The Application Of Artificial Neural Networks And Genetic Algorithms To The Estimation Of Electrode Response Characteristics And Stability Constants

Volume 1

By Margaret Kathleen Hartnett, School of Chemical Sciences, Dublin City University

Supervised by Dr. Dermot Diamond, School of Chemical Sciences, Dublin City University

A thesis submitted for the degree of Doctor of Philosophy

August 1994

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of doctor of philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Margaret ID No.: 90700368

Date: 30-9-94

Table Of Contents

CHAPTER 1 INTRODUCTION

CHAPTER 3 : POTENTIOMETRIC NON-LINEAR MULTIVARIATE CALIBRATION WITH GENETIC ALGORITHM AND SIMPLEX OPTIMISATION

Volume 2 APPENDICES

FIGURES

Appendix 7 TABLES

APPENDIX 9 SOFTWARE

* The use of the program EQUIL whose source code is included in programs GASTAB.C, GASIMP.C and STDEV.C is gratefully acknowledged. For further information of this program contact M. Bos at the Laboratory For Chemical Analysis in the Chemical Technology department of the University of Twente.

Chapter 1: Introduction

Abstract

This introductory chapter establishes the theoretical and contextual background for the application of neural networks and genetic algorithms to solving chemical problems. This chapter is divided into three major sections, namely neural networks, genetic algorithms and a literature review of previous applications of these techniques. Each of these sections are further subdivided into subsections. In the case of the neural networks section, the order of the subsections reflects a logical progression from small to large scale properties of biological neural systems. This progression is again expressed in the descriptions of artificial neural networks (ANNs). A number of different ANN architectures which have found chemical applications or have been discussed in a cognitive context are described, with particular emphasis on the backpropagation training algorithm for feedforward networks.

The genetic algorithms section mainly describes the formal framework underlying the use of the simple genetic algorithm (SGA) and Holland's Schema Theorem. The applications section is divided into those applications which involved neural networks and those which involved genetic algorithms.

1.1 Biological Basis For ANNs

1.1.1 Neural Structure [1-5]

Neurons are a class of cells which are highly specialised for impulse conduction in the nervous system. While they differ dramatically in morphology depending on their functionality, the motor neuron provides a good general model for neural structure. The neuron can be divided into three main regions, dendrites, cell body and axon (figure 1.1).

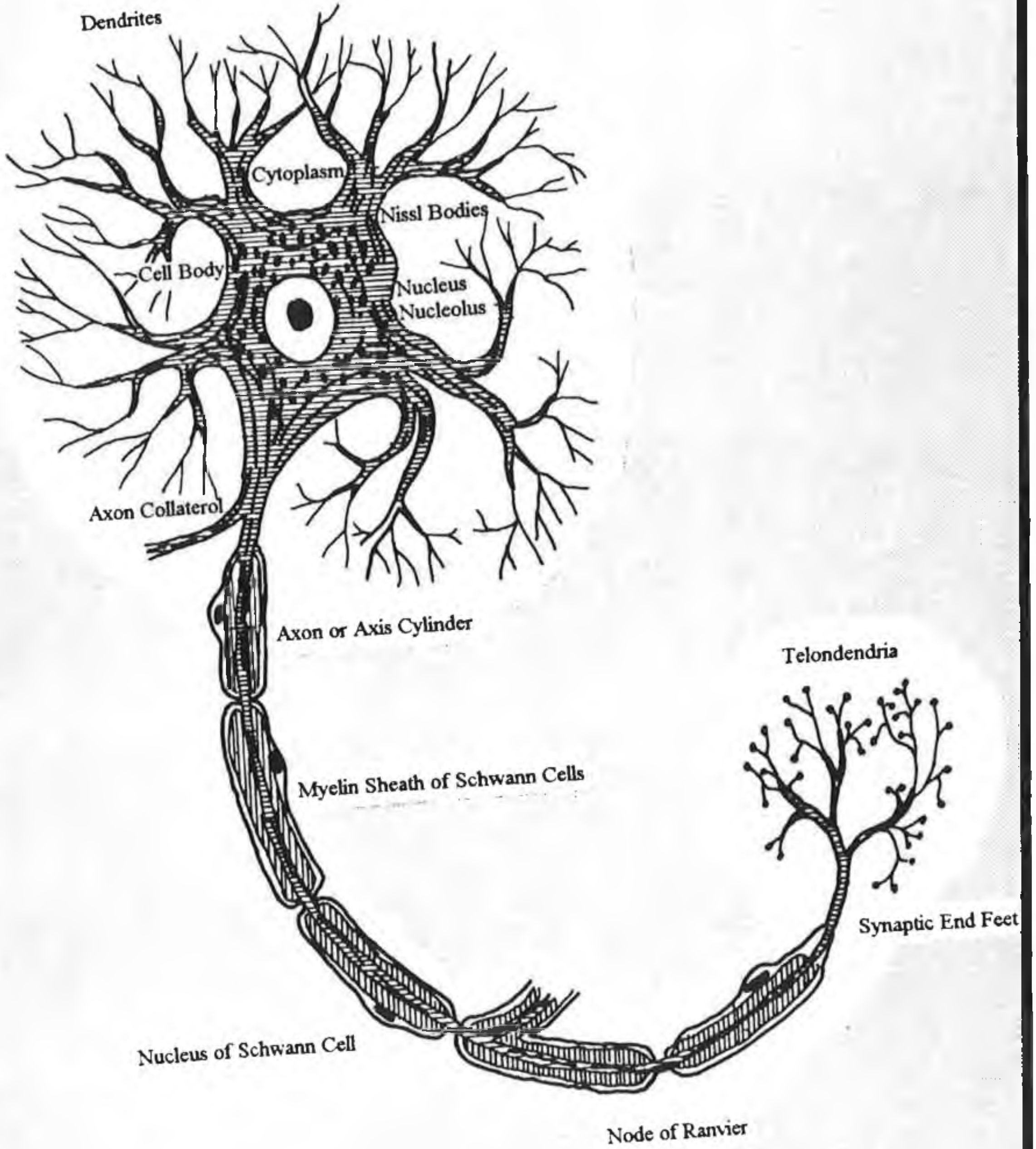


Figure 1.1 - Graphical depiction of a motor neuron (taken from reference 2)

The cell body contains a well defined nucleus and nucleolus surrounded by granular cytoplasm. It also contains Nissl bodies and neurofibrils. Nissl bodies are orderly arrangements of granular rough endoplasmic recticulum and free ribosomes whose function is protein synthesis. Neurofibrils are long thin fibres whose function is to provide structural support to the cell body.

Dendrites are a series of short extensions to the cell body which are responsible for receiving information and transmitting it to the cell body. The dendrites are profusely branched and have a spiny appearance, the resultant dendritic tree surrounds the neuron for an area of approximately 400µm in radius. An axon is an extension from the cell body which is responsible for transmitting information the cell body to other nearby neurons. It is generally longer and thicker than dendrites although the length of an axon can vary greatly depending on the type of neuron which it is a part. Axons for neurons inside the brain can be about 2cm in length but an axon which runs from the spinal cord to the feet can be over 1m long. Axons also vary in diameter and this is related to their rate of impulse conduction. Axons contain one or more side branches called axon collaterols and both the axons and collaterols each terminate by branching into numerous fine filaments called telondria or axon terminals. The distal ends of the telondria are themselves terminated by synaptic end bulbs which are responsible for transmitting information across a synapse to the dendrites of an adjacent neuron.

Variations on this structure are found in cases where a neuron may have no obvious axon but only extensions that seem to receive and transmit information an example of which includes some cells in the retina. Similarly axons may form synapses on other axons and dendrites, examples of which are found in the retina, thalamus and spinal cord although it is rare or absent in the cerebral cortex.

1.1.2 Propagation Of Signals

A neuron has a potential of -70 mV caused by differences in the concentrations of ions on either side of the neural cell membrane. A physical or chemical stimulus can trigger a sudden exchange of ions across the membrane which increases the potential of the neuron. If this potential exceeds a particular potential called the threshold a pulse or action potential (+30 to +40 mV) is stimulated.

The period between the membrane potential difference reaching its threshold and the onset of the action potential is known as the latent period. The peak in the potential of the neuron is then followed by a period of gradual decline (figure 1.2). The absolute refractory period is the time during which a stimulus no matter how strong cannot evoke a second action potential as the neuron re-establishes its chemical and electrical equilibrium. Following the absolute refractory period there is a relative refractory period during which the neuron will not respond to a stronger than normal stimulus. These refractory periods place a limit of about 200 action potentials which can be generated per second which is about 1 million times slower than the transmission of electrons in fast electronics. Similarly the axon which is analogous to an insulated cable is not that well developed for impulse conduction because of its low membrane resistance and high resistance across its axis, meaning that signals that enter the axon tend to be dissipated in 1 or 2 mm. To travel distances the action potential has to be regenerated periodically. This need to boost the signal also limits the maximum speed at which an impulse travels to about 100m/second.

However the nervous system has a major advantage over conventional serial electronics by processing information in parallel. Since there are about 10^{12} neurons in

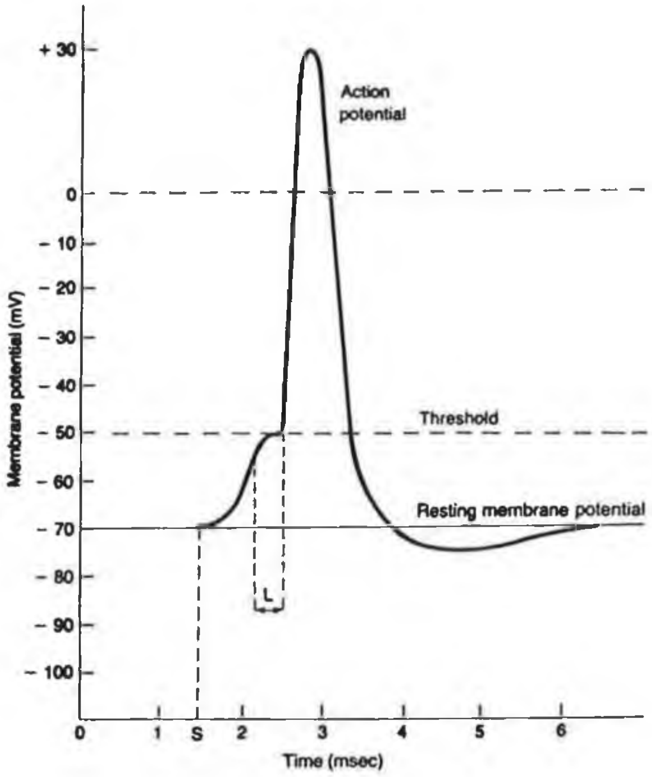


Figure 1.2 - Changes in membrane potential during an action potential. The stimulus is applied at time S. The latent period is shown by time L (taken from reference 2)

the central nervous system there is a potential maximum communication speed of about 10^{14} pulses/second.

1.1.3 Synapse

The space between the synaptic end bulbs of the axon and the synaptic contacts on the dendrites of an adjacent neuron is called the synaptic cleft. This cleft can be about 20-30 nm wide and has too high a resistance for the small current of the nerve impulse to cross. The synapse is the mechanism by which a nerve impulse is transmitted from one cell to another and is mainly chemical rather than electrical in nature. When an impulse travelling down the axon of the presynaptic neuron reaches it terminal, it stimulates the release of excitatory or inhibitory neurotransmitters.

A neurotransmitter which has an excitatory effect increases the membrane potential of the post-synaptic neuron to bring the potential closer to its threshold value. A neurotransmitter which has an inhibitory effect reduces the potential of the post-synaptic neuron.

The receiving neuron has thousands of dendrites receiving synapses from different presynaptic cells. The firing of the receiving neuron is thus dependent on the net effect of the excitatory and inhibitory neurotransmitters to produce a potential that may be large enough to exceed its threshold and trigger an impulse.

There are a wide variety of neurotransmitters including serotonin which has an inhibitory effect and noradrenaline which can be excitatory or inhibitory. Many drugs can affect the neural function by interfering with the synapse, these drugs include LSD which combines indiscriminately with the receptor sites for serotonin, and amphetamines which increase the release of noradrenaline in the brain.

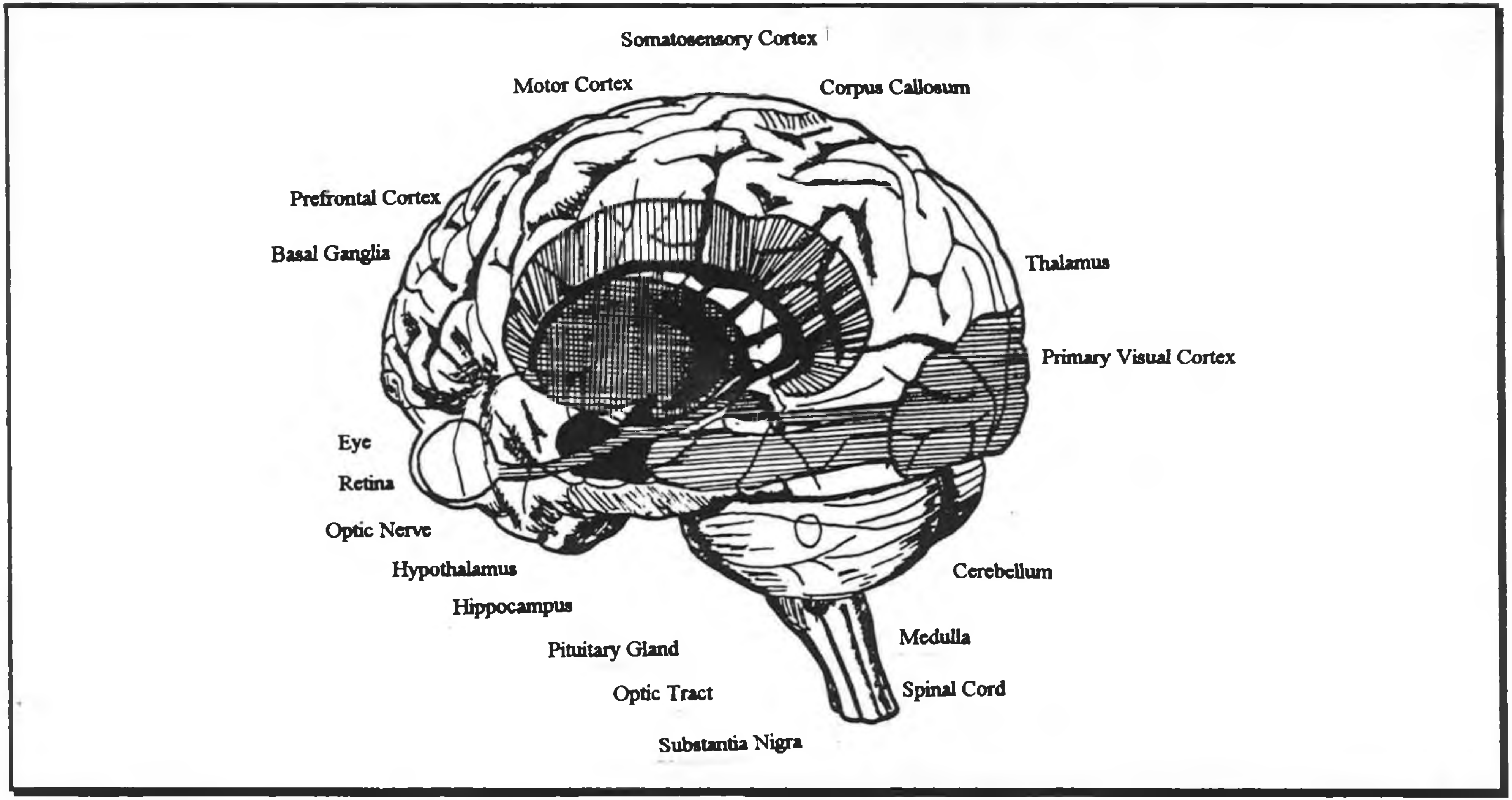


Figure 1.3 - Section through a human brain (taken from reference 5)

1.1.4 Neural Arrangement In The Brain [6-8]

The brain is composed of the brainstem, midbrain and forebrain. The brainstem contains the medulla oblongata and the pons which act as conduction pathways for motor and sensory impulses between the brain and spinal cord and connecting different parts of the brain to each other (figure 1.3).

The forebrain contains the thalamus, hypothalamus and cerebrum. The thalamus acts as the principle relay centre for sensory impulses to the cortex and also functions as an interpretation centre providing conscious recognition of pain, temperature, crude touch and pressure. The major function of the hypothalamus is the regulation of the pituitary gland and the synthesis of hormones stored there, providing a link between the endocrine and neural systems. The cerebrum makes up the bulk of the brain and is composed of the cerebral cortex and cerebral white matter.

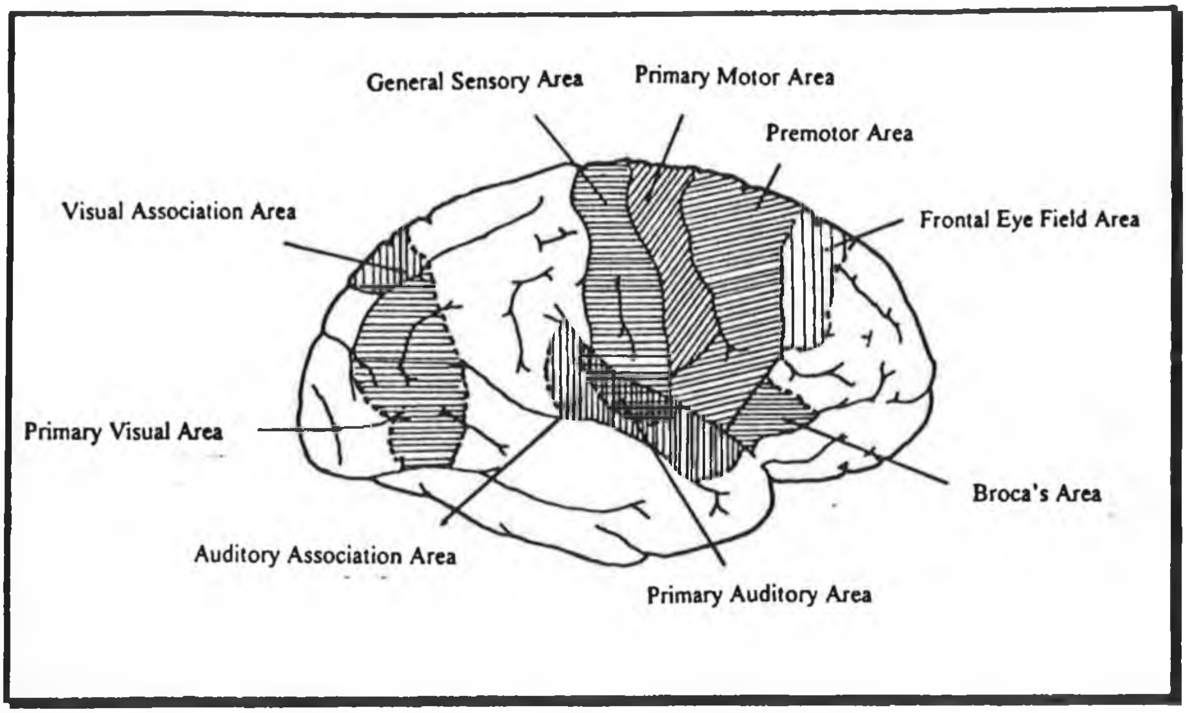


Figure 1.4 - Lateral view of the cerebrum (taken from reference 2)

The cerebral cortex weighs about 1.5 Kg and convolutions of its surface give it a surface area of approximately 0.2 m^2 which is approximately three times greater than if it had been smooth. It is about 2-4 mm thick and consists of about 100 billion neurons arranged in six layers. In many regions of the cortex, groups of adjacent neurons aggregate into higher functional units known as microcolumns. Such microcolumns act by responding to particular stimulus features. Adjacent microcolumns cannot be precisely separated, instead there tends to be a gradual transition in the membership of individual neurons. Microcolumns themselves are organised in specialised areas, each of these areas being a module for a specific task. Most of the cortical areas fall into one of three groups (figure 1.4) namely primary and secondary sensory areas, association fields and primary and secondary motor areas. The sensory or somesthetic areas receive sensations from cutaneous, muscular and visceral receptors in various different parts of the body. Specific neurons on the surface of the cortex receive these sensory stimuli. If a body part is rich in sensors then more neurons on the sensory cortex are required to receive those impulses. Hence the size of the cortical area which represents the body part is determined by its sensory importance with the lips, tongue and fingers being represented by large cortical areas. The representation of the sensory receptors on the cortex display a topographic ordering in which adjacent receptors are mapped to adjacent cortical neurons. In these topographically ordered sensory maps a spatially localised peak is generated from the activity pattern of the cortical neurons whose location represents the signal features being analysed. Topographic maps also exist in the motor regions in which a spatially localised peak creates an activity pattern among motor neurons that triggers a particular movement.

The association fields are responsible for integrating and interpreting sensory signals and also for the storage of memories of past sensory experiences allowing comparisons to be made with current sensory signals.

While it may not be possible at present to relate specific neural arrangements to particular cognitive functions, it can be seen that particular areas of the brain, which contain highly dense organisations of neurons can be related to these functions. Some of the networks which will be discussed later, function by closely mimicking naturally occurring neural arrangements and processing e.g. Kohonen's self-organising feature map.

1.2 Cognitive Aspects Of Neural Networks [9-16]

While there is increasing understanding of the cellular and molecular levels of the brain and nervous system, higher cognitive functions are much more difficult to study because of the complexity of the different levels of organisation of the nervous system. Modelling the brain can proceed by either using realistic memory models or simplified models. Realistic brain models involve large scale simulations of the brain that try to incorporate as much of its cellular detail as possible. But the realism of the modelling strategy produces its own problems. As the model is made increasingly realistic and hence more complex, by the addition of more variables and more parameters, the model runs the risk of becoming as poorly understood as the nervous system it is simulating. Another problem occurs because all the cellular details of the nervous system are not known yet, producing the risk that some important cellular features could be omitted from a model hence invalidating its results. Finally realistically modelling such complex systems tends to be highly computer intensive.

An alternative approach to modelling the function of the nervous system is to use a simplified brain model, connectionist or neural network models perform this function and allow the relationship between learning and representation to be studied.

Representation in the brain involves local coding of memories which are repeated in different regions of the cortex. This may have been a strategy developed by evolution as a security against damage to the cortex, because damage to any local part of the cortex would not block its signals, but rather cause them to be re-routed around the damaged area.

Memories are stored and retrieved in computers by sending binary encoded messages to specific memory locations described by an address. However information in the CNS is transferred in a more irregular fashion which makes it less likely that precise information is sent to a precise address. Rather memory is considered to be content-addressable which means that fragments of a memory can be used as an address to retrieve the whole memory. The storage of memories within a limited capacity network performed by short term memory (STM) can be mimicked by the Hopfield network which can be specified and completed to retain only the most recent information it received. The process of forgetting or erasure of STM traces can be described in Hopfield nets as resulting from interference of recent memories over older ones.

One of the features of the human cognitive system is its ability to adapt to environments whose rules may change unpredictably, without losing skills already learned. An adaptive resonance technique (ART) was developed by Grossberg and Carpenter to mimic this stable-plastic property. An ART is capable of plasticity to learn about new events but yet remain stable and retain pre-existing information.

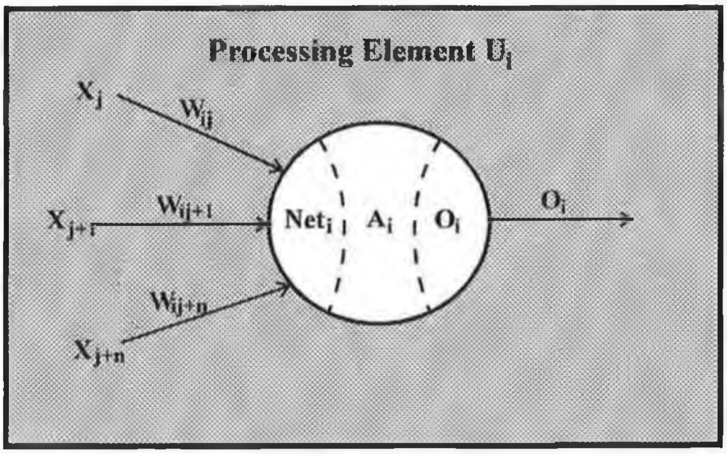
There are many different psychological classifications of learning including classical and operant conditioning and the classification according to whether learning requires a conscious record or not such as explicit and implicit learning.

Hebb bridged the divide between psychological views of learning and neurophysiology by suggesting that associative learning could be produced by a cellular mechanism.

"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process and metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B is increased."

1.3 General Introduction To Neural Networks [17-24]

The basic components of ANNs are a set if PEs known as processing elements (PEs) or neurons. Figure 1. 5 graphically depicts one such PE and the processes which occur within it.



Input j to processing element U_i : X_j

Connection weight between input j and unit U_i : W_{ij}

Total input to unit U_i : Net_i

Activity: $A_i = F_i(Net_i)$ [$F_i()$ represents the activation function]

Output: $O_i = \phi_i(A_i)$ [$\phi_i()$ represents the transfer function]

Figure 1. 5 - Schematic of a processing element employed in an ANN and the processes which occur in it.

The PEs are linked to each other in an ANN by means of network connections. PEs are able to send and receive signals to and from each other and the outside world by means of these connections. Every connection has an associated weight. The total input to a PE is the weighted sum of all the inputs to the PE as seen in eqn **1.1**

$$\mathbf{Net_i = \sum_j X_j W_{ij}} \qquad \textbf{(1.1)}$$

Net$_i$ = the total input to PE **U**$_i$ **X**$_j$ = input j to the PE **U**$_i$

W$_{ij}$ = weight of the connection transferring the input **X**$_j$ to the PE **U**$_i$

A positive weight represents an excitatory input to the PE and a negative weight represents an inhibitory input to the PE (drawing from the analogy of a biological neuron). The pattern of connections between PEs in a networks are often described by a weight matrix $\underline{\mathbf{W}}$ in which the element $\mathbf{W}_{ij}$ describes the strength of the connection between a PE $\mathbf{U}_i$ and another PE $\mathbf{U}_j$. As such, the weight **Wij** is a positive number if the PE $\mathbf{U}_j$ excites the PE $\mathbf{U}_i$ and is a negative number if $\mathbf{U}_j$ inhibits $\mathbf{U}_i$. The absolute value of $\mathbf{W}_{ij}$ specifies the strength of the connection between the PEs.

Once the total input to a PE has been determined, its activity must be calculated. The activity is calculated from the total input to a PE $\mathbf{U}_i$ by means of an activation function $\mathbf{F}_i$. The current activity of the PE may also depend on the previous value of its activity as seen in eqn **(1.2)** giving the activity a time dependent behaviour.

$$\mathbf{a_i\,(t)= F_i(a_i(t\text{-}1), Net_i(t))} \qquad \mathbf{(1.2)}$$

$\mathbf{a_i(t)}$ = activity of the PE $\mathbf{U}_i$ at time **t** $\mathbf{a_i(t\text{-}1)}$ = activity of the PE $\mathbf{U}_i$ at time **t-1**

$\mathbf{F_i()}$ = activity function of the PE $\mathbf{U}_i$ $\mathbf{Net_i(t)}$ = total input to the PE $\mathbf{U}_i$ at time **t**

In the majority of cases however, the activity is the same as the total PE input.

A PE outputs a signal, whose magnitude is related to the activity of the PE, by means of an output or transfer function.

$$\mathbf{O_i=\phi_i(a_i)} \qquad \mathbf{(1.3)}$$

$\mathbf{O}_i$ = output of PE $\mathbf{U}_i$ $\boldsymbol{\phi}_i$ = transfer function of PE $\mathbf{U}_i$ $\mathbf{a}_i$ = activity of PE $\mathbf{U}_i$

But since the activity of a PE usually equals its total input

$$\mathbf{O_i=\phi_i(Net_i)} \qquad \mathbf{(1.4)}$$

There are a wide variety of transfer functions, including

(a) the threshold linear function

$$\phi(z) = \begin{cases} 1 & \lambda z \geq 1 \\ 0 \text{ if} & \lambda z = 0 \\ \lambda z & \text{otherwise} \end{cases} \qquad (1.5)$$

(b) the hyperbolic tangent function

$$\phi(z)=\tanh(z) \qquad (1.6)$$

whose derivative (ϕ') is given by $\phi'=(1-\phi^2)>0$

(c) the sigmoid function

$$\phi(z) = \frac{1}{1+\exp(-z)} \qquad (1.7)$$

whose derivative (ϕ') is given by $\phi'=\phi\,(1-\phi)$. The sigmoid transfer function and its derivative are depicted in figure 1.6

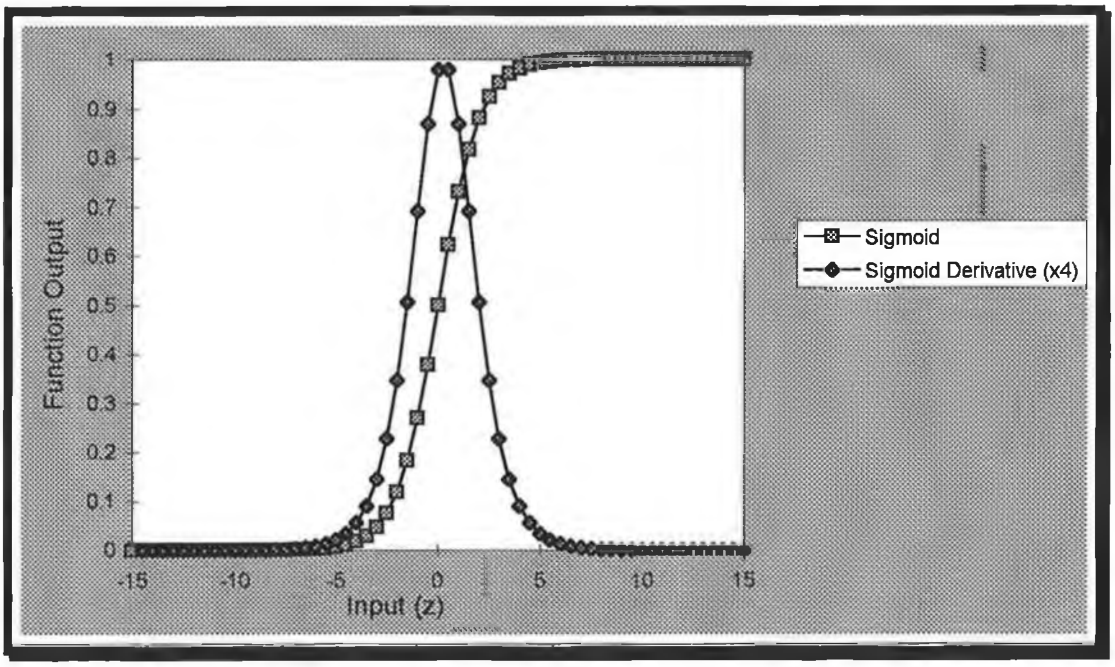


Figure 1.6 - The sigmoid transfer function and its derivative

When these transfer functions are used by processing elements the variable **z** is related to the activity of the processing element. If it is assumed that the activity of the PE is equal to its total input, then **z** refers to the total input to the processing element with an additional term called the bias which controls the horizontal offset of the transfer function.

$$\mathbf{z} = \mathbf{Net}_i + \theta \qquad \textbf{(1.8a)}$$
$$\theta = \text{bias}$$

but $\mathbf{Net}_{\mathbf{i}} = \sum_{\mathbf{j}} \mathbf{X}_{\mathbf{j}} \mathbf{W}_{\mathbf{ij}}$ (eqn 1.1) hence,

$$\mathbf{z} = \sum_{j} \mathbf{X}_j \mathbf{W}_{ij} + \theta \qquad \textbf{(1.8b)}$$

In the case of the one dimensional sigmoid depicted in figure 1.7, when the connection weight linking the input to a PE is positive, positive bias values shift the centre of the sigmoid to more negative values of the total input to the PE and negative offsets shift it to more positive values of the total input to the PE. A negative weight connecting the input to the PE produces the inverse behaviour.

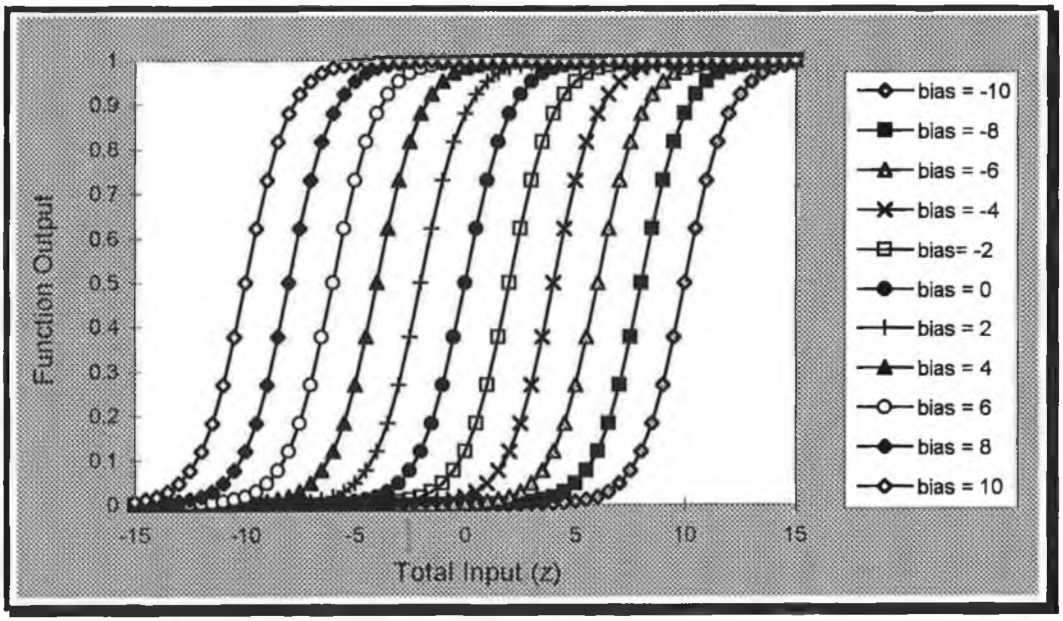


Figure 1.7 - The variation of the position of a one dimensional sigmoid function along the axis representing the total input to a single as a function of the value of the bias to the PE. The PE has one input whose connection weight has a value of 1.0.

The sigmoid is a centrosymmetric function with its centre located at an output value of 0.5. This output value occurs when the total input to the PE and the bias term summed together yield a value of 0.0.

$$\textbf{i.e. } \phi(\mathbf{z}) = \frac{\mathbf{1}}{\mathbf{1+\exp(-z)}} \textbf{=0.5 when z=0.0.} \qquad \textbf{(1.9)}$$

From expression **(1.8)** it can thus be said that the centre of the sigmoid occurs when

$$\theta = -\sum_{j} W_{ij} X_j \qquad \textbf{(1.10)}$$

The shape of the sigmoid (specifically its sharpness) is determined by the magnitude of the connection weights (||**W**||) to the PE. As can be seen in figure 1.8, increasing values of ||**W**|| yield a sharper sigmoid.

