The back-propagation algorithm is by far the most popular learning algorithm for multilayer networks and will be discussed in more detail in section 3.4.1.2.

3.3 Neural Networks Architecture

3.3.1 Types of interconnections between neurons

A network is said to be *fully connected* if the output from a neuron is connected to every other neuron in the next layer. A network with connections that pass outputs in a single direction only to neurons on the next layer is called a *feedforward network*. Nelson and Illingworth [1990] define a *feedback network* as one that allows its outputs to be inputs to preceding layers. They call networks that work with closed loops as *recurrent networks*. They also mention networks with *feedlateral connections* that would send some inputs to other nodes in the same layer. Feedforward networks are faster than feedback nets as they require only a single pass to obtain a solution. According to Nelson and Illingworth [1990] recurrent networks are used to perform functions like automatic gain control or energy normalization and selecting a maximum in complex systems.

Most ANN books, however, classify networks into two categories only: feedforward networks and recurrent networks. This is done by classifying all networks with feedback connections or loops as recurrent networks. Fully connected feedforward networks are often called *multi-layer perceptrons (MLPs)* and they are by far the most commonly used ANNs. All the ANNs used in this book are MLPs. They will be discussed in more detail in section 3.3.6.

3.3.2 The Number of Hidden Neurons

Hidden neurons are required to compute difficult functions known as nonseparable functions which are discussed in section 3.3.5. The number of input and output neurons are determined by the application at hand. However, there are no standard rules or theories in determining the number of neurons in the hidden layers although there are some rules of thumb suggested by various ANN researchers:

- Shih [1994] suggested that the network topology should have a pyramidal shape; that is to have the greatest number of neurons in the initial layers and have fewer neurons in the later layers. He suggested the number of neurons in each layer should be a number from mid-way between the previous and succeeding layers to twice the number of the preceding layer. The examples given suggest that a network with 12 neurons in its previous layer and 3 neurons in the succeeding layer should have 6 to 24 neurons in the intermediate layer.
- According to Azoff [1994], a rough guideline based on theoretical conditions of what is know as the Vapnik-Chervonenkis dimension[20], recommends that the number of training data should be at least ten times the number of weights. He also quoted a theorem due to Kolmogorov [Hecht-Nielsen 1990 and Lippman 1987] that suggests a network with one hidden layer and *2N+1* hidden neurons is sufficient for N inputs.
- Lawrence [1994, p. 237] gives the following formula for determining the number of hidden neurons required in a network:
  number of hidden neurons = training facts × *error tolerance.*
- Note: training facts refers to in-sample data while the error tolerance refers to the level of accuracy desired or acceptable error range.
- Baum and Haussler [1988] suggest that the number of neurons in the hidden layer should be calculated as follows: $j = \frac{me}{n+z}$ where *j* is the number of neurons in the hidden layer, *m* is the number of data points in the training set, *e* is the error tolerance, *n* is the number of inputs and *z* the number of outputs.

The latter two rules of thumb are very similar and may not be meaningful in cases where the error tolerances are significantly smaller than the number of training facts. For example, if the number of training facts is 100 and the error tolerance is 0.001, the number of hidden neurons would be 0.1 (meaningless!) in Lawrence's proposal; while Baum and Hassler's proposal would result in an even lower value. Most statisticians are not

[20] Azoff referred to an article by Hush and Horne [1993].

convinced that rules of thumbs are of any use. They argue that there is no way to determine a good network topology from just the number of inputs and outputs [Neural Network FAQ 1996].

The Neural Network FAQ [1996] suggests a method called *early stopping* or *stopped training* whereby a larger number of hidden neurons are used with a very slow learning rate and with small random initial weight values. The out-of-sample error rate is computed periodically during training. The training of the network is halted when the error rate in the out-of-sample data starts to increase. A similar method to *early stopping* is used in the development of the ANNs applications for the financial distress and foreign exchange trading problems of this book. However, those ANNs do not use 'lots of hidden units' as suggested by the article. Instead, they start with small numbers of hidden neurons with the numbers increased gradually only if the ANNs do not seem to 'learn'. In this way, the problem of overfit or curve-fit which can occur when there are more weights (parameters) than sample data can be avoided. However, a recent report by Lawrence et al. [1996] suggest that using "oversize" networks can reduce both training and generalization error.

3.3.3 The Number of Hidden Layers

According to the Neural Network FAQ [1996], hidden layers may not be required at all. It uses McCullagh and Nelder's [1989] paper to support this view. They found linear and generalized linear models to be useful in a wide variety of applications. They suggest that even if the function to be learned is mildly non-linear, a simple linear model may still perform better than a complicated nonlinear model if there is insufficient data or too much noise to estimate the nonlinearities accurately.

MLPs that uses the step/threshold/Heaviside transfer functions need two hidden layers for full generality [Sontag 1992], while an MLP that uses any of a wide variety of continuous nonlinear hidden-layer transfer functions requires just one hidden layer with 'an arbitrarily large number of hidden neurons' to achieve the 'universal approximation' property described by Hornik et al. [1989] and Hornik [1993].

3.3.4 The Perceptron

The perceptron model, as mentioned in earlier chapters, was proposed by Frank Rosenblatt in the mid 1960s. According to Carling [1992], the model was inspired by the discovery of Hubel and Wiesel [1962] of the existence of some mechanism in the eye of a cat that can determine line directions. Rosenblatt developed the perceptron learning theorem (that was subsequently proved by Arbib [1989]) which states that if a set of patterns is learnable by a perceptron, then the perceptron is guaranteed to find the appropriate weight set.

Essentially, Rosenblatt's perceptron model was an ANN model consisting of only an input layer and an output layer with no hidden layer. The input and output layers can have one or more neurons. Rosenblatt's model uses a threshold function as a transfer function although the perceptron model can use any of the transfer functions discussed in section 3.2.3. Therefore if the sum of the inputs is greater than its threshold value, the output neuron will assume the value of 1, or else a value of 0. Fu [1994] states that in terms of classification, an object will be classified by neuron *j* into Class A if

$$\sum w_{ij} x_i > \theta$$

Equation 3-4

where w_{ij} is the weight from neuron i to neuron j, x_i is the input from neuron i, and θ is the threshold on neuron j. If not, the object will be classified as Class B.

The weights on a perceptron model like the one shown in Figure 3-3 are adjusted by

Equation 3-5

where $w_{ij}(t)$ is the weight from neuron i to neuron j at time t (to the *t*th iteration) and Δw_{ij} is the weight adjustment. The weight change is computed by using the delta rule:

$$\Delta w_{ij} = \eta \delta_j x_i$$

Equation 3-6

where η is the learning rate ($0<\eta<1$) and δ_j is the error at neuron j;

$$\boldsymbol{\delta_j = T_j - O_j}$$

Equation 3-7

where T_j is the target output value and O_j is the actual output of the network at neuron j.

The process is repeated iteratively until convergence is achieved. Convergence is the process whereby the errors are minimized to an acceptable level. The delta rule is discussed in more detail in section 3.4.1.1

Ripley [1993] claims that the number of random patterns a perceptron with N inputs can classify without error is finite, since the patterns must be linearly separable. This is irrespective of the existence of an algorithm to learn the patterns. He states that Cover [1965] showed the asymptotic answer is 2N patterns. Ripley also proves the theorem in his paper.

Initially there was widespread optimism as the perceptron could compute a number of simple binary Boolean (logic) functions, i.e. AND, OR and NOT. However, the *caveat emptor* here is that the only patterns that a perceptron can learn are linear patterns which severely limit the type of problems that it could solve. This was the main criticism by Minsky and Papert [1969] leading them to conclude that the perceptron could not solve any ‘interesting problems’. One of the examples of a relatively simple problem that they showed the perceptron could not solve is the exclusive or (XOR) problem which is discussed in the next section.

3.3.5 Linear Separability and the XOR Problem

Linear separability refers to the case when a linear *hyperplane* exists to separate all instances of one class from another. A single plane can separate three-dimensional space into two distinct regions. Thus by extension, if there were *n* inputs where *n > 2*, then *Equation 3-4* becomes:

$$\sum_{i=1}^{n} w_{ij} x_j = \theta_j$$

Equation 3-8

forming a *hyperplane* of *n-1* dimension in the *n*-dimensional space (also called *hyperspace*), dividing the space into two halves. According to Freeman and Skapura [1991, pp. 24-30], many real life problems require the separation of regions of points in hyperspace into individual categories, or classes, which must be distinguished from other classes. This type of problem is also known as a classification problem. Classification problems can be solved by finding suitable arrangements of hyperplanes that can partition *n*-dimensional space into various distinct regions. Although this task is very difficult for *n>2* dimensions, certain ANNs (e.g. MLPs) can learn the proper partitioning by themselves.

As mentioned in the last section, the perceptron can solve most binary Boolean functions. In fact, all but two of the sixteen possible binary Boolean functions, which are the XOR and its complement, are linearly separable and can be solved by the perceptron. The XOR is a function that outputs a 1 if and only if its two inputs are not the same, otherwise the output is 0. The truth table for the XOR function is shown in Table 3-1.

Gallant [1993] showed that a perceptron model (which he called a *single-cell linear discriminant model)* can easily compute the AND, OR and NOT functions. Thus, he defined a Boolean function to be a *separable function* if it can be computed by a single-cell linear discriminant model; otherwise it is classified as a *nonseparable* function. He further states that the XOR is the simplest nonseparable function in that there are no nonseparable function with fewer inputs.

Application of the perceptron model of Figure 3-6 to the XOR problem yields:

$$\begin{aligned} \textit{Output, } O_j &= f(h_j) \\ &= f(w_{1j}x_1 + w_{2j}x_2, \theta) \\ &= \begin{cases} 1, & w_{1j}x_1 + w_{2j}x_2 \geq \theta \\ 0, & w_{1j}x_1 + w_{2j}x_2 < \theta \end{cases} \end{aligned}$$

Equation 3-9

where w_{ij} is the weight on the connection from neuron *i* to *j* and x_i is the input neuron *i, h_j* is the neuron *j*'s activation value and θ is the threshold value of the threshold function *f*.

A set of values must be found so that the weights can achieve the proper output value. We will show that this cannot be done.

From Equation 3-9, a line on the x_1 and x_2 plane is obtained:

$$\theta = w_{1j}x_1+w_{2j}x_2$$

Equation 3-10

By plotting the XOR function and this line for some values of θ, w_1 and w_2 on the x_1 and x_2 plane in

Figure 3-4, we can see that it is impossible to draw a single line to separate the 1s (represented by the squares) and the 0s (represented by the circles).

The next section will demonstrate how a multilayer perceptron (MLP) can be used to solve this problem.

Figure 3-3
A Simple Perceptron Model

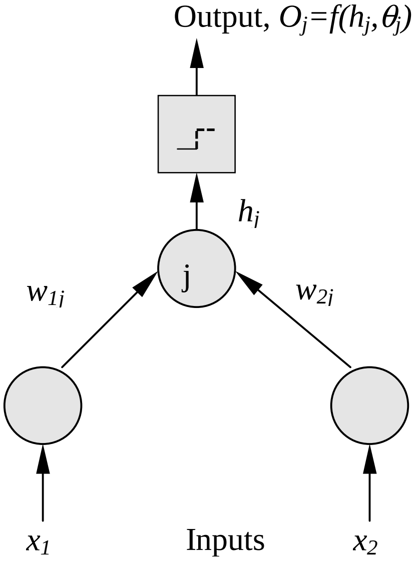


Figure 3-4
A plot of the Exclusive-Or function showing that the two groups of inputs (represented by squares and circles) cannot be separated with a single line.

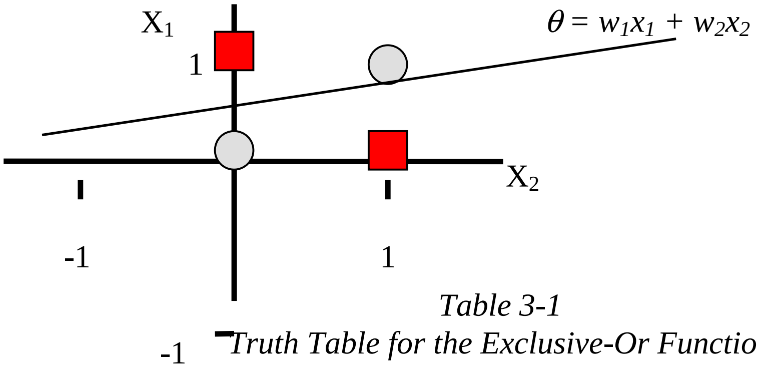


Table 3-1
Truth Table for the Exclusive-Or Function

X_1	X_2	Output
0	0	0
0	1	1
1	0	1
1	1	0

3.3.6 The Multilayer Perceptron

As mentioned in earlier sections, an MLP (also called a multilayer feedforward network) is an extension of the perceptron model with the addition of hidden layer(s) that have nonlinear transfer functions in the hidden neurons. We have also mentioned that an MLP having one hidden layer is a universal approximator, and is capable of learning any function that is continuous and defined on a compact domain[21] as well as functions that consist of a finite collection of points. According to Masters [1993, pp. 85-90], the MLPs can also learn many functions that do not meet the above criteria; specifically discontinuities can be theoretically tolerated and functions that do not have compact support (such as normally distributed random variables) can be learned by a network with one hidden layer under some conditions[22]. Masters states that in practice, a second hidden layer is only required if a function that is continuous has a few discontinuities. He further states the most common reason for an MLP to fail to learn is the violation of the compact domain assumption, i.e. the inputs are not bounded . He concludes that if there is a problem learning in an MLP, it is not due to the model itself but to either insufficient training, or insufficient number of neurons, insufficient number of training samples or an attempt to learn a supposed function that is not deterministic.

[21] A compact domain means that the inputs have definite bounds, rather than having no limits on what they can be.

[22] Kurkova [1995] has since, proven this theoretical assumption.

3.3.6.1 Solving the XOR Problem with A Multilayer Perceptron Model

An MLP model that successfully solves the XOR problem is shown in Figure 3-5. The model incorporates two hidden neurons in the hidden layer. The appropriate weights and threshold values for each neuron are also shown in the diagram. A plot of the XOR function and the two resulting lines from the model is shown in Figure 3-6. The lines have separated the plane into three regions; the central region is associated with the network output of 1 and the remaining two regions containing the points (0,0) and (1,1) are associated with the output of 0.

Figure 3-5
A Multilayer Perceptron Model That Solves the XOR Problem (adapted from Freeman and Skapura 1991, p.29)

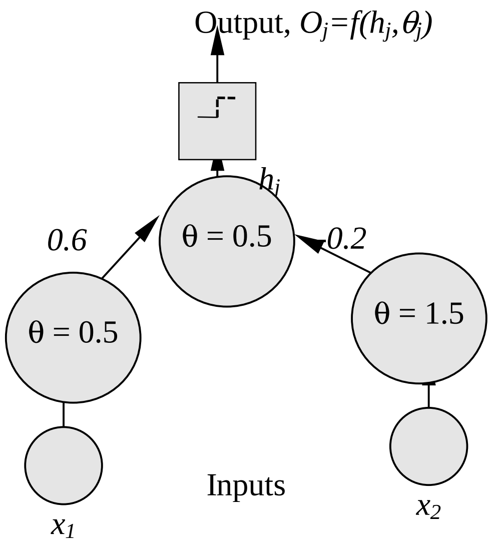
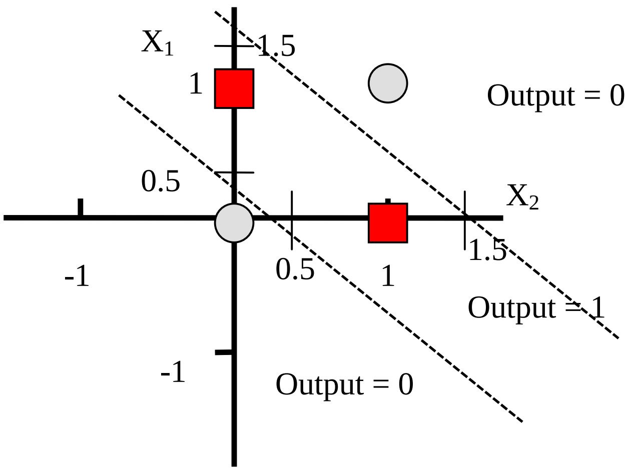


Figure 3-6

A Possible Solution to the XOR Problem By Using Two Lines to Separate the Plane into

Three Regions



3.4 Learning

Learning is the weight modification process of an ANN in response to external input. There are three types of learning:

1. *Supervised learning*

   It is by far the most common type of learning in ANNs. It requires many samples to serve as exemplars. Each sample of this *training set* contains input values with corresponding desired output values (also called target values). The network will then attempt to compute the desired output from the set of given inputs of each sample by minimizing the error of the model output to the desired output. It attempts to do this by continuously adjusting the weights of its connection through an iterative learning process called *training*. As mentioned in earlier sections, the most common learning algorithm for training the network is the back-propagation algorithm.

2. *Unsupervised learning*

   It is sometimes called self-supervised learning and requires no explicit output values for training. Each of the sample inputs to the network is assumed to belong to a distinct class. Thus, the process of training consists of letting the network uncover these classes. It is not as popular as supervised learning and is not used in this book and hence will not be considered further.

3. *Reinforcement learning*

   It is a hybrid learning method in that no desired outputs are given to the network, but the network is told if the computed output is going in the correct direction or not. It is not used in this book and hence will not be considered further.

3.4.1 Learning Algorithms

Although there are many learning algorithms (rules) in common used, this section will only discuss the two most popular ones: the delta rule and its generalization, the back-propagation algorithm. The learning procedures have to select the weights $\{w_{ij}\}$ and the 'biases' $\{\theta_j\}$ which is usually taken to be one [Ripley 1993] by minimizing the total squared error, E:

$$E = \frac{1}{2}\sum_p \left\| t^p - o^p \right\|^2$$

Equation 3-11

where o^p is the output for input x^p, t^p is the target output and the p indexes the patterns in the training set. Both the delta rule and the backpropagation algorithms are a form of the *gradient descent rule*, which is a mathematical approach to minimizing the error between the actual and desired outputs. They do this by modifying the weights with an amount proportional to the first derivative of the error with respect to the weight. The gradient descent is akin to trying to move down to the lowest value of an error surface from the top of a hill without falling into any ravine.

3.4.1.1 The Delta Rule/ Least Mean Squares (LMS) (Widrow-Hoff)

The Least Mean Square (LMS) algorithm was first proposed by Widrow and Hoff (hence, it is also called the Widrow-Hoff Rule) in 1960 when they introduced the ADALINE (Adaptive Linear), an ANN model that was similar to the perceptron model except that it only has a single output neuron and the output activation is a discrete bipolar function[23] that produces a value of 1 or -1. The LMS algorithm was superior to Rosenblatt's perceptron learning algorithm in terms of speed but it also could not be used on networks with hidden layers.

Most literature claims the Delta Rule and the LMS Rule are one and the same [Freeman and Skapura 1991, p. 96, Nelson and Illingworth 1991, p. 137, Carling 1992, p.74, Hecht-Nielsen 1990, p. 61]. They are, in terms of the weight change, Δw_{ij}, formula given in Equation 3-6:

$$\Delta w_{ij} = \eta \delta_j x_i$$

Equation 3-6

where η is the learning rate ($0<\eta<1$) and δ_j is the error at neuron j. However, Fu [1994, p. 30] states that the Widrow-Hoff (LMS) Rule differs from the Delta Rule employed by the perceptron model in the way the error is calculated for weight updating.

From Equation 3-6, the delta rule error was: $\delta_j = T_j - O_j$

The LMS error, δ_j, on the other hand is: $\delta_j = T_j - \Sigma \mathrm{w}_{\mathrm{ij}}\mathrm{x}_{\mathrm{j}}$

Equation 3-12

The LMS rule can be shown to be a gradient descent rule.

From Equation 3-11, if we substitute output O^p with $X_p W^p$

[23] This is the also the reason why it does not work in networks with hidden layers.

$$E = \frac{1}{2}\sum_{p}\left\| t^{p} - X_{p}\ W^{p} \right\|^{2}$$

Equation 3-13

where X_p is an input vector and W^p the weights vector.

Then, the gradient descent technique minimizes the error by adjusting the weights:

$$\Delta W = -\eta \frac{\delta E}{\delta W}$$

Equation 3-14

where η is the learning rate. From Equation 3-13 and Equation 3-14, the LMS rule can be rewritten as

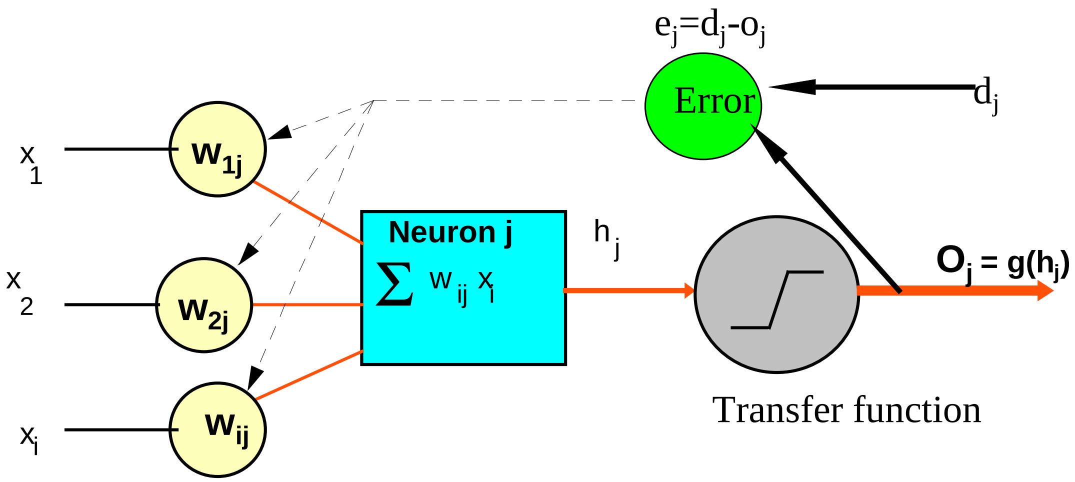
$$\Delta W = -\eta(t_{p} - X_{p}W^{p})X_{p}$$

Equation 3-15

3.4.1.2 The Back-propagation (BP)/Generalized Delta Rule

The back-propagation (BP) algorithm is a generalization of the delta rule that works for networks with hidden layers. It is by far the most popular and most widely used learning algorithm by ANN researchers. Its popularity is due to its simplicity in design and implementation.

Figure 3-7
This is similar to Figure 2-2 in chapter 2. Back-propagation of errors for a single neuron j.



The single neuron model of Figure 3-7 will be used to explain the BP algorithm. The BP algorithm is used mainly with MLP but a single neuron model is used here for clarity. The methodology remains the same for all models.

The BP algorithm involves a two-stage learning process using two passes: a forward pass and a backward pass. In the forward pass, the output O_j is computed from the set of input patterns, X_i:

$$O_j = g(h_j) = f(h_j, \theta_j)$$

$$h_j = \sum_{i=1}^{i} w_{ij} x_j$$

$$Therefore,\ O_j = f(\sum_{i=1}^{i} w_{ij} x_i, \theta_j)$$

Equation 3-16

where f is a nonlinear transfer function, e.g. sigmoid function, θ_j is the threshold value for neuron j, x_i is the input from neuron i and w_{ij} is the weights associated with the connection from neuron i to neuron j.

After the output of the network has been computed, the learning algorithm is then applied from the output neurons back through the network, adjusting all the necessary weights on the connections in turn. The weight adjustment, Δw_{ij}, is as in the LMS Rule Equation 3-6,

$$\Delta w_{ij} = \eta \delta_j x_i$$

Equation 3-6

where η is the learning rate ($0<\eta<1$) and δ_j is the error at neuron j;

$$\delta_j = e_j(O_j)(1-O_j) = \left(\sum \delta_k w_k\right) O_j \left(1 - O_j\right)$$

Equation 3-17

for hidden neurons where k is the neuron receiving output from the hidden neuron.

The adjustments are then added to the previous values:

$$New\ Weight\ Value: w_{ij} = w'_{ij} + \Delta \mathrm{w}_{\mathrm{ij}}$$

Equation 3-18

where w'_{ij} is the previous weight term.

The gradient descent method is susceptible to falling of a chasm and becoming trapped in local minima. If the error surface is a bowl, imagine the gradient descent algorithm as a marble rolling from the top of the bowl trying to reach the bottom (global minimum of the error term, i.e. the solution). If the marble rolls too fast, it will overshoot the bottom and swing to the opposite side of the bowl. The speed of the descent can be controlled with the learning rate term, η. On the other hand, if the learning rate is set to a very small value, the marble will descent very slowly and this translates to longer training time. An error surface of a typical problem is normally not a smooth bowl but may contain ravine and chasm where the marble could fall into. A momentum term is thus often added to the basic method to avoid the model's search direction from swinging back and forth wildly.

The weight adjustment term of Equation 3-6 will then translate to:

$$\Delta w_{ij} = (1 - M)\eta\delta_j x_i + M(w'_{ij} - w''_{ij})$$

Equation 3-19

where M is the momentum term and w''_{ij} is the weight before the previous weight w'_{ij}. The momentum term allows a weight change to persist for a number of adjustment cycles. Notice if M is set to zero, then the equation reverts to Equation 3-6.

Random noise is often added to the network to alleviate the local minima problem. The objective of the noise term is to ‘jolt’ the model out of a local minima. Fahlman [1992] states that BP networks can and do fall into local minima but they are often the ones that are needed to solve the particular problem. In other words, local minima solutions may suffice for some problems and there is no need to seek the global minimum[24].

There are many other variations to the BP algorithm but by far, BP still proves to be the most popular and is implemented in almost all commercial ANN software packages.

3.5 Statistical Aspects of Artificial Neural Networks

3.5.1 Comparison of ANNs to Statistical Analysis

In traditional statistical analysis, the modeller is required to specify the precise relationship between inputs and outputs and any restrictions that may be implied by theory. ANNs differ from conventional techniques in that the analyst is not required to specify the nature of the relationships involved; the analyst simply identifies the inputs and the outputs. According to Sarle [1994], no knowledge of ANN training methods such as back-propagation is required to use ANNs. In addition, Sarle states that the MLP’s main strength lies in its ability to model problems of different levels of complexity, ranging from a simple parametric model to a highly flexible, nonparametric model. For example, an MLP that is used to fit a nonlinear regression curve, using one input, one linear output, and one hidden layer with a logistic transfer function, can function like a polynomial regression or least squares spline. It has some advantages over the competing methods. Polynomial regression is linear in parameters and thus is fast to fit but suffers from numerical accuracy problems if there are too many wiggles. Smoothing splines are also linear in parameters and do not suffer from numerical accuracy problems but pose the problem of deciding where to locate the knots. MLPs with nonlinear transfer function, on the other hand, are genuinely nonlinear in the parameters and thus require longer computational processing time. They are more numerically stable than high-order polynomials and do not require knot location specification like splines. However, they may encounter local minima problems in the optimization process.

