1-1-1999

An adaptive hierarchical fuzzy logic system for modelling and prediction of financial systems

Mark Kingham
Edith Cowan University

1999

An adaptive hierarchical fuzzy logic system for modelling and prediction of financial systems

Mark Kingham
Edith Cowan University

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Declaration

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Mark Kingham

31 December 1998.

AN ADAPTIVE HIERARCHICAL FUZZY LOGIC SYSTEM FOR MODELLING AND PREDICTION OF FINANCIAL SYSTEMS

By

Mark Kingham $_{\text{BAppSci(IS) (ECU)}}$

A Thesis Submitted in Partial Fulfilment of the Requirements for the Award of

Master of Science (Computer Science)

at the Faculty of Science and Technology,
Edith Cowan University.

ABSTRACT

In this thesis, an intelligent fuzzy logic system using genetic algorithms for the prediction and modelling of interest rates is developed. The proposed system uses a Hierarchical Fuzzy Logic system in which a genetic algorithm is used as a training method for learning the fuzzy rules knowledge bases.

A fuzzy logic system is developed to model and predict three month quarterly interest rate fluctuations. The system is further trained to model and predict interest rates for six month and one year periods. The proposed system is developed with first two, three, then four and finally five hierarchical knowledge bases to model and predict interest rates.

A Feed Forward Fuzzy Logic system using fuzzy logic and genetic algorithms is developed to predict interest rates for three months periods. A back-propagation Hierarchical Neural Network system is further developed to predict interest rates for three months, six months and one year periods. These two systems are then compared with the Hierarchical Fuzzy Logic system results and conclusions on their accuracy of prediction are compared.

ACKNOWLEDGMENTS

I would like to thank the support of my supervisors, Dr Masoud Mohammadian of University of Canberra and Dr Jim Millar of Edith Cowan University for their ongoing support and encouragement.

I would also like to thank my friends for their support during the last few years. Their friendship and encouragement helped me through this thesis.

Finally, but definitely not least, I want to thank my family. Their support and encouragement was always there, helping me to overcome the frustration's that are unavoidable during any intensive work. They can always make me see the lighter side of things and cheer me up. Without their support, none of this would have been possible.

Mark Kingham, December 1998.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Publications

Mohammadian, M., Kingham, M. & Bignall, B. (1998). "Hierarchical and Feed Forward Fuzzy Logic Systems for Interest Rate Prediction", *Journal of Computational Intelligence in Finance*, May-June, Vol 6, No. 3, pp 5-12.

Mohammadian, M., Kingham, M. & Stonier, R.J. (1998). "Prediction of Interest Rate using Hierarchical Fuzzy Logic and Neural Networks", International Conference On Intelligent Systems – ICIS'98, Paris, France, July 1-3, 1998

Kingham, M & Mohammadian, M. (1997). "Hierarchical and Feed Forward Fuzzy Logic For Financial Modelling and Prediction", Artificial Intelligence 1997 (AI'97), Perth, Australia, December 1-4, 1997.

Mohammadian, M., Kingham, M. & Hassan, M. (1997). "Adaptive Holding Policies For IP Datagrams Over ATM Networks Using Fuzzy Logic and Genetic Algorithms", Singapore Int. Conf. On Intelligent Systems (SPICIS '97), Singapore, February 24 -27, 1997.

Mohammadian, M., Nainar, I. & Kingham, M. (1997). "Supervised and Unsupervised Fuzzy Concept Learning", 2nd ICSC Int. Sym. on Fuzzy Logic & Apps. Zurich, Switzerland, February 12-14, 1997 .

Kingham, M & Mohammadian, M. (1996). "Financial Modelling and Prediction of Interest Rate using Fuzzy Logic and Genetic Algorithms", Australian & New Zealand Conference on Intelligent Information Systems (ANZIIS'96), Adelaide, Australia, November 18-20, 1996

Chapter 1 Introduction

1.1 Introduction

The problem of modelling and predicting uncertain dynamic systems which are subject to external disturbances, uncertainty and sheer complexity is of considerable interest. Conventional modelling and prediction methods involve the construction of mathematical models describing the dynamic system to be controlled and the application of analytical techniques to the model to derive prediction and control laws (Vidyasagar, M. 1978, Kosko, B 1992). These models work well provided the system does meet the requirements and assumptions of synthesis techniques. However, due to uncertainty or sheer complexity of the actual dynamic system, it is very difficult to ensure that the mathematical model does not break down.

Fuzzy logic control is an active research area (Kosko, 1992, Ross, T. J. 1995, Karr 1994). It has been found useful when the process is either difficult to predict or difficult to model by conventional methods. Fuzzy modelling or fuzzy identification has numerous practical applications in control, prediction and inference (Zadeh. L. 1965, Kosko, B. 1992). The majority of fuzzy logic systems to date have been static and based upon knowledge derived from imprecise heuristic knowledge of experienced operators, and where applicable also upon physical laws that governs the dynamics of the process (Kosko, B. 1992, Ross, T. J. 1995).

Although its application to industrial problems has often produced results superior to classical control (Lee, C. C. 1990, Cox, E, 1993), the design procedures are limited by the heuristic rules of the system. It is simply assumed that a significant process change does not occur that is outside the fuzzy knowledge based system. This implicit assumption limits the application of fuzzy logic to the case of normal working conditions for which the fuzzy knowledge based system is capable of handling. To accommodate abnormal working conditions, however, adaptive functions should be introduced to adjust the parameters of a fuzzy control system to meet the unexpected case that may exist in the real world (Cox, E. 1993, Ross, T. J. 1995).

Designers who use fuzzy logic to develop sophisticated control systems are finding support from a related technology. By including adaptive learning techniques in their arrangements, they are able to design systems that can adjust to environmental changes - a critical factor in a wide array of applications (Karr, C. 1994, E. Cox 1993).

These adaptive systems obtain their power from, on the one hand, their ability to learn and on the other, from their capacity to be modified and extended. Striking such a balance between learned responsiveness and explicit human knowledge makes the system very robust, extensible, and suitable for solving a variety of problems.

Time series are a special form of data where past values in the series may influence future values, depending on the presence of underlying deterministic forces. These forces may be characterised by trends, cycles and non-stationary behaviour in the time series. Predictive models attempt to recognise the recurring patterns and nonlinear relationships. Whilst linear models, such as those based on regression techniques, have been the basis of traditional statistical forecasting models, their drawbacks have led to increased activity in nonlinear modelling.

Recently techniques from Artificial Intelligence fields such as Neural Networks (NNs), Fuzzy Logic (FL) and Genetic Algorithms (GA) have been successfully used in the place of the complex mathematical systems for forecasting of time series (Azoff, 1994, Bauer 1994). These new techniques are capable of responding quickly and efficiently to the uncertainty and ambiguity of the system.

Fuzzy logic controllers (Kingham, M. and Mohammadian, M., 1996. Azoff, E, 1994) and neural networks can be trained in an adaptive manner to map past and future values of a time series, and thereby extract hidden structure and relationships governing the data (Lapedes, A and Farber, R., 1987).

Investors and governments alike are interested in the ability to predict future interest rate fluctuations from current economic data. Investors are trying to maximise their gains on the capital markets, while government departments need to know the current position of the economy and where it is likely to be in the near future for the well being of a county's people.

1.2 Thesis Outline

Chapters 2 is an introduction to the concepts and methodology behind Artificial Intelligence and the Intelligent System areas of fuzzy logic, genetic algorithms and neural networks which are used in the thesis. The concepts and theory behind each system is described as well as some examples of real world applications.

Chapter 3 introduces the concepts of intelligent hybrid systems. These are systems that combine two or more of the intelligent systems together. These may be hybrid fuzzy logic/neural networks systems, hybrid genetic algorithm/neural networks systems or hybrid fuzzy logic/genetic algorithm systems.

Chapter 4 considers the development of hybrid fuzzy logic and genetic algorithm systems to create a fuzzy logic knowledge base. A genetic algorithm is used to find the rules used by the fuzzy logic system. This chapter also shows how the financial data can be used to predict the fluctuations of interest rates. The application of a hybrid fuzzy logic and genetic algorithm system for the prediction of quarterly interest rates in Australia is considered.

Chapter 5 introduces the Hierarchical Fuzzy Logic structure using a two layer model. The different relationships between the models are discussed and reviewed. The application of Hierarchical Fuzzy Logic for prediction of quarterly interest rates in Australia is considered. Comparison of the results from the single layer fuzzy logic system with the Hierarchical Fuzzy Logic system is made.

Chapter 6 considers a Feed Forward Fuzzy Logic System for interest rate prediction. It compares the results achieved with those from the Hierarchical Fuzzy Logic system for the prediction of quarterly interest rates in Australia.

Chapter 7 considers the use of neural networks instead of the hybrid fuzzy logic/genetic algorithm system previously developed for interest rate prediction. A Hierarchical Neural Network system is developed that includes all the economic indicators as used in the Hierarchical and Feed Forward Fuzzy Logic systems. Finally, performance analysis and comparisons between the Hierarchical Fuzzy Logic system, the Feed Forward Fuzzy Logic system and the Hierarchical Neural Network system is performed.

Chapter 8 considers the long term prediction of interest rates by increasing the forecast time to first six monthly then yearly time periods. Here again, Hierarchical and Feed Forward Fuzzy Logic systems are developed for prediction. The accuracy of these simulations is compared with those from the Hierarchical and the Feed Forward Fuzzy Logic systems for quarterly interest rate prediction. A Hierarchical Neural Network system for predicting six monthly then yearly time periods is then constructed and its long term results are compared to the Hierarchical and Feed Forward Fuzzy Logic systems.

Chapter 9 concludes this thesis and gives final conclusions and results about the research. The chapter also provides information about how this research may progress in future studies and what benefits this research may have on the results obtained by the system.

Chapter 2 Intelligent System Techniques

2.1 Introduction

This chapter introduces fuzzy logic, genetic algorithms and artificial neural networks. It provides information on the concepts and methodology of each of these AI techniques and how they relate to the task of modelling and prediction.

2.2 Fuzzy Systems

Fuzzy set theory was developed by Lotfi Zadeh in the late 1960's and early 1970's (Zadeh, 1965, 1973) in an attempt to simplify the extreme complexity associated with the more traditional mathematical processes. According to T. J. Ross, (Ross, 1995), fuzzy set theory provides a means for representing uncertainties. Fuzzy set theory can be used for modelling the kind of uncertainty associated with vagueness, imprecision and/or lack of information. The underlying power of fuzzy logic is its ability to represent imprecise values in an understandable form.

As an example, if we asked a number of people when a temperature changes from hot to cold, we would find that the people all have different ideas as to hot and cold. Traditional, crisp computing divides the temperature range into two distinctive parts, hot and cold. There is no overlapping between the two sets (see Figure 2.1).

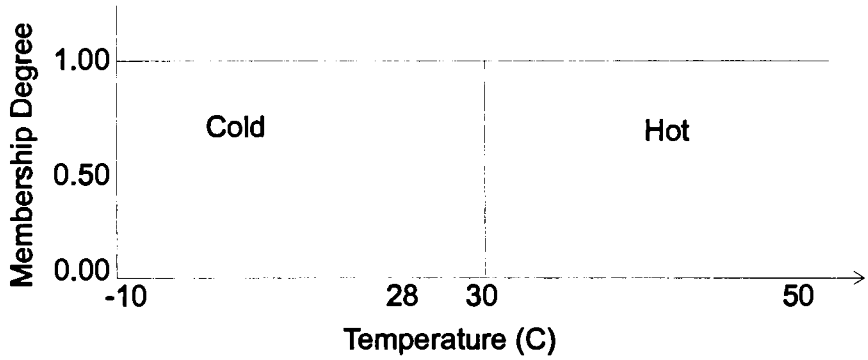

Figure 2.1 Traditional Sets for temperature

However, graphing the results would show us that there is no one place where a temperature changes from Hot to Cold. In fact, from asking this group of people, we find that the temperature ranges overlap each other (Figure 2.2).

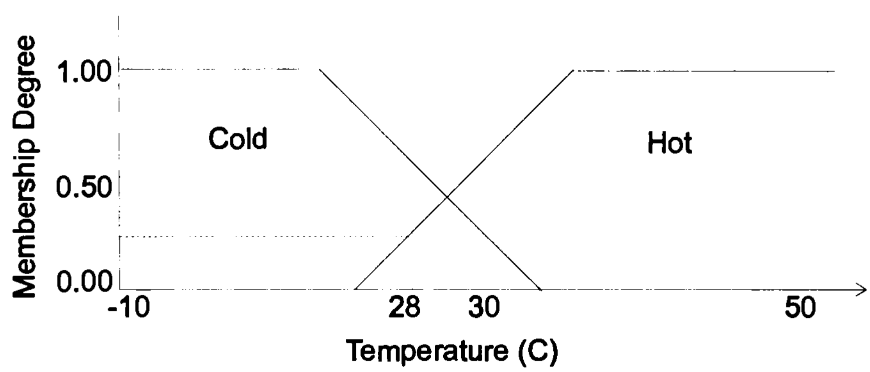

Figure 2.2 Simple Fuzzy Sets

While some people think 28 degrees is cold, others may think it is hot. There is no one stage in which the temperature changes from cold to hot, instead fuzzy logic assigns a degree of membership to each fuzzy set. From the above example, 28 degrees is seen to have a 0.5 membership to the set Cold while we have 0.25

membership to the set Hot. It is this simple idea that fuzzy logic is based on (Welstead 1994).

Ross (Ross, 1995) states that fuzzy logic systems are rule based systems which are capable of processing vague, imprecise data. In other words, fuzzy logic systems can be generalised as a form of rule based expert systems.

Below, we provide a mathematical framework for the impreciseness of fuzzy logic, an extension of the traditional Boolean (also called crisp or binary) logic.

2.2.1 Fuzzy Operations

Let X be a boolean (crisp) universe of discourse. A *fuzzy subset* A of X is shown by the *membership function*

$$\mu_A : X \to [0,1], \tag{2.1}$$

where for $x \in X$ the number $\mu_A(x)$ is interpreted as the degree of membership of x in the fuzzy set A, or as the truth value of the statement 'x is an element of A'.

A membership value of 0 means that x does not belong to the fuzzy set μ_A and a value of 1 means that x fully belongs to the fuzzy set μ_A. A value greater than 0 but less than 1 means that x partially belongs to the fuzzy set μ_A.

In classical set theory, where the output is boolean (or binary), the characteristic function of a set M is defined as

$$x_M : X \rightarrow \{0, 1\} \qquad (2.2)$$

$$x = 0 \quad \text{if } x \notin M$$

$$x = 1 \quad \text{if } x \in M$$

In order to calculate with fuzzy sets, we must generalise the set theoretical operations like *intersection* and *union*. These are given by

$$A \cap B = \{ x \in X \mid x \in A \wedge x \in B\} \qquad (2.3)$$

$$A \cup B = \{ x \in X \mid x \in A \vee x \in B\} \qquad (2.4)$$

In order to generalise these forms, we need triangular norms and conorms. (Terano, Asai and Sugeno, 1994). A triangular norm (or *t*-norm) is a binary operation on the unit interval

$$T : [0,1]^2 \rightarrow [0,1] \qquad (2.5)$$

which is commutative, associative, monotonic in both components and satisfies the boundary condition

$$T(x, 1) = x \qquad (2.6)$$

Similarly, if T is a *t*-norm, then the dual triangular conorm (*t*-conorm) $S : [0,1]^2 \rightarrow [0,1]$ is defined by

$$S(x,y) = 1 - T(1-x, 1-y) \qquad (2.7)$$

There are infinitely many *t*-norms and *t*-conorms, only a few of which are used in applications (Terano, Asai and Sugeno, 1994). The four most important representations of *t*-norms are

1. Minimum $$T_M(x, y) = \min(x, y) \tag{2.8}$$

which was introduced by L. Zadeh (Zadeh, 1965).

2. Lukasiewicz $$T_L(x, y) = \max(x + y - 1, 0) \tag{2.9}$$

was introduced by J. Lukasiewicz (Terano, Asai and Sugeno, 1994).

3. Product $$T_P(x, y) = x \bullet y \tag{2.10}$$

which is used in many areas due to its smoothness.

4. Drastic Product $$T_W(x, y) = \begin{cases} y & \text{if } x = 1 \\ x & \text{if } y = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

which is also called the Weakest *t*-norm.

According to McNeill and Freiberer (McNeill and Freiberer, 1993), it can be proven that T_M is the biggest and T_W is the smallest *t*-norm.

Given a *t*-norm T, *t*-conorm S, and fuzzy subsets A, B of the universe X, the membership functions of the intersection $A \cap B$, the union $A \cup B$, and the complement $\overline{A}$ are given by

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)) \tag{2.12}$$

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)) \tag{2.13}$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \tag{2.14}$$

These values describe the truth values of the statements

'x is an element of A AND x is an element of B',

'x is an element of A OR x is an element of B',

'x is NOT an element of A',

respectively.

2.2.2 Fuzzy Expert Systems

Traditional expert systems are programs that simulate the reasoning, or knowledge, of a human expert or professional in a given domain. There are a number of reasons why this knowledge may need to be captured:

1. The knowledge that the expert has is rare,
2. The work place is hazardous to humans,
3. The work is repetitive and boring,
4. Many other reasons…

These traditional expert systems have usually used classical logic and set theory to manipulate information. These systems use rules in the following format

IF condition A AND condition B OR condition C THEN action D where A, B and C are classical sets defining some state and D is some form of output. The IF clause is called the *antecedent* and the THEN clause is called the *consequent*.

By combining many of these types of rules, a *knowledge base* of rules is developed. There are three parts to an expert system: a knowledge base, an inference engine, and some form of memory. The knowledge base contains all the knowledge that has been collected from the expert about the domain, usually in the form of IF-THEN rules (as above). The inference engine fires the appropriate rules from the knowledge base depending on the current problem to solve. The working memory stores the running solution to the problem.

Expert systems using these classical logic and set theory represented uncertainty with many methods, including Dempster-Schaffer theory, certainty factors and Bayesian and probabilistic schemes (Yager, 1983). Mamdami (Mamdami, 1993) proposed the idea of using fuzzy logic in a rule based system. The rules in a typical fuzzy expert system are usually in the form of:

IF speed is **fast** AND distance is **small** THEN brake is **fast**

where speed and distance are linguistic input variables, brake is a linguistic output variable and **fast** and **small** are fuzzy sets. As can be seen, these rules are very similar to the classical expert system rules in the knowledge base, the difference being we now use fuzzy sets and fuzzy set theory instead of classical boolean logic to infer an output from these rules. Linguistic terms are used to describe the inputs and outputs of the fuzzy logic system. These terms are designed to handle imprecise concepts such as **small, fast, old** or **high** using language based evaluations.

2.2.3 Fuzzification and Defuzzification

Input data to be processed by the fuzzy system is usually in a numerical form. Some examples are :

1. Height is 1.82 meters,
2. Vehicle speed is 34 kph,
3. Unemployment Rate is 9.2%

In order for the fuzzy system to process this data using the knowledge base rules, the data must be converted to degrees of membership of the fuzzy set in which the input belongs.

The following is a simple example of a fuzzy knowledge base rule:

IF temperature is **hot** THEN air conditioner setting is **cold**

The fuzzy rule above has one input (temperature) and one output (air conditioner setting). The fuzzy system receives data from (in this case) the temperature sensor in a numerical form. To use the above rule, this numerical information must be fuzzified by assigning it a degree of membership in the relevant set.

Takagi and Sugeno (1983) suggested another form of fuzzy If-Then rule where the premise consisted of one or more fuzzy antecedents, while the consequence was described with a non-fuzzy equation. For example:

IF pressure is **high** and temperature is **high** THEN force = x * (pressure)^2

While both forms of fuzzy rules have been used extensively in modelling and prediction, this thesis uses fuzzy rules that have the consequence based on a fuzzy linguistic label.

In Figure 2.3, we show the fuzzy sets for the temperature input. It has been split into three fuzzy sets, **Hot, Warm** and **Cold**, starting from 0° C to 40° C.

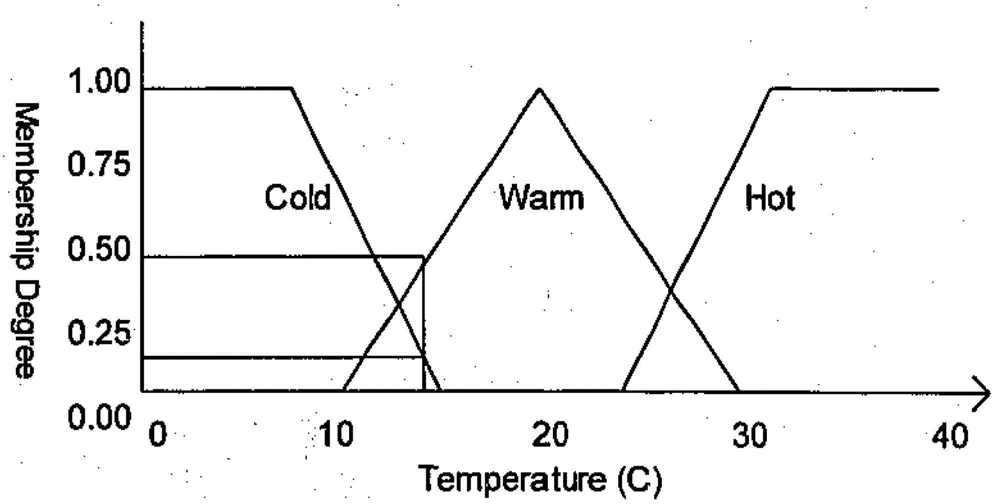

Figure 2.3 Temperature Fuzzy Sets

If the temperature was 14° C, as shown in the above figure, then it would be a member of the fuzzy set **Cold** by the extent (membership) of 0.1 and a member of the fuzzy set **Warm** by the extent of 0.45. This shows an important factor in fuzzy logic is that an input may belong to more than one set, even when the sets appear to be mutually exclusive under classical set theory.

2.2.4 Fuzzy Inferencing

Inferencing is where each rule of the fuzzy knowledge base is checked to see if it falls within the parameters required to fire, for example the rule is applicable to some degree. We first calculate the extent to which a rule fires (the amount of consequent of the rule that applies). There are a number of techniques that determine which rule to fire from the fuzzy knowledge base. In this thesis we use the Max-Min method, which takes the minium of the fuzzy rules fired. This is the union of the sets.

From the example in section 2.1.3, there is only one input so there are only a few rules for the system:

1. IF temperature is **hot** THEN air conditioner setting is **cold**
2. IF temperature is **warm** THEN air conditioner setting is **cool**
3. IF temperature is **cold** THEN air conditioner setting is **warm**

However most fuzzy systems have several inputs. This increases the size and complexity of the fuzzy knowledge base. As the number of inputs into the fuzzy system increases (and the number of fuzzy sets for input variables increases), the number of rules required increases exponentially (Raju, G.V.S. and Zhou, J. 1993). On a system that has a large number of inputs, this can quickly become a problem. For example,

1. IF temperature is **hot** AND humidity is **high** THEN sprinklers is **high**
2. IF temperature is **hot** AND humidity is **low** THEN sprinklers is **high**
3. IF temperature is **cold** AND humidity is **cold** THEN sprinklers is **low**
4. IF temperature is **cold** AND humidity is **medium** THEN sprinklers is **low**
5. ....

Figure 2.4 shows the fuzzy membership sets for the humidity input.

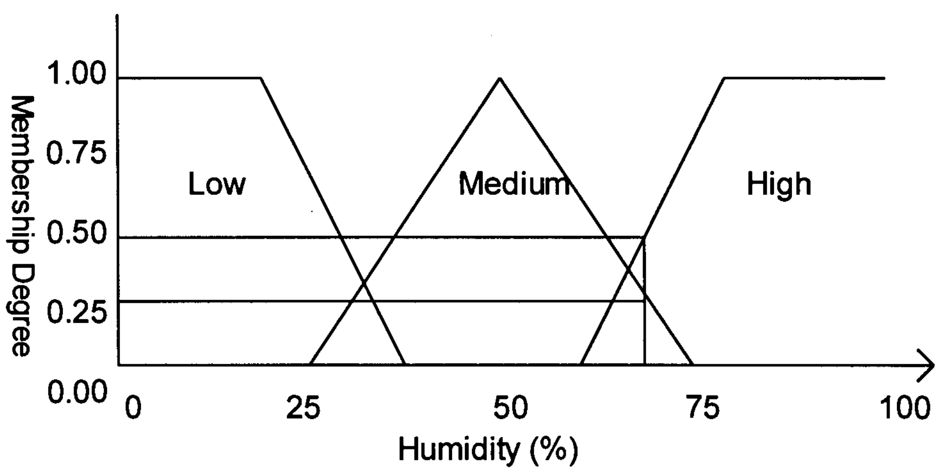

Figure 2.4 Humidity Fuzzy Sets

The figure below shows the process involved in developing a fuzzy inference system. The main components of a fuzzy logic system are:

1. A *rule base* containing a number of fuzzy If-Then rules.
2. A *data base* defining the fuzzy sets.
3. A *decision making unit* performing inference operations.

4. A *fuzzification unit* transforming crisp inputs into fuzzy memberships.

5. A *defuzzification unit* transforming the fuzzy results into crisp outputs.

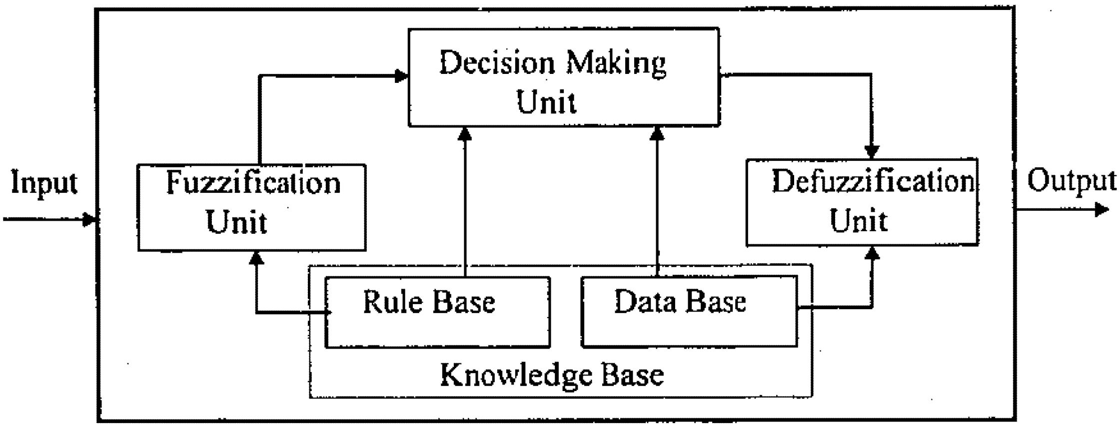

Figure 2.5 Fuzzy Inference System

The table below shows how the fuzzy rules for the system can be stored. The fuzzy rules are stored in a Fuzzy Knowledge Base (FKB). This FKB contains all the possible rules of the system.

Temperature

Humidity		Hot	Warm	Cold
	High	**High**	**High**	**High**
	Medium	**High**	**Medium**	**Low**
	Low	**Medium**	**Medium**	**Low**

Table 2.1 Fuzzy Knowledge Base

Table 2.1 shows the FKB rules for a fuzzy logic system which has two antecedents (the temperature and humidity) and the one consequent (sprinkler). Displaying a table with three inputs is represented as a cube. Higher numbers of inputs means a multi dimensional FKB, which are more difficult to show. Figure 2.6 below shows the fuzzy membership functions for the sprinkler output.

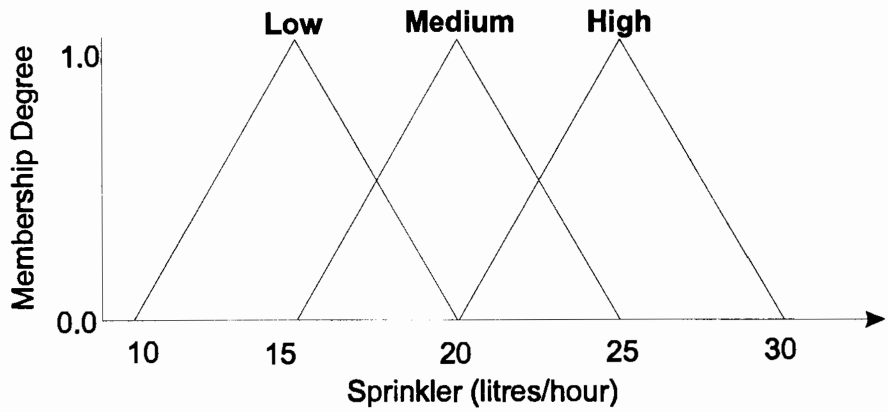

Figure 2.6 Sprinkler Outputs Sets

From the example above, if temperature falls in the fuzzy set **Hot** with a membership value 0.1 and humidity has a membership value of 0.5 in the fuzzy set **High**, then Rule 1 from the FKB would fire to an extent of *min*(0.1, 0.5) = 0.1. Rule 2 would fire to an extent of *min*(0.1, 0.25) = 0.1, while Rule 3 would fire *min*(0,0) = 0 as the membership of both antecedents to the rule is 0. If all the antecedents have a membership of 0 except for one, then the *max* would be taken, for example in Rule 4, temperature has a membership of 0 while humidity has a membership of 0.25, the rule would fire by the extent of *max*(0.0, 0.25).

To obtain an output from the fuzzy system, we must *defuzzify* the results. One common method of doing this is the *Centroid* defuzzification method (Kong, S and Kosko, B, 1990), also called centre of gravity. The output is obtained by finding the centre of gravity of the combined fuzzy output set. The figure below shows a representation of how this is achieved (Kosko, B. 1992).

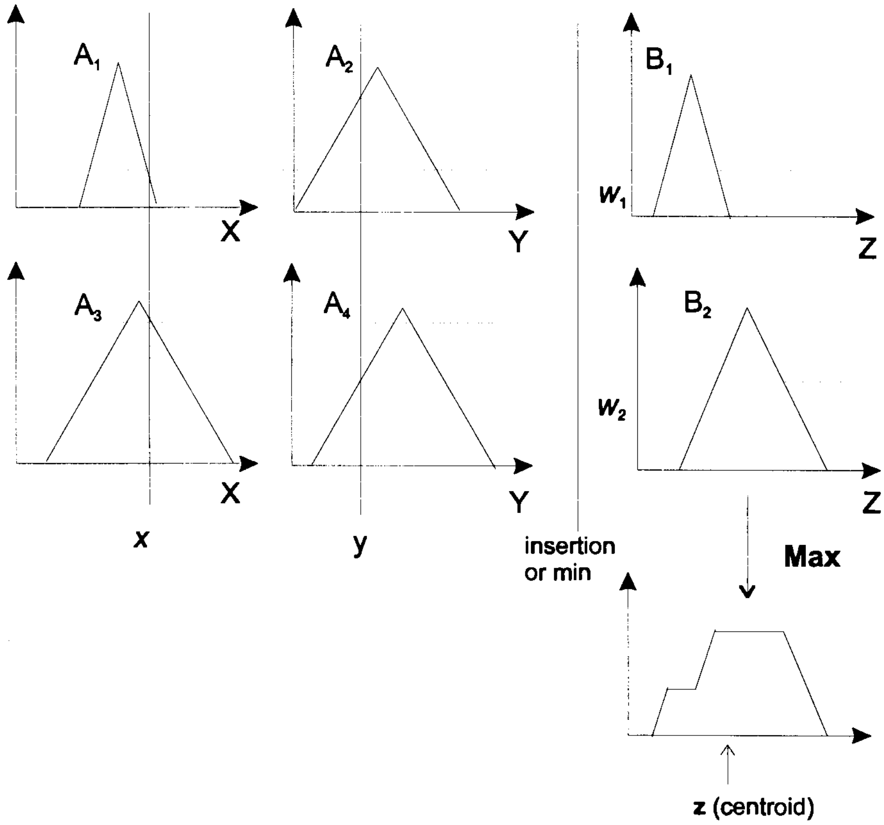

Figure 2.7 Centroid Defuzzification Method

Fuzzy systems have proved to be successful both in research and industry applications. It is fairly easy to design a fuzzy logic knowledge base with few inputs that provides adequate results most of the time, even in new or untested areas, however it is difficult to create FKB that display optimum results all the time.

Problems in complexity arise when the number of inputs to the fuzzy logic system is increased. This problem and a suggested solution are discussed in detail in later chapters.

2.3 Genetic Algorithm's

According to Goldberg (1989), Genetic Algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics. They are just one of a number of techniques which fall under the "Evolutionary Computing" paradigm. They are inspired by the process of Darwinian evolution which uses the principle of evolution through natural selection.

Although the ideas behind genetic algorithms have been around since Darwin and Mendel, it was not until practical computing power became available that the use of genetic algorithms becomes feasible. In 1975, John Holland published the book *Adaptation in Natural and Artificial Systems* (Holland, 1975), still regarded as one of the most important works in the genetic algorithm field. Since then, the genetic algorithm (or the larger Evolutionary Computing) field has developed with a number of annual conferences each year and a number of research groups around the world involved in such diverse areas as control, prediction, modelling, learning and cognitive science (for example, IEEE International Conference on Evolutionary Computing).

2.3.1 *Overview of Genetic Algorithms*

Some of the characteristics of this evolutionary process, also called "Survival of the Fittest" are (Davis, 1991):

1. Each individual tends to pass on its traits to its offspring,
2. However, individuals with different traits are produced by nature,
3. The fittest individuals, those with the most favourable traits for the current environment, tend to have more offspring than those with unfavourable traits. This drives the population as a whole towards favourable traits (Survival of the fittest),
4. Over a long period, variation can accumulate, producing entirely new species whose traits make them especially suited to particular environments.

Natural Evolution	Genetic Algorithm
genotype	coded string
phenotype	uncoded point (string)
chromosome	string individual
gene	string position
allele	value at certain position of a string
fitness	objective function table

Table 2.2 Analogies between natural evolution and the Genetic Algorithm paradigm

As parents that are better suited for the environment tend to have more offspring than those that are less suited, good traits are passed on to the next generation while poor traits gradually disappear.

Simply put, genetic algorithms solve a problem by generating, changing and evaluating a population of candidate solutions to the problem. Table 2.2 lists the analogies between natural evolution and the genetic algorithm paradigm (Goldberg, 1989):

The genetic algorithm cannot model the whole process of natural evolution as many of the factors in evolution are not known or understood. For example, the "fitness" of a chromosome (the DNA strand) in nature cannot be expressed by some value, it is more like a combination of strength, intelligence, health, etc.

The following shows the steps involved in a simple genetic algorithm.

1. Initialise and encode a random population of chromosomes (individuals),
2. Decode and evaluate the fitness of each chromosome in the population,
3. Produce a new population of strings by selecting current chromosomes as parents according to their fitness to generate new children,
4. Apply genetic operators (reproduction, crossover and mutation) to new chromosomes,
5. Repeat steps 2-4 until an adequate solution is found or for a certain number of generations.

Genetic Algorithms differ from traditional continuous optimisation methods, like gradient decent, in the following ways (Goldberg, 1989)

1. Genetic Algorithms manipulate coded versions of the problem parameters instead of the parameters themselves,
2. Most search methods start from a single point, genetic algorithms operate on a whole population of strings (points),
3. Genetic Algorithms use payoff (objective function) information, not derivatives or other auxiliary knowledge,
4. Genetic Algorithms use probabilistic transition operators (genetic operators) while traditional methods use deterministic rules.