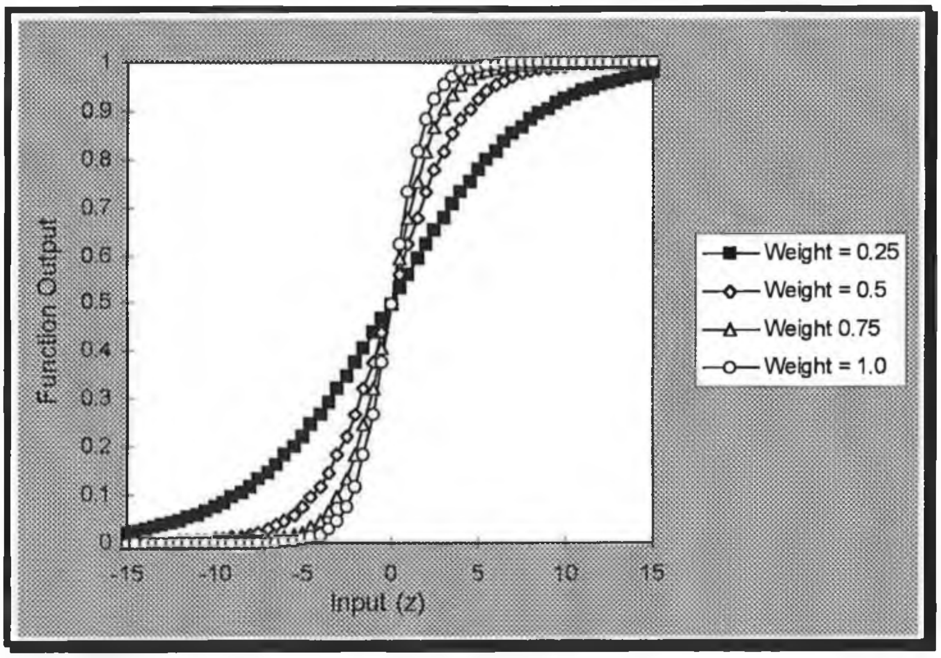


Figure 1.8 - Variation in sharpness of a one dimensional sigmoid for a single PE (with a bias of 0.0) as a function of the connection linking the input to the PE

When **θ=0**, increasing ||**W**|| does not affect the position of the centre of the sigmoid, however when **θ** is not equal to zero, the centre of the sigmoid is shifted by $\frac{-\theta}{\|\mathbf{W}\|}$. In many software applications the bias is treated as the weight of an extra input to processing element U_i. This extra input has a fixed value of 1, as such the bias can be treated as a weight variable

Individual PEs have a limited capability to form mappings. Early studies with neural networks composed of PEs with threshold transfer functions, known as perceptrons, demonstrated that single PEs can only perform classification of linearly separable problems. Linearly separable problems are classification problems in which the patterns in an N-dimensional space may be separated geometrically by planes of N-1 dimensionality (i.e. patterns in a two dimensional space are separated by a line, patterns in a three dimensional space are separated by a plane).

A famous example of a pattern classification task which cannot be performed by a single PE is provided by the Boolean exclusive OR (XOR) function. Since the XOR function is dependent on two binary variables, a perceptron attempting to represent the function would require two inputs, one for each variable (as depicted in figure 1.9)

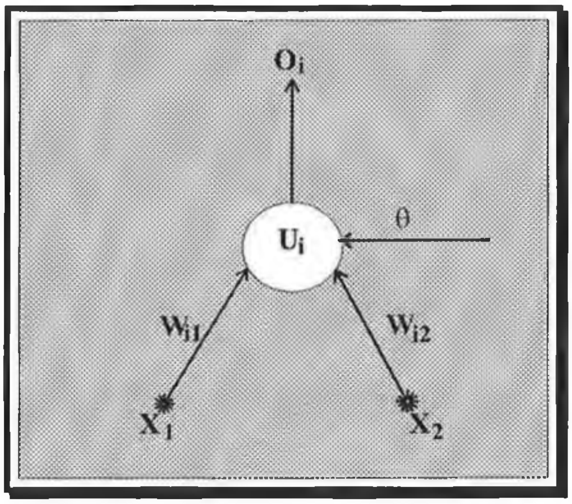


Figure 1.9 - Schematic of a perceptron to be used for the XOR problem. X_1 and X_2 represent the two binary inputs to the perceptron (U_i), W_{i1} and W_{i2} depict the connection weights for these inputs, θ represents the bias and O_i refers to the output of the perceptron

The total input to the unit depicted in figure 1.9 is thus

$$\mathbf{Net_i = X_1W_{i1} + X_2W_{i2}} \qquad \mathbf{(1.11)}$$

The bias of the perceptron acts as a threshold for its threshold transfer function as follows :

$$\mathbf{O_i = \phi(Net_i) = \begin{bmatrix} 1 & W_{i1}X_1 + W_{i2}X_2 \geq \theta \\ 0 & W_{i1}X_1 + W_{i2}X_2 < \theta \end{bmatrix}} \qquad \mathbf{(1.12)}$$

The border between the two classes categorised by the perceptron occurs when the total input to the perceptron equals its bias

$$\mathbf{\theta = W_{i1}X_1 + W_{i2}X_2} \qquad \mathbf{(1.13)}$$

This expression describes a line in the $\mathbf{X_1, X_2}$ plane. The plane and the four possible inputs to the perceptron are depicted in figure 1.10.

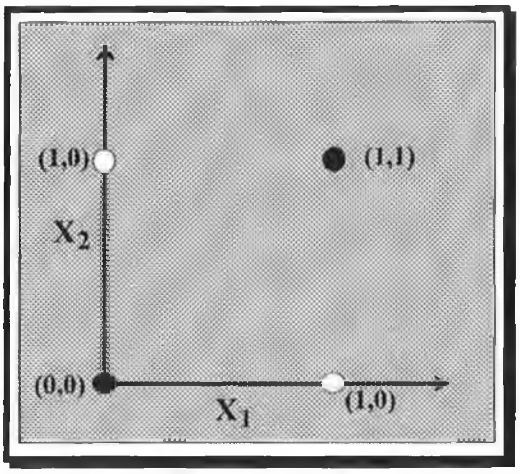


Figure 1.10 - The plane describing two binary variables (X_1 and X_2) used for a Boolean XOR function The resulting four possible inputs to the function are depicted as circles. Those inputs which produce an output of 1 from the XOR function are coloured in white, those which produce an output of 0 are coloured in black.

The task for the network is thus to find a line in the $\mathbf{X_1, X_2}$ plane which separates the inputs (1,1) and (0,0) which have an XOR output value of 0 from the inputs (1,0) and (0,1) which have an XOR output value of 1. There is no single line which can do this, as such a single perceptron cannot perform the XOR function.

The problem of linear separability was overcome by cascading PEs, so that the output of one PE acted as the input to another PE, thereby forming networks composed of multiple layers of PEs in which the outputs of PEs in one layer acted as the inputs for the PEs in another layer.

A two layered perceptron (as depicted in figure 1.11) can solve the XOR problem by forming convex regions in the $\mathbf{X_1}$, $\mathbf{X_2}$ plane (as depicted in figure 1.12).

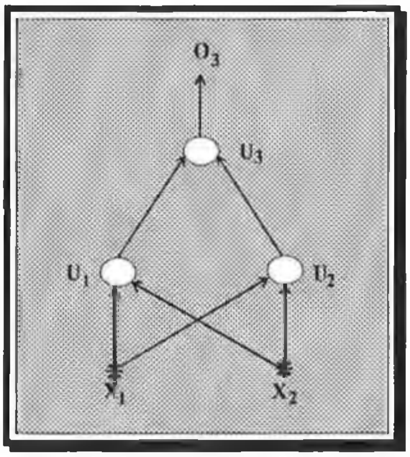


Figure 1.11 - A two layered perceptron to be used for solving the XOR problem. The two binary inputs to the XOR function are represented as X_1 and X_2.. X_1 and X_2 are inputs to the processing elements U_1 and U_2 in the first layer. The outputs from these processing elements act as inputs to the processing element U_3 in the next layer. The output of U_3 is described by O_3. O_3 acts as the final output from the two layered perceptron network.

A convex region is one in which any two points in the region can be joined by a line that does not leave the region. The PEs $\mathbf{U_1}$ and $\mathbf{U_2}$ (in figure 1.11) construct two lines in the $\mathbf{X_1}$, $\mathbf{X_2}$ plane, all the inputs to the left of the line constructed by $\mathbf{U_1}$ will produce an output of 0 from $\mathbf{U_1}$ all the data to the right of the line would produce an output of 1 from $\mathbf{U_1}$. Similarly all the data to the right of the line constructed by $\mathbf{U_2}$ would produce an output of 0 from $\mathbf{U_2}$ and all the data to the right of this line would produce an output of 1 from $\mathbf{U_2}$.. The unit $\mathbf{U_3}$ produces a convex region from the intersections of the lines constructed by $\mathbf{U_1}$ and $\mathbf{U_2}$. All the data within this region would produce an output of 0 from **U**$_3$ and all the data outside this region outside this region would produce an output of 1 from **U**$_3$.

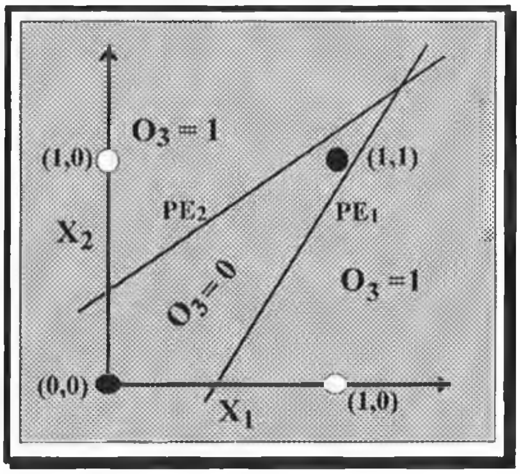


Figure 1.12 - Solving the XOR problem by means of a two layered perceptron network (figure 1.11).

Individual processing elements can be linked to each other in a number of different kinds of connections (e.g. feedforward, feedback and lateral connections) which are determined by the type of ANN model being used. Figure 1.13 depicts the flow of information between units linked by these connections. With feedforward connections, information from PEs in a lower layer (layer j) are transmitted to PEs in an upper layer (layer I). Connections of this type are to be found in multi-layer feedforward nets which will be discussed later

With feedback connections, information from PEs in an upper layer (layer I) are transmitted to units in a lower layer (layer j). Examples of this sort of connection are to be found in the bidirectional associative memory (BAM) type of network. The BAM is a hetero-associative network i.e. a network which accepts input data to which they produce some related but different output data. This is achieved by providing examples of the required input and output data, to allow the network to form a mapping between the input space (described by the input data) and the output space (described by the desired output data).

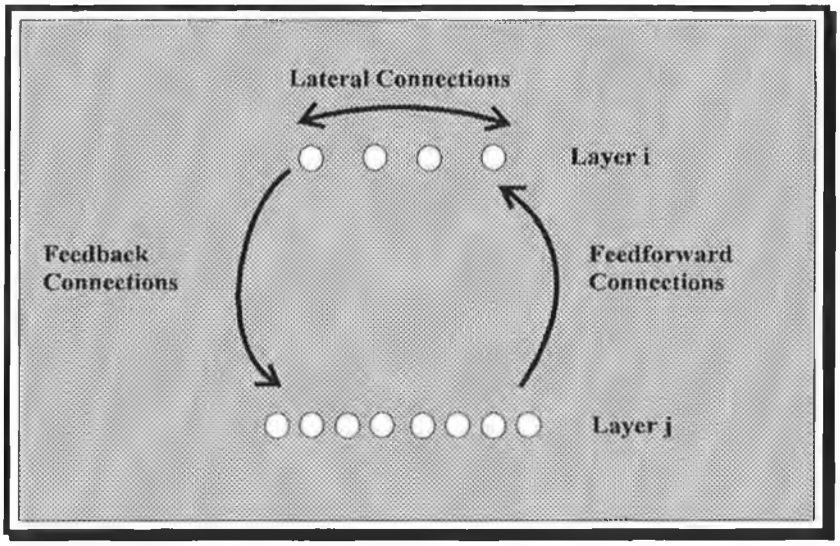


Figure 1.13 - Diagrammatic representation of the flow of information between processing elements in two layers by means of feedforward, feedback and lateral connections.

In a network with lateral connections, each unit interacts with other nearby units in the same layer. These kinds of connections can be found in self organising feature map type networks. This sort of network organises itself to reflect the topology of the space represented by the patterns which are input to it.

It can be seen from the preceding discussion concerning perceptrons that a network formed of individual PEs linked together can be used as a means of mapping a particular input space (described by the input data to the network) to a particular output space (described by the output from the network). If a function **F:** $\mathbf{R}^n \rightarrow \mathbf{R}^m$ maps the **n** vectors $\underline{\mathbf{X}}$ to the **m** vectors $\underline{\mathbf{Y}}$, then the network can be said to have learned the mapping corresponding to the function **F** if the network produces an output vector $\underline{\mathbf{Y}}$ when a vector $\underline{\mathbf{X}}$ acts as the input to the network (for all $\underline{\mathbf{X}}$). For a particular input space (e.g. vector $\underline{\mathbf{X}}$) the nature of the mapping formed by the network is determined solely by the connection weights since the behaviour of the PEs themselves (i.e. activation function, transfer function) do not change. As such, the connections and/or their weights need to be altered in order for a network to form the correct mapping.

These alterations may include the development of new connections between PEs and the loss of pre-existing connections. This thesis however is concerned with the weight adaptations of pre-existing links. The term learning rule is used to describe the strategy employed by a network to improve its mapping. There are two basic approaches by which this can be achieved, namely supervised and unsupervised learning.

In supervised learning, training proceeds using a set of vector pairs (usually called patterns). The vector pairs consist of an input vector and an associated output vector (which acts as a target for the training process). The aim of supervised learning is to adapt the connection weights between PEs so that the network produces the correct output (compared to the target output) when the input vector is presented as an input to the network. One of the most important rules used in supervised learning is the Widrow-Hoff rule also known as the delta rule or least mean squares (LMS) rule.

$$\Delta \mathbf{W}_{ij} = \eta(t_i - O_i)\mathbf{O}_j \qquad \textbf{(1.14)}$$

$\mathbf{t}_i$ = the desired output of the PE $\mathbf{U}_i$ $\mathbf{O}_i$ = the actual output from $\mathbf{U}_i$

$\mathbf{O}_j$ = the output of PE $\mathbf{U}_j$ to which $\mathbf{U}_i$ is connected

$\Delta\mathbf{W}_{ij}$ = change in the weight of the connection between $\mathbf{U}_i$ and $\mathbf{U}_j$

η = learning rate (used to control the size of the weight changes)

This learning rule was initially devised for a single layer network of units with differentiable transfer functions, but was later generalised and adapted to form the backpropagation algorithm, which will be discussed in more detail later.

In unsupervised training , no specific output vector is associated with an input vector to act as a target for the training procedure. Instead, unsupervised learning involves allowing the network to self-organise on the input vectors of the training set. On of the major rules used in unsupervised training is the Hebbian learning rule.

$$\Delta \mathbf{W}_{ij} = \eta \mathbf{a}_i \mathbf{O}_j \qquad \textbf{(1.15)}$$

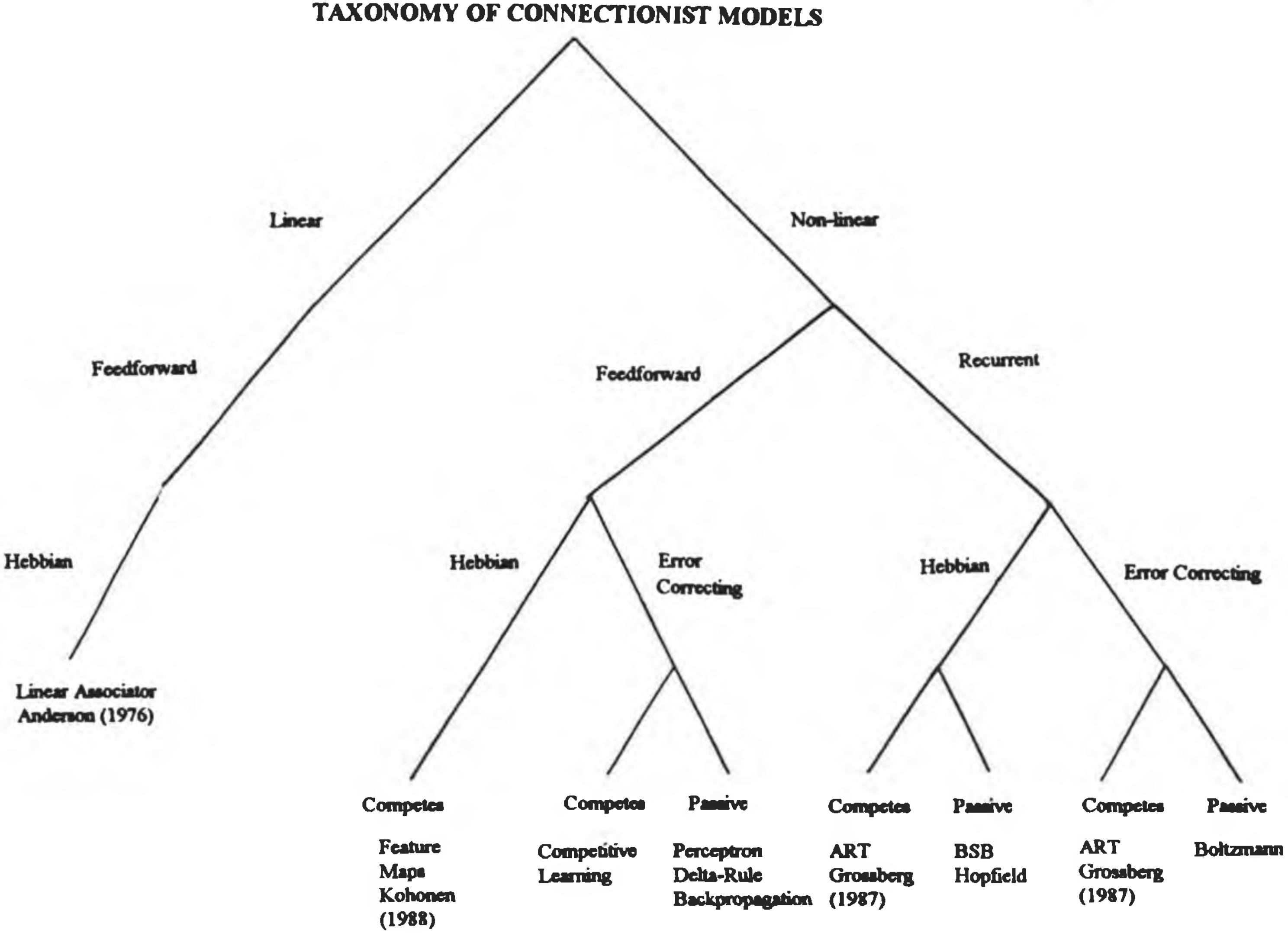


Figure 1.14 - Taxonomy of connectionist models according to connection type and learning rules

The terms $\mathbf{\Delta W_{ij}}$, η and $\mathbf{O_j}$ retain the same meaning as in the previous expression and $\mathbf{a_i}$ refers to the activity of the PE $\mathbf{U_i}$. The basic principle underlying this rule is that if a PE $\mathbf{U_i}$ receives an input from another PE $\mathbf{U_j}$, then if both PEs are highly active (as determined by their activation functions) the weight from PE $\mathbf{U_j}$ to PE $\mathbf{U_i}$ should be strengthened.

There are a number of different methods of classifying ANN models including according to architectures, transfer functions, learning algorithms and the nature of the inputs to the networks. Figure 1.14 depicts different taxonomies of ANNs using two possible sets of classification criterion. In the next few sections some of the ANN models which have found chemical application or which have been mentioned before in terms of cognitive function will be discussed in greater detail.

1.4 Examples Of Different ANN Models

1.4.1 Kohonen's Self-Organising Map [20, 25-29]

Kohonen's self-organising map is an associative net which uses an unsupervised form of training, it has been shown to be able to form topological maps similar to those found in the cerebral cortex. In this type of network, neighbouring cells compete and develop into specific detectors of different signal patterns. The responses from the cells become ordered in the trained network to reflect different features of the training set. Each cell or local group of cells then acts as a separate feature decoder for the same input pattern. As such it is the presence or absence of a response at a cell and not so much the input-output signal transformation or the magnitude of the response that provides an interpretation of the pattern.

1.4.2 The Hopfield Network [29-36]

The Hopfield network has a laterally connected single layer architecture. It can perform auto-association (retrieval of a complete pattern given partial information of the desired pattern) but can also be used for optimisation purposes. Hopfield's early work involved the use of neurons which could take values of 0 or 1 so that the state space over which the network operates is an n-dimensional binary hypercube.

The behaviour of the network was related to a physical system which is described by the components of its state vector $\underline{\mathbf{X}}$ (coordinates $\mathbf{X_1}$, $\mathbf{X_2}$,......$\mathbf{X_x}$). If the system has locally stable limit points $\underline{\mathbf{X_a}}$, $\underline{\mathbf{X_b}}$ etc. then if the system is started near any $\underline{\mathbf{X_a}}$ (eg. $\underline{\mathbf{X}} = \underline{\mathbf{X_a}} + \Delta$)

in Hamming distance and far from the other stable points it would tend to proceed to terminate at $\underline{\mathbf{X_a}}$.

If the locations of the stable points could be considered as the information of a particular memory stored in the network and the starting point $\underline{\mathbf{X}} = \underline{\mathbf{X_a}} + \Delta$ as partial information about the memory, then processes occurring within the network would generate the complete memory $\underline{\mathbf{X_a}}$ from the partial memory. Hence the memory is accessed by initially supplying a part of the memory rather than its address, suggesting that the network acts as a content addressable memory (see **1.2**).

Hopfield later used processing elements whose output $\mathbf{O_i}$ was a sigmoidal function of the input to the element in order to more closely model biological neurons.

The Hopfield net with **N** PEs has a limited capacity of about **0.15 N** patterns that can be simultaneously retained in the net before retrieval errors become significant as the net becomes overloaded. An interesting relationship between REM sleep and overloading of Hopfield nets was discussed by Hopfield and Crick. REM sleep occurs in almost all mammals and in most birds and is believed to play an important role in the development of the nervous system. A characteristic of REM sleep is bizarre intrusions which consist of a mixture of features, all or most of which can be related to events which have occurred recently. When a net is overloaded it can demonstrate different types of behaviour including :producing bizarre associations (fantasy), producing the same state (or one of a small group of states) irrespective of the input (obsession) and responding to inappropriate signals which would not normally evoke a response (hallucination). Hopfield developed a technique known as reverse learning to remove spurious memories created when learning. The technique is very similar to the standard learning procedure except that it starts from a random noisy input. The effect of this procedure is to decrease the total accessibility of spurious states by raising their energies and reducing their basins of attraction relative to the stable states corresponding to desired memories. The use of this technique mimics the proposed function of REM sleep.

1.4.3 Multi-Layered Feedforward Networks

As was described earlier, individual PEs have limited capabilities and more complex mappings may be created by linking the output of one PE to the input of another PE. The multilayered feedforward (MLF) is a network composed of multiple layers of PEs in which PEs in one layer are linked to PEs in other layers by means of feedforward connections. The MLF network is usually composed of an input layer, at least one intermediate layer and an output layer.

The input layer performs no processing on its inputs. It merely takes on the values of the input data to the network, which it distributes to the first hidden layer. The output layer provides the responses or outputs from the network to a particular set of input data. The hidden layer(s) enables the net to form its internal representation of problems, provided that the output values from these PEs are non-linear functions of their inputs. Theses intermediate layers are known as hidden layers because they receive no input from and produce no output to the outside world.

The number of PEs in the input layer is determined by the number of input variables of the task given to the network and the number of PEs in the output layer is determined by the number of output values required by the task.

Figure 1.15 depicts a three layered feedforward network. PEs in the input layer in figure 1.15 are represented by **U**$_i$, PEs in the hidden layer are represented by **U**$_j$ and PEs in the output layer are represented by **U**$_k$. The weight of the connection joining the PE **U**$_i$ in the input layer to the unit **U**$_j$ in the hidden layer is given by **W**$_{ji}$ and between the PE **U**$_j$ in the hidden layer and the PE **U**$_k$ in the output layer by **W**$_{kj}$.

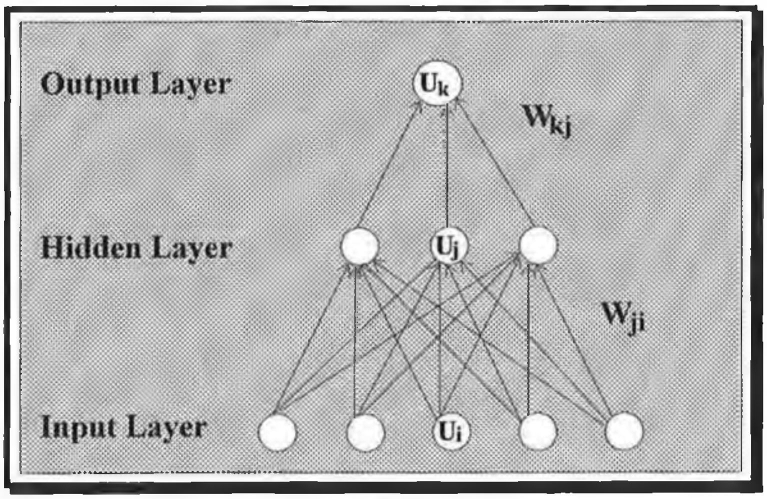


Figure 1.15 - A three layered feedforward network

If the output from a unit in the input layer is passed directly to a unit in the hidden layer without any scaling or other transformation then the input to the hidden unit U$_j$ which is given by $I_j = O_i$ where O$_i$ is the output of the unit U$_i$. As such, given a pattern p, the hidden PE **U**$_j$ receives a total input of

$$\mathbf{Net_j} = \sum_i \mathbf{W_{ji} I_j} = \sum_i \mathbf{W_{ji} O_i} \qquad \textbf{(1.16)}$$

and produces an output of $\mathbf{O_j{=}\phi_j(Net_j)}$ **(1.17)** (ϕ_j is the transfer function of PE **U**$_j$**)**.

A PE in the output layer **U**$_k$, thus receives a total input of

$$\mathbf{Net_k} = \sum_j \mathbf{W_{kj} I_k} = \sum_j \mathbf{W_{kj}} \phi_j \left(\sum_i \mathbf{W_{ji} O_i} \right) \qquad \textbf{(1.18)}$$

and produces a final output of

$$\mathbf{O_k} = \phi_k(\mathbf{Net_k}) = \phi_k\left(\sum_j \mathbf{W_{kj} I_k}\right) = \phi_k\left(\sum_j \mathbf{W_{kj}} \phi_j\left[\sum_i \mathbf{W_{ji} O_i}\right]\right) \qquad \textbf{(1.19)}$$

Determining a method of adapting the connection weights of hidden layer PEs posed a major problem for training MLF networks. The weights to the PEs in the output layer (output units) could be adapted according to the delta rule (equation **1.14)** because the patterns used to train the network contained data pertaining to the target output of the network. These output values of PEs in the output layer of the network could then compared with these target output values. The same is not true for the PEs in the hidden layer (hidden units) however.

1.5 The Backpropagation Algorithm (18, 20, 21, 24, 37-42)

Backpropagation is an algorithm based on the delta rule designed to address the problem of adapting connection weights from hidden units. The backpropagation algorithm employs the instantaneous summed squared error $\mathbf{E_p}$ for the current pattern **p** (given by $\underline{\mathbf{X}}^{\mathbf{p}}$, $\underline{\mathbf{Y}}^{\mathbf{p}}$) defined as follows:

$$\mathbf{E_p} = \tfrac{1}{2}\sum_{k}^{m}\left[\mathbf{Y_k^p} - \mathbf{O_k^p}\right]^2 \qquad \textbf{(1.20)}$$

$\mathbf{E_p}$ = instantaneous error for the training pattern **p**

$\mathbf{Y_k}^{\mathbf{P}}$ = component k of the target output vector $\underline{\mathbf{Y}}$ for pattern **p**

$\mathbf{O_k}^{\mathbf{P}}$ = component k of the network output vector $\underline{\mathbf{O}}$ when the network is presented with training pattern **p**

m= number of components in the output vector $\underline{\mathbf{O}}$

The total or cumulative error of all the patterns used to train the network is given by_

$$\mathbf{E} = \sum_{\mathbf{p}} \mathbf{E}_{\mathbf{p}} \qquad \textbf{(1.21)}$$

(**E**= cumulative error over all the training patterns)

Gradient descent involves movement through the weight space (space composed of the weights of the connections in the network) in the direction which causes the most rapid decrease in **E**. The linear relationship between **E** and $\mathbf{E}_\mathbf{p}$ depicted in equation **(1.21)** suggests that the gradient of the instantaneous error at pattern **p** can be estimated in the gradient descent procedure instead of estimating the gradient of the cumulative error up to the pattern **p**.

The gradient descent approach can thus be written as

$$\Delta_{\mathbf{p}}\mathbf{W}_{\mathbf{kj}} \propto -\frac{\partial \mathbf{E}_{\mathbf{p}}}{\partial \mathbf{W}_{\mathbf{kj}}} \qquad \textbf{(1.22)}$$

$\Delta_\mathbf{p}\mathbf{W}_\mathbf{kj}$ refers to the change in the weight $\mathbf{W}_\mathbf{kj}$ when the network is presented with the pattern **p**.

The derivative can be expressed as a product of two derivative terms via the chain rule.

$$\frac{\partial \mathbf{E}_{\mathbf{p}}}{\partial \mathbf{W}_{\mathbf{kj}}} = \frac{\partial \mathbf{E}_{\mathbf{p}}}{\partial \mathbf{Net}^{\mathbf{p}}_{\mathbf{k}}} * \frac{\partial \mathbf{Net}^{\mathbf{p}}_{\mathbf{k}}}{\partial \mathbf{W}_{\mathbf{kj}}} \qquad \textbf{(1.23)}$$

$\mathbf{Net}_\mathbf{k}^\mathbf{P}$ = the total input to the PE $\mathbf{U}_\mathbf{k}$ when the network is presented with pattern p.

However according to the discussion relating to figure 1.16

$\mathbf{Net}_\mathbf{k}^\mathbf{P}=\Sigma\mathbf{W}_\mathbf{kj}\mathbf{O}_\mathbf{j}^\mathbf{P}$ ($\mathbf{O}_\mathbf{j}^\mathbf{P}$ = output of PE $\mathbf{U}_\mathbf{j}$ when the network is presented with pattern **p**).

As such the second derivative term in equation **(1.23)** can be rewritten as

$$\frac{\partial \mathbf{Net}^{\mathbf{p}}_{\mathbf{k}}}{\partial \mathbf{W}_{\mathbf{kj}}} = \frac{\partial}{\partial \mathbf{W}_{\mathbf{kj}}}\left(\sum \mathbf{W}_{\mathbf{kj}}\mathbf{O}^{\mathbf{p}}_{\mathbf{j}}\right) = \mathbf{O}^{\mathbf{p}}_{\mathbf{j}} \qquad \textbf{(1.24)}$$

If a term $\boldsymbol{\delta}_\mathbf{k}^\mathbf{P}$ is defined as follows

$$\delta_k^p = -\frac{\partial E_p}{\partial Net_k^p} \qquad (1.25)$$

then expression **(1.23)** may be rewritten as

$$\frac{\partial E_p}{\partial W_{kj}} = \delta_k^p O_j^p \qquad (1.26)$$

Hence in order to perform gradient descent in **E**, weight changes should be made according to

$$\Delta_p W_{kj} = \eta \delta_k^p O_j^p \qquad (1.27)$$

η= a proportionality constant known as the learning rate

This expression has a similar form to the standard delta rule **(1.14)**. The difficulty with this approach is in determining δ_k^p for each PE in the hidden layer of the network (ie δ_j^p). As will be shown below however, the backpropagation algorithm can recursively calculate the δ_k^p values by propagating error terms backwards through the network.

The term δ_k^p can be rewritten as the product of two derivatives

$$\delta_k^p = -\frac{\partial E_p}{\partial Net_k^p} = -\frac{\partial E_p}{\partial O_k^p} \cdot \frac{\partial O_k^p}{\partial Net_k^p} \qquad (1.28)$$

O_k^p = output of PE U_k when the network is presented with pattern **p**.

The first term in the derivative product describes the change in instantaneous error as a function of the output of PE U_k and the second term describes the change in output of PE U_k as a function of changes in its total input.

As discussed earlier in the section concerning the processing which occurs in a single PE, the output of a PE is determined by the operation of a transfer function on its total input (assuming the activity of the PE is the same as its total input).

i.e. $\mathbf{O_k^p = \phi_k(Net_k^p)}$ **(1.29)**

ϕ_k = the transfer function of the PE $\mathbf{U_k}$

Thus the second term in the derivative product may be rewritten as follows:

$$\frac{\partial \mathbf{O_k^p}}{\partial \mathbf{Net_k^p}} = \frac{\partial}{\partial \mathbf{Net_k^p}}\left(\phi_k\left[\mathbf{Net_k^p}\right]\right) = \phi_k^{'}\left(\mathbf{Net_k^p}\right) \qquad \mathbf{(1.30)}$$

$\phi_k^{'}$ = the first derivative of the transfer function of the PE $\mathbf{U_k}$

The instantaneous error for the unit $\mathbf{U_k}$ is given by equation **(1.20)** [where case $\mathbf{O_k^P}$ refers to the output of PE $\mathbf{U_k}$ to pattern **p** and $\mathbf{Y_k^P}$ refers to the target output for this PE]

Hence $$\frac{\partial \mathbf{E_p}}{\partial \mathbf{O_k^p}} = -\left(\mathbf{Y_k^p - O_k^p}\right) \qquad \mathbf{(1.31)}$$

The derivation for the change in the weight between a hidden neuron and an input neuron is the same as that described by equations 1.23 to 1.30. However after this point the derivation proceeds differently because there is no way of explicitly calculating an instantaneous error for such neurons.

The total input to a PE $\mathbf{U_k}$ in the ouput layer in figure 1.15 is determined by the outputs of PEs $\mathbf{U_j}$ in the hidden layer and the weights of the connections linking the PEs.

$$\mathbf{Net_k^p} = \sum_j \mathbf{W_{kj} O_j^p} \qquad \mathbf{(1.32)}$$

$\mathbf{Net_k^P}$= total input to PE $\mathbf{U_k}$ in the output layer when the network is presented with pattern **p**

$\mathbf{O_j^P}$ = output of PE $\mathbf{U_j}$ in the hidden layer when the network is presented with pattern **p**

The term $\frac{\partial \mathbf{E_p}}{\partial \mathbf{O_j^p}}$ can thus be rewritten as

$$\frac{\partial \mathbf{E_p}}{\partial \mathbf{O_j^p}} = \sum_k \frac{\partial \mathbf{E_p}}{\partial \mathbf{Net_k^p}} \cdot \frac{\partial \mathbf{Net_k^p}}{\partial \mathbf{O_j^p}} \qquad \mathbf{(1.33)}$$

$$\sum_k\left(\frac{\partial E_p}{\partial Net_k^p}\cdot\frac{\partial}{\partial O_j^p}\left[\sum_j W_{kj}O_j^p\right]\right) \qquad (1.34)$$

$$\sum_k\left(\frac{\partial E_p}{\partial Net_k^p}\cdot W_{kj}\right) \qquad (1.35)$$

since $\delta_j^p = -\frac{\partial E_p}{\partial Net_{kj}^p}$ then

$$\frac{\partial E_p}{\partial O_j^p} = -\sum_k \delta_k^p W_{kj} \qquad (1.36)$$

The results of this derivation may be epitomised as follows:

(I) A weight connecting PEs in different layers should be modified by a quantity proportional to the product of the error term **δ** from the PE receiving its input along the connection **(U$_j$)** and the output of the PE sending its output data along the connection **(U$_i$)**.

$$\Delta_p W_{ji} = \eta\delta_j^p O_i^p \qquad (1.37)$$

(ii) The error term is specified according to the PE receiving it and is determined recursively starting with the output units.

If the PE exists in the output layer, then its error is given by

$$\delta_k^p = \left(Y_k^p - O_k^p\right)\phi_k^{'}\left(Net_k^p\right) \qquad (1.38)$$

If the PE exists in a hidden layer them its error term is given by

$$\delta_j^p = \phi_j^{'}\left(Net_j^p\right)\sum_k \delta_k^p W_{kj} \qquad (1.39)$$

An outline of the algorithm for the three layered feedforward network is as follows :

1. Apply an input vector **$\underline{X}^P$ =(X$_1^P$, X$_2^P$,........X$_n^P$)** to the PEs in the input layer of the network.