3.5.2 ANNs and Statistical Terminology

Although there are many similarities between ANN models and statistical models, the terminology used in both fields are quite different. For example, Sarle [1994] claims that the terminology ‘back-propagation’, should refer only to the simple process of applying the chain rule to compute derivatives for the generalized delta rule algorithm, and not the training method itself. He adds that this confusion is symptomatic of the general failure in ANN literature to differentiate between models and estimation methods. Sarle [1996]

[24] This assumes that the global minimum is not very far from the local minima.

gives a list of statistics terminology that has its equivalence in ANN literature. Some of the more common ones are listed in Table 3-2.

Table 3-2

Statistical and ANN Terminology

Statistical Terminology	***ANN Terminology***
variables	features
independent variables	inputs
predicted values	outputs
dependent variables	targets or training values.
residuals	errors
estimation	training, learning, adaptation, or self-organization.
an estimation criterion	an error function, cost function, or Lyapunov function
observations	patterns or training pairs
parameter estimates	(synaptic) weights
regression and discriminant analysis	supervised learning
cluster analysis or data reduction	unsupervised learning, self-organization or competitive learning
interpolation and extrapolation	generalization
intercept	bias
error term	noise
forecasting	prediction

3.5.3 Similarity of ANN Models to Statistical Models

In general, feedforward nets with no hidden layer are basically generalized linear models [Neural Nets FAQ, Part 1, 1996]. Sarle [1994] states that the perceptron model with different transfer functions has been shown to have equivalent statistical models. For example:

- A perceptron model with a linear transfer function is equivalent to a possibly multiple or multivariate linear regression model [Weisberg 1985; Myers 1986].
- A perceptron model with a logistic transfer function is a logistic regression model [Hosmer and Lemeshow 1989].
- A perceptron model with a threshold transfer function is a linear discriminant function [Hand 1981; McLachlan 1992; Weiss and Kulikowski 1991]. An ADALINE is a linear two-group discriminant.

MLP models have their statistical equivalent models too. For example:

- An MLP with one output is a simple nonlinear regression [Sarle 1994].
- An MLP with a moderate number of hidden neurons is essentially the same as a projection pursuit regression, except that an MLP uses a predetermined transfer function while the projection pursuit regression model uses a flexible nonlinear smoother [Sarle 1994].
- An MLP becomes a nonparametric sieve if the number of hidden neurons is allowed to increase with the sample size [White 1988]. This makes it a useful alternative to methods such as kernel regression [Hardle 1990] and smoothing splines.

Kuan and White [1995] states that when Cowan [1967] proposed the replacement of the Heaviside activation function with a smooth sigmoid function, specifically, the logistic function, the model proposed by McCulloch and Pitts [1943] becomes similar to the binary logit probability model [Ameniya 1981, 1985 p. 268]. Ameniya states that these models have great utility in econometric applications where binary classifications or decisions are involved.

White [1992, p. 84] states that the back-propagation algorithm is not new and is, in fact, a statistical method call *stochastic approximation*, first proposed by Robins and Munro in 1951. This had lead to an explosion of both theoretical and applied research of the field in the past 40 years. It has been used extensively in pattern recognition and systems identification literature. The major advantage of this, is that the considerable statistics and engineering literature on stochastic approximation can be applied to make general statements about the BP algorithm [White 1989a].

3.5.4 ANNs vs. Statistics

Sarle [1993] claims that "many ANN researchers are engineers, physicists, neurophysiologists, psychologists, or computer scientists, who know little about statistics and nonlinear optimization and they often reinvent methods that have been known in statistical or mathematical literature for decades or centuries without understanding how these methods work". He reasons that the common implementation of ANNs is biological or engineering criteria based, such as how easy it is to fit the net on a chip, rather than on well-established statistical and optimization criteria. Ripley [1993] seems to agree with Sarle (both being statisticians by training), stating that ANNs have been developed very rapidly by workers with diverse backgrounds, most with little or no experience in data analysis.

Ripley [1993] claims that although comparisons of ANNs to other methods are rare, however, when done carefully, often show that statistical methods can outperform the state-of-the-art ANNs. His paper includes a comment from Aharonian [1992] on ANNs and financial applications. Aharonian states that most ANNs papers on financial analysis either report results no more accurate than those obtained by traditional statistical techniques; or they fail to compare their results to traditional statistical analysis and by not doing so, invalidate any claims of a breakthrough.

Before the popularity of ANNs, few financial institutions used any form of statistical methods (except for technical analysis, which some may claim to be pseudo-statistics) for financial trading, and even fewer had a dedicated quantitative analysis unit for financial analysis which is now a common sight in most major banks' dealing rooms. As mentioned in chapter 1, financial institutions are second only to the US Department of Defense in sponsoring research into ANNs [Trippi and Turban 1996].

3.5.5 Conclusion of ANNs and Statistics

Sarle [1994] concludes it is unlikely that ANNs will supersede statistical methodology as he believes that applied statistic is highly unlikely to be reduced to an automatic process or 'expert system'. He claims that statisticians depend on human intelligence to understand the process under study and an applied statistician may spend more time defining a problem and determining what questions to ask than on statistical computation. He does, however, concede that several ANNs models are useful for statistical applications and that better communication between the two fields would be beneficial. White [1992, p. 81] agrees that statistical methods can offer insight into the properties, advantages and disadvantages of the ANN learning paradigm, and conversely ANN learning methods have much to offer in the field of statistics. For example, statistical methods such as Bayes analysis and regression analysis have been used in generating forecasts with confidence intervals that have deeper theoretical roots in statistical inference and data generating processes. ANN is superior for pattern recognition and is able to deal with any model whereas statistical methods require randomness.

ANNs have contributed more to statistics than statisticians would care to admit. They have enabled researchers from different disciplines and backgrounds to use modeling tools that were once only available to statisticians due to the complexities and restrictive conditions imposed by statistical models. By making modeling more accessible (and more interesting perhaps), ANNs researchers without statistical background are beginning to gain an appreciation of statistical methodologies due to the inevitable crossing of paths between ANNs and statistics.

There are definitely more visible ANN commercial applications than statistical applications even though the claim that some of the ANN methodologies were already been 'known for decades if not centuries in statistical and mathematical literature [Sarle 94]'.

3.6 References

1. Aharonian, G., *Comments on comp.ai.neural-nets*, Items 2311 and 2386 [Internet Newsgroup].
2. Ameniya, T., "Qualitative Response Models: A Survey", *Journal of Economic Literature*, No. 19, pp. 1483-1536, 1981.
3. Ameniya, T., *Advance Econometrics*, Cambridge, Harvard University Press, 1985.
4. Azoff, E. M., *Neural Network Time Series Forecasting of Financial Markets*, John Wiley & Sons, pp. 50-51, England, ISBN: 0-471-94356-8, 1994.
5. Baum, E. B. and Haussler, D., *Neural Computation 1*, 1988, 151–160.
6. Carling, A., *Introducing Neural Networks*, Sigma Press, ISBN: 1-85058-174-6, England, 1992.
7. Cover, T. M., "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition", *IEEE Trans. Elect. Comp.*, No. 14, pp. 326-334, 1965.
8. Falhman, S. E., *Comments on comp.ai.neural-nets*, Item 2198 [Internet Newsgroup].
9. Freeman, J. A. and Skapura, D. M., *Neural Networks: algorithms, applications, and programming techniques*, Addison-Wesley, ISBN 0-201-51376-5, October 1991.
10. Hand, D. J., *Discrimination and Classification,* John Wiley & Sons, New York, 1981.
11. Hardle, W., *Applied Nonparametric Regression*, Cambridge University Press, Cambridge, UK, 1990.
12. Haynes, J. and Tan, C.N.W., "An Artificial Neural Network Real Estate Price Simulator", *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES) (Addendum)*, University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993.
13. Hect-Nielsen, R., *Neurocomputing*, Addison-Wesley, Menlo Park, CA, USA, ISBN: 0-201-09355-3, 1990.
14. Hornik, K., "Some new results on neural network approximation*", Neural Networks*, 6, 1069-1072, 1993.
15. Hornik, K., Stinchcombe, M. and White, H., "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 359-366, 1989.
16. Hosmer, D. W. and Lemeshow, S., *Applied Logistic Regression*, John Wiley & Sons, New York, 1989.
17. Hubel, D. H. and Wiesel, T. N., "Receptive fields, binocular and functional architecture in the cat's visual cortex", *J. Physiol.*, 160: 106-154, 1962.
18. Hush, D. R. and Horne, B. G., "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine*, vol. 10, no. 1, pp. 8-39, January 1993.
19. Kalman, B. L, and Kwasny, S. C., "Why Tanh? Choosing a Sigmoidal Function", *International Joint Conference on Neural Networks*, Baltimore, MD, USA, 1992.

20. Kuan, C.-M. and White, H., “Artificial Neural Networks: An Econometric Perspective”, *Econometric Reviews*, vol. 13, No. 1, pp. 1-91, 1994.

21. Lawrence, J.*, Introduction to Neural Networks: Design, Theory, and Applications 6th edition*, edited by Luedeking, S., ISBN 1-883157-00-5, California Scientific Software, California, USA, July 1994.

22. Lawrence, S., Giles, C. L., and Tsoi, A. C., “What Size Neural Networks Gives Optimal Generalization? Convergence Properties of Backpropagation”, *Technical Report UMIACS-TR-96-22 and CS-TR-3617*, Institute of Advanced Computer Studies, University of Maryland, College Park, MD 20742, 1996.

23. Lippmann, R. P., “An Introduction to Computing with Neural Nets”, *IEEE ASSP Magazine*, pp. 4-23, April 1987.

24. Masters, T., *Practical Neural Network Recipes in C++*, Academic Press Inc., San Diego, CA., USA, ISBN: 0-12-479040-2, p.6, 1993.

25. McCullagh, P. and Nelder, J. A. *Generalized Linear Models*, 2nd ed., Chapman & Hall, London, UK, 1989.

26. McCulloch, W. S. and Pitts, W., “A Logical Calculus of Ideas Immanent in Nervous Activity”, *Bulletin of Mathematical Biophysics*, pp. 5:115-33, 1943.

27. McLachlan, G. J., *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, New York, 1992.

28. Minsky, M. and Papert, S. A., *Perceptrons. Expanded Edition*, MIT Press, Cambridge, MA, USA, ISBN: 0-262-63111-3, 1988.

29. Nelson, M. M. and Illingworth, W. T., *A Practical Guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., USA, ISBN: 0-201-52376-0/0-201-56309-6, 1991.

30. Neural Network FAQ, Maintainer: Sarle, W. S., “How Many Hidden Units Should I Use?”, July 27, 1996, *Neural Network FAQ Part 1-7*, [Online], Archive-name:ai-faq/neural-nets/part3, Available: ftp://ftp.sas.com/pub/neural/FAQ3.html, [1996, August 30].

31. Ripley, B. D., “Statistical Aspects of Neural Networks”, Networks and Chaos: Statistical and Probabilistic Aspects edited by Barndoff-Nielsen, O. E., Jensen, J.L. and Kendall, W.S., Chapman and Hall, London, United Kingdom, 1993.

32. Robbins, H. and Munro, S., “A stochastic approximation method”, *Annals of Mathematical Statistics*, No. 25, p. 737-44, 1951.

33. Sarle, W. S., “Neural Networks and Statistical Models”, *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute, USA, pp. 1538-1550, 1994.

34. Sarle, W. S., “Neural Network and Statistical Jargon?”, April 29, 1996, [Online], Archive-name:ai-faq/neural-nets/part3, Available: ftp://ftp.sas.com/pub/neural/jargon, [1996, August 24].

35. Shih, Y., *Neuralyst™ User’s Guide*, Cheshire Engineering Corporation, USA, pp.74, 1994.

36. Sontag, E. D. (), “Feedback stabilization using two-hidden-layer nets”, *IEEE Transactions on Neural Networks*, 3, 981-990, 1992.

37. Weiss, S. M., and Kulikowski, C. A., *Computer Systems That Learn*, Morgan Kauffman, San Mateo, CA, 1991.

38. White, H., *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell Publishers, Oxford, UK, ISBN: 1-55786-329-6, 1992.

39. White, H., "Some asymptotic results for learning in single hidden layer feedforward network models", *Journal of American Statistical Association*, No. 84, p. 1008-13, 1989.

40. Widrow, B. and Hoff, M. D., "Adaptive Switching Circuits", *1960 IRE WESCON Convention Record*, Part 4, pp. 96-104, 1960.

[25]Chapter 4: Using Artificial Neural Networks to Develop An Early Warning Predictor for Credit Union Financial Distress

"Economic distress will teach men, if anything can, that realities are less dangerous than fancies, that fact-finding is more effective than fault-finding"
Carl Becker (1873-1945), Progress and Power

[25] Part of this chapter has been published in Neural Networks in Finance and Investing edited by Trippi and Turban, Irwin, USA, Chapter 15 pp. 329-365, ISBN 1-55738-919-6, 1996

4. Using Artificial Neural Networks to Develop An Early Warning Predictor for Credit Union Financial Distress

4.1 Introduction

Since Beaver's [1966] pioneering work in the late 1960s there has been considerable interest in using financial ratios to predict financial failure[26]. The upsurge in interest followed the seminal work by Altman [1968] in which he combines five financial ratios into a single predictor (which he calls factor Z) of corporate bankruptcy[27]. An attractive feature of Altman's methodology is that it provides a standard benchmark for comparison of companies in similar industries. It also enables a single indicator of financial strength to be constructed from a company's financial accounts. While the methodology is widely appealing, it has limitations. In particular, Gibson and Frishkoff [1986] point out that ratios can differ greatly across industrial sectors and accounting methods[28].

These limitations are nowhere more evident than in using financial indicators to predict financial distress among financial institutions. The naturally high leverage of financial institutions means that models developed for the corporate sector are not readily transportable to the financial sector. The approach has nonetheless gained acceptance in its application to financial institutions by treating them as a unique class of companies. Recent examples in Australia include unpublished analyses of financial distress among non-bank financial institutions by Hall and Byron [1992] and McLachlan [1993]. Both of these studies use a Probit model to deal with the limited dependent variable nature of financial distress data.

This study examines the viability of an alternative methodology for the analysis of financial distress based on artificial neural networks (ANNs). In particular, it focuses on the applicability of ANNs as an early warning predictor of financial distress among credit unions. The ANN-based model developed in this chapter is compared with the Probit model results of Hall and Byron. In particular, this study is based on the same data set used by Hall and Byron. This facilitates an unbiased comparison of the two methodologies. The results reported in the paper indicate that the ANN approach is marginally superior to the Probit model over the same data set. The paper also considers ways in which the model design can be altered to improve the ANN's performance as an early warning predictor.

4.2 Existing Studies: Methodological Issues

Discriminant analysis is one of the most popular techniques used to analyze financial data in the context of financial distress. This method has been described by Jones [1987, p.

[26] See, for example, Beaver [1966], Ohlson [1980], Frydman Altman and Kao [1985], Casey and Bartczak [1985] and McKinley et al. [1983] and the works cited in these studies.

[27] The function is $Z = 0.12X_1+0.014X_2+0.033X_3+0.006X_4+0.999X_5$ where X_1 = Working capital/Total Assets (%), X_2 = Total retained earnings/total assets (%), X_3 = Earnings before interest and taxes (EBIT)/total assets (%), X_4 = Market value of equity/book value of total debt (%) and X_5 = Sales/total assets. Rowe et al. [1994, p. 373] states that in some cases, the Z-factor can be approximated with the simplified equation: $Z \approx \frac{sales}{total\_assets}$.

[28] These cautions are reinforced by Horrigan [1968] and Levy and Sarnat [1988].

143] as ‘a multivariate technique that assigns a score, z, to each company in a sample, using a combination of independent variables’. The analyst then decides a cutoff z-score based on the sample results; companies below the cutoff are predicted to experience bankruptcy while those above the cutoff are predicted to remain healthy. The main appeal of this approach is its ability to reduce a multidimensional problem to a single score. Altman [1968] was the first to use Discriminant Analysis in predicting bankruptcy. Studies using the Discriminant Analysis methodology generally find a high level of classification accuracy[29]

The main criticism of the Discriminant Analysis method is the restrictive statistical requirements posed by the model. For example, the requirement that the independent variables have a multivariate normal distribution is often violated as is the case when dummy independent variables are used. Further, the score that is produced by the model is of limited use in interpreting the results, since it is basically an ordinal ranking. There is also no simple method of determining the statistical significance of the contributions of the various independent variables to the overall score.

Binary choice models (or limited dependent variable techniques) such as Probit, Tobit and Logit, are able to overcome the main weaknesses of Discriminant Analysis. Martin’s paper (1977) on bank failure is the seminal work in the use of binary choice regression techniques in this area[30]

Martin compared the classification accuracy of a Logit regression based on the cumulative logistic function with Multiple Discriminant Analysis in analyzing financial distress among a large number of Federal Reserve supervised banks from 1970 to 1976. He found that, while Logit and Multiple Discriminant Analysis had similar levels of accuracy, both methods were superior to the linear discriminant model.

In a study of corporate failures Collins and Green [1982] found that the Logit model appeared to produce less Type I errors (misclassifying a failed firm as healthy) but that the method was not significantly better than Multiple Discriminant Analysis. They concluded that the additional computational effort required by the Logit model may not be justified unless the cost of Type I errors is very large.

The least supportive study of these general methodologies is that by Pacey and Pham [1990] who address three methodological problems in bankruptcy prediction models:

1. the use of choice-based and equally-distributed samples in model estimation and validation;
2. arbitrary use of cutoff probabilities; and
3. the assumption of equal costs of errors in predictions.

[29] See, for example, Deakin [1972], Libby [1975ab], Schipper [1977], Altman, Haldeman and Narayanan [1977], Dambolena and Khoury [1980], Gombola and Ketz [1983], Casey and Bartzak [1985], Gentry, Newbold and Whitford [1985a] and Sinkey [1975].

[30] Other studies that have used binary choice analysis in financial distress prediction include Ohlson [1980], Gentry, Newbold and Whitford [1985b], Casey and Bartzak [1985] and Zavgren [1985].

Using both Probit and multiple discriminant models to correct these problems, they found that neither the multiple discriminant model nor the Probit model outperformed a naive model which assumed all firms to be non-bankrupt.

The study that is used as the basis for comparison in this chapter is that by Hall and Byron. Hall and Byron use a Probit model with thirteen basic financial ratios to predict financial distress among credit unions in New South Wales. Of the thirteen ratios, four were found to make a significant contribution to predicting financial distress. The significant ratios were:

RA: Required Doubtful Debt Provision

RB: Permanent Share Capital + Reserves + Overprovision for Doubtful Debt to Total Assets (%)

RC: Operating Surplus to Total Assets (%)

RG: Operating Expenses to Total Assets (%)

Their estimated index function, Y, was:

$$Y = 0.330RA - 0.230RB -0.671RC + 0.162RG - 1.174 - 0.507Q1 -0.868Q2+0.498Q3$$

where the variables Q1 to Q3 are seasonal dummy variables to capture any seasonal effects in the data.

A conditional probability of financial distress is obtained by referring to the cumulative normal statistical tables. Any Credit Unions with a conditional probability greater than one were classified by Hall and Byron as being in ‘distress’.

4.3 Applications of ANNs in Predicting Financial Distress

Recently, ANNs have been used in predicting financial distress with a few reported successful applications. Odom and Sharda [1990] found that a back-propagation artificial neural network was superior to a Discriminant Analysis model in bankruptcy prediction of firms. The accuracy of their model has since been improved upon by Neuralware’s Applications Development Service and Support (ADSS) group [Coleman et al. 1991]. Coleman reported that the ADSS group had successfully developed an ANN-based system to detect bank failures for the accounting firm of KPMG Peat Marwick[31]. In their analysis they claim an accuracy rate of 90%. Salchenberger et al. [1992] showed that an ANN-based model performed as well as or better than the Logit model. They also observed that when the cutoff point (probability level) was lowered, the reduction in Type I errors (misclassifying a failed firm as healthy) was accompanied by a greater increase in Type II errors (misclassifying a healthy firm as failed) for the Logit model than for the ANN model.

In their survey of Savings and Loan Associations, Tam and Kiang [1992] argue that empirical results have shown that ANNs have better predictive accuracy than Discriminant Analysis, Logit, k Nearest Neighbor (kNN) and Decision Tree (ID3) analysis. They further argue that ANNs may be a better alternative to classification techniques under the following conditions:

[31] The results were subsequently published by Bell et al. [1990].

1. Multimodal distributions - improvement here is due to the ANN's ability to better represent the nonlinear discriminant function. Many classification tasks have been reported to have nonlinear relationships between variables. For example, Whitred and Zimmer [1985] find that the higher prediction accuracy of loan officers to linear Discriminant Analysis models is due to their ability to relate variables and loan outcomes in a nonlinear manner. Shepanski [1983] also finds that human judgments are better approximated by a nonlinear function.
2. Adaptive model adjustment - ANNs have the ability to adapt to the changing environment by adjusting the model, thus allowing the model to respond swiftly to changes.
3. Robustness - ANNs make no assumptions of any probability distribution or equal dispersion, nor are there any of the rigid restrictions found in other models, such as linearity.

In the same ANN framework, this chapter discusses a back-propagation model that uses financial ratios as input to build a model that for predicting financial distress in Credit Unions in New South Wales, Australia. The back-propagation model was chosen as it had been used quite successfully in bankruptcy prediction tasks [Coleman et al., Bell et all, Tam et al., Odom et al. and Salchenberger et al.] and tools for implementing it are readily available. ANNs have also gain prominence in the accounting field as a tool for predicting bankruptcies using funds flows, accrual rations and accounting data [Back et al. 1996] as well as in the insurance field for obtaining early warning of insurer insolvency [Brockett et al. 1994].

4.4 Data and Testing Methodology

As Hall and Byron note in their paper, defining failure of credit unions in Australia is not a clear cut process as many of the failed credit unions are not resolved in bankruptcy. They are mostly resolved by forced mergers, voluntary mergers or being placed under direction. Since this study uses the same data set as Hall and Byron, the definition of the distress category will be the same as theirs; namely, those Credit Unions which are placed under direction or placed under notice of direction.

The binary format for the output of the models is 1 for credit unions classified as 'Distress' and 0 for credit unions classified as 'Non-Distress'.

The data used in the study are quarterly financial data for 191 New South Wales Credit Unions from 1989 to 1991. The data were 'cleaned' by Hall and Byron to exclude all credit unions with total assets less than A$60,000. The total number of observations obtained for the study was 2144, of which 66 were classified as in distress. The input (independent) variables were financial ratios derived from the financial data used by Hall and Byron.

4.4.1 In-sample (Training) and Out-of Sample (Validation) Data Sets

There are two popular methods of validating bankruptcy prediction models. The first method is to separate a single data set into two, using one to build the model and the second to test the model. The second method involves using data from one time period as in-sample data and data from another similar time period as the out-of-sample test set. In this research the former method is adopted.

The data set was divided into two separate sets. Data for all quarters of 1989 to 1990 were used as the training set (in-sample data) to build the early warning predictor, while data for all quarters of 1991 were used as the validation set (out-of-sample data). The training sets contained a total of 1449 observations with 46 credit unions in the distress category. The validation set contained a total of 695 observations with 20 credit unions classified as in distress.

4.4.2 Input (Independent) Variables

The inputs used in the ANN are the same variables used by Hall and Byron. They consider thirteen financial ratios to reflect the stability, profitability and liquidity of a Credit Union plus four dummy variables to indicate the quarters in a year (See Table 4.1 below). Hall and Byron argue that the quarterly seasonal dummies are needed to adjust for the seasonality in some of the ratios. They also conducted a statistical analysis on the ratios to determine their significance to credit unions in distress.

Hall and Byron find only four of the thirteen ratios and three of the four quarterly dummy variables statistically significant as independent variables and thus incorporated only those variables in their final model. Using the ANN methodology, the ANN is allowed to determine the significance of the variables by incorporating all the available information as input in the model. The reason for this is that ANNs are very good at dealing with large noisy data sets and, in their learning processes, eliminate inputs that are of little significance by placing little or no weight values on the connections between the input nodes of those variables. The tradeoff is that larger networks require larger amounts of training time.

The financial ratios and Hall and Byron's comments on their significance are reproduced in Table 4-1.

Table 4-1 Hall and Byron's Financial Ratios

Ratio	**Definition**	**Comments**
RA:	Required Doubtful Debt Provision	Distress significantly larger
RB	Permanent Share Capital + Reserves + Overprovision for Doubtful Debt to Total Assets (%)	Distress significantly smaller
RC	Operating Surplus to Total Assets (%)	Distress significantly smaller
RD	Operating Surplus to Total Income (%)	Distress significantly smaller
RE	Required Doubtful Debt Provision to Actual Doubtful Debt Provision (%)	Distress significantly smaller
RF	Liquid Funds to Total Assets	Distress significantly smaller
RG	Operating Expenses to Total Assets (%)	Substantial seasonality
RH	Physical Assets to Total Assets (%)	No significant difference
RI	Loans Under 5 Years to Total Loans (%)	No significant difference
RJ	Delinquent Loans to Total Loans (%)	Distress significantly larger
RK	Required Doubtful Debt Provision to Total Loans (%)	Distress significantly larger
RL	Actual Doubtful Debt Provision to Total Loans (%)	No significant difference
RM	Gross Profit Margin = Total Income - Cost of Funds to Total Income (%)	No significant difference

The four dummy variables are as follows:

Q1=1 in March quarter and =0 otherwise

Q2=1 in June quarter and =0 otherwise

Q3=1 in September quarter and =0 otherwise

Q4=1 in December quarter and =0 otherwise

Application of the ratios in Table 4.1 leads to an input layer of the ANN consisting of 17 neurons with each neuron representing one of the above input variables. The output layer consists of only one output, indicating the status of the Credit Union as either distressed or not. The objective is for the ANN to predict the binary output of the status of the Credit Unions, with 1 indicating that the Credit Union is in distress and 0 indicating that is in non-distress. The output values of the ANN are continuous with upper and lower bounds of 0 and 1. Therefore, even though the objective or target values themselves are discrete, probability theory can be used to interpret the output values.

4.5 ANN Topology and Parameter Settings

The input data to the ANN were the financial ratios for each quarter of a Credit Union and the desired output (target) was the binary status of the Credit Unions i.e. ‘Distress’ or ‘Non-Distress’. The final model consisted of a hidden layer with five neurons, seventeen input neurons and one output neuron. The parameter settings are shown in Table 4.2.

The ANNs constructed in this study all used the same set of initial weights. This allowed the results obtained to be replicated. The ANNs were trained over 25,000 iterations, although the best model needed only 3,000 iteration to be fully trained. Further training did not improve results and actually reduced accuracy. This was probably due to ‘curve-fitting’ which as noted earlier results from overtraining (Where an ANN specifically models the training set rather than building a general model of the problem at hand).