As the standard genetic algorithm operations do not change, only the encoding of the chromosomes and the fitness operations change. Therefore, designers can concentrate on the domain dependant parts of the problem and let the genetic algorithm handle the search and optimisation routines.

2.3.2 Encoding and Genetic Operators

There are many ways in which to encode and initialise the chromosomes of the initial generation. Goldberg (Goldberg, 1989) uses simple structure chromosomes comprising of binary numbers and fixed size chromosomes. Other encoding methods such as *k*-ary codes, real (floating point) codes, permutation (order) codes, Lisp codes and variable length chromosomes have been reported as being successful by Davis (Davis, 1991).

For this introduction, simple binary numbers and a fixed length chromosome will be used. For the initial generation, the system randomly generates bits of each chromosome and encodes all the chromosomes as a population. The fitness of each chromosome in the population is evaluated and reproduction and selection is performed, based on the fitness value.

There are a number of ways to achieve effective selection and reproduction, including, ranking, tournament, and proportional schemes, but the most important objective is to give better individuals from the population a higher preference (Goldberg, 1989). The most common selection method of parents is the Roulette Wheel Parent Selection scheme. This method is described below:

1. Sum the fitness of all the chromosomes in the population.
2. Generate a random number between 0 and 1, and multiply by the sum of the fitness.
3. Return the first chromosome from the population whose fitness, accumulated from the first fitness of the population, is greater than or equal to the random number generated in the previous step.

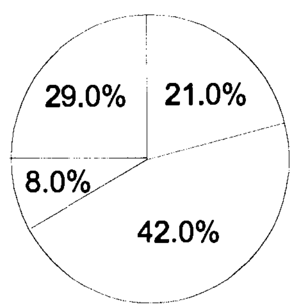

Figure 2.8 Roulette Wheel sized according to fitness

The Roulette Wheel parent selection method therefore returns a randomly selected parent whose chance of being selected is directly proportional to its fitness (see Figure 2.8).

The crossover operator is probably the most important of the genetic operators (Goldeberg 1994). Basically, crossover is the exchange of genes between the chromosomes of two parents. There are a number of different crossover operators, such as one-point, two-point, n-point and uniform crossover, with the simplest being the one-point crossover (Figure 2.9). There are three steps to the one-point crossover (Goldberg, 1989):

1. Two individuals are chosen from the population using a selection method,
2. A cross site along the individual length is chosen uniformly at random,
3. Values are exchanged between the two individuals following the cross site.

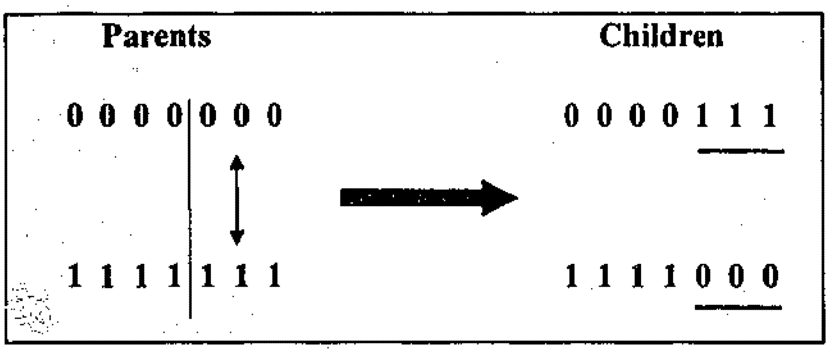

Figure 2.9 One-point crossover

After the crossover operation has taken place, these individuals would be placed in the new population. The crossover operation takes place until all the positions in the new population have been filled with offspring that have been constructed from selected parents.

The next genetic operator is the Mutation operator. The mutation operator is used to ensure genetic diversity on chromosomes and it avoids bits in a chromosome from becoming fixed at a certain position. It is performed by randomly selecting a position in the individual and altering the bit value (see Figure 2.10).

Mutation is used to maintain genetic diversity within a small population of individuals. There is a small probability that any allele in an individual will be flipped from its present value to another value within a specific range. This prevents certain alleles becoming fixed at a specific value due to every string in the population having that value, often a cause of premature convergence to a non-optimal solution.

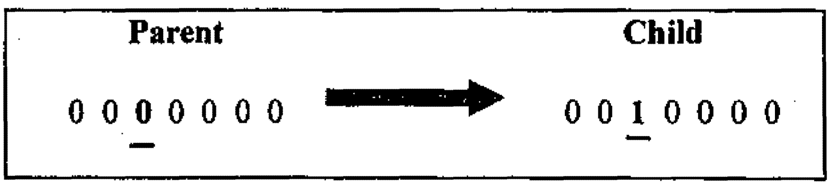

Figure 2.10 Mutation operator

Although selection and crossover provide most of the genetic algorithms search power, the mutation stops the genetic algorithm from becoming stuck in a local minimum and preventing premature convergence, a very important facet of optimisation (Goldberg, 1989).

2.4 Artificial Neural Networks

An Artificial Neural Network (ANN) can be considered to be a simplified mathematical model inspired by the way biological nervous systems, such as the brain, process information. Artificial neural networks are composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. The processing ability of the network is stored in the interconnected elements links as *weights*, obtained by a process of adaptation to, or learning from, a set of training patterns.

ANNs can learn from example in either a supervised or unsupervised manner. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. ANNs perform the same function by adjusting their weights.

The first forms of neural networks emerged in the early 1940's after the introduction of simplified neurons by McCullock and Pitts (McCullock and Pitts, 1943) and by work done by Hebb in the late 1940's (Hebb, 1949). Early work in the Artificial Intelligence field was divided between two camps, firstly, those that believed that intelligent systems could best be developed by modelling the biological representations of the brain, and secondly, those that believed that intelligence was fundamentally a symbol processing model, similar to the *von Neumann* (traditional) computer. Minsky and Papert (1969), summed up the general feeling of frustration (against neural networks) at the time by showing some of the deficiencies of the perceptron model. Funding was redirected into the symbol-process field and neural network research almost came to a complete stop, with only a few researchers remaining in the field.

This continued to be the case until the mid 1980's when a number of new theoretical developments rekindled interest in the neural networks field (most notably the discovery of error back-propagation algorithm). Today, the Artificial Neural Network field has many researchers from a number of different areas, including computer science, biology, economics and psychology and has increased funding for research, and also a number of annual conferences and journals (for example, IEEE International Conference on Neural Networks).

2.4.1 Biological Aspects of ANN's.

There are many theories about how the brain trains itself to process information. In the human brain, a *neuron* collects signals from other neurons through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. Figure 2.11 below shows the structure of a typical neuron (Chester, 1993).

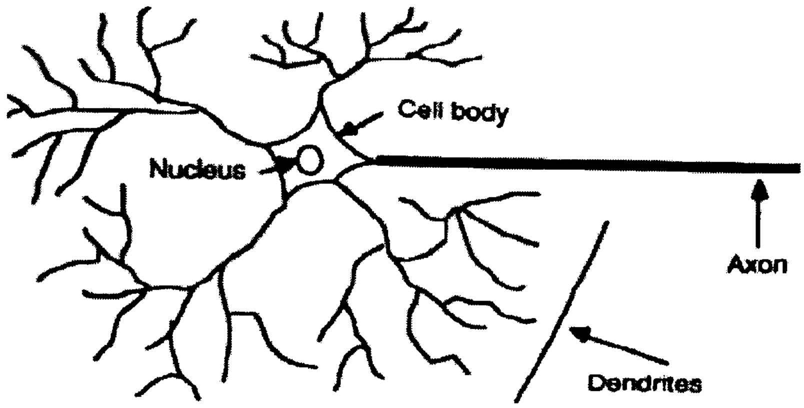

Figure 2.11 Structure of a Neuron

When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its own axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. The figure below shows the connection between neurons via axons to synapses.

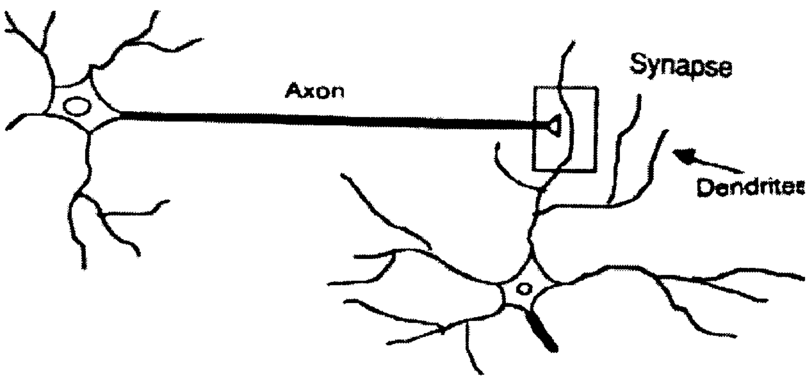

Figure 2.12 Synapse connection

The human brain is suspected to contain around 10^{11} neurons. Each neuron is connected to thousands of other neurons, and these are arranged in a rough layer like structure, as shown in Figure 2.13 below. The first few layers receive input from the senses, sight, smell, touch, taste and hearing, while the final layers produce some form of motor response, such as moving arms or legs. The layers in between these layers form the *associative cortex*. Biologists still have little understanding of how these inner layers work, but they believe the neurons in these layers are the most important part of the human brain, as it is these parts that are responsible for our conscious understanding of the world.

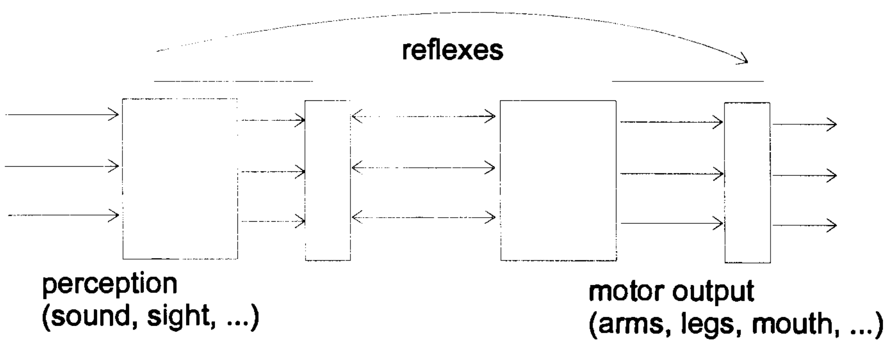

Figure 2.13 Neuron layers in the brain

Biologists believe that the brain learns in three ways:

1. By growing new axons,
2. By removing (killing off) old axons,
3. By changing the strength of existing axons.

The strength of an axon means that an axon must have some way to weight the signal passing along it. If it is larger than this weight, then the axon will pass the information on to the connected neuron, while if the signal is less than this weight, then the signal is not passed on (Gallant, S. 1993).

2.4.2 *Artificial Neural Networks*

An ANN is a model that emulates the biological neural network (the brain). Compared to even a simple biological brain, ANN's are still very primitive and limited in their power. However, they do provide an insight into how the biological brain works, or rather how researchers believe how the brain works, and as such can produce a number of productive and interesting systems that can be used in a number of different areas.

In ANN's, the basic processing elements are called *artificial neurons*, or simply *neurons* or *nodes*. These perform a similar task to the biological neuron in that the neurons communicate by sending signals to each other over a large number of weighted connections.

There are a number of different models of ANN's. Probably the most common (Rao & Rao, 1994) is the Feed Forward Neural Network. The Feed Forward ANN has:

1. An *input layer* where input patterns are presented to the network,
2. An *output layer* which contains the response of the network to the given input,
3. Zero or more *hidden layers* which lay between the input and output layers.

In Feed Forward networks, data flows from input to output neurons in a strictly forward method. The data processing can extend over a number of hidden layers of neurons, but there are no feed back loops. Figure 2.14 shows the structure of a Feed Forward Neural Network.

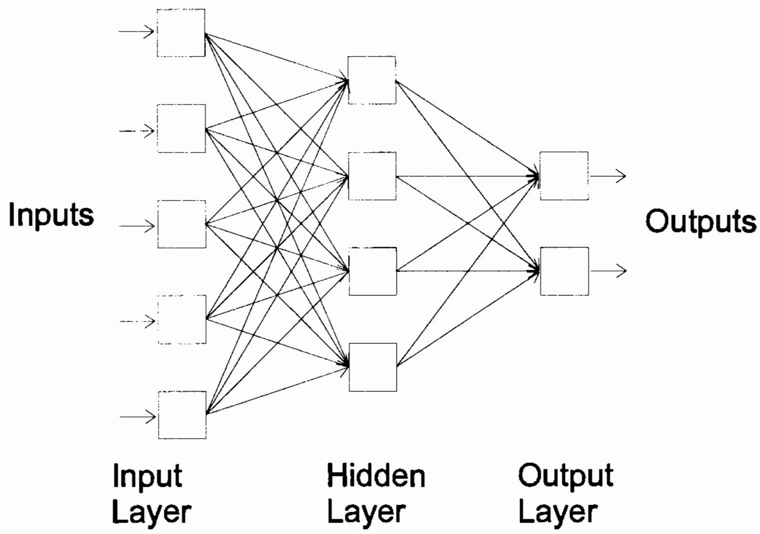

Figure 2.14 Feed Forward Neural Network

Each neuron (or processing unit) has:

1. An activity level which represents the state of polarisation of the neuron,
2. A set of input connections which represent the biological synapses and dendrites,
3. Some form of internal resting level,
4. The current state of activation,
5. A set of outputs from the neuron which represent a neurons axons.

Each processing unit receives input from other units and uses these to compute an output signal, which is propagated to other units. Figure 2.15 below shows an artificial neuron (processing unit).

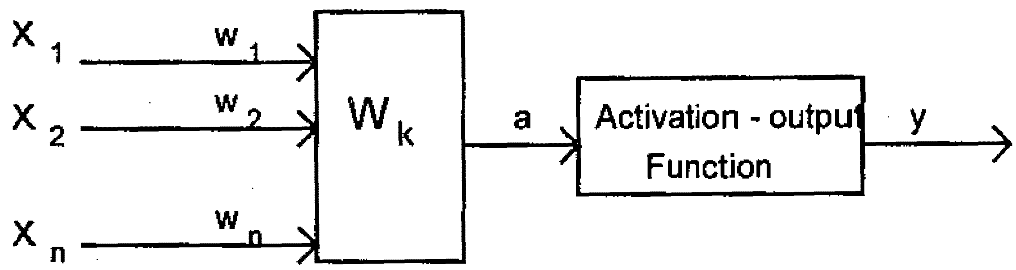

Figure 2.15 Artificial Neuron

2.4.3 Summation Function

The summation function finds the weighted average of all the input elements to each processing element. In an ANN neuron, there are *n* inputs with signals x_1 , x_2 ..., x_n and weights w_1 , w_2 ..., w_n. W_k is simply the weighted sum of the separate inputs into the neuron.

$$W_k = \sum_{i=0}^{n} w_i x_i \tag{2.15}$$

If W_k is positive then it is considered excitatory and if it is negative it is considered inhibitory. This is the internal activity level of the neuron. The most common activation level method is this linear weighted sum. Other methods may use thresholding or some form of non-linear activation, such as Boltzmann Machines and sigmoid functions (Grossberg, 1988).

2.4.4 Activation Function

The activation function (also called a transfer function) takes the activation level of the neuron and calculates its output. This is the signal it will send to other neurons. The activation function of *j*th node receives inputs (x_i) from other nodes. Each of these is multiplied by the corresponding synaptic weight (w_{ij}), and the resulting products are summed within the *j*th node to produce the activation, u. The activation is then transformed to produce the nodes output signal.

$$u_j = \sum w_{ij} x_i \tag{2.16}$$

There are a number of different activation functions (see Figure 2.16).

1. Hard limiting threshold,
2. Semi-Linear,
3. Nonlinear (such as sigmoid).

The Hard Limiting Threshold function is a simple step function. The Hard Limiting Threshold function is limited to output only binary numbers (0 or 1). According to Rumelhart and McClelland (Rumelhart and McClelland, 1986), this model has a number of problems, especially when dealing with multi layered neural networks. This is the type of activation function that McCullock and Pitts (Chester, 1993) used in their neural network models. This is also called a step or sign function.

$$f_i(x_i) = \begin{cases} 1 & x_i > 0 \\ 0 & x_i \leq 0 \end{cases} \tag{2.17}$$

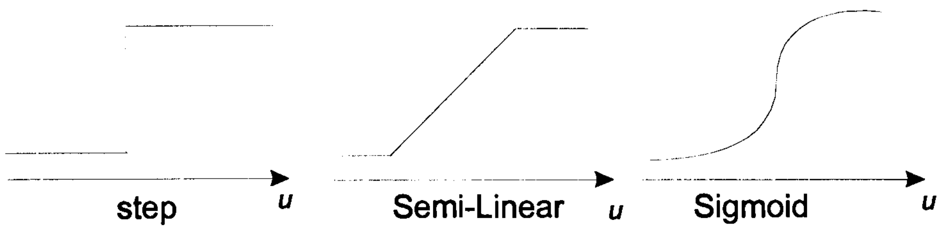

Figure 2.16 Different Activation Functions

The semi linear activation function is similar to the Hard Limiting Threshold function except there is a linear function between the boundaries. This allows the semi linear function to use values other than binary numbers,

$$f_i(x_i) = \begin{cases} 0 & x_i \le 0 \\ x_i & x_i \le a \\ 1 & x_i > a \end{cases} \tag{2.18}$$

Sigmoid functions are much more complex than the other activation functions as it is an increasing function and also continuous, that is it have a derivative at all points. The sigmoid activation function is also monotomically increasing, asymptotic to 0 and +1, as it arguments go to $-\infty$ and ∞, respectively.

$$f_i(x_i) = \frac{1}{1+e^{x_i}} \tag{2.19}$$

According to Chester (Chester, M. 1993), the activation functions purpose is to modify the output level to a reasonable value (for example, between 0 and 1). This transformation is performed before the output reaches any other neuron. The reason is that without this transformation, the value of the output may become very large, especially where there are a number of hidden layers.

2.4.5 Learning techniques

When a neural network is presented with a set of inputs, it needs to produce the desired output. There are a number of ways to train the network, one method is to set the weights of the connecting links in the network explicitly, using some form of *a priori* knowledge. Another method, more commonly used in ANN's is to train the network by showing it teaching patterns and letting it change its weights according to some learning algorithm.

There are two different ways in which the neural network can learn:

1. *Supervised learning*, where the network is trained by providing it with input and matching output patterns, provided by some form of knowledgeable teacher. The network compares its response with the desired response and modifies its weights in some way as to gradually output results similar to the desired output.
2. *Unsupervised learning*, where only the input patterns are shown to the network. This is commonly called a self-organising network as it organises its internal weights without any interaction with some form of teacher, that is no *a priori* information.

Both learning methods depend on modifying the weights of the connections between neurons using some learning rules. Two of the most common learning rules are:

1. *Hebbian Learning* where if two units are active at the same time, then the strength of the connections between them should be increased. For example if the neuron j receives input from neuron k, Hebbian learning prescribes to modify the weight w_{jk} with

$$\Delta w_{jk} = \gamma y_j y_k \tag{2.20}$$

where γ is a positive constant of proportionality representing the learning rate.

2. *Delta Rule Learning* where the difference between the actual and desired activation is used to adjust the weights

$$\Delta w_{jk} = \gamma y_j (d_k - y_k) \tag{2.21}$$

where d_k is the desired activation provided by a teacher. This is often called the *Widrow-Hoff* rule or the *Delta rule.*

2.4.6 Backpropagation

As stated earlier, Minsky and Papert (Minsky & Papert, 1969) showed that a single layer network has severe restrictions. They showed that a two layer feed-forward neural network can overcome many of these restrictions, however the problem of how to adjust the weights from input to hidden units could not be found. Rummelhart and McCelland (1986) presented a solution to this problem, with the main idea being that the errors of the units of the hidden layers are determined by backpropagating the errors of the units of the output layer. Since then, the backpropagation method has become the most widely implemented form of neural network (Welstead, 1994).

The main feature of the backpropagation model is the hidden layers of neurons. This allows the network to overcome the problems of the perceptron (Refenes, A. 1995). In this model, the network is fully connected, with every neuron in layer n-1 connected to every neuron in layer n. This was shown in the Feed Forward Neural Network in Figure 2.4. The backpropagation model is a supervised learning technique. The backpropagation method uses a gradient descent or steepest descent method to learn the connection weights, thus being a descendant of the Widrow-Hoff or delta rule.

Using a backpropagation algorithm, the weights of the connections are usually randomly set. The network inputs are then shown the training data. The neurons in the lower layers send impulses (output) to the next higher layer until they reach the output layer. The system then determines the amount of error in each neuron, which allows the connection weights then get updated based on the amount of error at the output layer. To do this, it uses the generalised delta rule.

The error found at the output layer is then propagated back through the hidden layers via the connections. The weights are then updated based on the generalised delta rule, continuing back through all the hidden layers. This process is continued until the system reaches some pre-determined measure of accuracy.

To determine how well the network is performing with the current weights, the error is summed over all the neurons:

$$E_p = \frac{1}{2}\sum_i (y_i - d_i)^2 \tag{2.22}$$

where y_j is the activity level of the jth unit in the top (input) layer and d_j is the desired output of the jth unit. The p subscript refers to the specific pattern shown to the input neurons.

To generalise this error over the whole training set, the average error is taken. This becomes the global error (GE) amount:

$$GE = \sum_p E_p \tag{2.23}$$

For learning to take place, the error measure (GE) given above must be minimised. This is achieved by continually changing the weights by an amount proportional to the derivative $\vartheta E / \vartheta W$. This is denoted by δ_i:

$$\Delta w_{ij}(t+1) = \lambda\, \delta_i y_{ij} \tag{2.24}$$

From above, the learning rate λ is the amount by which the global error is to be minimised during each training pass. It is kept constant for each training pass but may change after a training pass. The objective of this learning archetype is to find the Least Mean Square (LMS) error. This is achieved as the learning rate heads towards zero, a set of weights will be found that give the LMS error. According to

Chester (Chester, M. 1993), the value of δ_i is computed by differentiating the network error (equation 2.22) and the activation level of the neuron (equation 2.15):

$$\delta_i = (d_{j,c} - y_{j,c}) f^{*}(y_i) \tag{2.25}$$

where $y_{j,c}$ is the actual state of the output neuron j of pattern c and $d_{j,c}$ is its desired state.

Like any hill climbing technique, gradient descent has the problem of local minima rather than a global minimum. This means that the network may not find a valid solution to the problem, even if one does exist as it has converged in a local minimum.

2.4.7 Recall

Once the network has been trained to an acceptable accuracy, the weights are frozen This was the training phase. When a pattern is now presented to the inputs of the network, the response from the outputs will be based on what the network has learnt. It is important to recognise that the weights in the network will not be modified (Rao, V. and Rao, H. 1994).

Chapter 3 Hybrid Intelligent Systems

3.1 Introduction

In this chapter we consider hybrid intelligent systems and their importance and the different types of Hybrid systems. We focus on a hybrid system using a combination of genetic algorithm and fuzzy logic to create a Fuzzy Logic Knowledge Base. This is then applied to the development of a Fuzzy Logic Knowledge Base used in the prediction of interest rates in Australia.

3.2 Hybrid Systems

According to Goonatilake (Goonatilake, S and Treleaven, P. 1995), Intelligent Systems are a group of computing techniques that include neural networks, fuzzy logic and genetic algorithm's. These new techniques are capable of responding quickly and efficiently to the ambiguity and uncertainty that arises in many systems and have been used successfully in place of traditional complex mathematical systems (Welstead, T. 1994, Kosko, B. 1992, Karr, C. 1994).

A hybrid Intelligent system is said to combine two or more of the intelligent systems group (Medsher, 1995). Some of the more common hybrid systems are:

1. Neural network and genetic algorithm hybrids,
2. Neural networks and fuzzy logic hybrids, and

3. Fuzzy logic and genetic algorithm hybrids.

An introduction to all three types are given in the next few sections, concentrating on the fuzzy logic and genetic algorithm hybrid that is used later in this thesis.

3.3 Hybrid Neural Networks and Genetic Algorithm

This type of hybrid combines neural networks with genetic algorithms. This has become a rapidly expanding area since the late 1980's and early 1990's. The ability of genetic algorithms to search large complex spaces efficiently allow them to find an adequate, if not optimum, solutions more quickly than many other alternatives. They can be combined with neural networks in a number of ways (Medsker, 1995):

1. Representing in the chromosomes the relevant architectural information about a neural networks, such as hidden nodes and layers,
2. Tune the weights and parameters of the neural networks connections,
3. Using a neural networks to evaluate fitness of a chromosome, and
4. Allowing the neural networks to produce new chromosomes for the genetic algorithm.

In the first two cases, a genetic algorithm is used to learn the structure of the neural network. In the second two cases, a neural network is used to direct the search of the genetic algorithm.

The most popular use of the hybrid neural network genetic algorithm is for the genetic algorithm to search the data space to improve the performance of the neural network. This is performed by exploring the input data, tuning the neural networks parameters (number of layers and hidden nodes) and generally explaining the behaviour of the neural network, as in Figure 3.1 (Fukuda, T., Kohno, T. and Shibata, T. 1993). This makes easier the burden of manually training and designing of the neural network, and it speeds up development of such systems.

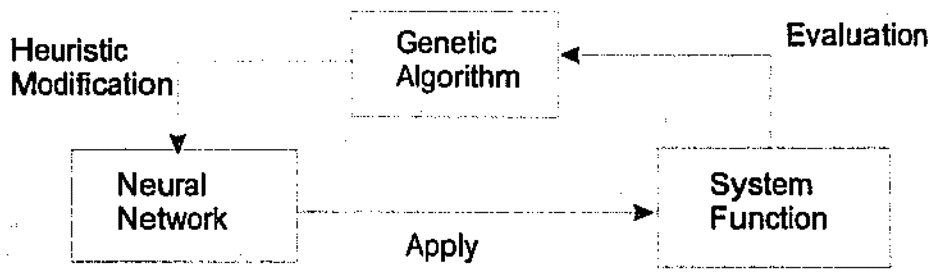

Figure 3.1 Hybrid Neural Network and Genetic Algorithm

Using a neural network to train and optimise the genetic algorithm has not received as much research attention but may offer a number of advantages in producing new genetic chromosomes that converge to an adequate solution quicker. Neural Networks may also be used to define the optimum structure of chromosomes and fitness calculations (Medsker, L. 1995).

Potentially more powerful is the more fully integrated hybrid system where a neural network and genetic algorithm work in tandem to modify both genetic and neural information to dramatically speed training and development (Schaffer, 1994).

3.4 Hybrid Neural Networks and Fuzzy Logic

This type of hybrid system combines neural networks with fuzzy logic. This has become a very important area of research into Intelligent Hybrid Systems since the introduction of the Soft Computing concept by Lotfi Zadeh at his Berkeley Institute in Soft Computing (Zadeh, 1994). Zadeh sees soft computing accommodating imprecision and uncertainty allowing reasoning and computation that is normally required for complex, real world applications. According to Zadeh (Zadah, L. 1994), soft computing involves fuzzy logic, neural networks, probabilistic reasoning, genetic algorithms, belief networks and Chaos theory.

Properties of Intelligent system	Fuzzy System	Neural Network
Function estimators	✓	✓
Trainable, dynamic	✓	✓
Improvements with use	✓	✓
Parallel implementations	✓	✓
Numerical	✓	✓
Tolerance for imprecision	✓	✗
Explicit knowledge representation	✓	✗
Adaptive	✗	✓
Optimising	✗	✓
Interpolative	✗	✓
Tolerance for noise	✗	✓

Table 3.1 Comparison of Fuzzy Systems and Neural Networks

Table 3.1 shows the characteristics of fuzzy systems and neural network systems (Medsker, 1995). As the table shows, there are a number of common characteristics , such as each are model-free function estimators that can be trained to improve their performance. They are both implicitly parallel in nature which allows parallel processing techniques to be used. Where they have no common characteristics is where the neural network and fuzzy logic system hybrid gains its strengths. The hybrid system uses the fuzzy logic imprecision and knowledge representation while also using the neural networks adaptation and optimising features (as well as the neural networks tolerance for noise). By combining these strengths, a larger number of applications can be developed.

According to Medsker (Medsker, 1995), applications that have used hybrid fuzzy neural systems have tended to be in the engineering and physical science areas, as well as the biological and medical areas. There are a number of different ways in which fuzzy logic systems can be integrated with neural network systems. One method is the Connectionist Expert System. These are similar to the combining neural networks and expert systems (Gallant, S. 1993). They replace the expert system components with fuzzy sets (Chapter 2 explained the relationship between expert systems and fuzzy logic).

A neural network can use fuzzy logic to pre-process the data the system receives. This allows the fuzzy system to process the incoming fuzzy data and output some value that the neural network can use. Additionally, after the neural network has processed the data, it can be passed to a fuzzy system to be processed. By fuzzifying the incoming data, the neural network learns to map the fuzzy inputs to either crisp or fuzzy outputs (Figure 3.2).

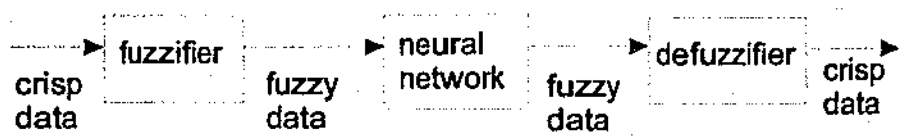

Figure 3.2 Fuzzy Neural system

To develop the fuzzy neural network application, the following steps must be taken (Cox, E. 1994):

1. Fuzzy rules and membership sets should be formed from application knowledge,
2. Connect the fuzzy outputs to the neural network inputs,
3. Initialise neural network connections and weights,
4. Train the neural network on the training data input/output pairs for the application,
5. Use the fuzzy neural network system for operational data.

Neural networks can also be used to train the fuzzy logic system. This may be done by finding the "best" membership sets and/or the fuzzy rules to govern the system (Hung, 1993). Figure 3.3 below shows the steps involved when using a neural network to find the rules for the fuzzy system. A similar model is used when a neural network is used to find the membership sets boundaries of the fuzzy system.

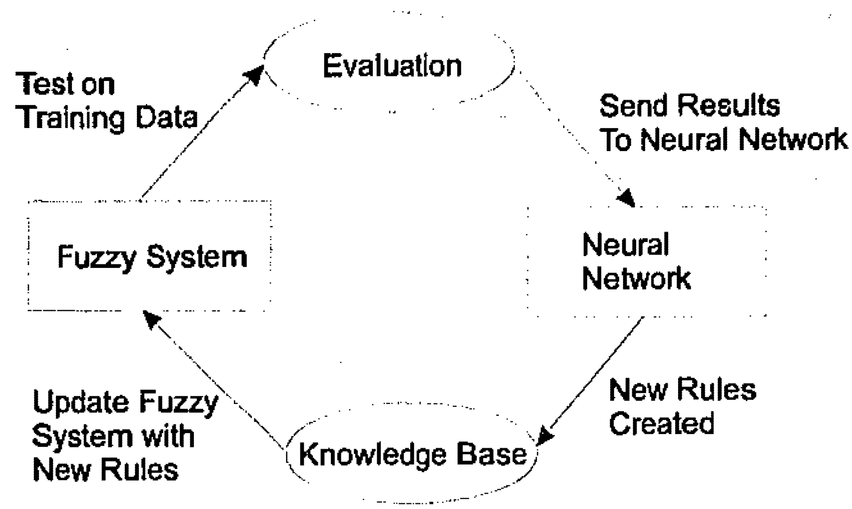

Figure 3.3 Neuro Fuzzy Controller

With the continuing development of the Soft Computing environment, the use of Hybrid Neuro-Fuzzy systems will progress into more advanced fields, perhaps providing systems that can deal with the complexity of many processes that are difficult to develop currently.

3.5 Hybrid Fuzzy Logic and Genetic Algorithm

This type of hybrid system combines fuzzy logic and genetic algorithms (FLGA). This is the newest of the Hybrid systems currently being developed with the first works only published a few years ago. One of the pioneering works on fuzzy logic genetic algorithm hybrids was by Charles Karr in 1991 (Karr, 1991). Since then, much research has been conducted in this field, showing many positive approaches. Currently, the most promising fields using this kind of hybrid system is the use of genetic algorithms to improve the performance of fuzzy systems.

Characteristics	Fuzzy System	Genetic Algorithm
Stores Knowledge	✓	✗
Learns	✗	✓
Optimises	✗	✓
Fast	✓	✓
Nonlinear Systems	✓	✓

Table 3.2 Comparison of Fuzzy Systems and Genetic Algorithms

Genetic Algorithms and fuzzy logic have a number of characteristics in common, as shown in Table 3.2 (Medsker, 1995). Both are fairly fast and can be used when dealing with nonlinear systems. Fuzzy logic is well suited to storing experts knowledge in the form of fuzzy rules. For well defined systems, constructing the fuzzy rules are simple, but for more complex systems where there are a large number of rules, developing the fuzzy knowledge base is difficult and very time-consuming task using trial and error techniques. The tuning and optimisation of membership functions of fuzzy logic systems is also a difficult task (Mohammadian, M and Stonier, R. 1994).

It is in these situations that genetic algorithms can be used to great effect. The ability to optimise and learn allow the genetic algorithm to learn and optimise the fuzzy membership functions and the fuzzy rules of the fuzzy logic system.

While most research has been in using a genetic algorithm to tune and optimise fuzzy systems, there has been some research in using fuzzy logic to control parameter selection for genetic algorithms. Lee and Takagi (1993), proposed a system where a fuzzy knowledge base system is used to control genetic algorithm's parameters. The inputs into the fuzzy knowledge base are the current genetic algorithm performance measures or settings such as population size, mutation rate, etc. The fuzzy system then outputs new control parameters for the genetic algorithm such as changing the population size (Figure 3.4).

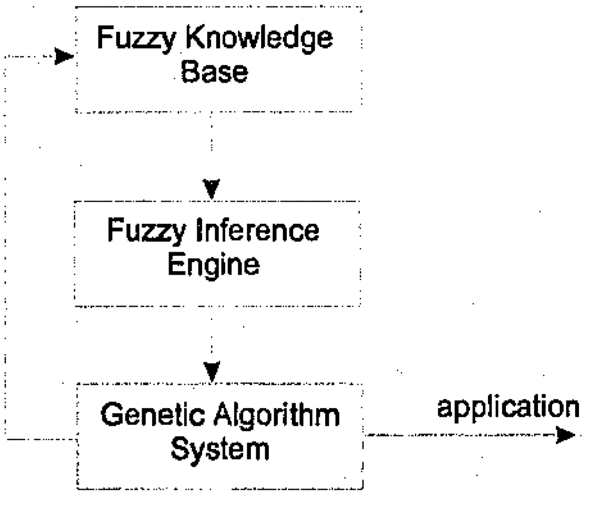

Figure 3.4 Fuzzy Logic System for Controlling a Genetic Algorithm

3.5.1 Tuning Fuzzy Membership Functions

There are a number of properties that make the genetic algorithm fundamentally different from conventional search techniques, making them more attractive for certain solutions. As mentioned in chapter 2, some of these properties are:

1. Genetic Algorithm's consider a whole population of points, not just the single point,
2. They work directly with strings that represent the parameter set, not the parameters themselves,
3. They use probabilistic rules to guide their search, not deterministic rules.