2. Calculate the total input values to PEs in the hidden layer of the network

$$\mathbf{Net_j^p = \sum_i^n W_{ji} X_i^p + \theta_j} \qquad \textbf{(1.40)}$$

θ_j = the bias of PE $\mathbf{U_j}$

3. Calculate the outputs from PEs in the hidden layer

$$\mathbf{O_j^p = \phi_j\left(Net_j^p\right)} \qquad \textbf{(1.41)}$$

4. Use the outputs from the PEs in the hidden layer to calculate the total inputs to PEs in the output layer

$$\mathbf{Net_k^p = \sum_j W_{kj} O_j^p + \theta_k} \qquad \textbf{(1.42)}$$

5. Calculate the outputs from the PEs in the output layer

$$\mathbf{O_k^p = \phi_k\left(Net_k^p\right)} \qquad \textbf{(1.43)}$$

The stage in the operation of the backpropagation algorithm represented by the steps so far is known as the *forward pass* phase of the algorithm.

6. Calculate the error term for the output units

$$\mathbf{\delta_k^p = \left(Y_k^p - O_k^p\right)\phi_k'\left(Net_k^p\right)} \qquad \textbf{(1.44)}$$

7. Use the error term from PEs in the output layer to calculate the error term for the PEs in the hidden layer

$$\mathbf{\delta_j^p = \phi_j'\left(Net_j^p\right)\sum_k \delta_k^p W_{kj}} \qquad \textbf{(1.45)}$$

8. Update the weights between the output and hidden layers

$$\mathbf{W_{kj}(t+1) = W_{kj}(t) + \eta \delta_k^p O_j^p} \qquad \textbf{(1.46)}$$

$\mathbf{W_{kj}(t)}$ = the connection weight before adaptation

$\mathbf{W_{kj}(t+1)}$ = the connection weight after adaptation

9. Update the weights between the hidden and input layers

$$\mathbf{W}_{ji}(t+1) = \mathbf{W}_{ji}(t) + \eta\delta_j^p X_i^p \qquad \textbf{(1.47)}$$

The procedures described in steps 6-9 are known as the *backward pass* phase of the backpropagation algorithm.

The steps discussed above can be extended to describe the learning procedures for networks with more than one hidden layer. In this case, the output from the first hidden layer acts as an input to the second hidden layer and steps 2 and 3 are repeated for as many times as there are hidden layers in the network. Similarly step 7 is repeated for the same number of times as steps 2 and 3, with the error terms from PEs in the hidden layer closer to the output layer being propagated to PEs in the next hidden layer to which they are connected. Step 9 is also repeated as many times as there are hidden layers in the network, updating the weights of the connections between the hidden layers. The procedure represented by steps 1-9 are repeated for every pattern in the training set. This number of repetitions is referred to as an *epoch* and is repeated until a particular termination criterion has been achieved.

This form of the backpropagation algorithm uses per-sampling adaptation of the connection weights. Another form of the backpropagation algorithm exists which uses batch updating of the connection weights. In this case the weights are adapted after all the patterns in the training set have been presented to the network (one epoch) by the cumulative or summed weight adaptations to each connection weight for each presented pattern.

The learning rate parameter η is important in controlling the learning process of the backpropagation algorithm. Unfortunately there are no set rules for what value to take for the learning rate, its values can vary depending on the type of transfer function being used by the PEs and the different modifications to the backpropagation algorithm being employed in the training process. Typical values for the learning rate exist in the range 0.1 to 10.

A common modification to the backpropagation algorithm involves the use of a momentum term in which a certain proportion of the previous change in a connection weight is added to its present weight change

$$\Delta W_i(t) = \eta \delta_i O_i + \alpha\left[\Delta W_i(t-1)\right] \qquad \textbf{(1.48)}$$

The momentum effectively filters out high frequency variations of the error surface of the network in the weight space (composed of the connection weights of the network). It is useful in situations in which a network is trying to move downhill in a valley with steep sides but a shallow slope along the valley floor. The momentum term must be less than 1.0 in order for learning to be stable. Otherwise the weight vector (composed of the connection weights) will move on the error surface of the network under greater influence from the previous gradient than the current gradient. Typical settings for this parameter range from 0.4 to 0.9.

Training can be considered as a search in the weight space of the neural network for the combination of weights that minimises the error measure. The error surfaces of feedforward networks are generally very complex. The surface has multiple copies of the global minimum due to degeneracy arising from the symmetry in the network architecture. Other convolutions of the surface can also complicate the training process. These include the existence of local minima and flat areas or troughs of very small slope. These complexities of the weight space can make it very difficult to train a network and there is no guarantee that when a net has trained that it has converged to a global minimum in the search space.

When discussing the performance of a network, it is necessary to consider both its cumulative error from the data on which it was trained and also the ability of the network to generalise. Generalisation refers to the ability of a network to extrapolate and interpolate from the data on which it was trained to new data drawn from the same distribution as the original training data. The ability of a network to generalise can be examined by applying the input patterns from the new data to the network inputs and observing the resulting network outputs. A comparison between the network outputs for the new data and the desired outputs for these data gives an indication of the network's ability to generalise. While the number of input PEs to a network and the number of output PEs in a network are defined by the problem to be solved by the network, the number of hidden layers and the number of PEs in these layers are variables for the network designer to decide upon.

Hornik and White (41) showed that multi-layered feedforward networks with as little as one hidden layer are a class of universal function approximators, that can approximate a function to any degree of accuracy, provided the network contains enough PEs in the hidden layers. Lippmann (20) showed that any classification task can be solved by a network with two hidden layer s containing enough PEs. This suggests that no more than two hidden layers are needed for a network to represent a non-linear transform or perform pattern classification tasks. The number of units in a hidden layer is itself dependent on the problem being solved by the network. With too few PEs in the hidden layer, the network will not be able to represent the non-linear transform. With too many PEs in the hidden layer, the computing time will increase, in addition it may limit the ability of the network to generalise, as the large network may store specific features of the patterns in the training set rather than their underlying features.

There are many other modifications to the backpropagation algorithm to improve its movement through the weight space, and also to improve the ability of the network to generalise outside its training set (for further details the reader is recommended to read the thesis by Albert Bos (42)). The backpropagation algorithm and its modifications has been applied to a broad range of chemical problems which will be discussed later in this introduction.

1.6 Genetic Algorithms

Artificial life is an area of study which views life as a property of the organisation of matter, rather than a property of the matter which is so organised. It takes a bottom up approach to the consideration of organisms, viewing an organism as a large population of simple components. It synthetically constructs large aggregates of these simple rule-governed components, which interact non-linearly with each other in the support of global dynamics frequently seen in natural systems.

In nature, an organism has a combination of phenotypic characteristics determined by it genotypic makeup. Genetic information about an individual is stored in the form of DNA which consists of long chains of adenine, thymine, cytosine and guanine nucleotides. Different sequences of nucleotides on the DNA strands known as genes encode the production of one or more related proteins. The term chromosome is the name give to the state of the DNA strands when they are densely packed together. Each gene has several alternative configurations, which produces differences in the manifestations of the characteristics associated with the gene.

There are tens of thousands of genes in the chromosomes of a typical vertebrate, each of which has several alleles. The complexity of these natural systems arises from the interactions of these genes. A major contributor to this complexity is the phenomenon known as epistasis, whereby the effect of a particular allele is strongly dependent on the effects of other alleles that are present in the chromosome. In this case small changes in the genetic makeup of a single gene can result in very large effects in the organism. Because of this phenomenon, there is no simple way to apportion credit to individual alleles for the performance of the resulting phenotype.

Given the ubiquity of epistasis, adaptation by means of change in genetic makeup becomes primarily a search for co-adapted sets of alleles, or alleles of different genes, which together significantly augment the performance of the corresponding phenotype. Various different environmental niches define different adaptational opportunities for the genetic system. In order to exploit these opportunities the adaptive system must select and use the sets of coadapted alleles which produce the phenotypic characteristics most appropriate for the survival of the organism to the environment.

Genetic algorithms (GAs) are a group of techniques that were initially designed to simulate the behaviour of biologically based adaptive systems. These techniques were developed to study what kind of emergent behaviour arose from a set of simple rules and how changes in the algorithm would affect this behaviour. Holland's Schema Theorem (43) showed that a simple genetic algorithm (SGA) functioned as a means of optimising a sequential decision process involving uncertainty in the form of lack of prior knowledge. Holland showed that GAs produce near optimal sequences of trials for problems with high levels of uncertainty. While the GA was not initially developed as a function optimiser itself, modifying the GA can produce powerful GA-based optimisers (44).

Genetic algorithms behave in a similar fashion to the class of *moderate* search techniques. These techniques make fewer assumptions about the response landscape than strong methods, and are less computationally demanding than weak methods. *Strong methods* (45) such as gradient search and simplex techniques use heuristics which concentrate on local areas of the response surface, by searching for an optimum, in the neighbourhood of a particular point. These methods make assumptions about the response surface (e.g. the existence of a derivative for gradient search) to aid the exploration. These two characteristics of local searching and assumptions about the response surface can limit the applicability of *strong methods* on surfaces which are rough, discontinuous or multimodal. However in cases when the surface is smooth and the search technique is close to an optimum, these methods will converge very rapidly and efficiently to the optimum position on the surface.

Weak methods, which include grid searching, make very few assumptions about the surface and tend to search in an enumerative or random manner. While these methods are very robust, they are also computationally inefficient and become quickly limited as the dimensionality of a problem increases. *Moderate techniques* such as genetic algorithms and simulated annealing are another class of optimisation techniques which bridge the division between the weak and strong methods. Genetic Algorithms (GA) use random events, which are directed by information about the previously unknown surface, accumulated during the search for the optimum, in order to identify and focus on regions of the search space which are more likely to contain the required optimum (46, 47). *Moderate techniques* tend to be particularly useful in situations where a problem is very complex (as the assumptions on which strong methods are based will make them fail), and also in cases where the dimensionality of the problem is high (as the computational time required for weak techniques to find an optimum becomes impractical).

GAs use concepts gleaned from Darwin's theory of evolution by natural selection and also from the mechanisms involved in the alteration and transfer of genetic information to individuals in a population (48). These processes enable a population to both survive in the surrounding environment and continuously adapt to any changes that may occur in it. A GA which mimics these evolutionary processes, is implemented as an iterative procedure which maintains a constant size population **P** of possible solutions to a particular optimisation problem. The **P** possible solutions to the optimisation problem contained within the population are known as chromosomes and the M variables of the search space are encoded on the chromosomes as M genes. The numeric values of these variables which might represent concentration, potential, wavelength or any other analytical parameter are encoded in a uniform alphabet on the genes. Quite commonly the alphabet used is binary, in which case the number of bits used to represent the variable in the gene determine the resolution with which the variables are optimised.

The basic configuration of the GA used in this study was composed of an initial population setup followed by a cyclic repetition of evaluation, scaled reproduction, single point crossover and single point mutation (as depicted in figure 1.16).

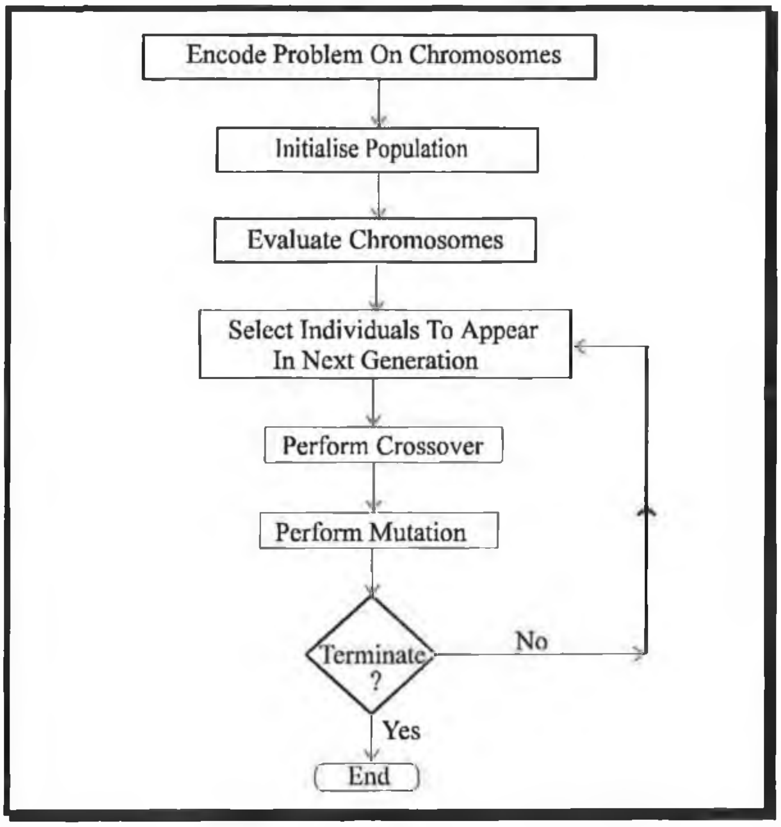


Figure 1.16 - Basic configuration of a genetic algorithm

1.6.1 Stages In The Operation Of A GA

1.6.1.1 Population Initialisation.

In the GA studies discussed in this thesis, the parameters to be optimised are encoded on a chromosome in a binary form as sixteen bit words. This was achieved by establishing a range for each parameter within which the residual sum of squares (SSE) of the calibration model would be expected to be minimised. The range for each parameter would thus define a continuous but finite area of the search space for each parameter within which the search would proceed. The maximum and minimum values of these ranges were then mapped into the numeric ranges 0 to 2^{16} -1 using the following transform.

$$\mathbf{rescale} = \frac{(2^{16}-1)[U - U_{min}]}{U_{max} - U_{min}} \qquad (1.49)$$

Where **[Umin, Umax]** represents the interval within which the optimisation is to proceed for a particular parameter **U**. The ranges for these variables were set with different widths depending on the significance of the contribution of each variable to the error of the calibration model. The individual genes corresponding to the parameters of the Nikolskii-Eisenman expression were then concatenated to form a single bit string or chromosome. Variable sized populations were filled using a random number generator to produce numbers in the range 0 to 2^{16} -1 for each gene. Press et al. discuss the difficulties associated with system supplied random number generators (49a) as such code for a portable random number generation based on Knuth's subtractive method (47, 49b) was developed and seeded from the PC clock using the include file **RANDHEAD.C** (see the software appendix).

1.6.1.2 Evaluation

Each candidate model described by a particular chromosome was evaluated in order to determine how closely it described the experimental data. This was achieved by decoding the binary genes on the chromosomes into values of the model parameters. The resulting model was then used to predict the values of the experimental data. A sum of squared errors (SSE) expression (as described by equation **1.50**) was thus used as an objective function to quantitatively express how well the candidate model described the experiemntal data.

$$\mathbf{SSE} = \sum_{\mathbf{s}=1}^{\mathbf{N}} \left(\frac{\mathbf{E} - \tilde{\mathbf{E}}}{\mathbf{E}} \right)^{2} \qquad \mathbf{(1.50)}$$

Where $\tilde{\mathbf{E}}$ is the estimated value of the experimental data predicted from the model encoded on a chromosome, **E** is the actual value of the experimental data and **N** refers to the number of samples used to form the experimental data seta calibration set. Since a genetic algorithm is normally concerned with maximising the performance or fitness of a population of chromosomes, the reciprocal of the error (i.e./SSE) was used to describe the fitness of each candidate chromosome. The overall evaluation procedure is depicted in figure 1.17.

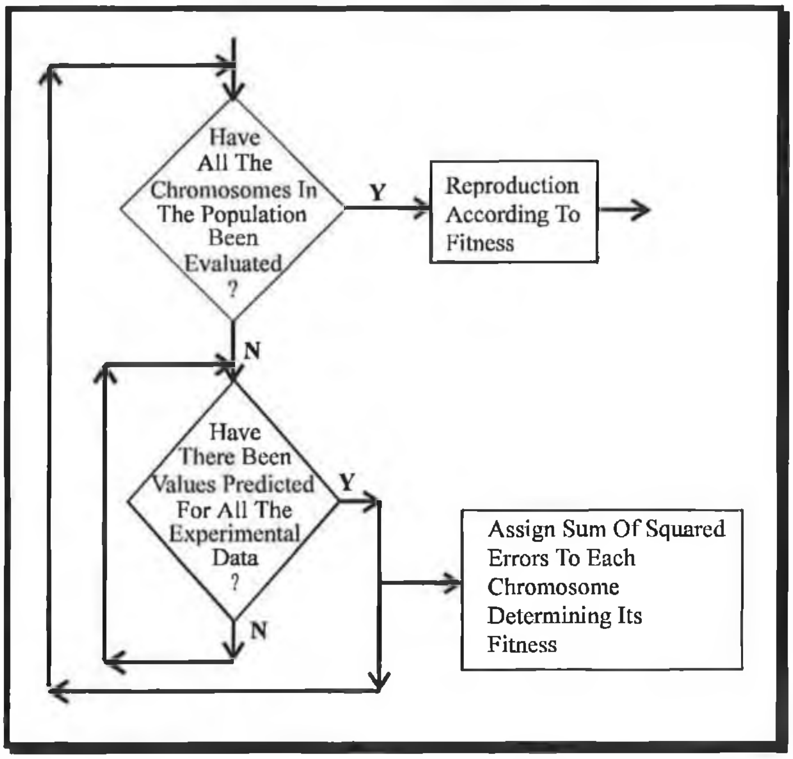

Have All The Chromosomes In The Population Been Evaluated ?
Y
Reproduction According To Fitness
N
Have There Been Values Predicted For All The Experimental Data ?
Y
N
Assign Sum Of Squared Errors To Each Chromosome Determining Its Fitness

Figure 1.17 - Schematic of evaluation procedure of the genetic algorithm

1.6.1.3 Scaled Reproduction

The existing calibration models in a particular population are exploited to improve the future performance of the population by performing biased reproduction ensuring that particularly good calibration models have greater probability of being reproduced in the next generation than less good models. This involves sorting the models according to their SSE by means of a quick-sort procedure. In this study different forms of reproduction and scaling are investigated namely roulette wheel selection, rank scaling and linear prescaling.

1.6.1.4 Crossover

In this study, single point crossover is implemented by randomly selecting two chromosomes from the current population by a variety of different methods. Figure 1.18 depicts the general processes occurring during the crossover operation.

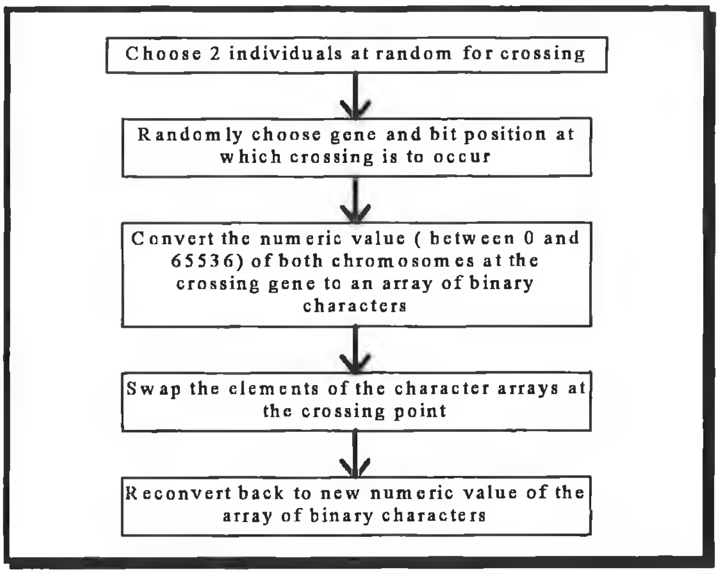

Choose 2 individuals at random for crossing
Randomly choose gene and bit position at which crossing is to occur
Convert the numeric value (between 0 and 65536) of both chromosomes at the crossing gene to an array of binary characters
Swap the elements of the character arrays at the crossing point
Reconvert back to new numeric value of the array of binary characters

Figure 1.18.- Schematic of general processes occurring during the crossover process of a genetic algorithm

The position at which crossover was to occur on both bitstrings was chosen at random. The genes on both chromosomes in which the cross point occurred were converted from integers in the range 0 to 2^{16} -1 into their binary equivalents. This was achieved by running a bit mask across the integer representations of the genes and storing the resulting binary representations in character arrays. The elements of the two character arrays (corresponding to the two genes within which crossover was to occur) to the right of the crosspoint were swapped as depicted in figure 1.19.

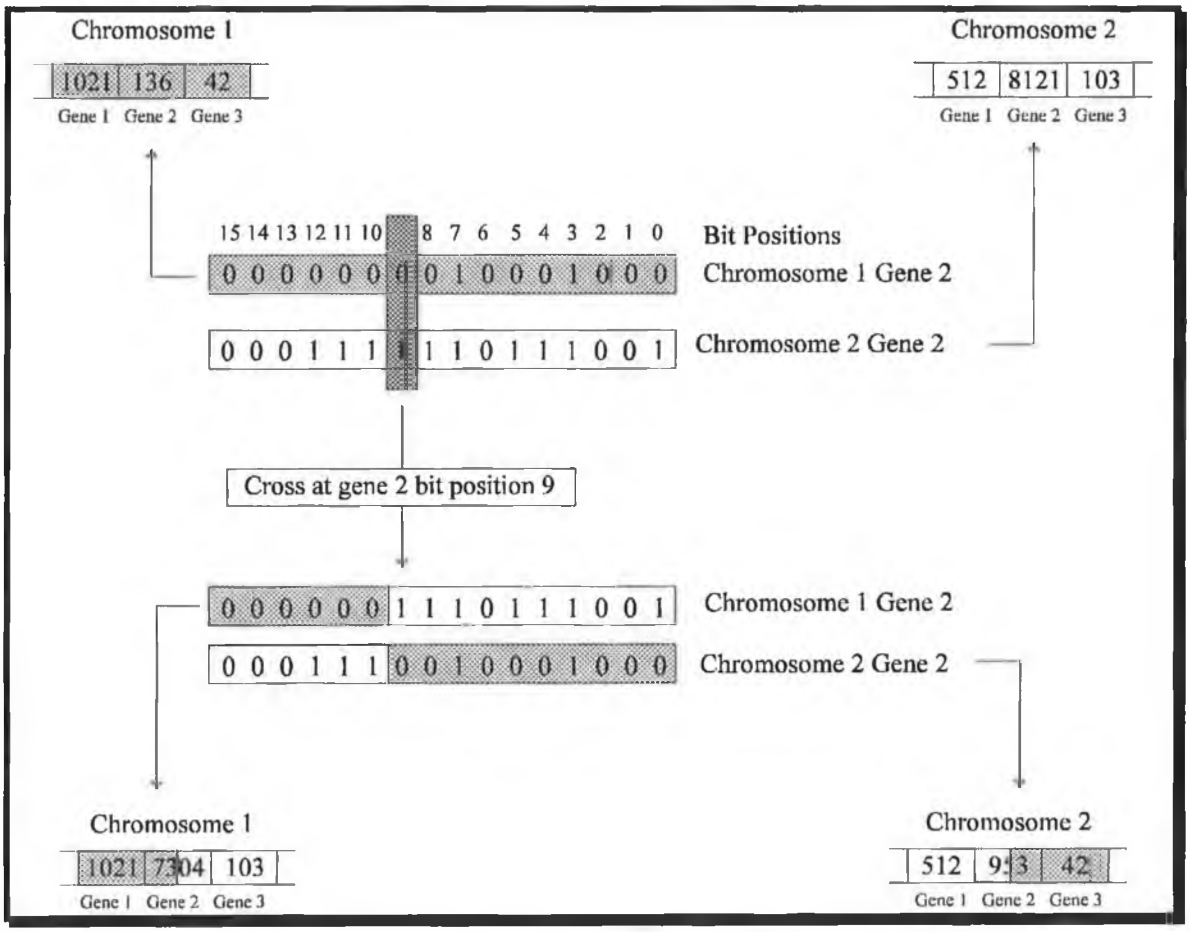


Figure 1.19 - Processes occurring during crossover at the bit level. Chromosome 1 (dotted) and Chromosome 2 (white) are chosen to cross at bit position 9 in gene 2. Bits to the right of position 9 in gene 2 are switched between the two chromosomes as are the genes to the right of gene 2 in each chromosome.

The two resulting binary character arrays were reconverted into genes of an integer form according to the following relation.

$$\mathbf{x} = \sum_{i=1}^{L} \mathbf{a}_i \mathbf{2}^{i-1} \qquad \textbf{(1.51)}$$

Where **x** represents the parameter (integer value) and $\mathbf{A=a_L, a_{L-1},.........a_2,a_1}$ represents the **L** element bit string

1.6.1.5 Mutation

Mutation is an operator which inverts the value of a bit which is chosen at random on a chromosome (figure 1.20).

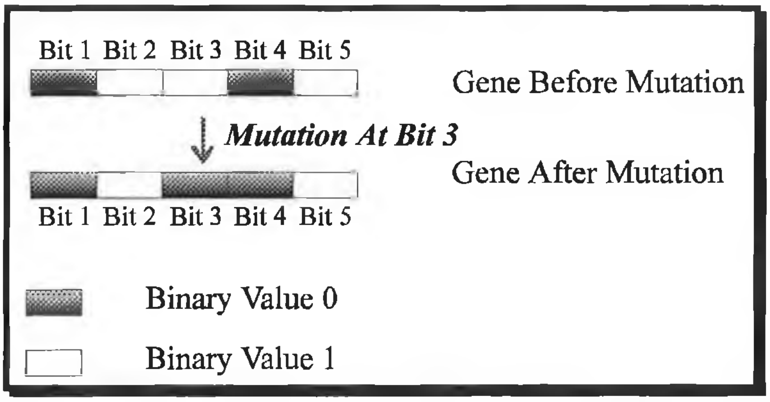


Figure 1.20 - Processes occurring during mutation. Bit 3 is has a value of 0 before mutation, after mutation bit 3 is toggled to a value of 1, (all the other bits in the gene remain the same)

If it were to be used as the main way of generating new individuals in the population, GAs would become a randomised search which, as described earlier, is inefficient for problems of high dimensionality. Instead, mutation acts to prevent genes which may occur in bad combinations with other genes, from being completely lost to the population (by selective reproduction), when they may become more useful later in the GA in a combination with better genes. In this study, chromosomes were mutated at randomly chosen points on the bit strings of chromosomes by inverting the binary value of the bit at the mutation point through a logical XOR operation with a bit of value 1.

In summary the simple genetic algorithm (SGA) involves encoding a problem of M variables onto a population of N chromosomes with M genes (figure 1.16). The chromosomes are allowed to reproduce according to their fitness determined by the objective function and are randomly subjected to genetic operators such as crossover and mutation. This iterative process is continued until some termination criterion is met. It can be seen that the GA is a highly parallel search technique whose parallelism is achieved by the simultaneous movement of the P search points through the M dimensional multivariate search space (43).

1.6.2 Premature Convergence As A Problem In The Implementation Of GAs

This is a problem encountered in the implementation of GAs whereby selective reproduction causes better individuals to occur in the population in increasing proportions, until the population is dominated by an individual chromosome and the GA converges to a sub-optimal solution. According to the theory of GAs (described in appendix 10), the number of copies of a particular schema should increase or decrease exponentially depending on whether its average fitness is above or below that of the population average. But, as it is not possible to directly calculate a schema average fitness, it must be estimated by the finite sequential sampling process of evaluating the individual chromosomes in the current population. This sampling process introduces its own sampling error, which when coupled with the variance associated with the stochastic procedures implemented in the GA, can lead to wide deviation between the number of schema which occur in the population and the number which would be predicted from the theory of GAs. Repeated iterations of the GA cause these errors to accumulate leading to search trajectories different from those probalistically predicted as useful schemata disappear from the population. This process is known as genetic drift and is manifested in practice as a premature loss of diversity in the population. In this chapter a number of different modifications of the genetic algorithm aimed at reducing this problem will be investigated.

For further details of the theory and use of GAs, references (47, 50-54), the tutorial by Davis (55), the textbook by Goldberg (47) and the seminal work by Holland (43) are recommended.

1.7 Literature Review

1.7.1 Applications Of The Feedforward Network In Chemistry

In recent years artificial neural network techniques and genetic algorithms have been successfully applied to a broad spectrum of problems from disciplines as diverse as commerce and engineering. However until recently these techniques have not found major use in analytical chemistry, although the following examples should demonstrate the virtual explosion in analytical applications of neural networks and genetic algorithms that has occurred in the last few years. It is clear that while a number of different types of networks have been used to solve various kinds of problems, the backpropagation training algorithm for feedforward nets and modifications of this training algorithm have been most commonly used for analytical applications. Similarly it is clear that a simple genetic algorithm is not sufficient to deal with a large number of analytical problems and instead is being modified and hybridised with other optimisation techniques.

Qian and Sejnowski (56) investigated the use of the backpropagation algorithm for predicting the secondary structure of local sequences of amino acids. The success rate obtained using this approach was 64% for the testing set of proteins non-homologous with the training set on three types of secondary structure namely α helix, β sheet and coils. The correlation coefficients for the structures were found to be higher than those obtained using other methods. This method was further developed by Kneller et al. (57) who improved prediction to 79% for all a proteins by adding neural network units to detect periodicities in the input sequence and by also using tertiary structural classes.

Thomsen and Meyer (58) investigated the use of the backpropagation algorithm for the classification of 500 MHz ^{1}H NMR spectra of six sugar alditols recorded at a concentration of 6 mg in 0.5 ml D_2O. The effect of a variety of distortions on the classification of the alditols was studied. It was found that right and left shifted specra were classified correctly as were spectra distorted by the addition of noise. Reducing the intensity of spectral features had a more dramatic effect on the ability of the network to classify the alditols.

Bos et al (59) applied a backpropagation net for the simultaneous determination of calcium and copper(II) ions in binary mixtures of copper(II) nitrate and calcium chloride, and also for the simultaneous determination of potassium, calcium, nitrate and chloride ions in mixtures of potassium chloride, calcium chloride and ammonium nitrate using arrays of ion-selective electrodes.

Aoyama (60,61) studied the application of a modification of the backpropagation algorithm for the development of quantitative structure activity relationships for carboquinones and benzodiazepines. In this modification to the backpropagation algorithm a new transfer function was used for the neurons to mix linear and non-linear operation, and a partial correlation coefficient was used to determine the influence of an input parameter on the network output.

Robb and Munks (62) applied a simple linear neural net (i.e. without a hidden layer) to the interpretation of the infra-red spectra of organic compounds in order to identify different functional groups present in the molecules. A detection level of 53.3% on a test set of 541 compounds with 24 functional groups was achieved, and of the detected groups, 91.5% were identified correctly. They later improved upon their linear model by adding a hidden layer (63) and demonstrated an improvement in the identification of functional groups in the training set using a net with a hidden layer when compared with the linear net.

Wythoff, Levine and Tomellini (64) used a fully connected three layer back-propagation net for the verification and recognition of infra-red spectral peaks of vapour phase species. The widths of some of the spectral features were of the order of the resolution with which the spectra were acquired, meaning that spectral peaks were sometimes represented by a single data point (which made the distinction between the spectral features and noise spikes rather difficult). It was found that the addition of a noise reference to the signals in the training set produced an improvement in the mean absolute difference between the desired identification of patterns in the test set and the actual output from the network from an original value of 0.357 to a value of 0.19.

Long, Gregoriou and Gemperline (65) used a backpropagation net for nonlinear multivariate calibration. They then applied neural network techniques to the determination of protein in wheat from near infra-red (NIR) spectroscopic data and to the quantitation of the ingredients in two pharmaceutical products using UV-visible spectroscopic data. It was found that principal components regression (PCR) performed better than neural networks when using perfectly linear simulated data. This was attributed to model error produced from fitting a sigmoid function to linear data. In the case of the wheat, PCR only slightly outperformed the neural network approach. The spectral data from the pharmaceuticals displayed non-linearities due to stray light and interactions between the pharmaceutical components at high concentrations. It was found that the non-linear response was inadequately modelled by the PCR, and consequently the neural network calibration results were found to be slightly better than the PCR results.

Gardner, Hines and Wilkinson (66) applied a back-propagation network to the pattern recognition of signals produced from an array of tin oxide gas sensors. It was found that vapour patterns of methanol, butan-1-ol, propan-2-ol, 2-methyl-1-butanol and ethanol could be recognised at the parts per million (ppm) level in all cases.

Curry and Rumelhart (67) investigated modifications of the conventional backpropagation algorithm for classification of low resolution mass spectra of unknown compounds according to the presence or absence of 100 organic substructures. They encountered similar overtraining problems as seen by Bos (68) and found that a weight reduction method reduced the overtraining effect more than the addition of random noise. It was found that the neural network approach classified mass spectra more reliably than STIRS, the most successful mass spectrum classifier (using a nearest neighbour classification algorithm) for which there was sufficient detail to allow comparison.

Nakamoto et al. (69) used a backpropagation algorithm to classify whiskies using quartz resonator array odour sensors. Recognition probability was improved using a statistical approach for choosing membranes for the array. An average recognition probability of 76% was achieved using this approach. The measurement system was improved (70) by modifying the flow system (using standard air as opposed to atmospheric air, minimizing temperature variations, maintaining constant flow rate during the measurement, using stainless steel for the production of the sensor cell and the distributor to minimise the adsorbtion of the odourant molecules) and introducing bilayer lipid materials to the sensing membrane set. The resulting recognition probability was improved to 94%. This approach was extended to identifying perfumes and flavours (71). In this case a 100% recognition probability was obtained. In addition the ability of the network to discriminate between a pure sample of orange flavouring and the orange flavouring contaminated with 2-butylidene cyclohexanone and decanal for different concentrations of the contaminant was examined. In both case a detection limit of 0.5% volume contaminant was determined for discrimination between pure and contaminated and pure samples, which is comparable to the detection limit of the human olfactory system for these compounds.

Elrod et al. (72) investigated the use of backpropagation for predicting the products of electrophilic aromatic substitution of monosubstituted benzenes. Information was represented to the network in the form of connectivity matrices and charge vectors. It was observed that the more descriptive, less specific representation derived from connectivity matrices gave better results than representation involving charge vectors. Overall, the predictions made by the neural network approach were found to be comparable with those made by expert systems and synthetic organic chemists.

Chang et al (73) used a backpropagation net for the recognition of patterns produced from an array of piezoelectric crystals coated with phosphatidylglycerol, phosphatidylethanolamine, phosphatidylserine and lipid A which responded to amyl acetate, acetoin, menthone, methanol, ethanol, propanol and butanol. It was found that odorants could be identified with 70% probability.

Glick and Hieftje (74) investigated the use of the backpropagation algorithm for classifying metal alloys according to the concentrations of their Si, Cu, Fe, Cr, Al, Mo and Ni constituents. it was found that the neural network approach performed slightly better than the KNN approach.

Sundgren, Winquist, Lukkari and Lundstrom (75) applied a three layer neural net to the quantification of the individual components in two types of gas mixtures. The first gas mixture was comprised of hydrogen, ammonia, ethanol and ethylene in air and the second mixture contained hydrogen and acetone in air. The components of the mixtures were to be quantified based on the responses of six metal oxide semiconductor MOSFETs which were exposed to the mixtures. It was found that both hydrogen and ammonia concentrations were predicted more accurately by the neural net than by partial least squares (PLS) and the same was found for the hydrogen and acetone in the two component mixture.

Meyer et al **(76),** used a backpropagation net for the identification of one-dimensional 1H-NMR spectra of oligosaccharides derived from xyloglucan (a plant cell wall hemicellulose). It was found that the spectra were identified correctly by the neural net even when the spectra were perturbed by slight variations to their chemical shifts.

Bos and Weber (77) carried out a comparison of the training of neural networks by backwards error propagation and genetic algorithms for quantitative x-ray fluorescence (XRF) spectrometry of iron, nickel and chromium samples. The backwards error propagation trained networks performed better than the genetic algorithm trained networks for these samples. However it was found that the two types of training procedures produced nets which performed equally well when trained on a larger data set composed of XRF spectra of thin iron and nickel layers on a substrate.

Long et al (78) used a backpropagation net for pattern recognition of jet fuel chromatographic data obtained by GC and GC/MS. It was found that classification by neural nets was dramatically better than that by K nearest neighbour and (SIMCA) for the water soluble fraction of the jet fuels.