Many ANNs were constructed with different network topologies and different parameter settings in an attempt to find the best model. Measuring the performance of the models is not without some subjectivity. The usual approach, including that adopted by Hall and Byron, is to judge performance by overall accuracy; for example, by the minimum root mean square error (rmse) of forecasts. Performance in any prediction, however, involves two types of errors: Type I error if the null hypothesis is rejected when it is true and Type II error which occurs if the null hypothesis is not rejected when it is false. Since the objective of the exercise is to predict distress, it is reasonable to assume that Type I errors involve the misclassification of financially distressed Credit Unions as healthy while Type II errors involve the misclassification of healthy Credit Unions as distressed. Type I errors can be very costly to regulators in that they could generate financial crisis or loss of confidence in the regulator. The cost of Type II errors is mainly that associated with the extra work required in analyzing Credit Unions with potential problems. Therefore, this research uses as its criterion for selecting the best model the minimization of Type I errors in the training (in-sample) data sets[32]. Chart 4-1 shows the optimal number of iterations for the training set is 3000. The ANN model constructed after the 3000 iterations also provided one of the lowest Type I errors committed in the test set. Further training resulted in less accurate ANN models.

[32] Incidentally, the ANN that gave the minimum Type I errors in the in-sample data set also gave the minimum Type I errors for the combined in-sample and the out-of-sample data sets. See Chart 4-1.

Chart 4-1: Training and Test Set Type I Error

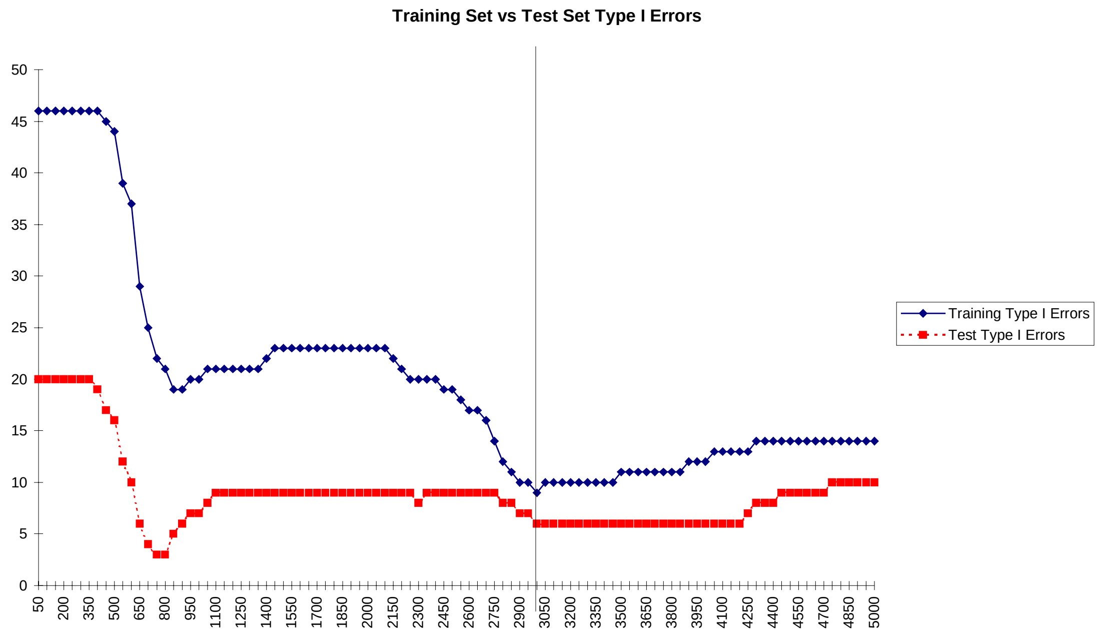


Table 4-2 Artificial Neural Networks Parameters

Network Parameters	
Learning rate	0.05
Momentum	0.1
Input Noise	0
Training Tolerance	0.9
Testing Tolerance	0.9

A brief description of each of the parameters is discussed below:

4.5.1 Learning Rate

The learning rate determines the amount of correction term that is applied to adjust the neuron weights during training. The learning rate of the neural net was tested with values ranging from 0.05 to 0.1.

Small values of the learning rate increase learning time but tend to decrease the chance of overshooting the optimal solution. At the same time, they increase the likelihood of becoming stuck at local minima. Large values of the learning rate may train the network faster, but may result in no learning occurring at all. Small values are used so as to avoid missing the optimal solution. The final model uses 0.05; the lowest learning rate in the range.

4.5.2 Momentum

The momentum value determines how much of the previous corrective term should be remembered and carried on in the current training. The larger the momentum value, the greater the emphasis placed on the current correction term and the less on previous terms. The momentum value serves as a smoothing process that ‘brakes’ the learning process from heading in an undesirable direction.

4.5.3 Input Noise

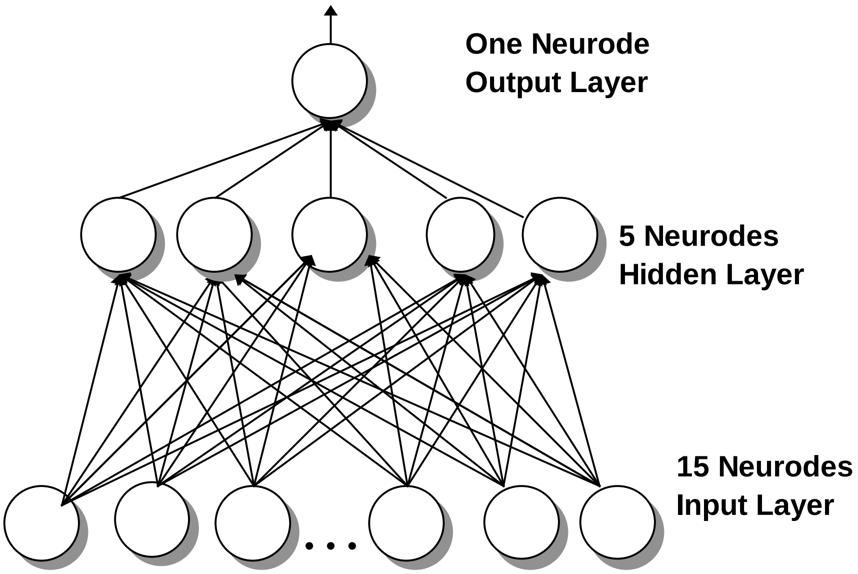
Random noise is used to perturb the error surface of the neural net to jolt it out of local minima. It also helps the ANN to generalize and avoid curve fitting. No input noise is used in this study as good results were obtained without it. It could be that the data set itself may already be noisy.

4.5.4 Training and Testing Tolerances

The training and testing tolerances are similar to the cut-off point or the level of probability in determining into which category a Credit Union should fall. A 0.1 cutoff point is equal to a tolerance of 0.9. This means that any output values that fall within the 90% range of the target are considered correct. Thus when the target is 1, an output with any value greater than 0.1, indicating that the Credit Union is in Distress, is classified as correct.

The ANN topology can be seen in Figure 4-1.

Figure 4-1
The ANN Topology



4.6 Results

A summary of the overall accuracy of both models training (in-sample) data set and validation (out-of-sample) data set, as well as selected Credit Unions is displayed in a similar fashion to the Hall and Byron’s paper so as to allow for a direct comparison of the two models. The full results for all the Credit Unions (except for Credit Unions numbered as 1058, 1093, 1148 and 1158 that were too small) from both models are in Appendix B of this research.

In the tables below, the Type I errors are highlighted by the box shading and the Type II errors are highlighted by a plain background box. The accuracy of the models is computed by taking the percentage of the total number of correct classifications in both categories from the total number in both categories.

$$Accuracy = \frac{\sum Distress\ CUs\ classified\ as\ Distress\ +\ \sum Non\text{-}Distress\ CUs\ classified\ as\ Non\text{-}Distress}{\sum CUs}$$

where CUs = Credit Unions.

[Equation 4.1]

4.6.1 Training Set (In-sample) Discussion

The training set consists of the data for all quarters of 1989 to 1990. The summary results are shown in Table 4-3.

The cut-off point or level of probability that is used to categorize a Credit Union as “Distress” is varied to see the effect on the results. By decreasing the cut-off point, less Type I errors are committed; i.e. less misclassification of Credit Unions that were actually in distress as non-distress, with the tradeoff of more Type II errors being committed; i.e. non-distress Credit Unions being misclassified as in distress. This holds true for both the Probit and the ANN model.

Type I errors are lower in all cases of the ANN model though at the 0.1 level, the Type II Errors committed are marginally higher than the Probit model. The ANN model with a cut-off at 0.1 gives the lowest Type I error, committing only 10 misclassification of distress Credit Unions as non-distress, while the Probit model yielded 13 misclassifications. The tradeoff, however, is an increase in Type II errors to 145 misclassifications for the ANN model and 109 for the Probit model. This increase in Type II errors resulted in the Probit model reporting a higher total accuracy rate. The tradeoff of improving the Type I errors committed must be weighed against the increase in Type II errors. The optimal cut-off point for the ANN model in terms of the total number of Type I errors committed in both the in-sample and out-of-sample data is 0.1.

Table 4-3 Training Set Results

	Type I Error: Predicting Credit Unions in Distress as Non-Distress Type 2 Error: Predicting Credit Unions Not in Distress as Distress						
Probit Model In Sample Results	**PREDICTIONS**						
Actual Groups	**Distress= Pr(Distress)>0.5**		**Distress= Pr(Distress)>0.25**		**Distress= Pr(Distress)>0.1**		**Total**
	Non-Distress	*Distress*	*Non-Distress*	*Distress*	*Non-Distress*	*Distress*	
Non-Distress	1403	0	1382	21	1294	109	1403
Distress	37	9	21	25	13	33	46
Total	1440	9	1403	46	1307	142	1449
Accuracy	97.45%		97.10%		91.58%		
ANN Model In Sample Results	**PREDICTIONS**						
Actual Groups	**Distress= Pr(Distress)>0.5**		**Distress= Pr(Distress)>0.25**		**Distress= Pr(Distress)>0.1**		**Total**
	Non-Distress	*Distress*	*Non-Distress*	*Distress*	*Non-Distress*	*Distress*	
Non-Distress	1399	4	1376	27	1258	145	1403
Distress	22	24	15	31	10	36	46
Total	1421	28	1391	58	1268	181	1449
Accuracy	98.21%		97.10%		89.30%		

4.6.2 Validation Set (Out-of-sample) Result Comparison

The validation data set are all the quarterly data of 1991. The summary results are shown in Table 4-4.

The ANN model performed better in predicting correctly Credit Unions that were actually in distress. The Type I error committed by the ANN model is 10% lower than the Probit model. The Type II error in using the ANN model is a little over half a percent higher than the Probit model. However, there are no statistical differences in the results at $\alpha = 0.05$ in all cases.

Table 4-4: Validation Set Results

	Type I Error: Predicting Credit Unions in Distress as Non-Distress **Type 2 Error: Predicting Credit Unions Not in Distress as Distress**						
Probit ***Out of Sample***	**Prediction**			***ANN*** ***Out of Sample***	**Prediction**		
Actual Groups	**Distress=** **Pr(Distress)>0.1**		**Total**	**Actual Groups**	**Distress=** **Pr(Distress)>0.1**		**Total**
	Non-Distress	*Distress*			*Non-Distress*	*Distress*	
Non-Distress	631	44	675	*Non-Distress*	627	48	675
Distress	8	12	20	*Distress*	6	14	20
Total	639	56	695	**Total**	633	62	695
Accuracy	92.52%			**Accuracy**	92.23%		

4.6.3 Validation Set (Out-of-sample) Evaluation

Any output of greater than 0.1 from the ANN model classifies the Credit Union as in Distress; any values less than 0.1 classifies it as Non-Distress. The 0.1 cutoff value is chosen from observing the results in the in-sample data. It gives the least number of Type I errors with a marginal increase in Type II errors as was discussed earlier. The Probit model classifies a Credit Union as Distress if the conditional probability is greater than 0.1.

In the tables that follow, the output results of the training set from the ANN model and the fitted conditional probability values of the Probit model are shown for all the quarters of 1989 to 1990 as well as 1991 predicted values based on the financial ratios for the 1991 quarters. The actual status (ranging from 1 being non-distress to 5 being distress) of each Credit Union is shown together with the normalized status of 1 being distress and 0 being non-distress.

A direct comparison will now be made on the Credit Unions (except Credit Union 1148 due to its small size) that were highlighted in the Hall and Byron’s study.

4.6.3.1 Credit Unions under Direction/Notice in 1991

The ANN model performed as well as, or better than the Probit model in most of the cases here. The overall results, in terms of percentage correctly predicted, could be misleading as the models in most cases were able to predict a few quarters ahead that a credit union would be in distress. However in the overall reporting of the predictive accuracy of the models, the early warning signals would have shown up as Type II errors.

4.6.3.1.1 Credit Union 1023

This Credit Union has been under direction since 1989. The ANN model clearly shows that it has not resolved its problems yet and correctly predicts its distress status on all the relevant quarters. The Probit model fail to predict it to be in distress in the first two quarters of 1991 although it manage to get the other quarters correctly.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1023	8903	0.0622	0	0	0	8903	0.061
1023	8906	0.4895	1	0	1	8906	0.285
1023	8909	0.6929	1	1	1	8909	0.412
1023	8912	0.7401	1	1	1	8912	0.489
1023	9003	0.9468	1	1	1	9003	0.667
1023	9006	0.9850	1	1	1	9006	0.84
1023	9009	0.2813	1	1	1	9009	0.181
1023	9012	0.1250	1	1	1	9012	0.112
1023	9103	0.1635	1	1	0	9103	0.094
1023	9106	0.1235	1	1	0	9106	0.095
1023	9109	0.7649	1	1	1	9109	0.213
1023	9112	0.5606	1	1	1	9112	0.216

4.6.3.1.2 Credit Union 1149

None of the models are able to correctly predict that this Credit Union would be in distress from the third quarter of 1990 to the second quarter of 1991.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1149	8903	0.0041	0	0	0	8903	0
1149	8906	0.0004	0	0	0	8906	0
1149	8909	0.0037	0	0	0	8909	0.001
1149	8912	0.0050	0	0	0	8912	0.001
1149	9003	0.0101	0	0	0	9003	0.003
1149	9006	0.0012	0	0	0	9006	0
1149	9009	0.0280	0	1	0	9009	0.003
1149	9012	0.0154	0	1	0	9012	0.001
1149	9103	0.0237	0	1	0	9103	0.001
1149	9106	0.0029	0	1	0	9106	0.001
1149	9109	0.0058	0	0	0	9109	0
1149	9112	0.0067	0	0	0	9112	0

4.6.3.1.3 Credit Union 1061

Both the models classify this credit union as in distress from the late 1990 onwards although it was not put under direction till the first quarter of 1991. The Type 2 error committed by both models is not indicative of their usefulness as early predictors of financial distress in this case. This classification problem is discussed in greater detail under the further research section.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1061	8903	0.0158	0	0	0	8903	0.01
1061	8906	0.0051	0	0	0	8906	0.004
1061	8909	0.0979	0	0	0	8909	0.032
1061	8912	0.0264	0	0	0	8912	0.011
1061	9003	0.0576	0	0	0	9003	0.02
1061	9006	0.0568	0	0	0	9006	0.042
1061	9009	0.1631	1	0	0	9009	0.071
1061	9012	0.1920	1	0	1	9012	0.35
1061	9103	0.4856	1	1	1	9103	0.774

4.6.3.1.4 Credit Union 1062

All the models predict from the outset that this Credit Union was in distress. However it was only put under direction in the fourth quarter of 1991. Again in this case, the Type 2 errors made by the models are not consistent with their early warning predictive capability.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1062	8903	0.1953	1	0	0	8903	0.089
1062	8906	0.1524	1	0	1	8906	0.128
1062	8909	0.2898	1	0	1	8909	0.137
1062	8912	0.3477	1	0	1	8912	0.198
1062	9003	0.5972	1	0	1	9003	0.246
1062	9006	0.8097	1	0	1	9006	0.459
1062	9009	0.2256	1	0	1	9009	0.116
1062	9012	0.1879	1	0	1	9012	0.117
1062	9103	0.3316	1	0	1	9103	0.207
1062	9106	0.4154	1	0	1	9106	0.195
1062	9109	0.3401	1	0	0	9109	0.085
1062	9112	0.3006	1	1	1	9112	0.151

4.6.3.1.5 Credit Union 1078

Both models fail to predict this credit union from being put under direction.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1078	8903	0.0139	0	0	0	8903	0
1078	8906	0.0134	0	0	0	8906	0
1078	8909	0.0140	0	0	0	8909	0
1078	8912	0.0137	0	0	0	8912	0
1078	9003	0.0134	0	0	0	9003	0
1078	9006	0.0135	0	0	0	9006	0
1078	9009	0.0136	0	0	0	9009	0
1078	9012	0.0148	0	0	0	9012	0
1078	9103	0.0150	0	1	0	9103	0
1078	9106	0.0185	0	1	0	9106	0.001

4.6.3.1.6 Credit Union 1153

The ANN model provided an early warning of distress in the first quarter of 1990 though the warning waned in the second quarter. However stronger signals were given from the third quarter of 1990 onwards. The Probit model predicted distress in the late 1990 but failed to predict the distress in the first quarter of 1991 when the credit union was put under direction.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1153	8903	0.0401	0	0	0	8903	0.017
1153	8906	0.0338	0	0	0	8906	0.028
1153	8909	0.0665	0	0	0	8909	0.055
1153	8912	0.0484	0	0	0	8912	0.053
1153	9003	0.1172	1	0	0	9003	0.072
1153	9006	0.0494	0	0	0	9006	0.083
1153	9009	0.1796	1	0	1	9009	0.129
1153	9012	0.1367	1	0	1	9012	0.109
1153	9103	0.1978	1	1	0	9103	0.1
1153	9106	0.9782	1	1	1	9106	0.994
1153	9109	0.6603	1	1	1	9109	0.849
1153	9112	0.3328	1	1	1	9112	0.567

4.6.3.1.7 Credit Union 1174

The ANN model committed a Type 2 error in the first quarter of 1991 though its predictions agreed with the Probit predictions of no problems in the other quarters of 1991.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1174	8903	0.7985	1	1	1	8903	0.42
1174	8906	0.6926	1	1	1	8906	0.318
1174	8909	0.8491	1	1	1	8909	0.607
1174	8912	0.7088	1	1	1	8912	0.441
1174	9003	0.6450	1	1	1	9003	0.297
1174	9006	0.3613	1	1	1	9006	0.221
1174	9009	0.3089	1	1	1	9009	0.268
1174	9012	0.2415	1	1	1	9012	0.114
1174	9103	0.1501	1	0	0	9103	0.024
1174	9106	0.0409	0	0	0	9106	0.009
1174	9109	0.0936	0	0	0	9109	0.084
1174	9112	0.0787	0	0	0	9112	0.024

4.6.3.2 Credit Union Transferring in 1991

None of the models seems to be able to predict voluntary transfer. The reason for this could be that the actual status of the voluntary transfer credit unions was classified as non-distress since they had a status score of 2. If an actual status of higher than one was used to categorize the credit unions as in distress, the predictive ability of the models on this type of credit unions should improve.

4.6.3.2.1 Credit Union 1002

None of the models predict any problems with this credit union. This credit union was a voluntary transfer in early 1992.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1002	8903	0.0000	0	0	0	8903	0
1002	8906	0.0000	0	0	0	8906	0
1002	8909	0.0000	0	0	0	8909	0
1002	8912	0.0000	0	0	0	8912	0
1002	9003	0.0000	0	0	0	9003	0
1002	9006	0.0000	0	0	0	9006	0
1002	9009	0.0000	0	0	0	9009	0
1002	9012	0.0000	0	0	0	9012	0
1002	9103	0.0000	0	0	0	9103	0
1002	9106	0.0000	0	0	0	9106	0
1002	9109	0.0001	0	0	0	9109	0
1002	9112	0.0001	0	0	0	9112	0

4.6.3.2.2 Credit Union 1071

This credit union was a voluntary transfer in the third quarter of 1991. None of the models indicated any problems with it.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1071	8903	0.0171	0	0	0	8903	0
1071	8906	0.0031	0	0	0	8906	0
1071	8909	0.0028	0	0	0	8909	0
1071	8912	0.0018	0	0	0	8912	0
1071	9003	0.0005	0	0	0	9003	0
1071	9006	0.0002	0	0	0	9006	0
1071	9009	0.0005	0	0	0	9009	0
1071	9012	0.0005	0	0	0	9012	0
1071	9103	0.0003	0	0	0	9103	0
1071	9106	0.0002	0	0	0	9106	0

4.6.3.2.3 Credit Union 1150

None of the models predict any problems with this credit union which subsequently became a voluntary transfer in the second quarter of 1991. The Probit model however did manage to give a very weak signal in the third quarter of 1990 that the credit union could be in distress.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1150	8903	0.0562	0	0	0	8903	0.008
1150	8906	0.0059	0	0	0	8906	0.011
1150	8909	0.0872	0	0	0	8909	0.056
1150	8912	0.0496	0	0	0	8912	0.059
1150	9003	0.0828	0	0	0	9003	0.063
1150	9006	0.0841	0	0	1	9006	0.113
1150	9009	0.0159	0	0	0	9009	0.019
1150	9012	0.0157	0	0	0	9012	0.019
1150	9103	0.0081	0	0	0	9103	0.008
1150	9106	0.0040	0	0	0	9106	0.008

4.6.3.2.4 Credit Union 1190

Again none of the models give any indication of problems with this credit union which was a voluntary transfer in the second quarter of 1991.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1190	8903	0.0057	0	0	0	8903	0.008
1190	8906	0.0087	0	0	0	8906	0.027
1190	8909	0.0069	0	0	0	8909	0.011
1190	8912	0.0051	0	0	0	8912	0.01
1190	9003	0.0069	0	0	0	9003	0.011
1190	9006	0.0033	0	0	0	9006	0.016
1190	9009	0.0101	0	0	0	9009	0.011
1190	9012	0.0038	0	0	0	9012	0.005
1190	9103	0.0016	0	0	0	9103	0.002

4.6.3.3 Credit Unions with Predicted Problems in 1991

4.6.3.3.1 Credit Union 1013

This credit union came out of direction in the first quarter of 1991 after being put in direction during 1990. The Probit model seems to indicate that the direction may have been lifted too early which the ANN model seems to agree with except for the last quarter of 1991.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1013	8903	0.4031	1	0	1	8903	0.195
1013	8906	0.3883	1	0	1	8906	0.278
1013	8909	0.4678	1	0	1	8909	0.254
1013	8912	0.4810	1	0	1	8912	0.26
1013	9003	0.8718	1	1	1	9003	0.483
1013	9006	0.9742	1	1	1	9006	0.793
1013	9009	0.8493	1	1	1	9009	0.544
1013	9012	0.8532	1	1	1	9012	0.635
1013	9103	0.8979	1	0	1	9103	0.653
1013	9106	0.7730	1	0	1	9106	0.571
1013	9109	0.1377	1	0	1	9109	0.215
1013	9112	0.0775	0	0	0	9112	0.075

4.6.3.3.2 Credit Union 1025

Both models seem to indicate problems with this credit union since the first quarter of 1990.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1025	8903	0.0140	0	0	0	8903	0.001
1025	8906	0.0027	0	0	0	8906	0
1025	8909	0.0152	0	0	0	8909	0.007
1025	8912	0.0471	0	0	0	8912	0.01
1025	9003	0.1037	1	0	0	9003	0.018
1025	9006	0.3713	1	0	1	9006	0.138
1025	9009	0.2483	1	0	1	9009	0.14
1025	9012	0.3402	1	0	1	9012	0.262
1025	9103	0.4763	1	0	1	9103	0.281
1025	9106	0.5262	1	0	1	9106	0.266
1025	9109	0.2241	1	0	1	9109	0.167
1025	9112	0.2738	1	0	1	9112	0.175

4.6.3.3.3 Credit Union 1044

The ANN model seems to agree with the weak Probit model signal that this credit union may have some potential problems.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1044	8903	0.0188	0	0	0	8903	0.004
1044	8906	0.0411	0	0	0	8906	0.038
1044	8909	0.0585	0	0	0	8909	0.029
1044	8912	0.1317	1	0	0	8912	0.041
1044	9003	0.1117	1	0	0	9003	0.026
1044	9006	0.0434	0	0	0	9006	0.051
1044	9009	0.1401	1	0	0	9009	0.07
1044	9012	0.2357	1	0	0	9012	0.095
1044	9103	0.2998	1	0	0	9103	0.096
1044	9106	0.2789	1	0	1	9106	0.142
1044	9109	0.1438	1	0	1	9109	0.117
1044	9112	0.1252	1	0	0	9112	0.076

4.6.3.3.4 Credit Union 1052

Both models indicate potential problems with this credit union from the first quarter of 1989.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1052	8903	0.3865	1	0	1	8903	0.211
1052	8906	0.4993	1	0	1	8906	0.318
1052	8909	0.1777	1	0	1	8909	0.129
1052	8912	0.1357	1	0	0	8912	0.057
1052	9003	0.1926	1	0	0	9003	0.063
1052	9006	0.5066	1	0	1	9006	0.25
1052	9009	0.1915	1	0	1	9009	0.13
1052	9012	0.4002	1	0	1	9012	0.196
1052	9103	0.6107	1	0	1	9103	0.243
1052	9106	0.7639	1	0	1	9106	0.445
1052	9109	0.4644	1	0	1	9109	0.278
1052	9112	0.3862	1	0	1	9112	0.208

4.6.3.3.5 Credit Union 1056

The models are in agreement here with potential problems for this credit union.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1056	8903	0.2163	1	0	0	8903	0.095
1056	8906	0.0690	0	0	0	8906	0.083
1056	8909	0.0518	0	0	0	8909	0.076
1056	8912	0.2690	1	0	1	8912	0.12
1056	9003	0.1405	1	0	0	9003	0.061
1056	9006	0.2893	1	0	1	9006	0.155
1056	9009	0.1652	1	0	1	9009	0.145
1056	9012	0.2168	1	0	1	9012	0.102
1056	9103	0.5385	1	0	1	9103	0.201
1056	9106	0.6372	1	0	1	9106	0.372
1056	9109	0.2212	1	0	1	9109	0.179
1056	9112	0.2050	1	0	1	9112	0.103

4.6.3.3.6 Credit Union 1169

The Probit indicates potential problems with this Credit Union in 1991. Hall and Byron postulated in their paper that the high conditional probabilities may be caused by distinct seasonal patterns in some of the financial ratios of this credit union. The ANN model seems to have captured this seasonal pattern with its output that seems to predict problems on the third and fourth quarter of every year except for the last two quarters of 1991.