One method to design a robust fuzzy logic systems is to establish the rule base with the appropriate rules that have been obtained (say, from an expert, or trial and error) and use a genetic algorithm to determine the "optimal" membership functions. When using triangular functions, the parameters are the centres and widths for each category (trapezoidal shaped membership functions are a simple extension of this parameter set) (Mohammadian, M and Stonier, R. 1994, Ng, K. C. and Li, Y. 1994) . The genetic algorithm generates an initial population of possible solutions for the membership functions (see Figure 3.5).

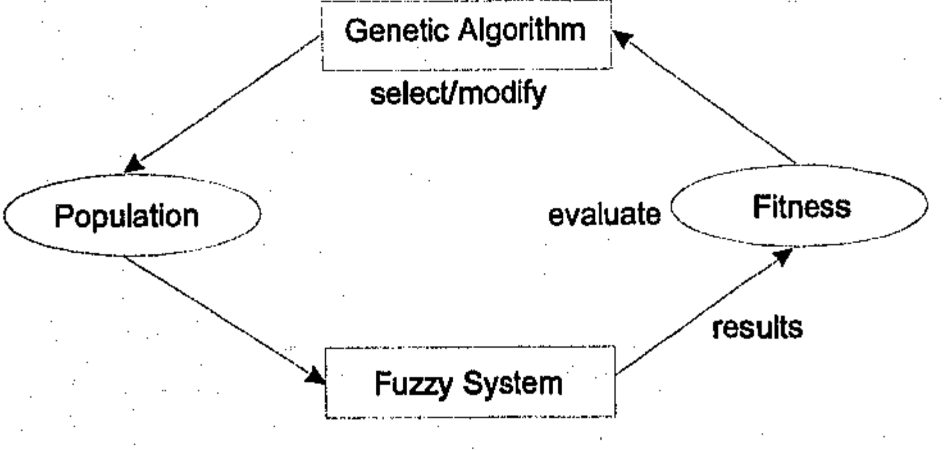

Figure 3.5 Using a GA to improve performance of a fuzzy system

Each chromosome of the genetic algorithm is evaluated and assigned a fitness value. New generations are generated by the genetic algorithm using the genetic operators. This cycle continues until an acceptable solution has been found (Kingham, M and Mohammadian, M. 1996). Figure 3.5 shows this cycle (Medsker, 1995).

The figure below shows an example of how this would work on a fuzzy systems that contains three membership functions (**Small**, **Medium** and **Big**). The genetic algorithm creates an initial population and generates a number of possible candidates solutions. The initial membership function is shown in Figure 3.6 (a). The genetic algorithm continues to generate new populations until an "optimal" (or adequate) solution is found. The fuzzy logic system now has the optimal membership functions and can be used for its designed application (Figure 3.6 (b)).

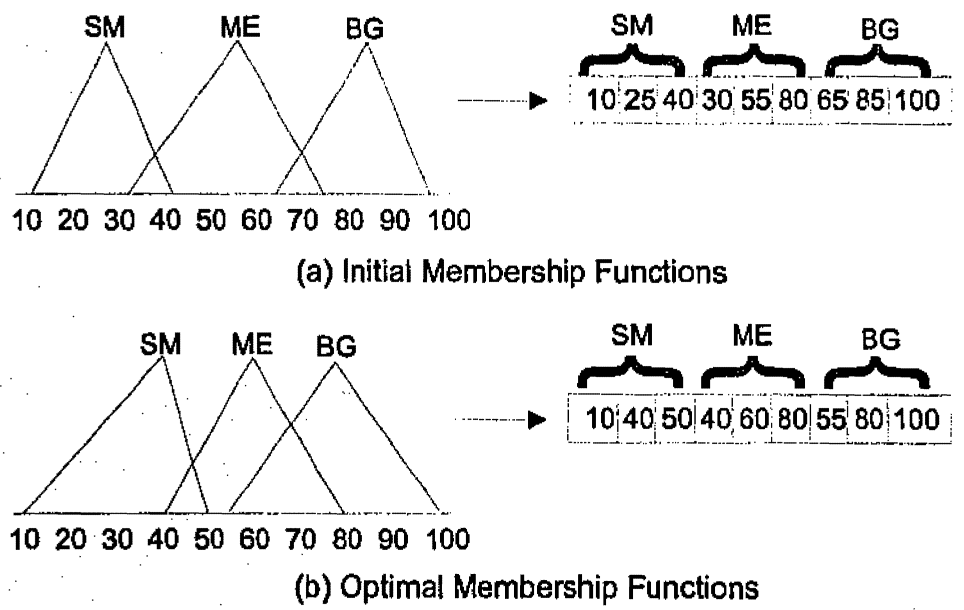

Figure 3.6 Fuzzy membership functions optimised by GA

3.5.2 *Finding Fuzzy Rules by Genetic Algorithms*

Finding the fuzzy rules of complex systems with large number of input parameters is a difficult task. A genetic algorithm can be used to learn the fuzzy rules of a fuzzy system in a similar method as performed to find the "optimal" membership functions.

As mentioned in Chapter 2, genetic algorithms are powerful search algorithms based on the mechanism of natural selection and use operations of reproduction, crossover, and mutation on a population of strings. A set (population) of possible solutions, in this case, a coding of the fuzzy rules of a FL system, represented as a string of numbers. Figure 3.7 shows the combination of FL and GAs for generating fuzzy rules of a FL system.

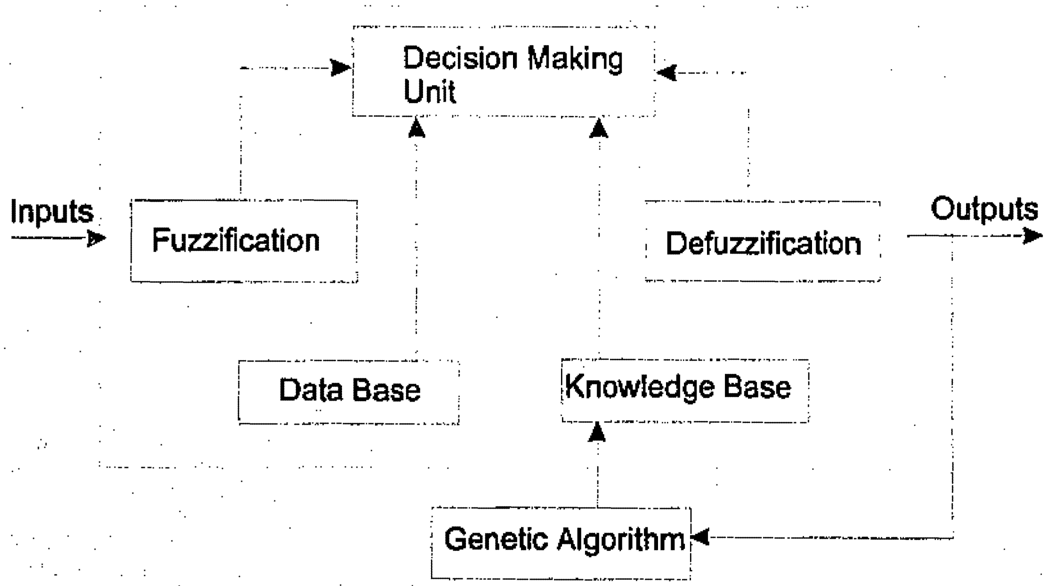

Figure 3.7 Combination of FL and GAs for fuzzy rule generation

3.5.3 Encoding and Decoding of Fuzzy rules by GAs

First the input parameters of the fuzzy logic system is divided into fuzzy sets. Assume that the FL system has two inputs α and β and a single output δ. Assume also that the inputs and output of the system is divided into 5 fuzzy sets. Therefore a maximum of twenty five fuzzy rules can be written for the FL system. The consequent for each fuzzy rule is determined by genetic evolution. In order to do so, the output fuzzy sets are encoded. It is not necessary to encode the input fuzzy sets because the input fuzzy sets are static and do not change. The fuzzy rules relating the input variables (α and β) to the output variable (δ) have twenty five possible combinations. The consequent of each fuzzy rule can be any one of the five output fuzzy sets.

Assume that the parameter (δ) has five fuzzy sets with the following fuzzy linguistic variable: **NB** (Negative Big), **NS** (Negative Small), **ZE** (Zero), **PS** (Positive Small), and **PB** (Positive Big). The output fuzzy sets are encoded by assigning 1 = **NB** (Negative Big), 2 = **NS** (Negative Small), 3 = **ZE** (Zero), 4 = **PS** (Positive Small), and 5 = **PB** (Positive Big). GA randomly encodes each output fuzzy set into a number ranging from 1 to 5 for all possible combinations of the input fuzzy variables. A string encoded this way can be represented as (Figure 3.8):

4	3	5	3	----------	--
1	2	3	4		25

Figure 3.8 Fuzzy Rules encoded in GA string

Each individual string is then decoded into the output linguistic terms. The set of fuzzy rules thus developed is evaluated by the fuzzy logic system based upon a fitness value, which is specific to the system. At the end of each generation, two copies of the best performing string from the parent generation is included in the next generation to ensure that the best performing strings are not lost. GA then performs the process of selection, crossover and mutation on the rest of the individual strings. Selection and crossover are the same as a simple GAs while the mutation operation is modified.

Crossover and mutation take place based on the probability of crossover and mutation respectively. The mutation operator is changed to suit this problem. For mutation, an allele is selected at random and it is replaced by a random number ranging from 1 to 5. The process of selection, crossover and mutation are repeated for a number of generations till a satisfactory fuzzy rule base is obtained. We define a satisfactory rule base as one whose fitness value differs from the desired output of the system by a very small value.

In the following chapter, a hybrid system that combines fuzzy logic and genetic algorithms together will be developed (FLGA). The FLGA system will be used to predict the quarterly interest rates in Australia.

Chapter 4 Interest Rate Prediction using Integrated FL and GAs

4.1 Introduction

Investors and governments alike are interested in the ability to predict future interest rate fluctuations from current economic data. Investors are trying to maximise their gains on the capital markets, while government departments need to know the current position of the economy and where it is likely to be in the near future for the well being of a countries people.

Economists, and investors, have been unable to find all the factors that influence interest rate fluctuations. However as mentioned in Chapter 3, there are some major economic indicators released by the government (Madden, R. 1995) that are commonly used to determine the current position of the economy.

4.2 Economic Indicators

4.2.1 Interest Rate

In this thesis, we look at the prediction of Ten Year Treasury Bonds which is used by the government to calculate long term interest rates. Interest is the compensation paid to a lender for deferring the price paid by a borrower for the use of the funds.

There are different rates of interest which vary according to the amount borrowed, the length of time of the borrowing and the financial stability of the borrower. Treasury Bonds are long term securities issued over a long period of time (in this case ten years).

4.2.2 Unemployment Rate

Unemployment exists when people without a job are looking for but unable to find employment. The labour force is made up of the civilian population aged 15 to 65 who are already working and those that are actively looking for work and are unable to find employment. The ABS classes people as unemployed as those who are actively looking for work and can start work immediately. Looking for work includes writing, telephoning, faxing or meeting an employer or registering with the Department of Social Security and Commonwealth Employment Service. The Unemployment rate is the percentage of the labour forces that are not employed but are actively looking for employment.

4.2.3 Job Vacancies

A job vacancy is a job available for immediate filling for which some form of recruitment action has been taken (such as advertising in some form of media available to the public, informing job centres). When the demand for labour is high, the number of job vacancies increases while when the demand is low the number of job vacancies falls. Recessions generally have a low job vacancy rate while economic growth periods generally have high job vacancy rates. The job vacancy is recorded as the number of vacancies per 1000 unemployed.

4.2.4 Gross Domestic Product

The Gross Domestic Product (GDP) is an aggregate measurement of the flow of goods and services produced in an economy. Only goods used for final consumption or capital goods are included to remove the possibility of "double counting". This can occur if the a good is required to make another good further down the chain. GDP describes the domestic product because it does not include income earned outside the country. There are a number of different measures of GDP calculated by the ABS. These are GDP(P) which is the sum of goods and services produced at each stage of the production less the cost of production, GDP(I) which is the sum of incomes generated by production, and GDP(E) which is the sum of final expenditure on goods and services produced plus exports minus imports. This thesis uses the average of these three indicators and is referred to as GDP(A). According to Madden (Madden, 1995), analysis has shown that *constant price* GDP(A) has provided the most satisfactory indicator of short term trend growth in GDP.

4.2.5 Consumer Price Index

The Consumer Price Index (CPI) is an indication of the rate of change in prices paid by consumers for a fixed list of goods and services. This list of items has been selected to represent purchases by the average household and includes items in the following areas: alcohol, clothing, education, food, health care, household equipment and operation, housing, recreation, tobacco and transportation. Every 5 years the list of items is revised to reflect the changing economy. The price of the CPI in the base period 1989-90 is used as the reference point and is set to 100.0 and the prices in later periods are identified as a percentage of the base period.

$$\text{CPI} = \frac{\text{total cost of goods in given period}}{\text{total cost of goods in reference period}} \times 100 \tag{4.1}$$

4.2.6 Household Saving Ratio

Savings is often defined as the income not spent on goods and services which are used for current consumption (Pearce, 1983). It can be expressed as giving up current consumption to derive a future benefit. Savings are used to finance investments which will increase the productive capability to produce a greater quantity of goods and services in the future. The household's disposable income is the amount of income that households have available for spending after deducting taxes paid, interest payments and transfers to overseas. The Household Savings Ratio is the ratio of household income saved to household's disposable income.

4.2.7 Home Loans

Most home purchases are financed by a home loan from a financial institution such as banks. The demand for housing loans is dependent on consumers perceived ability to repay the home loan debt. This ability can be influenced by the interest rate, home prices, the consumer's income level and financial/employment stability.

In general, when interest rates are high, the total home loan's decline while when interest rates fall, the demand for home loans financing increases.

4.2.8 Average Weekly Earnings

Earnings are used to describe the total payments an individual receives from his employment. The weekly earnings are the workers gross earnings, which includes their basic pay together with any payments for shifts and overtime and any form of incentive scheme. The average weekly earnings are the average of the wages received by both men and women for an entire week. When the average weekly earnings increases by a greater amount than the CPI, then there has been real wage increases.

4.2.9 Current Account

The current account is a running account for a person, business or country. The indicator here relates to the countries Balance of Payments Current Account, which is the sum of the balances on goods and services, income and unrequited transfers. If the sum of the balances is positive, then the country has a current account surplus, while a nation that has a negative sum of the balances has a current account deficit. Goods and services balance is the difference between the total export value and the total import value of goods and services. Net income balance is the difference between the value of income and interest earned by residents from non-residents and that payable by residents to non-residents. An unrequited transfers is required when real or financial resources are provided without something of economic value being received (for example foreign aid).

4.2.10 Trade Weighted Index

The Australian trade weighted index is an index of the average value of the Australian dollar ($A) compared to the currencies of Australia's major trading partners. The weight given to each currency is relative to the level of trade between Australia and the trading partner. The Reserve Bank trade weighted index includes 23 countries which account for more than 90 percent of Australia's foreign trade.

4.2.11 RBA Commodity Price Index

The Reserve Bank of Australia (RBA) Commodity Price index consists of 19 commodities which represents about half of total merchandise exports and more than two-thirds of Australia's commoditv exports. The index was developed to provide an early indication in the country's export prices. The weights for each commodity vary over time and relate to the share of the export market by volume.

4.2.12 All Industrial Index

The All Industrial Index provides an indication of aggregate price movements on the Australian Stock Exchange. The index is calculated from a sample of shares which includes about 260 companies. The index only measure the capital gain or loss experienced by share holders through fluctuations of the share market. The index does not take into account any dividend earned.

4.2.13 Company Profits

Company profits are taken from a wide range of industries in Australia, including, manufacturing, mining, construction, wholesale and retail trade to name a few. Companies excluded include those primarily engaged in agriculture, forestry, banking and insurance activities. The data relates to companies employing more than 30 people. Profits for a company is defined as net operating profits or losses before income tax. During periods of economic growth, the level of company profits generally increases, while declining economies usually see smaller company profits.

4.2.14 New Motor Vehicle Registrations

New Motor Vehicle Registrations is a measure of the number of new cars purchased which have been registered with the appropriate registration authority. It gives an indication of the number of new cars sold. One of the major purchases for any household is a new motor vehicle. During periods of economic growth, the level of new registrations generally increases, while declining economies usually see a smaller number of registrations.

4.3 Fuzzy Logic and Genetic Algorithms for the Prediction of Interest Rates

To predict fluctuations in the interest rate, a fuzzy logic system was created. The fuzzy rules (knowledge base) were discovered by the use of a genetic algorithm. There are a number of steps to perform to create the fuzzy knowledge base of the fuzzy logic system:

1. Identify the inputs and outputs,
2. Pre-process data if required, and split into training and test suites,
3. Create Fuzzy sets and Fuzzy membership functions for input and output parameters of the system,
4. Set parameters for genetic algorithm training (mutation and crossover),
5. Create an "optimum" FKB using the genetic algorithm with training data,
6. Use the developed FKB for fuzzy logic on test data.

4.4 Identify the inputs and outputs

To design a fuzzy logic system, the actual inputs and outputs must first be determined. For this system, we are using a number of economic indicators to predict the following quarters interest rate. There are a number of possible indicators that could be used to predict the interest rate. Three of these indicators are:

- *Interest Rate* which is the indicator being predicted. The Interest Rate used here is the Australian Commonwealth government 10-year treasury bonds.

- *Gross Domestic Product* (GDP) is an average aggregate measure of the value of economic production in a given period.
- The *Consumer Price Index* (CPI) is a general indicator of the rate of change in prices paid by consumers for goods and services.

A fuzzy logic system can be developed that uses the above indicators as its inputs (Figure 4.1). Appendix A shows the Australian economic data for these indicators from Quarter1, 1983 through to Quarter 4, 1997 (McLenan, W, 1997). This gives us sixty quarters of data in which to train and test the system.

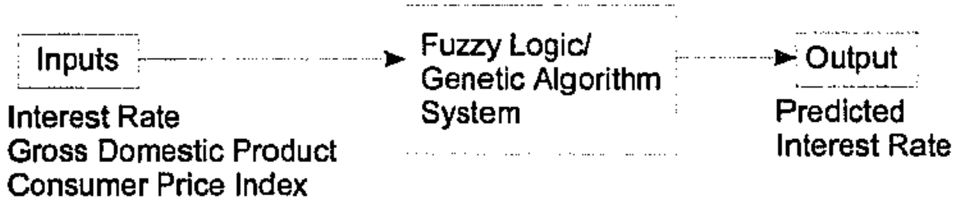

Figure 4.1 Fuzzy Logic/Genetic Algorithm system to predict Interest Rate

The current interest rate is included in the input indicators to the system as the predicted interest rate is highly dependent on the current rate as there is only likely to be a small fluctuations in the interest rate. The current interest rate gives the FLGA system an indication as to the expected "ball park" area of the predicted rate (in fact, this gives an indication as to what part of the fuzzy knowledge base to look at).

4.5 Pre-Process Data

In most time series predictions, there is some pre-processing of the data so that it is in a format that the system can use. This may be where data is normalised so it fits within certain boundaries, formatted into an appropriate form for the FLGA system to use. It is also where decisions on how the data is represented is made.

There are a number of ways in which the raw data from the above indicators could be represented. Firstly, the system could just use the data "as is" and make its predictions from that. Alternatively, the system could instead use the difference from the previous quarter to the current quarter. The system could also take into consideration the effects of inflation on the raw data and compensate appropriately.

In this system, the change from one quarter to the next is used for the GDP and CPI indicators, while the interest rate is the actual reported rate from the Australian Bureau of Statistics. For example, the data for interest rates is stored in the following format:

Year	**Quarter**	**Data**
1983	1	14.00
1983	2	12.95
1983	3	15.00
1984	4	14.85
1985	1	14.15

Table 4.1 Format of Economic Data

Once the data has been pre-processed, it must be split into some groups for the training and testing of the system. For this system, the first 2/3rds of the data was assigned to the training set while the other 1/3rd was assigned to the test set. The system uses the training set to learn the FKB for the fuzzy logic system using the genetic algorithm. The fuzzy logic system is then tested on the test set using the "best" FKB.

4.6 Fuzzy Membership Functions

Now that the data has been pre-processed and assigned into a training and test set, the fuzzy logic system can be built. The data must be split into a number of fuzzy sets. For each fuzzy set a fuzzy membership function is assigned. The membership functions used in this study are triangular membership functions.

Each input and output parameter is divided into five fuzzy sets as this provided the necessary precision of the indicators for the system without increasing the number of fuzzy rules required by the FKB to a computationally expensive amount. The linguistic values assigned for the input and output fuzzy sets are:

PB - Positive Big,

PS - Positive Small,

ZE - Zero,

NS - Negative Small,

NB - Negative Big.

The first step in finding the fuzzy sets and membership functions is to find the minimum and maximum range of the data for the indicator. The minimum and the maximum point become the mid point (centre) for the first and last membership function respectively. The range is then divided into 4 equal parts, which become the mid point for the membership functions in between the first and last sets. It was decided that the membership functions have an equal overlap of 1/3rd size of the membership function as this was found to give good results over the entire range of data. Therefore, if we had two fuzzy sets that ranged from 3 to 7 inclusive, their membership functions would be defined as shown in Figure 4.2.

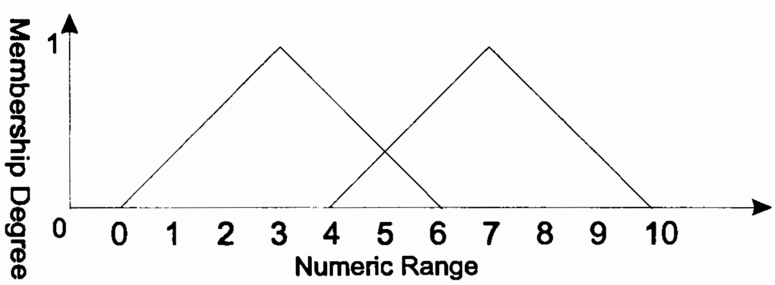

Figure 4.2 Fuzzy Membership Functions

For the interest rate indicator, the data ranged from 6.7 to 15. Using the above method, the fuzzy membership functions for interest rates would be found as per Figure 4.3.

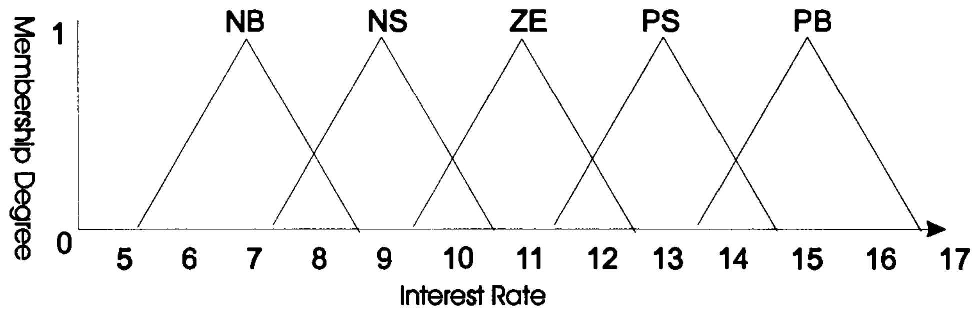

Figure 4.3 Membership functions for Interest Rate

The boundaries for the fuzzy sets for Interest Rate are shown in Table 4.2 below.

Fuzzy Set	Triangle Start	Middle	End
NB	5.14	6.70	8.26
NS	7.22	8.77	10.33
ZE	9.29	10.85	12.41
PS	11.37	12.92	14.48
PB	13.44	15.00	16.56

Table 4.2 Interest Rate membership function boundaries

By applying the same technique to the Gross Domestic Product and Consumer Price Index indicators, we find that CPI has a minimum of -0.30 and a maximum of 2.70 while GDP has a minimum of -1023.00 and a maximum of 2464.00. The fuzzy membership functions for GDP and CPI are shown in Figure 4.4 and 4.5 respectively.

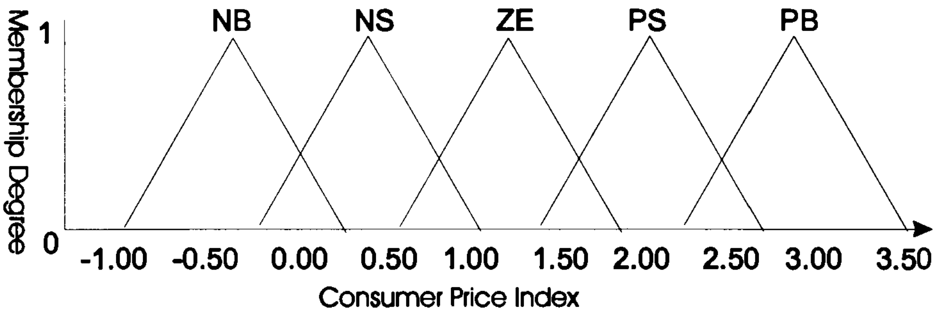

Figure 4.4 Consumer Price Index membership functions

Fuzzy Set	Triangle Start	Middle	End
NB	-0.86	-0.30	0.26
NS	-0.11	0.45	1.01
ZE	0.64	1.20	1.76
PS	1.39	1.95	2.51
PB	2.14	2.70	3.26

Table 4.3 Consumer Price Index membership function boundaries

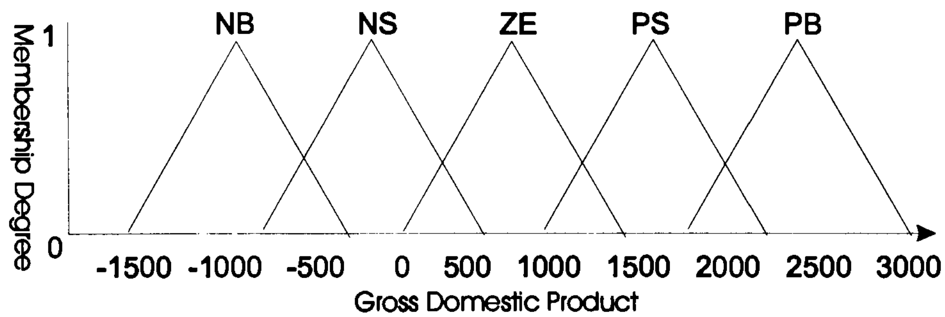

Figure 4.5 Gross Domestic Product membership functions

Fuzzy Set	Triangle Start	Middle	End
NB	-1676.81	-1023.00	-369.19
NS	-805.06	-151.25	502.56
ZE	66.69	720.50	1374.31
PS	938.44	1592.25	2246.06
PB	1810.19	2464.00	3117.81

Table 4.4 Gross Domestic Product membership function boundaries

The output for the fuzzy logic system is the same indicator as per the input Interest rate, we use the same fuzzy sets and membership functions of Interest Rate for the input Interest Rate parameter.

4.7 Initialise Fuzzy Logic Knowledge Base

In order for the fuzzy system to predict the following quarters interest rate, it must have a knowledge base in which to extract the rules. As discussed in chapter 2, a fuzzy logic rule is an IF…THEN statement with linguistic terms.

For example, a typical fuzzy rule may be

IF interest rate is **PB** and CPI is **NS** and GDP is **NB** THEN

predicted interest rate is **ZE**

The size of the FKB depends on the number of fuzzy set inputs parameters of the system. In this fuzzy logic system, we have three inputs each with five membership functions. Therefore, the size of the FKB is

$$5 \times 5 \times 5 = 125 \text{ rule.}$$

If we had added another input parameter for the fuzzy logic system with five fuzzy sets then we would have

$$5 \times 5 \times 5 \times 5 = 625 \text{ rules.}$$

This can quickly add up to a very inefficient FKB (Raju, G.V.S. and Zhou, J., 1993). In the next chapter, we discuss how this problem can be overcome with the use of a Hierarchical Fuzzy Logic System.

As the input fuzzy sets are static and do not change, only the consequence for each fuzzy rule needs to be determined. To initialise the FKB, each possible output membership function is allocated a number. There are five fuzzy sets for output parameter, we assigned 1…5 to the fuzzy sets (**NB** = 1, **NS** = 2, **ZE** = 3, **PS** = 4, **PB** = 5). The FKB is then randomly encoded using genetic algorithm with numbers ranging from 1 to 5.

A FKB for the system may look as follows:

Rule 1: IF interest Rate is **NB** and CPI is **NB** and GDP is **NB** THEN consequent

Rule 2: IF interest Rate is **NB** and CPI is **NB** and GDP is **NS** THEN consequent

Rule 3: IF interest Rate is **NB** and CPI is **NB** and GDP is **ZE** THEN consequent

Rule 4: IF interest Rate is **NB** and CPI is **NB** and GDP is **PS** THEN consequent

Rule 5: IF interest Rate is **NB** and CPI is **NB** and GDP is **PB** THEN consequent

Rule 6: IF interest Rate is **NB** and CPI is **NS** and GDP is **NB** THEN consequent

...

Rule 124: IF interest Rate is **PB** and CPI is **PB** and GDP is **PS** THEN consequent

Rule 125: IF interest Rate is **PB** and CPI is **PB** and GDP is **PB** THEN consequent

where the consequent is the action performed when the rule fires.

4.8 Genetic Algorithm Parameters

As discussed in the hybrid fuzzy logic and genetic algorithm section form Chapter 3, a genetic algorithm is able to learn the rules that govern the fuzzy system. In order for the genetic algorithm to learn the mapping for the FKB, it must encode its strings to represent the rules in the FKB. Instead of using a binary string as suggested by Goldberg (Goldberg, 1989), the string for the genetic algorithm is encoded using the integer numbers that represent the fuzzy sets. The figure below shows an example of a string used by the genetic algorithm.

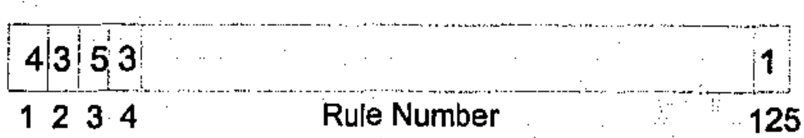

Figure 4.6 A string created by GA to encode a FKB

There are a number of reasons why the use of an integer string is used instead of the traditional binary string for the chromosome. By using an integer number, we can easily extract the fuzzy sets from the string as each integer represents a fuzzy rule in the FKB. Instead of performing bit manipulations on the chromosome, which can be a time consuming task, an integer representation allows faster encoding and decoding of the chromosomes used by the genetic algorithm (Mohammadian, M. and Stonier, R. 1996).

The trade off is the amount of memory required to store the fuzzy sets as integers instead of bits strings. For a FKB consisting of 125 rules, 125 bytes would be required to store the FKB. Using bit strings, the five fuzzy sets can be represented with 3 x 125 bits (47 bytes).

In order for the genetic algorithm to learn the FKB, a number of parameters must be set for the genetic algorithm. Finding the correct combination for these parameters is often a matter of trial and error. For this system, the genetic algorithm had the following parameters:

population size = 150,

crossover rate = 0.6 ,

mutation rate = 0.01,

Chromosome length = 125.

These figures were decided on after a number of trials and were found to give the best results overall.

The genetic algorithm was run for 5000 generations over a period of 40 quarters (10 years) using the training data. The genetic algorithm used an elitist strategy so that the best population generated was not lost. The best performing popula! .on is saved and entered into the next generation of the genetic algorithm. This procedure prevents a good string from being lost by the probabilistic nature of reproduction and also speeds convergence to a good solution.

Fitness of each chromosome was calculated as the sum of the absolute differences from the predicted quarter and the actual quarters interest rate. The aim is to minimise the difference between the predicted interest rate and actual interest rate.

The fitness was subtracted from an "optimal" fitness amount, which was decided to be 40 as it is unlikely that the error amount would be higher than 40 over 40 quarters of data used for training of the system. The fitness of the system is calculated by the following formula:

$$fitness = 40 - \sum_{i=0}^{40} abs(PI_i - I_{i+1}) \tag{4.2}$$

where PI_i is the predicted interest rate,

i is the current quarter and

I_{i+1} is the actual interest rate for the next quarter.

4.9 Learning the FKB with a Genetic Algorithm

Once the genetic algorithm parameters have been set, the genetic algorithm is run to find the FKB for the system. The genetic algorithm randomly creates an initial population of strings that represent possible FKB's solutions. Using the processes of selection, crossover and mutation, the genetic algorithm is able to find a satisfactory FKB. Simulation results are shown below:

After the initial generation, the following statistics were produced:

maximum fitness	3.04
average fitness	1.19
minimum fitness	0.00
sum of fitness	178.50

Table 4.5 Generation 0 statistics

As Table 4.5 shows, the initial generation has very low fitness values. Figure 4.7 below shows the results of the best FKB generated after generation 0 on the training data. The dotted line is the predicted interest rate while the solid line is the actual interest rate for that quarter. The Error Amount between the actual interest rate and the predicted interest rate is shown in the bottom part of the figure. As can be seen, the error amount between the actual and predicted interest rate for each quarter is very large for a number of quarters. This is what we are trying to reduce. The FKB generated can be seen to provide only a few rules that give good results. As this generation has not provided an optimal FKB, and the maximum number of generations has not been reached, a new generation is created.

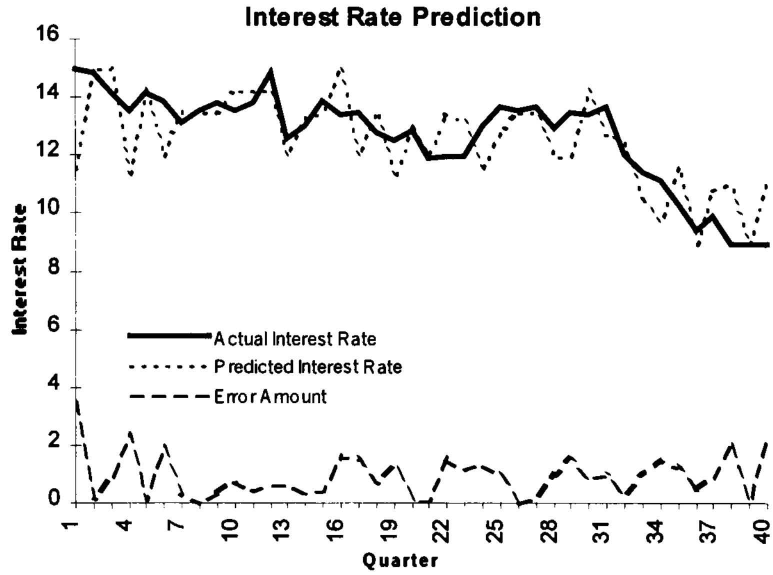

Figure 4.7 Prediction of Interest Rate using Initial FKB

To obtain the next generation of chromosomes, the genetic algorithm performs the following steps (as discussed in Chapter 2):

1. Choose two chromosomes from the generation probabilistically according to fitness using the Roulette wheel selection method described in Chapter 2,
2. Randomly choose a position (cut point) within the chromosomes and perform crossover operation and add to next generation of chromosomes,
3. Evaluate the fitness of the two new chromosomes,
4. Repeat steps 1 - 3 until the population size maximum has been reached,
5. Repeat steps 1-4 until the number of generations to perform has been reached or an optimal (or adequate) chromosome has been found.