Gemperline, Long and Gregoriou (79) investigated the use of different multivariate techniques and neural networks for the detection and modelling of non-linear regions of spectral response in multivariate, multicomponent spectroscopic assays of pharmaceutical products. It was found that neural networks could be used to develop non-linear calibration models that performed better than PCR or PLS.

Anker and Jurs (80) used a three layered fully connected network employing a backpropagation algorithm for predicting C^{13} NMR chemical shifts of keto-steroid carbon atoms. The results obtained from this approach demonstrated a 77% improvement on classification of the training set and 63% improvement for the test set when compared with linear regression analysis.

Smits et al. (81) investigated the application of backpropagation to the identification and counting of phytoplankton based on various optical parameters obtained using a flow cytometer. It was demonstrated that the neural network was capable of identifying specific kinds of algal species such as poisonous and non-poisonous species and was also capable of identifying a number of different algal species that appear in one sample. The robustness of the neural network to changes in the flow cytometer settings were also studied. Improved performance was obtained if the settings were used as an additional input parameter to the network.

Weijer et al. (82) studied the application of backpropagation algorithm to the development of a model describing the relationship between the physical structure and mechanical properties of poly(ethylene terephthalate) yarns. The physical structure of the polymers produced under widely varying manufacturing conditions were studied by means of x-ray diffraction, density measurements, sonic pulse propagation and measurements of birefringence. These properties were then subjected to PCA to produce five independent structure parameters. The mechanical properties of the yarns were determined from their stress-strain curves and their shrinkage at different temperatures and strain conditions. The neural network approach compared favourably with results obtained from PCR and PLS.

Song et al. (83) investigated the use of the backpropagation algorithm to develop a model describing the relationship between the spectral structure of Eu(II) ion transition emission in complex fluorides and the structures of the host complex fluoride lattices. A recognition rate of 98% was obtained for the training set and 92.6% for the test set, which compared favourably with results obtained using conventional pattern recognition techniques.

Bos et al (68) investigated different modifications of the backpropagation algorithm, applied to the prediction of water content of cheese from milk composition and process parameters. The complex nature of the data and the limited number of patterns available for training and testing caused difficulties when they trained networks with the standard backpropagation algorithm. While the error from the training set decreased during training, the error from the test set would decrease during the early stages of training, but start to increase as training progressed. It was suggested that this was caused by the network learning the noise in the training set and not the underlying principle which would enable the network to generalise to the test set. The modifications to the backpropagation which were studied to alleviate this problem, included learning with extra neurons in the output layer, the addition of random noise to the training patterns, addition of uniform noise to the training patterns, Weigand's weight reduction, Hinton's weight decay and descending epsilon approaches.

It was found for neurons employing a conventional sigmoid transfer function, that the addition of noise both random and uniform, had little or no effect on errors in the test set. The descending epsilon, weight reduction and weight decay approaches all reduced the overtraining effect, with the weight decay approach producing the lowest error on the test set. Adding extra units to the output layer caused a worsening in the network's performance, in which case the weight reduction approach worked best. Neurons with radial basis transfer functions reversed the trend found with neurons employing sigmoid transfer functions, whereby descending epsilon, weight reduction and weight decay approaches performed worst and little change was observed between the performance obtained when using the conventional backpropagation and the addition of noise. Symmetrical sigmoidal neurons performed in a similar fashion to neurons using radial basis functions, but in general an increased performance was observed using symmetrical sigmoids in comparison with conventional sigmoids. Best results overall were obtained when using radial basis function neurons with noise.

Bos et al. (84) studied the application of a modified form of the backpropagation algorithm known as the scaled conjugate gradient approach to modelling iron, chromium and nickel concentrations in stainless steel, determined by XRF. The neural network approach was compared with the Rasberry-Heinrich model which is based on empirically modelled interelement effects. The calibration coefficients of this model were determined using genetic algorithms. The neural network approach was also compared against a linear modelling procedure using singular value decomposition. It was shown that the neural network could produce robust models which outperformed the other methods for Ni and Fe calibration and outperformed the Rasberry-Heinrich model for Cr calibration.

Song and Yu (85) applied the backpropagation algorithm to investigate the quantitative structure activity relationship of dihydropteridine reductase inhibitors derived from 1-methyl-4-phenyl-1,2,3,6 tetrahydropyridine. The results obtained compared favourably with those obtained by stepwise multidimensional linear regression analysis. They introduced a term known as the partial correlation index in order to determine the influence of individual input variables on the network's output. It was found that a molecular shape parameter indicating the distance of the heteroatomic ring nitrogens was most important in determining the network's output. The three most important parameters determined with this index agreed with those used in previous QSAR studies

Li et al. (86) investigated modifications to the conventional backpropagation algorithm, applied to the prediction of furnace lining durability based on six parameters describing the chemical composition of the furnace lining and the conditions under which the furnace was operated. The modifications to the backpropagation algorithm involved the use of neurons with a symmetrical sigmoid transfer function and also a conjugate gradient training algorithm. The predictive results obtained from the neural network approach were better than those obtained from PCR and PLS regression. Convergence was accelerated using the symmetrical sigmoid function when compared with backpropagation using conventional sigmoid functions. Convergence was also accelerated with the conjugate gradient training technique, furthermore the conjugate gradient technique converged with fewer hidden nodes than could be achieved using the backpropagation algorithm, which is an important feature in preventing overfitting.

Smits et al. (87) investigated the use of a modular feedforward network (as opposed to a flat feedforward network) for the classification of alcohol and carbonyl functional groups. It was found that the neural networks involved in the study had difficulty interpreting spectra of multi-functional compounds, especially if the network had not been trained using examples of the combinations of functional groups involved. The incorporation of simulated spectra containing the combinations of functional groups, improved the performance of the network but did not completely solve the problem. Networks dedicated to determining the presence of a particular class of functional group e.g. alcohol, performed better than a flat network designed to classify all the functional groups.

When the dedicated networks were combined in a hierarchical structure, it was found that false positive errors or doubts in the coarse classification module were compensated for by the finer sub-class classification modules. The performance of the modular network was also found to be comparable with that of a human expert.

1.7.2 Applications Of Genetic Algorithms In Chemistry

Wu and Freeman (88) used a GA to optimise pulse shaping functions used in NMR spectroscopy, in order to develop an absorbtion mode profile that had a single central response with decaying side lobes and a dispersion mode profile that was weak across the entire frequency range. These pulse shapes were then used for ω-COSY and soft-COSY techniques.

Lucasius and Kateman (89) used GA techniques to select a set of wavelengths which were optimally selective for the measurement of the absorbances of individual species in multi-component samples. The spectral regions of four RNA nucleotides with ranges of thirty six wavelengths each were used for this study. Generally a set of six or seven wavelengths were needed to be chosen for the analysis because of overlap of the spectra of the nucleotides. The GA were intialised with a population of 100 chromosomes and within 70 generations had converged. This was noted to be ten times faster than convergence by a process of stepwise removal of the wavelengths. The stepwise wavelength removal process proved to be robust under different initial wavelength distributions, indicating that local minima were not a big problem, and hence the GA may not have been the best approach to the optimisation problem. However if the wavelength selection problem had been further complicated by the inclusion of more components into the samples and the use of spectra with more than thirty six wavelengths, the use of the GA may have been more justified.

Lucasius et al. (90) investigated the application of a GA for the determination of conformational parameters of DNA hairpins. The conformational parameters used in this study included torsion angles in the DNA backbone, torsion angles of the bonds connecting the furanose ring and the base ring system and internal rotational parameters for the endocyclic torsion angles of the furanose rings. These parameters were coded as bit strings using Gray's code. Strongly correlated conformational parameters were also represented as tightly linked genes, to reduce the positional bias in the crossover operator and minimize its disruptive effect.

The resultant chromosomes were evaluated using a cascaded hybrid evaluation criterion, such that the fitness of a chromosome was determined by two different procedures. The first procedure involved the calculation of H-H distances based on the DNA conformation proposed by the chromosome in the GA. These distances were then compared with those calculated from two dimensional (2D) Nuclear Overhauser Enhancement (NOE) NMR spectrometry. The second procedure involved the calculation of a theoretical NOE table from the proposed DNA conformation and its comparison with the experimental NOE table. It was found that the GA optimisation time was dependent on the choice of the initial population, but that the conformations finally determined were not strongly determined by this parameter. On simulated experimental NOE tables created for a mixture of DNA conformations, among which one was chosen to be ten times more probable than the others, it was found that the GA always converged towards the prominent conformation, although there were sometimes stable subpopulations for the other conformations. A similar study was performed by Blommers et al. (91) using a GA to determine optimal torsion angles, phase angles and pucker amplitude to determine the conformation of the photodimer cis, syn-dUp[ ] dT from experimental NMR data.

Fontain (92) used a GA to develop different permutation vectors in order to optimise the minimal chemical distance (CDm) between isomeric ensembles of molecules. The CDm refers to the constitutional similarity of the molecules being studied and can be used as a means of developing reaction trees used to postulate mechanistic and synthetic pathways of organic reactions. It was found that there was a dramatic reduction in computing time required for optimisation when using the GA as opposed to other optimisation techniques.

Wienke et al. (93) compared five optimisation techniques namely, pattern search, simplex search, gradient search, simulated annealing and GA for the determination of an optimum set on a simulated three dimensional surface based on response surfaces found in chromatography and spectroscopy. The simulated surface had one global minimum and two larger local minima and numerous hills, valleys and walls. The search space was divided into smaller windows and the search techniques were applied several times in each window. Each algorithm had difficulty finding the global minimum with the GA being the most successful, finding the global minimum 94% of the time and the steepest ascent method being the least successful, finding the minimum 14% of the time. The GA was successful when intialised at the top of the two local optima whereas all the other techniques were unsuccessful in these cases. The GA was also applied to the optimising the intensities of atomic emission lines of Cu, Ni, Cr, Mn, Fe and Si for the determination of these trace elements in highly purified alumina powder. The variables of current intensity of the electric arc and the concentration of the added purified NaCl were adapted by the GA to produce an optimal emission intensity.

The global optimum for the analysis was determined within a few generations under constrained and unconstrained conditions.

Li, Lucasius and Kateman (94) developed a modification of the simple genetic algorithm (SGA) using diversity functions based on principles from ANOVA to control the operation of the GA and hence reduce the problem of premature convergence by reduced population diversity. The diversity functions involved the measure of the diversity between the chromosomes in the population and also the diversity between the alleles in all the chromosomes. The SGA was modified by adding two new parameters namely a duplication parameter, producing a form of elitism in the process and a selective or mating parameter, controlling the selection probability distributions of different individuals in the population. A dynamic genetic algorithm (DGA) was developed in which the probabilities of crossover and mutation were allowed to vary during evolution according to the values of the new diversity functions in order to maintain diversity and guarantee a broad search of the search space. The DGA and SGA were compared on a problem for optimising a calibration set of NIR data for estimation of percentage protein. On short run modes, the DGA maintained a higher level of diversity than SGA. In a long run mode the performance of the most fit individual for DGA was better than with SGA. It was also shown that the SGA was sensitive to the setting of mutation probability, with the performance of the population higher with a low value of the mutation probability than with a higher value, suggesting that the SGA was converging to a local minimum. It was found that the predictive residual error sum of squares (PRESS) of the calibration set optimised by this procedure was greatly reduced when compared with the original data set and also when compared with manual removal of the outliers. The DGA was also used for the optimisation of LC-DAD data from a two component mixture of anthracene and phenanthracene. It was found that the predictive quality of the calibration model was greatly improved by the optimised calibration subset.

Wienke et al. (95) used a GA to optimise the slope and shape of the photometric calibration curve of a biochemical dry reagent test strip for the determination of glucose in human urine. The properties of the calibration curve were dependent on twelve different parameters including the concentrations of the different reagents used in the preparation of the strip and the chemical processing techniques of the strip itself, in a theoretically unpredictable fashion. The calibration graph for the study was developed using six different samples involving different concentrations of glucose and different aging conditions. PLS was also used as a means of directly deducing the required variables, it was found that the GA with low resolution predicted optima which were close to those predicted by PLS.

Wehrens et al. (96) developed a self-adapting expert system hybridised with a GA for the interpretation of two dimensional NMR spectra of proteins. A set of known spectra were used to refine and optimise the rules contained within the expert system which were used or the identification of patterns belonging to single amino acids in the protein and also for the classification of the patterns as amino acids or specific groups of amino acids. The GA was then used to construct a sequence of patterns that mapped onto the sequence of amino acids . It was found that for a particular test protein that 77% of all assignments agreed with the true sequence.

Hibbert (97) studied different variants of the SGA used to optimise the rate coefficients for the hydrolysis of adenosine 5'-triphosphate by fitting a kinetic model to concentration Vs. time data by optimising the sum of squares difference between the calculated and measured concentrations of phosphate. The response surface studied had several local minima and a global minimum that was neither symmetrical in the parameter space nor well defined. The kinetic model contained four rate constants which were to be optimised. The genes were encoded as bit strings and also as real numbers, the selection process used was stochastic remainder. Diversity maintenance was investigated using a similarity algorithm in which individuals which are very similar to the rest of the population are penalised, incest prevention was also studied for this purpose. The hybridisation of the GA with pseudo-Newton steepest descent (SDO) was investigated with respect to firstly feeding candidates from the GA into the SDO during training, secondly using a feedback system from the SDO into the GA and allowing cycling during evolution and thirdly running the GA to completion and using the SGA on each member of the resultant population. It was found that the SGA did not produce a good optimum and it also suffered from poor search precision. It was also found that increasing the size of the population improved the final solution but that increasing the number of bits involved in the representation of the rate constants did not improve the final solution.

Using the similarity algorithm improved the final solution but also had an associated time penalty. The prevention of incest improved the quality of the final result and also dramatically reduced the variance between the final solutions after repeated runs of the GA. The use of real coding of the rate constants was found not to make any major change in the final results of the GA. However it ran about 50% faster than the SGA because of the removal of the encoding and decoding stages of the GA.

When hybridising the SGA with the SDO during training, it was found that the better the guess provided by the SGA better the result provided by the SDO. When recycling was used a good estimate of the optimum was quickly found but there was a discretisation error produced by the conversion of the rate constants from the SDO into the bit strings of the GA. The best final solution was found when the SDO was applied to the population produced when the GA had terminated, although strangely it was not the most fit individual in the final population from the GA which produced the best result from the SDO.

It is evident from these examples that the neural network and GA can be used for a wide variety of different chemical applications and it seems likely that as both techniques become increasingly understood that the number of applications investigated will increase. It is in the context of applications to analytical chemistry that this thesis will discuss some of the features of these techniques and their practice.

1.8 Symbol Conventions

1.8.1 Neural Networks

ϕ:	**Transfer function**
λ:	**Scaling constant or gain on the transfer function**
U_i:	**PE Number**
a_i:	**Activity of the PE U_i**
O_i:	**Output of the PE U_i**
W_{ij}:	**Connection weight from unit i to unit j**
η:	**Learning rate**
α:	**Momentum**
ΔW_{ij}:	**Change in weight of connection between PE U_i and PE U_j**
$t_i(t)$:	**Desired output of PE U_i**
$\underline{X}$:	**Input vector $\underline{X}=[x_1,x_2,......x_n]^T \in R^n$**
d_j:	**Euclidean distance metric between two vectors**
σ:	**Width of kernel**
$\underline{Y}$:	**Desired output vector**
F:	**Mapping function $\underline{Y}=F(\underline{X})$**
$\underline{X}'$ and $\underline{Y}'$:	**Approximations to $\underline{X}$ and $\underline{Y}$**
I_i:	**External input to the Hopfield network**
Th_i:	**Threshold for Hopfield network**
$\underline{X}_a$:	**Stable point for Hopfield network**

Ep: **Instantaneous summed squared error**

E: **Cumulative error**

1.9 References

1. W.T. Keeton and J.L. Gould, **Biological Science 4th Edition.** W.W. Norton and Co. Inc. USA 1986
2. G.J. Tortora and R.L. Evans, **Principles of Human Physiology 2nd Edition**, Harper and Row Publishers N.Y. 1986
3. D.S. Levine, **Introduction to Neural And Cognitive Modelling,** Lawrence Erlbaum Associates Inc. N.J., 1991
4. H.Ritter, T. Martinez and K. Schulten, **Neural Computation and Self-Organising Maps**, Addison Wesley Publishing Company, USA 1992
5. G.D. Fischbach. **Mind and Brain**, Scientific American, September 1992, 24-33
6. D.R. Hofstader, **Godel, Escher, Bach: An Eternal Golden Braid,** Penguin Books, U.K. 1979
7. R. Penrose, **The Emporer's New Mind**, Vintage, Oxford University Press 1989
8. F. Crick and C. Asanuma, **Chapter 20, Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex,** J.L. Mc Lelland, D.E. Rumelhart and the PDP Research Group (1986)
9. P.N. Johnson-Laird, **The Computer and the Mind: An Introduction to Cognitive Science**, Fontana Press, London, 1989
10. J.A. Anderson, **IEEE Trans. Syst. Man and Cybern**., SMC-13 (5), 1983, 799-815
11. J.R. Anderson, **Artificial Intelligence**, 40, 1989, 313-351
12. S.J. Hanson and D.J. Burr, **Neural Networks Theory and Applications,** Richard J. Mammone and Yehoshua Y. Zeevi (Editors), Academic Press Inc., 1991.
13. T.J. Sejnowski, C. Koch and P.S. Churchland, **Science**, 241, 1988,1299-1306
14. E.R. Kandel and R.D. Hawkins, **Scientific American**, September 1992, 53-60
15. J.-P. Changeux and S. Dehaene, **Cognition,** 33, 1989, 63-109
16. S. Grossberg, **Cognitive Science,** 11, 1987, 23-63

17. D.E. Rumelhart, G.E. Hinton and J.L. M^{c}Clelland in **Parallel Distributed Processing Volume 1 Chapter 2**, D.E. Rumelhart, J.L. M^{c}Clelland and the PDP Research Group, MIT Press, Cambridge MA 1986

18. B. Kosko, **Neural Networks and Fuzzy Systems**, Prentice -Hall International Inc. N.J. 1992

19. S.J. Hanson and D.J. Burr, **Learning and Representation in Connectionist Networks**, p. 169-201 in Editor R.J. Mammone andY.Y. Zeevi, **Neural Networks: Theory and Applications**, Academic Press Inc., 1991

20. R.P. Lippmann, **IEEE ASSP Magazine,**1987 April, 4-22

21. B.J. Wythoff, **Chemometrics and Intelligent Laboratory Systems,** 18, 1993, 115-155

22. T. Kohonen, **Neural Networks,** 1, 1988, 3-16

23. J.R.M. Smits, W.J. Melssen, L.M.C. Buydens and G. Kateman, **Chemometrics and Intelligent Laboratory Systems,** 22, 1994, 165-189.

24. S.Y. Kung, **Digital Neural Networks**, PTR Prentica Hall N.J., 1993

25. T. Kohonen, **Self-Organisation and Associative Memory 2nd Edition**, Springer Verlag, Germany 1988

26. T. Kohonen, **Biological Cybernetics,** 43, 1982, 59-69

27. T. Kohonen, **Applied Optics,** 26(23), 1987, 4910-4918

28. T. Kohonen, **Proceedings of the IEEE,** 78 (9), 1990, 1464-1480

29. W.J. Melssen, J.R.M. Smits, L.M.C. Buydens and G. Kateman, **Chemometrics and Intelligent Laboratory Systems**, 23, 1994, 267-291

30. F. Crick and G. Mitchison, **Nature,**304, July 1983, 111-114

31. F. Crick and G. Mitchison, **The Journal of Mind and Behavior,** 7(2,3), 1986, 229-250

32. J.J. Hopfield, **Proceedings of the National Academy for Sciences,** 79, 1982, 2554-2558

33. J.J. Hopfield, D.I. Feinstein and R.G. Palmer, **Nature,** 304, 1983, 158-159

34. J.J. Hopfield, **Proceedings of the National Academy of Sciences,** 81, 1984, 3088-3092

35. J.J. Hopfield and D.W. Tank, **Biological Cybernetics,** 52, 1985, 141-152

36. P.D. Wasserman, **Neural Computing Theory and Practice**, Van Nostrand, Reinhold N.Y. 1989

37. B. Widrow and M.A. Lehr, **Proceedings of the IEEE,** 78 (9), 1990, 1415-1441

38. J.A. Freeman,and D.M. Skapura, **Neural Networks, Algorithms, Applications and Programming Techniques**. Addison Wesley, Reading Mass., 1991

39. D.E. Rumelhart, G.E, Hinton and R.J. Williams, **Nature,** 323, 1986, 533-536

40. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Chapter 8 (318-362), **Parallel Distributed Processing,** J.L. M^{C}Clelland, D.E. Rumelhart and the PDP Research Group, MIT Press, Cambridge MA, 1986

41. K. Hornik, M. Stinchcombe and H. White, **Neural Networks**, 2, 1989, 359-366

42. A. Bos, **Artificial Neural Networks as a Tool in Chemometrics**, Ph. D. Thesis, University of Twente, Enschede, 1993.

43. J.H. Holland, **Adaptation in Natural and Artificial Systems**, MIT Press, Bradford Books Edition, USA, 1992

44. K.A. De Jong, **Are Genetic Algorithms Function Optimizers ?**, in R. Männer and B. Manderick (Editors), **Parallel Problem Solving From Nature 2**, Elsevier Publishers B.V., Amsterdam,1992, p. 3-15.

45. J.C. Nash, **Compact Numerical Methods for Computers 2nd Edition**, Adam Hilgen, Bristol, 1990.

46. C.B. Lucasius, Towards Genetic Algorithm Methodology in Chemometrics, Ph. D. thesis, Katholieke Universiteit Nijmegen, Netherlands, 1993.

47. D.E. Goldberg, **Genetic Algorithms in Search, Optimization and Machine Learning,** Addison-Wesley, Reading, MA., 1989.

48. Editors J.D. Becker, I. Eisele and F.W. Mundemann, **Parallelism, Learning, Evolution,** Workshop on evolutionary models and strategies, Germany 1989. Workshop on parallel processing: logic, organisation and technology, WOPPLOT 89, Germany 1989. Springer Verlag, 1991.

49. W.H. Press, S.A. Teukolsky, B.P. Flannery and W.T. Vetterling, **Numerical Recipes in C : The Art of Scientific Computing,** Cambridge University Press, 1991.
    (a) System supplied random number generators p. 206
    (b) Knuth's subtractive method p.212-213

50. K. De Jong, **Machine Learning** 3 (1988) 121-138.

51. G.E. Liepins and M.R. Hilliard, **Annals of Operations Research** 21 (1989) 31-58.

52. G.E. Liepins and M.D. Vose, **J.Expt. Theor. Artif. Intell.** 2 (1990) 101-115.

53. J. J. Grefenstette, **IEEE Trans. Syst. Man Cybern.** SMC-16 (1) (1986) 122-128.

54. D.B. Hibbert, **Chemom. Intell. Lab. Syst,** 19 (1993) 319-329

55. Editor L. Davis, **Handbook of Genetic Algorithms,** Van Nostrand Reinhold, New York, 1991.

56. N. Qian and T.J. Sejnowski. **J. Mol. Biol.,** 202, 1988, 865-884

57. D.G. Keller, F.E. Cohen and R. Langridge, **Journal Of Molecular Biology,** 214, 1990, 171-182

58. J.U. Thomsen and B. Meyer, **Journal Of Magnetic Resonance,** 84, 1989, 212-217

59. M. Bos, A. Bos and W.E. Van der Linden, **Analytica Chimica Acta,** 233, 1990, 31-39

60. T. Aoyama and H. Ichikawa, **Chem. Pharm. Bull.,** 39 (2),1991, 358-366

61. T. Aoyama and H. Ichikawa, **Chem. Pharm. Bull.,** 39(2), 1991, 372-378

62. E.W. Robb and M.E. Munk, **Mikrochim. Acta,** I, 1990, 131-155

63. M.E. Munk, M.S. Madison and E.W. Robb, **Mikrochim. Acta,** II, 1991, 505-514

64. B.J. Wythoff, S.P. Levine and S.A. Tomellini, **Analytical Chemistry,** 62, 1990, 2702-2709

65. J.R. Long, V.G. Gregoriou and P.J. Gemperline, **Analytical Chemistry**, 62, 1990, 1791-1797

66. J.W. Gardner, E.L. Hines and M. Wilkinson, **Meas. Sci. Technol.,** 1, 1990, 446-451

67. B. Curry and D.E. Rumelhart, **Teterahedron Computer Methodology,** 3 (3/4), 1990, 213-237

68. A.Bos, M.Bos and W.E. Van Der Linden, **Analytica Chimica Acta**, 256, 1992, 133-144

69. T. Nakamoto, K. Fukunishi and T. Moriizumi, **Sensors and Actuators**, B 1, 1990, 473-476

70. T. Nakamoto, A. Fukada, T. Moriizumi and Y. Asakura, **Sensors and Actuators**, B 3, 1991, 221-226

71. T. Nakamoto, A. Fukada and T. Moriizumi, **Sensors and Actuators**, 10, 1993, 85-90

72. D.W. Elrod, G.M. Maggiora and R.G. Trenary, **Journal of Chem. Inf. Comput. Sci.,** 30, 1990, 477-484

73. S.-M. Chang, Y. Iwasaki, M. Suzuki, E.Tamiya, I. Karube and H.Muramatsu, **Analytica Chimica Acta**, 249, 1991, 323-329

74. M. Glick and G.M. Hieftje, **Applied Spectroscopy**, 45(10), 1991, 1706-1716

75. H. Sundgren, F. Winquist, I. Lukkari and I. Lundstrom, **Meas. Sci. Technol**., 2, 1991, 464-469

76. B. Meyer, T. Hansen, D. Nute, P. Albersheim, A. Darvill, W. York and J. Sellers, **Science**, 251, 1991, 542-544

77. M.Bos and H.T. Weber, **Analytica Chimica Acta**, 247, 1991, 97-105

78. J.R. Long, H.T. Mayfield, M.V. Henley and P. R. Kromann, **Analytical Chemistry**, 63, 1991, 1256-1261

79. P.J.Gemperline, J.R.Long and V.G. Gregoriou, **Analytical Chemistry**, 63**,** 1991, 2313-2323

80. L.S. Anker and P.C. Jurs, **Analytical Chemistry**, 64, 1992, 1157-1164

81. J.R.M. Smits, L.W. Breedveld, M.W.J. Derksen, G. Kateman, H.W. Balfoort, J. Snoek and J.W. Hofstraat, **Analytica Chimica Acta**, 258, 1992, 11-25

82. A.P. de Weijer, L. Buydens, G.Kateman and H.M. Heuvel, **Chemometrics and Intelligent Laboratory Systems**, 16, 1992, 77-86

83. X.-H. Song, Z. Chen and R.-Q. Yu, **Chemometrics and Intelligent Laboratory Systems**, 16, 1992, 213-219

84. A. Bos, M. Bos and W.E. van der Linden, **Analytica Chimica Acta**, 277, 1993, 289-295

85. X.-H. Song and R.-Q. Yu, **Chemometrics and Intelligent Laboratory Systems,** 19, 1993, 101-104

86. Z. Li, Z. Cheng, L. Xu and T. Li, **Analytical Chemistry**, 65, 1993, 393-396

87. J.R.M. Smits, P. Schoenmakers, A. Stehmann, F. Sijstermans and G. Kateman, **Chemometrics and Intelligent Laboratory Systems**, 18, 1993, 27-39

88. X.-L. Wu and R. Freeman, **Journal of Magnetic Resonance**, 85, 1989, 414-420

89. C.B. Lucasius and G. Kateman, **Trends in Analytical Chemistry**, 10 (8), 1991, 254-261

90. C.B. Lucasius, M.J.J. Blommers, L.M.C. Buydens and G. Kateman, Chapter 18, pages 251-281, in **Handbook of Genetic Algorithms**, Editor L. Davis, Van Nostrand Reinhold, N.Y. 1991

91. M.J.J. Blommers, C.B. Lucasius, G. Kateman and R. Kaptein, **Biopolymers**, 32, 1992, 45-52

92. E. Fontain, **Analytica Chimica Acta,** 265, 1992, 227-232

93. D. Wienke, C. Lucasius and G. Kateman, **Analytica Chimica Acta**, 265**,** 1992, 211-225

94. T.-H. Li, C.B. Lucasius and G. Kateman, **Analytica Chimica Acta**, 268, 1992, 123-134

95. D. Wienke, C. Lucasius, M.Ehrlich and G. Kateman, **Analytica Chimica Acta**, 271, 1993, 253-268

96. R. Wehrens, C. Lucasius, L. Buydens and G. Kateman, **Analytica Chimica Acta**, 277, 1993, 313-324

97. B.B. Hibbert, **Chemometrics and Intelligent laboratory Systems**, 19, 1993, 319-329

Chapter 2: Neural Network Application to Pattern Recognition of Potentiometric Flow Injection Analysis Peaks

Abstract

This chapter is a discussion of the application of backpropagation neural networks to the detection and identification of waveforms produced by the transient responses of electrodes selective to sodium, potassium and calcium ions injected into a flowing stream using the technique known as flow injection analysis (FIA).

The ability of the network to classify noisy data is studied by deliberately distorting the waveforms by a variety of mechanisms sometimes associated with potentiometric detection. In addition an investigation of processes occurring within the network during training is performed by studying the variation of the lengths of the weight vectors and the angles which they make to a vector with unit co-ordinates.

2.1 Introduction

This section provides the context for this chapter by discussing some of the basic principles of FIA and potentiometry.

2.1.1 Flow injection analysis

With increased public awareness and concern about health, food and environmental issues, there has been a concomitantly dramatic increase in the volume of routine analyses performed by laboratories. The problem of producing large volumes of accurate and precise data motivated the design and development of rapid and inexpensive automatic analytical systems, which are now employed in a diverse range of applications including the monitoring and control of industrial processes, diagnostic and screening clinical tests, environmental monitoring of a wide variety of species in the air, soil and water and quality control in the food processing and pharmaceutical industries.

Automated analytical instruments can be broadly classified as being either discrete or continuous, although instruments which are hybrids of the two classes also exist. Discrete or segmented flow instruments simulate the operations which would be carried out in the manual version of a particular analysis. Samples are retained in discrete vessels throughout the various analytical operations such as dilution, reagent addition and mixing, leading to the final measurement. In contrast, in a continuous flow instrument such as FIA which is depicted in figure 2.1, the sample becomes a plug in a flowing stream and is carried from the injection port to the detector and then to waste.

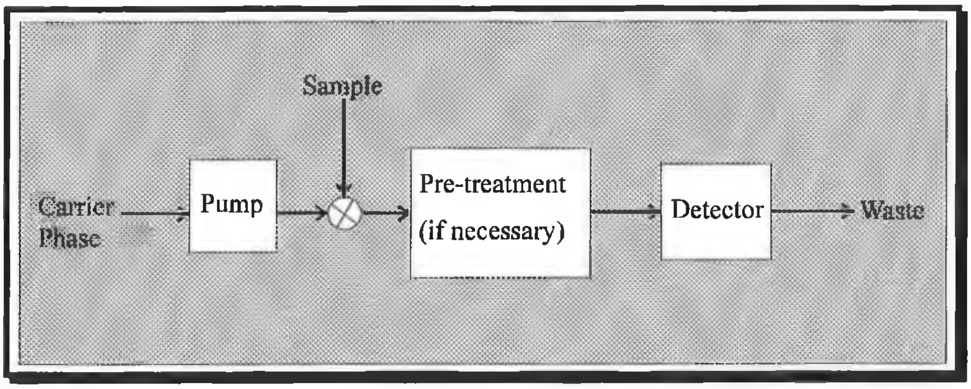


Figure 2.1 - Schematic of an apparatus used for flow injection analysis

A wide variety of different detectors and operating conditions such as carrier phase composition, mixing reagents and flow rate have provided the means to analyse for a diverse range of species. Some applications include: spectrophotometric detection of phosphate and chloride in blood sera (1) and glycerol in water (2), kinetic determination of glucose in blood sera (3), and potentiometric determination of chloride in tap and sewage water (4), fluoride in tap water, beverages and urine (5) and sodium, potassium and calcium in mineral water and plasma (6). For further information concerning the theory and implementation of flow injection analysis the reader is recommended the texts by Ruzicka and Hansen (7) and Valcarel and Luque de Castro (8).

2.1.2 Potentiometry

Electrochemical techniques can be broadly divided into voltammetric and potentiometric methods. Potentiometry is a technique which involves the measurement under zero current conditions of potentials generated by electrochemical cells. One half of the cell is an ion-selective electrode (ISE) which generates a potential related to the activity of the ion of interest (primary ion). Changes in the potential are measured against that of a reference electrode which is also in contact with the sample solution. Ideally the reference electrode potential is unaffected by changes in the sample composition. The basic apparatus used for a potentiometric measurement is depicted in figure 2.2.

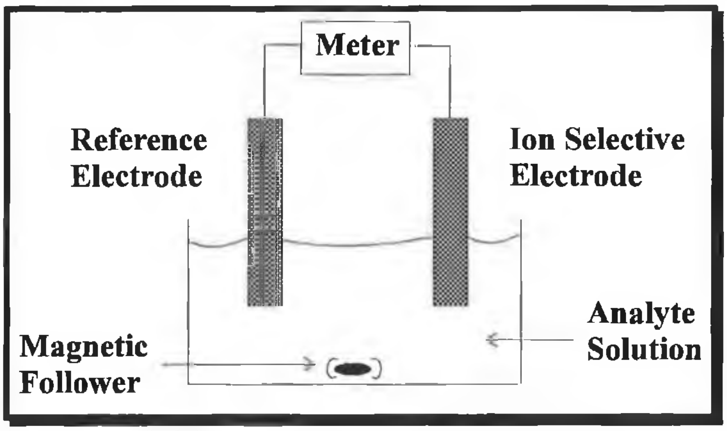


Figure 2.2 - Diagrammatic representation of a basic experimental system used for performing potentiometric measurements

The potential measured between an ideal electrode and the reference electrode is mathematically described by the Nernst equation (equation **2.1**).

$$\mathbf{E = E^0 + S\log_{10} a_i} \qquad \textbf{(2.1)}$$

Where **E** is the measured cell potential, **E⁰** is the standard cell potential, **S** is the slope of the electrode, which has a value of 59.2 mV divided by the charge of the species involved in the electrochemical process, when measurements are performed at 25°C. The variable $\mathbf{a_i}$ in equation **2.1**, describes the activity of the ion being measured. Activity is a thermodynamic quantity (9) related to the concentration of the ion **($\mathbf{c_i}$)** in a manner described by equation **2.2**.

$$\mathbf{a_i = \gamma_i c_i}. \qquad \textbf{(2.2)}$$

γ_i in equation **2.2** is a term known as the activity coefficient which may be calculated by means of a number of expressions including the Davis equation (10) depicted in equation **2.3**.

$$\mathbf{\log\gamma_i = -0.5z_i^2\left(\frac{\sqrt{I}}{1+\sqrt{I}} - 0.2I\right)} \qquad \textbf{(2.3)}$$

The term $\mathbf{z_i^2}$ in equation **2.3** refers to the charge of the ion and **I** refers to the ionic strength of the solution containing the ion. The ionic strength itself can be calculated from the concentrations of all the ions present in the analyte solution by means of equation **2.4**.

$$\mathbf{I = 1/2\sum_{i=1}^{n} c_i z_i^2} \qquad \textbf{(2.4)}$$

Where $\mathbf{c_i}$ and $\mathbf{z_i}$ refer to the concentration and charge of any ion **i** in the solution. Unfortunately electrodes do not respond to the primary ion alone, i.e. the electrodes are selective but not specific. The additional effects of other ions present in the system are accounted for by the Nikolskii-Eisenman expression.

$$\mathbf{E = E^o + S\log_{10}\left(a_i + \sum_j k_{ij}^{pot} a_j^{Z_i/Z_j}\right)} \qquad \textbf{(2.5)}$$

The variables **E, E$^{\mathbf{O}}$, S** and $\mathbf{a_i}$ in equation **2.5** have the same meaning as they did in equation **2.1**. $\mathbf{k_{ij}^{pot}}$ is a weighting factor or selectivity coefficient which describes the selectivity of an electrode for its primary ion relative to the other ions present in the system. Because potentiometry is a high impedance technique (the resistance of freshly made electrodes of the kind used in this study [PVC/liquid membrane electrodes] is typically 0.5 MΩ [for bench type electrodes] and the resistance gradually increases with use (11)), it is particularly prone to the uptake of noise from its surroundings. Hence all the cables used in potentiometric experiments are generally shielded to reduce the effects of noise, but there can still be problems with inductive and RF noise and static. Other sources of error for this technique include parallel drift of the electrode potential (12), this is a shift in the calibration line for an electrode system in which the slope of the line does not change. This may be caused by a number of reasons including difficulties with the liquid junction between the reference half cell and the sample solution. References (13-14) are recommended for further reading about the theory and application of potentiometric analysis.