ANN Model					PROBIT Model		
		ANN	ANN	Actual	Probit		Probit
Identity	Quarter	Output	Predicted	Status	Predicted	Quarter	Probability
1169	8903	0.0171	0	0	0	8903	0.01
1169	8906	0.0221	0	0	0	8906	0.021
1169	8909	0.3215	1	0	1	8909	0.445
1169	8912	0.2130	1	0	1	8912	0.404
1169	9003	0.0200	0	0	0	9003	0.009
1169	9006	0.0167	0	0	0	9006	0.018
1169	9009	0.1997	1	0	1	9009	0.399
1169	9012	0.2462	1	0	1	9012	0.383
1169	9103	0.0114	0	0	0	9103	0.005
1169	9106	0.0101	0	0	0	9106	0.006
1169	9109	0.0996	0	0	1	9109	0.198
1169	9112	0.0886	0	0	1	9112	0.189

4.6.4 Result Summary of Type I and Type II Errors

The ANN overall Type I error for the entire data set is 16 vs. 21 for the Probit model out of the 66 distress credit unions. The Type II error committed by the ANN model over the entire data set is 193 vs. 153 for the Probit model. The breakdown of Type I and Type II errors for both models in the training and validation sets are shown in Table 4-5.

Table 4-5 Type I and Type II Errors

Type of Error	**Type I**		**Type II**	
Data Set	**ANN Model**	**Probit Model**	**ANN Model**	**Probit Model**
Training Set	10	13	145	109
Validation Set	6	8	48	44

The ANN model is marginally superior (7.5% better) to the Probit scores method in terms of the fewest number of Type I errors committed. The Type II errors that the ANN model committed are only 1.8% worse in terms of the number of Type II errors committed. Therefore it may be a worthwhile tradeoff in using the ANN model over the Probit model.

4.7 Assumptions and Limitation of Methodology

The major assumption made in this research is the assumption of the accuracy and integrity of the historical data. The reliability of the models are dependent on this assumption as the data are used to construct the models. This methodology also assumes that future data are just as reliable.

The performances of the models are largely dependent on honest reporting by the credit unions. Therefore, they are vulnerable to fraudulent reporting. Some of the historical data used contain fraudulent reporting by credit unions that subsequently went into financial distress. However, the data provider is unwilling to identify nor provide this information due to confidentiality reasons. It is very likely that these credit unions are the ones that both models are unable to detect; i. e., the Type I errors.

4.8 Conclusion

The ANN model has been demonstrated to perform as well and in some cases better than the Probit model as an early warning model for predicting Credit Unions in distress. The overall accuracy of the ANN model vs. the Probit model is almost the same at around 90% for the ‘in’ sample data and 92% for the out-of-sample data. The results of the two models are not statistically significant different at $\alpha = 0.05$.

However, care should be taken in interpreting the accuracy results as explained in earlier sections that the Type II errors (predicting a Credit Union in distress when it is not) may actually be an early warning indicator of problems that do not surface until later quarters. Therefore the results from the models may actually be better than those reflected in the overall accuracy. A better benchmark would be the model with the fewest number of Type I errors.

The models provided early warning signals in many of the credit unions that eventually were in financial distress but were unjustifiably penalized with Type II errors due to the classification technique employed by Hall and Byron. In their technique, a credit union was classified in distress only after it has been put under direction or under notice of direction. This has a severe effect on ANNs as they learn through mistakes and being told that predicting a credit union in distress when the supervisors have not put it under direction or notice of direction is wrong even though it actually goes into financial distress in the near future. As a result in the ANN will build a suboptimal model that cannot, by design, provide early warning. This may hold true for the Probit model too. The data set needs to be reconstructed in future studies so that credit unions that failed in *n* number of quarters will be classified as in potential distress in order to allow for *n* number of quarters forecast.

One of the elements that seems to be vital to this type of research but missing from the models in this study is the temporal effect of the independent variables. The temporal effect of the financial data time series was ignored because this study is meant to be a comparison with Hall and Byron’s work which did not use any time-dependent variables. They state in their paper that they find no significance in the one period change of any of the financial ratios. The models constructed thus are severely restricted in their time horizon forecast which are only able to predict financial distress for the quarter that financial ratios are obtained. This seems to be contrary to the objective of achieving an early predictor system.

4.9 Managerial and Implementation Issues

This problem highlights the potential difficulty in getting a new technology accepted in an establishment that has been running without it. In this case, potential problems may arise with the intended users, which are the supervisors of regulatory bodies that oversee the credit unions. If the ANN predicts a few potential credit unions that may go into distress of which a supervisor may disagree with, based upon his/her personal experiences, and thus choose to ignore the ANN warning, he/she does so on his/her own peril! If the predicted credit union does go into distress, he/she would have to answer to his/her superiors for ignoring the warning. On the other hand, the ANN model may be seen to be an additional burden for the supervisors to shoulder. In addition, many of its predictions may be false warning and to err on the side of caution, the supervisors may require additional resources to be expended. The supervisor has to bear the ultimate responsibility for any decisions made and thus may have a difficult time in deciding whether to trust a computer generated opinion or not.

The ethical issue of auditing healthy credit unions due to the false warnings also needs to be addressed. Before the system is implemented, all the credit unions under the system, should be made aware of the limitation of the system and that is can commit Type II errors, and they may be targeted for audit by the supervisory board even if they are not aware of any financial problems with credit unions. People should be made aware of the limitations of the system so that their expectations will not be too high. The security and accessibility of the information provided by the credit unions also need to be addressed, in order to minimize a breach of confidentiality as the system will result in more people having access to potentially sensitive information.

A cost-benefit analysis needs to be conducted to determine if the implementation of an early warning predictor system is indeed viable. The projected cost of Type II errors need to be considered with the potential gain in terms of both monetary gain (from prevention of a credit union going under) and public confidence. The cost of extra resources required to implement the system will have to be justified. The personnel resources required include a team of system builders, system maintenance personnel and additional monitoring and auditing staff (in anticipation of an increase in the number of credit unions audit due to Type II errors). Other resources required will include computer equipment, the design and drafting of new compliance rules for the credit unions, staff training, integrating the system with existing information systems and a facility to house the new department.

A prototype of the system may need to be constructed to demonstrate to the management the tangible benefits that can be derived from full implementation of the system. and to convince them to commit resources to the project. It is important to gain acceptance from management and also the people who will be working with it. Constructing a prototype will also provide the system builders with experience that will be valuable in the actual full implementation of the project.

4.10 Future Research

Future research will concentrate on the time series component of the financial data including perhaps new ratios such as growth in assets and liabilities. ANNs have been applied to many time series problems such as weather forecasting, financial market forecasting (See Tan [1993b], Tsoi, Tan and Lawrence[1993]), EEG signal analysis to detect mental illness and ECG signal analysis to predict heart attacks, etc. Applying ANNs to financial distress prediction from a time series context may provide better results by potentially providing earlier warnings to financial distress. The data selection is the most vital and time consuming process. New financial ratios that represent the time series component of the financial data need to be developed.

However, there may not be sufficient data to conduct a time series analysis, as data collection of the credit unions did not start until the late 1980s. Furthermore, due to the deadline required for submission of the information by credit unions, many of the data gathered are erroneous and have to be corrected in the subsequent quarters. The frequency of the data may also pose a problem as currently the credit unions are only required to submit quarterly reports.

Most studies just report the accuracy of the models in terms of percentage correctly classified without regard to the difference in the cost of error as pointed out by Pacey and Pham [1990]. The Type I errors tend to be more serious as failing to provide early warning to a credit union that is in financial distress and this could prove to be a very costly affair as observed in the US S & L failures. The costs of Type II errors are normally restricted to loss of extra labor or resources in auditing or analyzing the credit unions. Therefore, it would be prudent in future studies to use the minimization of Type I error as the objective function of the models. This is assuming that the Type II errors are kept at acceptable rate.

The Hall and Byron method of classifying the Credit Unions also does not allow for voluntary transfers and mergers of credit unions to be built into the models. It may turn out that the voluntary transfers could be a result of potential financial distress in a credit union. Closer examination of those credit unions will need to be conducted to determine if there are any common characteristics that may provide valuable information in predicting financial distress.

The ten largest conditional probabilities of both models for each quarter of 1991 are provided in Appendix A. The only credit union from the Hall and Byron study that was missing from the table is Credit Union number 1148 which was omitted from this study due to its small size. The appendix will be used in future research to analyze the relationship of the ratios to the ANN model output.

Since one of the major weaknesses of ANNs is the difficulty in explaining the model, future research will concentrate on studying the interaction of the input variables in relation to the outputs as well as the associated weights of the networks' structures. The ANN parametric effects on the result will be studied in a similar method used by Tan and Wittig [1993] in their parametric study of a stock market prediction model. Sensitivity analysis on input variables, similar to those performed by Poh [1994], can be conducted to determine the effect each of the financial ratios have on the financial health of the credit unions.

Different types of artificial neural networks such as the Kohonen type of network will be constructed to see if the results can be improved. The Kohonen network has been used by Prof. A. C. Tsoi of the University of Queensland quite successfully in predicting medical claims fraud.

The utilization of genetic algorithms to select the most optimal ANN topology and parameter setting will be explored in future research. Hybrid type of models discussed by Wong and Tan [1994], incorporating ANNs with fuzzy logic and/or expert systems will also be constructed in future to see if the results can be improved. The benefits of incorporating ANNs with rule-based expert systems as proposed by Tan [1993a] for a trading system will be examined to see if the same concept can be implemented in the context of financial distress prediction of credit unions.

4.11 References

2. Altman, E, Haldeman R. and Narayanan, P., “Zeta Analysis”, *Journal of Banking and Finance*, pp. 29054, June 1977.

3. Altman, E., Financial ratios, “Discriminant Analysis and the Prediction of Corporate Bankruptcy”, *Journal of Finance*, pp. 589-609, September 1968.

4. Back, B., Laitinen, T. and Sere, K., “Neural Networks and Bankruptcy Prediction: Fund Flows, Accrual Ratios, and Accounting Data”, *Advances In Accounting*, ISBN: 0-7623-0161-9, Vol. 14, pp. 23-37, 1996.

5. Beaver, W., “Financial Ratios as Predictors of Failure”, Journal of Accounting Research, pp. 71-111, 1966.

6. Bell, T. B., G. S. Ribar and J. R. Verchio, “Neural Networks vs. Logistic Regression: A Comparison of Each Model’s Ability to Predict Commercial Bank Failures”, *Deloitte & Touche/University of Kansas Auditing Symposium*, May 1990.

7. Brockett, P. L., Cooper, W. W., Golden, L. L. and Pitakong, U., “A Neural Network Method for Obtaining an Early Warning of Insurer Insolvency”, *Journal of Risk and Insurance*, Vol. 61, No. 3, pp. 402-424, 1994.

8. Coleman, K. G., Timothy J. Graettinger and William F. Lawrence, “Neural Networks for Bankruptcy Prediction: The Power To Solve Financial Problems”, *AI Review*, pp. 48-50, July/August 1991.

9. Collins, R. A. and R. D. Green, “Statistical Methods for Bankruptcy Forecasting”, *Journal of Economic and Business*, vol. 32, pp. 349-354, 1982.

10. Dambolena, I. and Khoury, S., “Ratio Stability and Corporate Failure”, *Journal of Finance*, pp. 1017-26, September 1980.

11. Gentry, J., Newbold, P. and Whitford, D., “Bankruptcy: If Cash Flow’s Not the Bottom Line, What Is?”, Financial Analysts Journal (September/October), pp. 17-56, 1985b.

12. Gentry, J., Newbold, P., Whitford, D., “Classifying Bankrupt Firms with Fund Flow Components”, *Journal of Accounting Research (Spring)*, pp. 146-59, 1985a.

13. Gibson, C. H., Frishkoff, P. A., *Financial Statement Analysis: Using Financial Accounting Information*, 3rd. ed., Kent Publishing Company, Boston, 1986.

14. Gombola, M., and Ketz, J., “Note on Cash Flow and Classification Patterns of Financial Ratios”, *The Accounting Review*, pp. 105-114, January 1983.

15. Hall, A. D. and Byron, R., “An Early Warning Predictor for Credit Union Financial Distress”, *Unpublished Manuscript for the Australian Financial Institution Commission.*

16. Horrigan, J. O., "A Short History of Financial Ration Analysis", *The Accounting Review*, vol. 43, pp. 284-294, April 1968.

17. Jones, F., "Current Techniques in Bankruptcy Prediction", *Journal of Accounting Literature*, vol. 6, pp. 131-164, 1987.

18. Levy, H. and Sarnat, M., "Caveat Emptor: Limitations of Ratio Analysis", *Principles of Financial Management*, Prentice Hall International, pp. 76-77, 1989.

19. Libby, R., "Accounting ratios and the Prediction of Failure: Some Behavioral Evidence", *Journal of Accounting Research (Spring)*, pp. 150-61, 1975.

20. Maclachlan, I., "Early Warning of Depository Institutions Distress: A Study of Victorian Credit Unions", *Master of Commerce Thesis*, University of Melbourne, December 1993.

21. Martin, D., "Early Warning of Bank Failure-A Logit Regression Approach", *Journal of Banking and Finance*, vol. 1, pp. 249-276, 1977.

22. McKinley, J. E., R. L Johnson, G.R Downey Jr., C. S. Zimmerman and M. D. Bloom, "Analyzing Financial Statements", *American Bankers Association*, Washington, 1983.

23. Odom, M. D., & Ramesh Sharda, "A Neural Network Model for Bankruptcy Prediction", *Proceedings of the IEEE International Conference on Neural Networks, pp. II163-II168, San Diego, CA, USA*, June 1990.

24. Ohlson, J., "Financial ratios and Probabilistic Prediction of Bankruptcy", *Journal of Accounting Research (Spring)*, pp. 109-31, 1980.

25. Pacey, J., Pham, T., "The Predictiveness of Bankruptcy Models: Methodological Problems and Evidence", *Journal of Management*, 15, 2, pp. 315-337, December 1990.

Poh, H. L., "A Neural Network Approach for Decision Support", *International Journal of Applied Expert Systems*, Vol. 2, No. 3, 1994.

26. Rowe, A. J., Mason, R. O., Dickel, K. E., Mann, R. B., and Mockler, R. J., *Strategic Management: A Methodological Approach 4th Edition*, Addison-Wesley Publishing Company, USA, 1994.

27. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge Mass., 1986.

28. Salchenberger, L. M., E. M. Cinar and N. A. Lash, "Neural Networks: A New Tool For Predicting Thrift Failures", *Decision Sciences*, Vol. 23, No. 4, pp. 899-916, July/August 1992.

29. Shepanski, A., "Test of Theories of Information Processing Behavior in Credit Judgment", *Accounting Review*, 58, pp. 581-599, 1983.

30. Sinkey, J., "A Multivariate Statistical Analysis of the Characteristics of Problem Banks", *Journal of Finance,* vol. 30, no. 1, pp. 21-36, March 1975.

31. Tam, K. Y. and M. Y. Kiang, "Managerial Applications of Neural Networks: The Case of Bank Failure Predictions", *Management Science,* Vol. 38, No. 7, pp. 926-947, July 1992.

32. Tan, C.N.W. and Wittig, G. E., "Parametric Variation Experimentation on a Back-propagation Stock Price Prediction Model", *The First Australia and New Zealand Intelligent Information System (ANZIIS) Conference*, University of Western Australia, Perth, Australia, December 1-3, 1993, IEEE Western Australia Press, 1993.

33. Tan, C.N.W., "Incorporating Artificial Neural Network into a Rule-based Financial Trading System", *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES),* University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993a.

34. Tan, C.N.W., "Trading a NYSE-Stock with a Simple Artificial Neural Network-based Financial Trading System", *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES),* University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993b

35. Whitred, G. and Zimmer, I., "The Implications of Distress Prediction Models for Corporate Lending", *Accounting and Finance,* 25, pp. 1-13, 1985.

36. Wong, F. and Tan, C., "Hybrid Neural, Genetic and Fuzzy Systems", *Trading On The Edge: Neural Genetic and Fuzzy Systems for Chaotic Financial Markets*, John Wiley and Sons Inc., pp. 243-261, 1994.

4.12 Appendix A: The 10 largest predicted conditional probabilities.

The ten largest conditional probabilities of both the ANN and the Probit model for each quarter of 1991 are provided in this appendix. These are the credit unions which the models predict are most likely to go into distress. Although many of the credit unions here have not gone into distress at the time of writing, it would be of interest to continue monitoring these credit unions, as the models may be giving advanced warning on them. Future research will involve analyzing these results with the financial ratios being used. A detailed comparison analysis of results from both models may reveal some useful information that can be used in improving the early detection of financial distress models.

Table 4-6
The 10 largest predicted conditional probabilities

ANN Model					Probit Model			
Identity	Quarter	Output	Predicted	Actual	Predicted	Quarter	Probability	Identity
1044	9103	0.2998	1	0	0	9103	0.096	1044
1062	9103	0.3316	1	0	0	9103	0.1	1153
1127	9103	0.3588	1	0	1	9103	0.201	1056
1147	9103	0.4613	1	0	1	9103	0.207	1062
1025	9103	0.4763	1	0	1	9103	0.243	1052
1061	9103	0.4856	1	1	1	9103	0.281	1025
1056	9103	0.5385	1	0	1	9103	0.361	1147
1052	9103	0.6107	1	0	1	9103	0.653	1013
1013	9103	0.8979	1	0	1	9103	0.774	1061
1129	9103	0.9979	1	1	1	9103	0.884	1129
1069	9106	0.2863	1	0	1	9106	0.111	1005
1127	9106	0.3052	1	0	1	9106	0.142	1044
1147	9106	0.3344	1	0	1	9106	0.195	1062
1062	9106	0.4154	1	0	1	9106	0.266	1025
1025	9106	0.5262	1	0	1	9106	0.372	1056
1056	9106	0.6372	1	0	1	9106	0.429	1147
1052	9106	0.7639	1	0	1	9106	0.445	1052
1013	9106	0.7730	1	0	1	9106	0.571	1013
1153	9106	0.9782	1	1	1	9106	0.795	1129
1129	9106	0.9957	1	1	1	9106	0.994	1153
1191	9109	0.1845	1	0	1	9109	0.164	1191
1194	9109	0.1855	1	0	1	9109	0.167	1025
1110	9109	0.2157	1	0	1	9109	0.179	1056
1056	9109	0.2212	1	0	1	9109	0.187	1147
1025	9109	0.2241	1	0	1	9109	0.198	1169
1062	9109	0.3401	1	0	1	9109	0.213	1023
1052	9109	0.4644	1	0	1	9109	0.215	1013
1153	9109	0.6603	1	1	1	9109	0.278	1052
1023	9109	0.7649	1	1	1	9109	0.849	1153
1129	9109	0.9993	1	1	1	9109	0.998	1129
1005	9112	0.1400	1	0	0	9112	0.085	1005
1127	9112	0.1782	1	0	1	9112	0.103	1056
1056	9112	0.2050	1	0	1	9112	0.111	1191
1069	9112	0.2677	1	0	1	9112	0.151	1062
1025	9112	0.2738	1	0	1	9112	0.175	1025
1062	9112	0.3006	1	1	1	9112	0.189	1169
1153	9112	0.3328	1	1	1	9112	0.208	1052
1052	9112	0.3862	1	0	1	9112	0.216	1023
1023	9112	0.5606	1	1	1	9112	0.567	1153
1129	9112	0.9989	1	1	1	9112	0.997	1129

4.13 Appendix B: In-sample and Out-of-sample Results

This appendix contains the results of both the models for the 1989 to 1991 quarters of all the Credit Unions except for Credit Unions numbered 1058,1093,1148 and 1158. They were excluded entirely since they were not used in the estimation sample of Hall and Byron's study due to their small size. The prediction of the Probit model incorporate both the conditional probabilities of the Probit scores model. The prediction of the Probit model is 1 if the conditional probability of the Probit model is greater than 0.1 and 0 if not. The ANN model prediction is 1 if the output from the ANN is greater than 0.1 and 0 if not.

Tables Withheld Due to Non-Disclosure Agreement with the Australian Financial Institution Commission.

Chapter 5: Applying Artificial Neural Networks in Finance: A Foreign Exchange Market Trading System Example with Transactions Costs

"There is no sphere of human thought in which it is easier to show superficial cleverness and the appearance of superior wisdom than in discussing questions of currency and exchange",
Sir Winston Churchill at the House of Commons, September 28, 1949.

5. Applying Artificial Neural Networks in Finance: A Foreign Exchange Market Trading System Example with Transactions Costs

5.1 Introduction

This chapter focuses on the application of Artificial Neural Networks (ANNs) to financial trading systems. A growing number of studies have reported success in using ANNs in financial forecasting and trading[33]. In many cases, however, transaction costs and, in the case of foreign exchange, interest differentials, have not been taken into account. Attempts are made to address some of these shortcomings by adding the interest differentials and transaction costs to the trading system in order to produce a more realistic simulation.

The complexity and problems encountered in designing and testing ANN-based foreign exchange trading systems as well as the performance metrics used in the comparison of profitable trading systems are discussed. The idea of incorporating ANNs into a rule-based trading system has been raised in earlier work by the author [Tan 1993a, Wong and Tan 1994]. The particular trading system used in this chapter is based on an earlier model constructed by the author and published in the proceedings of the ANNES '93 conference [Tan 1993b]. The system uses ANN models to forecast the weekly closing Australian/US dollar exchange rate from a given set of weekly data. The forecasts are then passed through a rule-based system to determine the trading signal. The model generates a signal of either 'buy', 'sell' or 'do nothing' and the weekly profit or loss is computed from the simulated trading based on the signals. The various attitudes towards risk is also approximated by applying a range of simple filter rules.

An appendix is provided at the end of this chapter which discusses the different foreign exchange trading techniques in use, including technical analysis, fundamental analysis and trading systems.

This chapter builds on another earlier study by the author that was reported at the TIMS/INFORMS '95 conference at Singapore in June 1995 [Tan 1995a] and the Ph.D. Economics Conference at Perth in December 1995 [Tan 1995b]. It introduces the idea of a simple hybrid Australian/US dollar exchange rate forecasting/trading model, the ANNWAR, that incorporates an ANN model with the output from an autoregressive (AR) model. In the earlier study, initial tests find that a simple ANN-based trading system for the Australian/US dollar exchange rate market fail to outperform an AR-based trading system, thus resulting in the development of the hybrid ANNWAR model. The initial ANNWAR model results indicates that the ANNWAR-based trading system is more robust than either of the independent trading systems (that utilize the ANN and the AR models on their own). The earlier study however, uses a smaller data set and the advantage of the ANNWAR model over the simple ANN model in terms of returns alone is quite marginal.

Furthermore, the best ANNWAR and ANN models were simple linear models, thus casting doubt on the usefulness of ANNs' ability in solving non-linear problems. One of the reasons for that result may have been the nature of the out-of-sample (validation) data

See for example, . Widrow et al. [1994], Trippi and Turban [1996][33] set which was clearly in a linear downward trend. This property of the data may also have explained why the AR model outperformed the ANN model in the earlier study. In this chapter, the tests are repeated with additional data. The results from the larger data set show that the ANN and ANNWAR models clearly outperform the AR model. The ANNWAR model also clearly outperforms the ANN model.

For the rest of this book, the ANN model with the AR input is referred to as an ANNWAR while the ANN model without the AR is referred to as an ANNOAR. I will refer to both the ANNWAR and ANNOAR models collectively as ANN models in general if no differentiation is needed.

5.2 Literature Review

5.2.1 Development of the Australian Dollar Exchange Rate Market

A substantial amount of the research into foreign exchange markets has focused on the issue of market efficiency. If the *efficient market hypothesis* (EMH) holds, there should be no unexploited profit opportunities for traders. Thus, trading systems should not be able to generate excess returns in an efficient market. Section 5.2.2 discusses previous EMH studies in the foreign exchange market while section 5.2.3 and the Appendix 5.13 discusses trading systems in more detail.

Despite the move by the major currencies to floating exchange rates in 1973, after the breakdown of the Bretton Woods system of 1944, the Australian dollar was not floated until December 9, 1983. The Australian dollar was initially pegged to the US dollar until 1976 when it was moved to a crawling peg (based on a Trade Weighted Index (TWI) of currencies. The Government’s prime motivation in floating the Australian dollar in 1983 was the perceived inefficiency imposed on monetary policy by the crawling peg. By mid 1984, when all regulatory controls were removed from the “official market”, and with the introduction of currency futures on the Sydney Futures Exchange, the Australian foreign exchange market had developed into a free and competitive market.

The Reserve Bank of Australia [1996] reported that the latest survey conducted by the Bank for International Settlements [BIS95] put the total daily foreign exchange turnover at over US$1 trillion with the Australian Dollar accounting for 3 percent of the global turnover, up one percent from the previous survey conducted in April 1992. The survey also ranked the Australian dollar as the world’s eighth most actively traded currency, up one rank from the previous survey, displacing the European Currency Unit (ECU). This was despite a decrease in the percentage of Australian dollar trade conducted in the Australian market from 45% to 40%. This clearly shows that the increased activity in the Australian dollar market are mostly due to speculators, indicating a growing interest by international fund managers in holding Australian-dollar investments.

5.2.2 Studies of the Efficiency of the Australian Foreign Exchange Market

Most studies of foreign exchange market efficiency test the unbiasedness of the forward rate as a predictor of future spot exchange rates. Studies from the Pre-Float period (using quarterly data from September 1974 to June 1981) by Levis [1982] and Turnovsky and Ball [1983] on the efficiency of the exchange market generally support the efficiency hypothesis. According to Bourke [1993], however, the evidence should be viewed with caution because of inconsistency in their estimation procedures.

Studies from the Post-Float period generally provide a better test of the efficiency of the foreign exchange market. These studies also benefited from advances in econometric methodology. Using the weekly spot rates and forward rates of three maturities in the period up to January 1986, Tease [1988] find the market to be less efficient subsequent to the depreciation in February 1985. Kearney and MacDonald [1991] conduct a similar test on changes in the exchange rate, using data from January 1984 to March 1987. They conclude that the change in the spot foreign exchange rate does not follow a random walk[34] and that there was strong evidence for the existence of a time-varying risk premium.

Sheen [1989] estimate a structural model of the Australian dollar/US Dollar exchange rate and find some support for the argument that structural models are better predictors than a simple random walk model using weekly data from the first two years of the float. There have been other studies that apply multivariate cointegration techniques to the exchange rate markets, reporting results that do not support the efficient market hypothesis [see for example, Karfakis and Parikh 1994]. However, the results of these studies may be flawed as it has been shown that cointegration does not mean efficiency and vice-versa [See Dwyer and Wallace 1992 and Engel 1996].

5.2.3 Literature Review on Trading Systems and ANNs in Foreign Exchange

Despite the disappointing result from White's [1988] initial seminal work in using ANNs for financial forecasting with a share price example, research in this field has generated growing interest. Despite the increase in research activity in this area however, there are very few detailed publications of practical trading models. In part, this may be due to the fierce competition among financial trading houses to achieve marginal improvements in their trading strategies which can translate into huge profits and their consequent reluctance to reveal their trading systems and activities.