After fifty generations of the population have evolved, the following fitness' were obtained (see Table 4.6).

maximum fitness	16.28
average fitness	9.43
minimum fitness	0.00
sum of fitness	1415.04

Table 4.6 Generation 50 statistics

As these statistics show, the maximum fitness achieved has increased significantly. This means that the genetic algorithm is now generating some chromosomes that provide good rules for the FKB.

By graphing the results of the training data, we can see that although some quarters have a large discrepancy between the actual interest rate and predicted rate, there are a number of quarters where the difference is very small.

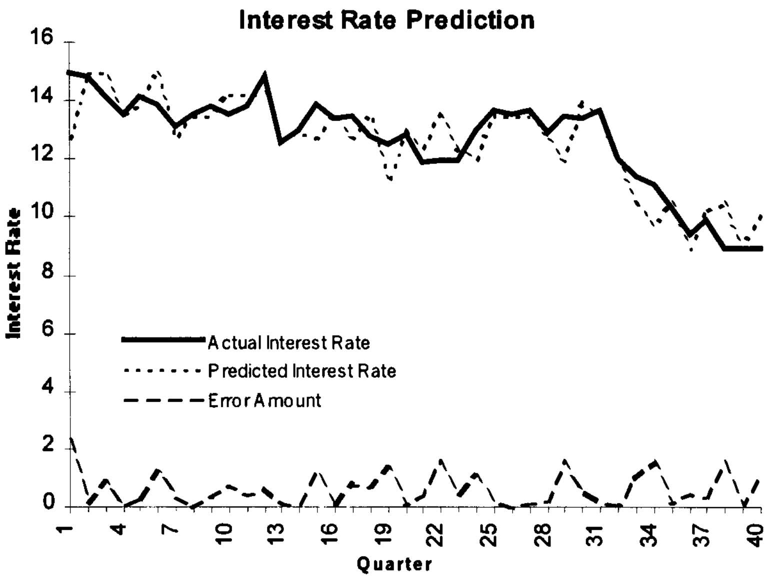

Figure 4.8 Results after 50 Generations

As Figure 4.8 shows, the Error Amount has reduced over the entire learning period. Comparing these results to those obtained with the initial FKB (Figure 4.7), we can see that whilst the initial FKB had error amounts greater than 3 percent, the largest error amount after 50 generations is a little over 2 percent. This shows that the FLGA system is gradually finding a FKB that can model the fluctuations in the interest rate from one quarter to the next.

The genetic algorithm continuous to generate new populations until it has reached the maximum number specified earlier (5000 generations). At the end of the learning cycle, the best population had a fitness of 26.1250. Table 4.7 below shows the final statistics of the genetic algorithm search for the best FKB for the system.

maximum fitness	26.1250
average fitness	19.200
minimum fitness	8.497
sum of fitness	3141.108

Table 4.7 Generation 5000 statistics

Figure 4.9 shows how the final FKB predicted the quarterly interest rate for the training data. Compared to the previous FKB in generation 50, there has been a detectable rise in performance. The error amount for the training data has been reduced, with only a few quarters having an error amount greater than 1 percent.

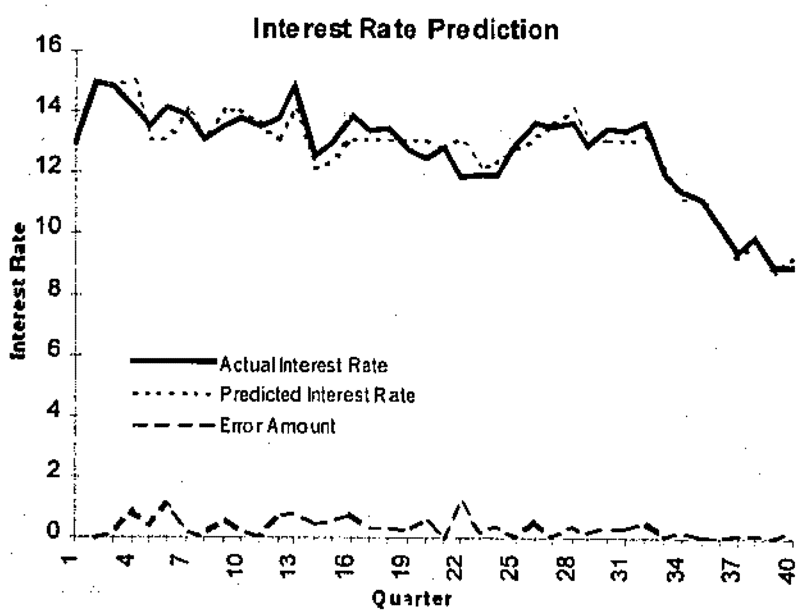

Figure 4.9 Results of best FKB after training completed

Figure 4.10 below shows how the systems maximum fitness progressed from the initial generation through to the last generation. As Figure 4.10 shows, there is a steep rise at the start of the learning progress which then evens out after 1000 generations. Although better chromosomes (and hence FKB's) are found after this generation, there is not the steep learning curve we found in the earlier generations, in fact after generation 1725 there has been little improvement in chromosome's fitness found that improve the performance of the FKB.

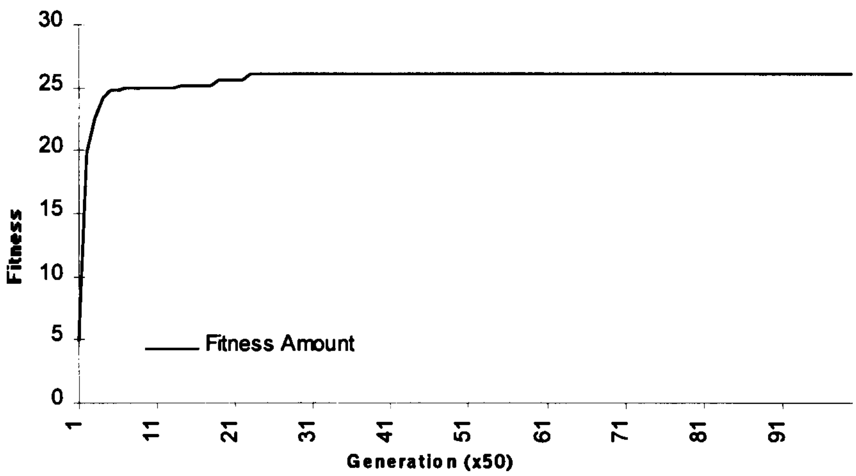

Figure 4.10 Maximum Fitness of best FKB after each generation

One of the problems identified after the genetic algorithm has found its best chromosome (after the maximum number of generations) is that there are a number of rules that have not been tested. Ruelle (Ruelle, 1989) makes the observation that there are many pitfalls that can occur when using time series predictions, in particular the time series should be of sufficient length. If the amount of data is large enough, then all possible rules would be covered and the genetic algorithm could learn all the correct rules for the FKB. Unfortunately, as pointed out, some rules in the FKB cannot be tested on the training data set. As it is unlikely that the all the rules are correct, some method must be developed that will allow the fuzzy system to use an unlearnt rule and still be reliable on the output the Fuzzy Logic system generates.

One method to achieve this is to set unused rules to some default value. In this thesis, unused rules have been assigned to the fuzzy set in which their interest rate falls. As the interest rate has been split into five fuzzy sets, the FKB rule can have five possible values. For the fuzzy logic system, there are 125 FKB rules. Splitting this into five equal regions gives us 25 rules. As discussed in earlier in this chapter, the FKB has been structured in the following method:

Rule 1: IF interest Rate is **NB** and CPI is **NB** and GDP is **NB** THEN consequent

Rule 2: IF interest Rate is **NB** and CPI is **NB** and GDP is **NS** THEN consequent

Rule 3: IF interest Rate is **NB** and CPI is **NB** and GDP is **ZE** THEN consequent

Rule 4: IF interest Rate is **NB** and CPI is **NB** and GDP is **PS** THEN consequent

Rule 5: IF interest Rate is **NB** and CPI is **NB** and GDP is **PB** THEN consequent

Rule 6: IF interest Rate is **NB** and CPI is **NS** and GDP is **NB** THEN consequent

...

Rule 125: IF interest Rate is **PB** and CPI is **PB** and GDP is **PB** THEN consequent

It can be seen that the first 25 rules look at interest rate when it is **NB**, the next 25 is where it is **NS** and so on. If Rule 2 was not learnt by the genetic algorithm, then as it is in the first 25 rules, the rule would be set to **NB**. If Rule 30 had not been learnt, it would be set to **NS**. As the current interest rate is the most important indicator in determining the following quarter's interest rate, it seems likely that the consequence would be the same as the current interest rates membership function.

4.10 Testing FKB on Interest Rate Test set

When the genetic algorithm has completed learning the FKB and unlearnt rules have been set to their default values, the fuzzy logic system can be tested on all the available data. Figure 4.11 shows the performance of the fuzzy logic system using the FKB. The fuzzy logic system can predict each quarter's interest rate for the training data and the test data.

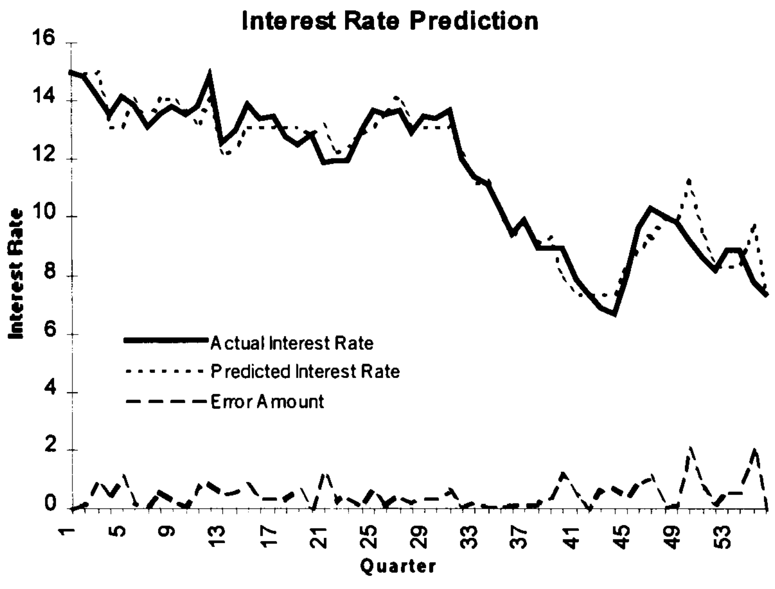

Figure 4.11 Interest rate prediction using test and training quarters

As Figure 4.11 shows, the system is quite capable of predicting the interest rate in most situations with a good degree of accuracy. However, as to be expected, the system accuracy falls away for data it has not been trained on. There are several reasons why this may occur. The two main reasons are:

1. It has not seen the data pattern before and is using the fuzzy rules that have not been trained,
2. There are more economic indicators that affect the following quarters interest rate than used in the system.

The first reason can only be rectified by using more data in the training cycle, however this option may be unavailable if there is only a small set of data. This then falls into the problems of time series prediction as mentioned by Ruelle (Ruelle, 1989). For the predictions used in this chapter, we used only three indicators. There are many more economic indicators that effect the fluctuations of the interest rate. The next chapter considers additional economic indicators for the fuzzy logic system and uses genetic algorithm to learn the FKB of the fuzzy logic system.

Chapter 5 Hierarchical Fuzzy Logic system

5.1 Introduction

In this chapter, we develop a number of single fuzzy logic systems. These are then combined into a Hierarchical Fuzzy Logic system to predict interest rates. Specifically, we look at why a Hierarchical Fuzzy Logic system is important compared to a single fuzzy logic system and compare the results from single fuzzy logic systems with a Hierarchical Fuzzy Logic system.

In the previous chapter, we developed a fuzzy logic system that predicts the following quarters interest rate using the current quarters interest rate, gross domestic product (GDP) and consumer price index (CPI), available from the Australian Bureau of Statistics (Madden, 1995). From the results obtained, the system was able to predict some of the fluctuations in the interest rate, but in many instances there was a fairly large difference from the actual rate. We concluded that by adding more indicators, we may be able to increase the performance of the system.

In fuzzy systems, there is a direct relationship between the number fuzzy sets of input parameters of the system and the size of the FKB. Kosko and Isaka, (Kosko, B & Isaka, S., 1993) calls this the "Curse of Dimensionallity". The "curse" in this instance is that there is exponential growth in the size of the FKB.

$$k = m^n \tag{5.1}$$

where k is the number of rules in the FKB, m is the number of fuzzy sets for each input and n is the number of inputs into the fuzzy system.

As the number of fuzzy sets of input parameters increase, the number of rules increases exponentially.

There are a number of ways that this exponential growth in the size of the FKB can be contained. The most obvious is to limit the number of inputs that the system is using. However, this may reduce the accuracy of the system, and in many cases, render the system being modelled unusable. Another approach is to reduce the number of fuzzy sets that each input has. Again, this may reduce the accuracy of the system (Kosko, B. 1992). The number of rules in the FKB can also be trimmed if it is known that some rules are never used. This can be a time-consuming and tedious task, as every rule in the FKB may need to be looked at.

Some of the main economic indicators released by the Australian Government were discussed in Chapter Four. To reconsider, those indicators include:

- *Interest Rate* which is the indicator being predicted. The Interest Rate used here is the Australian Commonwealth government 10-year treasury bonds.
- *Job Vacancies* is where a position is available for immediate filling or for which recruitment action has been taken.
- *The Unemployment Rate* is the percentage of the labour force actively looking for work in the country.

- *Gross Domestic Product* is an average aggregate measure of the value of economic production in a given period.
- *The Consumer Price Index* is a general indicator of the rate of change in prices paid by consumers for goods and services.
- *Household Saving Ratio* is the ratio of household income saved to households disposable income.
- *Home Loans* measure the supply of finance for home loans, not the demand for housing.
- *Average Weekly Earnings* is the average amount of wages that a full time worker takes home before any taxes.
- *Current Account* is the sum of the balances on merchandise trade, services trade, income and unrequited transfers.
- *Trade Weighted Index* measures changes in our currency relative to the currencies of our main trading partners.
- *RBA Commodity Price Index* provides an early indication of trends in Australia's export Prices.
- *All Industrial Index* provides an indication of price movements on the Australian Stock Market.
- *Company Profits* are defined as net operating profits or losses before income tax.
- *New Motor Vehicles* is the number of new vehicles registered in Australia.

By creating a system that contained all these indicators, we would be in a much better position to predict the fluctuations in interest rates. Unfortunately, a fuzzy logic system that used every indicator and had five fuzzy sets for every indicator would result in a large FKB consisting of over six billion rules! As can be imagined, this would require large computing power to not only train the fuzzy logic system with a genetic algorithm, but also large storage and run-time costs when the system is operational. Even if a computer could adequately handle this large amount of data, there is still the problem, as discussed by Ruelle (Ruelle, 1989), about having enough data to properly train every possible rule. It is very unlikely that the time series being modelled or predicted has enough data to properly train the FKB. Raju, G.V.S., Zhou, J. and Kisner, R.A. (Raju, G.V.S., Zhou, J. and Kisner, R.A., 1991) suggested using a Hierarchical Fuzzy Logic structure for the fuzzy logic system to overcome this problem. By using a Hierarchical Fuzzy Logic system, the number of fuzzy rules of the system is reduced, hence computational times are decreased resulting in a more efficient system.

5.2 Hierarchical Fuzzy Logic Systems

The Hierarchical Fuzzy Logic structure is formed by having the most influential inputs as the system variables in the first level of the hierarchy, the next important inputs in the second layer, and so on. If the Hierarchical Fuzzy Logic structure contains n system input parameters and L number of hierarchical levels with n_i the number of variables contained in the ith level, the total number of rules k is then determined by:

$$k = \sum_{i-1}^{L} m^{n_i} \tag{5.2}$$

where m is the number of fuzzy sets.

The above equation means that by using a Hierarchical Fuzzy Logic structure, the number of fuzzy rules for the system is reduced to a linear function of the number of system variables n, instead of an exponential function of n as is the conventional case (Raju et al, 1991).

The first level of the hierarchy gives an approximate output, which is then modified by the second level rule set, and so on. This is repeated for all succeeding levels of the hierarchy. One problem occurs when it is not known which inputs to the system have more influence than the others. This is the case when using the economic indicators discussed earlier in the chapter. Statistical analysis could be performed on the inputs to determine which ones have more bearing on the interest rate, however, without the advise of a statistician, it may be difficult to decide which statistical method to use.

The method used in this thesis is to split the inputs into a number of related groups. These inputs in these groups are related because they have some common connection between the inputs, such as dealing with employment, or imports and exports. This changes the hierarchy into a two level hierarchy, with the outputs from all the groups in the top layer giving their results as inputs into the bottom layer (Figure 5.1).

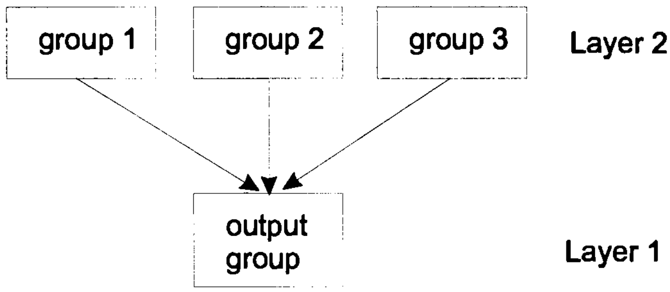

Figure 5.1 Example of Hierarchical Fuzzy Logic Structure

Using the economic indicators already indicated we can develop five separate groups. These groups are as follows:

1. **Country Group** -This group contains Gross Domestic Product and Consumer Price Index.
2. **Employment Group** - This group contains the unemployment rate and the job vacancies indicators.
3. **Savings Group** - This group contains House Hold Saving Ratio, Home Loans and Average Weekly Earnings.
4. **Company Group** - This group contains All Industrial Index, Company Profit and New Motor Vehicles indicators.
5. **Foreign Group** - This group contains Current Account, Trade Weight Index and also the RBA Commodity Index.

These five groups each produce a predicted interest rate for the next quarter. These are then fed into the next layer of the hierarchy where the final predicted interest rate is found, as shown in Figure 5.2 below. For each of these groups, the current quarter's interest rate is included in the indicators used, as hypothesised in chapter four, the current interest rate has the biggest influence on the following quarters interest rate.

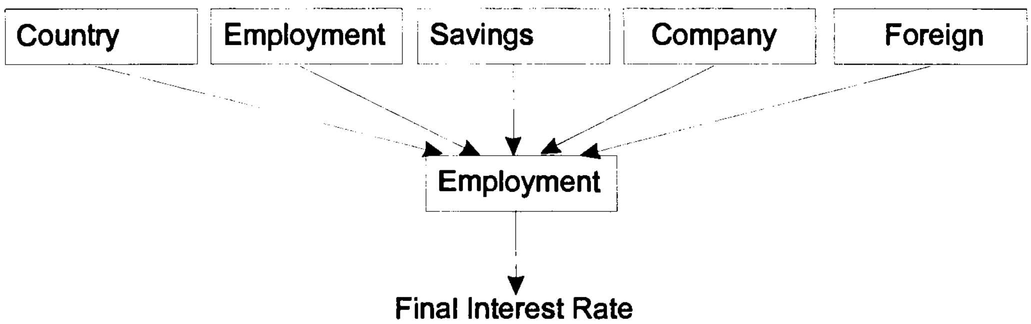

Figure 5.2 Hierarchical Fuzzy Logic system for interest rate prediction

The five FKB created form the top layer of the hierarchy. They are connected together to form a final FKB. The final FKB uses the predicted interest rate from the five above groups to produce a final interest rate prediction.

The advantage of using this Hierarchical Fuzzy Logic structure is that the number of rules used in the FKB's has been reduced substantially. Using the five fuzzy sets for each indicator, the following number of rules is required for each group.

Country Group 125 fuzzy rules,

Employment Group 125 fuzzy rules,

Savings Group 625 fuzzy rules,

Company Group 625 fuzzy rules,

Foreign Group 625 fuzzy rules.

The fuzzy logic system in the bottom layer of the hierarchy uses 3125 rules. This results in the total number of rules being required to be 5250. This is a significant reduction in the number of rules required from the 6 billion if we used a traditional fuzzy logic system with a single FKB.

In the following sections, we first create each of the fuzzy logic systems required for the top layer of the hierarchy. We then combine first 2, then 3, 4 and finally all 5 groups together to form the final Hierarchical Fuzzy Logic system to predict the quarterly interest rate in Australia.

5.3 Learning a Fuzzy Knowledge Base for each Fuzzy Logic system group

In this section we develop the FKB for each group using the genetic algorithm as described in Chapter Four. The genetic algorithm uses the same crossover and mutation rate as discussed in Chapter Four (section 4.7).

5.3.1 Country Knowledge Base

The country knowledge base contains information relating to countries current economic performance. These indicators are

Consumer price index,

Gross domestic product.

These are the same indicators that we used in the previous chapter (see Figure 4.11).

As a measure of how well a group predicts the following quarters interest rate, we calculate the average error of the system for the training set and tests sets. This is calculated using the following formula:

$$E = \frac{\sum_{i=1}^{n} abs(Pi - Ai)}{n} \tag{5.3}$$

where E is the average error, *Pi* is the Predicted interest rate at time period *i*, *Ai* is the actual interest rate for the quarter and *n* is the number of quarters predicted.

The table below shows the results for the Country group.

Training Average	Test Average	Overall Average
0.357	0.653	0.447

Table 5.1 Average error for Country Group

Table 5.1 shows the training average error is less than the test average error as the test set uses data that has not been used in the training of the FLGA system. The overall average error, which includes all the available data, is 0.447.

5.3.2 *Company Knowledge Base*

The Company knowledge base contains information relating to the corporate sector of the market. This information includes :

All Industrial Index,

Company Profit,

New Motor Vehicle Registrations.

These three indicators, combined with the Interest Rate, are used to predict the following quarters interest rate. Figure 5.3 shows the progress the GA made in learning the "optimal" FKB.

It started with a fitness of just 3.48 and finished with a fitness value of 30.18. The fitness of the "best" FKB found by the genetic algorithm started to even out after the 1000th generation when the fitness was 29.02. There were a few advances in the fitness value after this generation, however these were only small compared to the initial learning.

Figure 5.4 shows the predicted interest rate for both the training and test data on the "best" FKB found by the genetic algorithm.

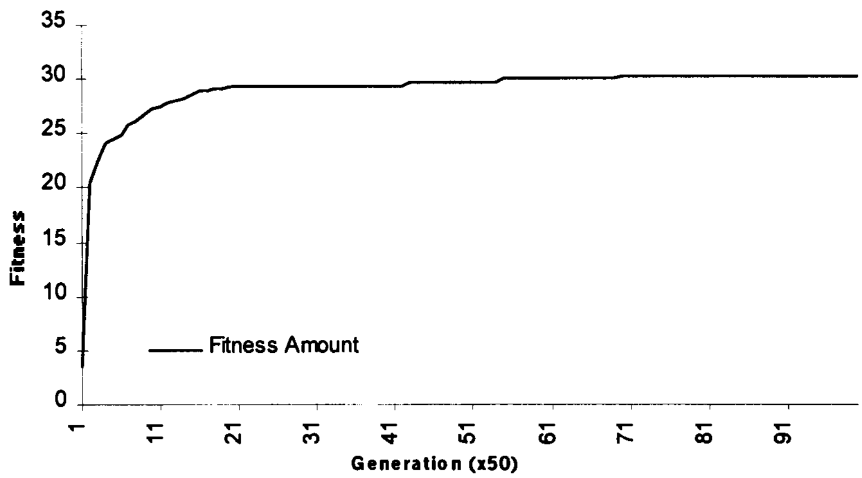

Figure 5.3 Fitness amount over training generations

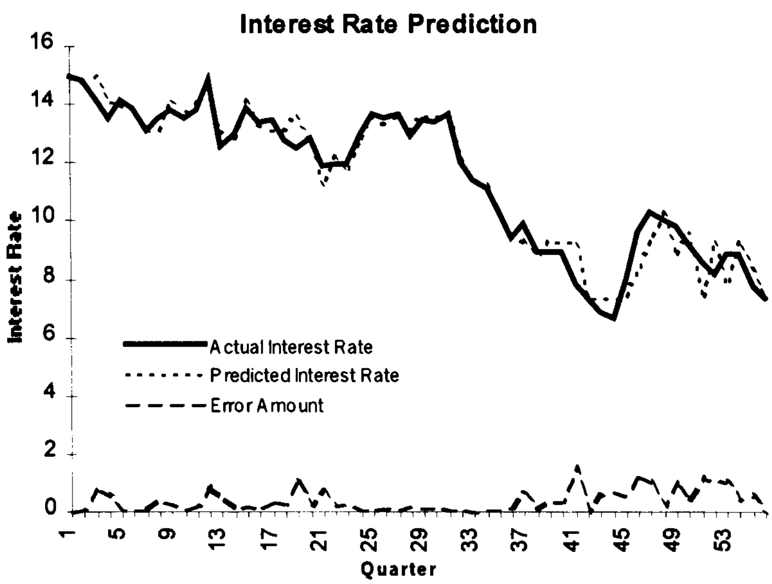

Figure 5.4 Predicted Interest Rate using FL of Company group

As Figure 5.4 shows, it follows a similar pattern as to the predictions from the Country group used in the previous chapter. Using the training data, the system is able to learn a FKB that predicts the following quarters interest rate with only a few fluctuations from the actual interest rate. However, the same problems occur when predicting the interest rate on the test data. The performance of the FKB for predicting the interest rate on the test data is slightly better than that achieved by the FKB in the Country group.

The average error of the Company group of indicators is shown in Table 5.2. It shows that there was a slight decrease in the average error of the simulation when compared to the Country group in Table 5.1. However, the test set average error is larger than that of the Country group, suggesting that the test data uses a number of rules that were not trained while the country group used more trained rules from the FKB.

Training Average	Test Average	Overall Average
0.252	0.691	0.385

Table 5.2 Average error for Company Group

5.3.3 Employment Knowledge Base

The Employment knowledge bas contains information relating to the employment sector of the economy. This information includes :

Unemployment Rate,

Job Vacancies.

These two indicators, combined with the Interest Rate, are used to predict the following quarters interest rate. Figure 5.5 shows the progress the GA made in learning the "optimal" FKB. It started with a fitness of 5.74 and finished with a fitness value of 23.3825. The fitness of the "best" FKB found by the genetic algorithm started to even out after only 300 generations when the fitness was found to be 22.656. This is a lot quicker than the previous groups we have looked at. There were a few advances in the fitness value after this generation, however these were only small compared to the initial learning.

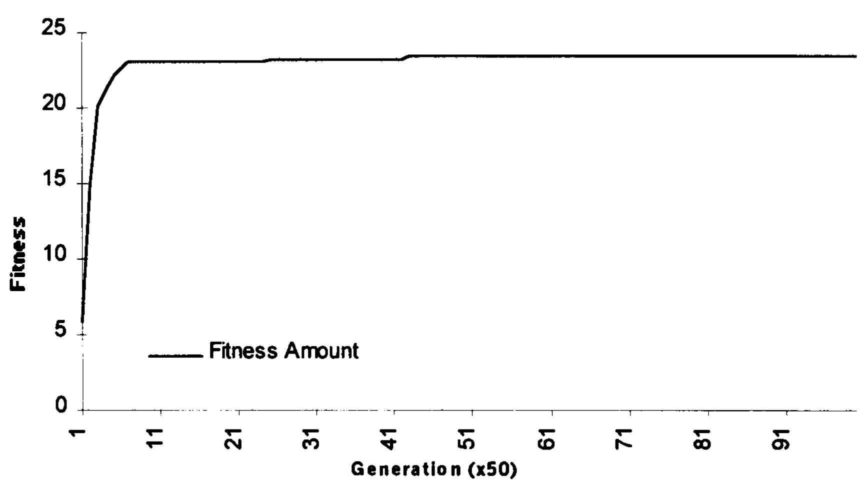

Figure 5.5 Fitness amounts over training generations

Figure 5.6 shows the predicted interest rate for both the training and test data on the "best" FKB found by the genetic algorithm.

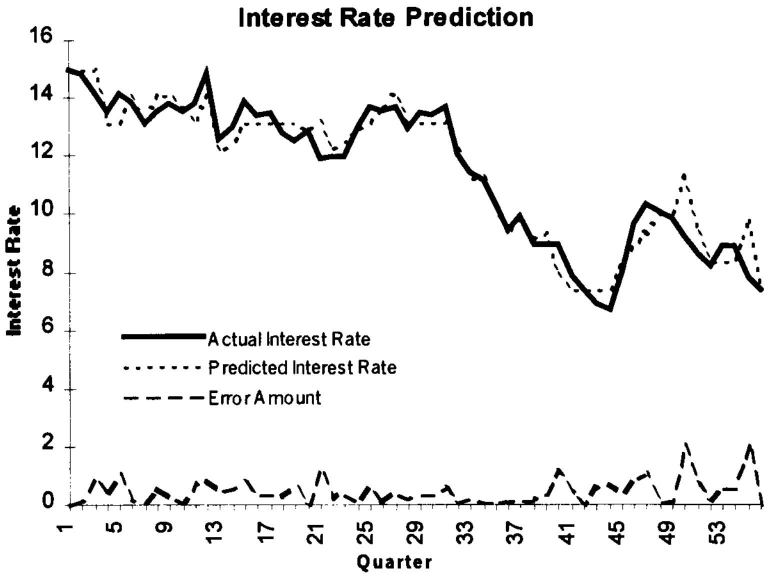

Figure 5.6 Predicted Interest Rate using Employment Group

As Figure 5.6 shows, the FKB generated for this group does not perform quite as well as the previous groups. Using the training data, the system is able to learn a FKB that predicts most of the following quarters interest rate, with some fluctuations from the actual interest rate. The performance of the FKB for predicting the interest rate on the test data looks slightly worse than that achieved by the FKB in the Country or Company groups.

The average error of the Employment group of indicators is shown in Table 5.3. It shows that there was an increase in the amount of average error of the fuzzy logic system for both the training and test data sets. This confirms our initial conclusions of the FKB from Figure 5.6 that the Test Average error was larger than the Training period Average Error.

Training Average	Test Average	Overall Average
0.427	0.578	0.473

Table 5.3 Average Error for the Employment Group

5.3.4 Savings Knowledge Base

The Savings knowledge base contains the following indicators:

Savings Ratio,

Home Loan approvals,

Average Weekly Earnings.

These three indicators, combined with the Interest Rate, are used to predict the following quarters interest rate. Figure 5.7 shows the progress the GA made in learning the "optimal" FKB. It started with a fitness of 2.04 and finished with a fitness value of 30.137. Unlike the previous groups looked at, the Savings group had a gradual increase in the fitness of the "optimal" FKB found by the FLGA system. There was a rapid increase in the fitness for the first 100 generations and then a gradual increase in the fitness until the 3205th generation where the "best" fitness was found to be 30.137. From this generation on there were no further increases in the fitness amount.

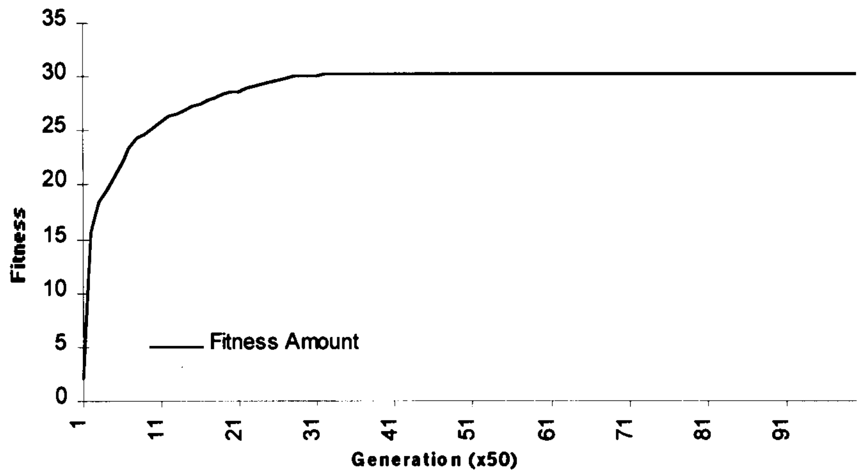

Figure 5.7 Fitness Amounts over training generations

Figure 5.8 shows the predicted interest rate for both the training and test data on the "best" FKB found by the genetic algorithm.

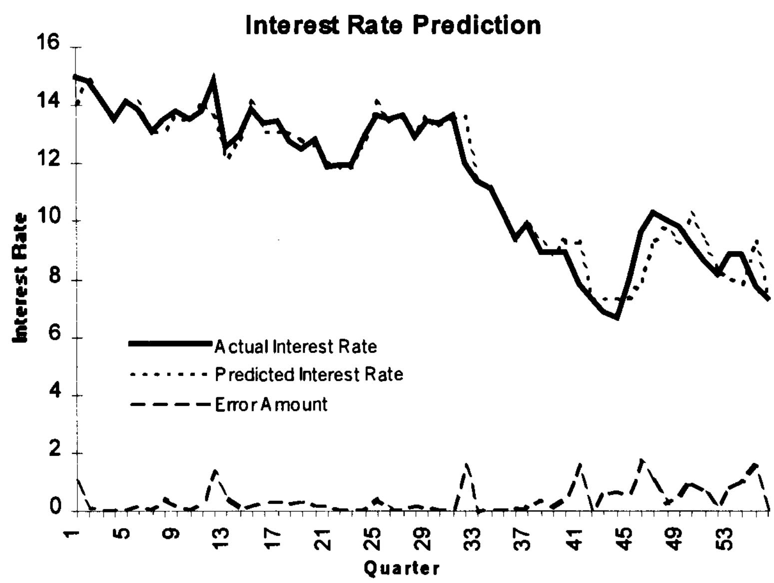

Figure 5.8 Predicted Interest Rate using Savings group

Figure 5.8 shows the FKB generated for the Savings group has a number of error amount peaks up as high as 2 percent, however during the training period, there is only the two peaks with the others happening during the test period. This compares well with the other groups looked at so far, as while there are some fairly large error amounts, there are a number of quarters where there is a very low amount of error.

The average error of the Savings group of indicators is shown in Table 5.4 below. The table shows that the savings group error amount compares with that achieved with the Company group (Table 5.2). However, the average error for the test period is the highest found, mainly due to a few quarters in the test period with a large difference between the actual and predicted interest rate.

Training Average	Test Average	Overall Average
0.256	0.725	0.398

Table 5.4 Average Error for Savings Group

5.3.5 Foreign Knowledge Base

The Foreign knowledge base contains information relating to Australia's current economic position in relation to the rest of the world. The indicators used are :

Current Account,

Reserve Bank of Australia Commodity Price Index,

Trade Weight Index.

These three indicators, when combined with the Interest Rate, are used to predict the following quarters interest rate. Figure 5.9 below shows the progress the GA made in learning the "optimal" FKB. It started with a fitness of 3.06 and finished with a fitness value of 28.26. The fitness achieved by the Foreign group follows a similar pattern to that of the Savings group. It has a rapid rise at the start of the training session and then it gradually improves the fitness.