2.1.3 Potentiometric Flow Injection Analysis

Figure 2.3 depicts a typical trace obtained from FIA apparatus using three ion-selective electrodes for the detection of sodium, potassium and calcium ions. The traces represent the potential differences measured between the different ion selective electrodes and the reference electrode over a period of time in which a sample containing these three ions is injected into the carrier phase and is swept past the electrodes by the carrier phase to waste.

The traces show a rapid rise as the sample reaches the electrodes and then a more gradual descent as the tail of the sample passes the electrodes on its way to waste. The duration of a particular peak is dependent on several variables including the volume of the sample injected into the flowing stream, the concentration of the carrier phase, the composition of the sample, the flow rate and dispersion characteristics of the FIA systems and the response characteristics of the sensing membrane, but the typical duration of a peak would be 60 seconds or less for an optimised system. The potential difference measured between the reference and each ion selective electrode is acquired and digitised on separate channels of a data acquisition card fitted inside a computer. A voltage range can then be defined allowing the digitised values corresponding to the responses of the electrodes at different times during the analysis to be prescaled for further processing.

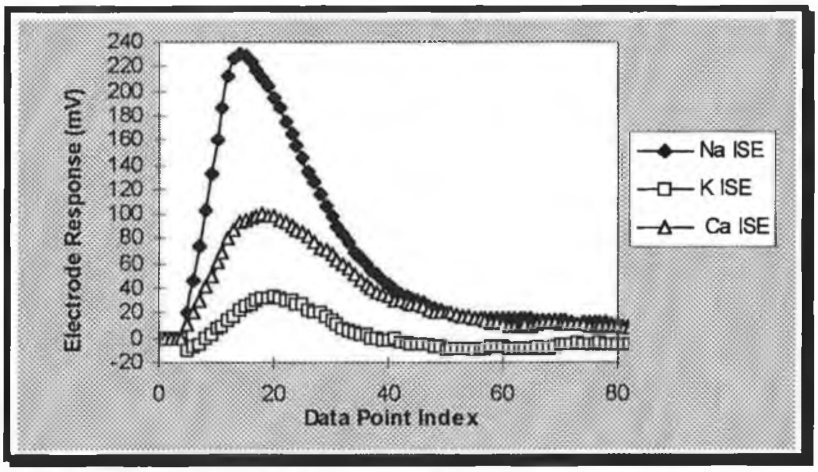


Figure 2.3 - Typical responses of sodium, potassium and calcium ISEs to a sample containing 0.1M sodium, potassium and calcium, injected into an FIA system at a flow rate of 1 ml/minute.

2.2 Experimental Method

The continuous flow system used for transient ion is as described by Forster and Diamond (6). The electrodes used for the detection of sodium, potassium and calcium were based on three ionophores (p-t-butylcalix[4]methyl acetate (15) valinomycin (Fluka) and ETH 129 (Fluka), respectively), each of which were immobilised within a plasticised PVC matrix (15). Membranes were cast from solutions of the ionophore, ion excluder, plasticiser and PVC in tetrahydrofuran (THF). The membranes were then mounted in the electrode block which was reconnected to the rest of the continuous flow system.

The experimental data were captured via an Analog Devices RTI-815 data acquisition card fitted inside an IBM 286 compatible PC. Data acquisition and processing software was written in Microsoft QuickBASIC . The FIA patterns were provided courtesy of Mr.F.J. Sáez de Viteri, School of Chemical Sciences, Dublin City University.

The FIA patterns used in this study were acquired when 200µl samples of the solutions (whose composition is described in table 2.1) were injected into the described FIA system operating at a flow rate of 1 ml/min. The key for each solution in table 2.1 will be used to refer to these solutions in further discussions. The entry NONE at the bottom of the solution key column refers to a synthetic profile corresponding to the absence of any of the cations

Solution Key	Sodium Ion Concentration (M)	Potassium Ion Concentration (M)	Calcium Ion Concentration
Na	0.1	0.0	0.0
K	0.0	0.1	0.0
Ca	0.0	0.0	0.1
NaK	0.1	0.1	0.0
NaCa	0.1	0.0	0.1
KCa	0.0	0.1	0.1
NaKCa	0.1	0.1	0.1
NONE	0.0	0.0	0.0

Table 2.1 - Compositions of the sample solutions used for the studies in this chapter

The term binary solution will be used to refer to a solution containing two ions namely solutions NaK, NaCa and KCa in table 2.1, the term tertiary solution will be used to refer to the solution NaKCa in table 2.1, containing all three ions.

Two types of neural network software were investigated namely, Neural Technologies Ltd. NT5000 running on an 80486 PC and NeuralWorks Professional II running on a SUN SPARC workstation. Data preparation for the neural networks and results analyses from the neural nets were performed using software written in Borland Turbo C and SUN C version 1.1.

2.3 Results

The subject of the investigations discussed in this chapter is the application of feedforward neural networks to the problem of classifying patterns produced by a series of ISEs used in a flow injection analysis regime. To this end, different aspects of the use of feedforward networks to this specific problem were studied. These studies included :

(i) An investigation of the effect of the learning rate and momentum parameters (see chapter 1) on the rate at which a neural network would learn how to correctly classify simple FIA patterns (as depicted in appendix 1).

(ii) An investigation of how a network trained on these simple FIA patterns would perform when confronted with these patterns when distorted by noise addition, baseline shifting and variation of the heights of the peaks in the FIA traces. This study was devised in order to investigate the ability of a network to generalise from the simple patterns on which it had been trained to the distorted patterns.

(iii) The redesign of the training and testing sets for a network based on conclusions drawn from the previous study.

(iv) An investigation of how a network trained with this new training set would perform when confronted with the same patterns as used in study (ii).

(v) An investigation of some of the processes occurring within a network as it is training.

Studies (i) to (iv) were performed with the NT5000 software and study (v) was performed with the NeuralWorks software. Study (i) was performed with the FIA traces from the solutions described in table 2.1. Studies (ii) to (v) were performed with patterns which were derived from these FIA traces and distorted as described above. A three layer network topology (as depicted in figure 2.4) was used for all the studies.

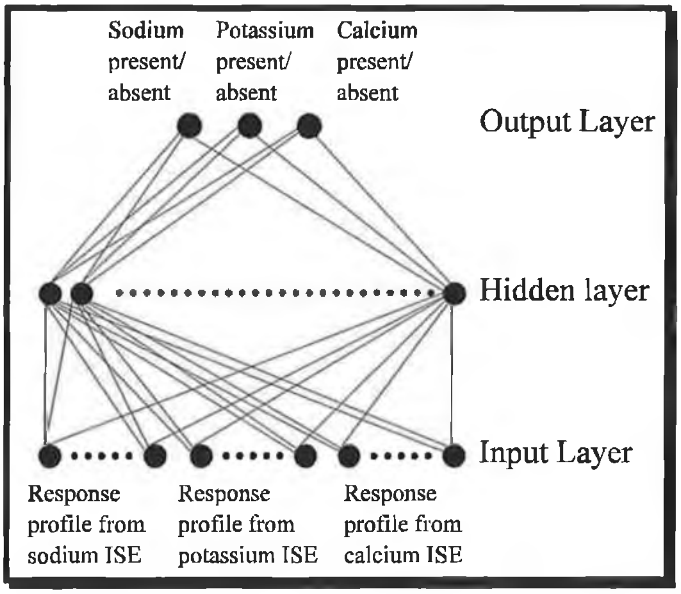

Sodium present/ absent
Potassium present/ absent
Calcium present/ absent
Output Layer
Hidden layer
Input Layer
Response profile from sodium ISE
Response profile from potassium ISE
Response profile from calcium ISE

Figure 2.4 - A diagrammatic representation of the three layer feedforward networks used for the studies discussed in this chapter

The data presented to the input layer of the networks corresponded to the simple (or distorted) electrode response profiles to the solutions described in table 2.1 (i.e. the 7 possible combinations of the cations and the synthetic profile of the absence of any cations (key NONE in table 2.1)). The data files describing the responses of the electrodes were composed of 80 data points per electrode (240 data points in total corresponding to 240 neurons in the input layer). The task of the network was to make a decision concerning whether or not a particular ion was present in the solution to which the ISEs produced the response profiles, which were presented to the input layer. The target outputs for the network, specifying the desired classification of the FIA patterns were binary numbers referring to the presence (described by a 1) or the absence (described by a 0) of a particular ion, as seen in table 2.2

Solution Key	Desired Output For The Neuron Describing The Absence/Presence Of The Sodium Ion	Desired Output For The Neuron Describing The Absence/Presence Of The Potassium Ion	Desired Output For The Neuron Describing The Absence/Presence Of The Calcium Ion
Na	1	0	0
K	0	1	0
Ca	0	0	1
NaK	1	1	0
NaCa	1	0	1
KCa	0	1	1
NaKCa	1	1	1
NONE	0	0	0

Table 2. 2- Desired outputs of neurons to solutions described in table 2.1.

Three neurons were used in the output layer of a network. The outputs of these units represented the network's decision concerning the presence or absence of a specific ion. To simplify the discussion, the output neurons whose outputs represent the result of the classification concerning sodium, potassium and calcium will be called the Na, K and Ca neurons respectively (i.e. if the Na neuron produces an output of 0 then the network has decides that sodium is not present). To simplify the discussions in studies (ii) and (iv), the classification decisions of the network of the network trained with the simple FIA patterns will be differentiated from those made by the network trained with the distorted patterns, by referring to the former as restricted training (rt) and the latter as broad training (bt). As such, the classification decisions of the network using restricted training will be discriminated from those of the network using broad training by means of a subscript bt on the output of the Na, K and Ca neurons of networks using broad training and rt for the corresponding neurons in the networks using restricted training (i.e. $Na_{(rt)}$ =1, indicates that a network using restricted training has decided the sodium is present)

2.3.1. Initial Studies Using The NT5000 software

2.3.1.1 A study of the effect of learning rate and momentum on the learning speed of a network trained for the recognition of simple FIA patterns.

This study was performed with the FIA traces from the solutions described in table 2.1, the patterns are depicted in appendix 1. The potentials of the electrodes recorded in the FIA traces were linearly prescaled to have a magnitude similar to the desired output values of the network (16).

Training in the NT5000 is terminated according to two criterion (a) an upper limit on the number of epochs for training (b) a lower limit on the maximum output error. The upper limit on the number of epochs for training means that learning will stop if a certain number of epochs have been exceeded (an epoch is a cycle of the backpropagation algorithm in which all the training patterns are presented to the network). While the desired output classifications for the patterns are binary the actual outputs from the neurons in the output layer which employ a sigmoid transfer function are continuous. This is similar to the principle of fuzziness, which mathematically means multi-valuedness or multivalence. Three valued fuzziness in this study would mean the presence (output =1), absence (output =0) or ambiguity

($0 < \text{output} < 1$) (concerning the presence or absence of a particular species).

Mathematically a fuzzy set is described as follows;

if **X** is a collection of objects **x**, then a fuzzy set **A** in **X** is a set of ordered pairs such that $\mathbf{A} = \{(\mathbf{x},\mu_A(\mathbf{x}))|\mathbf{x}\in\mathbf{X}\}$. The term $\mu_A(\mathbf{x})$ is called the membership function the value of which describes the degree to which the element **x** belongs to the set **A**. This value also describes the degree to which the measurement **x** is compatible with the concept of **A** (e.g. the degree to which the measured pattern from the ISEs is compatible with the concept of the presence of a particular species). If $\mu_A(\mathbf{x})$ has only two values, 0 and 1, then **A** is not fuzzy (17-18).

The maximum output error refers to an error calculated from the difference between the continuous outputs of neurons in the output layer of the network and the binary target output for those neurons specified in the training file. Training will not stop until the outputs from the neurons in the output layer of the network agree with their target values (as specified in the training file) to within the maximum output error. The maximum output error was set to a default value of 0.1. The effect of choosing different termination criterion will be described more fully in the discussion section of this thesis. Since the desired outputs of the network were had two possible values (0 and 1) the value of 0.1 for the maximum output error represents a 10% difference between the desired and actual outputs of each neuron in the output layer of a network. The number of epochs needed for the maximum output error to converge to the desired termination levels was investigated with respect to variation of the learning rate (η in equation **1.27**) and momentum (α in equation **1.48**) by means of a 6 level 2 factor full factorial experimental design. The network topology used for this study employed seven neurons in the hidden layer. The rationale behind the initial choice of seven neurons for this study was based on an initial reading of the Kolmogorov network existence theorem (19). The theorem was misinterpreted as meaning that any mapping from R^n to R^m could be represented by means of a network with 2m +1 hidden layer neurons (2*3(number of output neurons) +1 = 7). In actuality, the theorem says that for any continuous mapping from R^n to R^m there must exist a three layer network with a hidden layer of (2n+1) neurons. However, the use of this theorem itself is controversial (see the discussion section of this thesis). Table 2.3 depicts the results of this study. In those cases where the table entries read as no convergence it means that the network did not converge to the required levels of the maximum output error within a limit of 1000 epochs.

Unfortunately, it was not possible to obtain ranges for the number of iterations until convergence. Repetitions of training under the same conditions of learning rate, momentum and network topology yielded the same number of iterations until convergence. If training was stopped (prematurely) after a fixed number of iterations, the studied connection weights had the same values when training was repeated. It would be beneficial to determine whether this is due to the initialisation of the weights in a network prior to training.

It can be seen that at a very high momentum of 2.0, the network did not converge to the desired levels of maximum output error within 1000 epochs. This suggests that using a momentum greater than one causes the weight vector (the vector describing the weights of the connections to a neuron) to move during training on the error surface (the surface describing the error for the neuron for different values of the connection weights) under greater influence from the previous gradient than from the current gradient for a given step, leading to unstable behaviour of the network (see **equation 1.48**). Strangely however, at a slightly lower momentum of 1.5 the network converged with very low learning rates i.e. $0.05 \leq \eta \leq 0.25$. This suggests that low learning rates stabilise the learning process when used with high momentum. However, having a low learning rate slows down the rate of learning so that a compromise between having stable learning (associated with low learning rates and high momentum values) and speeding up learning with a slightly higher learning rate is achieved with the learning rate of 0.1. At a lower momentum of 1.25, a higher learning rate of 0.25 provides the optimal rate of convergence for the network, although the number of epochs required for convergence is slightly higher than with the learning rate of 0.1 and the momentum of 1.5.

Momentum	Learning Rate	Number of Iterations Until Convergence
2	1.0	Not Converged
2	0.75	Not Converged
2	0.5	Not Converged
2	0.25	Not Converged
2	0.1	Not Converged
2	0.05	Not Converged
1.5	1.0	Not Converged
1.5	0.75	Not Converged
1.5	0.5	Not Converged
1.5	0.25	125
1.5	0.1	94
1.5	0.05	132
1.25	1.0	Not Converged
1.25	0.75	Not Converged
1.25	0.5	Not Converged
1.25	0.25	100
1.25	0.1	152
1.25	0.05	381
1.0	1.0	Not Converged
1.0	0.75	Not Converged
1.0	0.5	140
1.0	0.25	84
1.0	0.1	98
1.0	0.05	98
0.75	1.0	Not Converged
0.75	0.75	Not Converged
0.75	0.5	74
0.75	0.25	85
0.75	0.1	116
0.75	0.05	115
0.5	1.0	Not Converged
0.5	0.75	56
0.5	0.5	82
0.5	0.25	106
0.5	0.1	129
0.5	0.05	135
0.25	1.0	Not Converged
0.25	0.75	79
0.25	0.5	137
0.25	0.25	188
0.25	0.1	221
0.25	0.05	240
0.1	1.0	Not Converged
0.1	0.75	168
0.1	0.5	325
0.1	0.25	455
0.1	0.1	548
0.1	0.05	578
0.05	1.0	Not Converged
0.05	0.75	319
0.05	0.5	609
0.05	0.25	919
0.05	0.1	Not Converged
0.05	0.05	Not Converged

Table 2.3- Variation of number of epochs of a network with seven neurons in the hidden layer, trained with the eight FIA patterns depicted in appendix 1 to converge for different learning rate and momentum values

However it is difficult to ascertain the reason for this difference because the momentum is itself dependent on the learning rate (16). The previous statements concerning the use of low learning rates with high momentum values agree with the conclusions derived by Tollenaere (20) and described by Bos (16), that high momentum rates do not make learning unstable if small enough learning rates are used. However in Bos's thesis, this related to momentum values up to 0.9. It is interesting that the same effect is seen in this case for momentum values greater than 0.9 at which learning should be unstable. The software was tested in a similar fashion on a different problem and demonstrated again the ability of the network to converge at momentum values greater than one. It would be of benefit to determine whether this behaviour is a feature of the software rather than the problem being solved.

It can also be seen from table 2.3, that the learning rate with the lowest number of epochs to convergence of the network decreased as the momentum increased (see table 2.4). This also agrees with the conclusions of Tollenaere (20) described by Bos (16).

Momentum	Learning Rate With The Lowest Number Of Epochs Until Convergence
0.05	0.75
0.1	0.75
0.25	0.75
0.5	0.75
0.75	0.5
1.0	0.25
1.25	0.25
1.5	0.1

Table 2.4 - Variation of the learning rate with the lowest number of epochs until convergence (determined from table 2.3) of the network with 7 neurons in its hidden layer applied to the classification of the patterns produced from the response of the sodium, potassium and calcium ISEs to the solutions whose composition is given in table 2.1.

It can be noticed from table 2.3 that the network did not converge to the required level of maximum output error within the required number of iterations for any value of the momentum when the learning rate is fixed at 1.0. This suggests that training with such a large learning rate on the patterns in appendix 1 was so unstable that the network could not converge to the required level irrespective of the stabilisation provided by the momentum. It can be seen table 2.3 that an apparent minimum in the number of epochs (presentations of the training set) required for the network to converge to the required level of maximum output error exists for a learning rate coefficient in the region of 0.5 to 0.75 and a momentum value of 0.25 to 0.75 . The optimal conditions found (i.e. lowest number of epochs required to convergence) were a learning rate equal to 0.75 and momentum equal to 0.5. Since further training would involve more noisy patterns, it was decided to use a slightly higher momentum of 0.75 (to provide stabilisation) and a lower learning rate of 0.5 as this would provide stabilisation without unduly affecting the performance.

2.3.1.2 A study of the effect of the number of neurons in the hidden layer of a network on the number of epochs until convergence and the computational time involved in training until convergence

Figure 2.5 and 2.6 shows how the number epochs and computational time required for networks (trained with a learning rate of 0.5 and a momentum of 0.75) to converge to the same termination criterion as used for the previous study (maximum output error = 0.1) varies with respect to the number of neurons used in the hidden layer of the networks . The training set used for this study is the same as used for the previous study (the FIA traces from the solutions described in table 2.1 and the synthetic trace describing the absence of the ions). As with the previous study, it was unfortunately not possible to obtain ranges for the values of the number of epochs and computational time by repeating the experiments. Without these ranges it is difficult to objectively describe the significance of the observed trends and as such, the following discussion will be of mainly descriptive value. Figure 2.5 shows a gradual decrease in the number of presentations of the training set required for network to converge with increasing number of neurons in the hidden layer. As is discussed by Bos, while a larger number of neurons in the hidden layer increase the number of computations that must be performed by the backpropagation algorithm (because there are more connection weights) increasing the number of neurons in the hidden layer may lead to faster convergence rates because of the larger number of weight adaptations per epoch.

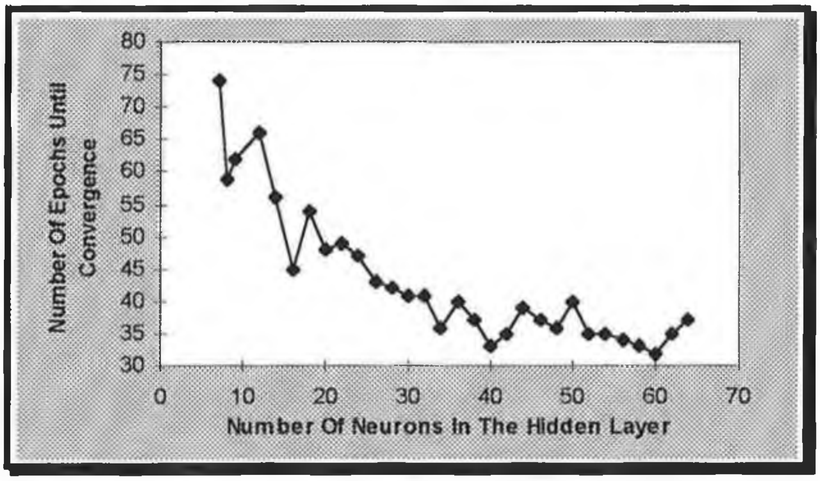


Figure 2.5 - Variation of the number of presentations of the training set (epoch) to networks with varying number of neurons in their hidden layers until convergence to a maximum output error of 0.1. The training set was composed of the FIA traces of the sodium, potassium and calcium ISEs to the solutions described in table 2.1. The learning rate was 0.5 and the momentum was 0.75

The computational time required for convergence of the networks depicted in figure 2.6 is dependent on both the software and the hardware used for the study but it can be seen that there is a general increase in the computational time required for convergence as the number of neurons in the hidden layers increase. Increasing the number of neurons in the hidden layer from 20 to 60 leads to a reduction in the number of epochs required for convergence from 48 to 32 but results in an increase in the computational time required for convergence from 27 to 52 seconds. It can be seen that an approximately three-fold increase in the number of connection weights leads to an approximately two-fold increase in computational time, suggesting that the larger number of weight adaptations per epoch partly compensates for the associated increase in the number of computations per epoch. These results agree with those quoted by Bos (16) when he discussed the variation in number of epochs required for convergence and computational time as a function of the number of neurons in the hidden layer of a network. Bos (16) also showed that the number of epochs and computational time required for convergence also varies widely depending on the learning rate, indicating that the final performance of a network on a training set is dependent on the learning rate and the number of epochs chosen as a termination criterion for training and is also dependent on the number of neurons in the hidden layer of the network.

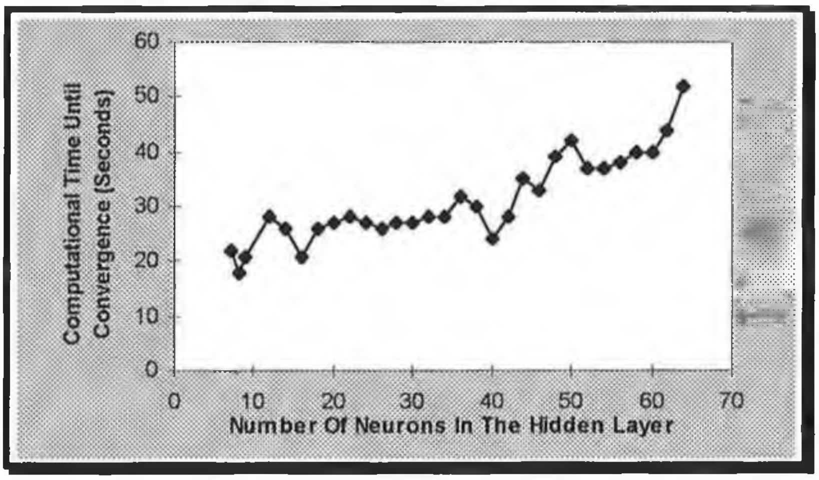


Figure 2.6- Variation of the computational time of networks with varying number of neurons in their hidden layers until convergence to a maximum output error of 0.1. The training set was composed of the FIA traces of the sodium, potassium and calcium ISEs to the solutions described in table 2.1. The learning rate was 0.5 and the momentum was 0.75

This suggests that the balance achieved between the increased computational time and reduced number of epochs required for training large networks should be considered when deciding upon a particular network topology. However, the performance of a network should also consider its ability to generalise outside its training set and as such the possibility of overfitting by large networks should not be disregarded.

Observations of overtraining will be described in more detail in sections 2.3.5.1, the specific problems of overtraining as it relates to this study will be described in the conclusion and a more general view of overtraining will be described in the discussion section of this thesis.

In order to investigate the effect of various distortions on the ability of the networks to correctly predict qualitatively the composition of an unknown solution, test sets of distortions of response profiles in the training set were composed. A mean squared error (MSE) described in equation **2.6** was used as a measure of the performance of the networks.

$$\mathbf{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{m}\left(y_{ij}-o_{ij}\right)^{2}\right] \qquad \textbf{(2.6)}$$

$\mathbf{y_{ij}}$ in this expression refers to the jth component of the desired output to pattern i, $\mathbf{o_{ij}}$ refers to the actual output of the neuron j to the pattern i, **m** refers to the number of components in the vector describing the desired output of the network to a given pattern (or the number of neurons in the output layer of the network) and **n** refers to the number of patterns involved in the testing of the network.

For the studies in sections 2.3.1.3 to 2.3.3, networks with 50 to 55 neurons in the hidden layer were used. The choice of this topology arose from the study depicted in figure 2.5, which suggested that networks with this topology yielded the lowest number of epochs until convergence.

2.3.1.3 Classification Of Test Patterns With Varying Noise Levels

An investigation was performed into how networks trained with the FIA patterns depicted in appendix 1 would perform when tested with patterns derived from these FIA traces by the addition of variable amounts of noise. Noise was added by using the system supplied random number generator to produce numbers in the ranges of 0 to 20%, 50%, 100% and 150%, 200%, 300% and 400%. The resulting random numbers were then converted positive and negative numbers in the ranges +10% to -10%, +25% to -25%, +50% to -50%, +75% to -75%, +100% to -100%, +150% to -150% and +200% to -200% of the maximum peak height in each pattern composed of the responses of the sodium, potassium and calcium ISEs in the training set. These random numbers were then added to FIA traces to the solutions described in table 2.1. The resulting patterns are depicted in appendix 2. These patterns were then presented to networks with variable numbers of neurons in the hidden layer which had been trained on the FIA traces to the solutions described in table 2.1 to a maximum output error of 0.1 and a maximum total network error of 0.01, with a learning rate of 0.5 and a momentum of 0.75. Figures 2.7 a to 2.7g depict the outputs from a network (with 55 neurons in its hidden layer) to the FIA patterns as a function of the noise added to the valid FIA peak.

As might be expected, it can be seen from figures 2.7a to 2.7g that the outputs of neurons corresponding to species which were absent , were more affected by the addition of noise than those of neurons corresponding to species which were present. Even at the highest noise levels the outputs of neurons corresponding to species which were present (desired output =1) remained high. It appears from figure 2.7a that at noise levels of ±10% ±50% of the sodium ISE peak height that the network is sensitive to noise added at levels of ±10% to ±50% to the traces from the potassium and calcium ISEs to solutions only containing sodium (as observed from the increasing outputs of the $K_{(rt)}$ and $Ca_{(rt)}$ neurons). In contrast, the $Na_{(rt)}$ neuron output is relatively unaffected. At noise levels of ±200% of the sodium ISE peak height, the network is totally incapable of differentiating between the FIA peak and the noise, and every ion is classified as being present.

If the output from neuron $Na_{(rt)}$ in figure 2.7b is compared with the output of neuron $K_{(rt)}$ in figure 2.7a it can be seen that at noise levels of ±10% and ±25% of the viable peak height, that the network demonstrated a lower sensitivity to noise on the sodium ISE trace from solutions containing only potassium than to noise on the trace from the

Figures 2.7a to 2.7g The outputs from the Na(rt), K(rt) and Ca(rt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the simple FIA patterns depicted in appendix 1, when the network is tested with the patterns distorted by noise addition.

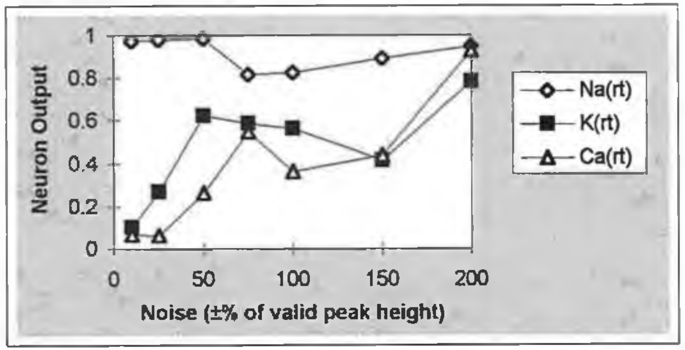


Figure 2.7a Solution containing sodium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =0

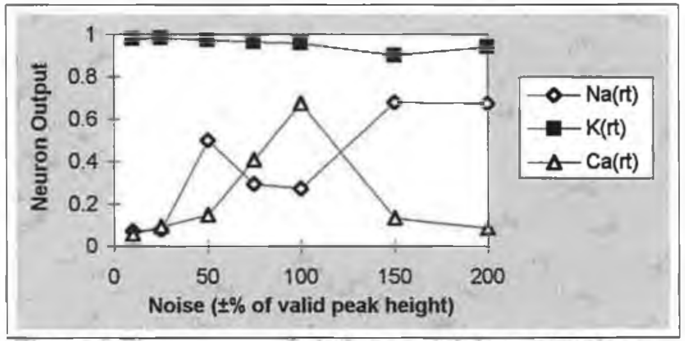


Figure 2.7b Solution containing potassium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =0

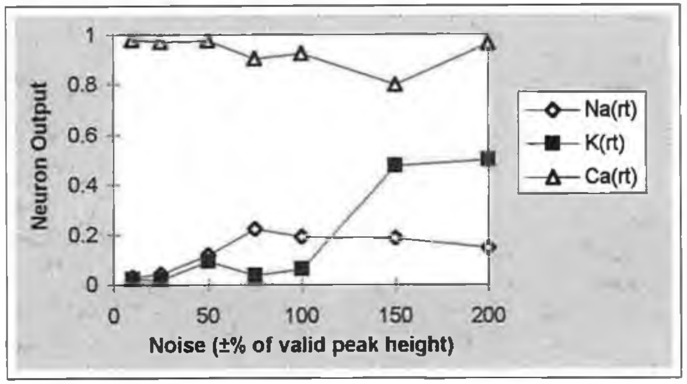


Figure 2.7c Solution containing calcium
Desired Output: Na neuron =0, K neuron =0, Ca neuron =1

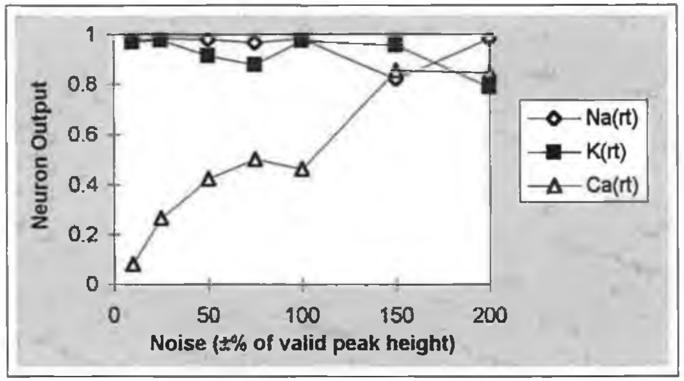


Figure 2.7d Solution containing sodium and potassium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =0

Figures 2.7a to 2.7g continued The outputs from the Na(rt), K(rt) and Ca(rt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the simple FIA patterns depicted in appendix 1, when the network is tested with the patterns distorted by noise addition.

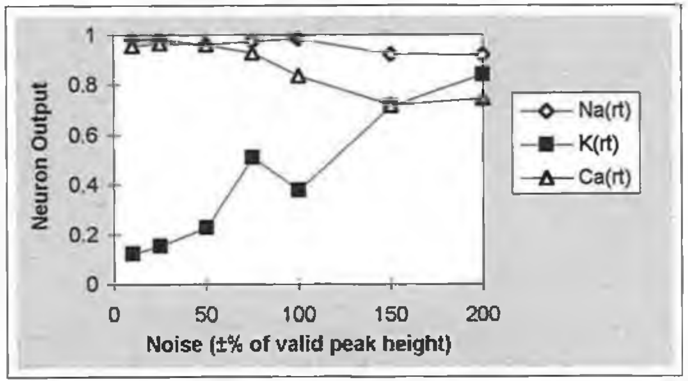


Figure 2.7e Solution containing sodium and calcium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =1

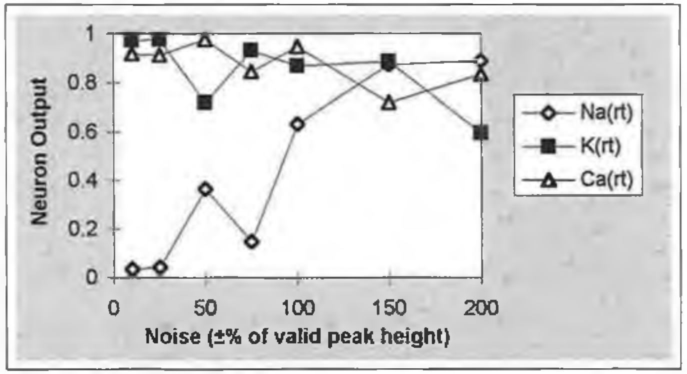


Figure 2.7f Solution containing potassium and calcium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =1

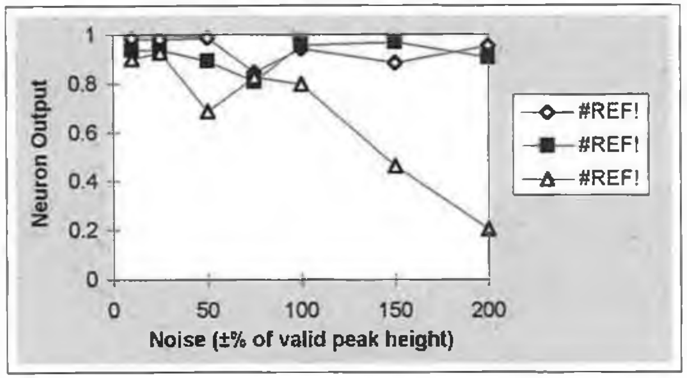


Figure 2.7g Solution containing sodium, potassium and calcium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =1

potassium ISE from solutions only containing potassium (i.e. the output of neuron $K_{(rt)}$ in the noise range ±10% and ±25% of the sodium ISE peak height in figure 2.7a is larger than the output of the neuron $Na_{(rt)}$ in the same noise range in figure 2.7b). This is interesting because on examination of the original FIA peaks in appendix 1, it appears that the sodium ISE demonstrates a larger response to the presence of potassium than the potassium ISE demonstrates to the presence of sodium (this coincides with values determined for the selectivity coefficients for the electrodes [studies in chapter 3] $\mathbf{k}^{\mathbf{pot}}_{\mathbf{NaK}} \cong 0.1$, $\mathbf{k}^{\mathbf{pot}}_{\mathbf{KNa}} < 10^{-6}$, literature sources $\mathbf{k}^{\mathbf{pot}}_{\mathbf{NaK}} \cong 3.9\text{x}10^{-3}$ (21), $\mathbf{k}^{\mathbf{pot}}_{\mathbf{KNa}} \cong 4.1\text{x}10^{-4}$ (22)). It is difficult to follow trends in the outputs from the $Na_{(rt)}$ and $Ca_{(rt)}$ neurons in figure 2.7b in the noise range ±50% to ±150% of the potassium ISE peak height, because the increase in output from $Ca_{(rt)}$ in the noise range ±50% to ±100% of the potassium ISE peak height coincides with a decreased output from the $Ca_{(rt)}$ neuron, similarly the decrease in output of the $Na_{(rt)}$ neuron between the noise levels of ±100% and ±150% of the potassium peak height coincides with an increase in the output of $Ca_{(rt)}$.