This reluctance notwithstanding, as reported by Dacorogna et al. [1994], a number of academicians have published papers on profitable trading strategies even when including transaction costs. These include studies by Brock et al. [1992], LeBaron [1992], Taylor and Allen [1992], Surajaras and Sweeney [1992] and Levitch and Thomas [1993].

From the ANN literature, work by Refenes et al.[1995], Abu-Mostafa [1995], Steiner et al.[1995], Freisleben [1992], Kimoto et al.[1990], Schoneburg [1990], all support the proposition that ANNs can outperform conventional statistical approaches. Weigend et al. [1992] find the predictions of their ANN model for forecasting the weekly Deutshmark/US Dollar closing exchange rate to be significantly better than chance. Pictet et. al. [1992] reports that their real -time trading models for foreign exchange rates returned close to 18% per annum with unleveraged positions and excluding any interest gains. Colin [1991] reports that Citibank's proprietary ANN-based foreign exchange trading models for the US Dollar/Yen and US Dollar/Swiss Franc foreign exchange market achieved simulated

[34] Random walk hypothesis states that the market is so efficient that any predictable fluctuations of price are eliminated thus making all price changes random. Malkiel [1973] defines the broad form of the random-walk theory as "Fundamental analysis of publicly available information cannot produce investment recommendations that will enable an investor consistently to outperform a buy-and-hold strategy in managing a portfolio. The random-walk theory does not, as some critics have claimed, state that stock prices move aimlessly and erratically and are insensitive to changes in fundamental information. On the contrary, the point of the random-walk theory is just the opposite: The market is so efficient — prices move so quickly when new information does arise — that no one can consistently buy or sell quickly enough to benefit".

trading profits in excess of 30% per annum and actual trading success rate of about 60% on a trade-by-trade basis. These studies add to the body of evidence contradicting the EMH.

5.3 Foundations of Trading Methodology

The motive for foreign exchange trading is profit. In order to profit from transactions, a trader must gain sufficiently from a transaction to cover not only the transaction costs involved, but also any opportunity cost or funding cost involved.

For example, in the professional currency market, spot transactions incur a bid/ask spread of around seven basis points. (or seven pips[35], i.e. 0.0007) in buying or selling Australian Dollar (AUD) in exchange for US Dollar (USD). Thus, if the reported spot exchange rate is .7950 (USD per AUD), the bid/ask prices are likely to be in the order of 0.7947 (bid) and 0.7954 (ask).

Based on these figures, a speculator expecting the AUD to depreciate against the USD, will sell, for example, 100 AUD in exchange for 79.47 USD. This purchase of USD must be funded (i.e. assuming that the speculator borrows the 100 AUD to deliver to the dealer buying AUD) at the going Australian interest rate, i_A. In similar fashion, the USD received will be invested in the USA at the going US interest rate, i_{US}. The interest rates are quoted on a per annum basis.

Funding transactions in the professional money markets also incur a bid/ask spread. This is typically around two basis points (i.e., 0.02%). Thus, if the reported Australian interest rate is 10.00%, the bid/ask lending rates are likely to be in the order 9.98% (bid) and 10.02% (ask). The speculator then has to borrow the 100 AUD at 10.02%. If the US interest rate quoted is 8%, the bid/ask quote for investment is likely to be 8.02%/7.98%. The speculator thus earns an interest rate of 7.98% on his/her 79.47 USD.

If the local rate is higher than the foreign interest rate, arbitrage in the forward exchange market resulting in the local currency trading at a *forward discount* to the foreign currency. If the converse is true, the local currency is said to trade at a *forward premium*. In the example above, the AUD will be at a forward discount to the USD.

Assuming the speculator forecasts correctly that the exchange rate in one week will be .7955. The trader will incur transaction costs in purchasing the USD, funding the purchase in AUD, investing the proceeds in USD and in converting the USD back to AUD in one week.

The ‘profit’ in AUD on this transaction would be:

[35] The term *pip* is used to describe the smallest unit quoted in the foreign exchange market for a particular currency. For example a pip in the US Dollar is equivalent to US$0.0001 or .01 of a cent while a pip in Yen is 0.01 Yen.

$$\left[ForeignExchange\,\mathrm{Pr}\,ofit\ /\ Loss \right] - \left[NetFundingCost \right]$$

Equation 5-1

$$\Rightarrow \left[100 - \left(\frac{100 \times .7947}{.7959} \right) \right] - \left[\left(\frac{10.02}{52 \times 100} \times 100 \right) - \left(\frac{7.98}{52x100} \times (100 \times .7947) \right) \right]$$

$$\Rightarrow 0.1507 - 0.0707 = 0.0800$$

The profit rate is $\frac{.08}{100}$ per week or an effective, annualized interest rate of 8.32%.

If the interest rate differential is wider, with the Australian interest rate at $i_A = 12\%$ and the US interest rate at $i_{US} = 5\%$, the result from this transaction will be a loss:

$$\left[ForeignExchange\,\mathrm{Pr}\,ofit\ /\ Loss \right] - \left[NetFundingCost \right]$$

$$\Rightarrow \left[100 - \left(\frac{100 \times .7947}{.7959} \right) \right] - \left[\left(\frac{12.02}{52 \times 100} \times 100 \right) - \left(\frac{4.98}{52x100} \times (100 \times .7947) \right) \right]$$

$$\Rightarrow 0.1507 - 0.1550 = -0.0043$$

Clearly, a correct forecast of depreciation is a necessary, but not a sufficient condition for the speculator to profit. The following factors will affect the profitability of a transaction:

As transaction costs increases, profits will decrease and vice-versa.

As the interest differential widens, the net funding cost will be higher and profits will decrease.

5.4 Data

5.4.1 ANN Data Sets: Training, Testing and Validation

The data used in this study were provided by the Reserve Bank of Australia. The data consist of the weekly closing price of the US dollar/Australian Dollar exchange rate in Sydney, the weekly Australian closing cash rate in Sydney and the weekly closing US Fed Fund rate in New York from 1 Jan 1986 to 14 June 1995 (495 observations).

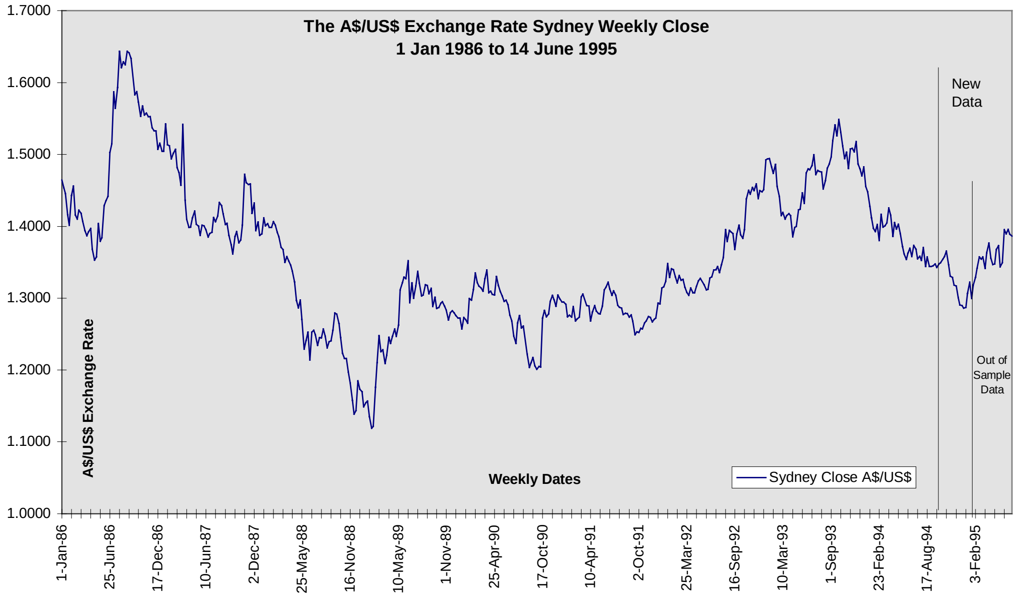
In the earlier study reported by the author [Tan 1995ab], the data used extend only from 1 January 1986 to 16 September 1994 (453 observations). The additional data used in this study are indicated to the right of the vertical line in Chart 1. This additional data has significantly improved the ANNs but has resulted in dismal performance by the AR. This is probably due to the higher volatility in this new data set and the less linear nature of the out-of-sample data. The previous out-of-sample data were clearly on a strong downtrend.

Of the total 495 observations for this study, the last 21 observations (27 January 1995 to 14 June 1995) are retained as out-of-sample data with the remaining 474 observations used as in-sample data set for both the ANN and the AR models[36].

[36] Note that in constructing the models, the last observation (14 June 1995) was used only in forecast comparison as a one step forecast. In addition, the first two observations were only used to generate lagged inputs before the first forecast on 15 January 1986.

In the case of the ANN, the observations in the in-sample data set are divided again into training and testing data sets. The first 469 observations are used as the training set to build the ANN model; the remaining 5 observations are used to determine a valid ANN model and to decide when to halt the training of the ANN[37]. Statistical, mathematical and technical analysis indicators such as the logarithmic values, stochastic oscillators, relative strength index and interest differentials, are derived from the original data set and used as additional inputs into the ANNs. Interestingly, the final ANN model disregards all the additional variables; the best ANN model uses only the closing price of the exchange rate and the AR output as input variables with a time window size of three periods.

Chart 1: The Australian Dollar/US Dollar Weekly Exchange Rate Data from 1 Jan 1986 to 15 June 1995



5.4.2 AR Data Set: In-sample and Out-of-Sample

The AR data set is divided into the in-sample (training + testing in the ANN) and out-of-sample (validation in the ANN) data sets. All the observations in the training and testing data sets are used to build the AR model, while the out-of-sample data set is the same as that used with the ANN models.

In the case of the AR model, only the previous two lagged observations are statistically significant as independent variables, i.e. x_{t-1} and x_{t-2}. The final AR model is therefore an AR(2) model that utilizes only two independent variables.

[37] The number of observations for the test set may seem small but this book uses an additional 21 observations for the out-of-sample data set for validation of the model. The purpose of this test set is mainly to determine when to stop the training of the ANN. This limitation will be alleviated as more observations are obtained. However, at the time of research for this book, the amount of data available was limited to the 495 observations.

5.5 Financial Trading System Structure

Figure 5-1 shows the entire trading system structure with the different processes involved. A more detailed discussion of each of the processes is provided in the sections below. The entire system is constructed on a Microsoft Excel 5.0/7.0 spreadsheet with the addition of the Neuralyst[38] 1.41, neural network program that runs as a macro in Excel.

The raw data discussed in the previous section are used as input to the trading system. The AR process uses the previous two exchange rates, x_{t-1} and x_{t-2} as inputs and generates an output, the AR forecast, y_t, one period ahead. The estimated AR model has the following parameters:

$y_t = 0.021359 + 0.\ 90107x_{t-1} + 0.\ 08297x_{t-2}$ with an r^2 of 0. 98364.

This output is passed on to the next process which is the ANN process. The ANN process uses the past three AR output; y_t, y_{t-1} and y_{t-2}; and the past three exchange rates, x_t, x_{t-1} and x_{t-2} as input and produces the following week's forecast, x^*_{t+1} as output. The ANN model and process is discussed in detail in section 5.6.

The computation of the interest rate differentials with transaction costs which is discussed in detailed in section 5.5.2, determines arbitrage boundaries (explained in the next section) in the decision-making process for trading.

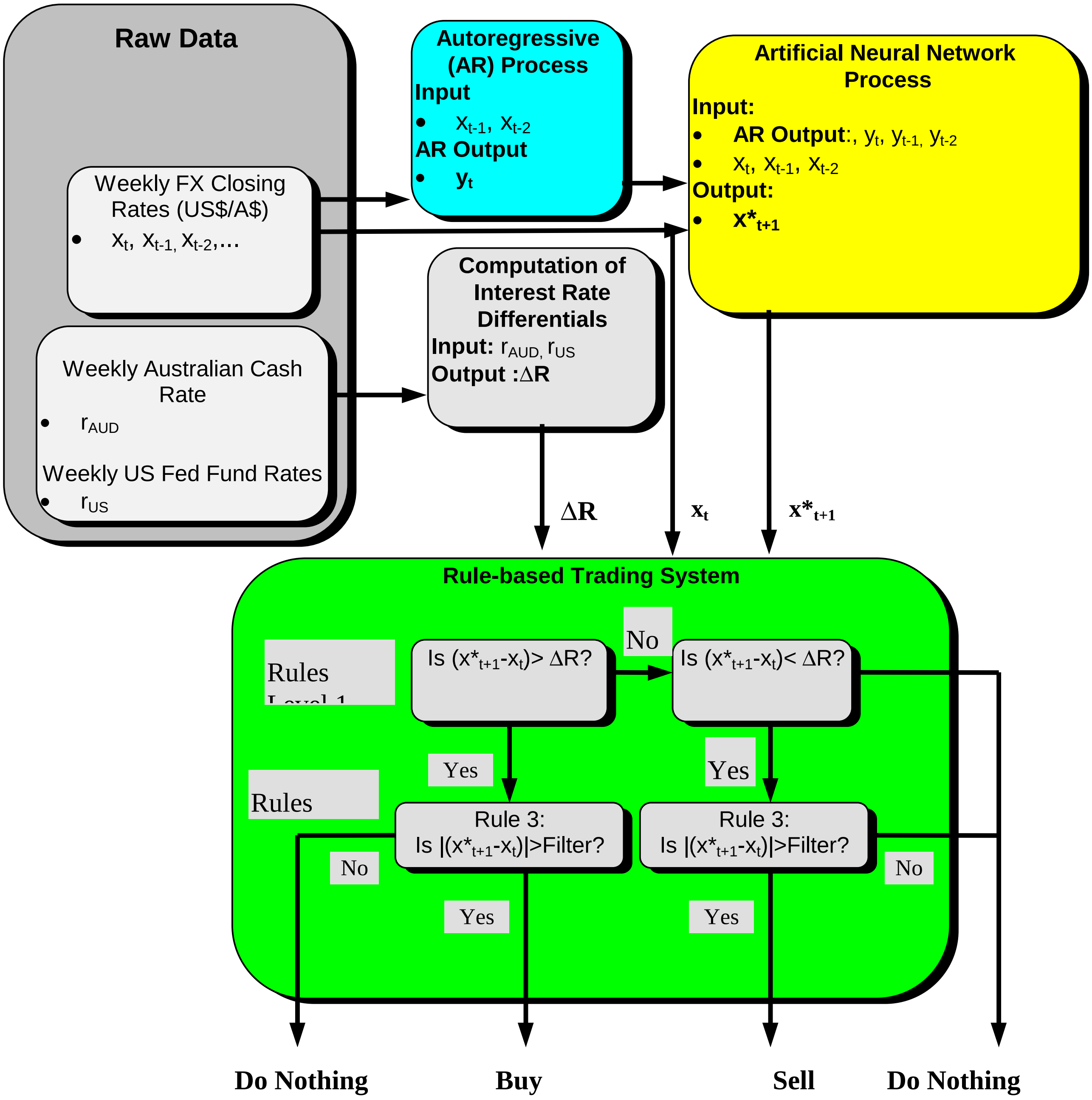
The final stage of the system consists of the rule-based trading system. It uses the ANN forecast, x^*_{t+1} and the interest differential, ΔR, to generate the trading signals. [39]A basic buy signal is indicated when the next week's forecast closing rate is higher than the current closing rate plus the interest differential. This basic rule is then modified to allow for risk aversion by adding filter values, or risk band, around the buy/sell signal. These are discussed in section 5.5.4. If a buy (sell) signal is indicated, the system will 'purchase' (sell) the US currency at the current closing rate and sell (buy) it back at the following week's closing rate, thus closing the position. In this experiment, the actual profits and losses are maintained for the entire period of simulated trading.

[38] Neuralyst 1.4 is a neural network program that runs as a Microsoft Excel macro. The company responsible for the program, Cheshire Engineering Corporation, can be contacted at 650 Sierra Madre Villa Avenue, Suite 201, Pasadena, CA 91107, USA.

[39] Note that this is just one of the many ways the rule can be constructed; e.g. a buy signal may be generated if the closing price has been declining for the past three periods.

Figure 5-1

Artificial Neural Net-AR-based Trading System


Raw Data
Weekly FX Closing Rates (US$/A$)
• x_t, x_{t-1}, x_{t-2},…
Weekly Australian Cash Rate
• r_{AUD}
Weekly US Fed Fund Rates
• r_{US}
Autoregressive (AR) Process
Input
• x_{t-1}, x_{t-2}
AR Output
• y_t
Artificial Neural Network Process
Input:
• AR Output:, y_t, y_{t-1}, y_{t-2}
• x_t, x_{t-1}, x_{t-2}
Output:
• x^*_{t+1}
Computation of Interest Rate Differentials
Input: r_{AUD}, r_{US}
Output : ΔR
ΔR
x_t
x^*_{t+1}
Rule-based Trading System
Rules Level 1
Rules
Is $(x^*_{t+1}-x_t)$> ΔR?
No
Is $(x^*_{t+1}-x_t)$< ΔR?
Yes
Yes
Rule 3: Is $|(x^*_{t+1}-x_t)|$>Filter?
Rule 3: Is $|(x^*_{t+1}-x_t)|$>Filter?
No
No
Yes
Yes
Do Nothing
Buy
Sell
Do Nothing

5.5.1 The Rules Structure Underlying Approach

A simple rule structure is used to determine the amount of money that the trading systems are capable of making. The structure is similar to that constructed by Tan [Tan 1993b, Wong and Tan 1994]. The structure in this research differs from the earlier one, in that the structure assumes that the foreign exchange trade involves borrowing or lending in domestic (Australian) and foreign (US) assets. Thus, interest differentials as well as transaction costs are taken into account in deciding which trades should be undertaken, given a one week ahead forecast of the foreign exchange rate. The home currency is assumed to be the Australian dollar, although switching this to the US dollar does not change the exercise materially. A more detailed discussion on the trading rules follows in section 5.5.3.

Buy and sell signals are determined by reference to the pricing boundaries implied by arbitrage pricing for futures contracts. Under arbitrage pricing, the theoretical futures price lies between the following upper and lower bounds.

$$f^{u}_{t,T} = S_t(1+i_t - y_t) + \tau_t$$

Equation 5-2

$$f^{l}_{t,T} = S_t(1+i_t - y_t) - \tau_t$$

Equation 5-3

where f is the futures contract price, S is the spot price at the trading time, t is the trading date, T is the future date, i is the funding cost, y is the earnings from placement, *u* is the upper bound, *l* is the lower bound and τ is the transaction costs.

When the actual futures price lies above the upper bound, arbitrageurs can make risk free profits by buying spot, funding the position and selling futures. When the actual futures price lies above the lower bound, profits can be made by selling spot and buying futures. A similar logic applies to speculative positions where the forecast future spot rate replaces the theoretical futures price. The fundamental difference between the two structures is that arbitrage involves risk-free profits. In contrast, speculation involves profits, with all the attendant uncertainties.

It is not possible to define a universal trading rule. Ultimately, attitudes towards risk govern the choice of a trading rule. A risk-neutral speculator will undertake any trade for which the forecast exchange rate lies outside the boundaries implied by arbitrage pricing. A risk-averse speculator will require a greater spread between the forecast rate and the arbitrage boundaries, where the size of the spread will depend on the degree of risk aversion; a higher spread is consistent with a higher expected return on the transaction. For example, a highly risk-averse speculator might only trade when the forecast rate is outside the arbitrage range and a very high level of statistical significance. Increasing the level of significance reduces the number of trades, but increases the probability of profit on the trades undertaken. The following section describes the calculation of the arbitrage boundaries as though the investor is risk neutral. Section 5.5.4 below defines the filter rules.

5.5.2 Calculating the Arbitrage Boundaries

The trading system assumes that when a trade is transacted, the transaction is funded through borrowing, in either the domestic (in the case of buying foreign currency), or foreign money market (in the case of selling the foreign currency), and placing the transacted funds in the appropriate money markets at the prevailing rates. For example, when a buy signal is generated, it is assumed that the trader will buy the foreign asset by borrowing local currency (A$) funds from the domestic market (in this case the Australian money market) at the weekly domestic cash rate (in this case the Australian Weekly Cash rate), purchase the foreign currency (US Dollar), and invest it for one week at the foreign money market rate (US Fed Funds weekly rate). In the case of a sell signal, the trader will borrow from the foreign money market at the prevailing rate (US Fed Funds), sell the foreign currency (US Dollar) for domestic currency (A$), and invest the proceeds for one week in the domestic money market at the prevailing rate (Australian Cash rate). Transaction costs are important in short-term transactions of this type. Bid-offer spreads in the professional AUD/USD market are normally around 7 basis points. Spreads in short-term money markets are usually around 2 basis points. Since the data available for these rates are mid rates, a transaction cost of 7 basis points is assumed for a two-way foreign exchange transaction while a transaction cost of 2 basis points is assumed in the money market transactions.

Since these are the normal interbank spread, sensitivity analysis of the spread is not carried out. However, sensitivity analysis of the different filter values is performed, and this is similar to performing a sensitivity analysis on the foreign exchange rate spread.

The interest differential and the spread in both the money market and foreign exchange transactions represent the cost of funds for performing such transactions. Thus, a risk-averse investor will trade if the forecast exchange rate change lies outside the band set by the interest differentials and transaction costs. As noted in the previous sections, the limits of this band is referred to as the arbitrage boundaries, since they correspond to the arbitrage boundaries for futures pricing.

The formula for computing the interest differential in terms of foreign exchange points in deciding whether to buy foreign currency (US dollar) is as follows:

Interest Differential = Foreign Asset Deposit Interest - Local Funding Cost

$$= \left[\frac{1}{\left(x_t + \frac{fxspread}{2} \right)} \times \left(\frac{1 + foreign\_interest\_rate - \frac{intspread}{2}}{52} \right) \times \left(x^*_{t+1} - \frac{fxspread}{2} \right) \right] - \left[\frac{1 + local\_interest\_rate + \frac{intspread}{2}}{52} \right]$$

Equation 5-4

and the formula for selling foreign currency is as follows:

Interest Differential = Domestic Asset Deposit Interest - Foreign Funding Cost

$$= \left[\frac{1 + local\_interest\_rate - \frac{intspread}{2}}{52}\right] - \left[\frac{1}{\left(x_t - \frac{fxspread}{2}\right)} \times \left(\frac{1 + foreign\_interest\_rate + \frac{intspread}{2}}{52}\right) \times \left(x^*_{t+1} + \frac{fxspread}{2}\right)\right]$$

Equation 5-5

where $\boldsymbol{x_t}$ is the current closing exchange rate expressed as units of domestic currency per foreign currency unit, $\boldsymbol{x^*_{t+1}}$ is the forecast following week's closing rate, *fxspread* is 7 basis points or 0.0007 representing the foreign exchange transaction cost, *intspread* is 2 basis points or 0.02% representing the money market transaction cost, foreign_interest_rate is the US Fed Fund rate in percentage points and the local_interest_rate is the Australian cash rate in percentage points.

5.5.3 Rules Structure

The first level rules check if the difference between the forecast and the current closing rate (x^*_{t+1} - x_t) lies outside the arbitrage boundaries set by ΔR. The second level rules are the filter rules discussed in the next section. A 'buy' signal is generated if the difference is beyond the upper boundary and the filter value. Likewise, a 'sell' signal is generated if it is beyond the lower boundary and passes the filter rule. In all other cases, a 'do nothing' signal is generated.

The signals in summary are:

For x^*_{t+1} - x_t > *Upper Boundary + Filter Value,* *Buy*

< *Lower Boundary - Filter Value,* *Sell*

else, *Do Nothing*

Equation 5-6

The model assumes all trades can be transacted at the week's closing exchange rates and interest rates in the calculation of the profitability of the trades.

5.5.4 Filter Rules

The idea of a filter rule is to eliminate unprofitable trades by filtering out the small moves forecast in the exchange rate. The reason for this is that most whipsaw losses in trend following trading systems occur when a market is in a non-trending phase. The filter rule values determine how big a forecast move should be before a trading signal is generated. Obviously, small filter values will increase the number of trades while large values will limit the number of trades. If the filter value is too large, there may be no trade signals generated at all.

This research uses filter values ranging from zero to *threshold values*. Threshold values are filter values at which beyond, all trades are eliminated (filtered out) for each of the three models. The filter rules are linear in nature but their relationship to the proft results are nonlinear, as can be observed in the results discussed in later sections. More rules can of course be added to the system. These additional rules can be the existing rules in technical analysis indicator-based trading systems, or econometric models that are based on fundamental information. However, it is necessary to determine whether additional rules will enhance the trading system.

5.6 ANN Topology and Parameter Settings

5.6.1 Model Selection

The input data for the ANN are the closing foreign exchange rates of the previous three weeks and derivatives of the closing rates; i.e. the AR outputs. This chapter refers to the ANN model with the AR input as an ANNWAR and the ANN model without the AR as ANNOAR. As mentioned earlier, both the ANNWAR and ANNOAR models is referred to collectively as ANN models in general if no differentiation is needed. Experimentation with adding more inputs using technical indicators, i.e. relative strength index, moving averages and momentum; and fundamental indicator inputs, i. e. interest rates, did not improve the ANN models' performance.

All ANNs constructed in this study use the same set of initial weights. This allows the results obtained to be easily replicated. The ANNs were trained over 25,000 iterations although the best ANNOAR model needed only 3,000 iterations to be fully trained while the best ANNWAR model required 10,000 iterations. Further training did not improve results and actually reduced accuracy. This could be due to 'curve-fitting' which, as noted earlier, occurs when an ANN starts to specifically model the training set rather than build a general model of the problem at hand.

As there are no hard and fast rules on setting the correct parameter values for ANNs nor are there any in determining the best ANN architecture/topology, many ANNs were constructed with different network topologies and different parameter settings in an attempt to find the best model through trial and error. The performance of the models was measured by total profitability generated from the trades initiated by each model. Genetic algorithms (GAs) were used to determine the best combination of parameter settings and architecture. The initial architecture consisted of 9 different input variables and 2 hidden layers. The fitness criteria or objective of the GAs was to determine the best ANNOAR and ANNWAR models in terms of the smallest Root Mean Square Error (RMSE) of weekly forecasts.

The GAs found the best ANNWAR model to be one using only two variables and an architecture with one hidden layer. The final ANNWAR model consist of an ANN with one hidden layer with three hidden neurons, six input neurons (the last three periods of the two variables mentioned above) and one output neuron. The best ANNOAR model is one with one variable and an architecture with one hidden layer. The ANNOAR model has 3 input neurons, three hidden neurons and one output neuron. It must, however, be noted that the ANN models that gave the best profit for this research do not necessarily preclude the existence of a better ANN model. It is almost impossible to do an exhaustive search for the best ANN model to be conclusive.

The parameter settings determined by the GAs did not perform well. Therefore, the ANNWAR and ANNOAR models parameter settings were chosen based on experiences and are shown in Table 5-1.