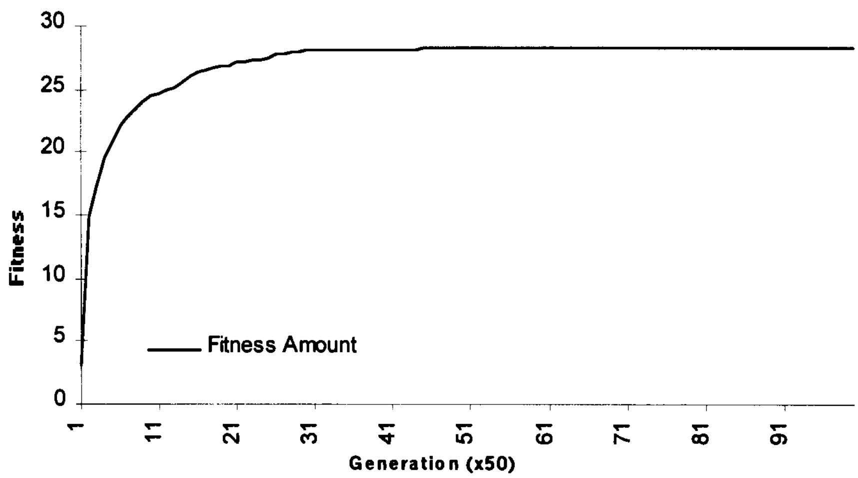

Figure 5.9 Fitness Amounts over training generations for Foreign Group

Figure 5.10 shows the predicted interest rate for both the training and test data on the "best" FKB found by the genetic algorithm.

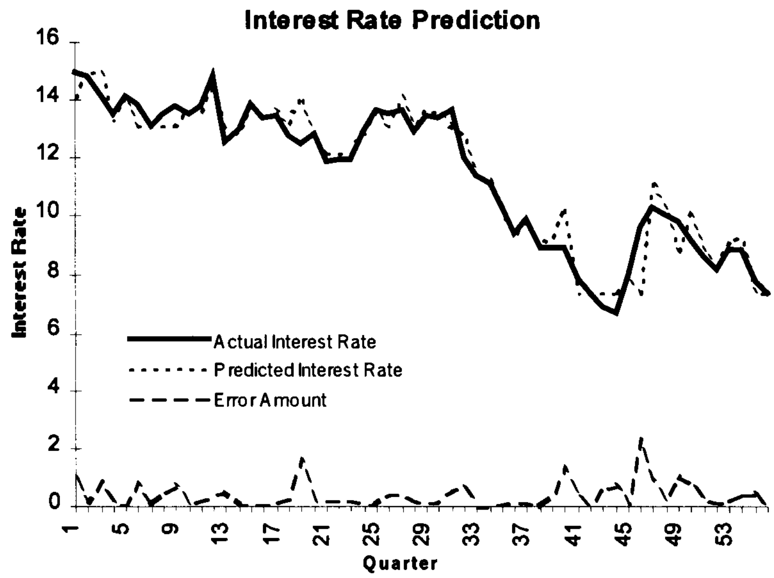

Figure 5.10 Predicted Interest Rate using Foreign group

As Figure 5.10 shows, the FKB generated for this group has a number of fluctuations in interest rate that is not accurately predicted by the FKB. In one quarter (quarter 46), there is a difference between the actual and predicted interest rate of more than three percent. However, the rest of the quarters perform better than this and compare favourably with previously generated FKB for other groups.

The average error of the Foreign group of indicators is shown in Table 5.5. It shows that the training average error amount is larger than that achieved by both the Savings and Company groups, but the average error for the combined data of training and test sets is the lowest achieved of any group. One of the reasons for this is that even though there was one quarter that performed badly, the other quarters compensated by performing well.

Training Average	Test Average	Overall Average
0.301	0.559	0.379

Table 5.5 Average Error for Foreign Group

5.4 Building the Hierarchy by combining the Fuzzy Logic systems

After creating the above fuzzy knowledge bases for each fuzzy logic system, we must combine them so that we can utilise the information they present and obtain better predictions of each quarters interest rate than any of the fuzzy logic system previously created.

To show the improvement in prediction using a Hierarchical Fuzzy Logic system, the following simulations were performed. A Hierarchical Fuzzy Logic system was randomly created by combining first two , then three, four and finally all five of the fuzzy logic systems from the previous section. The way these groups were combined to form the Hierarchical Fuzzy Logic system is shown in Table 5.6.

Combine 2 groups	**Combine 3 groups**	**Combine 4 groups**	**Combine 5 groups**
Company group	Company group	Company group	Company group
Country group	Country group	Country group	Country group
	Employment group	Employment group	Employment group
		Savings group	Savings group
			Foreign group

Table 5.6 Hierarchical system groups

5.4.1 Two Group Hierarchy

The two group Hierarchy combines the current quarters interest rate with the results from the Company and Country groups fuzzy logic systems (as shown in Figure 5.11). The results from these two groups are the predicted interest rate for the next quarter. The FLGA is run to find the "best" FKB using the same techniques as described in Chapter Four. The input and output parameters of the combined group fuzzy logic system had five fuzzy sets each.

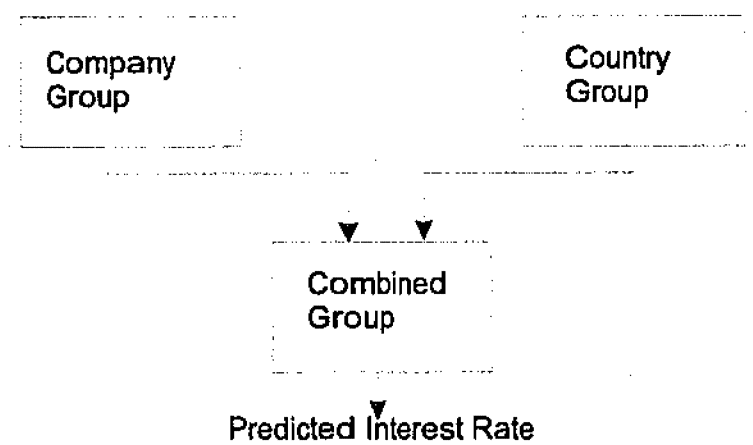

Figure 5.11 Combining Two Groups in the Hierarchy

Figure 5.12 shows the how the fitness improved over the training generations. There was a steep rise from the initial fitness of 7.42 to the maximum fitness value of 27.64. In fact, the FLGA found a "good" FKB after just 392 generations.

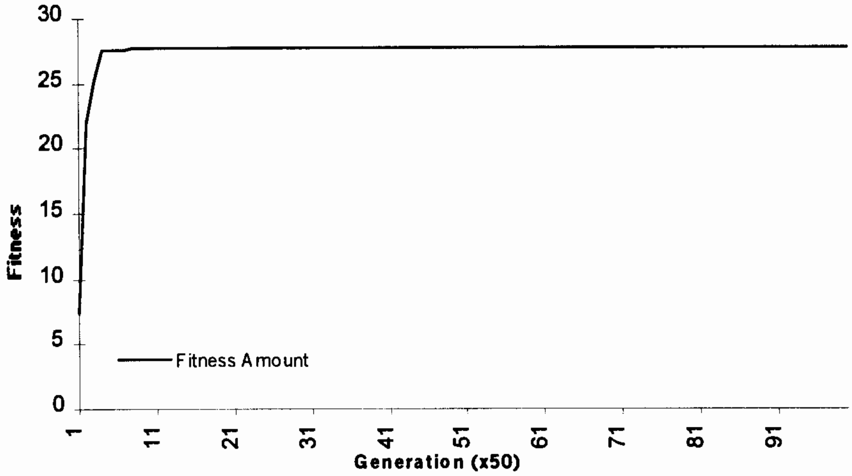

Figure 5.12 Fitness amounts during training for 2 group hierarchy

When the final FKB generated is tested on the training and test sets, the following results were obtained (Figure 5.13).

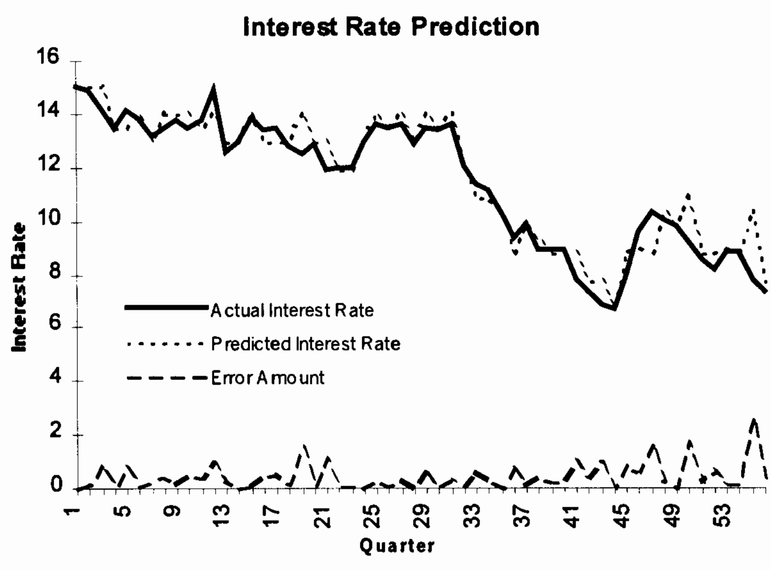

Figure 5.13 Predicted interest rate combining 2 groups

Figure 5.13 shows that when combining the Company and Country groups together in the hierarchy, the combined two group fuzzy logic system is able to predict the results over the training period with only a few fluctuations from the actual interest rate. During the test period, the system had a number of fluctuations between the actual interest rate and the predicted interest rate.

The average error for the system, shown in Table 5.6. The Hierarchical Fuzzy Logic system had a higher error amount for both the training and the testing periods when compared to the company group alone, but had lower error amounts when compared to the country group alone.

Training Average	Test Average	Overall Average
0.323	0.623	0.425

Table 5.7 Average Error Amounts for 2 group hierarchy

5.4.2 Three Group Hierarchy

The three group hierarchy is made by combining the outputs of three of the single layer groups into one final group (Figure 5.14). The inputs used in this system are:

Country Group output,

Company Group output, and

Employment Group output.

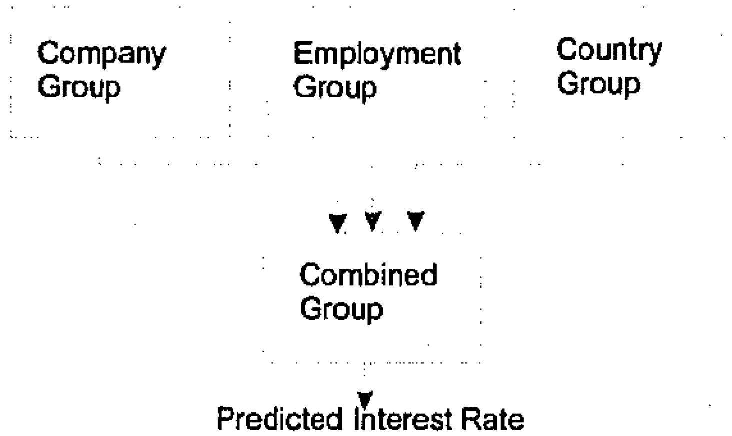

Figure 5.14 Combining Three Groups in the Hierarchy

These inputs are combined with the current quarters interest rate to create a system that predicts the following quarters interest rate. Figure 5.15 below shows the progress the FLGA system made in finding the "best" FKB. The system started with an initial fitness of 9.03, and rapidly increased to 28.29 by generation 264. By generation 1196, the maximum fitness value of 28.97 had been reached. This learning rate is very similar to that of the previous hierarchy (two groups hierarchy).

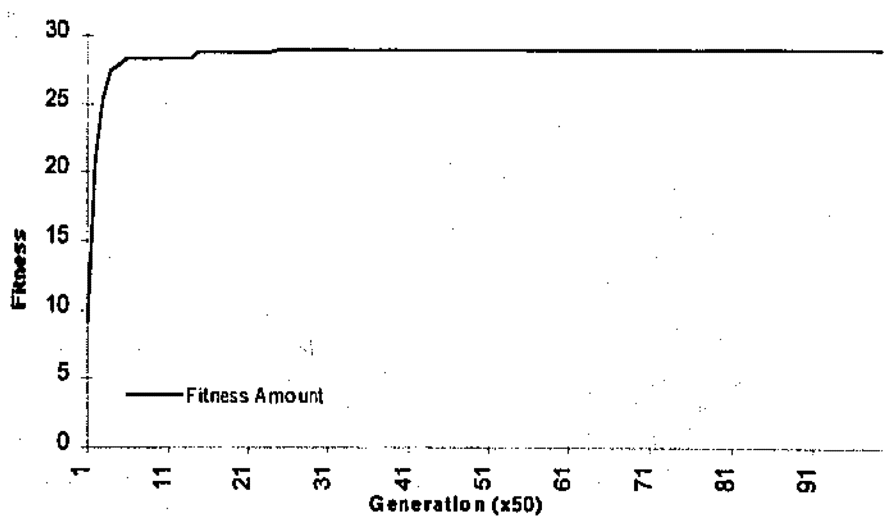

Figure 5.15 Fitness amount during training for 3 group hierarchy

The following figure (Figure 5.16) shows the results of the final FKB learnt by the system applied to both the training and test data.

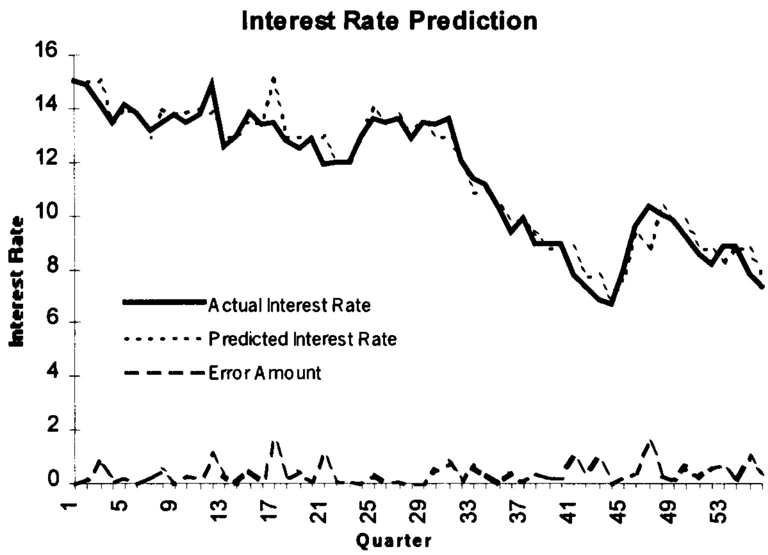

Figure 5.16 Predicted interest rate combining 3 groups

Figure 5.16 shows the results of the "best" FKB found in predicting the following quarters interest rate. There are a few minor differences in the training data, with three quarters having an error amount of greater than 1 percent. The results on the test data shows that the fuzzy logic system is able to model most of the fluctuations, even though it was not trained on that data. When compared to the previous hierarchy which used only two groups, the three group Hierarchical Fuzzy Logic system performed with more accuracy.

The average errors of the three group Hierarchical Fuzzy Logic system is shown in Table 5.8. The table shows that the three group Hierarchical Fuzzy Logic system had a lower average error over all sections of the data when compared to the two group system. When compared to the single layer systems, we find that the average error for the training data was similar to the best achieved, whilst the test average and overall average is the best of any group we have created so far. The best test average error previously achieved was 0.559 by the Foreign group, as compared to 0.494 shows a marked improvement in the prediction capabilities.

Training Average	Test Average	Overall Average
0.289	0.494	0.352

Table 5.8 Average Error Amounts for 3 group hierarchy

5.4.3 Four Group Hierarchy

The four group hierarchy is made by combining the outputs of four of the single layer groups into one final group (Figure 5.17). The inputs used in this system are:

Country Group output,

Company Group output,

Employment Group output, and

Savings Group output.

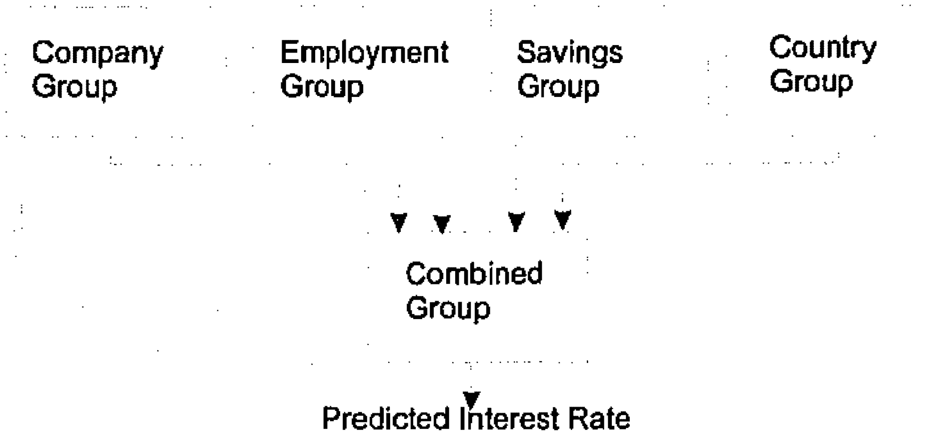

Figure 5.17 Combining Four Groups in the Hierarchy

The output from these groups are combined with the current quarters interest rate as inputs to create a fuzzy logic system that predicts the following quarters interest rate. The fuzzy logic system will be the largest one so far created as it will contain five inputs each split into five fuzzy sets which leads to 3125 rules in the FKB. Figure 5.18 below shows the progress the FLGA system made in finding the "best" FKB. The four group Hierarchical Fuzzy Logic system started with an initial fitness of 4.43, and increased to 31.04 by generation 394. Between this generation and generation 2621 there was a slow increase which led to the "best" fitness of 33.045 being achieved.

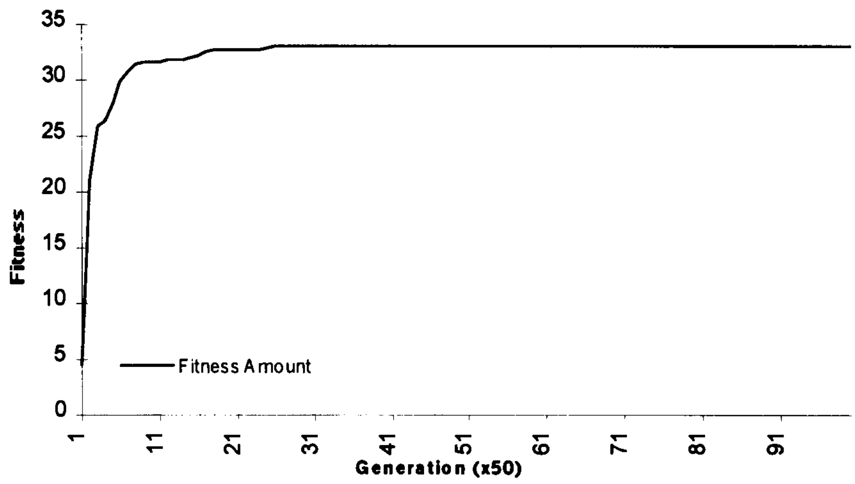

Figure 5.18 Fitness amount during training for 4 group hierarchy

Figure 5.19 shows the results of the final FKB learnt by the system applied to both the training and test data.

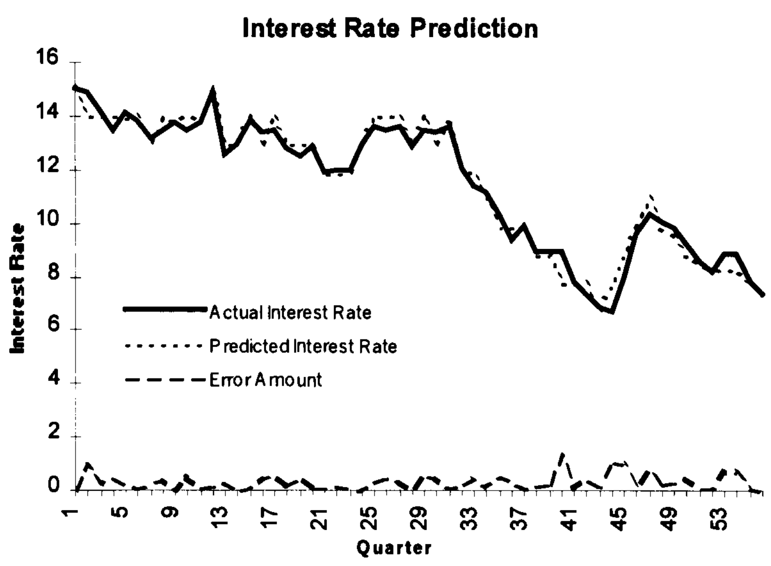

Figure 5.19 Predicted interest rate combining 4 groups

Figure 5.19 shows the results of the "best" FKB found in predicting the following quarters interest rate. The four group Hierarchical Fuzzy Logic system is able to predict the interest rate during the training period with a good accuracy. The error amount is fairly consistent with no large peaks, with the largest error amount being just under 1 percent. When predicting the interest rate for the test set, the error amount increases, but not by the amount previous graphs show (see Figure 5.16 and Figure 5.13). This means that the system is well able to predict the following quarters interest rate. Compared to the previous best system, that of the three group hierarchy, the graph indicates that the performance of the four group hierarchy is better.

Table 5.9 displays the average errors of the system. The table shows that the four group Hierarchical Fuzzy Logic system had the lowest average error for the test period and the overall data than any preceding system looked at whilst the training average error was the second best found, only beaten by the company group.

Training Average	Test Average	Overall Average
0.255	0.416	0.304

Table 5.9 Average Error Amounts for 4 group hierarchy

5.4.4 Final Combined Hierarchy

The final combined hierarchy is made by combining all the outputs from the groups in the single layer into one final group (Figure 5.20). The inputs used in this system are:

Country Group output,

Company Group output,

Employment Group output,

Savings Group output, and

Foreign Group output.

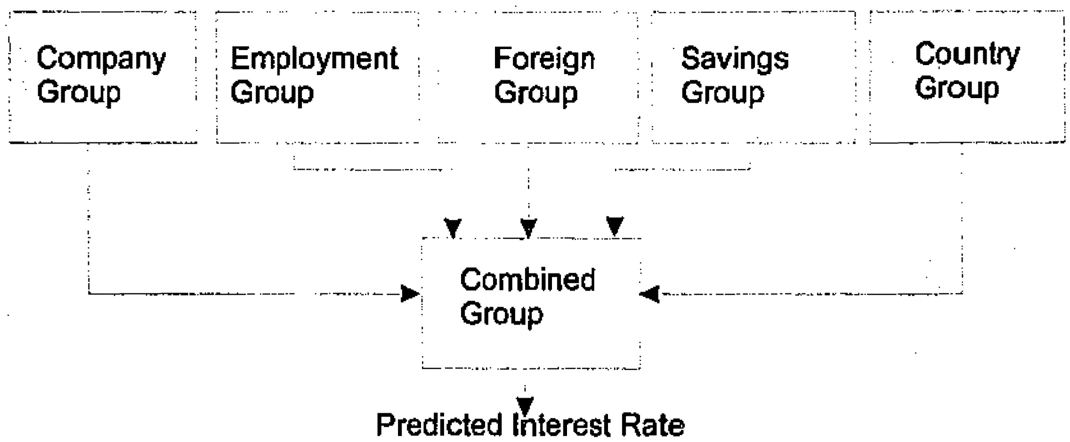

Figure 5.20 Combining all Five Groups in the Hierarchy

For this system, we will not include the current quarters interest rate, due to the fact that the size of the FKB has already rapidly grown to 3125 rules. If we included the current quarters interest rate, the size of the FKB would become over 15000 rules in size.

Figure 5.21 shows the progress the FLGA system made in finding the "best" FKB. The system started with an initial fitness of 6.77, and increased to 28.04 by generation 225. Between this generation and generation 2822 there was a slow increase which led to the "best" fitness of 32.20 being achieved. However, when compared to previous fitness results, this is not the best fitness result achieved on the training data, as the earlier four group hierarchy had a better fitness of 33.05.

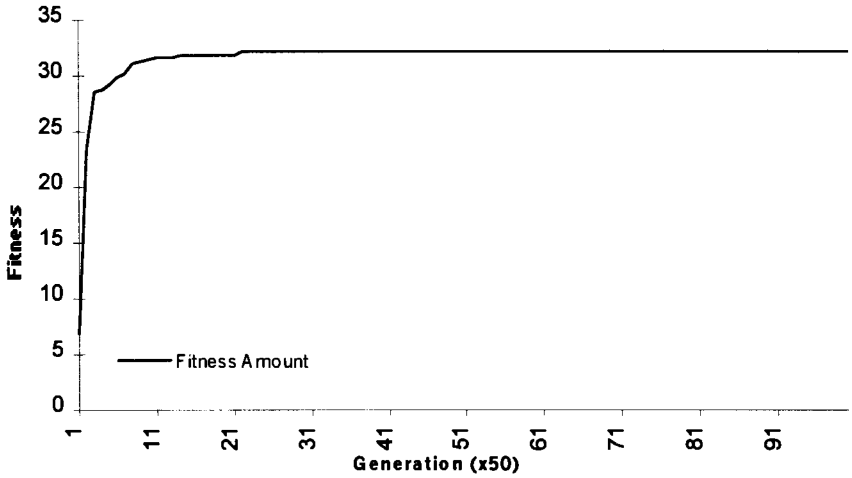

Figure 5.21 Fitness amount during training for final combined hierarchy

Figure 5.22 shows the results of the final FKB learnt by the system applied to both the training and test data.

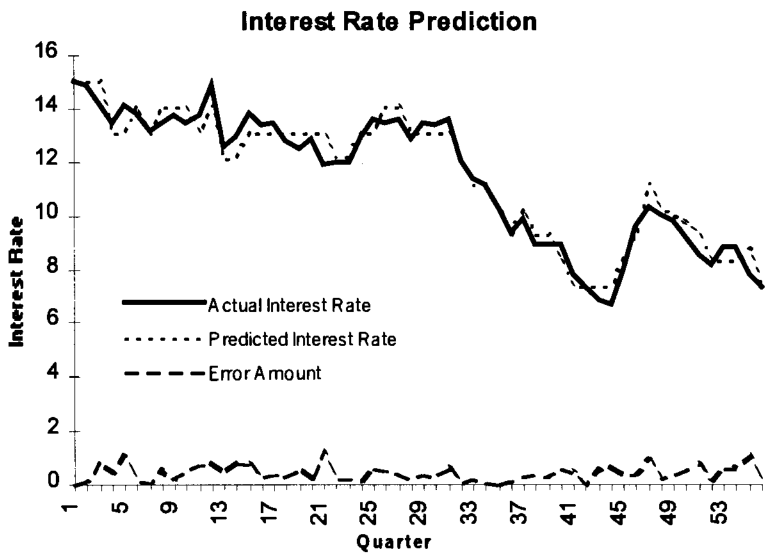

Figure 5.22 Predicted interest rate for final combined hierarchy

Figure 5.22 shows the results of the "best" FKB found in predicting the following quarters interest rate. Compared to the results from the four group hierarchy, the results achieved by the final combined hierarchy don't seem to be quite as good. Although the system was still able to predict the interest rate during the training period with a good deal of accuracy, by comparing the error amount there seems to be a larger error in the final combined hierarchy. However, comparing the test period prediction show that similar accuracy was found between the final combined and the four group hierarchies.

Table 5.10 displays the average errors of the system. The table shows that this system has a higher average error for the training period, in fact it is the second worse of all the systems tested (the worst being the Employment group - Table 5.3). However, the test period proved to be the second best found with only the four group hierarchy beating the performance. As the training average error was so high, this led to the overall average error to being higher than would otherwise be expected.

Training Average	Test Average	Overall Average
0.402	0.465	0.421

Table 5.10 Average Error Amounts for final combined hierarchy

5.5 Conclusions on Hierarchical Fuzzy Logic systems

When the separate fuzzy logic system groups are combined to form a Hierarchical Fuzzy Logic system, there is a definite improvement in the accuracy of the results in

Hierarchical Fuzzy Logic systems. Simulation results show that there is a gradual improvement in predicted results as we move from combining two groups to combining four groups in a hierarchy structure.

The final combined hierarchy had slightly worse results during the training period, mainly due to the fact that its inputs fully relied on the predictions of the single layer groups and didn't include the current interest rate as an input. The other Hierarchical Fuzzy Logic systems all combined the current interest rate with the their predicted interest rate inputs, which led to the four group hierarchy to be the best performing FKB, however the final combined hierarchy FKB still had the second best average error for the most important period - the test period. Table 5.11 displays all the errors amounts for the single groups and the combined groups.

	Training Error	Testing Error	Overall Error
Country Group	0.357	0.653	0.477
Company Group	0.252	0.691	0.385
Employment Group	0.427	0.578	0.473
Savings Group	0.256	0.725	0.398
Foreign Group	0.301	0.559	0.379
Two Group Hierarchy	0.323	0.623	0.425
Three Group Hierarchy	0.289	0.494	0.352
Four Group Hierarchy	0.255	0.416	0.304
Five Group Hierarchy	0.402	0.465	0.421

Table 5.11 Comparison of Average Errors

It seems that by combining the current interest rate with the single layer groups predicted interest rate gives the best result. If a single fuzzy logic system that used all the inputs had been developed, the system would have had over 6 billion rules. In the Hierarchical Fuzzy Logic system created in this chapter, we have shown that the accuracy of the system can be increased by using more indicators in a hierarchical structure, without the exponential growth in the number of fuzzy rules that occurs in a single fuzzy logic system with the same number of inputs.

In the next chapter, we look at using a new structure called the Feed Forward FLGA system, which uses the current interest rate in every group and passes the result from one group onto the next group in the structure.

Chapter 6 Feed Forward Fuzzy Logic System

6.1 Introduction

In the last chapter, we developed a Hierarchical Fuzzy Logic system where the economic indicators were split into a number of groups, and the results from each of these groups was fed into a final fuzzy logic system. In this chapter we look at another kind of hierarchical system called a Feed Forward system, where instead of the results all being passed into one combined system, the results from each group is passed on to next group in the list (hence its name of Feed Forward). Figure 6.1 below shows the structure of the Feed Forward system.

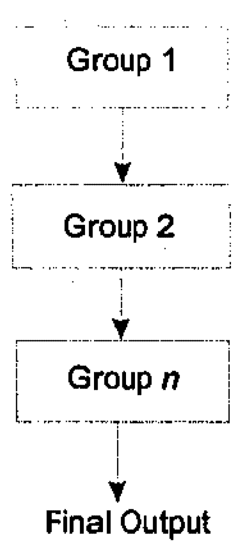

Figure 6.1 Feed Forward Fuzzy Logic structure

The output from Group 1 is passed onto Group 2 as one of its inputs, then the output from Group 2 is passed on to Group 3 as one of its inputs, and so on. The final output from Group *n* is the actual result for the system.

Using the same economic indicators as discussed in chapter four (and chapter five), these were split into the same groups as chapter five, as shown below.

Group Name	**Indicators**
Country Group	Consumer Price Index Gross Domestic Product
Company Group	Company Profit All Industrial Index New Motor Vehicles
Employment Group	Unemployment Rate Job Vacancies
Savings Group	Home Loans Savings Ratio Average Weekly Earnings
Foreign Group	Current Account Trade Weight Index RBA Commodity Price Index

Table 6.1 Economic indicators split into groups

For each group, the current quarters interest rate was also included as an indicator. Each group is trained to find a FKB that can predict the following quarters interest rate. This output is then used as an input for the next layer, and so on. The last group in the Feed Forward Fuzzy Logic structure provides the final predicted interest rate for the following quarter. The order in which the groups are processed is shown in Figure 6.2.

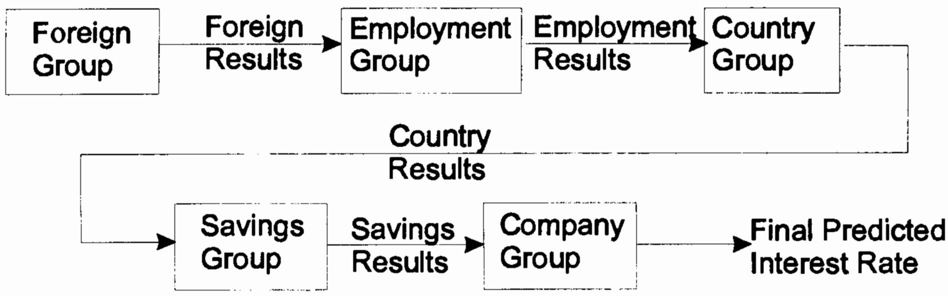

Figure 6.2 Interest rate prediction using Feed Forward Fuzzy Logic structure

Figure 6.2 shows that the output from the Foreign group is fed into the Employment group as another input, the output of the Employment group is fed into the Country group as an input, and so on until the output from the Company group is found. This output then represents the overall output for the system.

In the next sections, we show the results for each group in the Feed Forward Fuzzy Logic system, and then compare the results achieved to those from chapter five when a Hierarchical Fuzzy Logic system structure was used.

6.2 Foreign group

The Foreign group uses the following inputs:

Current Account,

Trade Weight Index,

RBA Commodity Index,

Current Quarter's interest rate.

These are the same indicators as used in the Foreign group in the previous chapter for Hierarchical Fuzzy Logic systems (section 5.1.5), so the same FKB as found in that section will be used here. For an explanation of the results of this group, refer to section 5.1.5.

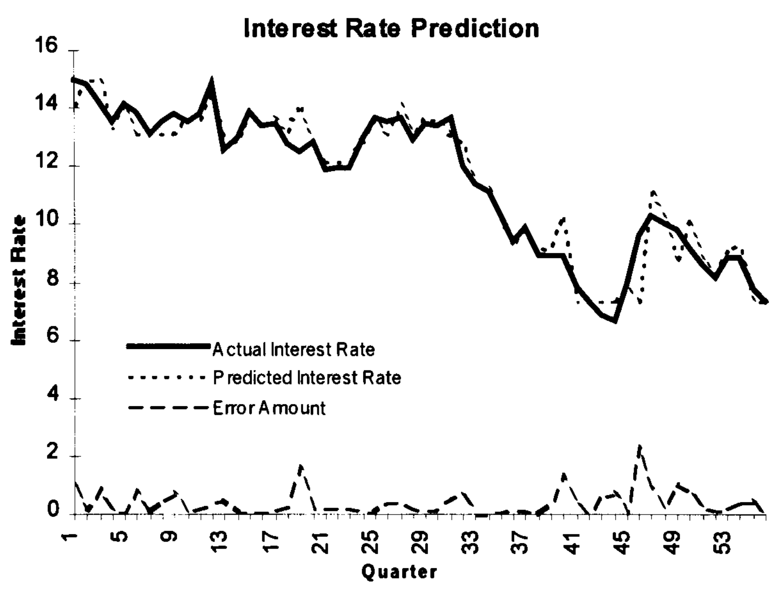

Figure 6.3 Predicted Interest Rate by Foreign Group

Training Average	Test Average	Overall Average
0.301	0.559	0.379

Table 6.2 Average Error for Foreign group using Feed Forward

6.3 Two group Feed Forward Fuzzy Logic structure

The second group in the Feed Forward Fuzzy Logic structure (Figure 6.4) consists of the following indicators:

Unemployment Rate,

Job Vacancies,

Current Quarters Interest Rate,

Foreign Group Results.