Comparing the output of $Na_{(rt)}$ in figure 2.7b and in figure 2.7c, it appears that the network, is less sensitive to noise on the trace from the sodium ISE to solutions containing calcium than to solutions containing potassium (as judged by the fact that the output from $Na_{(rt)}$ in figure 2.7c does not achieve as large an output value as in figure 2.7b at noise levels greater than ±25% of the valid peak height.

This may be related to the presence of a negative potential feature on the traces of the sodium and potassium ISEs to solutions containing only calcium as seen in pattern 1.3 in appendix 1. This negative feature is also present in the traces from the sodium, potassium and calcium ISEs to solutions containing only sodium and only potassium, but are not as large (solution containing only sodium, negative potential features : potassium ISE = -6 mV, calcium ISE = -13 mV, solution containing only potassium, negative potential feature: calcium ISE = -15 mV, solution containing only calcium, negative potential features: sodium ISE = -27 mV, potassium ISE = -12 mV). If a network was trained with a pattern with negative features to recognise as the absence of a species it might have less difficulty recognising patterns with negative noise elements[1] .

There can be seen a gradual decrease in the output from the $Ca_{(rt)}$ neuron in figure 2.7c as the noise level increased from 10 to 150% of the calcium ISE peak height. The increased output from the $Ca_{(rt)}$ neuron at noise levels of ±200% of the calcium ISE peak height coincides with a slightly increased output from the $Na_{(rt)}$ neuron.

It can be seen from figure 2.7d that the output of the $Ca_{(rt)}$ neuron increases dramatically as the added noise increases from ±10% to ±200% of the sodium and potassium ISE peak heights. This suggests that the network is highly sensitive to noise on the trace from the calcium ISE trace to binary solutions containing sodium and potassium. This might happen because the network has only got the trace from the calcium ISE to use as a judge for the presence of a baseline for patterns produced by this type of binary solution. Since the magnitude of the peak from the calcium ISE in figure 1.3 (of appendix 1) is smaller than the peaks from the sodium and potassium ISEs (because the calcium ISE has a lower slope than the sodium and potassium ISEs),

[1] As an aside, the negative feature discussed above usually indicates that there is a lower concentration of a primary ion in an injection than in the carrier stream, a method by which this negative feature itself could be recognised could be used to alert the FIA user to this problem.

the network trained with the patterns in appendix 1 would probably have been classifying the presence of calcium on the basis of low magnitude peaks from the calcium ISE. In the absence of any other form of baseline the network may have confused the noise on the calcium ISE trace as the presence of the ion in the solution. It can be seen however that there is little degradation of the output from the $Na_{(rt)}$ and $K_{(rt)}$. neurons with increased noise.

It can be seen from figures 2.7e and 2.7f that the networks perform well (in terms of the low output from the $K_{(rt)}$ neuron and high output from the $Na_{(rt)}$ and $Ca_{(rt)}$ neurons in figure 2.7e and the low output from the $Na_{(rt)}$ and high outputs from the $K_{(rt)}$ and $Ca_{(rt)}$ in figure 2.7f) at noise levels of ±10% and ±25% of the sodium ISE peak height (in figure 2.7e) and the potassium ISE peak height (in figure 2.7f).

At higher noise levels the outputs from the $K_{(rt)}$ neuron (in figure 2.7e) and the $Ca_{(rt)}$ neuron (in figure 2.7f) rapidly increase as the network becomes more sensitive to noise on the traces, apparently acting as a judge for the baseline. In the tertiary solutions containing sodium, potassium and calcium (depicted in figure 2.7g) , the network demonstrates an ability to classify correctly the presence of sodium and potassium, but as the noise level increases, the ability of the network to classify the presence of calcium is reduced dramatically. However noise levels of ±50% of the height of the tallest FIA peak in the pattern (potassium ISE) corresponds to ~±100% of the height of the peak from the calcium ISE. When this is taken into account, it can be seen that the $Ca_{(rt)}$ neuron classification is on a par with the others in the range 0-±100% noise.

In general, it can be said that if a species is present in the original solution, that it is never classified as being absent (except for calcium in the tertiary mixture in figure 2.7g). However there is a tendency for the network to confuse noise with analytical peaks when the noise becomes relatively large (i.e. > ± 50% of the analytical signal). Whether this is happening because the network is confusing the spiked features in the noise with the shape of the FIA peak or is simply assuming that any feature above a particular potential represents a peak will become more clear with the studies involving baseline shifting and peak height reduction. However it certainly suggests the possibility of training with patterns distorted by noise addition to alleviate this problem.

The raw data corresponding to the actual classifications made by the network in this study is depicted in table 2.1 in the tables appendix.

Studies on the effect of varying the number of neurons in the hidden layer on the classification ability of the networks of the noisy patterns in appendix 2 did not yield any conclusive information pertaining to the generalisation abilities of variably sized networks.

2.3.1.4 Classification Of Patterns With Variable Peak Heights

An investigation was performed into how networks trained with the FIA patterns depicted in appendix 1 would perform when tested with patterns derived from these FIA traces by reducing the height of the peaks from ISEs selective for ions which were present in the solutions which produced the original FIA patterns. The heights of the peaks were reduced by 25%, 50%, 75% and 90% of their original amplitudes without altering the traces from the other ISEs. The resulting patterns are depicted in appendix 3. These patterns were then presented to networks with variable numbers of neurons in the hidden layer which had been trained on the original FIA in the previous section. The classifications of the patterns in appendix 3 by a network with 55 neurons in its hidden layer are depicted in figures 2.8a to 2.8f as a function of the resulting signal to noise

Figures 2.8a to 2.8f The outputs from the Na(rt), K(rt) and Ca(rt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the simple FIA patterns depicted in appendix 1, when the network is tested with the patterns distorted by reduction of the height of the valid FIA peak.

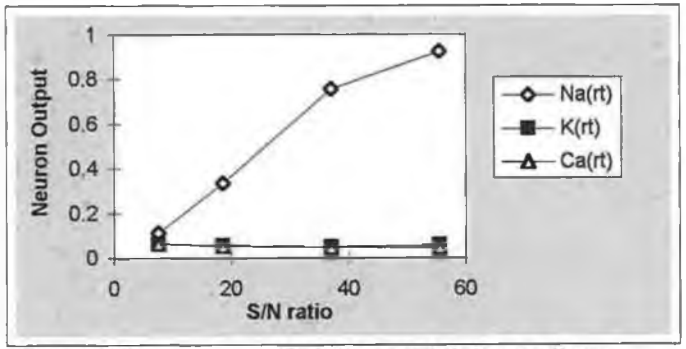


Figure 2.8a Solution containing sodium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =0

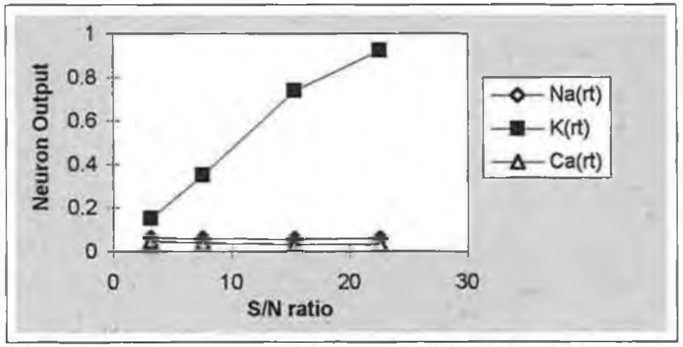


Figure 2.8b Solution containing potassium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =0

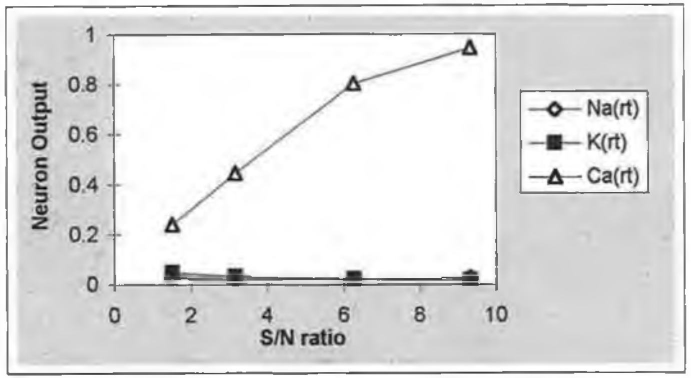


Figure 2.8c Solution containing calcium
Desired Output: Na neuron =0, K neuron =0, Ca neuron =1

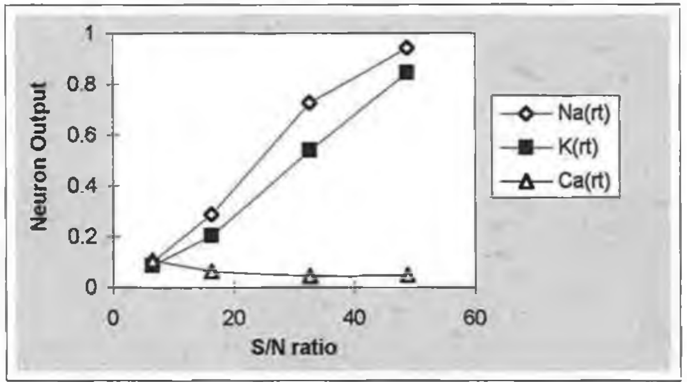


Figure 2.8d Solution containing sodium and potassium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =0

Figures 2.8a to 2.8f continued The outputs from the Na(rt), K(rt) and Ca(rt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the simple FIA patterns depicted in appendix 1, when the network is tested with the patterns distorted by reduction of the height of the valid FIA peak.

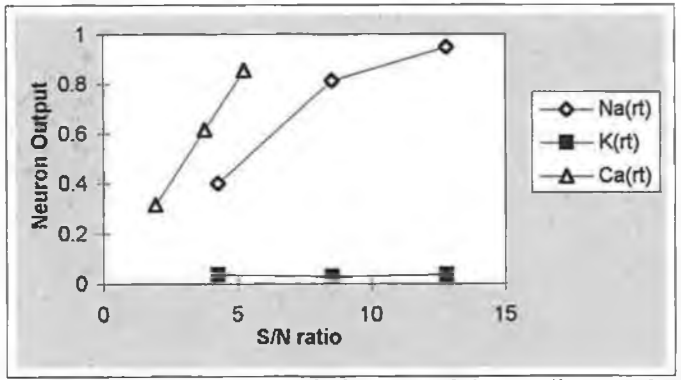


Figure 2.8e Solution containing sodium and calcium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =1

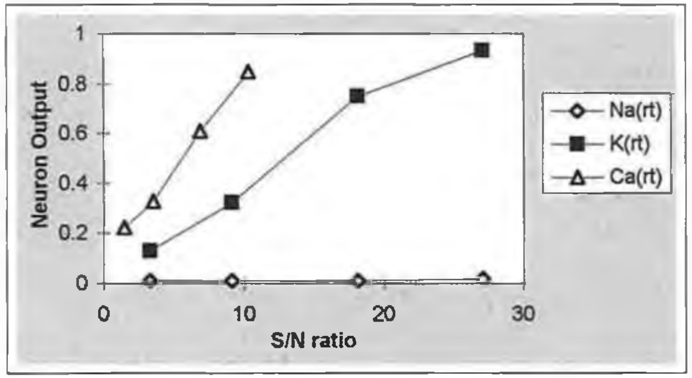


Figure 2.8f Solution containing potassium and calcium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =1

(S/N) ratios[2]. The actual outputs of the network to the patterns are described in appendix table 2.2.

It can be seen from figure 2.8a that the output of the $Na_{(rt)}$ neuron increases very rapidly as the S/N ratio increases from 7 to 37, further increase in the S/N ratio beyond this point, does not produce as large an increase in the output from the $Na_{(rt)}$ neuron. This suggests that the network confidence of the presence of a species increases rapidly with increased S/N ratio, until a point is reached at which the network is almost totally sure of the presence of the species. Increasing the S/N ratio thereafter, merely serves to affirm that decision.

A similar pattern is seen for solutions containing potassium in figure 2.8b and solutions containing calcium in figure 2.8c. However, the point at which the increase in S/N ratio has a reduced affect on the increase in output from the $K_{(rt)}$ neuron in figure 2.8b, occurs at a much lower S/N ratio of approximately 15. This is not too surprising because the peak from potassium ISE had been reduced whereas the positive potential feature in the trace from the sodium ISE (seen in figure 1.2 in appendix 1, arising from the contribution of the potassium to the response of the sodium ISE) was unaltered. Since this positive feature in the sodium ISE trace was included in the calculations of noise, the apparent noise level for the patterns from the solution containing only potassium was higher than from the patterns from the solution containing only sodium. meaning that the amplitude of the potassium peak was comparable to the height of the positive potential feature from the sodium ISE.

[2] The noise level was calculated for primary solutions as the standard deviation of the trace from an ISE selective to an absent ion. The signal was calculated as the amplitude of a valid FIA peak.

The S/N ratios for the solution containing only calcium are very low[3] and increasing the S/N ratio has a reduced affect on the increased output from the $Ca_{(rt)}$ neuron beyond an S/N ratio of approximately 6.

It is difficult to draw similar conclusions concerning the affect of increasing the S/N ratio on the increase in output from the $Na_{(rt)}$ and $K_{(rt)}$ neurons in figure 2.8d. However, it can be seen that the increase in output from $Na_{(rt)}$ and $K_{(rt)}$ coincides with a decreased output from $Ca_{(rt)}$ as the S/N ratio increases. When the heights of the sodium and potassium ISE peaks are reduced by 90% (i.e. S/N ratio of approximately 6) the network can no longer discriminate between the sodium and potassium ISE peaks and the calcium ISE baseline (as judged by the similar outputs from the $Na_{(rt)}$, $K_{(rt)}$ and $Ca_{(rt)}$ neurons).

For the other binary solutions (i.e. solutions containing (a) sodium and calcium and (b) potassium and calcium) depicted in figures 2.8e and 2.8f, it can be seen that increasing the S/N ratio causes rapid increase in the outputs from the $Ca_{(rt)}$ neuron whereas the increase in output from the $Na_{(rt)}$ neuron (in figure 2.8e) and the $K_{(rt)}$ neuron(in figure 2.8f) show a similar pattern of increase as was seen for the $Na_{(rt)}$ neuron in figure 2.8a. the outputs of neurons corresponding to ions present (e.g. $Na_{(rt)}$ in figure 2.8e) were always much higher than the outputs from neurons corresponding to ions which were absent (e.g. $K_{(rt)}$ in figure 2.8e) even at the lowest S/N ratios.

If it is assumed that an output of 0.5 of greater from a neuron indicates the presence of a species, then it can be seen from table 2.6 that the network is capable of classifying the

[3] One might wonder why the S/N ratios quoted for the calcium containing solutions in table 2.2 in the table appendix are not approximately half those for the sodium containing solutions, given that the peaks form both ISEs were subjected to the same degree of peak height reduction. However an examination of pattern 1.3 in appendix 1, it can be seen that there is a negative potential feature on the traces form the sodium and potassium ISEs, which was included in the noise calculations in order to determine the S/N ratio. Hence the apparent noise level was is higher than that for the sodium ISE, meaning that the S/N ratio is lower than might be expected on the basis of the different slopes of the electrodes.

presence of a species correctly up to a peak height reduction of 50% of the original height.

2.3.1.5 Classification Of Patterns With Baseline Shifting

An investigation was performed into how networks trained with the FIA patterns depicted in appendix 1 would perform when tested with patterns derived from these FIA traces by the shifting of the baselines of the traces from the ISEs. The baselines of the peaks were shifted by +50, +100 and -40 mV, which corresponds to approximately +20%,+40% and -16% of the original monovalent ion peak heights. The resulting patterns are depicted in appendix 4. These patterns were then presented to networks with variable numbers of neurons in the hidden layer which had been trained on the simple FIA patterns in the fashion described in section 2.3.1.3. The classifications of the patterns in appendix 4 by a network with 55 neurons in its hidden layer are depicted in table 2.5.

It can be seen from this table that shifting the baseline in the positive or negative direction appears to have a greater influence on misclassifying an absent species as being present than misclassifying a present species as being absent. It can be seen that for solutions containing only sodium that a negative baseline shift increases the output $K_{(rt)}$ neurons when compared with its output to patterns with the positive baseline shift. In contrast to this, a negative baseline shift decreases the output of the $Ca_{(rt)}$ neuron.

			$Na_{(rt)}$		$K_{(rt)}$		$Ca_{(rt)}$	
Pattern Number	Solution Key	Baseline Shift	Desired Output	Actual Output	Desired Output	Actual Output	Desired Output	Actual Output
1	Na	+50 mV	1	0.978	0	0.222	0	0.321
2	Na	-40 mV	1	0.992	0	0.363	0	0.080
3	K	+50 mV	0	0.422	1	0.965	0	0.275
4	K	-40 mV	0	0.370	1	0.996	0	0.119
5	Ca	+50 mV	0	0.400	0	0.164	1	0.813
6	NaK	+50 mV	1	0.984	1	0.957	0	0.577
7	NaK	+100 mV	1	0.987	1	0.947	0	0.835
8	NaCa	+50 mV	1	0.986	0	0.363	1	0.826
9	NaCa	+100 mV	1	0.989	0	0.760	1	0.835
10	KCa	+50 mV	0	0.288	1	0.958	1	0.798
11	KCa	+100 mV	0	0.831	1	0.948	1	0.803
12	NaKCa	+50 mV	1	0.987	1	0.947	1	0.840
13	NaKCa	+100 mV	1	0.987	1	0.947	1	0.840

Table 2.5 - Classification results from a network with 55 neurons in its hidden layer trained as described in the text to a test set composed of the FIA patterns depicted in appendix 4.

On examination of the original FIA trace in appendix 1 it can be seen that the trace from the calcium ISE has a larger negative potential feature than the potassium ISE for a solution containing only sodium (pattern 1.1). If the network was trained to recognise a negative feature in the calcium ISE trace for a solution containing only sodium as being related to the absence of the calcium, it may not have been as sensitive to a negative displacement of the baseline from the calcium ISE as to a negative baseline shift of the trace from the potassium ISE. It can also be seen that for solutions containing only potassium, the outputs from the $Na_{(rt)}$ and $Ca_{(rt)}$ neurons decreased with the negative baseline shift relative to their outputs with a positive baseline shift. On examination of the original FIA pattern (pattern 1.2 in appendix 1) it can be seen that there is a negative potential feature in the trace from the calcium ISE and a positive potential feature in the trace from the sodium ISE. The network may not have been as sensitive to negative displacements of the baseline of the calcium ISE as to positive displacements because of the presence of this negative feature in the original FIA pattern. Applying a negative shift to the baseline of the sodium ISE would have brought the peak of its original positive feature into the general region where the negative feature existed in the original calcium ISE FIA trace and this may have influenced the reduction in the output from the $Na_{(rt)}$ neuron. It can be seen from table 2.5 that a baseline shift of +100 mV on the traces from solutions containing sodium and calcium, produced an output from the $K_{(rt)}$ neuron comparable to that achieved with the addition of noise at 150% of the peak height of the sodium peak (but the classification of the presence of calcium is better with the baseline shift than with the noise). Similarly for FIA traces from solutions containing potassium and calcium, a baseline shift of +100 mV produces an output from the $Na_{(rt)}$ neuron comparable to that achieved with addition of noise at 150% of the height of the peak from the potassium ISE (as before the baseline shift produced a better output from the $Ca_{(rt)}$ neuron than the noise addition). This is suggestive that the presence of baselines at different positions in the potential range within which the original FIA traces were acquired had a comparable role in the misclassification of patterns as the presence of noise on the patterns. The study did not yield conclusive information concerning the role of variable numbers of neurons in the hidden layer on the classification of these patterns unfortunately.

2.3.1.6 Summary

From the previous sections it can be seen that the different effects of noise addition, baseline shifting and peak height variation have different effects on the classification of patterns by networks trained on undistorted patterns. Noise addition and baseline shifting appear to play an important role in determining the outputs of neurons corresponding to the decisions concerning the presence or absence of species which were absent in the solutions responsible for the original FIA traces in appendix 3, whereas peak height variation played an important role in determining the outputs from neurons corresponding to the decisions concerning the presence or absence of species which were present in the solutions which produced the patterns in appendix 1. If it is assumed that an output from a neuron of 0.5 or greater indicates the presence of a species and an output of less than 0.5 indicated the absence of the species, then the networks trained on the patterns in appendix 1 were capable of classifying patterns correctly up to noise levels of approximately 50% of the largest peak in a particular FIA pattern, baseline shifts of approximately 50 mV (approximately 20% of the height of the peak of the original FIA traces from the sodium and potassium ISEs) and reduction of the heights of peaks by approximately 50%. The next stage in this study was to determine whether this performance could be improved by redesigning the set of patterns used to train the networks to contain variable amounts of these distortions and to test the performance of the resulting networks against new sets of testing patterns also containing variable amounts of these distortions.

2.3.2 Performance Of Networks Trained With Patterns Containing Noise, Baseline Shifts and Variation Of Peak Heights

A training set composed of 70 random patterns representing the flow injection peaks distorted by different combinations of baseline shift, peak reduction and noise addition, at levels much worse than normally encountered under experimental conditions was designed and a test set of 56 patterns was composed in a similar fashion. Table 2.6 describes the contents of the training set used for subsequent studies. The patterns in the training set are graphically depicted in appendix 5.

For a net with 50 neurons in its hidden layer, 77% of the patterns in the testing set were classified correctly, based on the assumption that an output of greater than 0.9 indicated the presence of the species and an output of less than 0.1 indicated its absence.

Tables 2.7 to 2.13, list the outputs of the network with 50 neurons in its hidden layer to the different patterns in the test set, it also lists the ratios of the network outputs to the ions which are supposed to be present in the sample to the ions which are supposed to be absent from the sample. The tables also contain a listing of the different distortions applied to the original experimental trace required for the generation of the particular pattern.

Pattern Number	Solution Key	Noise (at % of maximum peak height)	Baseline shift (at % of maximum peak height)	Reduction in height of sodium ISE peak	Reduction in height of potassium ISE peak	Reduction in height of calcium ISE peak
1	NaKCa	-	-20%	-	-	-
2	NaK	25%	-	75%	75%	75%
3	NaCa	-	-	30%	-	75%
4	NaK	50%	+20%	35%	35%	-
5	Na	-	+16%	60%	-	-
6	Na	-	-	25%	-	-
7	Ca	-	+222%	-	-	75%
8	NaCa	-	+21%	-	-	75%
9	Na	-	+42%	-	-	-
10	Ca	50%	+56%	-	-	-
11	KCa	10%	-	-	-	-
12	Ca	-	-	-	-	50%
13	NaK	-	+20%	50%	25%	-
14	NaK	-	-	25%	50%	-
15	NaKCa	-	-	75%	25%	50%
16	Ca	-	+56%	-	-	25%
17	KCa	50%	+19%	-	35%	35%
18	NaK	-	+20%	-	-	-
19	NaCa	-	+22%	50%	-	-
20	NaK	-	+20%	25%	50%	-
21	Na	-	-	10%	-	-
22	K	-	-	-	25%	-
23	Ca	25%	-	-	-	-
24	K	25%	-	-	35%	-
25	Na	50%	+21%	80%	-	-
26	NaCa	-	+22%	-	-	-
27	Na	-	-17%	75%	-	-
28	Ca	10%	-	-	-	-
29	KCa	25%	-	-	75%	75%
30	K	50%	+19%	-	80%	-
31	K	75%	-	-	-	-
32	KCa	-	+29%	-	-	-
33	K	-	+38%	-	60%	-
34	NaKCa	-	-	25%	50%	75%
35	NaKCa	-	+20%	-	-	-
36	NaCa	-	-17%	-	-	-
37	K	-	-15%	-	75%	-

Table 2.6 - A list of the patterns composing the training set for further studies. The solution key refers to the composition of the original solutions used to produce the patterns (determined from table 2.1, and the subsequent columns refer to the distortions applied the relevant FIA pattern

Pattern Number	Solution Key	Noise (at % of maximum peak height)	Baseline Shift (at % of maximum peak height)	Reduction in height of Sodium ISE Peak	Reduction in height of Potassium ISE Peak	Reduction in height of Calcium ISE Peak
38	NaCa	75%	-	-	-	-
39	NaKCa	75%	-	-	-	-
40	NaKCa	-	-20%	40%	40%	80%
41	NaCa	50%	+4%	-	-	-
42	NaK	75%	-	-	-	-
43	K	-	-	-	35%	-
44	NaKCa	25%	-	50%	75%	25%
45	KCa	-	+19%	-	25%	50%
46	NaK	10%	-	-	-	-
47	Na	75%	-	-	-	-
48	NaCa	-	-	50%	-	50%
49	KCa	-	+19%	-	25%	50%
50	KCa	75%	-	-	-	-
51	NaKCa	10%	-	-	-	-
52	NaK	-	-	25%	50%	-
53	NaK	-	-	25%	50%	-
54	NaCa	25%	-	75%	-	75%
55	Ca	-	-	-	-	50%
56	NaKCa	50%	+4%	80%	80%	60%
57	Na	25%	-	35%	-	-
58	NaKCa	-	+20%	40%	40%	40%
59	KCa	-	-	-	25%	50%
60	Ca	25%	-	-	-	75%
61	KCa	-	-	-	50%	35%
62	K	-	-	-	25%	-
63	K	10%	-	-	-	-
64	Na	-	-13%	-	-	-
65	Na	10%	-	-	-	-
66	Ca	-	+111%	-	-	-
67	KCa	-	+38%	-	-	-
68	K	-	+38%	-	-	-
69	Na	-	+250%	25%	-	-
70	Ca	50%	+330%	-	-	-

Table 2.6 continued- A list of the patterns composing the training set for further studies. The solution key refers to the composition of the original solutions used to produce the patterns (determined from table 2.1, and the subsequent columns refer to the distortions applied the relevant FIA pattern

Table 2.7 depicts the outputs from the network to the distorted pattern generated from FIA traces to a sample which contained sodium only. The natures of the distortions employed and the resultant network responses suggest that as the height of the sodium ISE peak became smaller and hence closer to the level of the noisy signals from the potassium and calcium ISEs, that it became more difficult to distinguish between the presence of sodium or calcium in the sample, because of the sensitivity of the network to low amplitude calcium ISE signals relative to the sodium and potassium ISE peak amplitudes. The worst response from the network obtained for this series of patterns was obtained for pattern 7 in which the height of the sodium ISE peak had been reduced by 70% and noise at 40% of the original peak height had been added. The network response to this pattern was 0.999 (Na), 0.032 (K), 0.546 (Ca), however even with this level of distortion it is evident that the network output indicates a degree of confidence in the presence of sodium almost twice that of the presence of calcium.

Table 2.8 depicts the outputs of the network to the distorted patterns generated from FIA traces to samples which contained only potassium. Interference signals from the sodium ISE to the potassium present in the sample produces the relatively low output ratio of potassium to sodium seen in pattern 13 (6.566) in which the baseline had been shifted in the negative direction by approximately 8% of the original peak height, and the height of the potassium ISE peak had been reduced by 80%. Noise at 60% of the original peak height also produced low output ratios in pattern 16 (6.098 for K/Na and 2.874 for K/Ca).

Table 2.7. Classification results to the test patterns 1-8 in appendix 6 of a network with 50 neurons in its hidden layer trained with the patterns depicted in appendix 5 and described in table 2.6. Learning rate = 0.5, momentum = 0.75, maximum output error = 0.1. These test patterns are derived from the FIA traces to a solution containing sodium only.

	Network Outputs			Decision Output Ratio		Distortions Of Original Pattern Employed[4]		
Pattern	Na neuron	K neuron	Ca neuron	Na/K	Na/Ca	Noise Addition	Baseline Shift	Peak Height Reduction
1	1.000	0.067	0.017	14.926	58.823		~62%	
2	1.000	0.011	0.010	90.909	100.000			60%
3	1.000	0.038	0.02	26.316	50.000	30%	~38%	40%
4	1.000	0.245	0.035	4.082	28.571	60%	~42%	50%
5	0.986	0.017	0.021	58.000	46.952		~10%	80%
6	1.000	0.021	0.104	47.619	9.615	50%		
7	0.999	0.032	0.546	31.219	1.830	40%		70%
8	1.000	0.251	0.023	3.984	43.478	70%	~30%	

Table 2.8. Classification results to the test patterns 9-16 in appendix 6 of a network described as per table 2.7. These test patterns are derived from the FIA to a solution containing potassium only

	Network Outputs			Decision Output Ratio		Distortions Of Original Pattern Employed[4]		
Pattern	Na neuron	K neuron	Ca neuron	K/Na	K/Ca	Noise Addition	Baseline Shift	Peak Height Reduction
9	0.020	1.000	0.005	50.000	200.000	40%		60%
10	0.008	1.000	0.007	125.000	142.857	30%	~23%	60%
11	0.042	1.000	0.023	23.809	43.478			30%
12	0.029	0.999	0.006	34.448	166.500	50%	~30%	70%
13	0.152	0.998	0.006	6.566	166.333		~-8%	80%
14	0.023	1.000	0.028	43.478	35.714		~10%	
15	0.013	1.000	0.095	76.923	10.526	70%	~20%	
16	0.164	1.000	0.348	6.098	2.874	60%		

[4] The distortions are expressed as percentages of the valid peak height in the original FIA patterns

Table 2.9 depicts the outputs of the network to the distorted patterns generated from FIA traces to samples which contained only calcium. The high sensitivity of the network to low amplitude signals from the calcium ISE and the high selectivity of the other ISEs to calcium meant that all the patterns were classified correctly with a very high degree of confidence (worst output ratio for Ca/Na is 12.500 for pattern 21. Interestingly, the output ratios for Ca/K were around an order of magnitude larger than for Ca/Na, suggesting that under the conditions of this study, the system is able to discriminate potassium interference much better than sodium interference on signals arising from calcium.

Table 2.10 depicts the network outputs to distorted FIA traces generated from a sample containing sodium and potassium. In these cases the FIA traces were distorted by shifting the baseline and adding noise as before, but the heights of the sodium and potassium ISE peaks were changed independently of each other. The worst network output to any of the patterns in the test set was produced to pattern 26. Although the sodium and potassium were positively identified as being present, the high levels of noise on the calcium ISE signal made it difficult to determine whether there was a calcium peak present too. However, even in this difficult case, the output ratios of 1.704 (Na/Ca) and 1.709 (K/Ca) the network is almost twice as confident of the presence of the correct ions in the test solution.

Table 2.11 depicts the network outputs to distorted FIA traces generated from a sample containing sodium and calcium. The network had most difficulty with patterns 38 and 40. For pattern 38, the noise addition at 60% the original peak height and the interference signals from the potassium ISE to the sodium and calcium present reduced the output ratios for Na/K and Ca/K to just over six. Similarly for pattern 40 in which noise at 50% of the maximum peak height had been added and the heights of the sodium and calcium ISE peaks had been reduced, the same output ratios were reduced to 10.753 and 8.581, respectively.

Table 2.9. Classification results to the test patterns 17-24 in appendix 6 of a network described as per table 2.7. These test patterns are derived from the FIA traces of pattern to a solution containing calcium

	Network Outputs			Decision Output Ratio		Distortions of original pattern employed[5]		
Pattern	Na neuron	K neuron	Ca neuron	Ca/Na	Ca/K	Noise Addition	Baseline Shift	Peak height Reduction
17	0.062	0.020	0.910	14.677	45.500		~213%	60%
18	0.017	0.008	0.995	58.529	124.375	60%	~22%	40%
19	0.080	0.012	0.989	12.362	82.417			65%
20	0.037	0.006	0.998	26.973	166.333	40%	~75%	20%
21	0.080	0.002	1.000	12.500	500.000	30%		
22	0.038	0.010	1.000	26.316	100.000	75%		40%
23	0.049	0.010	1.000	20.408	100.000		~85%	
24	0.037	0.020	1.000	27.027	50.000	80%	~43%	

Table 2.10 Classification results to the test patterns 25-32 in appendix 6 of a network described as per table 2.7. These test patterns are derived from the FIA traces to a solution containing sodium and potassium

	Network Outputs			Decision Output Ratio		Distortions Of Original Pattern Employed[6]		
Pattern	Na neuron	K neuron	Ca neuron	Na/Ca	K/Ca	Noise Addition	Baseline Shift	Peak Reduction
25	1.000	1.000	0.136	7.353	7.353	30%		40% (Na), 60%(K)
26	0.997	1.000	0.585	1.704	1.709	60%	~40%	60% (Na), 40% (K)
27	1.000	1.000	0.023	43.478	43.478		~-20%	20% (Na), 20% (K)
28	0.986	1.000	0.024	41.083	41.667			70%(Na), 60% (K)
29	1.000	1.000	0.007	142.857	142.857		~12%	
30	0.993	1.000	0.021	47.286	47.619		~28%	60%(Na), 30% (K)
31	1.000	1.000	0.000	-	-	40%		
32	1.000	1.000	0.032	31.250	31.250	20%	~8%	

[5] The distortions are expressed as percentages of the valid peak height in the original FIA patterns

[6] The noise distortion is expressed as a percentages of the height of the largest peak in the original FIA patterns, the baseline shift and peak height reduction are expressed as a percentage of the height of the valid FIA peak

Table 2.11 Classification results to the test patterns 32-40 in appendix 6 of a network described as per table 2.7 These test patterns are derived from the FIA traces to a solution containing sodium and calcium

	Network Outputs			Decision Output Ratio		Distortions Of Original Pattern Employed[7]		
Pattern	Na neuron	K neuron	Ca neuron	Na/K	Ca/K	Noise Addition	Baseline Shift	Peak Height Reduction
33	1.000	0.025	0.971	40.000	38.840			40% (Na), 40% (Ca)
34	0.996	0.050	0.546	19.920	10.920		~43% (Na), ~127%(Ca)	60% (Na), 60% (Ca)
35	0.999	0.003	0.982	333.000	327.333	40%		70% (Na), 20% (Ca)
36	1.000	0.020	0.988	50.000	49.400		~30% (Na), ~90% (Ca)	
37	1.000	0.005	0.988	200.000	197.600	40%		
38	1.000	0.152	0.993	6.579	6.118	60%	~13% (Na), ~38% (Ca)	
39	0.798	0.020	0.995	39.900	49.750	20%	~26% (Na), ~76% (Ca)	85% (Na), 30% (Ca)
40	1.000	0.093	0.798	10.753	8.581	50%	~13% (Na), ~138% (Ca)	45% (Na), 65% (Ca)

[7] The noise distortion is expressed as a percentages of the height of the largest peak in the original FIA patterns, the baseline shift and peak height reduction are expressed as a percentage of the height of the valid FIA peak

Table 2.12 details the network outputs to distorted FIA traces generated from a sample containing potassium and calcium. In all cases described in this table the patterns were identified correctly with a high degree of confidence, the worst case being pattern 48 where increasing the noise added to 70% of the original peak heights caused the output ratios for K/Na and Ca/Na to fall to around 20 from values of over 300 (pattern 47), even though other distortions were not as severe.

Table 2.13 which details the network outputs to distorted FIA traces generated from a sample containing all three cations. The network was able to cope confidently with all distortions investigated. The least confident result was a 0.760 prediction for Ca (pattern 56), which may have been caused by a larger shift in the baseline (150% of peak height compared to 70% for sodium and 60% for potassium).