Table 5-1
Summary of the ANN Parameter Settings

Network Parameters	
Learning rate	0.07
Momentum	0.1
Input Noise	0.1
Training Tolerance	0.01
Testing Tolerance	0.01

A brief description of each of the parameters is discussed below:

5.6.1.1 Learning Rate

The learning rate of the neural net was tested with fixed values ranging from 0 to 1 using GAs. However, through experience, 0.07 is chosen as it is always better to use small values of learning rates to ensure that a solution is not missed during the training of the ANN. Large learning rates tend to overshoot good solutions. Small learning rates, while ensuring a solution (if one exist) will be found, do require longer training time, and have the tendency to be trapped in local minimas. However, financial market data that are normally noisy enough to jolt the ANN out of local minimas coupled with the addition of adding small amount of random noise to the ANN, resolve this problem.

5.6.1.2 Momentum

Momentum values were varied by the GAs from 0 to 1. Again from experience, good results are obtained when small momentum values are used in conjunction with small learning rates. Therefore, a relatively low momentum value of 0.01 is used.

5.6.1.3 Input Noise

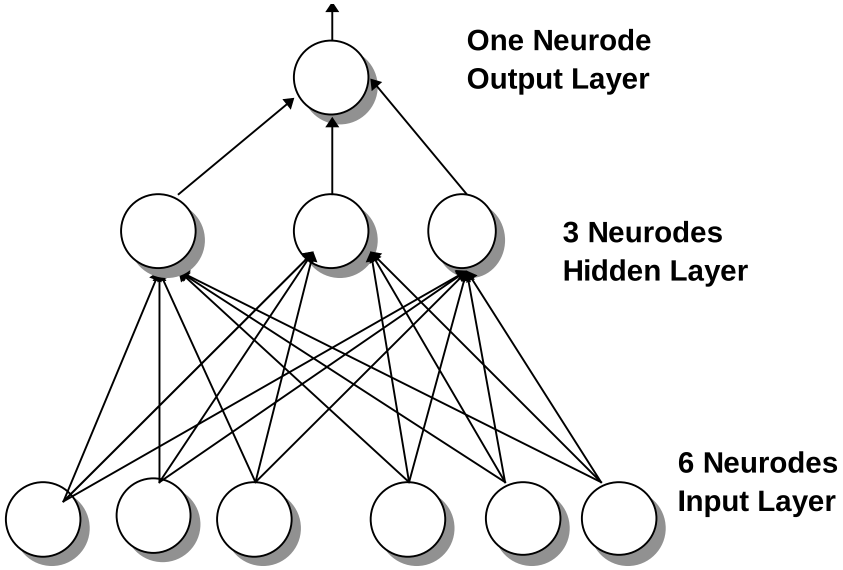
As mentioned earlier, input noise was added after the other parameter settings were set. The level chosen was 1% of the input range that was small but sufficient to improve the results of a similar model without noise.

5.6.1.4 Training and Testing Tolerances

Since the standard error was about 0.35% of the mean of the data set, the training and tolerance level were set to 0.1%. A training tolerance that is too small will cause an ANN to ‘overtrain’ or curve-fit while a value that is too large will result in a failure to learn. The testing tolerance value plays no significant role in this study and is used only as a reporting tool to indicate how many of the predictions fall within the tolerance band.

The ANN topology is shown in Figure 5-2.

Figure 5-2 The ANN Architecture



5.7 Problems encountered in the research

The main problem encountered was the multidimensional complexity in determining the best model for the financial time series, from parameter settings for the ANN to the filter values for the trading model. The selection of the ideal parameter settings, architecture and input variables is very time consuming as these factors are not mutually exclusive and there is no easy method of selection. The GA program assists in the selection of the best ANN model in terms of RMSE of the forecast but the GA program itself takes a very long time to run, especially if it is required to search through a larger problem space. A typical run took a Pentium 100 MHz computer more than forty-eight hours, with no guarantee that the model found is optimal, or even profitable. However, once the ANN model is selected, new data can be quickly retrained if desired, without needing to readjust the parameter settings or architecture. It is the initial task of model selection that is long and tedious.

Another time consuming process occurs in the analysis of the results obtained from each model in the model selection process Model selection is based on the highest return obtained by the models from the test set data. Since the ANN model changes with each iteration in the learning process, determining the number of iterations required to obtain the optimal model is as important as the parameter settings in order to allow the model to be replicated.

To do this requires the periodic halting of the ANN learning process, forecasting the test set data, passing the forecast through the trading system and recording the profit or loss generated. The entire process needs to be repeated every time changes are made to the input variables, trading rules, filter values and parameter settings.

This process was automated through the use an Excel macro that was used in earlier papers the author co-authored [Tan and Wittig 1993ab]. The macro automatically halts the ANN at fixed iteration intervals, tests the ANN, records the profits generated the ANN from the training and test set, i.e., the in-sample data set, and continues training the ANN till the next iteration interval or the maximum number of iterations has been reached. At the end of it, the results for each iteration interval are sorted according to the best profits in the test and training data sets.

5.8 Results

The results are reported in terms of profitability of the trading systems. Earlier studies have shown that the direction of the forecast is more important that the actual forecast itself in determining the profitability of a model [Tsoi, Tan and Lawrence 1993ab, Sinha and Tan 1994]. Only the out-of-sample results are reported; i.e. the last 21 observations; as this is the only data set that provides an informative and fair comparison of the models. The profits and losses are given in terms of foreign exchange points in local currency terms (Australian dollar); for example, a profit of 0.0500 points is equivalent to 5 cent for every Australian dollar traded or 5% of the traded amount. The Mean Square Errors (MSE)of the models' forecasts are reported in Table 5-2 as well as a brief analysis on the forecast results in section 5.8.8.

5.8.1 Perfect Foresight Benchmark Comparison

The two models are compared with a "perfect foresight" (PF) model benchmark. This model assumes that every single trade is correctly executed by the trading system when it is given perfect foresight into the future and knowledge of the actual closing exchange rates for the following week. Under this model, all profitable trades are executed.

5.8.2 Performance Metrics

The results are reported with different filter values. Table 5-3 breaks the results down into different trading performance metrics to help asses the impact of the filter values. A set of performance metrics is used to provide a more detail analysis on the trading patterns generated by each model. They are as follows:

i. Total gain

This is the total profit or loss generated by each model for each of the different filter values.

$$\textit{Total gain} = \sum \textit{profit} / \textit{loss}$$

Equation 5-7

ii. Largest loss per trade

This is the trade with the greatest loss. One of the reasons for using filter values is to reduce this value of this loss. Increasing filter values should reduce this loss.

$$\text{Largest loss per trade} = \textit{Minimum}\left[\textit{profit} / \textit{loss}\right]$$

Equation 5-8

iii. Largest gain per trade

This is the converse of the previous metric. Filter values may eliminate some of the good trades too.

$$\text{Largest gain per trade} = \mathit{Maximum}\left[\mathit{profit} / \mathit{loss}\right]$$

Equation 5-9

iv. Average profit per trade

This metric provides an indication of the profitability of each trade transacted.

$$\text{Average profit per trade} = \frac{Total\_Gain}{Total\_No.\_of\_Trades}$$

Equation 5-10

v. Winning trades

This is the number of trades that generated a profit.

$$\text{Winning trades} = \sum trades \big| (profit / loss > 0)$$

Equation 5-11

vi. Percentage of Winning trades

This is the percentage of trades from the total number of buy or sell trading signals generated. The formula is:

$$\mathit{Percentage\ of\ Winning\ trades} = \frac{\sum Winning\_Trades}{\sum Buy\_Signals + \sum Sell\_Signals} \times 100$$

Equation 5-12

vii. Percentage of correct trades to perfect foresight

A correct trade is classified as a trade that has the same signal as the PF model. A correct trade may not necessarily be a winning trade as a ‘Do nothing’ signal may or may not be a correct trade but all winning trades are correct trades. This metric measures how close the ANN model is to simulating the PF model.

$$\text{Correct trades} = \sum trades \,\middle|\, signal = signal_{PF}$$

Equation 5-13

$$\text{Percentage of correct trades to perfect foresight} = \frac{\sum Correct\_trades}{\sum trades} \times 100$$

Equation 5-14

viii. Buy, Sell and Do Nothing signals

These metrics report on the number of each of the three types of signal generated by the models. Increasing filter values should increase the 'Do Nothing' signals as trades (hopefully, the unprofitable ones) are eliminated. The condition for the signals are as stated in Equation 5-6.

5.8.3 Profitability of the Models

Table 5-2 shows a summary comparison results of all three models with the PF benchmark model over a range of filter values from 0.0000 to 0.0020. Tables 5-3 (a,b,c & d) give a more detailed comparison over a wider range of filter values from 0.0000 to 0.0100 with the various performance metrics discussed in the previous section. For all models, filter values greater than 0.0100 eliminated too many profitable trades and degraded the performance of the trading models; thus the performance metrics are only reported for filter values up to 0.0100.

From Table 5-2, the ANNWAR model clearly outperforms the ANNOAR and AR models in all cases. However, the ANNWAR model's profitability is still significantly below the maximum achievable profit that is indicated by the PF model, indicating there may be room for improvement. The AR model gives the worst performance, suffering losses in all cases except at filter value of 0.0020. The ANNOAR model is the second best performer in all cases except when the filter value was set to 0.0020. However, its profitability was more susceptible than the ANNWAR model to filter value increments.

Charts 5-1, 5-2 and 5-4 give a visual result on the effect of filter values over the PF, AR, ANNOAR and ANNWAR models respectively. Chart 5-5 compares the effect on all three models while Chart 5-6 compares the effect on the two ANN models.

The results indicate that increasing the filter values generally reduces overall profitability. By reducing the number of trades, however, the filter rules also lower risk and the volume of funds committed to speculation.

To judge the impact of the filter rules, it is useful to assess the impact of the filter on the number of trades undertaken and the profitability per trade metrics. The results show that the threshold filter values[40] were different for each model. They were:

[40] A threshold filter value is the limit value for the filter before all trading signals are eliminated.

i.	AR:	0.0325
ii.	ANNOAR:	0.0195
iii.	ANNWAR:	0.0162

The filter rules obviously have little or no impact on the PF model.

Charts 5-7, 5-8 and 5-9 show a comparison of the actual A$/US$ exchange rate against the forecast of each of the three models, AR, ANNOAR and ANNWAR respectively. Chart 5-9 compares the ANNOAR's forecast against the ANNWAR's forecast with the actual exchange rates as a benchmark. This comparison is made as the ANNOAR and ANNWAR graphs seems to be very similar, but yet their profitability performances are quite different.

5.8.4 PF Model's Profitability Performance

The PF as mentioned earlier, serves a benchmark and reflects the ideal model. From Table 5-3a&b and Chart 5-1, an increase in filter values results in a decrease in total profits; from 0.1792 at zero filter value to 0.1297 at a filter value of 0.0100. There is also an increase in the average profit per trade, revealing that only small profitable trades are filtered out. The average profit per trade increases from 0.0090 at zero filter value to 0.0162 at filter value of 0.0100. Increasing the filter values reduces the overall total number of trades. The largest total gain per trade is 0.0326 and is not filtered out by the range of filter values used in the test. The PF model's profit remains steady at 0.0326 at 0.0.0210 before finally being filtered out at 0.0326. This steady state profit is derived from just one trade which is the trade with the largest total gain.

5.8.5 AR Model's Profitability Performance

The AR model's profitability performance is erratic as observed in Chart 5-2 and Tables 5-3a&b. They show that the AR model is quite sensitive to the filter values; a change in filter values by a mere 0.0005 could reverse profits to losses and vice-versa. From Table 5-3a&b, the highest total gain achieved by the AR model is 0.0193 when using filter values of 0.0030 to 0.0040 while the biggest loss is -0.0184 at a filter value of 0.0015. Filter values of 0.0100 and 0.0145 are the only other values that give significant profitable performance; profits of 0.0175 and 0.0192 respectively. Chart 5-2 shows that a constant profit of 0.0036 is achieved from 0.0200 to the threshold value of 0.0325. The AR model has the largest threshold value. This is in contrast to the previous study by the author [Tan 1995ab] where that AR model achieved significant profits and outperformed a simple ANN (called an ANNOAR in this study) but had all its trades filtered out from 0.0010.

The AR model's worst average loss per trade is -0.0011 at filter value of 0.0015 while its best average profit per trade is 0.0018 at a filter value of 0.0100. The average profit/loss per trade fluctuates over the different filter values. The AR model's largest gain per trade is 0.0207 while the largest loss per trade is -0.0347 at filter values of 0.0000 to 0.0015. This loss reduces to -0.0104 at a filter value of 0.0100. The AR model did not manage to capture the single biggest possible gain per trade of 0.0326 as indicated by the PF model in Table 5.3a&b.

The percentage of winning trades for the AR model does not improve significantly with the increments in filter values. From Table 5-3a&b, the highest percentage of winning trades is 60% at a filter value of 0.0100 while the lowest is 46.15% at filter values range of 0.0050 to 0.0065. The percentage of correct trades to PF never exceeds 47.62% in the range of filter values (0.0000 to 0.0100) tested. It does not seem to have a clear positive correlation total profit. In some cases, filter values with higher profits actually has lower percentages of correct trades to PF. For example, the filter value of 0.0030 corresponds to the highest total profit (0.0193) but also to the second lowest percentage of correct trades to PF (38.10%).

In contrast to the earlier study [Tan 1995ab], the filter value increment from 0.0000 to 0.0005 improved the percentage of winning trade from 72.73% to 100% but the percentage of correct trades to PF decreased from 45% to 35%.

5.8.6 ANNOAR Model's Profitability Performance

The ANNOAR model in this study, significantly outperformed the AR model but fails to achieve the standard of the ANNWAR model. This is in contrast to the earlier study by the author [Tan 1995ab] where the AR model outperformed the simple stand-alone ANN model (referred to as ANNOAR in this study). In fact, that is the main motivation for the experimentation with hybrid ANN models, which subsequently resulted in the development of the ANNWAR model.

From Tables 5-2 and 5-3c&d show that the ANNOAR model's highest profit of 0.0546 is achieved without any filter values. Incrementing the filter value to 0.0005 reduced profit by more than 50% to 0.0220. This profit was further reduced by 50% to 0.0110 when the filter value is incremented to 0.0010. However, the total profit gradually increases again to a maximum of 0.0220 before decreasing again to a stable state profit of 0.0036 at the filter value of 0.0095. Chart 5-3 indicates that the profitability of the model remains constant at this level with all subsequent filter values up to the threshold value of 0.0195.

The ANNOAR model's highest average profit per trade is 0.0089 at a filter value of 0.0090 while its lowest is 0.0006 at filter values of 0.0010 to 0.0015. The largest gain per trade is 0.0326 with zero filter value, but this trade is filtered out when the filter value is incremented by 0.0005. The largest loss per trade is -0.0122. However, when the filter value is increased to 0.0070, the largest loss per trade is reduced to -0.0049 and further increments to 0.0090 and beyond, eliminate all unprofitable trades.

The lowest percentage of winning trades is 46.06% at filter values of 0.0010 and 0.0015. The highest percentage of winning trades is 100% when the largest loss per trade is reduced to zero at filter value of 0.0090 to the threshold value. The highest percentage of correct trades to PF is 61.90% at the filter value of 0.0090 while the lowest is 23.81% at the filter value of 0.0040. Generally, the higher filter values (from 0.0080) improve the percentage of correct trades to PF, though not as significantly as the improvement to percentage of winning trades. This means that some profitable trades are eliminated together with the unprofitable trades; i.e. some 'buy' or 'sell' signals in the PF model are incorrectly filtered out resulting in instead in a 'Do Nothing' signal

5.8.7 ANNWAR Model's Profitability Performance

Chart 5-5 and Chart 5-6 suggest the ANNWAR model is the best of the three models in terms of overall profitability. The total profit gained by the model was significantly higher than the AR and ANNOAR at all filter values tested up to 0.0075. The ANNOAR only outperformed the ANNWAR at filter values of 0.0085 to 0.0090. This is due to the fact that both the ANNOAR and ANNWAR have the same stable state profit value of 0.0036 but the ANNWAR hit its stable state profit value earlier at 0.0090.

Table 5.2 and Table 5.3c&d show a total profit gain of 0.0685 with filter values of zero to 0.0009. The total profit dips slightly to 0.0551 when filter value is set to 0.0010 and is reduced by 50% to 0.0225 with filter values of 0.0015 to 0.0025. Total profits gradually increase again to 0.0409 when the filter values are incremented to 0.0335. Total profits then drops to 0.0258 at filter value of 0.0040 but increases to 0.0445 and remains steady there, from filter values of 0.0045 to 0.0050. Total profits dips again at 0.0055 to 0.0286 but recovers to 0.0334 at filter values of 0.0060 to 0.0065. From that value onward, total profits gradually decreases to a stable state profit value of 0.0036 at filter value of 0.0090 where it remains till the threshold value of 0.0162 is reached. All trades beyond that value are eliminated.

The ANNWAR model's average profit per trade ranges from 0.0015 (at filter values of 0.0015 to 0.0020) to 0.0084 (at filter values of 0.0060 to 0.0065). The largest gain per trade achieved by this model is 0.0326. This is the same value achieved by the ANNOAR model. However, unlike the ANNOAR, this highly profitable trade is not eliminated at 0.0005. In fact, it is only eliminated at filter value of 0.0015. The largest loss per trade is -0.0122 at filter values of zero to 0.0030. It is eliminated with a filter value of 0.0035 resulting in the next largest loss per trade of -0.0104. This trade is quickly eliminated when the filter value is set to 0.0045, which reduces the largest loss per trade to -0.0049. At a filter value of 0.0060, all unprofitable trades are eliminated.

The percentage of winning trade decreases from 58.82% at filter values of 0.0000 to 0.0005 to 53.33% at filter values of 0.0015 to 0.0025. It gradually improves from 53.85% at a filter value of 0.0030 to a 100% from the filter value of 0.0060. The percentage of winning trades to PF initially increases from 47.62% zero filter value to 52.38% before decreasing to a low of 28.57% at a filter value of 0.0040. However, from a filter value of 0.0045, it gradually increases to 57.14% at 0.0090. Interestingly, the initial filter values that give the highest average profit per trade and a 100% of winning trades do not correspond to the highest percentage of correct trades to PF. This indicates that some profitable trades are eliminated at those filter values but the majority of the remaining trades are highly profitable.

Table 5-2
Summary of the Models' Profitability: Perfect Foresight (PF), Autoregressive (AR), Artificial Neural Networks with no AR (ANNOAR) and Artificial Neural Networks with AR (ANNWAR)

Models	**PF**	**AR**	**ANNOAR**	ANNWAR
Filter = 0.0000 Total Gain in A$.	0.1792	-0.0074	0.0546	0.0685
Filter = 0.0005 Total Gain in A$.	0.1792	-0.0074	0.0220	0. 0685
Filter = 0.0010 Total Gain in A$.	0.1776	-0.0074	0.0110	0.0551
Filter = .0015 Total Gain in A$.	0. 1765	-0.0184	0. 0110	0.0225
Filter = .0020 Total Gain in A$.	0. 1763	0.0163	0. 0138	0. 0225

Chart 5-1
Effect of Filter Values on the Profitability of the PF Model

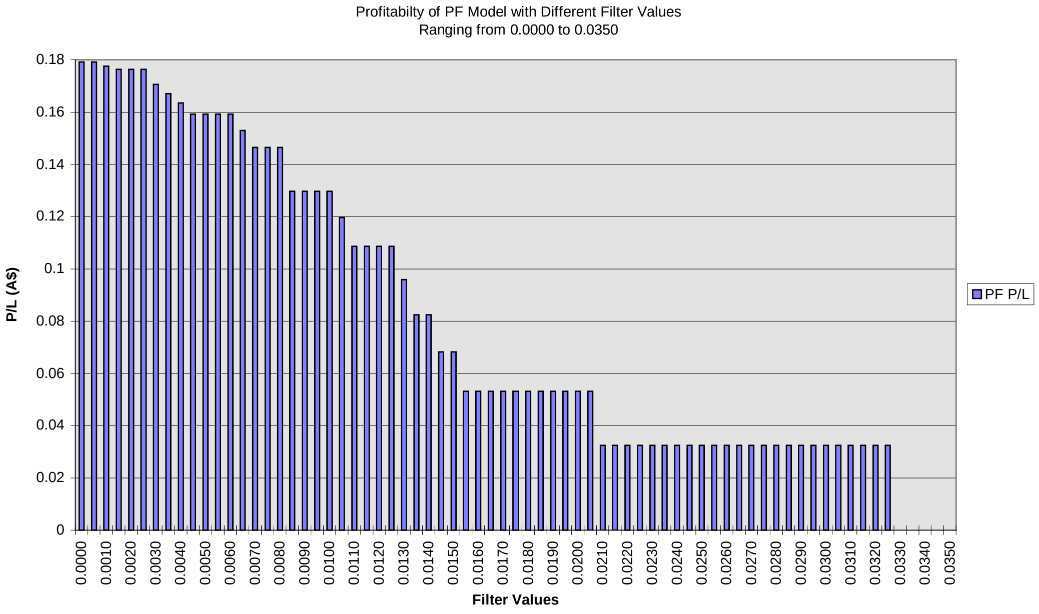


Chart 5-2
Effect of Filter Values on the Profitability of the AR Model

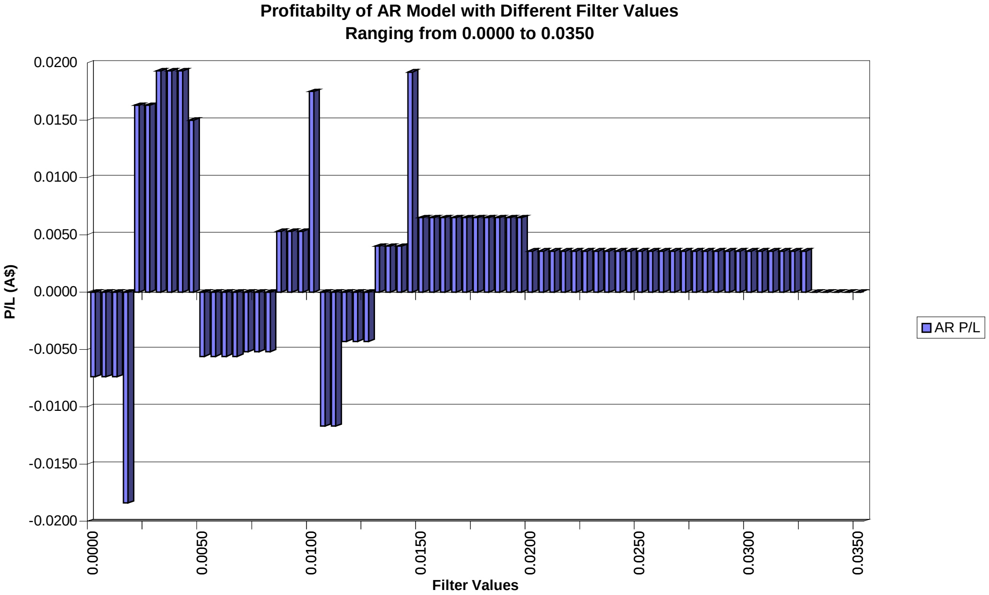


Chart 5-3
Effect of Filter Values on the Profitability of ANNOAR Model

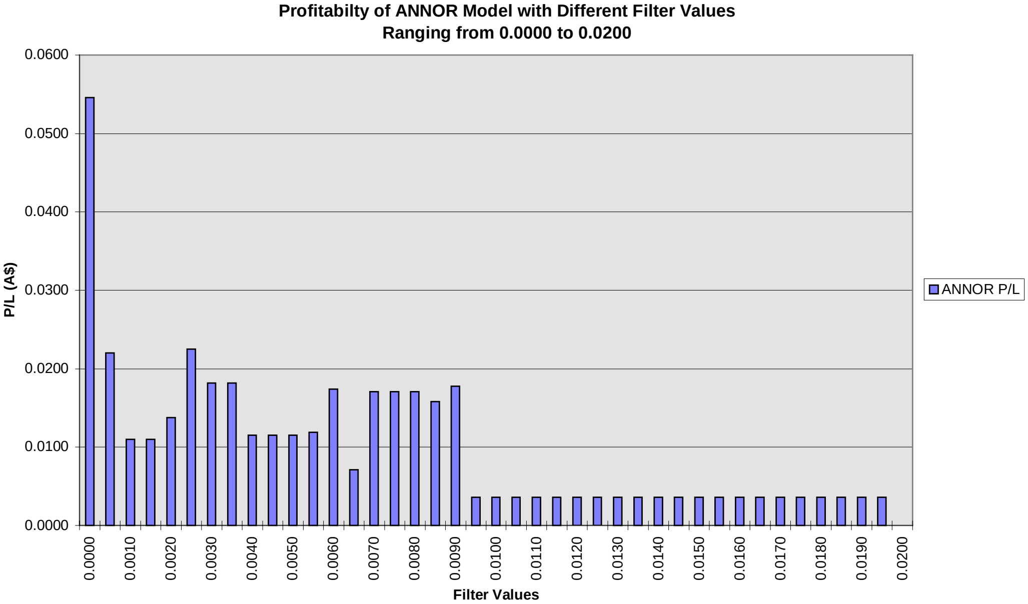


Chart 5-4
Effect of Filter Values on the Profitability of ANNWAR Model

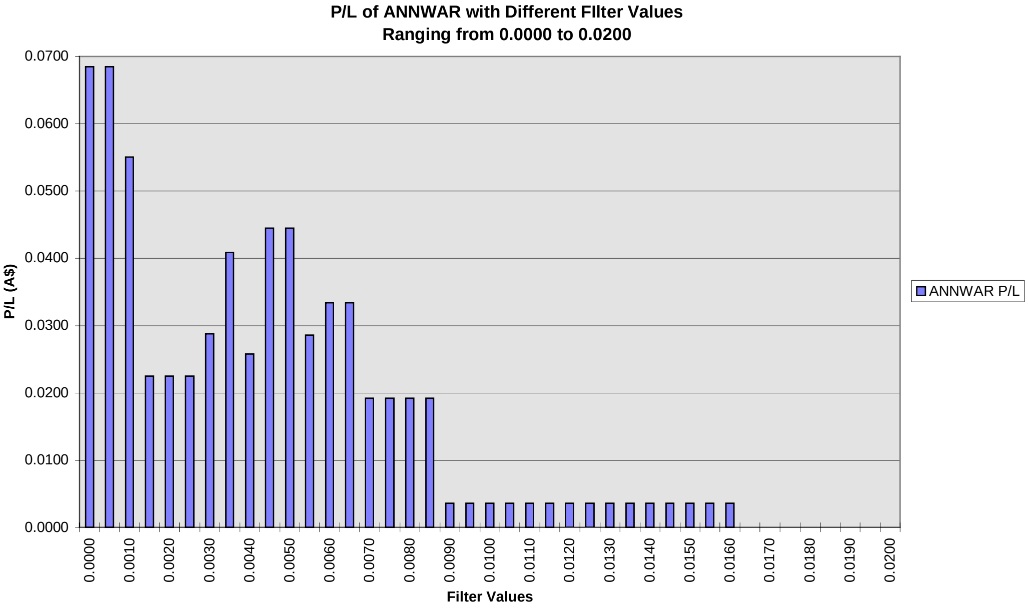


Chart 5-5
Comparison of the Effect of Filter Values on the Profitability of the AR, ANNWAR and ANNOAR Models