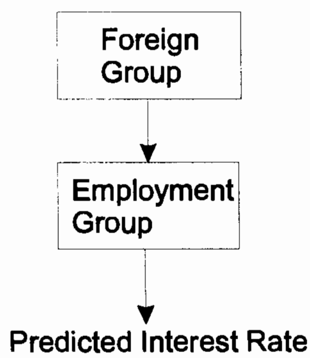

Figure 6.4 Two Group Feed Forward Fuzzy Logic System

These are the same indicators as used in the Employment group created in Chapter Five (5.1.3) except that the results from the Foreign group is also included as an indicator. Figure 6.5 shows the results of the Employment group when predicting the following quarter's interest rate.

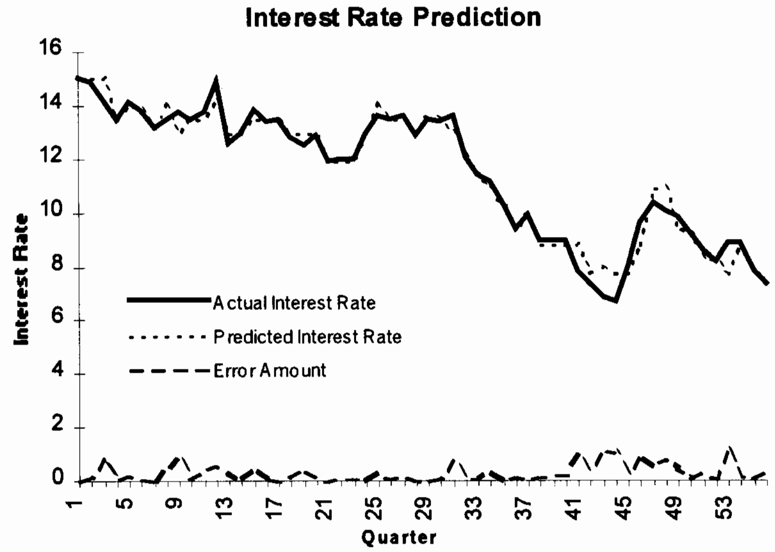

Figure 6.5 Results using Two group Feed Forward Fuzzy Logic system

As Figure 6.5 shows, by including the Foreign results as an input for the Employment group the performance has markedly improved from both the Foreign group result and the results achieved by the Employment group in section 5.1.3. Through both the training and also the test period, the average error is fairy small.

Training Average	Test Average	Overall Average
0.200	0.510	0.294

Table 6.3 Average Error for Two Feed Forward Fuzzy Logic groups

Table 6.3 shows the average error achieved by the system. As can be seen, the error amount has been reduced compared to the results from the Employment group in Section 5.1.3. (Table 5.3).

6.4 Three group Feed Forward Fuzzy Logic structure

The three group Feed Forward Fuzzy Logic structure (Figure 6.6) consists of the following indicators:

Gross Domestic Product,

Consumer Price Index,

Current Quarter's Interest Rate,

Result from Employment Group.

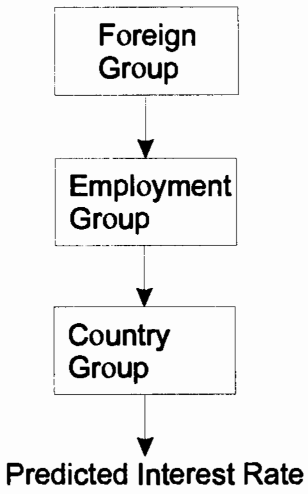

Figure 6.6 Three Group Feed Forward Fuzzy Logic System

These indicators are used to build the Feed Forward Fuzzy Logic structure that can predict the following quarter's interest rate. The results when using the Feed Forward Fuzzy Logic structure is shown in Figure 6.7.

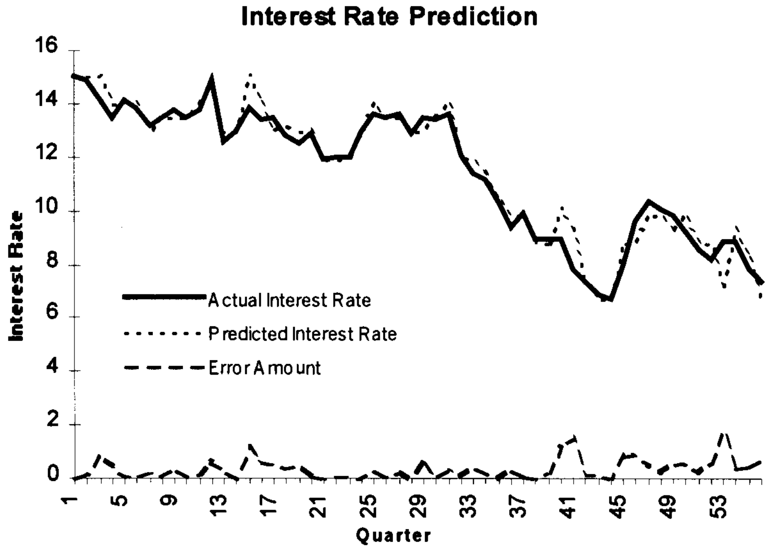

Figure 6.7 Results using a Feed Forward Fuzzy Logic system for Three groups

Figure 6.7 shows that the Country group didn't perform quite as well as the Employment group when predicting the following quarter's interest rate. During the training period, there are a couple of small peaks, with the largest error amount being about 1.5 percent, however, during the test period there were a number of peaks in the error amount with errors greater than 1 percent. Compared to the results for the previous section, there are more error peaks, but when compared with the country group results from chapter four, the peaks are fewer and these peaks are smaller than previously achieved.

Training Average	Test Average	Overall Average
0.248	0.618	0.360

Table 6.4 Average Error for Three Feed Forward Fuzzy Logic groups

The Average error achieved for the Country group in the Feed Forward Fuzzy Logic structure is shown in Table 6.4. When compared to the results of the Country group from chapter four, we see that the average error amounts have fallen, representing a better performance of the FKB for the Country group (Table 5.1).

6.5 Four group Feed Forward Fuzzy Logic structure

The Four group Feed Forward Fuzzy Logic system (Figure 6.8) consists of the following indicators:

Savings Ratio,

Home Loan approvals,

Average Weekly Earnings,

Current Quarter's Interest Rate,

Result from the Country Group.

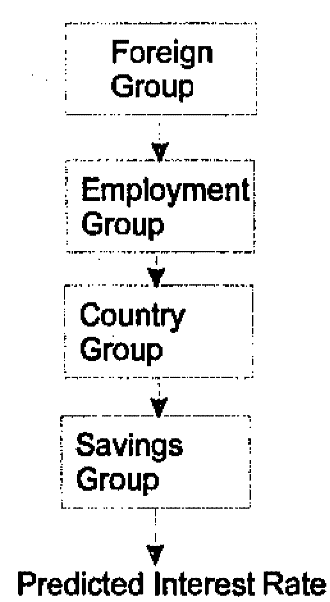

Figure 6.8 Four Group Feed Forward Fuzzy Logic System

These are the same indicators as used in section 5.1.4 when the Savings group FKB was created, but we now include the results from the Country group as an indicator. The FLGA system is run to find the "best" FKB that predicts the following quarter's interest rate. The results are shown in Figure 6.9 below.

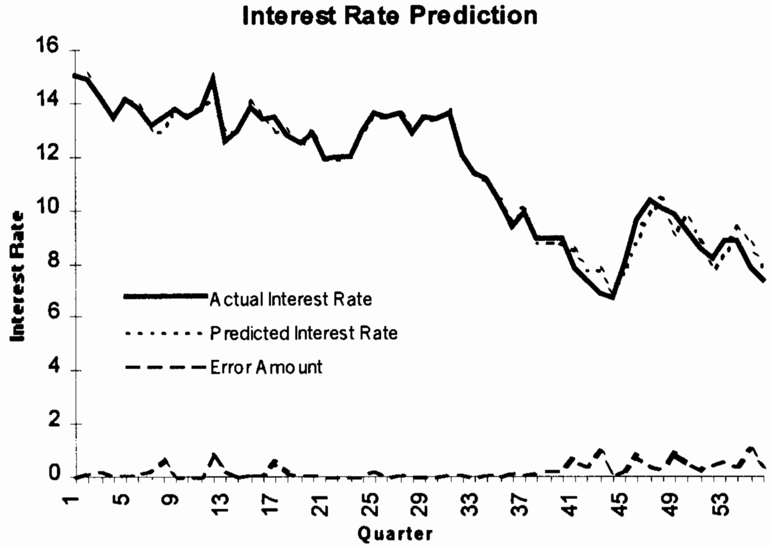

Figure 6.9 Results using Four Feed Forward Fuzzy Logic groups

The results from Figure 6.9 show that the system was very successful in predicting the following quarters interest rate for both the training period and the test period. During the training period, there are only a couple of very small peaks in the error amount, all these being 1 percent or lower, and during the test period, although there are a number of peaks in the error amount, they are all under 2 percent. Compared to the results from the two pervious groups in the Feed Forward Fuzzy Logic structure, the Savings group appears to be the best performance found so far. When compared to the Savings group in Chapter Five (5.1.4) the results during the test period seem to indicate that the Feed Forward Fuzzy Logic system results are better than those previously achieved (Figure 5.8).

Training Average	Test Average	Overall Average
0.120	0.472	0.227

Table 6.5 Average Error for Four Feed Forward Fuzzy Logic groups

Table 6.5 shows the average error that was achieved by the system. The table shows that the Training Average Error is significantly lower than that achieved so far in the Feed Forward Fuzzy Logic system. The test and overall average error is also lower than that previously achieved in the Feed Forward Fuzzy Logic system. When compared to the results from the Savings group in Chapter Five, the all three averages (namely Training, Test and Overall average) are significantly lower when using a Feed Forward Fuzzy Logic structure (Table 5.4).

6.6 Five group Feed Forward Fuzzy Logic structure

The Five group Feed Forward Fuzzy Logic system (Figure 6.10) contains the following indicators:

All Industrial Index,

Company Profit,

New Motor Vehicle Registration,

Current Quarter's Interest Rate,

Result from Savings Group.

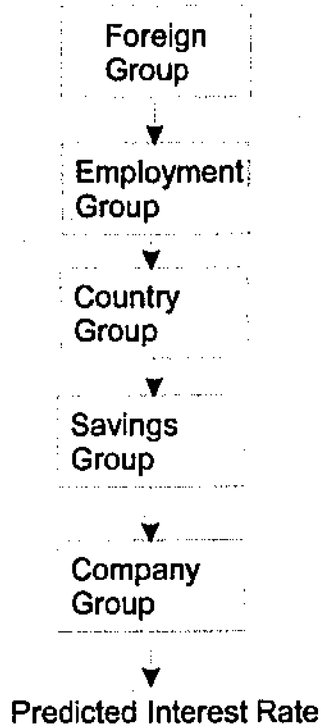

Figure 6.10 Five Group Feed Forward Fuzzy Logic System

These are the same indicators used as the Company group in chapter five (5.1.2) with the results from the Savings group added to the indicators. Figure 6.11 below shows the results when the "best" FKB that has been found during the training period is used to predict the following quarter's interest rate.

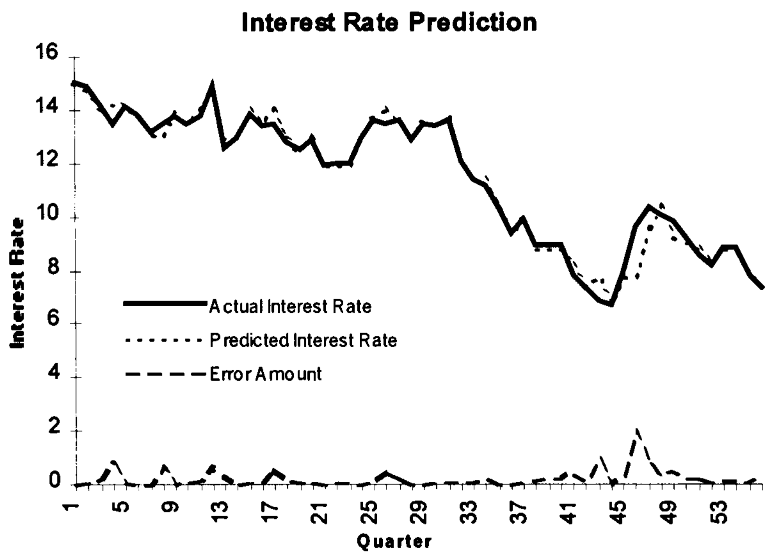

Figure 6.11 Results using Five Feed Forward Fuzzy Logic groups

The above Figure shows that the Company group was able to find a “good” FKB that was able to predict most of the fluctuations in the interest rate. During the training period, the interest rate followed very closely the actual interest rate for the following quarter. The test period shows that there was one main peak in the error amount, at the 47th quarter, but other than this peak, the error amount was fairly small.

Compared to the rest of the groups in the Feed Forward Fuzzy Logic system, the system perform better than all other groups, with the exception of the Four group Feed Forward Fuzzy Logic system (Figure 6.9) which had a number of small peaks in the error amount during the test period, compared to the one large peak in the Company group. Compared to the Company group in chapter five (5.1.2) the performance of the Feed Forward Fuzzy Logic system is superior both during the training and testing periods.

Training Average	Test Average	Overall Average
0.146	0.379	0.217

Table 6.6 Average Error for Five Feed Forward Fuzzy Logic groups

Table 6.6 shows that the average error obtained was the best performing of the Feed Forward Fuzzy Logic groups.

6.7 Comparison of Feed Forward Fuzzy Logic system with Hierarchical Fuzzy Logic system

There are a number of differences between the Hierarchical Fuzzy Logic system used in chapter five and the Feed Forward Fuzzy Logic system developed in this chapter. The Hierarchical Fuzzy Logic system has a number of self contained groups. The output from these groups is then combined into a final system which produces the final, predicted interest rate. In comparison, the groups in the Feed Forward Fuzzy Logic system rely on the output from the previous group in the system before they can predict the following quarters interest rate.

Table 6.7 shows the Average Error for the Hierarchical Fuzzy Logic system and the Feed Forward Fuzzy Logic system.

	Training Error	Test Error	Overall Error
Two Group Hierarchy	0.323	0.623	0.425
Three Group Hierarchy	0.289	0.494	0.352
Four Group Hierarchy	0.255	0.416	0.304
Five Group Hierarchy	0.402	0.465	0.421
One FFFL Group	0.301	0.559	0.379
Two FFFL Groups	0.200	0.510	0.294
Three FFFL Groups	0.248	0.618	0.360
Four FFFL Groups	0.120	0.472	0.227
Five FFFL Groups	0.146	0.379	0.217

Table 6.7 Comparison of Average Error Between Hierarchical and Feed Forward systems

Table 6.7 shows that the Feed Forward Fuzzy Logic results for the five group Feed Forward Fuzzy Logic system has a much lower average error for the training period compared to the hierarchical groups. However, the test period average errors are of similar values, resulting in the overall average error for the Feed Forward Fuzzy Logic system to be slightly better than the Hierarchical Fuzzy Logic system.

One reason that the Feed Forward Fuzzy Logic system seems to perform slightly better than a Hierarchical Fuzzy Logic system may be that if there is a large error in the error amount for a quarter, the other indicators in the Feed Forward Fuzzy Logic group compensate for this error and therefore reduce its significance. In the Hierarchical Fuzzy Logic system, if there is a large error in the predicted interest rate for the following quarter, then the system only has other predicted interest rates with which to work with, which may also have this large error.

One advantage that the Hierarchical Fuzzy Logic system has over the Feed Forward Fuzzy Logic system is that each group of the Hierarchical Fuzzy Logic system can be trained and run in parallel, reducing the amount of time necessary to train the system to obtain a final result. Figure 6.12 below shows this concept.

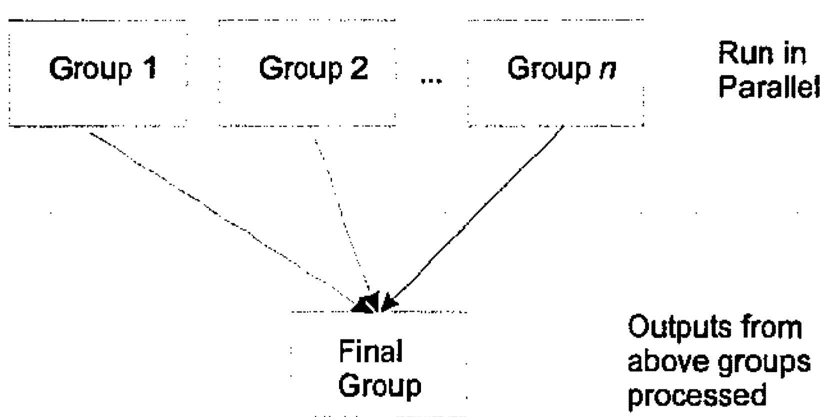

Figure 6.12 Parallel structure of Hierarchical groups

The top groups in the Hierarchical Fuzzy Logic system can be processed in parallel as they do not rely on information between them. It is only the final combined group that requires all the previously processed groups to be finished so it can process their information. The Feed Forward Fuzzy Logic system on the other hand can only operate in a sequential fashion. That is, the first group is processed, then the output from that group is passed on the next group and so on until the final group is processed.

Chapter 7 Artificial Neural Network for Prediction of Interest Rates

7.1 Introduction

Artificial Neural Networks (ANN's), which were introduced in chapter two, can be compared to a simplified mathematical model of a biological nervous system such as the brain. They are composed of a large number of highly interconnected processing elements (neurons) working together to solve specific problems. The processing ability of the ANN is stored as weights for each of the interconnected links between the neurons, usually obtained by some method of training or learning.

In this chapter, an ANN is created to predict the following quarter's interest rate in Australia. The system uses the same data as used by the FLGA system in the previous chapters. Two different methods using an ANN were used to predict the following quarter's interest rate. The first model uses the same structure as the FLGA where the input parameters are split into a number of smaller related groups and their output is fed into the final group which then produces the final interest rate prediction. This was the Hierarchical Neural Network system. The second model used the traditional neural network system where all the input parameters were presented to the system and an interest rate prediction was made.

7.2 Pre-Process Data

In order for the neural network to use the economic data for predicting the following quarter's interest rate, a number of pre-processing steps must be performed. This allows the data to be presented to the neural network in a format that it can easily work with. Data presented to the neural network must fall within certain ranges (usually 0 to +1 or -1 to +1 (Rao and Rao, 1994)) due to the fact that the network uses a Sigmoid Activation function (see Chapter 2) in its middle (or hidden) layers.

7.2.1 Calculate Difference

The neural network system formats the data for processing in a similar manner to the FLGA system where the difference from the current quarter to the previous quarter is used as the data for the input. The change from one quarter to the next is used by all the indicators except the interest rate, where the actual interest rate is used. For example the Gross Domestic Product would be formatted as:

Year	Quarter	Data	Difference
1986	1	79856.0	
1986	2	79520.0	-336.0
1986	3	79619.0	99.0
1986	4	79319.0	-300.0
1987	1	80201.0	882.0

Table 7.1 Difference in Data from Current Quarter to Previous Quarter

7.2.2 Normalise the Data

As Table 7.1 shows, there can still be a large range between the smallest and largest values. To reduce this into a more useable range, the data is modified by the following equation:

$$\textit{New Data} = (\textit{current data} - \textit{Mean}) / \textit{standard deviation} \tag{7.1}$$

The new data that has been calculated represents the distance from the mean value as a fraction of the standard deviation. This gives a good variability to the data, with only a few values that are out of the 0 to +1 or -1 to +1 range.

7.2.3 Squash the Data

The next step in the pre-processing stage is to squash the data so that it falls between the required range of 0 to +1 for this simulation. For this system, a Sigmoid squashing function is performed. The equation for this step is:

$$\textit{Squash data} = 1 / (1 + \exp(-\textit{Norm Data})) \tag{7.2}$$

After performing the sigmoid squashing function on the data, all the values fall in the range 0 to +1.

7.2.4 Moving Difference

As well as using the indicators as inputs for the neural network, we also present data to the system that relates to the rate of change in the data, which is the second derivative of the data set (Rao and Rao, 1994). This accents changes in the data set between one quarter and the next. The equation for this is:

$$\textit{mov diff} = (\textit{current val} - \textit{previous val}) / (\textit{current val} + \textit{previous val}) \tag{7.3}$$

The above equation is performed on the original data (before any pre-processing steps are performed) and will give a value between -1 and 1. The result from this equation becomes an additional input to the system. Therefore, for each input into the system, an additional input is created, doubling the number of input parameters to the neural network.

7.3 Training and results of the Neural Network

The indicators are split into the same groups as used by the FLGA system. A back-propagation neural network is used with two hidden layers, each consisting of 20 neurons. This was found to produce a quicker and more accurate result than using a single hidden layer. Sigmoid learning is used to predict the following quarters interest rate. The error tolerance was set to 0.0001, the Learning Parameter (Beta) was set to 0.6, momentum (alpha) and Noise Factor were both set to 0. The neural network was trained for 10000 cycles.

7.3.1 Country Group

The country knowledge base contains information relating to country's current economic performance. These indicators are:

Interest Rate,

Consumer price index,

Gross domestic product.

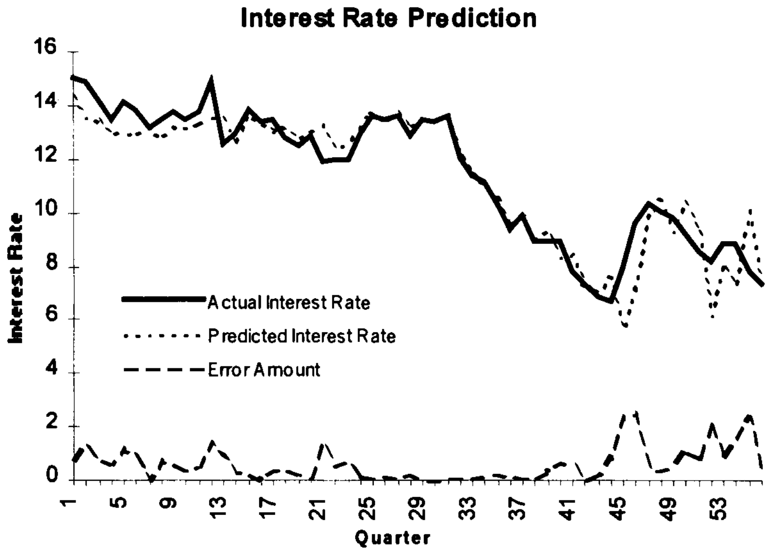

Figure 7.1 Neural Network Prediction for Country Group

Training Average	Test Average	Overall Average
0.401	1.023	0.591

Table 7.2 Average Error for Neural Network Country Group

7.3.2 Company Knowledge Base

The Company knowledge base contains information relating to the corporate sector of the market. This information includes:

Interest Rate,

All Industrial Index,

Company Profit,

New Motor Vehicle Registrations.

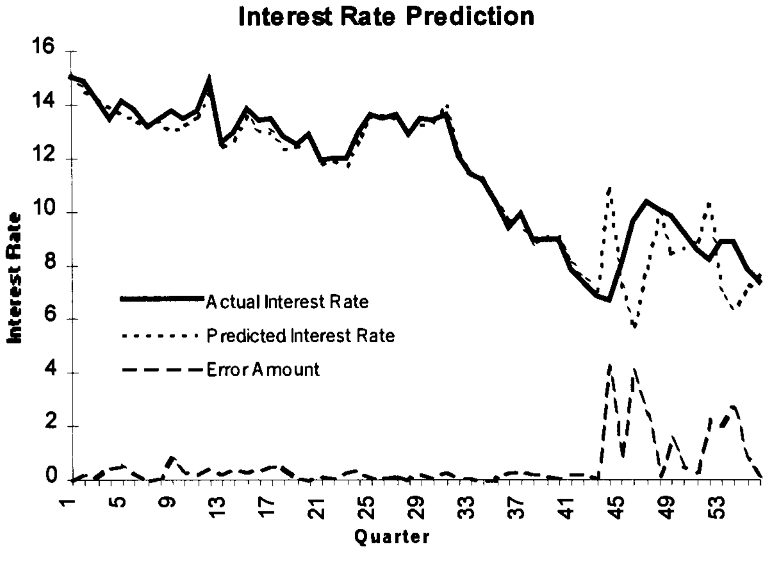

Figure 7.2 Neural Network Prediction for Company Group

Training Average	Test Average	Overall Average
0.228	1.290	0.548

Table 7.3 Average Error for Neural Network Company Group.

7.3.3 Employment Knowledge Base

The Employment knowledge base contains information relating to the employment sector of the economy. This information includes:

Interest Rate,

Unemployment Rate,

Job Vacancies.

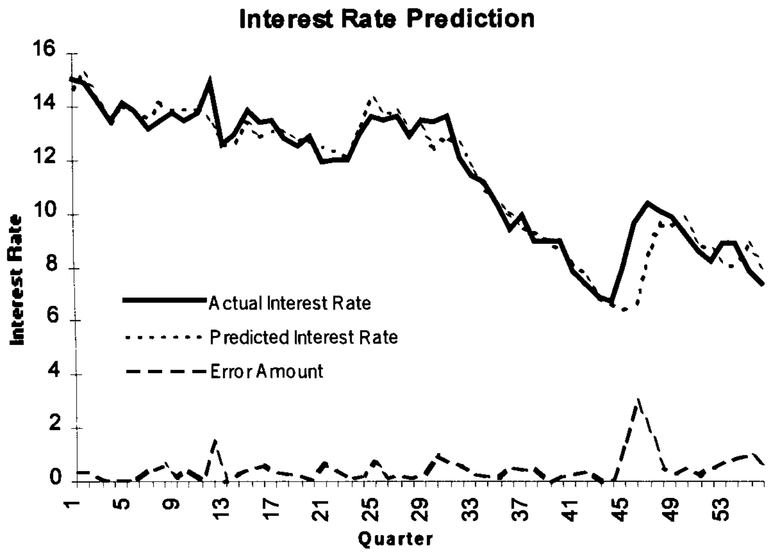

Figure 7.3 Neural Network Prediction for Employment Group

Training Average	Test Average	Overall Average
0.352	0.742	0.471

Table 7.4 Average Error for Neural Network Employment Group

7.3.4 Savings Knowledge Base

The Savings knowledge base contains the following indicators:

Interest Rate,

Savings Ratio,

Home Loan approvals,

Average Weekly Earnings.

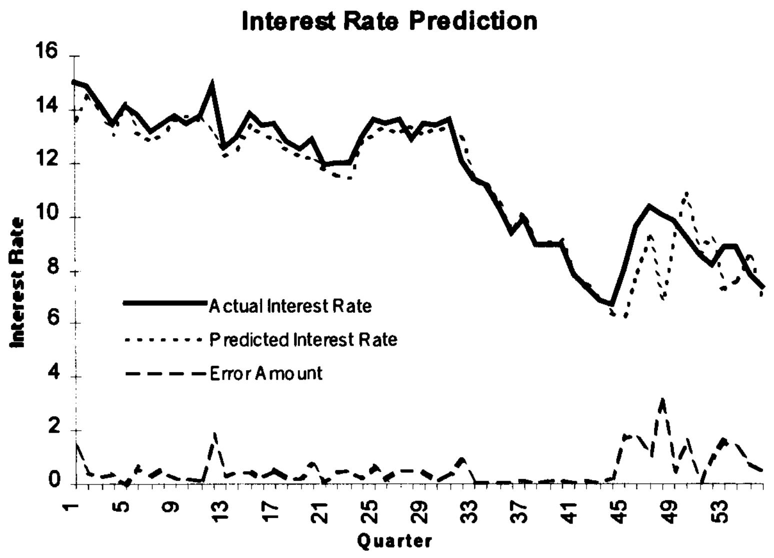

Figure 7.4 Neural Network Prediction for Savings Group

Training Average	Test Average	Overall Average
0.378	0.923	0.534

Table 7.5 Average Error for Neural Network Savings Group

7.3.5 Foreign Knowledge Base

The Foreign knowledge base contains information relating to Australia's current economic position in relation to the rest of the world. The indicators used are:

Interest Rate,

Current Account,

Reserve Bank of Australia Commodity Price Index,

Trade Weight Index.

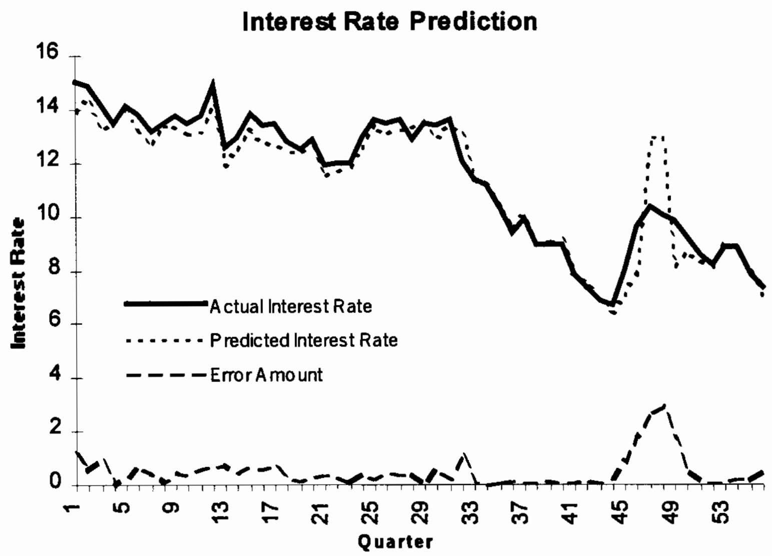

Figure 7.5 Neural Network Prediction for Foreign Group

Training Average	Test Average	Overall Average
0.387	0.714	0.487

Table 7.6 Average Error for Neural Network Foreign Group

7.3.6 Four Combined Hierarchical Neural Network System

The Four Combined Hierarchical Neural Network system uses the same inputs as the 4 group Hierarchical Fuzzy Logic system:

Interest Rate,

Country Group output,

Company Group output,

Employment Group output,

Savings Group output.

These inputs were normalised for the neural network in the same manner as previously stated in section 7.2. This gave 10 indicators to be presented to the neural network for training and testing of the system.

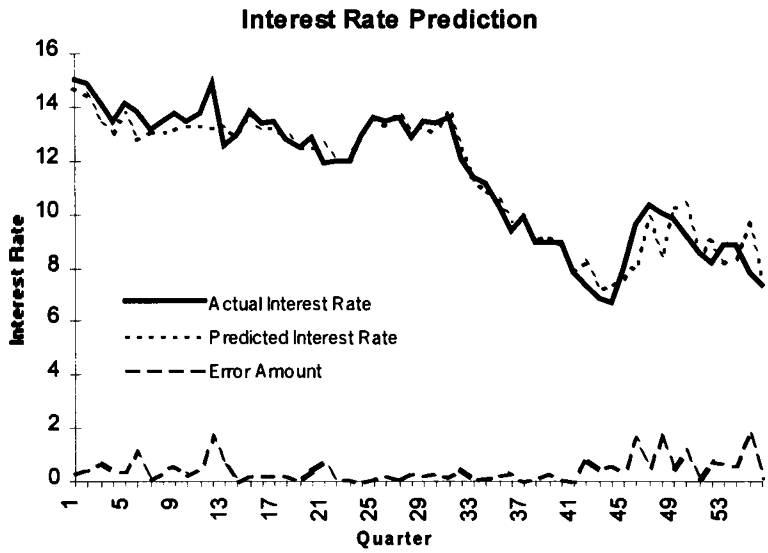

Figure 7.6 Neural Network Prediction for 4 Combined Groups

Training Average	Test Average	Overall Average
0.318	0.681	0.428

Table 7.7 Average Error for 4 Combined Neural Network Groups

7.3.7 Final Combined Hierarchical System

The final Hierarchical Neural Network system uses the same inputs as the Final Hierarchical Fuzzy Logic system. As can be seen, it does not include the current quarter's interest rate:

Country Group output,

Company Group output,

Employment Group output,

Savings Group output,

Foreign Group output.

These inputs were normalised for the neural network in the same manner as previously stated. This gave 10 indicators to be presented to the neural network for training and testing of the system.

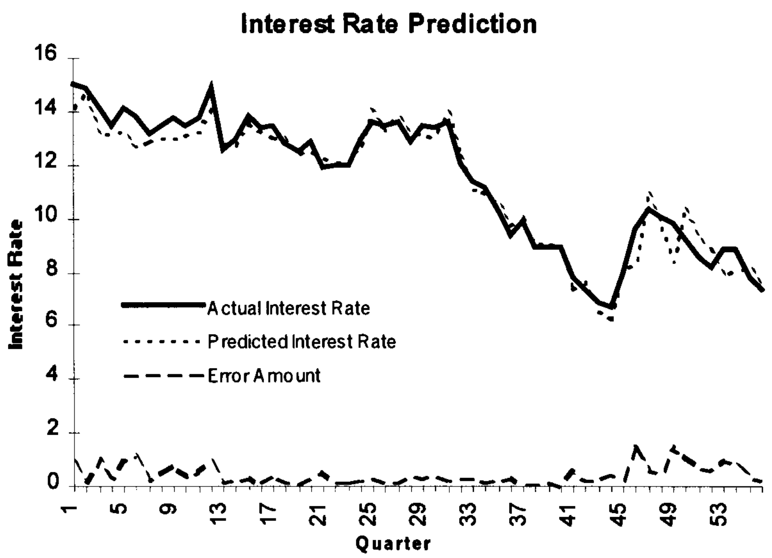

Figure 7.7 Neural Network Prediction for Final Combine Hierarchy

Training Average	Test Average	Overall Average
0.354	0.607	0.431

Table 7.8 Average Error for Final Combined Neural Network Groups

7.4 Neural Network with All Indicators

This system includes all the indicators as inputs into the neural network. This gives a system with 28 inputs (including the moving difference for each indicator). The same training parameters as used previously in Section 7.2 were used in this system. After 10000 cycles, the following results were achieved:

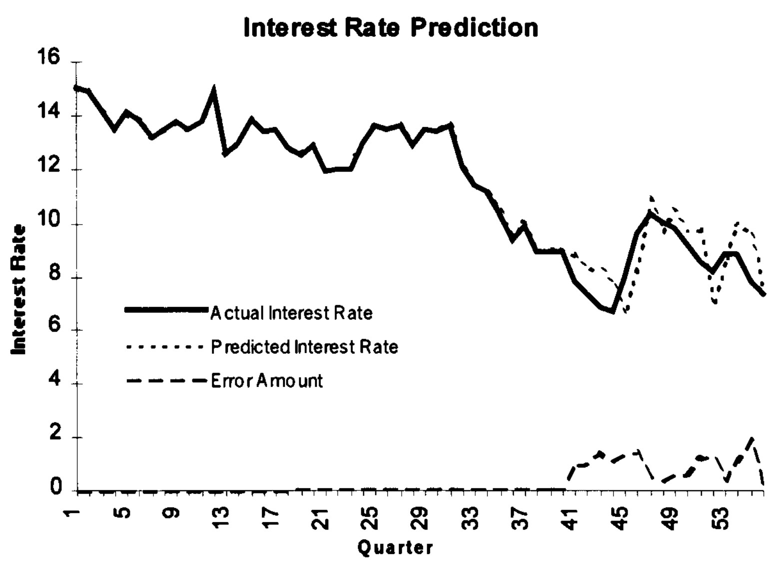

Figure 7.8 Neural Network results for All Indicators

Training Average	Test Average	Overall Average
0.039	0.880	0.296

Table 7.9 Average Error for All Indicators Neural Network

7.5 Comparison of Results

The results from the Hierarchical Neural Network system showed disappointing results when compared to the HFKB and Feed Forward systems. Table 7.10 shows the results of the neural network systems and the HFKB and Feed Forward Fuzzy Logic systems.