It can be seen from these tables that for 44 of the 56 ion combinations investigated, the patterns in the testing set were classified correctly, based on the assumption that an output of greater than 0.9 indicated the presence of the species and an output of less than 0.1 indicated its absence. However for those patterns which were not classified correctly according to these criteria there was still a greater degree of confidence in the presence of the correct ion(s) rather than the other ions which may have been present in the sample. If we consider the outputs of the neural network to ions which are present relative to ions which are absent, it can be seen that in around 38 of the 48 possible patterns studied, the classification of the pattern indicated a degree of confidence of at least 10 times greater for the presence of the correct ion relative to the other possible ion permutations. In fact, in every case, the network favoured the correct composition, with the worst case being the output ratio of 1.7 for the correct prediction of the presence of sodium and potassium compared to the presence of calcium as discussed above (see table 2.10, pattern number 26).

Table 2.12. Classification results to the test patterns 41-48 in appendix 6 of a network described as per table 2.7. These test patterns are derived from the FIA traces to a solution containing potassium and calcium

	Network Outputs			Decision Output Ratio		Distortions Of Original Pattern Employed[8]		
Pattern	Na neuron	K neuron	Ca neuron	K/Na	Ca/Na	Noise Addition	Baseline Shift	Peak Reduction
41	0.001	1.000	0.999	1000.000	999.000	40%	-	-
42	0.001	0.998	0.998	998.000	998.000	60%	~10% (K), ~25% (Ca)	60% (K), 40% (Ca)
43	0.005	1.000	1.000	200.000	200.000	30%		40% (K), 20% (Ca)
44	0.003	1.000	1.000	333.333	333.333	-	~38% (K), ~103% (Ca)	-
45	0.002	1.000	0.997	500.000	498.501	-	~-15% (K), ~-40% (Ca)	70% (K), 60% (Ca)
46	0.000	1.000	0.989	-	-	45%	~23% (K), ~61% (Ca)	45% (K), 45% (Ca)
47	0.003	1.000	0.987	333.333	329.000	-	-	25% (K), 50% (Ca)
48	0.047	1.000	1.000	21.276	21.276	70%	~13% (K), ~36% (Ca)	-

[8] The noise distortion is expressed as a percentages of the height of the largest peak in the original FIA patterns, the baseline shift and peak height reduction are expressed as a percentage of the height of the valid FIA peak

Table 2.13. Classification results to the test patterns 41-48 in appendix 6 of a network described for table 2.7. These test patterns are derived from the FIA traces to a solution containing sodium, potassium and calcium

	Network Outputs			Distortions Of Original Pattern Employed[9]		
Pattern	Na neuron	K neuron	Ca neuron	Noise Addition	Baseline Shift	Peak Reduction
49	1.000	1.000	0.990		~47%(Na), ~40%(K), ~100%(Ca)	
50	1.000	1.000	0.992			40% (Na), 40% (K), 30% (Ca)
51	1.000	1.000	0.991	60%		
52	0.991	1.000	0.965		~35% (Na), ~30% (K), ~75% (Ca)	70% (Na), 40% (K), Ca (55%)
53	1.000	0.917	0.984	40%		30% (Na), 70% (K), 35% (Ca)
54	0.999	1.000	0.999	60%	~12% (Na), ~10% (K), ~25% (Ca)	50% (Na), 30% (K), 20% (Ca)
55	0.999	0.999	0.981	30%	~28% (Na), ~24% (K), ~60% (Ca)	55% (Na), 55% (K), 25% (Ca)
56	1.000	1.000	0.760	70%	~70% (Na), ~60% (K), ~150% (Ca)	

[9] The noise distortion is expressed as a percentages of the height of the largest peak in the original FIA patterns, the baseline shift and peak height reduction are expressed as a percentage of the height of the valid FIA peak

2.3.3 Comparison Of The Ability Of A Network Trained With A Data Set Composed Of Simple FIA Peaks And A network Trained With A Data Set Composed Of Distorted FIA Peaks To Recognise Patterns Distorted By Noise Addition, Baseline Shifting And Peak Height Variation

In this section the performance of networks trained with the new data set composed of the distortions of the simple FIA peaks (depicted in appendix 5) will be tested with the same data sets as were used in the studies described in 2.3.1.3-2.3.1.5 and will be compared with the performance of the network trained with the simple FIA patterns (depicted in appendix 1).

Test 1: Varying Noise Levels

Figures 2.9 a to 2.9g graphically depict the results of this for a network with 55 neurons in its hidden layer trained with the patterns described in table 2.6 to a maximum output error of 0.1. Training was performed with a learning rate of 0.5 and a momentum of 0.75. The actual classifications made by the network in this study are described in table 2.3 in the table appendix. Comparing figures 2.7a (from 2.3.1.3) and 2.9a, it can be seen that for patterns generated from solutions containing sodium that the network trained with the distorted training set described in table 2.8 (broad training) performs better than the network trained with the simple FIA patterns (restricted training) up to a noise level of ±75% of the sodium ISE peak height. At a noise level of ±75% there is an unusual increase in the output from the $K_{(bt)}$ neuron. A corresponding effect is seen in the case of solutions containing potassium where there is an increase in output from $Ca_{(bt)}$ neuron and from the $Na_{(bt)}$ neuron at noise levels of ±50% and ±100% of the potassium ISE peak height respectively. This is difficult to explain as the this effect does not appear in patterns from the other solutions and examination of the relevant patterns in appendix 2 does not provide any visual clues. Although it is extremely debatable whether a decision should be made on this basis, if the increased output from the $K_{(bt)}$

[10]neuron is disregarded, the network performs a correct classification of patterns up to a noise level of ±150% of the sodium peak height (using the criteria that an output greater than 0.5 indicates the presence of a species and a n output of less than 0.5 indicates its absence). This compares with the network trained with the simple FIA patterns in appendix 1, which can classify the patterns to a noise level of ±25% of the original peak height. Given that the original training set (table 2.8) contained patterns with noise levels up to ±75% of the maximum peak height, it is surprising that the network is capable of extrapolating to higher noise levels. At noise levels of ±200% of the sodium peak height the network is incapable of distinguishing between the pattern from the sodium ISE and the noisy patterns from the other ISEs and classifies every species as being present.

In the case of solutions containing potassium only, the classification of patterns becomes very unpredictable at noise levels greater than ±50% of the peak height and there does not appear to be much of an improvement in classification performance of the network trained with the distorted patterns and the network trained with the simple FIA patterns. In contrast the network trained with the distorted patterns classifies the patterns from solutions containing calcium much better than the network trained with the simple FIA patterns, in terms of the outputs from the $Na_{(bt)}$ and $K_{(bt)}$ neurons and is capable of classifying patterns correctly to a noise level of ±200% of the peak height using the previously discussed classification criteria. In the case of patterns from solutions containing sodium and potassium, the network trained with the distorted patterns

[10] It is disturbing that the network trained with the simple FIA patterns performs better than the network trained with the distorted training set for these cases. When the experiment was repeated with a different sequence of random numbers used as added noise, this effect was dramatically reduced for the sodium containing solutions. The outputs from the $Na_{(bt)}$, $K_{(bt)}$ and $Ca_{(bt)}$ neurons were equal to 1.000, 0.113 and 0.000 respectively for the addition of noise at ±50% of the sodium ISE peak height and 1.000, 0.060 and 0.085 for the addition of noise at ±75% of the peak height.. Unfortunately the same cannot be said for the patterns from potassium containing solutions. Although the output form the $Ca_{(bt)}$ neuron decreased dramatically to 0.004 for the pattern with noise added at ±50% of the calcium ISE peak height, it increased with noise levels of ±75% to 0.777. At noise levels of ±100% the outputs for the $Na_{(bt)}$ and $Ca_{(bt)}$ neurons had decreased to 0.001 and 0.067 respectively. These results are suggestive that the network trained with the distorted data set may have been sensitive to patterns in the noise on which it was trained, rather than an underlying feature in the FIA trace. Altering the noise pattern with a different random number sequence, may have had a deleterious on its classification abilities. This suggests that a more comprehensive training set should be designed, probably with different seeds for the random number generator

produces a much lower output from the $Ca_{(bt)}$ neuron than the network trained with the simple FIA patterns up to a noise level ±100% of the maximum peak height. At noise levels greater than this the network misclassifies the presence of calcium (this is not too surprising because it can be seen from table 2.6 that the network was trained with noise levels of up to ±75% of the maximum peak height from the FIA trace to this solution). This means that the network interpolated well within the noise range with which it had been trained, but did not extrapolate outside this noise range).

Similarly the network trained with the distorted patterns misclassiffied solutions containing sodium and calcium (seen in figure 2.9e) and potassium and calcium (seen in figure 2.9f) at noise levels grater than ±100% of the maximum peak height.

For the solution containing sodium, potassium and calcium (seen in figure 2.9g), the network trained with the distorted patterns correctly classified the patterns from this solution, up to a noise level of ±200% of the maximum peak height. At this point it classified the calcium ion as being absent (however noise at ±200% of the peak height of the trace from the sodium ISE would correspond to noise at approximately ±400% of the peak height of the trace from the calcium ISE).

Figures 2.9a to 2.9g The outputs from the Na(bt), K(bt) and Ca(bt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the distorted FIA patterns depicted in appendix 5, when the network is tested with the patterns distorted by noise addition.

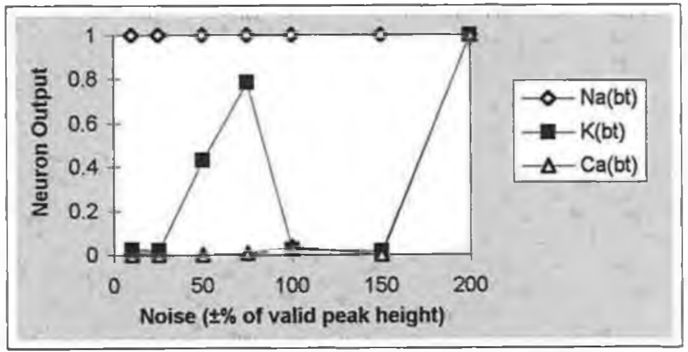


Figure 2.9a Solution containing sodium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =0

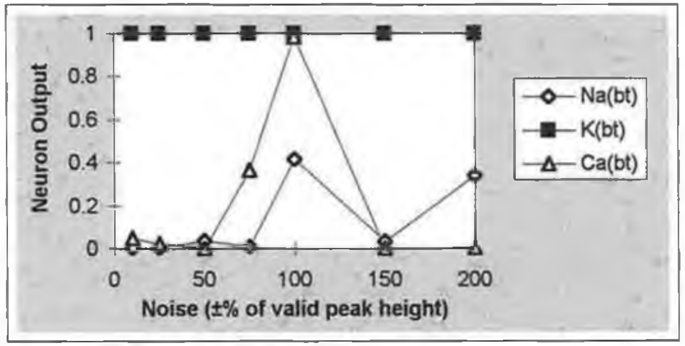


Figure 2.9b Solution containing potassium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =0

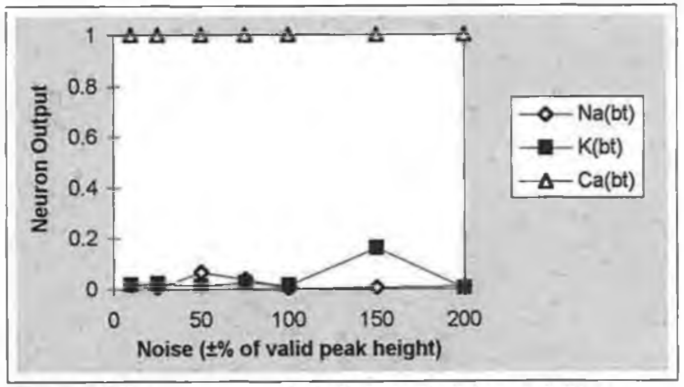


Figure 2.9c Solution containing calcium
Desired Output: Na neuron =0, K neuron =0, Ca neuron =1

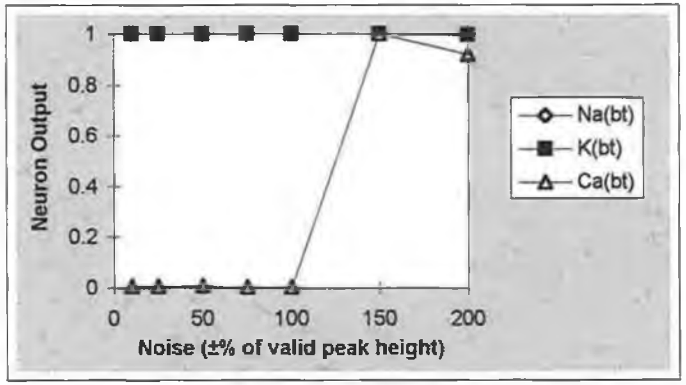


Figure 2.9d Solution containing sodium and potassium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =0

Figures 2.9a to 2.9g continued The outputs from the Na(bt), K(bt) and Ca(bt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 and a maximum total network error of 0.01 with the distorted FIA patterns depicted in appendix 5, when the network is tested with the patterns distorted by noise addition.

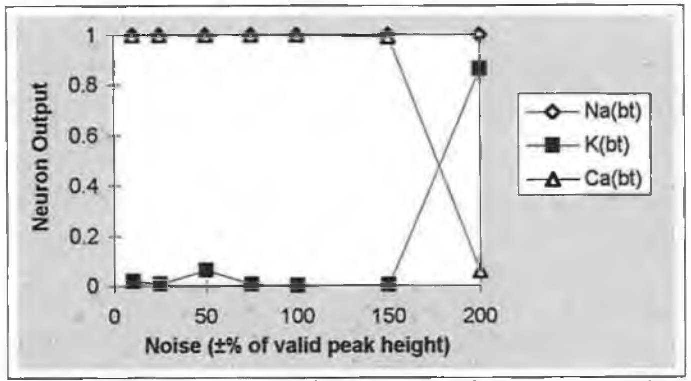


Figure 2.9e Solution containing sodium and calcium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =1

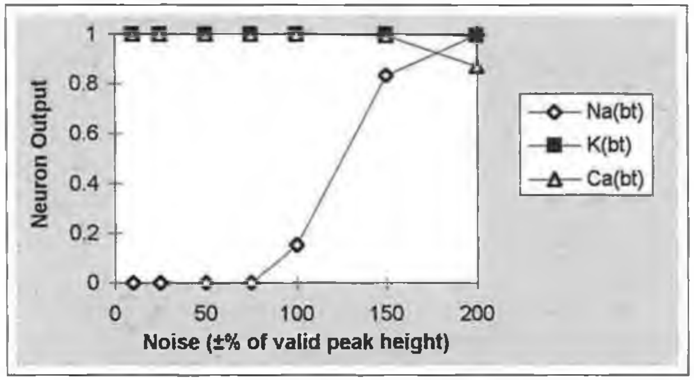


Figure 2.9f Solution containing potassium and calcium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =1

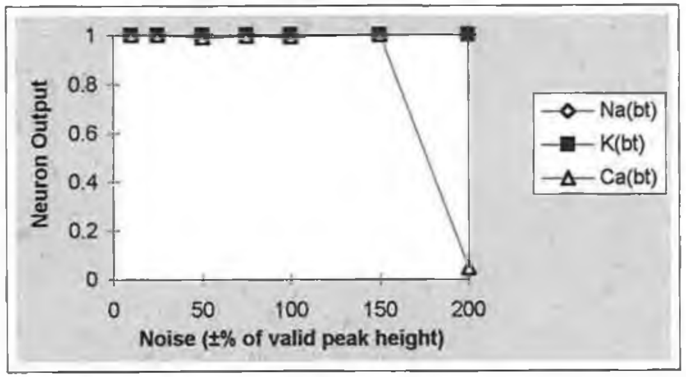


Figure 2.9g Solution containing sodium, potassium and calcium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =1

Test 2: Variable Peak Heights

Figures 2.10a to 2.10b graphically depict the results of this study for a network with 55 neurons in its hidden layer trained with the patterns described in table 2.6 to a maximum output error of 0.1. Training was performed with a learning rate of 0.5 and a momentum of 0.75. The actual outputs of the network in this study are depicted in table 2.4 in the table appendix.

Comparing figures 2.10a to 2.10g with figures 2.8a to 2.8g (in 2.3.1.4) it can be seen that the network trained with the distorted patterns (see table 2.6) perform much better than the networks trained with the simple FIA patterns (see appendix 1) when confronted with patterns distorted by varying the height of the peak from a species which was present in the solution which generated the FIA trace.

Comparing figures 2.8a and 2.10a it can be seen that the output of the $Na_{(bt)}$ neuron is much higher than the $Na_{(rt)}$ neuron for the S/N ratio of 7.5 (output of $Na_{(bt)}$ = 0.912, output of $Na_{(rt)}$ = 0.113). This indicates that the network trained with the distorted patterns is much more sure of the presence of the sodium than the network trained with the simple FIA patterns even at the lowest S/N ratios used in the study. On increasing the S/N ratio beyond a value of 19, the $Na_{(bt)}$ neuron is totally sure of the presence of sodium and its output is virtually unaffected by further increase in the S/N ratio. This compares with the output of the $Na_{(rt)}$ neuron which can be clearly seen from figure 2.10b to be highly sensitive to increasing S/N ratio.

It can be seen from figure 2.10b that while the output from the $K_{(bt)}$ neuron is very high (0.915) at low S/N ratios (indicating that the network was very sure of the presence of potassium), the output from the $Na_{(bt)}$ neuron was also quite high (0.355), indicating that the network was unsure of the presence of sodium. Moreover the network trained with the simple FIA patterns produced a smaller output for the $Na_{(rt)}$ neuron to this same pattern. However, the outputs from all the neurons trained with the simple FIA patterns were very low to this pattern (see table 2.2 in the table appendix). In this case, the increased output from the $Na_{(bt)}$ neuron at low S/N ratios may be indicative of a general increased sensitivity of the network to positive potential features, including that arising

Figures 2.10a to 2.10 The outputs from the Na(bt), K(bt) and Ca(bt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 with the distorted FIA patterns depicted in appendix 5, when the network is tested with the patterns distorted by reduction of the height of the valid FIA peak.

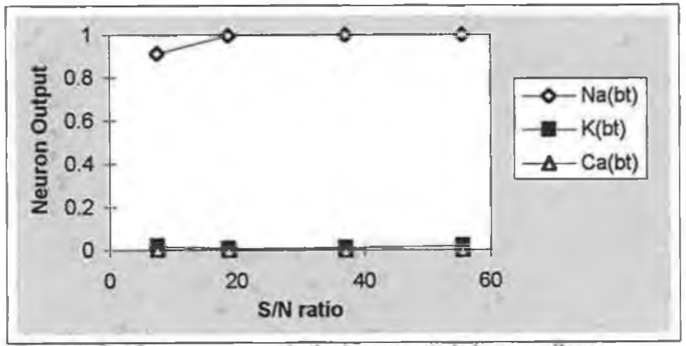


Figure 2.10a Solution containing sodium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =0

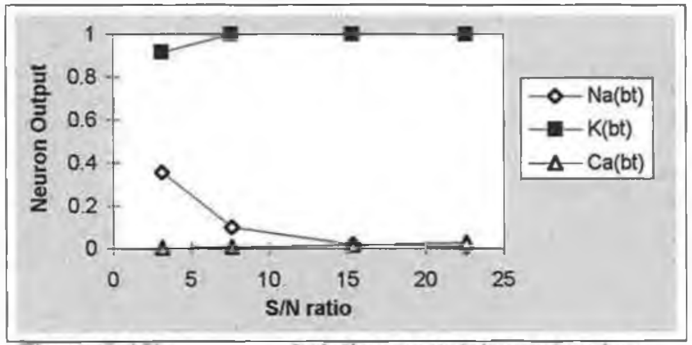


Figure 2.10b Solution containing potassium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =0

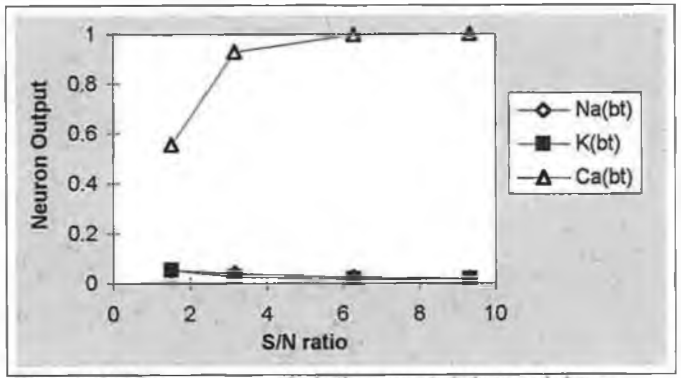


Figure 2.10c Solution containing calcium
Desired Output: Na neuron =0, K neuron =0, Ca neuron =1

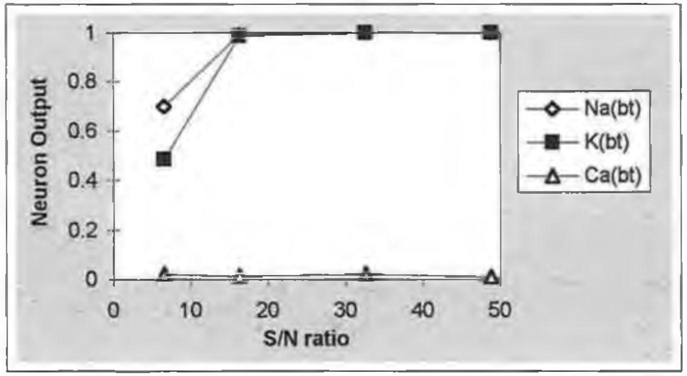


Figure 2.10d Solution containing sodium and potassium
Desired Output: Na neuron =1, K neuron =1, Ca neuron =0

Figures 2.10a to 2.10g continued The outputs from the Na(rt), K(rt) and Ca(rt) neurons of a network with 55 neurons in its hidden layers trained to a maximum output error of 0.1 and a maximum total network error of 0.01 with the simple FIA patterns depicted in appendix 1, when the network is tested with the patterns distorted by reduction of the height of the valid FIA peak.

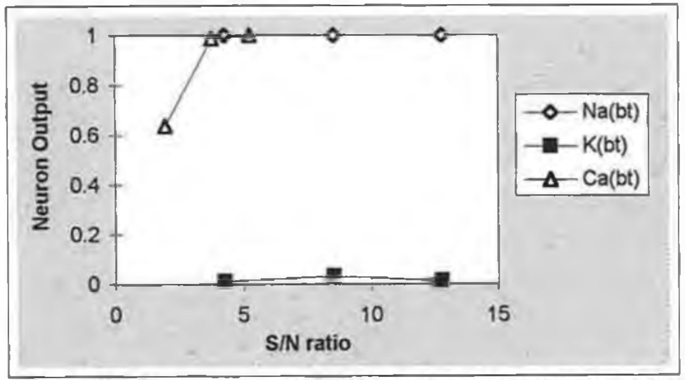


Figure 2.10e Solution containing sodium and calcium
Desired Output: Na neuron =1, K neuron =0, Ca neuron =1

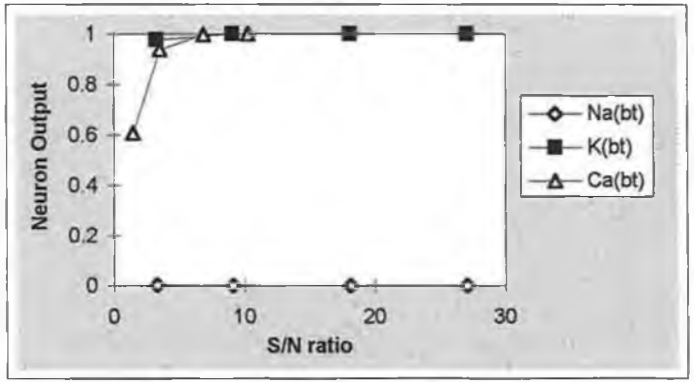


Figure 2.10f Solution containing potassium and calcium
Desired Output: Na neuron =0, K neuron =1, Ca neuron =1

from the contribution of the potassium ion to the response of the sodium ISE. It can be seen from figure 2.10b however that the network is sure of the presence of potassium where a S/N ratio of approximately 7.5 is reached and is virtually sure of the absence of sodium by an S/N ratio of approximately 15.

The increase in output between the $Ca_{(rt)}$ neuron and the $Ca_{(bt)}$ neuron for the S/N ratio of 1.5 is not as large as that seen between the $K_{(bt)}$ neuron and the $K_{(rt)}$ neuron (i.e., $Ca_{(bt)}$=0.554, $Ca_{(rt)}$=0.104 [S/N ratio ~ 3]; $K_{(bt)}$ =0.915, $K_{(rt)}$ = 0.152). This may be due to the different magnitudes of the peak from a calcium ISE and the sodium or potassium ISEs (due to their different slopes).

If an output from a neuron of 0.5 or greater corresponds to a classification that a species is present and an output of less than 0.5 indicates that the species is absent then it could be said that the network trained on the patterns in table 2.8 performed well to a reduction to 10% of the original peak heights of the peaks in the FIA patterns. This compares with the performance of the network trained with the simple FIA patterns which was capable of classifying patterns correctly according to these criteria to a reduction to 50% of the original peak heights in the FIA patterns.

Test 3: Baseline Shifting

Table 2.14 depicts the results of this study for a network with 55 neurons in its hidden layer trained with the patterns described in table 2.8 to a maximum output error of 0.1. Training was performed with a learning rate of 0.5 and a momentum of 0.75.

It can be seen by comparing this table with table 2.5 that a network trained with the distorted patterns (described in table 2.6) performs much better than a network trained with the simple FIA patterns (described in table 2.1) when confronted with patterns distorted by baseline shifting. This is particularly evident in the outputs of neurons corresponding species which were not present in the solutions which produced the associated FIA traces. For example the output of the $Ca_{(rt)}$ neuron to pattern 1 is 0.321 (see table 2.5), whereas the output from the $Ca_{(bt)}$ neuron to the same pattern is 0.000 (desired output = 0.0). Similarly the output of the $K_{(rt)}$ neuron to pattern 9 is 0.760

whereas the output from the $K_{(bt)}$ neuron to the same pattern is 0.008 (desired output = 0.0).

A slight increase in the outputs of neurons corresponding to species which were actually present in the solutions which generated the original FIA peaks is also observed. For example the output of the $Ca_{(rt)}$ neuron to pattern 9 is 0.835 (table 2.5) whereas the output of the $Ca_{(bt)}$ neuron to the same pattern is 0.998 (desired output = 1.0). These results suggest that a network trained with patterns distorted by noise addition, baseline shifting and peak variation were less sensitive to the position of the baseline in the potential range with which the patterns were acquired compared to a network trained with simple FIA patterns.

			$Na_{(bt)}$		$K_{(bt)}$		$Ca_{(bt)}$	
Pattern Number	Solution Key	Baseline Shift	Desired Output	Actual Output	Desired Output	Actual Output	Desired Output	Actual Output
1	Na	+50 mV	1	1.000	0	0.044	0	0.000
2	Na	-40 mV	1	1.000	0	0.034	0	0.000
3	K	+50 mV	0	0.006	1	1.000	0	0.033
4	K	-40 mV	0	0.004	1	1.000	0	0.029
5	Ca	+50 mV	0	0.014	0	0.022	1	1.000
6	NaK	+50 mV	1	1.000	1	1.000	0	0.006
7	NaK	+100 mV	1	1.000	1	1.000	0	0.010
8	NaCa	+50 mV	1	1.000	0	0.009	1	0.999
9	NaCa	+100 mV	1	1.000	0	0.008	1	0.998
10	KCa	+50 mV	0	0.002	1	1.000	1	1.000
11	KCa	+100 mV	0	0.005	1	1.000	1	1.000
12	NaKCa	+50 mV	1	1.000	1	1.000	1	0.998
13	NaKCa	+100 mV	1	1.000	1	1.000	1	0.992

Table 2.14 - Classification results from a network with 55 neurons in its hidden layer trained as described in the text with the patterns in table 2.8 to a test set composed of the FIA patterns depicted in appendix 4.

2.3.4 A Comparison Of Neural Network And Statistically Based Pattern Recognition Techniques

Lippmann (23) discussed the similarities between the operation of different neural network models and the clustering and classification activities of traditional statistical pattern recognition techniques. In the sense that its use yields the lowest probability of committing classification errors on average, Baye's decision rule provides an optimal statistical classification rule. However many statistical techniques based on this rule make assumptions concerning the distributions of the joint probability density functions p(x, c_i) ,i.e. the probability density that a pattern be x and be in the class c_i (24-25).

The technique of linear discriminant analysis in which patterns are classified according to whether they are on one side or another of a set of hyperplanes makes the assumption that the joint probability density functions for the different classes have normal distributions with equal variances. Techniques which classify patterns according to which distribution centre or cluster centre is nearest requires that the distributions of the joint probability density functions of all the classes have equal variance and equal probabilities that a pattern belongs to each class regardless of the identity of the class.

Neural networks make fewer assumptions and as such may prove to be more robust than these statistical techniques when distributions are generated by non-linear processes or are strongly non-Gaussian (23, 27). Whilst both linear discriminant analysis (LDA) and feedforward networks use a supervised approach to the development of their discriminants between the different classes, LDA is a parametric technique whereas ANNs are non-parametric. However, as has been discussed previously in chapter 1, the search space of a neural network can be very complex and as such it can be difficult to guarantee that a network has converged to a global minimum on the error surface when it has finished training, whereas if the assumptions of normality and equal variances are true for the data being studied, the statistical techniques derived from Baye's rule are guaranteed to find the optimal classification rule for the data. Smits et al. (26) discussed the performance of LDA and ANNs on the classification of data containing outliers. It was shown that ANNs can cope better with outliers in the training data than LDA because LDA tends to focus on the mean pattern in each class. Hence patterns deviating from this mean cause deterioration of the LDA performance, whereas an ANN at the end of its training focuses on boundary objects and as such is not as impeded by outliers.

Unfortunately multivariate statistical pattern recognition techniques were not applied to the patterns in this study. Without the results of such a study it is not possible to directly compare the performance of the neural network and the more conventional approaches. The degree of distortion of the patterns used for training (broad training set) and testing the networks was very severe and it might be expected that conventional statistical approaches would have difficulties with these distortions.

2.3.5 Using NeuralWorks II

Following the apparent success of the pattern classification demonstrated by the neural networks trained using the NT5000 software, it was decided to investigate the learning process more closely using a different type of neural network simulation software namely NeuralWorks Professional II which would provide more information concerning processes occurring within the network.

Using this software it was possible to study the MSE on the training and testing sets during the training period, it was also possible to save the connection weights at intervals during training. It was decided to investigate some of the processes occurring in the neural nets during training by studying the lengths of the vectors represented by the connection weights between the input and hidden layers and hidden and output layers, and also to study the variation of the angle formed between these vectors and the unit vector during training.

If all the connection weights from one layer to another are treated as a single vector, then its length can be described by :

$$\|\mathbf{W}\| = \sqrt{\sum_{\mathbf{i}}\sum_{\mathbf{j}} \mathbf{W}_{\mathbf{ij}}^{2}} \qquad \textbf{(2.7)}$$

Where W_{ij} describes the connection weight between a neuron I and a neuron j in another layer (alternatively described by the vector W), $\|\mathbf{W}\|$ is the length of the vector referring to the connection weights between one layer and another. The length of the vector U with unit co-ordinates is given by $\sqrt{n}$, (n being the dimension of the weight vector)

From the definition of the dot product of two vectors:

$$\mathbf{U.W} = \|\mathbf{U}\|\|\mathbf{W}\|\cos\theta \qquad \textbf{(2.8)}$$

$$\cos\theta = \frac{\mathbf{U}.\mathbf{W}}{\|\mathbf{U}\|\|\mathbf{W}\|} = \frac{\sum_i\sum_j \mathbf{W}_{ij}}{\sqrt{n}\sqrt{\sum_i\sum_j \mathbf{W}_{ij}^2}} \qquad \textbf{(2.9)}$$

$$\theta = \cos^{-1}\left(\frac{\sum_i\sum_j \mathbf{W}_{ij}}{\sqrt{n}\sqrt{\sum_i\sum_j \mathbf{W}_{ij}^2}}\right) \qquad \textbf{(2.10)}$$

θ refers to the angle between the vector describing the connection weights between one layer and another (W) and the vector with unit co-ordinates (U).

2.3.5.1 Comparison of training and testing set errors

Figure 2.11 shows the variation of MSE values from the training and testing sets with the number of presentations of the training set for a backpropagation net with 20 neurons in its hidden layer. It can be seen that the MSE values from both sets of patterns undergo their most rapid decrease within 100 epochs after which the decrease in the MSE value is more gradual.

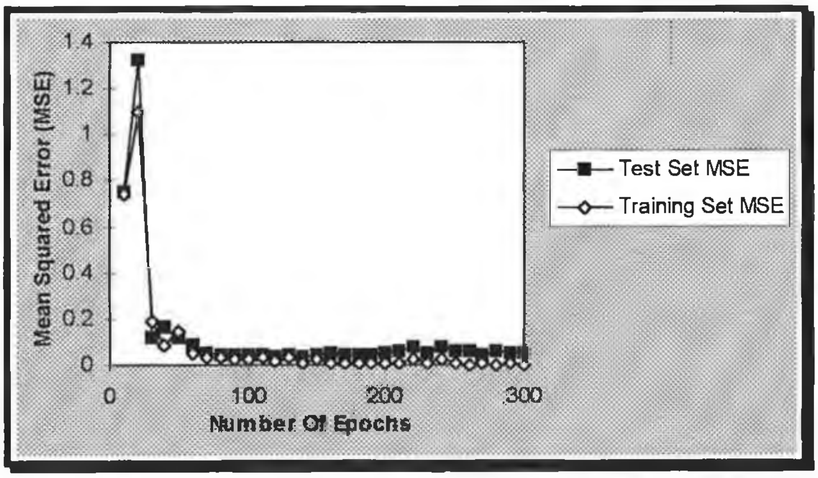


Figure 2.11 - Variation during training of MSE from the patterns used to train (see table 2.6) and test (see tables 2.7-2.13) a network with 20 neurons in its hidden layer, trained with a learning rate of 0.5 and a momentum of 0.75 using the NeuralWorks Software

Figure 2.12 zooms in on the later stages of training of this net and demonstrates that while it appears from figure 2.11 that there is not much change in the MSE after 100 epochs that in actuality the MSE of the training set is continuing to decrease (because the network is training on this set of patterns) while the MSE of the test set is beginning to increase. This effect is a feature of overtraining.

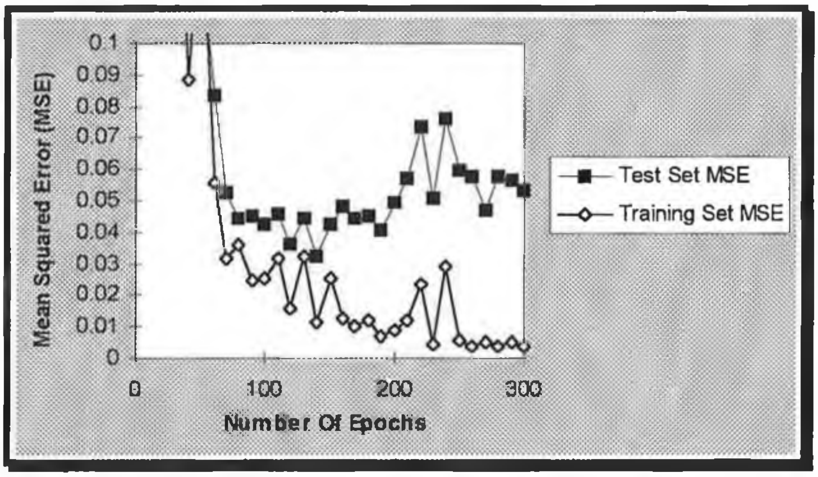


Figure 2.12 - Variation during the later stages of training of MSE from the patterns used to train (see table 2.6) and test (see tables 2.7-2.13) a network with 20 neurons in its hidden layer, trained with a learning rate of 0.5 and a momentum of 0.75 using the NeuralWorks Software.

It was observed that overtraining or the potential for overtraining was a problem for all the neural nets studied. For some of the nets the overtraining was much more obvious and occurred before the convergence point defined in the NT5000 software, indicating the difficulties associated with the interpretation of the MSE of the test set as a function of the number of hidden layer neurons.