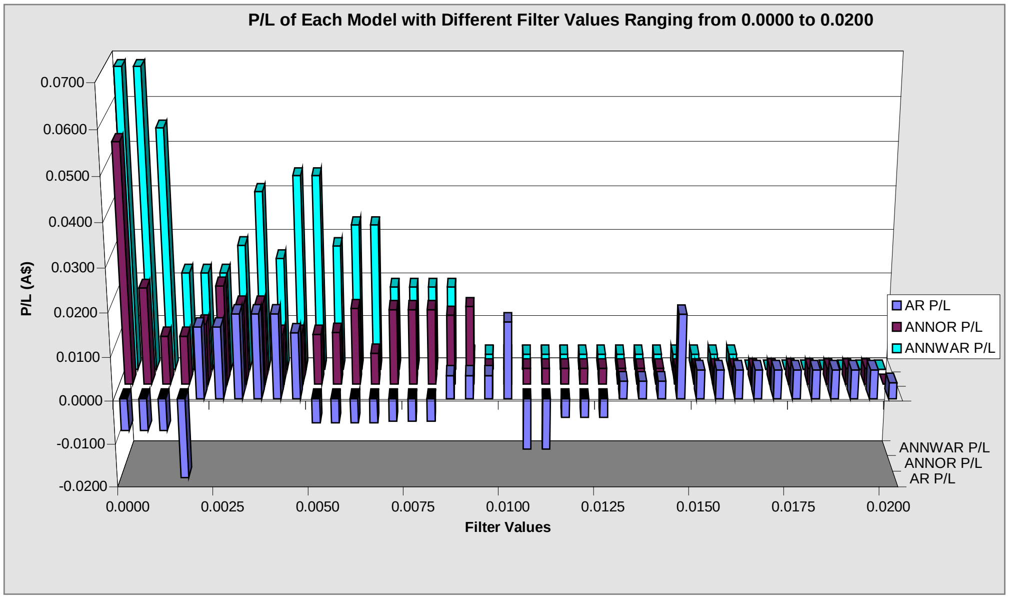


Chart 5-6

Comparison of the Effect of Filter Values on the Profitability of the ANNOAR and ANNWAR Models

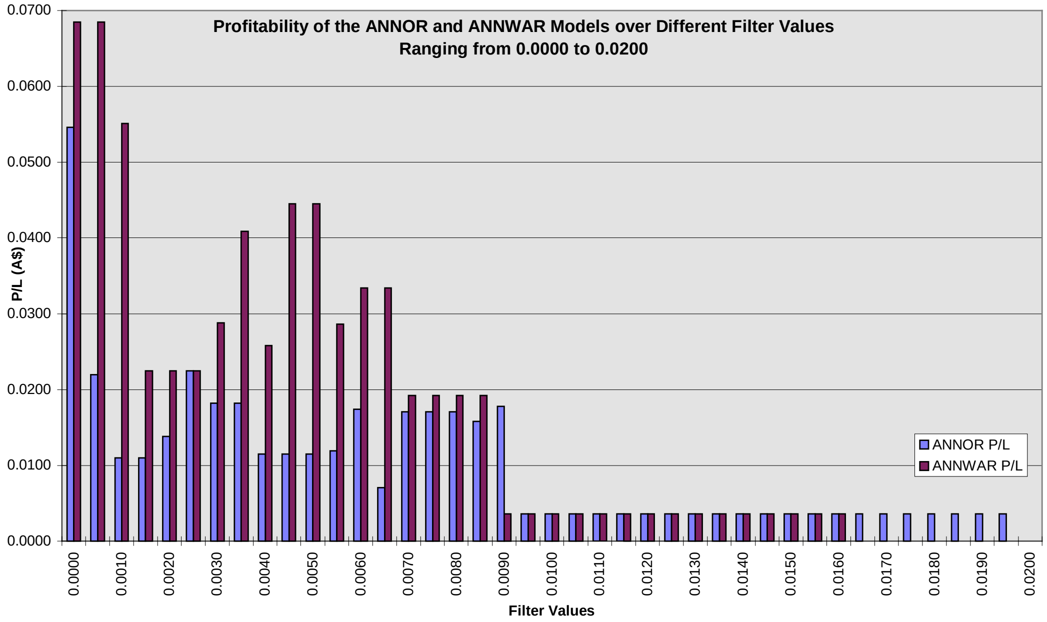


Table 5-3a
Detailed trading comparison of the PF and AR with filter values varied from 0.0000 to 0.0050 basis points.

Filter	**0.0000**	**0.0005**	**0.0010**	**0.0015**	**0.0020**	**0.0025**	**0.0030**	**0.0035**	**0.0040**	**0.0045**	**0.0050**
Perfect Foresight											
Average profit per trade	0.0090	0.0090	0.0099	0.0104	0.0104	0.0104	0.0114	0.0119	0.0126	0.0133	0.0133
Largest loss per trade	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Largest gain per trade	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326
Total gain	0.1792	0.1792	0.1776	0.1763	0.1763	0.1763	0.1706	0.1671	0.1635	0.1592	0.1592
Percentage winning trades	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% of correct trades to PF	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
No of winning trades	20	20	18	17	17	17	15	14	13	12	12
Buy	12	12	11	11	11	11	9	8	8	8	8
Sell	8	8	7	6	6	6	6	6	5	4	4
Do Nothing	1	1	3	4	4	4	6	7	8	9	9
AR											
Average profit per trade	-0.0004	-0.0004	-0.0004	-0.0011	0.0010	0.0010	0.0013	0.0013	0.0013	0.0011	-0.0004
Largest loss per trade	-0.0347	-0.0347	-0.0347	-0.0347	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122
Largest gain per trade	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0150
Total gain	-0.0074	-0.0074	-0.0074	-0.0184	0.0163	0.0163	0.0193	0.0193	0.0193	0.0150	-0.0056
Percentage winning trades	50.00%	50.00%	50.00%	47.06%	50.00%	50.00%	53.33%	53.33%	53.33%	50.00%	46.15%
% of correct trades to PF	42.86%	42.86%	47.62%	38.10%	38.10%	38.10%	38.10%	42.86%	38.10%	38.10%	33.33%
No of winning trades	9	9	9	8	8	8	8	8	8	7	6
Buy	6	6	6	6	6	6	5	5	5	5	5
Sell	12	12	12	11	10	10	10	10	10	9	8
Do Nothing	3	3	3	4	5	5	6	6	6	7	8

Table 5-3b
Detailed trading comparison of the PF and AR with filter values varied from 0.0055 to 0.0100 basis points.

Filter	**0.0055**	**0.0060**	**0.0065**	**0.0070**	**0.0075**	**0.0080**	**0.0085**	**0.0090**	**0.0095**	**0.0100**
Perfect Foresight										
Average profit per trade	0.0133	0.0133	0.0139	0.0146	0.0146	0.0146	0.0162	0.0162	0.0162	0.0162
Largest loss per trade	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Largest gain per trade	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326
Total gain	0.1592	0.1592	0.1530	0.1465	0.1465	0.1465	0.1297	0.1297	0.1297	0.1297
Percentage winning trades	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% of correct trades to PF	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
No of winning trades	12	12	11	10	10	10	8	8	8	8
Buy	8	8	8	7	7	7	5	5	5	5
Sell	4	4	3	3	3	3	3	3	3	3
Do Nothing	9	9	10	11	11	11	13	13	13	13
AR										
Average profit per trade	-0.0004	-0.0004	-0.0004	-0.0004	-0.0004	-0.0004	0.0005	0.0005	0.0005	0.0018
Largest loss per trade	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0104
Largest gain per trade	0.0150	0.0150	0.0150	0.0150	0.0150	0.0150	0.0150	0.0150	0.0150	0.0150
Total gain	-0.0056	-0.0056	-0.0056	-0.0052	-0.0052	-0.0052	0.0053	0.0053	0.0053	0.0175
Percentage winning trades	46.15%	46.15%	46.15%	50.00%	50.00%	50.00%	54.55%	54.55%	54.55%	60.00%
% of correct trades to PF	33.33%	33.33%	33.33%	38.10%	38.10%	38.10%	42.86%	42.86%	42.86%	42.86%
No of winning trades	6	6	6	6	6	6	6	6	6	6
Buy	5	5	5	4	4	4	4	4	4	4
Sell	8	8	8	8	8	8	7	7	7	6
Do Nothing	8	8	8	9	9	9	10	10	10	11

Table 5-3c
Detailed trading comparison of the ANNOAR and ANNWAR with filter values varied from 0.0000 to 0.0050 basis points.

Filter	0.0000	0.0005	0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0045	0.0050
ANN without AR (ANNOAR)											
Average profit per trade	0.0029	0.0012	0.0006	0.0006	0.0009	0.0015	0.0013	0.0013	0.0010	0.0010	0.0010
Largest loss per trade	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122
Largest gain per trade	0.0326	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207	0.0207
Total gain	0.0546	0.0220	0.0110	0.0110	0.0138	0.0225	0.0182	0.0182	0.0115	0.0115	0.0115
Percentage winning trades	52.63%	50.00%	47.06%	47.06%	50.00%	53.33%	50.00%	50.00%	50.00%	50.00%	50.00%
% of correct trades to PF	47.62%	42.86%	42.86%	38.10%	42.86%	42.86%	33.33%	33.33%	23.81%	28.57%	28.57%
No of winning trades	10	9	8	8	8	8	7	7	6	6	6
Buy	6	5	5	5	5	5	5	5	3	3	3
Sell	13	13	12	12	11	10	9	9	9	9	9
Do Nothing	2	3	4	4	5	6	7	7	9	9	9
ANN With AR (ANNWAR)											
Average profit per trade	0.0040	0.0040	0.0034	0.0015	0.0015	0.0015	0.0022	0.0034	0.0026	0.0056	0.0056
Largest loss per trade	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0122	-0.0104	-0.0104	-0.0049	-0.0049
Largest gain per trade	0.0326	0.0326	0.0326	0.0207	0.0207	0.0207	0.0207	0.0207	0.0150	0.0150	0.0150
Total gain	0.0685	0.0685	0.0551	0.0225	0.0225	0.0225	0.0288	0.0409	0.0258	0.0445	0.0445
Percentage winning trades	58.82%	58.82%	56.25%	53.33%	53.33%	53.33%	53.85%	58.33%	60.00%	75.00%	75.00%
% of correct trades to PF	47.62%	47.62%	52.38%	42.86%	42.86%	42.86%	33.33%	33.33%	28.57%	33.33%	33.33%
No of winning trades	10	10	9	8	8	8	7	7	6	6	6
Buy	7	7	6	5	5	5	5	5	5	4	4
Sell	10	10	10	10	10	10	8	7	5	4	4
Do Nothing	4	4	5	6	6	6	8	9	11	13	13

Table 5-3d

Detailed trading comparison of the ANNOAR and ANNWAR with filter values varied from 0.0055 to 0.0100 basis points.

Filter	0.0055	0.0060	0.0065	0.0070	0.0075	0.0080	0.0085	0.0090	0.0095	0.0100
ANN without AR (ANNOAR)										
Average profit per trade	0.0011	0.0017	0.0009	0.0034	0.0034	0.0034	0.0040	0.0089	0.0036	0.0036
Largest loss per trade	-0.0122	-0.0122	-0.0122	-0.0049	-0.0049	-0.0049	-0.0049	0.0000	0.0000	0.0000
Largest gain per trade	0.0207	0.0207	0.0142	0.0142	0.0142	0.0142	0.0142	0.0142	0.0036	0.0036
Total gain	0.0119	0.0174	0.0071	0.0171	0.0171	0.0171	0.0158	0.0178	0.0036	0.0036
Percentage winning trades	54.55%	60.00%	62.50%	80.00%	80.00%	80.00%	75.00%	100.00%	100.00%	100.00%
% of correct trades to PF	33.33%	38.10%	38.10%	38.10%	38.10%	38.10%	52.38%	61.90%	57.14%	57.14%
No of winning trades	6	6	5	4	4	4	3	2	1	1
Buy	2	2	2	1	1	1	1	0	0	0
Sell	9	8	6	4	4	4	3	2	1	1
Do Nothing	10	11	13	16	16	16	17	19	20	20
ANN With AR (ANNWAR)										
Average profit per trade	0.0057	0.0084	0.0084	0.0064	0.0064	0.0064	0.0032	0.0036	0.0036	0.0036
Largest loss per trade	-0.0049	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Largest gain per trade	0.0142	0.0142	0.0142	0.0128	0.0128	0.0128	0.0036	0.0036	0.0036	0.0036
Total gain	0.0286	0.0334	0.0334	0.0192	0.0192	0.0192	0.0065	0.0036	0.0036	0.0036
Percentage winning trades	80.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
% of correct trades to PF	38.10%	42.86%	47.62%	47.62%	47.62%	47.62%	52.38%	57.14%	57.14%	57.14%
No of winning trades	4	4	4	3	3	3	2	1	1	1
Buy	2	2	2	2	2	2	1	0	0	0
Sell	3	2	2	1	1	1	1	1	1	1
Do Nothing	16	17	17	18	18	18	19	20	20	20

Chart 5-7

Comparison of Forecast of the Actual Vs AR on Out-of-sample Data

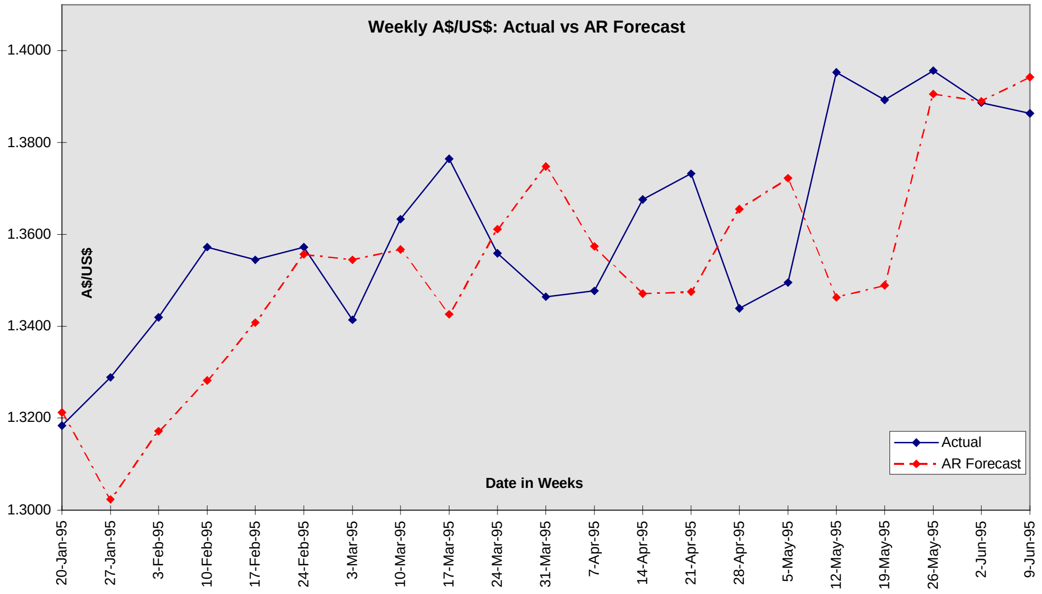


Chart 5-8

Comparison of Forecast of the Actual Vs ANNOAR on Out-of-sample Data

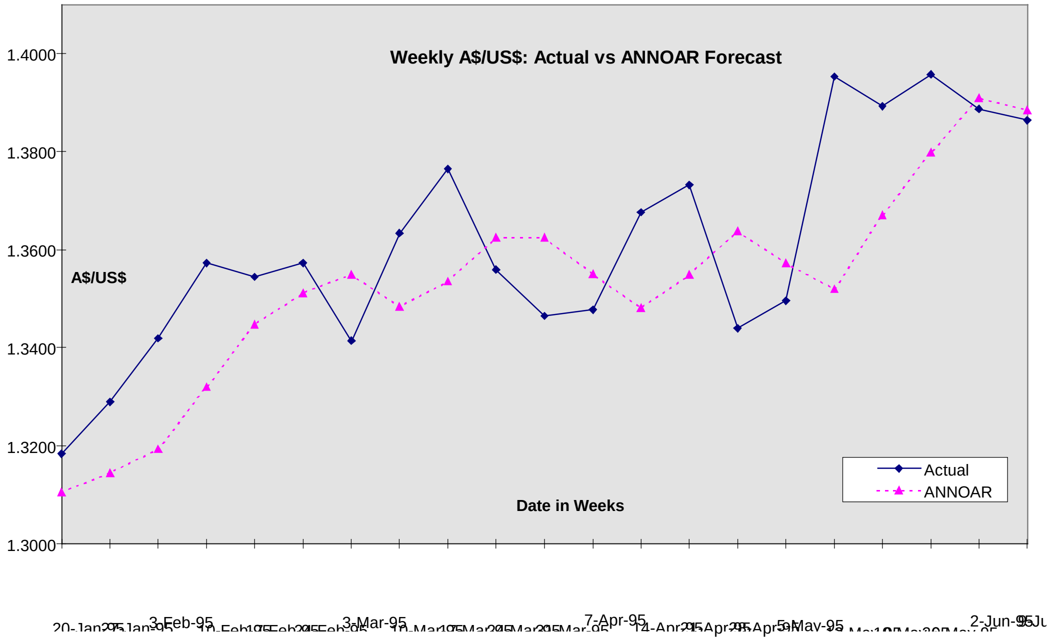


Chart 5-9

Comparison of Forecast of the Actual Vs ANNWAR on Out-of-sample Data

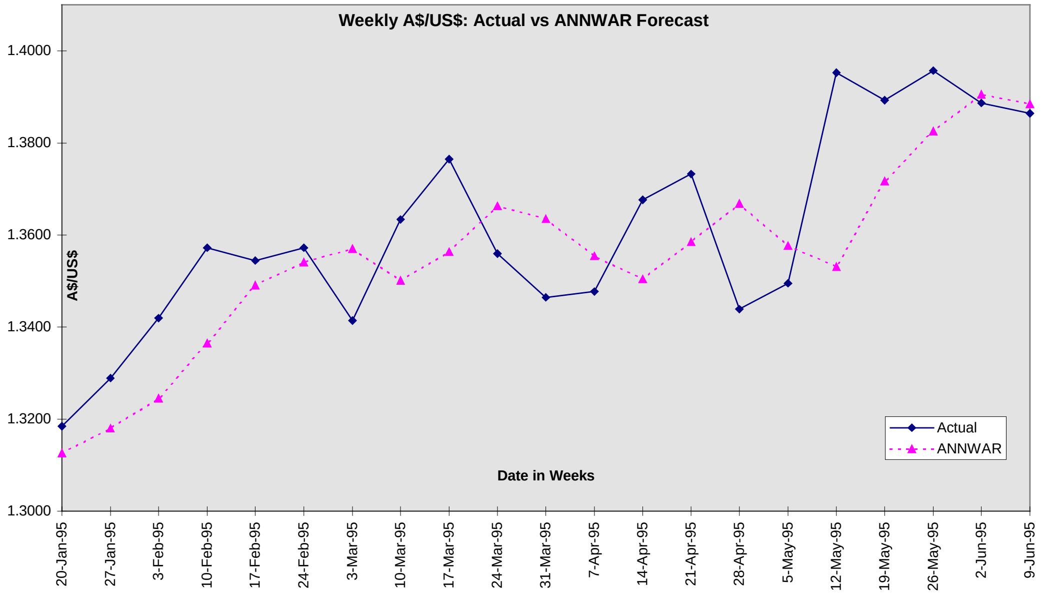


Chart 5-10

Comparison of Forecast of the Actual Vs ANNWAR and ANNOAR on Out-of-sample Data

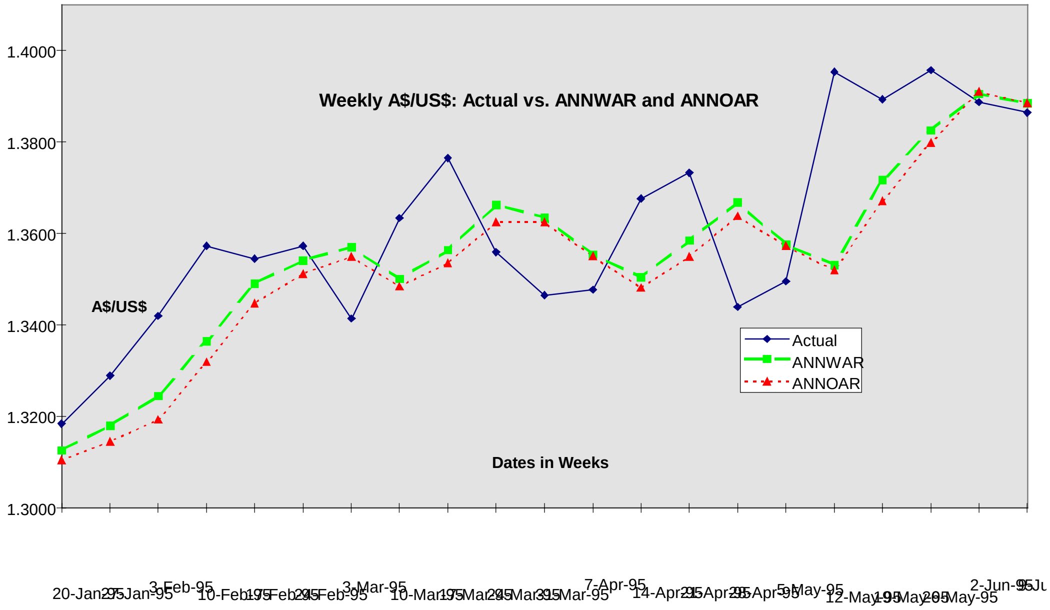


Table 5-4
Comparison of Mean Square Error(MSE) of the Different Models on the out-of-sample data

Models	**MSE**
AR	0.000516
Random Walk Theory (RWT)	0.000278
ANNOAR	0.000317
ANNWAR	0.000266

5.8.8 Forecast Comparisons

Chart 5-7, Chart 5-8 and Chart 5-9 compares the actual A$/US$ exchange rate data to the AR, ANNOAR and ANNWAR models respectively. Chart 5-10 compares the ANNOAR and ANNWAR models to the actual A$/US$ exchange rate.

The charts show that the AR model has a lag of two periods while the ANN models have a slight lag of one. The ANNOAR and ANNWAR models' forecast seem to be very similar.

Table 5-4 demonstrates the model with the lowest Mean Square Error (MSE) for the out-of-sample data is the ANNWAR model. It even outperforms the Random Walk Theory (RWT) model which is marginally the second best model. The RWT model uses the last known exchange rate as the forecast for the next rate and obviously the RWT model thus generates no trades. The ANNOAR model is the third best forecasting model while the AR model is the poorest.

However, as discussed earlier, the reduction in forecast errors does not necessarily translate to better profits. In this case though, this seems to be the case with the ANNWAR performing best, both in terms of exchange rate forecasting, and trading profitability.

5.9 Assumptions and Limitation of Methodology

This methodology assumes that the trader is able to get the stated exchange rate and interest rate spread on every trade. The spread on these rates can differ over time, depending on market volatility and liquidity. An assumption is also made on the trading rules. It assumes that all trades can be executed at the week’s closing rates.

The main limitation of this methodology is that it forces the trader to perform at most, only one trader per week, and every week’s trade has to be closed out at the following week’s closing exchange rate. More favorable rates may have been available at different time of the week for closing out the position. Also, it may be cheaper for the trader to carry a current position for more than a week; especially when the market is trending; as this eliminates the transaction costs.

5.10 Conclusion

The results reported in this chapter suggest that the ANNWAR model, incorporating the AR output into an ANN, can improve the robustness and the profitability of trading systems relative to those based on AR models or ANNs in isolation. The results for this experiment indicate the ANNWAR model to be a more profitable and robust trading system in that it performs better and over a wider range of filter values than the other models.

There appear to be opportunities to exploit some inefficiency in the Australian/US dollar foreign exchange market, as all models return profits after taking account of interest differential and transaction costs. This concurs with studies that have found abnormal profits can be obtained from technical trading and filter rules [Sweeney 1986, Brock et al. 1992, LeBaron 1992]. The utilization of the ANN with other established technical trading rules may improve profitability.

The AR model has been shown to perform poorly in this book. The results of the AR model from this study differ quite significantly from the earlier study [Tan 1995ab]. In that study, the AR model by itself seems ideal for the risk-averse trader as it generates a smaller number of trades and in conjunction with an appropriate filter value, give the best average profit per trade as well as the highest number of winning trades. However, its sensitivity to the filter values, with all trades filtered out at a mere 5 basis points, questions its reliability and stability for use in a real life trading environment.

In this study, however, the AR is unprofitable at most filter values, does not perform well in any of the profitability metrics but is quite insensitive to the filter values, as it is the model that has the highest threshold value. A reason for this could be the more linear nature of the out-of-sample data in the earlier study, allowing the AR model to perform better. However, in this study, the out-of-sample data has a more volatile nature with no clear trend.

In the earlier study, the best ANN architecture is one with no hidden layer. This is the architecture used in the ANNWAR and the ANN in isolation model in this research. This suggest that the best model then, may be a linear forecasting model. It is therefore surprising that the AR model, which is, by definition, a linear best-fit method can be improved upon by incorporating the AR output into the ANN. Many studies have suggest that most financial markets are nonlinear in nature, so the results from that time series are quite interesting as it seems to contradict this view. One explanation could be that the filter rules have added a non-linear dimension to the trading system in terms of the performance as measured in terms of profitability.

In this study, the best ANN architecture was a network with one hidden layer. The additional data may have helped the ANN to pick up the non-linearity nature of the exchange rate market. Indeed, Hsieh [1989] and Steurer [1995] have shown that there is considerable evidence of nonlinear structure in the Deutschmark/US Dollar (DEM/USD) exchange rate. Steurer's study suggests that there is a low-dimensionality chaos in the DEM/USD exchange rate and the use of nonlinear nonparametric techniques can produce significantly better results. Artificial Neural Networks have been shown to 'capture chaos because they learn the dynamical invariants of a chaotic dynamical system' [Deco et al. 1995].

The accuracy of the ANNWAR model as measured by the percentage of winning trades reached a 100%. A level above 60% is sufficient for a market maker with low transaction cost to run a profitable foreign exchange desk [Orlin Grabbe 1986]. However, this high percentage of winning trade requires relatively high filter values. By eliminating all of the unprofitable trades, many profitable trades are also eliminated thus reducing the total profit.

The more risk-averse trader may choose to accept a lower total return with the use higher filter values to minimize the possibility of any trading loss, while a more speculative trader may be willing to take the risk of having some unprofitable trades in expectation of a higher return. Further research should investigate if the returns are commensurate with the additional risk.