	Training Error	Test Error	Overall Error
Four Group Hierarchical Fuzzy Logic System	0.255	0.416	0.304
Five Group Hierarchical Fuzzy Logic System	0.402	0.465	0.421
Final Feed Forward Fuzzy Logic System	0.146	0.379	0.217
Four Group Hierarchical Neural Network	0.318	0.681	0.428
Five Group Hierarchical Neural Network	0.354	0.607	0.431
All Indicators Neural Network	0.039	0.880	0.296

Table 7.10 Compairson of Neural Network and FLGA systems

As Table 7.10 shows, the range of results for the different systems is very diverse. The Training Average Error, which is the average error recorded during the training period of 40 quarter's, ranges from a high value of 0.402 for the Five Group Hierarchical FKB down to an almost perfect 0.039 for the All Indicator Neural Network. The All Indicator Neural Network was able to learn the training data almost perfectly, with the Final Feed Forward FKB having the second best results of 0.146. The Hierarchical Fuzzy Logic systems did not learn the training data as well as these two systems, but still had a quite low training error.

The Test Average Error is the average error recorded during the test period, which is where the system is presented with inputs that it has not been trained on. The best result for the test average error was achieved by the Final Feed Forward Fuzzy Logic system with a result of 0.379. The two Hierarchical Fuzzy Logic systems achieved the next best results with 0.416 for the Four combined Fuzzy Logic system and 0.465 for the Final combined fuzzy logic system. These compare favourably with all the Neural Network systems which, although having similar overall average error results to the Hierarchical Fuzzy Logic systems, had disappointing test average error results. In fact, the neural network system with the best training average error had the worse test result of all the tested systems. This may be a case of overtraining the network.

These results show that the FLGA systems (both the Hierarchical and Feed Forward Fuzzy Logic systems) produce better Test Average Errors results than the neural network systems when trained for just 40 quarter's. These results may change when the system is trained for longer time periods.

Chapter 8 Long Term Predictions

8.1 Introduction

So far we have looked at predicting the following quarter's interest rate, that is three months from the current quarter. There are a number of situations in which this time period is too short a prediction length, such as when investors have to decide whether to move from the bond market into the property market before the end of the financial year.

In this chapter, a Hierarchical Fuzzy Logic system for predicting interest rates six months (half yearly) from the current quarter is developed, followed by a Hierarchical Fuzzy Logic system that predicts the interest rate one year ahead of the current quarter. We then compare these results to a Hierarchical Neural Network system for predicting the same time periods.

8.2 Predicting Six Months Ahead using a Hierarchical Fuzzy Logic system

Predicting three months ahead is sometimes not a valid length of time, especially in government where they need to decide economic policy that will effect the country in the next half of the financial year. To accommodate these requirements, the

Hierarchical Fuzzy Logic system we developed over chapters four and five will be extended to now predict two quarters ahead (six months) instead of one quarter.

The indicators used to predict two quarters ahead are the same as used previously in chapter four, as are the groups they have been split into. The following sections show the results of the FKB that the FLGA has produced on the separate groups in predicting the interest rate, followed by the combined hierarchical groups.

8.2.1 Company Group prediction

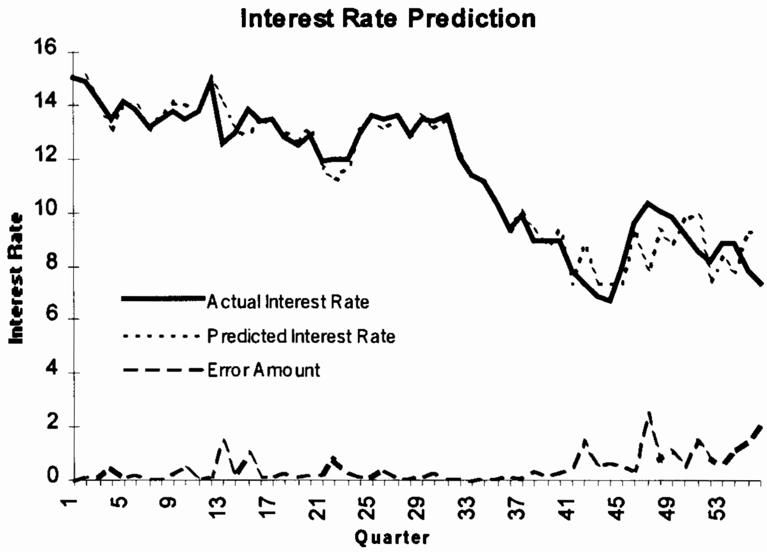

Figure 8.1 Six Month predicted rates using Company group

Training Average	Test Average	Overall Average
0.233	0.963	0.455

Table 8.1 Six Month Average Errors for Company Group

8.2.2 *Country Group prediction*

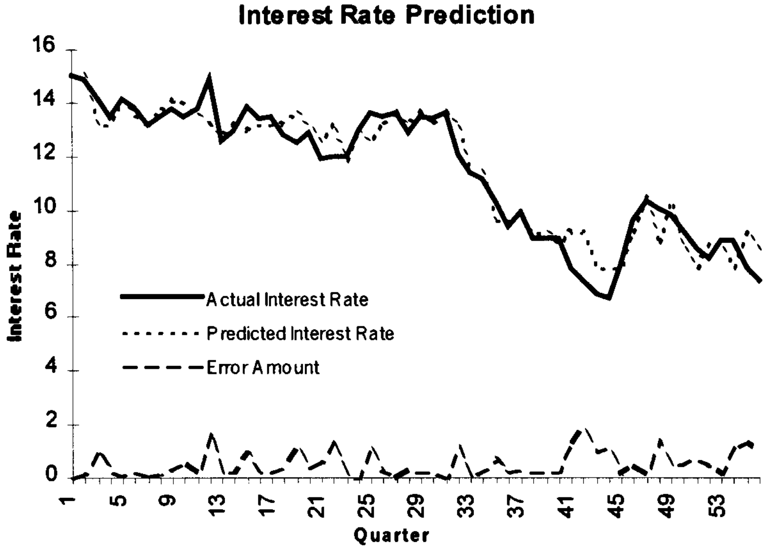

Figure 8.2 Six Month predicted rates using Country group

Training Average	Test Average	Overall Average
0.393	0.798	0.516

Table 8.2 Six Month Average Errors for Country Group

8.2.3 Foreign Group prediction

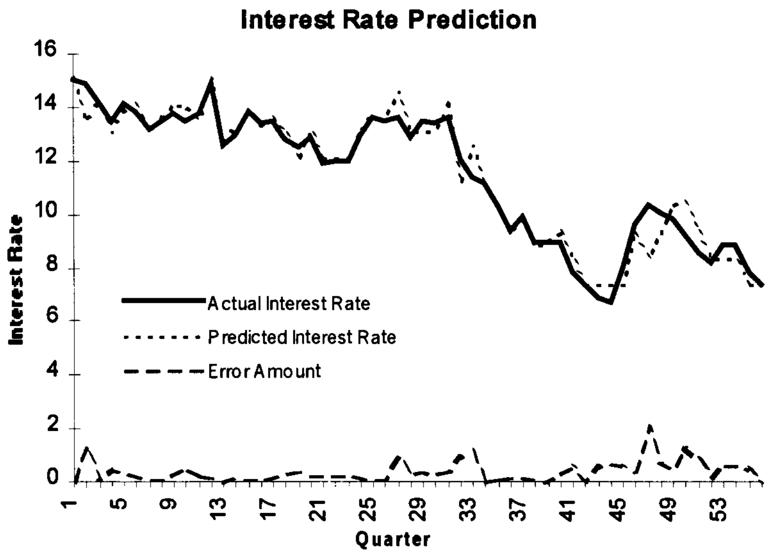

Figure 8.3 Six Month predicted rates using Foreign group

Training Average	Test Average	Overall Average
0.267	0.585	0.363

Table 8.3 Six Month Average Errors for Foreign Group

8.2.4 Savings Group prediction

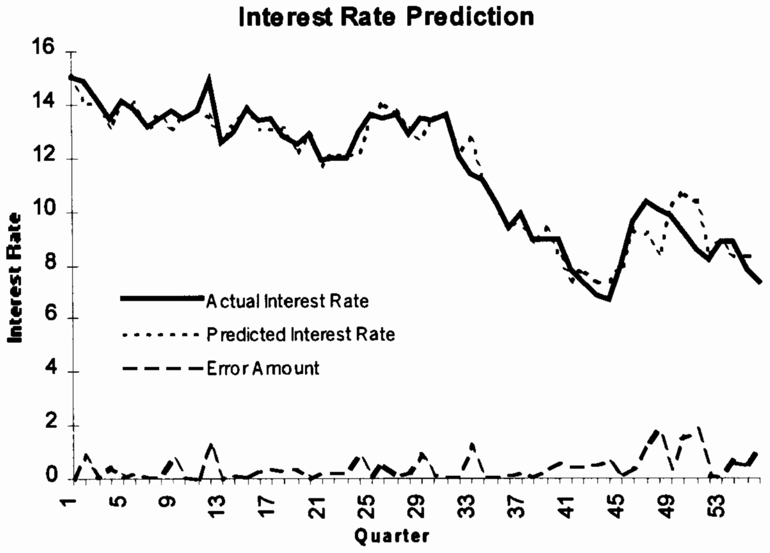

Figure 8.4 Six Month predicted rates using Savings group

Training Average	Test Average	Overall Average
0.281	0.708	0.410

Table 8.4 Six Month Average Errors for Savings Group

8.2.5 Employment Group prediction

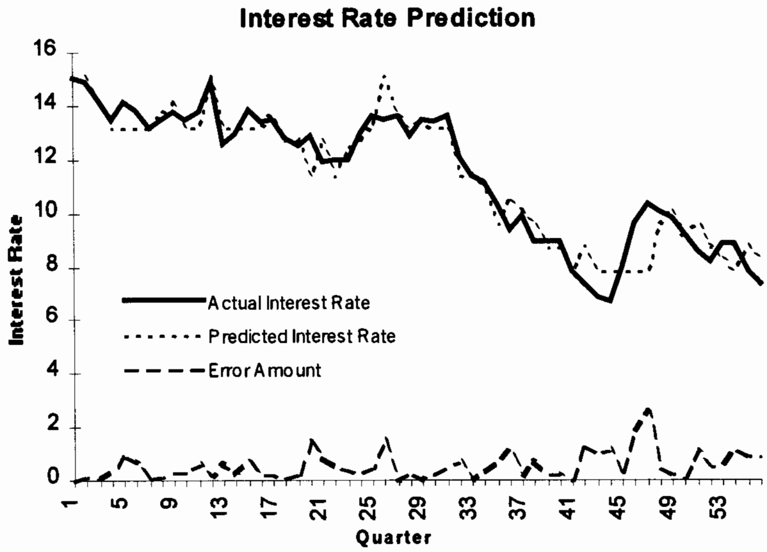

Figure 8.5 Six Month predicted rates using Employment group

Training Average	Test Average	Overall Average
0.408	0.819	0.533

Table 8.5 Six Month Average Errors for Employment Group

8.2.6 Two Group Hierarchy

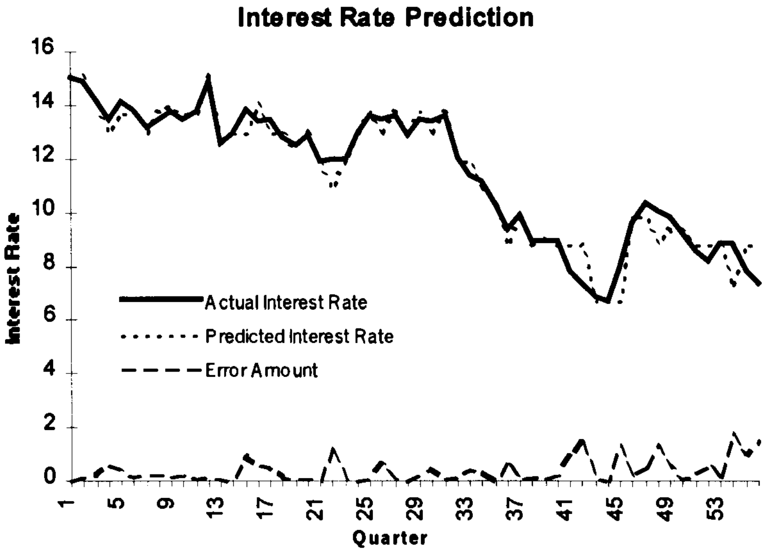

Figure 8.6 Six Month predicted rates Combining Two groups

Training Average	Test Average	Overall Average
0.408	0.819	0.533

Table 8.6 Six Month Average Errors for 2 Group Hierarchy

8.2.7 Three Group Hierarchy

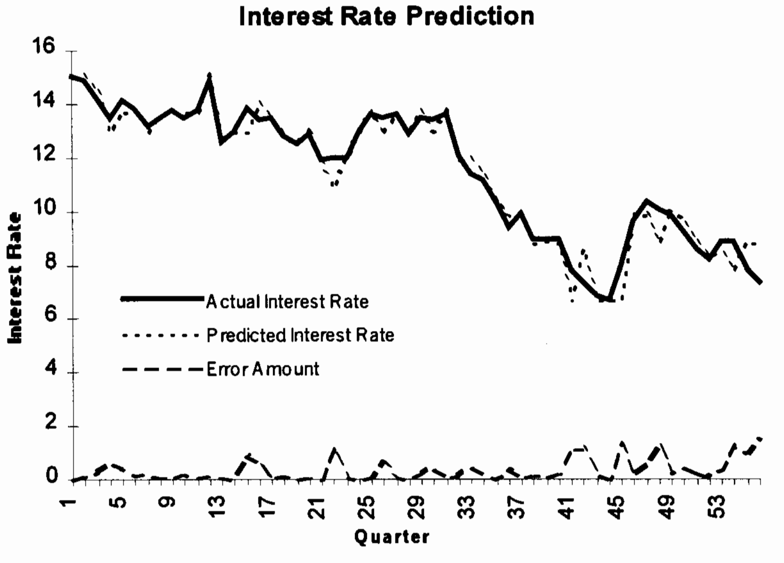

Figure 8.7 Six Month predicted rates Combining Three groups

Training Average	Test Average	Overall Average
0.221	0.629	0.345

Table 8.7 Six Month Average Errors for 3 Group Hierarchy

8.2.8 *Four Group Hierarchy*

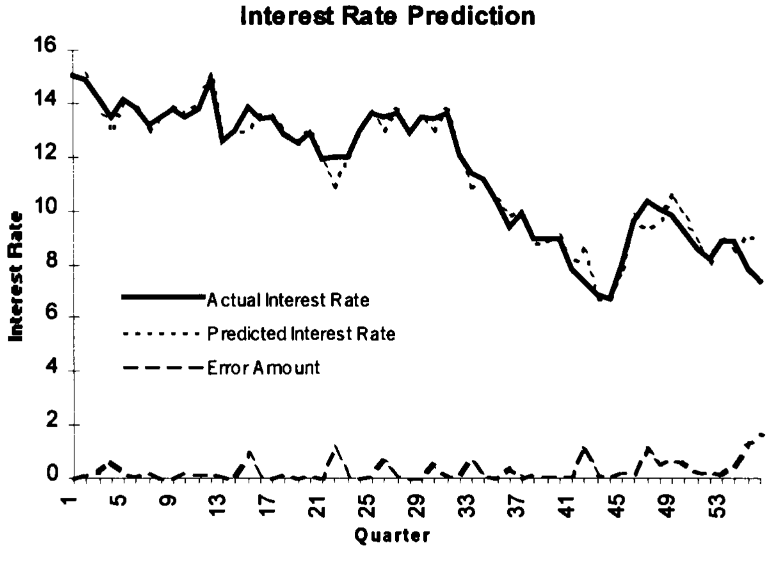

Figure 8.8 Six Month predicted rates Combining Four groups

The four group hierarchy consists of the results from the country, company, employment and savings groups, and the results for the training and test periods is shown in Figure 8.8. The Hierarchical Fuzzy Logic system is able to predict the interest rate during the training period with a good accuracy with only a couple of quarters having an error amount greater than one percent. During the test period the system had some good predictions but the last few quarters of the test period (quarter 54 onwards) showed an increasing error amount in the range of 1.5 percent. This follows a similar pattern as the Three Group hierarchy.

Training Average	Test Average	Overall Average
0.184	0.502	0.280

Table 8.8 Six Month Average Errors for 4 Group Hierarchy

8.2.9 Final Combined Hierarchy

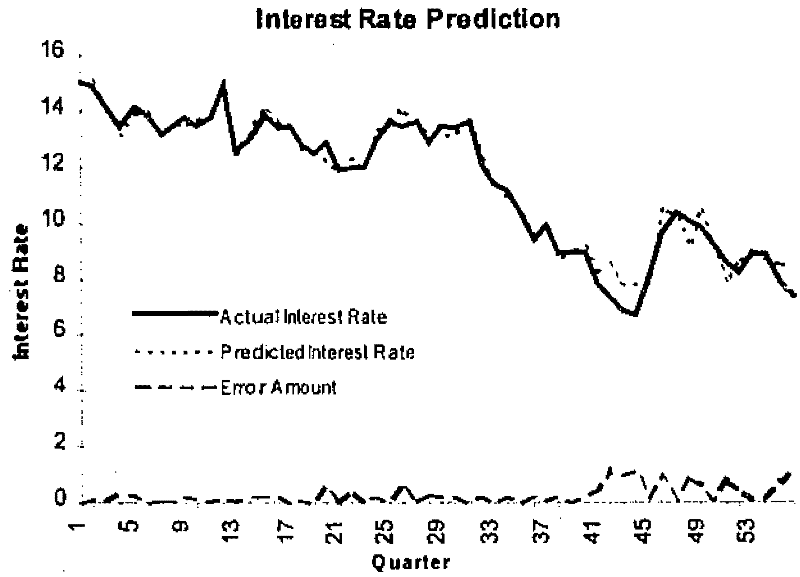

Figure 8.9 Six Month predicted rates for Final Combined Hierarchy

Figure 8.9 above shows the results when combining all the groups into the final hierarchical system. The results in the training period show an great ability to predict the six monthly interest rate with almost a zero error amount for the whole training period. The results during the test period however do not reflect this accuracy with a number of quarters having an error amount greater than 1 percent. The training average error is the one of the lowest so far encountered, where the test average is still quite high.

Training Average	Test Average	Overall Average
0.152	0.565	0.277

Table 8.9 Six Month Average Errors for Final Combined Hierarchy

8.3 12 Month predictions using Hierarchical FKB

Extending the prediction time for interest rates allows financial institutions to make decisions with the knowledge that an interest rate will move to a certain value in the future. So far we have looked at three and six month predictions. In this section the Hierarchical Fuzzy Logic system is trained to predict interest rates one year (twelve months) from the current quarter.

The indicators used to predict one year ahead are the same as used previously (section 4.1), as are the groups they have been split into. The following sections show the results of the fuzzy logic system that the FLGA has produced on the separate groups in predicting the interest rate, followed by the combined hierarchical groups.

8.3.1 Company Group prediction

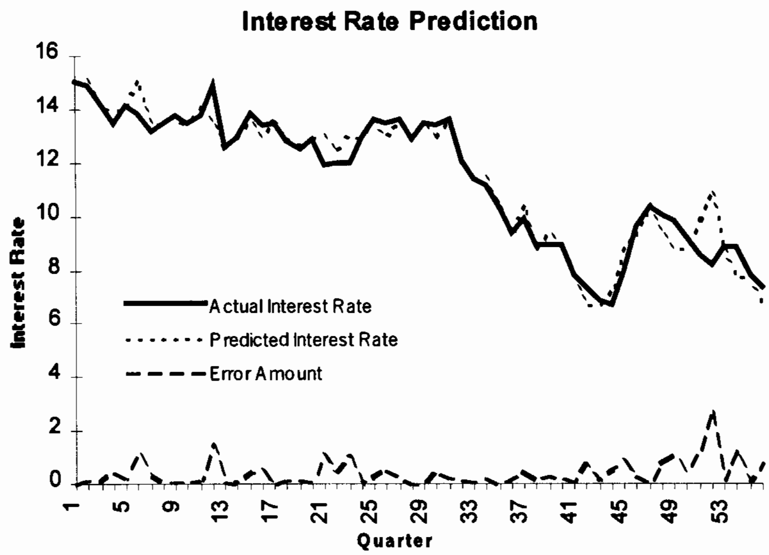

Figure 8.10 One Year predicted rates using Company group

Training Average	Test Average	Overall Average
0.284	0.649	0.395

Table 8.10 One Year Average Errors for Company Group

8.3.2 Country Group prediction

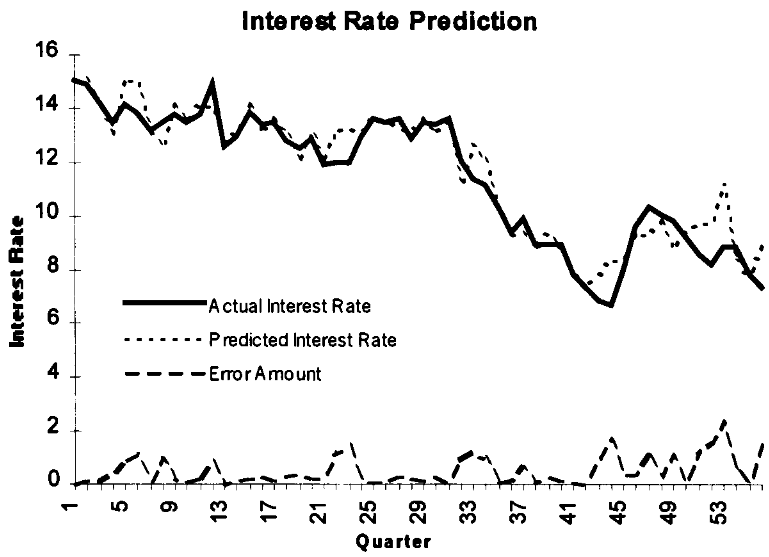

Figure 8.11 One Year predicted rates using Country group

Training Average	Test Average	Overall Average
0.384	0.771	0.501

Table 8.11 One Year Average Errors for Country Group

8.3.3 Foreign Group prediction

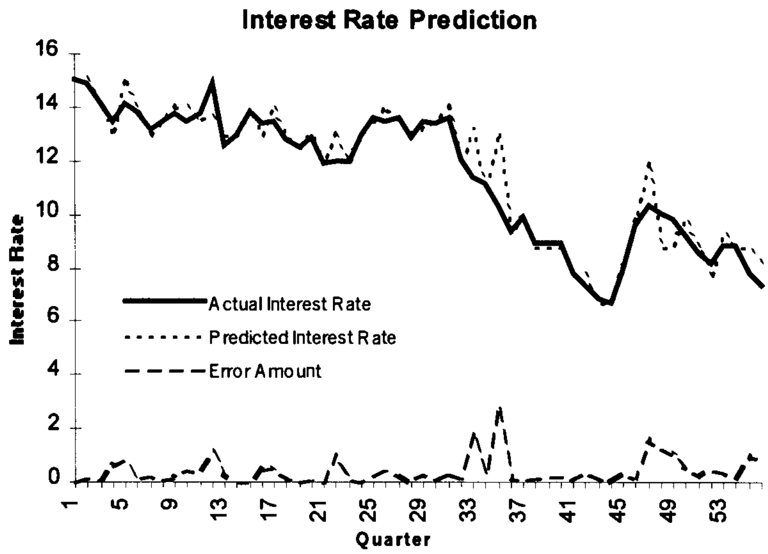

Figure 8.12 One Year predicted rates using Foreign group

Training Average	Test Average	Overall Average
0.357	0.517	0.405

Table 8.12 One Year Average Errors for Foreign Group

8.3.4 Savings Group prediction

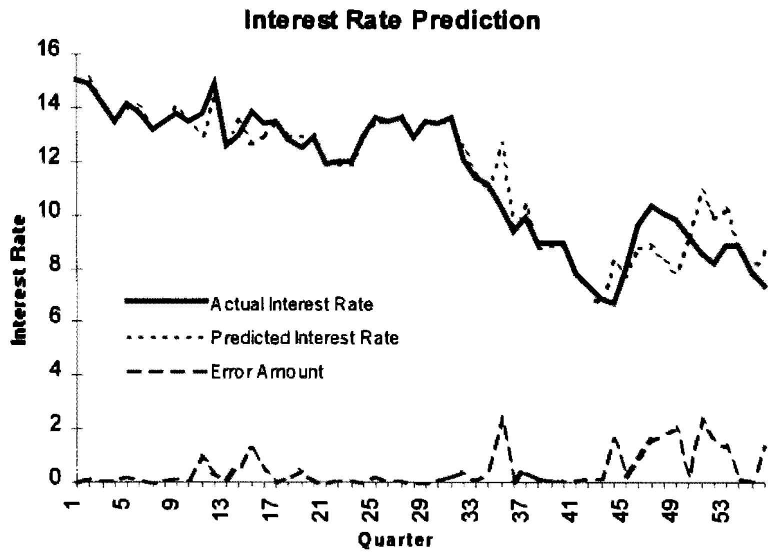

Figure 8.13 One Year predicted rates using Savings group

Training Average	Test Average	Overall Average
0.243	0.901	0.443

Table 8.13 One Year Average Errors for Savings Group

8.3.5 Employment Group prediction

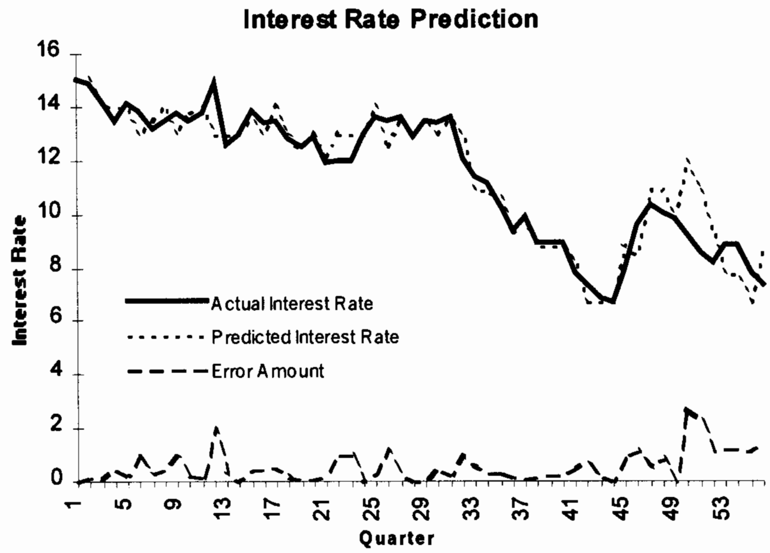

Figure 8.14 One Year predicted rates using Employment group

Training Average	Test Average	Overall Average
0.381	0.919	0.544

Table 8.14 One Year Average Errors for Employment Group

8.3.6 Two Group Hierarchy

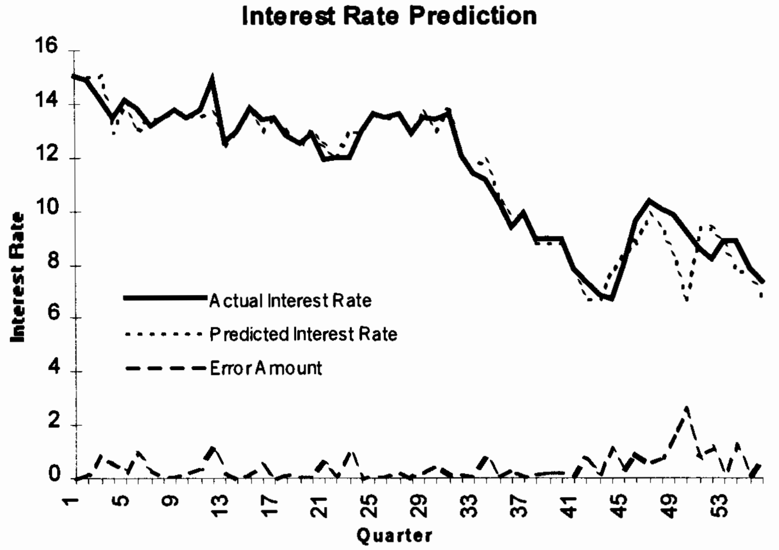

Figure 8.15 One Year predicted rates Combining 2 groups

Training Average	Test Average	Overall Average
0.269	0.741	0.412

Table 8.15 One Year Average Errors for 2 group Hierarchy

8.3.7 Three Group Hierarchy

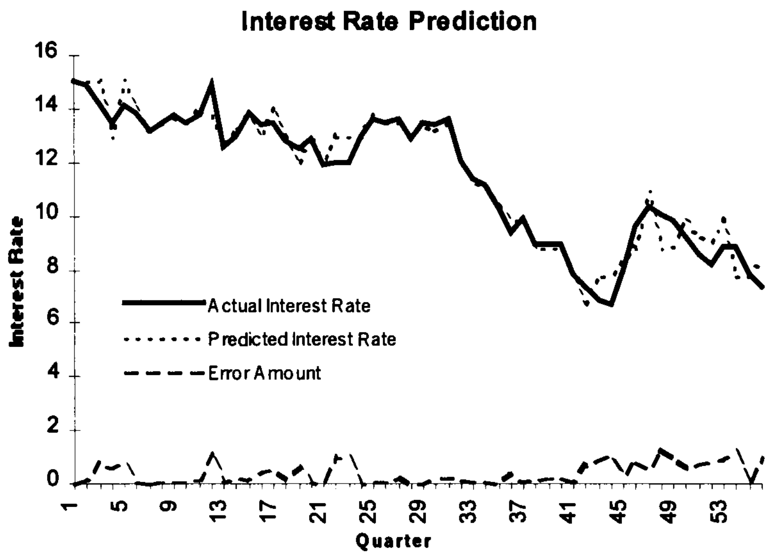

Figure 8.16 One Year predicted rates Combining 3 groups

Training Average	Test Average	Overall Average
0.262	0.701	0.395

Table 8.16 One Year Average Errors for 3 group Hierarchy

8.3.8 Four Group Hierarchy

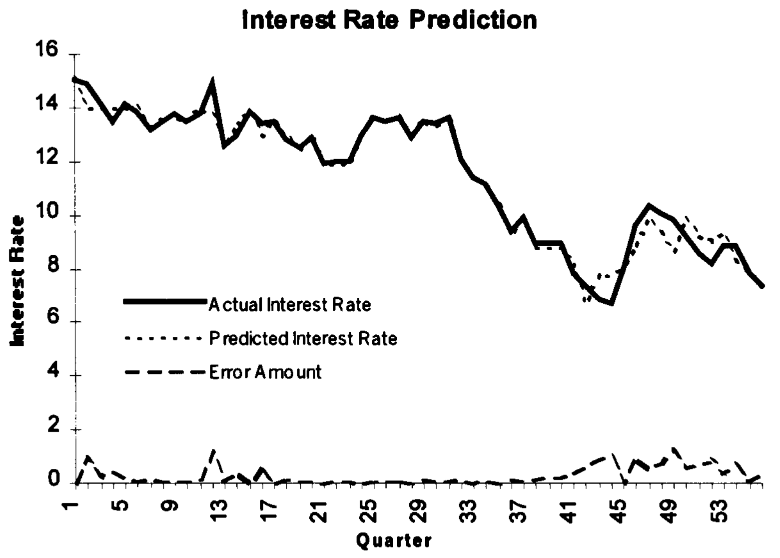

Figure 8.17 One Year predicted rates Combining 4 groups

Figure 8.17 shows the results of the Four Group Hierarchical Fuzzy Logic system. The system is able to predict the interest rate during the training period with a fairly good degree of accuracy with only a two quarters having an error amount greater than one percent. During the test period the system was not able to predict the yearly interest rate with the same degree of accuracy. There were a number of quarters with an error amount greater than one percent. This is reflected in the Average Error amount in Table 8.17 below.

Training Average	Test Average	Overall Average
0.158	0.593	0.291

Table 8.17 One Year Average Errors for 4 Group Hierarchy

8.3.9 Final Combined Hierarchy

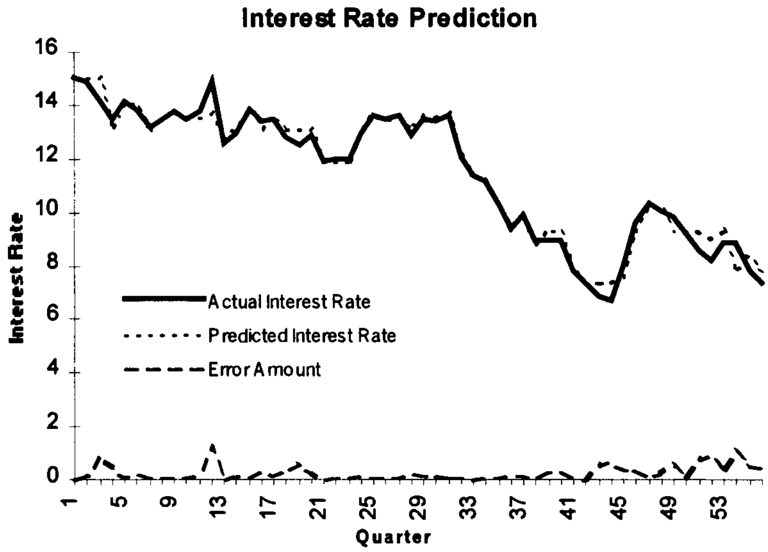

Figure 8.18 One Year predicted rates for Final Combined groups

Training Average	Test Average	Overall Average
0.177	0.424	0.252

Table 8.18 One Year Average Errors for Final Combined Hierarchy

From the above Figures and Tables, we can see that the system was able to predict some of the fluctuations in interest rates one year in the future. The single groups faced the same problems as they did in the shorter time periods in that they had a number of quarters in the test period where there was a large error amount between the actual and predicted interest rate. However, the Four Combined Hierarchy (Figure 8.17) and the Final Combined Hierarchy (Figure 8.18) shows that the system produces similar results for the twelve month predictions when compared to the three month and six month predictions. In fact, the Final Combined Hierarchy produced slightly better results than the Four Combined Hierarchy, where in the 3 and 6 month systems the Four Combined Hierarchy produced better results.

8.4 Long Term Prediction with Neural Networks

A Hierarchical Neural Network system is used with the same indicators (and groupings) as previously looked at. Six month interest rate prediction is looked at first with the results of the Four Combined system and the Final Combined system shown. A neural network system is also created that includes all the indicators in the one system and its results are shown. We then show the same results for the prediction of interest rates one year in the future.

8.4.1 6 Month Predictions with Neural Networks

A neural network was used to predict the interest rate 6 months from the current quarter. The economic indicators were split into the same groups as used previously. The results when combining four groups, all five groups and finally all the indicators in the single neural network are shown below.

8.4.1.1 Four Group Hierarchy

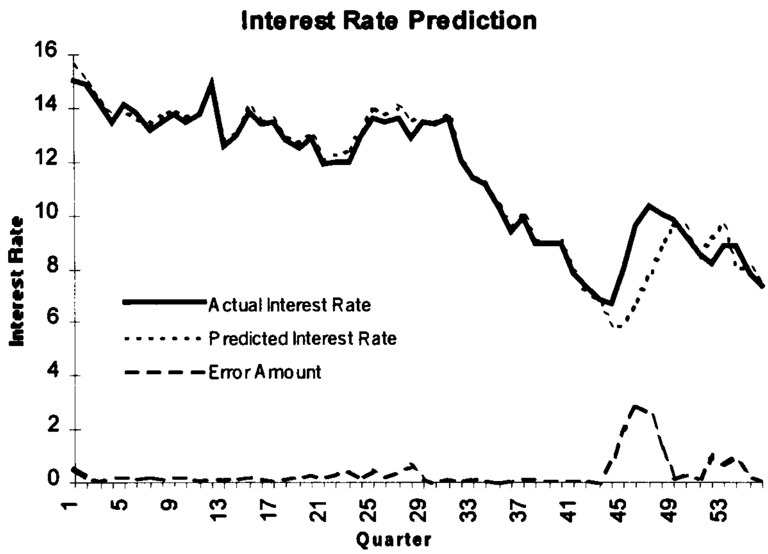

Figure 8.19 Neural Network Six month Prediction using Four Groups

Training Average	Test Average	Overall Average
0.196	0.794	0.378

Table 8.19 Six Month Average Errors for Neural Network Four group Hierarchy

8.4.1.2 Final Combined Hierarchy

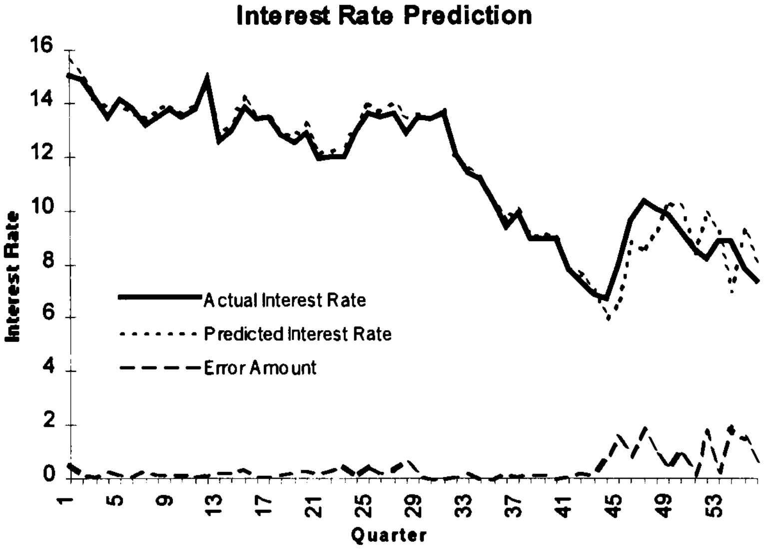

Figure 8.20 Neural Network Six month Prediction using Final Combined Hierarchy

Training Average	Test Average	Overall Average
0.197	0.813	0.383

Table 8.20 Six Month Average Errors for Neural Network Final Combined Hierarchy

8.4.1.3 All Indicators System

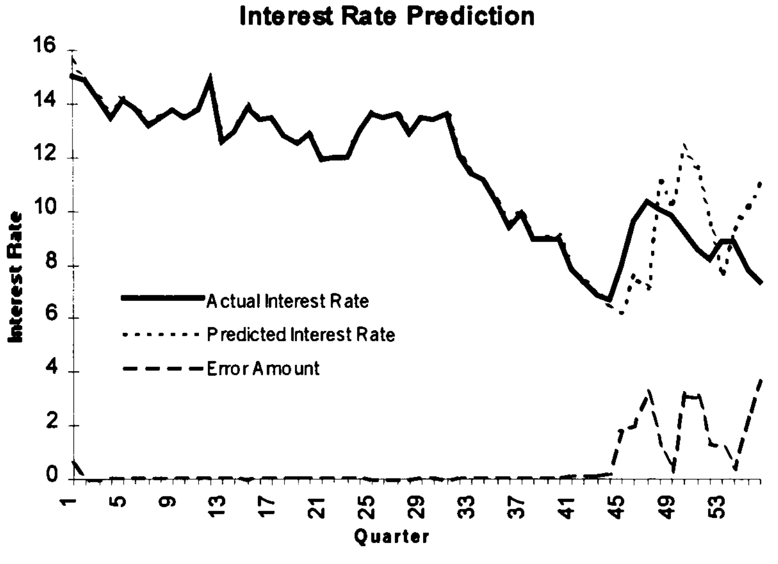

Figure 8.21 Neural Network Six month Prediction using All Indicators

Training Average	Test Average	Overall Average
0.079	1.421	0.480

Table 8.21 Six Month Average Errors for Neural Network using All Indicators

The prediction accuracy of the neural network for predicting Six month interest rates show a similar pattern to the neural network results for predicting interest rates one quarter on the future. The Four Group Hierarchical Neural Network shows that it is able to predict the fluctuations of the interest rate for some quarter's, but seems to be slightly behind in predictions for other quarters. This is also the case for the Final Combined Hierarchical Neural Network where its six month predictions show it does follow the fluctuations of the interest rate, but seems to be one to two quarters behind the actual change. The All Indicators System Neural Network has the best training average error of all the Six month neural networks, but its test average error is extremely disappointing. The neural network does not seem to be able to predict the interest rate with any degree of accuracy, leading to a test average error of 1.421.

8.4.2 12 Month Predictions with Neural Networks

A neural network was used to predict the interest rate one year from the current quarter. The economic indicators were split into the same groups as used previously. The results when combining four groups, all five groups and finally all the indicators in the single neural network are shown below.

8.4.2.1 Four Group Hierarchy

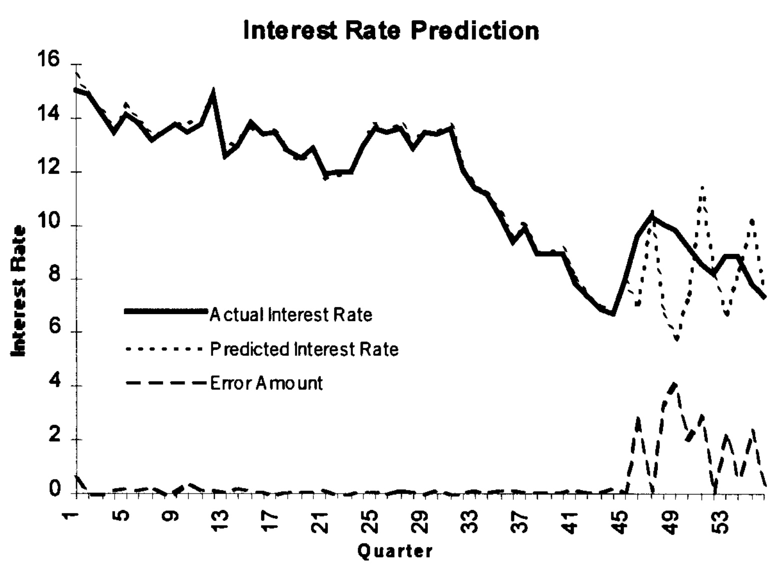

Figure 8.22 Neural Network One Year Prediction using Four Groups

Training Average	Test Average	Overall Average
0.110	1.248	0.456

Table 8.22 One Year Average Errors for Neural Network Four group Hierarchy

8.4.2.2 Final Combined Hierarchy

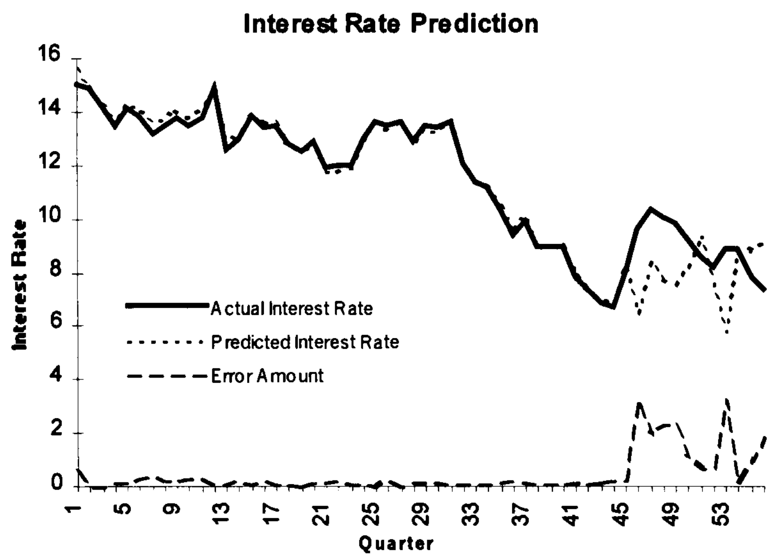

Figure 8.23 Neural Network One Year Prediction using Final Combined Hierarchy

Training Average	Test Average	Overall Average
0.139	1.114	0.435

Table 8.23 One Year Average Errors for Neural Network Final Combined Hierarchy

8.4.2.3 All Indicators System

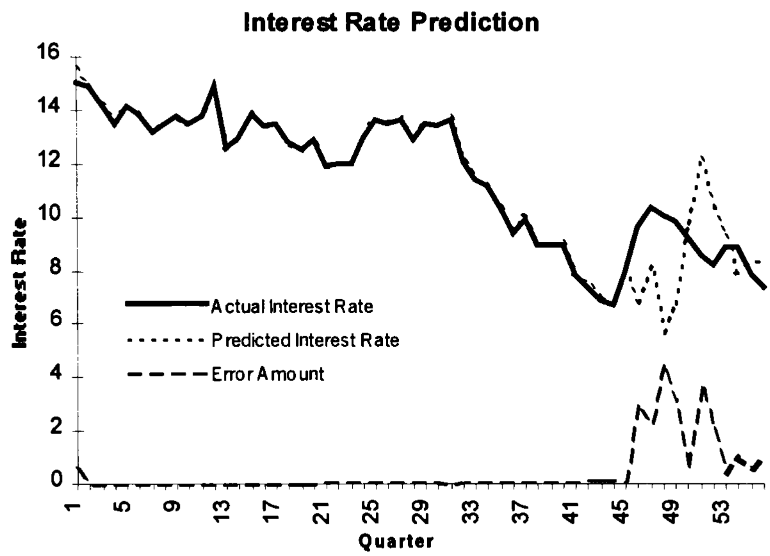

Figure 8.24 Neural Network One Year Prediction using All Indicators

Training Average	Test Average	Overall Average
0.054	1.320	0.439

Table 8.24 One Year Average Errors for Neural Network using All Indicators

8.5 Comparison between Hierarchical Fuzzy Logic systems and Neural Networks for Long Term Predictions

From the above Figures and Tables, it can be seen that the results for long term predictions produce similar results to the predictions for the following quarter. Table 8.25 below shows a comparison between the Hierarchical Fuzzy Logic system predictions and the neural network predictions for six months and one year.

6 Month Predictions	Training Error	Test Error	Overall Error
Four Group HKB	0.184	0.502	0.280
Five Group HFKB	0.152	0.565	0.277
Four Group HNN	0.196	0.794	0.378
Five Group HNN	0.197	0.813	0.383
All Indicators NN	0.079	1.421	0.480
One Year Predictions			
Four Group HKB	0.158	0.593	0.291
Five Group HFKB	0.177	0.424	0.252
Four Group HNN	0.110	1.248	0.456
Five Group HNN	0.139	1.114	0.435
All Indicators NN	0.054	1.320	0.439

Table 8.25 Comparison of Results for Long Term Predictions

From these results, it can be seen that the Hierarchical Fuzzy Logic systems have a much better Test Average Error when compared to the neural network systems. The Training average error results were similar for most of the prediction systems, with only the All Indicator Neural Network system, for both six month and one year predictions, which had very low training results. However, the test error amounts for the All Indicator systems were the highest of all the systems.

From these results, we can conclude that using 14 economic indicators and training the system for 40 quarters, the Hierarchical Fuzzy Logic systems provide much better prediction results than the neural network systems. These results are similar to the comparisons found in the previous chapter (section 7.5) where the Interest Rate predictions for one quarter by the Hierarchical Fuzzy Logic system provided better prediction results than the neural network system.

Chapter 9 Conclusion and Further Investigations

This thesis has presented a method in which a Hybrid Fuzzy Logic and Genetic Algorithm system can be used to model and predict the fluctuations of the 10-year Australian treasury bond using Australian economic data.

The research showed that, using a Hierarchical Fuzzy Logic system, the number of fuzzy rules in the Fuzzy Knowledge Base could be reduced to a linear equation as opposed to an exponential equation. This meant the number of fuzzy rules of the system is reduced significantly, hence computational times are decreased resulting in a more efficient system.

The application of the proposed method to modelling and prediction of interest rate using Australian economic indicators is considered. Genetic Algorithms are used to obtain the fuzzy rules for each FL system as well as the mapping between different FL systems. Untrained rules (those rules never used during the training period) were set to a default value depending upon their position within the FKB.

From simulation results it was found that the Hierarchical Fuzzy Logic system is capable of making accurate predictions of the following quarter's interest rate. These results were compared to a Feed Forward Fuzzy Logic system which produced slightly better results for a similar number of rules. One advantage of the Hierarchical Fuzzy Logic systems over the Feed Forward Fuzzy Logic systems is that they can operate in parallel while the Feed Forward Fuzzy Logic system must operate in a sequential manner. This results in a speed increase during the training period.

The results from the Hierarchical Fuzzy Logic systems and the Feed Forward Fuzzy Logic system were compared to a Hierarchical Neural Network, which split the economic indicators into the same groupings, and also a single Neural Network that used all the indicators as inputs. The results from the neural network systems were disappointing when compared to the Hierarchical Fuzzy Logic systems as they produced far larger error amounts when tested on different (non-training) data.

Long term predictions for six months and one year from the current quarter were then undertaken, with the Hierarchical Fuzzy Logic systems proving to be more accurate in their predictions than the Neural Network systems. These results were found to be similar to those obtained when quarterly interest rates were predicted in chapter 7.

An advantage that the Fuzzy Logic Genetic Algorithm systems have over the Neural Network systems is that, in many cases, the reasoning behind a prediction must be explained (either to the customer or senior management). As the Hierarchical Fuzzy Logic systems used a FKB which contains all the rules of the system, this allows the rules used in the prediction to be shown and allows an expert to make any modifications if necessary. This can be a very difficult task with a neural network system.

Further research in this area may look at training the prediction systems up to the last available data. Currently, the system is trained for 40 quarters and then tested with the remaining quarters. An alternative to this approach would be to train the Hierarchical Fuzzy Logic system with all the available data. This would allow indicators that become more important over time to gradually become more dominant within the FKB's, while indicators that have less relevance slowly become less influential.

Having a time lag for some economic indicators may also increase prediction accuracy. There are some indicators whose effect is not felt on the interest rate for a number of quarters, such as Consumer Price Index (Larrain, M. 1991). Delaying the indicator results in the system using the indicator when it has more effect on the interest rate. The accuracy may also be increased if an indicator that fluctuates greatly between quarters is smoothed out using some form of moving average (such as two quarter (six month) moving average). This would then remove any sudden peaks (or valleys) that the indicator may exhibit which could greatly effect the prediction accuracy.

Finally, some combination of Hierarchical and Feed Forward Fuzzy Logic systems may provide better results by adding more indicators without the exponential growth in the fuzzy rule base.

Chapter 10 References.

Azzof, E. M. (1994). *Neural Networks Time Series Forecasting of Financial Markets*, New York: John Wiley & Sons

Bauer, R. J. (1994). *Genetic Algorithms and Investment Strategies*. John Wiley & Sons, NewYork.

Chester, M. (1993). *Neural Networks: A Tutorial.* Prentice-Hall, Englewood Cliffs, New Jersey.

Cox, E. (1994). *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems*. Academic Press, New York.

Cox, E. (1993). "Adaptive fuzzy systems". IEEE Spectrum, February, pg 27-31.

Croughanowr, D. R. and Koppel, L. B. (1965). *Process Systems Analysis and Control*. New York: McGraw-Hill.

Davis, L. (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Gallant, S. I. (1993). *Neural Network Learning and Expert Systems*, MIT Press, Cambridge, MA.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning.* Reading, Massachusetts: Addison Wesley.

Grossberg, S. (1988). "Nonlinear Neural Networks: Principles, mechanisms, and achitectures", Neural Networks. Ed 1 pg 17-68.

Hebb, D. (1949). *The Organisation of Behaviour.* John Wiley & Sons, New York.

Holland, H.(1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press.

Hung, C. (1993). "Building a neuro-fuzzy learning control system", AI Expert, vol 8, no. 11, pg 40-49.

Karr, C. (1994). Adaptive Control with Fuzzy Logic and Genetic Algorithms, Fuzzy Sets and *Neural Networks, and Soft Computing*, Edited by R. R. Yager, L. A. Zadeh, Van Nostrand ReinHold, NY, USA.

Karr, C. (1991). "Genetic algorithms for fuzzy controllers", AI Expert, vol6, no 2, pg 26-33.

Kingham, M. and Mohammadian, M. (1996). "Financial Modelling and Prediction of Interest Rate using Fuzzy Logic and Genetic Algorithms", 4th Australian and New Zealand Intelligent Information Systems Conference (ANZIIS '96), November 1996.

Kingham, M. and Mohammadian, M. (1996). "Hierarchical and Feed Forward Fuzzy Logic for Financial Modelling and Prediction", Artificial Intelligence 1997 (AI'97), Perth, Australia, December 1-4.

Kong, S and Kosko, B. (1990). "Comparisons of Fuzzy and Neural Truck Backer-Upper Control Systems", Proc of International Joint Conference on Neural Networks, June. Pg 339-361.

Kosko, B. (1992). *Neural networks and fuzzy systems, a dynamic system.* Prentice-Hall: Englewood Cliff.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*. EngleWood Cliffs, N.J.: Prentice Hall.

Kosko, B. & Isaka, S. (1993). "Fuzzy Logic", *Scientific American.* July, pg 76-81.

Lapedes, A and Farber, R., Nonlinear Signal Processing Using Neural Networks, Prediction and system Modelling. Los Alamos Report LA-Ur-87-2662, Los Alamos National Laboratory, Los Alamos, NM, 1987.

Larrain, M. (1991). "Testing Chaos and Non Linearities in T-Bill Rates", *Financial Analysts Journal*, Sep-Oct, pg 51-62.

Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Controllers - part I, part II. *IEEE Transactions on Systems, Man and Cybernetics.*, Vol 2092, pg 404-435.

Lee, M. A. and Takagi, H. (1993). "Dynamic control of Genetic Algorithm using Fuzzy Logic techniques", *Proceedings of the 5th International Conference on Genetic Algorithms*, Urbana-Champaign, IL, July 17-22, pg 76-83.

Madden, R (1995). *Measuring Australian Economy*, Australian Bureau of Statistics, Catalogue No 1360.0

Mamdani, E. H. (1993). "Twenty years of fuzzy control: experiences gained and lessons learnt", *Proceedings of 2nd IEEE International Conference on Fuzzy Systems*, San Diego, USA, pg 339-344.

McClelland, J. L. and Rumelhart, D. E. (1988). *Explorations in Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts.

McCulloch, W. and Pitts, W. (1943). "A Logical calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, Vol 9, pg 127-147.

McLennan, W. (1997). *Australian Economic Indicators, February 1997*, Australian Bureau of Statistics, Catalogue No 1350.0.

Medsker, L. (1995). *Hybrid Intelligent Systems*, Kluwer Academic Publishers, Norwell, Massachusetts.

Michalewicz, Z. (1992). "*Genetic Algorithms + Data Structure = Evolution Programs*", Spring Verlag, U.S.A.

Minsky, M. and Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Mohammadian, M., Yu, X. H. and Smith, J. D. (1993). "A self-learning control system- docking a truck", *29th Australian Applied Mathematics Conference*, Adelaide, Australia, February 9-11.

Mohammadian, M. and Stonier, R. J. (1994). "Tuning and optimisation of Membership functions of Fuzzy Logic Controllers by Genetic Algorithms", *3rd International Workshop on Robot Human Communication RO-MAN'94*, Japan, Nagoya, 18-20 July.

Mohammadian, M., Kingham, M. and Bignall, B. (1998). "Hierarchaical and Feed-Forward Fuzzy Logic Systems for Interest Rate Prediction", *Journal of Computation Intelligence in Finance*, Vol 6, No 3, pg 5-12.

Nguyen H. D. and Widrow, B, (1990). "Neural network for self-learning control systems", IEEE Control Systems Magazine, April.

Pearce, D. (1983). *The Dictionary of Modern Economics*, MacMillan Press, London.

Raju, G. V. S. and Zhou, J. (1993). Adaptive Hierarchical Fuzzy Controller, *IEEE Transactions on Systems, Man & Cybernetics*, Vol 23, No 4, pg 973-980.

Rao, V.B. and Rao, H.V. (1994). *C++ Neural Networks and Fuzzy Logic,* MIS Press.

Refenes, A. (1995). *Neural Networks in the Capital Markets.* Wiley, New York.

Ross, T. J. (1995). *Fuzzy Logic with Engineering Applications.* Prentice Hall, Englewood Cliffs, New Jersey.

Ruelle, D. (1989). *Chaotic Evolution and Strtange Attractors: The Statistical Analysis of Time Series for Deterministic Nonlinear Systems.* Cambridge Uni Press:

Rumelhart, D.E., and McClelland, J.L. (1986). *Parallel Distributed Processing: Exploration in the Microstructure of Cognition.* MIT Press.

Schaffer, J. D. (1994). “Combinations of Genetic Algorithms with Neural Networks of Fuzzy Systems”, in (Zurada, Marks, and Robinson, Editors), pg 371-382.

Takagi, T. and Sugeno, M. (1983). “Derivation of fuzzy control rules from human operator’s control actions”, Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pg 55-60.

Terano, T., Asai, K. and Sugeno, M. (1994). *Applied Fuzzy Systems.* AP Professional, Cambridge, Massachusetts.

Vidyasagar, M. (1978). *Nonlinear systems and Analysis.* Prentice Hall, Englewood Cliffs, New Jersy.

Wang, X. and Mendel, J, M, (1992). "Generating fuzzy rules by learning from example", IEE Trans. Sys., Man and Cybernetics, vol. 22, pp, 1414-1427.

Welstead, T. (1994). *Neural networks and Fuzzy Logic applications in C/C++.* Wiley, New York.

Yager, R. R. (1983). "An introduction to applications of possibility theory", Human Systems Management, vol 3, pg 246-269.

Zadeh, L. (1965). Fuzzy Sets. *Inf. Control*, vol 8, pg 338-353.

Zadeh, L. (1973). "Outline of a new approach to the analysis of complex systems and decision processes". IEEE Trans. Syst. Man. Cybern. Vol SMC-3, pg 28-44.

Zadeh, L. (1994). "Fuzzy logic, Neural Networks, and soft computing", *Communications of the ACM*, vol 37, no. 3, pg 77-84.

Zurada, L., Marks, R., and Robinson, C. (eds). (1994) Computational Intelligence Imitating Life. IEEE Press, New York.

Chapter 11 Appendix A

Time Series Data.

The following table shows the raw data from the Australian Bureau of Statistics (REF) as of June, 1997. The data is shown as quarterly data starting from the first quarter of 1983.

Year/ Quarter	Interest Rate	All Industrial	Company Profit	Consumer Price Index	Current Account
1983 - 1	14.00	631.7	1830.0	58.6	-1793.0
1983 - 2	12.95	610.7	1850.0	60.3	-1690.0
1983 - 3	15.00	650.1	1890.0	61.6	-1498.0
1983 - 4	14.85	744.9	1900.0	62.9	-1834.0
1984 - 1	14.15	875.1	1993.0	64.0	-1800.0
1984 - 2	13.50	974.0	2168.0	65.5	-1439.0
1984 - 3	14.15	960.0	2639.0	65.2	-2004.0
1984 - 4	13.85	915.5	2668.0	65.4	-2100.0
1985 - 1	13.15	1032.8	2309.0	66.2	-2497.0
1985 - 2	13.50	1072.5	2312.0	67.2	-2794.0
1985 - 3	13.80	1172.8	2696.0	68.1	-2837.0
1985 - 4	13.50	1228.9	3186.0	69.7	-2783.0
1986 - 1	13.80	1401.5	3327.0	71.3	-3026.0
1986 - 2	14.85	1459.9	2675.0	72.7	-3640.0
1986 - 3	12.60	1739.2	2526.0	74.4	-3710.0
1986 - 4	12.95	1936.2	1960.0	75.6	-3698.0
1987 - 1	13.85	1936.4	2522.0	77.6	-3182.0
1987 - 2	13.40	2294.1	2774.0	79.8	-3143.0
1987 - 3	13.45	2584.6	2931.0	81.4	-2741.0
1987 - 4	12.80	2599.3	3041.0	82.6	-2618.0
1988 - 1	12.50	3302.7	3256.0	84.0	-2342.0
1988 - 2	12.85	1919.8	3565.0	85.5	-2629.0
1988 - 3	11.90	2210.2	3865.0	87.0	-2333.0
1988 - 4	11.95	2506.0	4022.0	88.5	-2989.0
1989 - 1	11.95	2554.4	4050.0	90.2	-3032.0
1989 - 2	12.95	2447.5	4177.0	92.2	-4279.0
1989 - 3	13.65	2458.1	4350.0	92.9	-4829.0
1989 - 4	13.50	2498.3	4713.0	95.2	-5349.0
1990 - 1	13.65	2756.3	4551.0	97.4	-5720.0
1990 - 2	12.90	2575.4	4278.0	99.2	-5792.0
1990 - 3	13.45	2412.8	4006.0	100.9	-5947.0

1990 - 4	13.40	2367.9	3811.0	102.5	-3902.0
1991 - 1	13.65	2167.5	3514.0	103.3	-4071.0
1991 - 2	12.05	1979.4	3093.0	106.0	-4721.0
1991 - 3	11.40	2202.8	2709.0	105.8	-3721.0
1991 - 4	11.15	2330.7	2567.0	106.0	-2994.0
1992 - 1	10.30	2402.4	2700.0	106.6	-2817.0
1992 - 2	9.40	2511.5	2979.0	107.6	-2736.0
1992 - 3	9.90	2449.5	3169.0	107.6	-2838.0
1992 - 4	8.90	2550.0	3394.0	107.3	-3297.0
1993 - 1	8.95	2334.0	3694.0	107.4	-3987.0
1993 - 2	8.95	2373.4	4047.0	107.9	-3777.0
1993 - 3	7.80	2598.6	4397.0	108.9	-3412.0
1993 - 4	7.37	2665.7	4630.0	109.3	-3845.0
1994 - 1	6.85	3037.8	4854.0	109.8	-3801.0
1994 - 2	6.68	3191.6	5312.0	110.0	-3724.0
1994 - 3	7.95	3275.9	5729.0	110.4	-3631.0
1994 - 4	9.63	2984.7	6214.0	111.2	-5443.0
1995 - 1	10.33	2926.1	6475.0	111.9	-7023.0
1995 - 2	10.04	2741.0	6435.0	112.8	-6875.0
1995 - 3	9.83	2850.0	6180.0	114.7	-7535.0
1995 - 4	9.21	3012.1	6103.0	116.2	-6608.0
1996 - 1	8.57	3169.3	6293.0	117.6	-5567.0
1996 - 2	8.18	3276.0	6152.0	118.5	-6342.0
1996 - 3	8.88	3365.1	5823.0	119.0	-4824.0
1996 - 4	8.88	3305.8	5478.0	119.8	-4450.0
1997 - 1	7.79	3459.2	5183.0	120.1	-4744.0
1997 - 2	7.37	3660.8			
1997 - 3					
1997 - 4					

Year/ Quarter	GDP (A)	Home Loans	Job Vacancies	New Motor Vehicles	RBA Commodity
1983 - 1	69714.0	18100	46.7	39100	68.79
1983 - 2	68691.0	18600	28.1	46400	68.43
1983 - 3	68473.0	25100	23.1	39500	75.21
1983 - 4	68291.0	21900	24.0	35900	76.85
1984 - 1	70755.0	22700	24.6	37100	75.71
1984 - 2	71814.0	25200	47.3	38400	72.61
1984 - 3	73467.0	30300	60.3	40700	70.91
1984 - 4	74114.0	26700	54.4	40900	72.55
1985 - 1	74814.0	25400	62.5	40700	73.41
1985 - 2	75431.0	23500	68.2	41300	73.10
1985 - 3	76904.0	27600	81.6	46800	84.58
1985 - 4	78455.0	23900	93.0	41900	86.35
1986 - 1	79856.0	24700	91.3	44800	83.36
1986 - 2	79520.0	21300	92.7	41400	84.59
1986 - 3	79619.0	18900	89.0	35400	79.91
1986 - 4	79319.0	20500	92.0	33900	80.18
1987 - 1	80201.0	23200	80.3	33100	85.12
1987 - 2	80799.0	23424	85.4	30500	83.83
1987 - 3	81068.0	22919	91.0	29700	88.31
1987 - 4	82541.0	24699	87.4	30600	86.06
1988 - 1	83822.0	28142	90.4	33979	89.56
1988 - 2	85357.0	29841	88.2	36986	96.76
1988 - 3	86079.0	34452	97.5	35060	105.08
1988 - 4	86150.0	39774	102.3	36796	103.71
1989 - 1	87287.0	32184	116.5	40515	98.14
1989 - 2	88887.0	32228	129.8	39406	94.15
1989 - 3	89391.0	26910	125.2	41460	97.07
1989 - 4	91007.0	22934	148.0	41799	101.35
1990 - 1	91775.0	22608	128.7	42996	99.60
1990 - 2	91736.0	22128	133.0	37271	99.61
1990 - 3	92831.0	26451	108.0	49754	101.25
1990 - 4	92768.0	23823	95.6	40858	95.22
1991 - 1	92109.0	24313	76.6	40401	95.89
1991 - 2	91828.0	22722	51.2	35070	95.71
1991 - 3	91483.0	23746	38.3	33584	89.81
1991 - 4	90834.0	28689	32.6	34203	91.60
1992 - 1	91426.0	29683	29.9	34304	85.80
1992 - 2	91647.0	30838	29.1	35268	88.37
1992 - 3	92751.0	33304	29.9	43002	89.70
1992 - 4	92772.0	35547	27.9	43930	88.71
1993 - 1	93729.0	35708	28.8	40034	92.18
1993 - 2	94749.0	36279	31.3	41021	92.17
1993 - 3	95412.0	39384	30.7	42975	89.72

1993 - 4	96320.0	41048	35.1	42570	93.29
1994 - 1	96863.0	43340	38.2	35681	94.25
1994 - 2	98146.0	47593	40.8	40746	93.53
1994 - 3	100271.0	48730	49.7	43715	92.65
1994 - 4	101195.0	45676	62.7	45241	90.62
1995 - 1	102200.0	40576	76.3	42373	93.65
1995 - 2	103285.0	39756	82.6	43948	92.48
1995 - 3	103382.0	33867	70.8	47264	97.14
1995 - 4	104105.0	36044	77.5	47511	101.73
1996 - 1	105845.0	37058	75.6	41305	95.58
1996 - 2	106719.0	37717	72.7	42944	97.28
1996 - 3	108924.0	38665	81.8	45564	94.80
1996 - 4	109004.0	35727	76.6	52082	93.40
1997 - 1	109852.0	41699	72.1	42893	91.60
1997 - 2			81.0		89.70
1997 - 3					
1997 - 4					

Year/ Quarter	Savings Ratio	Trade Weight Ind	Unemploy ment	Avg Week Earnings
1983 - 1	8.65	83.8	7.5	329.90
1983 - 2	8.26	83.4	9.4	337.60
1983 - 3	7.30	76.1	10.0	341.00
1983 - 4	6.02	77.7	10.3	343.30
1984 - 1	10.16	80.4	10.4	349.70
1984 - 2	9.10	81.1	9.4	362.00
1984 - 3	9.36	82.9	9.3	370.60
1984 - 4	9.47	79.2	9.3	383.80
1985 - 1	9.92	80.3	8.8	386.20
1985 - 2	8.88	81.3	8.5	389.00
1985 - 3	8.27	69.2	8.3	393.00
1985 - 4	7.71	65.0	7.9	397.20
1986 - 1	7.79	64.8	7.9	403.10
1986 - 2	7.00	60.7	7.7	413.90
1986 - 3	8.37	61.1	7.8	422.70
1986 - 4	6.89	56.3	7.6	425.50
1987 - 1	6.75	51.9	8.3	437.20
1987 - 2	7.17	55.0	8.4	446.30
1987 - 3	4.87	55.4	8.4	444.50
1987 - 4	6.04	56.6	8.0	450.90
1988 - 1	7.71	56.2	7.8	457.00
1988 - 2	3.82	52.0	7.7	470.00
1988 - 3	5.39	53.8	7.6	474.90
1988 - 4	6.52	59.8	7.4	481.70
1989 - 1	6.10	60.0	7.0	486.20
1989 - 2	6.87	63.2	6.8	505.20
1989 - 3	6.12	62.2	6.3	511.60
1989 - 4	6.75	59.4	6.1	519.10
1990 - 1	6.55	59.8	6.1	527.10
1990 - 2	7.36	61.1	5.9	540.00
1990 - 3	8.17	59.8	6.2	546.30
1990 - 4	6.93	61.6	6.7	555.80
1991 - 1	7.19	61.6	7.4	562.70
1991 - 2	6.29	57.3	8.1	578.20
1991 - 3	6.83	59.7	9.2	585.60
1991 - 4	2.87	59.7	9.3	569.90
1992 - 1	5.85	60.6	10.1	575.40
1992 - 2	5.32	55.9	10.4	589.70
1992 - 3	5.54	58.6	10.5	598.90
1992 - 4	3.10	55.2	11.0	597.40
1993 - 1	4.72	51.7	10.8	597.70
1993 - 2	3.93	52.4	11.2	599.50
1993 - 3	3.82	52.9	10.9	611.20

1993 - 4	3.97	49.5	11.0	612.50
1994 - 1	3.18	47.3	10.8	618.10
1994 - 2	2.54	50.8	10.6	619.00
1994 - 3	3.32	52.1	10.3	625.60
1994 - 4	4.67	53.0	9.9	625.10
1995 - 1	2.24	53.4	9.4	634.50
1995 - 2	3.17	56.2	8.9	643.10
1995 - 3	2.60	50.7	8.7	650.10
1995 - 4	1.50	48.4	8.3	652.70
1996 - 1	3.06	53.8	8.5	654.80
1996 - 2	2.40	53.9	8.3	662.70
1996 - 3	2.28	56.8	8.5	668.10
1996 - 4	2.99	58.1	8.3	671.50
1997 - 1	2.87	58.5	8.7	673.50
1997 - 2		59.4	8.6	
1997 - 3				
1997 - 4				