This study also confirms the robustness of the ANNWAR model that was introduced in my earlier work. The ANNWAR model in this study not only significantly outperformed the other models in terms of profitability but also in terms of exchange rate forecast as measured by the MSE term.

5.11 Managerial and Implementation Issues

The justification for this system can be demonstrated by the amount of excess return it can generate, albeit on historical data. There is a high level risk that this system may not sustain its profit record. Even the top human traders cannot get it right all the time. This is why strict risk and money management controls needs to be implemented with the system to avert financial disaster. However, confidence on the system can be gained, if the system tests well with different sets of historical data. This is similar to the track record of a human trader. The system needs to be monitored continuously to ensure that the ANN models in use are performing well. The ANN models may need to be retrained should the system starts showing signs of diverging from its profitability targets.

Resources required for implementing the system include a reliable source for data, a computer system, personnel to ensure compliance and risk management controls are in place, maintenance of database, operational staff to execute the trades and the training of the personnel that will use the system.

5.12 Future Research

Future work will improve upon the naive money management technique of the present research where a fixed amount is traded and the position closed off at the end of the week. The current model assumes all trades can be transacted at the week's closing exchange rates. In real life, the exchange rate may change significantly between the time a trading signal is generated and the actual execution of the trade.

Future research will also focus on constructing ANNs to forecast the direction of the market rather than the absolute foreign exchange rate. Unsuccessful attempts had been made in the course of this research to forecast the market direction or turning points. Future attempts may use different ANN architectures and learning algorithms.

Hybrid intelligent systems, combining fuzzy logic, expert systems, genetic algorithms and ANNs, need to be explored to determine if a more profitable trading system can be implemented. ANNs can be used to forecast the time series, while fuzzy logic and expert systems can assist in determining the trading signals. Genetic algorithms can be used to select the input variables as well as the optimal parameters for the system.

Sensitivity analysis of input variables in ANNs, similar to the analysis conducted by Poh [1994], can be performed in future research to assist in determining the effect of a particular variable on the exchange rate. Trends can be identified by continuous monitoring of the system as new data are obtained.

Another area for further research is to develop a methodology to search and detect chaotic dynamical systems in financial time series. This will help in the selection of financial time series to be used for modeling with ANNs. Financial time series that do not exhibit chaotic behavior may truly be random in nature and thus will prove to be difficult if not impossible to model. Further research needs to be undertaken too, with other foreign exchange rate time series to determine if other foreign exchange markets have similar characteristics and if the ANNWAR models can continue to perform better. Finally, the effect of filter values on the trading systems needs to be explored further. Perhaps a better method of selecting the filter values can be developed.

5.13 References

1. Abu Mostafa, Y. S., "Financial Market Applications of Learning Hints", *Neural Networks in the Capital Market edited by Refenes, A.*, ISBN 0-471-94364-9, John Wiley & Sons Ltd., England, pp. 220-232, 1995.
2. Bourke, L., "The Efficiency of Australia's Foreign Exchange Market in the Post-Float Period", *Bond University School of Business Honours Dissertation*, Australia, September 1993.
3. Brock, W. A., Lakonishok, J. and LeBaron, B., "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns", *The Journal of Finance*, 47:1731:1764, USA, 1992.
4. {BIS95], Central Bank Survey of Foreign Exchange Market Activity in April 1995, Bank for International Settlements Press Communiqué, Basel, October 1995.
5. Colin, A, "Exchange Rate Forecasting at Citibank London*", Proceedings, Neural Computing 1991*, London, 1991.
6. Colin, A. M., "Neural Networks and Genetic Algorithms for Exchange Rate Forecasting", Proceedings of International Joint Conference on Neural Networks, Beijing, China, November 1-5, 1992.
7. Dacorogna, M. M., Muller, U. A., Jost, C., Pictet, O. V., Olsen R. B. and Ward, J. R., "Heterogeneous Real-Time Trading Strategies in the Foreign Exchange Market", *Preprint by O & A Research Group MMD.1993-12-01*, Olsen & Associates, Seefeldstrasse 233, 8008 Zurich, Switzerland, 1994.
8. Davidson, C., June 1995, *Development in FX Markets* [Online], Olsen and Associates: Professional Library, Available: http://www.olsen.ch/library/prof/dev_fx.html, [1996, August 5].
9. Deboeck, G. J., *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets,* ISBN 0-471-31100-6, John Wiley and Sons Inc., USA, 1994.
10. Deco, G., Schuermann, B. and Trippi, R., "Neural Learning of Chaotic Time Series Invariants, *Chaos and Nonlinear Dynamics in the Financial Markets edited by Trippi, R*, Irwin, USA, ISBN 1-55738-857-1, pp.467-488, 1995.
11. Dwyer G. P., and Wallace, M. S., 'Cointegration and Market Efficiency", *Journal of International Money and Finance*, Vol. 11, pp. 318-327.
12. Engel, C., "A Note on Cointegration and International Capital Market Efficiency", *Journal of International Money and Finance*, Vol. 15, No. 4, pp. 657-660, 1996.
13. Fishman, M., Barr, D. S. and Heaver, E., A New Perspective on Conflict Resolution in Market Forecasting, *Proceedings The 1st International Conference on Artificial Intelligence Applications on Wall Street*, NY, pp. 97–102, 1991.
14. Fishman, M., Barr, D. S. and Loick, W. J., Using Neural Nets in Market Analysis, *Technical Analysis of Stocks & Commodities*, pp. 18–20, April 1991
15. Freedman, R. S., AI on Wall Street, *IEEE Expert*, pp. 3–9, April 1991.
16. Freisleben, B., Stock Market Prediction with Backpropagation Networks, *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems 5th*

International Conference IEA/AIE-92, Paderborn Germany, June 9-12, 1992 Proceedings, pp. 451–460, 1992.

17. Jang, G. J., Lai, F., Jiang and B., Chien, L., An Intelligent Trend Prediction and Reversal Recognition System Using Dual–module Neural Networks, *Proceedings The 1st International Conference on Artificial Intelligence Applications on Wall Street*, NY, pp. 42–51, 1991.

18. Johnson, R. A., and Wichern, D. D., "Clustering", *Applied Multivariate Statistical Analysis Second Edition*, Prentice-Hall International, pp. 543-589, 1988.

19. Kamijo, K. and Tanigawa, T., Stock Price Recognition A Recurrent Neural Net Approach, *Proceedings the International Joint Conference on Neural Networks*, pp. 589–221, June 1990.

20. Karfakis, C. I. and Parikh, A., "Exchange Rate Convergence and Market Efficiency", *Applied Financial Economics*, vol. 4**,** pp. 93-98**,** 1994

21. Kearney, C. and MacDonald, R., "Efficiency in the Forward Foreign Exchange market: Weekly Tests of the Australian /US Dollar Exchange Rate January 1984-March 1987", *Economic Record*, vol. 67, No. 198, pp. 237-242, September 1991.

22. Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M., Stock Market Prediction System With Modular Neural Networks, *Proceedings IJCNN 1990 Vol. 1*, pp. 1–6, 1990.

23. LeBaron, B., "Do Moving Average Trading Rule Results Imply Nonlinearities in Foreign Exchange Markets?", *Working Paper #9222*, University of Wisconsin-Madison, Social Systems Research Institute, USA, 1992.

24. LeBaron, B., "Technical Trading Rules and Regime Shifts in Foreign Exchange", *Technical Report, University of Wisconsin*, Madison, USA, 1991.

25. Levis, M., "The Behaviour of the Australian Forward Exchange Market", *Australian Journal of Management*, vol. 6, pp. 61-74, 1981.

26. Levitch R. M., and Thomas, L. R. III, "The Significance of Technical Trading -Rule Profits in the Foreign Exchange Market: A Bootstrap Approach", *Journal of International Money and Finance*, vol. 12, No. 5, pp. 451-474.

27. Malkiel, B. C., *A Random Walk Down Wall Street*, New York, W. W. Norton, 1973.

28. Murphy, J. J., *Technical Analysis of the Futures Market*, NYIF, New York, ISBN 0-13-898009-X, pp. 2-4, 1986.

29. Neural Computer Sciences, Using Neural Nets for Financial Forecasting, *PC Business Software Vol. 17 No. 2*, pp. 8–11, 1992.

30. Orlin Grabbe, J., *International Financial Markets*, New York, Elsevier, 1986.

31. Pictet O. V., Dacorogna M. M., Muller U. A., Olsen R. B., and Ward J. R., "Real-time trading models for foreign exchange rates", *Neural Network World*, vol. 2, No. 6, pp. 713-744, 1992.

32. Poh, H. L., "A Neural Network Approach for Decision Support", *International Journal of Applied Expert Systems*, Vol. 2, No. 3, 1994.

33. Refenes, A. and Saidi, A., "Managing Exchange-Rate Prediction Strategis with Neural Networks", *Neural Networks in the Capital Market edited by Refenes, A.,* ISBN 0-471-94364-9, John Wiley & Sons Ltd., England, pp. 213-219, 1995.

34. *Reserve Bank of Australia Bulletin*, “Australian Financial Markets”, ISSN 0725-0320, Australia, May 1996.

35. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., “Learning Internal Representations by Error Propagation”, *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge Mass., 1986.

36. Schawrtz, T. J., AI Applications on Wall Street, *IEEE Expert*, pp. 69–70, Feb. 1992.

37. Schoneburg, E., Stock Prediction Using Neural Networks: A Project Report, *Neurocomputing Vol. 2, No. 1*, 17–27, June 1990.

38. Sheen, J., “Modeling the Floating Australian Dollar: Can the Random Walk be Encompassed by a Model Using a Permanent Decomposition of Money and Output?”, *Journal of International Money and Finance*, vol. 8, pp. 253-276, 1989.

39. Sinha, T. and Tan, C. , “Using Artificial Neural Networks for Profitable Share Trading”, JASSA: *Journal of the Security Institute of Australia*, Australia, September 1994.

40. Steiner, M. and Wittkemper, H., “Neural Networks as an Alternative Stock Market Model”, *Neural Networks in the Capital Market edited by Refenes, A.*, ISBN 0-471-94364-9, John Wiley & Sons Ltd., England, pp. 137-148, 1995.

41. Steurer, E., “Nonlinear Modeling of the DEM/USD Exchange Rate”, *Neural Networks in Capital Markets edited by Refenes*, A., John Wiley and Sons, England, ISBN 0-471-94364-9, pp. 199-212, 1995.

42. Surajaras, P. and Sweeney, R. J., “Profit-Making Speculation in Foreign Exchange Markets, *The Political Economy of Global Interdependence*, Westview Press, Boulder, 1992.

43. Sweeney, R. J., “Beating the Foreign Exchange Market”, *The Journal of Finance*, 41:163-182, Vol. XLI, No. 1, USA, March 1986

44. Tan, C. N. W., “Incorporating Artificial Neural Network into a Rule-based Financial Trading System”, *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES)*, University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993a.

45. Tan, C. N. W., “Trading a NYSE-Stock with a Simple Artificial Neural Network-based Financial Trading System”, *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES)*, University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993b.

46. Tan, C.N.W., Wittig, G. E., A Study of the Parameters of a Backpropagation Stock Price Prediction Model, *The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES)*, University of Otago, Dunedin, New Zealand, November 24-26, 1993, IEEE Computer Society Press, ISBN 0-8186-4260-2, 1993a.

47. Tan, C.N.W., Wittig, G. E., Parametric Variation Experimentation on a Backpropagation Stock Price Prediction Model, *The First Australia and New Zealand*

Intelligent Information System (ANZIIS) Conference, University of Western Australia, Perth, Australia, December 1-3, 1993, IEEE Western Australia Press, 1993b.

48. Tan, C., "Using Artificial Neural Networks as a Financial Trading Tool: A Foreign Exchange Market Example with Transactions Costs", *Abstracts of INFORMS '95*, Singapore, June 1995a.

49. Tan, C., "Applying Artificial Neural Networks in Finance: A Foreign Exchange Market Trading System Example with Transactions Costs", *Proceedings of the Ph.D. Conference in Economics and Finance*, Perth, Western Australia, November 1995b.

50. Taylor M. P. and Allen H., "The Use of Technical Analysis in the Foreign Exchange Market", *Journal of International Money and Finance*, vol. 11, pp. 304-314, 1992.

51. Tease, W. J., "Speculative Efficiency and the Exchange Rate: Some Evidence Since the Float", *Economic Record*, vol. 64, pp. 2-13, 1988.

52. Tsoi A. C., Tan, C. N. W., Lawrence, S., "Financial Time Series Forecasting: Application of Artificial Neural Network Techniques", *1993 International Symposium on Nonlinear Theory and its Applications*, Hawaii, USA, December 5-10 1993, Publication of Proceedings TBA, 1993a.

53. Tsoi A. C., Tan, C. N. W., Lawrence, S., "Financial Time Series Forecasting: Application of Recurrent Artificial Neural Network Techniques", 1993b *First International Workshop Neural Networks in the Capital Markets*, November 18-19, 1993, London Business School, London , United Kingdom,. Publication of Proceedings TBA, 1993b.

54. Turnovsky, S. J. and Ball, K. M., "Covered Interest Parity and Speculative Efficiency: Some Empirical Evidence for Australia", *Economic Record*, vol. 59, pp. 271-280, 1983.

55. Weigend, A. S., B. A. Huberman, and D. E. Rumelhart, "Predicting Sunspots and Exchange Rates with Connectionist Networks". In Nonlinear Modeling and Forecasting, edited by M. Casdagli and S. Eubank. Sante Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, Redwood City, CA, Addison-Wesley, pp. 395-432, 1992.

56. White, H., "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns", *IEEE International Conference on Neural Networks vol. 2*, 1988.

57. Widrow, B., Rumelhart, D. E. and Lehr, M. A., *Neural Networks: Applications in Industry, Business and Science*, Journal A, vol. 35, No. 2, pp. 17-27, July 1994.

58. Wong, F. and Tan, C., "Hybrid Neural, Genetic and Fuzzy Systems", *Chapter 14: Trading On The Edge: Neural Genetic and Fuzzy Systems for Chaotic Financial Markets edited by G. J. Deboeck*, ISBN 0-471-31100-6, John Wiley and Sons Inc., USA, pp. 243-261, 1994.

59. Yamaguchi, T. and Tachibana, Y., A Technical Analysis Expert System with Knowledge Refinement Mechanism, Proceedings The 1st International Conference on Artificial Intelligence Applications on Wall Street, NY, pp. 86–91, 1991.

5.14 Appendix C: Introduction to Foreign Exchange Trading Techniques

The techniques applied by foreign exchange traders broadly fall into two main categories, technical analysis and fundamental analysis. The trading system that is described in this chapter falls under the technical analysis technique.

5.14.1 Technical Analysis

Technical analysis (TA) is defined as the study of *market (price) action*[41] for the purpose of forecasting future price trends [Murphy 1986]. It is probably the most widely used decision-making tool for traders who make multi-million dollar trading decisions. According to Davidson [1995], the Bank of England reported in its quarterly bulletin in November 1989 that 90% of foreign exchange dealing institutions uses some form of charting or technical analysis in foreign exchange trading with two thirds claiming charts are as important as fundamentals for short-term forecasting (intraday to one week). He concludes that, since intraday traders account for 90% of the foreign exchange volume, technical analysis plays an important role in decision making in the market.

One of the reason for TA's popularity is that it forces a discipline and control on trading by providing traders with price and profit/loss objectives before trades are made. It is also a very useful tool for short-term as well as long-term trading strategies as it does not rely on any information other than market data. Another reason for its popularity is that, while its basic ideas are easy to understand, a wide variety of trading strategies can be developed from these ideas.

Currently the major areas of technical analysis are:

Charting: The study of price charts and chart patterns; e.g. trendlines, triangles, reversal patterns, and Japanese candlesticks.

Technical/Statistical Indicators: The study of technical indicators; e.g., momentum, relative strength index (RSI), stochastic and other oscillators.

Trading Systems: Developing computerized or automated trading systems, as well as mechanical trading systems, ranging from simple systems using technical indicators with a few basic rules (to generate trading signals such as moving averages) to complex rule-based systems incorporating soft computing methods such as artificial neural networks, genetic algorithms, and fuzzy logic. The traditional trading systems are based on rigid rules for entering and exiting the market. The main advantage of these systems is that they impose discipline on traders using them to be discipline.

Esoteric methods e.g. Elliot Waves, Gann Lines, Fibonacci ratios, and astrology.

[41] Although the term "price action" is more commonly used, Murphy [1986] feels that the term is too restrictive to commodity traders who have access to additional information besides price. As his book focuses more on charting techniques for commodity futures market, he uses the term "market action" to include price, volume and open interest and it is used interchangeably with "price action" throughout the book.

Murphy [1986] summarizes the basis for technical analysis into the following three premises:

Market action discounts everything. The assumption here is that the price action reflects the shifts in demand and supply which is the basis for all economic and fundamental analysis and everything that affects the market price is ultimately reflected in the market price itself. Technical analysis does not concern itself in studying the reasons for the price action and focuses instead on the study of the price action itself.

Prices move in trends. This assumption is the foundation of almost all technical systems that try to identify trends and trading in the direction of the trend. The underlying premise is that a trend in motion is more likely to continue than to reverse.

History repeats itself. This premise is derived from the study of human psychology which tends not to change over time. This view of behavior leads to the identification of chart patterns that are observed to recur over time, revealing traits of a bullish or a bearish market psychology**.**

5.14.2 Fundamental Analysis

Fundamental analysis studies the effect of supply and demand on price. All relevant factors that affect the price of a security are analyzed to determine the intrinsic value of the security. If the market price is below its intrinsic value then the market is viewed as undervalued and the security should be bought. If the market price is above its intrinsic value, then it should be sold.

Examples of relevant factors that are analyzed are financial ratios; e.g. Price to Earnings, Debt to Equity, Industrial Production Indices, GNP, and CPI. Fundamental analysis studies the causes of market movements, in contrast to technical analysis, which studies the effect of market movements. Interest Rate Parity Theory and Purchasing Power Parity Theory are examples of the theories used in forecasting price movements using fundamental analysis.

The problem with fundamental analysis theories is that they are generally relevant only in predicting longer trends. Fundamental factors themselves tend to lag market prices, which explains why sometimes market prices move without apparent causal factors, and the fundamental reasons only becoming apparent later on. Another factor to consider in fundamental analysis is the reliability of the economic data. Due to the complexity of today's global economy, economic data are often revised in subsequent periods therefore posing a threat to the accuracy of a fundamental economic forecast that bases its model on the data. The frequency of the data also poses a limitation to the predictive horizon of the model.

5.14.3 ANNs and Trading Systems

Today there are many trading systems being used in the financial trading arena with a single objective in mind; that is; to make money. Many of the trading systems currently in use are entirely rule-based, utilizing buy/sell rules incorporating trading signals that are generated from technical/statistical indicators such as moving averages, momentum, stochastic, and relative strength index or from chart patterns formation such as head and shoulders, trend lines, triangles, wedge, and double top/bottom.

The two major pitfalls of conventional rule-based trading systems are the need for an expert to provide the trading rules and the difficulty of adapting the rules to changing market conditions. The need for an expert to provide the rules is a major disadvantage in designing a trading system as it is hard to find an expert willing to impart his/her knowledge willingly due to the fiercely competitive nature of trading. Furthermore, many successful traders are unable to explain the decision-making process that they undergo in making a trade. Indeed, many of them just put it down to *'gut feel'*[42]. This makes it very difficult for the knowledge engineer[43] to derive the necessary rules for the inference engine[44] of an expert system to function properly.

The inability to adapt many rule-based systems to changing market conditions means that these systems may fail when market conditions change; for example, from a trending market to a non-trending one. Different sets of rules may be needed for the different market conditions and, since market are dynamic, the continuous monitoring of market conditions is required. Many rule-based systems require frequent optimization of the parameters of the technical indicators. This may result in curve fitting of the system.[45]

ANNs can be used as a replacement of the human knowledge engineer in defining and finding the rules for the inference engine. An expert's trading record can be used to train an ANN to generate the trading rules [Fishman 1991]. ANNs can also be taught profitable trading styles using historical data and then used to generate the required rules. In addition, they can learn to identify chart patterns, thereby providing valuable insight for profitable trading opportunities. This was demonstrated by Kamijo and Kanigawa [1990] who successfully trained a neural network to identify triangular patterns of Japanese candlestick charts.

Finally, ANNs which are presented with fundamental data can find the rules that relate these fundamental data (such as GNP, interest rates, inflation rates, unemployment rates, etc.,) to price movements. Freisleben [1992] incorporated both technical and fundamental analysis in his stock market prediction model while Kimoto and Asakawa [1990] used fundamental/economic data such as interest rate and foreign exchange rate in their forecasting model. The research reported in this book incorporates technical analysis into an ANN, to the extent that it incorporates historical price data and a statistical value (from the AR model).

5.14.4 Basic Structure of a Rule-based Financial Trading System

The two possible trading actions and the associated minimum basic rules for a financial trading system are:

[42] It is interesting that some recent studies have linked the neurons in the brain to activities in the stomach. Therefore, the term 'gut feel' may be more than just a metaphor!

[43] A knowledge engineer is a term used to describe expert system computer programmers. Their job function is to translate the knowledge they gather from a human expert into computer programs in an expert system.

[44] The inference engine is a computer module where the rules of an expert system are stored and used.

[45]A system is said to be curve fitting if excellent results are obtained for only a set of data where the parameters have been optimized but is unable to repeat good results for other sets of data.

Opening a position:

Buy rule

b. Sell rule

Closing a position

a. Stop/Take Profit rule

According to R. S. Freedman [Freedman 1991], the two general trading rules for profiting from trading in securities markets are:

i Buy low and sell high.

ii Do it before anyone else.

Most trading systems are trend following systems, e.g., moving averages and momentum. The system works on the principle that the best profits are made from trending markets and that markets will follow a certain direction for a period of time. This type of system will fail in non-trending markets. Some systems also incorporate trend reversal strategies by attempting to pick tops or bottoms through indicators that signal potential market reversals. A good system needs to have tight control over its exit rules that minimize losses while maximizing gains.

5.14.4.1 Opening Position rules

Only one of the following rules below can execute for a specific security at any one time, thus creating an open position. None of these rules can be executed for a security that has an existing open position. A position is opened if there is a high probability of a security price trending. A position is said to be *open* if either a buy or a sell rule is triggered.

a. Buy Rule

This rule is generated when the indicators show a high probability of an increase in the price of the security being analyzed. Profit can be made by buying the security at this point in time and selling it later after the security price rises. Buying a security opens a *long* position.

b. Sell Rule

This rule is generated when the indicators show a high probability of a drop in price of the security being analyzed. Profit can be made by selling the security at this point in time and buying it later after the security price declines. Selling a security opens a *short* position.

5.14.4.2 Closing Position rules

A position can only be closed if there is an open position. A position is closed if there is a high probability of a reversal or ending of a trend.

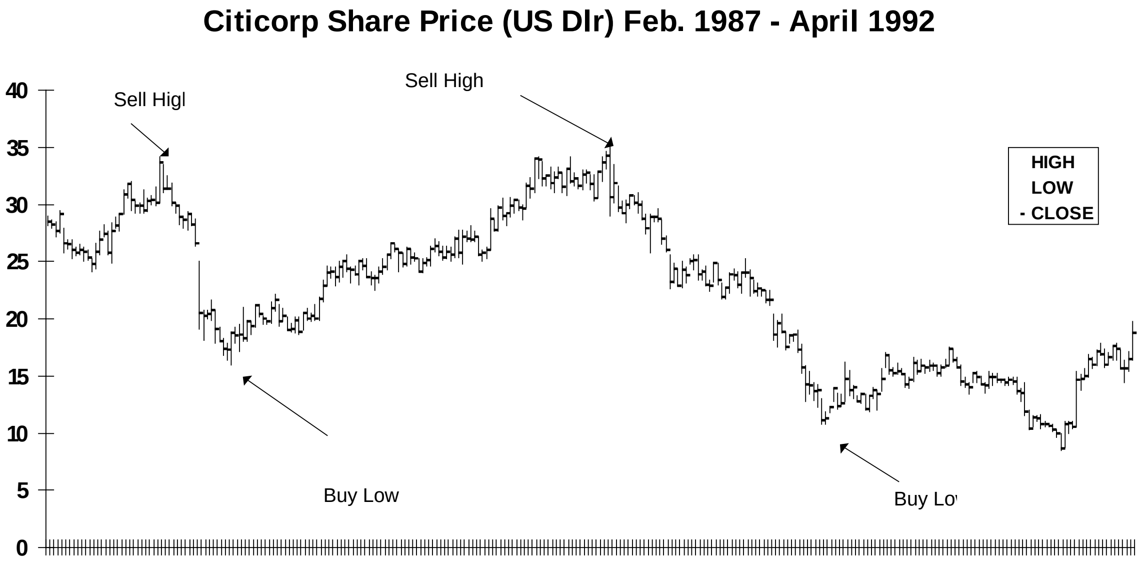
a. Stop/Take Profit rule

This rule can only be generated when a position (either long or short) has been opened. It is generated when indicators show a high probability of a reversal in trend or a contrary movement of the security price to the open position. It can also be generated if the price of the security hits a certain level thus causing the threshold level of loss tolerance to be triggered.

Systems that set a profit-taking target when a position is open call the closing position rule, a take-profit rule, while systems that place stops on an open position call the closing position rule a stop loss rule.

Chart 5-11 is an example of the technical charts analyzed by traders for pattern formations such as head and shoulders, triangles, and trend lines. The main components of the chart are the high, low and closing price of the security plotted against time. Sometimes the opening price and volume of transactions completed are also plotted. For a profitable trade to be made, it is obvious that one needs to buy when the price has bottomed out and sell when the price has topped out.

Chart 5-11
A Typical Technical Price Chart



5.14.5 Selection of Indicators/Data Input to ANN

The selection of technical and economic indicators/data to be used will depend on the following factors:

i. *Availability:*
The data must be easily obtainable.

ii. *Sufficiency of the historical databases:*
There must be enough sample data for the ANN learning and system testing process.

iii. *Correlation of the indicators to the price:*
The data should have some relevancy to the price of the security (whether it is lagging, leading, coincidental or noise).

iv. *Periodicity of the data:*
The data must be available in a predictable frequency (quarterly, monthly, weekly, yearly).

v. *Reliability of the data:*
The fast changing pace of today's global financial world and the increased in financial market volatility has resulted in difficulty to obtain reliable economic

data. This results in economic bodies having to frequently revise their data. Thus, if a price forecasting model is built on revised historical input data, the model's immediate forecast may not be reliable as the new data that is fed into the model will probably be erroneous.

Two sets of historical data are used. The first set is used to train the ANN to develop trading strategies and generate rules. The second set is used to test the profitability and reliability of the system. The system developer must be careful not to use the second set as training data inadvertently by modifying the system if it performs badly on the second set of data.