In addition to the overtraining problem there was some oscillation in the MSEs of the test set in the later stages of training which, while being very small relative to the error at the start of training, made direct comparison of the networks in terms of their MSEs after a fixed number of iterations difficult. However, it was observed that were differences between the variations and values of the MSEs between the NT5000 software and the NeuralWorks II software. This may be due to variations in the random values of the connection weights at the start of training.

2.3.5.2 Variation of vector length and angle formed with the unit vector during training

Figure 2.13 shows how the lengths of the vectors corresponding to the connection weights between the input and hidden layers and hidden and output layers varied during training for a backpropagation net with 25 neurons in the hidden layer.

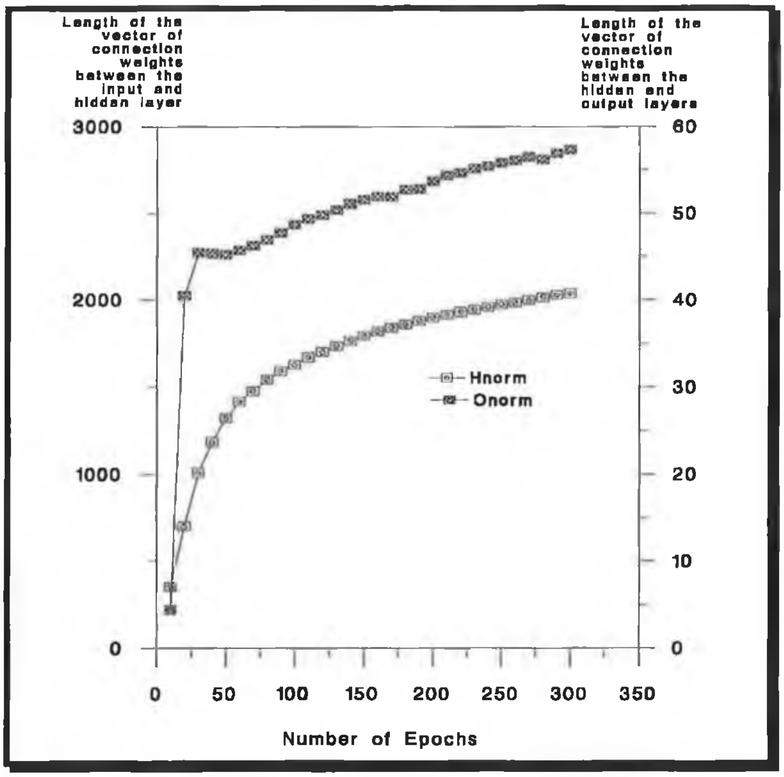


Figure 2.13 - Variation during training of the length of the vector corresponding to the connection weights between the input and hidden layer (Hnorm in the graph) and hidden and output layer (Onorm in the graph) for a network with 25 neurons in its hidden layer. The network was trained with the patterns described in table 2.6 with a learning rate of 0.5 and momentum of 0.75 with the NeuralWorks software.

Figure 2.14 shows how the angles between the unit vector and the vectors corresponding to the connection weights between the input and hidden layers and hidden and output layers varied during training for the backpropagation net in figure 2.13.

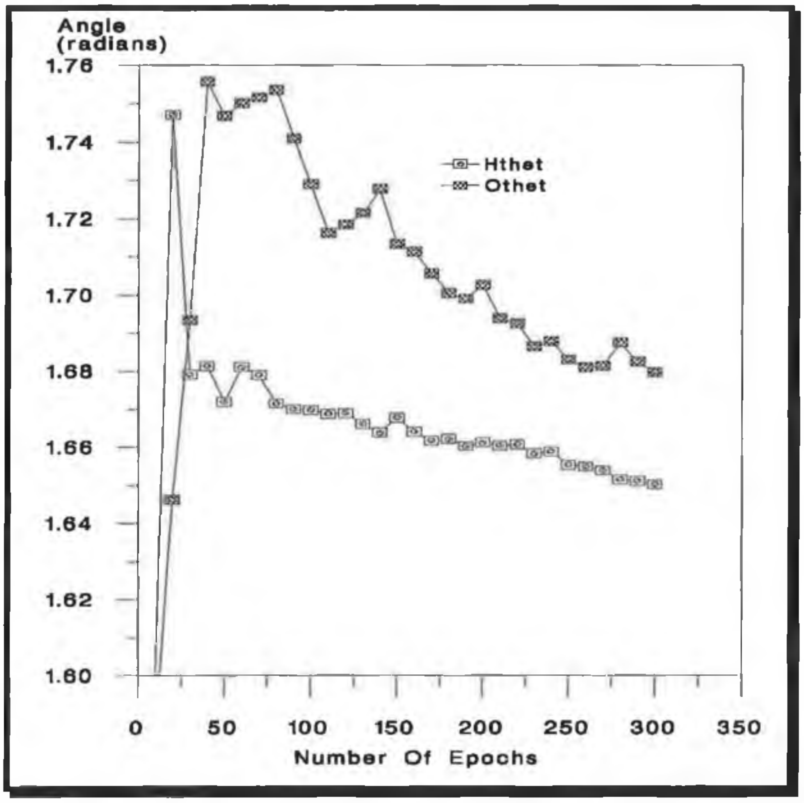


Figure 2.14 - Variation during training of the angle between the vector corresponding to the connection weights between the input and hidden layer (Hthet in the graph) and hidden and output layer (Othet in the graph) and the vector with unit co-ordinates for a network with 25 neurons in its hidden layer. The network was trained with the patterns described in table 2.6 with a learning rate of 0.5 and momentum of 0.75 with the NeuralWorks software.

It can be seen in figure 2.13 that the length of the input to hidden layer vector was greater than the length of the hidden to output layer vector throughout the training process, which can be attributed to the greater dimensionality of the first vector. Similarly it can be seen in figure 2.14 that the hidden to output layer vector made a larger angle with the unit vector than did the input to hidden layer vector.

The initial rapid decrease in MSE during training is characterised by a rapid increase in the length of the hidden to output layer vector in figure 2.13, whereas the input to hidden layer vector length tends to display a smooth continuous trend throughout training. The angle that the input to hidden layer vector makes to the unit vector in figure 2.14 forms a spike during this period, while the hidden to output layer vector angle increases slowly. This suggests that during the initial stages of training that the input to hidden layer vector gradually increases in length but fluctuates in direction, whereas the input to hidden layer vector shows a rapid change in length and direction. After this initial period of rapid change the length of the hidden to output layer vector continues to increase, but at a much slower rate than before and the input to hidden layer vector continues its smooth increase in length.

Those networks which displayed signs of overtraining also suggested some interesting properties of their vector lengths and angles. Figure 2.15 shows how the angle formed between the unit vector and the vectors corresponding to the input to hidden layers and hidden to output layers vary during training for the network whose MSE change is shown in figure 2.11.

As can be seen in figure 2.11 a net with 20 neurons in its hidden layer was clearly overtraining after about 120 iterations. This period, in which the model for the training set was improved to the detriment of the test set, was associated with a rapid decrease in the angle that the hidden to output layer vector made with the unit vector compared to the rate of decrease of the angle formed by the hidden to output layer vector (figure 2.15).

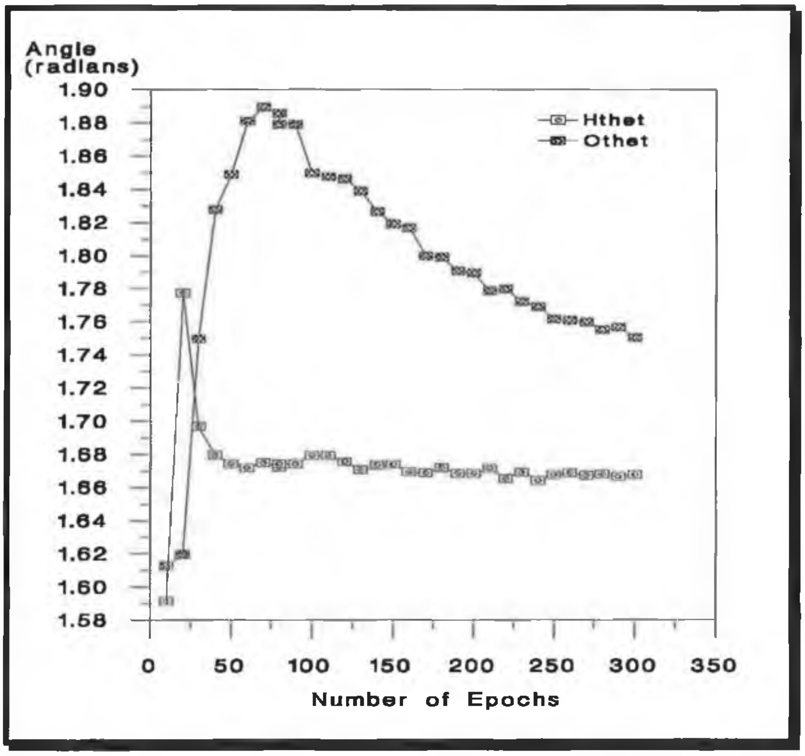


Figure 2.15 - Variation during training of the angle between the vector corresponding to the connection weights between the input and hidden layer (Hthet in the graph) and hidden and output layer (Othet in the graph) and the vector with unit co-ordinates for a network with 20 neurons in its hidden layer. The network was trained with the patterns described in table 2.6 with a learning rate of 0.5 and momentum of 0.75 with the NeuralWorks software.

Figure 2.16 shows the variation of MSE of the training and testing sets during training for a backpropagation net with 45 neurons in the hidden layer. As can be seen from this figure the MSE on the training set is improving as the magnitude of the oscillations in the MSE of the test set are reducing during the later stages of training.

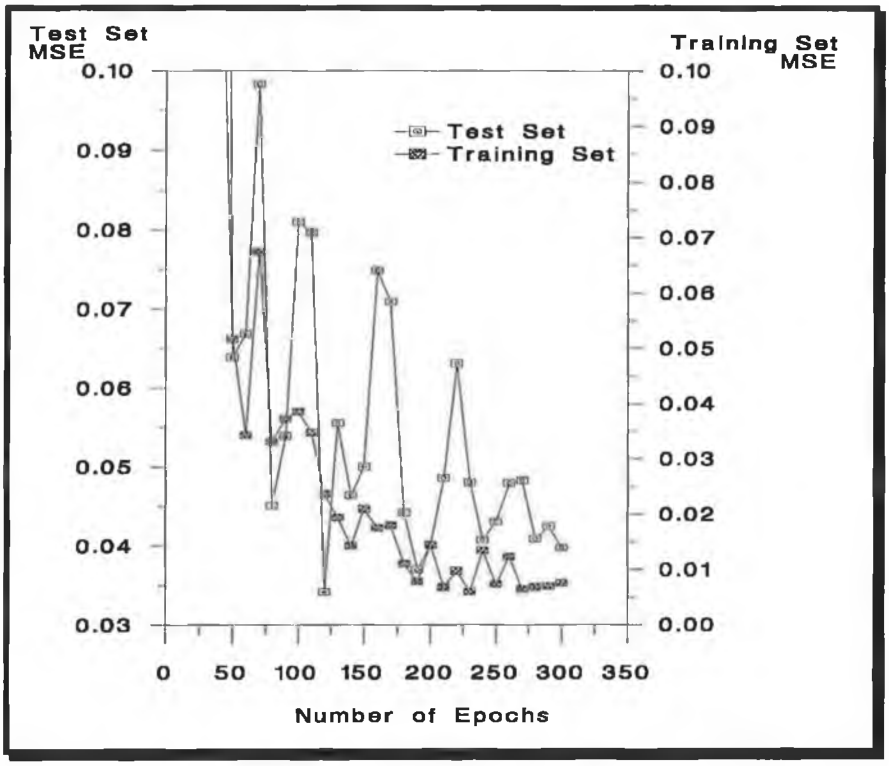


Figure 2.16 - Variation during the later stages of training of MSE from the patterns used to train (see table 2.6) and test (see tables 2.7-2.13) a network with 45 neurons in its hidden layer, trained with a learning rate of 0.5 and a momentum of 0.75 using the NeuralWorks Software.

Figure 2.17 shows how the lengths of the vectors corresponding to the connection weights between the input and hidden layers and hidden and output layers vary during training for the same net .

As with the other networks studied, two distinct regions can be identified in the variation of the hidden to output layer vector length during training. It can be seen that the normal rapid increase in the vector length during the early stages of training is present, followed by a slower increase in the vector length while the network model of the training set is being more gradually improved, until about 130 presentations of the training set, at which point the vector length appears to be increasing more rapidly.

This suggests the presence of another period of change in the vector length corresponding to the period in the training of the network in which the network model stabilises.

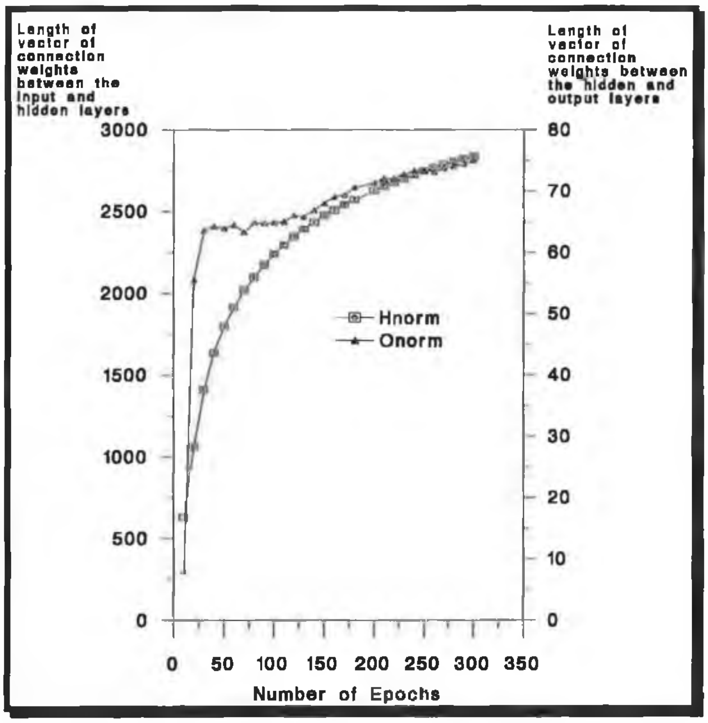


Figure 2.17 - Variation during training of the length of the vector corresponding to the connection weights between the input and hidden layer (Hnorm in the graph) and hidden and output layer (Onorm in the graph) for a network with 45 neurons in its hidden layer. The network was trained with the patterns described in table 2.6 with a learning rate of 0.5 and momentum of 0.75 with the NeuralWorks software.

Figure 2.18 shows how the angles formed between unit vector and the vectors corresponding to the connection weights between the input and hidden layers and hidden and output layers vary during training for this network.

The three regions already seen in figure 2.17 can also be seen in figure 2.18, which shows how the angles formed between unit vector and the vectors corresponding to the connection weights between the input and hidden layers and hidden and output layers vary during training for this network. A rapid increase in the angle is seen during the initial stage of training, followed by a more gradual increase corresponding to the second region in the variation of vector length plot and followed again by a region where the angle gradually decreases, corresponding to the stabilisation of the network model.

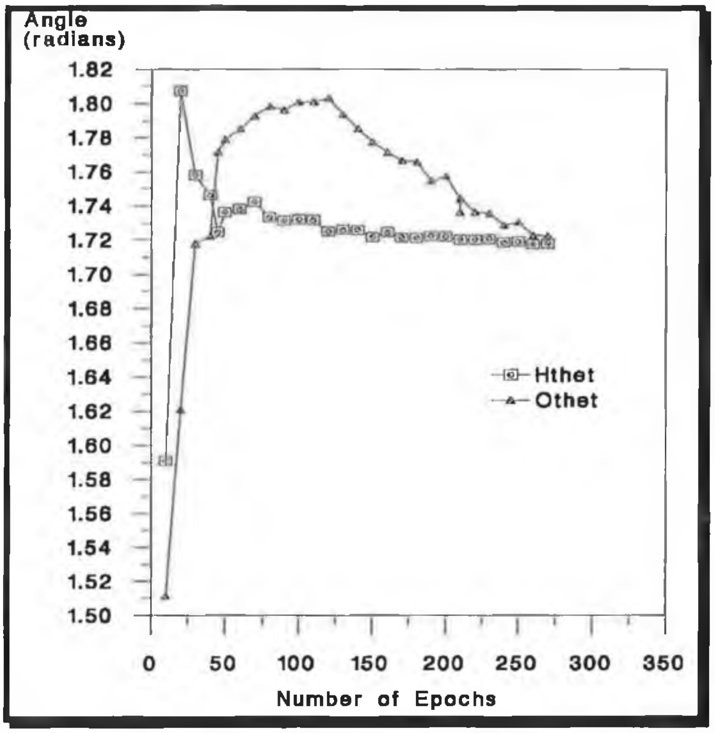


Figure 2.18 - Variation during training of the angle between the vector corresponding to the connection weights between the input and hidden layer (Hthet in the graph) and hidden and output layer (Othet in the graph) and the vector with unit co-ordinates for a network with 45 neurons in its hidden layer. The network was trained with the patterns described in table 2.6 with a learning rate of 0.5 and momentum of 0.75 with the NeuralWorks software.

2.4 Summary

Backpropagation networks have been investigated for the detection and identification of metal ions in solution based on the transient response profiles of ion selective electrodes to these ions when they are injected into a flowing stream. The effects of distorting the patterns on the ability of the networks to perform their identification task was studied, and training and testing sets were devised to consider these deleterious effects. The networks performed well on the test sets even at distortion levels much higher than those normally found in real-life measurements.

Networks trained on a patterns containing typical distortions performed better than those trained on simple patterns, when tested on patterns containing similar distortions, indicating the importance of the choice of the training set for a network.

The same testing and training sets were used to investigate the training process further using a second type of neural network software. There were some variations between the results produced by the two pieces of software which was attributed to possible differences in the initial connection weights. The processes occurring within some networks during training were investigated by studying the variation of the lengths of the vectors representing the connection weights between the input and hidden layers and the hidden and output layers and also by studying the variation of the angle formed between these vectors and the unit vector as training progressed.

2.5 Discussion

This investigation has studied some of the features associated with the use of feedforward neural networks to the recognition of patterns from arrays of ISEs used in an FIA system. However, there are a number of failings in this investigation which limit the scope of its conclusions. The biggest problem with these studies has arisen from the limited availability of experimental data with which to construct the training and testing sets for the networks. These studies were performed with eight experimentally acquired patterns, the training and testing sets were then composed of distortions of these patterns. Ultimately however, the conclusions drawn from these distorted patterns are limited by the fact that the training and testing sets are derived from the same set of experimental data.

Another feature related to this problem arises from the dimensionality of the patterns studied and the topologies of the networks used. In general, the networks used for these studies were oversized for the number of available training patterns. This is especially evident in the studies in 2.3.1.3 to 2.3.1.5. In these studies the eight original FIA patterns (described in table 2.1) were used to train networks with 55 neurons in the hidden layer. This implies a total of 240 (number of input layer neurons) *55 (number of hidden layer units) +55 (bias) connection weights between the input and hidden layer (i.e. 13255 connection weights) and 55 (number of hidden layer neurons) *3(number of output layer neurons) +3(bias) connection weights between the hidden and output layers (i.e. 168 connection weights), leading to a total of 13423 connection weights in the network. It is very clear that the number of training patterns available (8) are inadequate for the task of training a network of this size. It is very probable that a network trained in this fashion would encode the specific training patterns in its connection weights effectively forming a "look-up" table for the patterns. While the networks trained with the broad training set had 70 patterns available for training, these patterns were derived from the same original eight FIA patterns. Given this fact, it is interesting that a network trained on this broad data set performed better than a network trained on the original FIA traces when confronted with test sets containing similar distortions.

Another issue to consider relates to the nature of the distortions used for these studies. These distortions were simulated and applied digitally to the original FIA patterns. The levels of the distortions used would not be found in practical situations and are so extreme (e.g. noise at 200% of the peak height) to make the conclusions drawn from them of limited practical use.

The discussion section of this thesis deals with the issues associated with the choice of different thresholds (on the outputs of the output neurons) for classifying patterns. It is necessary however, to clarify the choice of different thresholds in different studies of this investigation. In sections 2.3.1.4, 2.3.1.6 and 2.3.3 there is reference made to a threshold of 0.5, i.e. the two thresholds for deciding the presence or absence of a species were merged into one threshold such that if the output of a unit was greater than 0.5, the species was classified as being present, if the output was less than 0.5 then the species was classified as being absent (i.e. no uncertainty region). In sections 2.3.2, a lower threshold of 0.1 and an upper threshold of 0.9 were used such that if a neuron's output was less than 0.1 the corresponding species was considered absent if the output of the neuron was greater than 0.9 the species was considered present. The reasons for the use of the two different sets of thresholds is that they were being used for two different studies with different objectives in mind. In sections 2.3.1.4 and 2.3.1.6 the study's objective was to investigate how a network trained on a restricted set of data (given that the network was overdetermined) would perform when confronted with a set of progressively distorted data. It was expected that a network trained in this fashion would not be able to classify the patterns and as such the tightness of the requirements for classification were relaxed by broadening the classification output ranges for each conclusion (to 0.49 for each conclusion). Section 2.3.3 compares the performance of a network trained on a broader training set to classify the same test set of distorted patterns. In order to compare the classification performance of the networks it was necessary to use the same threshold criteria. Section 2.3.2 describes a completely different study, its objectives were to study the performance of a network trained on the broad data set when tested with a similarly distorted testing set. It would be desired and expected that a network trained in this fashion would be able to classify the testing set. As such the requirements for definitive classification were tighteened by reducing the output ranges for definitive classification to 0.1 for both classification decisions. In addition the relative merits of the classification in this study are described by ratioing the outputs of the neurons.

Section 2.3.5 refers to another completely different study, the purpose of this study was to investigate some of the internal processes occurring within a network during training. The motivation behind the choice of network topologies invetsiagted in this study had very little to do with the previuos studies described in this cahpter. Instead, the networks discussed in this section were chosen because the results of the studies demonstrated some interesting features such as overtraining. This is not too say that the other network topologies would not demonstrate this behaviour, rather that in the study described this behaviour was not as clear. Further studies in this area might yield information concerning the role of the topology of a network on this behaviour. While some interesting features were observed, they do not have an immediate impact or use for those employing ANNs. However, it is hoped that it may contribute to future research which may provide some practical help for users of ANNs.

In terms of future research, it would be desirable to continue this work with training sets composed of more experimental data. A particularly useful approach would be to acquire patterns from the ISEs over a range of concentrations of the cations particularly because the selectivity of an ISE against interferences (as determined from the selectivity coefficients in the Nikolskii-Eisenman expression **(2.5)**) is dependent on the activity of the interfering ion (when the charges of the primary and interfering ions are dissimilar) (28).

It would also be of interest to study the effect of different schemes of presenting the patterns to the network. In these studies a crude approach to presentation was taken whereby the potential (mV) responses of the electrodes were prescaled and presented to the network. Alternative approaches for presenting data to a network are described more fully in the discussion section of this thesis.

2.6 Bibliography

1. J. Ruzicka and E.H. Hansen, **Anal. Chim. Acta**, 87, 1976, 353-363
2. D. Betteridge and J. Ruzicka, **Talanta**,23, 1976, 409
3. E.H. Hansen, J. Ruzicka and B. Rietz, **Anal. Chim. Acta**, 89, 1977, 241-254
4. L. Ilcheva and K. Cammann, **Fresenius Z. Anal. Chem**., 322, 1985, 323-326
5. W. Frenzel and P. Bratter, **Anal. Chim. Acta**, 188, 1986, 151-164,
6. R.J. Forster and D. Diamond, **Anal. Chem.**, 64, 1992, 1721-1728,
7. J. Ruzicka and E.H. Hansen, **Flow Injection Analysis**, Chemical Analysis Series Volume 62, Wiley Interscience Publication, N.Y. 1981.
8. M. C. Valcarcel and M.D. Luque de Castro, **Automatic Methods Of Analysis**, Elsevier, Amsterdam, 1988
9. A.W. Adamson, **A Textbook Of Physical Chemistry (2nd Edition)**,[section 12.7], Academic Press N.Y., 1979
10. A.L. Horvath, **Handbook of Aqueous Electrolyte Solutions: Physical Properties, Estimation and Correlation Methods**, p. 213, Ellis Horwood Ltd., Chichester, W. Sussex, England, 1985
11. F.Regan, **Applications and Characteristics of Polymer Membrane Ion Selective Electrodes**, Chapter 1, Resistance Measurements As A Simple Diagnostic Tool For Ion-Selective Electrode Performance, M.Sc. Thesis, Dublin City University , 1990

12 R.A. Durst in H. Freiser (Editor), **Ion-Selective Electrodes In Analytical Chemistry (Volume 1)**, Chapter 5 Sources Of Error In Ion-Selective Electrode Potentiometry, Plenum Press N.Y., 1981

13. W.E. Morf, **The Principles of Ion-Selective Electrodes and of Membrane Transport, Studies in Analytical Chemistry (Volume 2)**, Elsevier, Amsterdam, 1981

14. H.H. Willard, L.L. Merritt Jr., J.A. Dean and F.A. Settle Jr., **Instrumental Methods of Analysis (7th edition)**, Wadsworth Publishing Company, Belmont, California, 1988

15. A.M. Cadogan, D. Diamond, M.R. Smyth, M. Deasy, M.A. McKervey and S.J. Harris, **Analyst**, 114, 1989, 1551-1554,

16. A. Bos, **Neural Networks as a Tool in Chemometrics**, Ph. D. Thesis, University of Twente, Enschede 1993.

17 A. Kandel, **Fuzzy Techniques In Pattern Recognition**, Wiley N.Y., 1982.

18. B. Kosko, **Neural Networks and Fuzzy Systems: A Dynamical Systems Approach To Machine Intelligence**, Prentice Hall, Englewood Cliffs, N.J., 1992

19. R. Hecht-Nielsen, **Theory of the backpropagation neural network** chapter III 3, p.65-93 in **Neural Networks for Perception Volume 2, Computation, Learning and Architectures**, H. Wechsler (Editors), Academic Press Inc 1991

20. T. Tollenaere, **Neural Networks**, 3, 1990, 575

21. R.J. Forster, F.Regan and D. Diamond, **Anal. Chem**., 63, 1991, 876-882

22. K. Cunningham, G. Svehla, S.J. Harris and M.A. M^{c}Kervey, **Analyst**, 118, 1993, 341-345

23. R.P. Lippmann, **IEEE ASSP magazine**, April 1987, 4-22

24. M.James, **Pattern Recognition**, BSP Professional Books, U.K. 1987

25. Y.-H. Pao, **Adaptive Pattern Recognition and Neural Networks**, Addison Wesley Publishing Co. Inc., U.S. 1989

26. J.R.M. Smits, W.J. Melssen, L.M.C. Buydens and G. Kateman, **Chemometrics And Intelligent Laboratory Systems**, 22, 1994, 165-189

27. B.J. Wythoff, **Chemometrics and Intelligent Laboratory Systems**, 18, 1993, 115-155

28. F.J. Sáez de Viteri and D. Diamond, **Analyst,** 119, 1994, 749-758

Chapter 3 : Potentiometric Non-Linear Multivariate Calibration With Genetic Algorithm and Simplex Optimisation

Abstract

In this chapter a genetic algorithm (GA) and a modified simplex technique are investigated as a means of developing non-linear multivariate calibration models for an array of ion-selective electrodes. The responses of an array of ammonium, sodium, potassium and calcium selective electrodes employed in a flow injection analysis system were modelled over the concentration range of $1\text{x}10^{-4}$ M to $1\text{x}10^{-2}$ M using the GA and simplex techniques to optimise the cell potentials, slopes and selectivity coefficient parameters of the Nikolskii-Eisenman equation for each electrode. Correlations between activities predicted from the calibration model and the actual activities of the solutions presented to the array ranged from 0.98 to 0.88 for the four ions.

A variety of different modifications to the simple genetic algorithm (SGA) configuration are investigated, including fitness and rank based scaling, roulette wheel and stochastic remainder sampling and post-hybridisation of the genetic algorithm with the simplex technique.

3.1 Introduction

In this contribution, we consider the application of GAs to modelling the response of an array of ion-selective electrodes (ISEs) used in a flow injection Analysis (FIA) system. The response of an ISE is described by the well-known Nikolskii-Eisenman expression

$$\mathbf{E}_{ij} = \mathbf{E}_j^o + \mathbf{S}_j \log_{10}\left(\mathbf{a}_{ik} + \sum_{l \neq k} \mathbf{k}_{jkl}^{pot} \mathbf{a}_{il}^{z_k/z_l}\right) + \mathbf{e}_i \qquad \mathbf{(3.1)}$$

where $\mathbf{E_{ij}}$ is the measured potential of the jth electrode to the ith sample , E_j^o is the standard cell potential of the electrode and $\mathbf{S_j}$ is the slope of the electrode or the change in potential of the electrode per decade change in activity of the primary ion in the absence of interferents. The activities of the primary ion **k** is $\mathbf{a_{ik}}$ and the activity of the interfering ions **l** are given by $\mathbf{a}_{il}$. The term $\mathbf{k}_{jkl}^{pot}$ refers to the selectivity coefficient of the electrode against the lth interferent with respect to the primary ion, $\sum_{l \neq k} \mathbf{k}_{jkl}^{pot} \mathbf{a}_{il}^{z_k/z_l}$ represents the error arising from the net contribution of all the interferents and $\mathbf{e_i}$ represents the experimental error.

Previous work of this kind had already been performed using simplex techniques (1-3). However, Betteridge et al. (4) noted that the final result of a simplex is dependent on the estimates of the function variables with which the simplex is initialised. It was also clear that the size of the simplex could have quite a dramatic effect on the final result obtained as the complexity of the surfaces studied grows due to increased possibility of local minima existing. With this problem in mind, it was decided to investigate the use of the GA because it is useful in high dimensional search spaces and makes less assumptions about the search space than strong optimisation methods like the simplex technique (for further details see section 1.6).

A simplified form of the Nikolskii-Eisenman expression was used for modelling the responses of the ISEs in these studies. This simplified expression is described in equation **3.2.**

$$\mathbf{E = E^{o} + S\log\left(a_{i} + \sum_{j} K^{*}_{ij} a_{j}\right).} \qquad \textbf{(3.2)}$$

In this expression $\mathbf{K^{*}_{ij}}$ is a "conditional selectivity constant" which is related to the selectivity coefficient defined in equation **3.1** as follows:

$$\mathbf{K^{*}_{ij} a = K^{pot}_{ij} a^{z_i/z_j}} \qquad \textbf{(3.3)}$$

Hence $\mathbf{K^{*}_{ij}=K^{pot}_{ij}}$ when $\mathbf{z_i=z_j}$. As discussed in (1), $\mathbf{K^{*}_{ij}}$ is not a global constant but is rather a constant within the constraints of the conditions of the experiment and calibration design. This investigation involved the use of an array of ammonium, sodium potassium and calcium selective electrodes used in an FIA system. As such, the simplified Nikolskii-Eisneman expression for each electrode is as follows

$$\mathbf{E_{iNH_4^+} = E^{o}_{NH_4^+} + S_{NH_4^+} \log_{10}\left(a_{iNH_4^+} + k^{*}_{NH_4^+Na^+} a_{iNa^+} + k^{*}_{NH_4^+K^+} a_{iK} + k^{*}_{NH_4^+Ca^{2+}} a_{iCa^{2+}}\right)} \qquad \textbf{(3.4)}$$

$$\mathbf{E_{iNa^+} = E^{o}_{Na^+} + S_{Na^+} \log_{10}\left(a_{iNa^+} + k^{*}_{Na^+NH_4^+} a_{iNH_4^+} + k^{*}_{Na^+K^+} a_{iK} + k^{*}_{Na^+Ca^{2+}} a_{iCa^{2+}}\right)} \qquad \textbf{(3.5)}$$

$$\mathbf{E_{iK^+} = E^{o}_{K^+} + S_{K^+} \log_{10}\left(a_{iK^+} + k^{*}_{K^+NH_4^+} a_{iNH_4^+} + k^{*}_{K^+Na^+} a_{iNa^+} + k^{*}_{K^+Ca^{2+}} a_{iCa^{2+}}\right)} \qquad \textbf{(3.6)}$$

$$\mathbf{E_{iCa^{2+}} = E^{o}_{Ca^{2+}} + S_{Ca^{2+}} \log_{10}\left(a_{iCa^{2+}} + k^{*}_{Ca^{2+}NH_4^+} a_{iNH_4^+} + k^{*}_{Ca^{2+}Na^+} a_{iNa^+} + k^{*}_{Ca^{2+}K^+} a_{iK^+}\right)} \qquad \textbf{(3.7)}$$

In these expressions $\mathbf{E_{iNH_4^+}}$, $\mathbf{E_{iNa^+}}$, $\mathbf{E_{iK^+}}$ and $\mathbf{E_{iCa^{2+}}}$ refer to the potential (mV) responses of the ammonium, sodium, potassium and calcium ISEs to the ith sample. $\mathbf{E^{o}_{NH_4^+}}$, $\mathbf{E^{o}_{Na^+}}$, $\mathbf{E^{o}_{K^+}}$ and $\mathbf{E^{o}_{Ca^{2+}}}$ refer to the standard cell potentials of the ammonium, sodium, potassium and calcium ISEs respectively. $\mathbf{S_{NH_4^+}}$, $\mathbf{S_{Na^+}}$, $\mathbf{S_{K^+}}$ and $\mathbf{S_{Ca^{2+}}}$ refer to the slopes of the ammonium, sodium, potassium and calcium ISEs respectively, $\mathbf{a_{iNH_4^+}}$, $\mathbf{a_{iNa^+}}$, $\mathbf{a_{iK^+}}$ and

$\mathbf{a}_{iCa^{2+}}$ refer to the activities of ammonium, sodium, potassium and calcium in the ith sample. $\mathbf{k}^{*}_{NH_4^+Na^+}$, $\mathbf{k}^{*}_{NH_4^+K^+}$ and $\mathbf{k}^{*}_{NH_4^+Ca^{2+}}$ refer to the conditional selectivity coefficient of the ammonium ISE against sodium, potassium and calcium. $\mathbf{k}^{*}_{Na^+NH_4^+}$, $\mathbf{k}^{*}_{Na^+K^+}$ and $\mathbf{k}^{*}_{Na^+Ca^{2+}}$ refer to the conditional selectivity coefficient of the sodium ISE against ammonium, potassium and calcium. $\mathbf{k}^{*}_{K^+NH_4^+}$, $\mathbf{k}^{*}_{K^+Na^+}$ and $\mathbf{k}^{*}_{K^+Ca^{2+}}$ refer to the conditional selectivity coefficients of the potassium ISE against ammonium, sodium and calcium. Similarly $\mathbf{k}^{*}_{Ca^{2+}NH_4^+}$, $\mathbf{k}^{*}_{Ca^{2+}Na^+}$ and $\mathbf{k}^{*}_{Ca^{2+}K^+}$ refer to the conditional selectivity coefficients of the calcium ISE against ammonium, sodium and potassium.

The objective of the GA is to determine the values of the cell potentials, slopes and the conditional selectivity coefficients of each electrode (e.g. for the ammonium ISE, $\mathbf{E}^{o}_{iNH_4^+}$, $\mathbf{S}_{iNH_4^+}$, $\mathbf{k}^{*}_{NH_4^+Na^+}$, $\mathbf{k}^{*}_{NH_4^+K^+}$ and $\mathbf{k}^{*}_{NH_4^+Ca^{2+}}$) which best describes the response of the electrode. This is achieved by using the parameters and the Nikolskii-Eisenman expression for each electrode **(3.4-3.7)** to predict the potential (mV) responses of the electrodes to a series of calibration solutions. The compositions of these calibration solutions are depicted in table 3.1. The predicted potentials are then compared with the experimentally measured values for the potential (mV) responses of the electrodes to the same solutions.

The extent to which the predicted and measured potentials differ is described by a sum squared error (SSE) term defined in equation **3.8**.

$$\mathbf{SSE} = \sum_{s=1}^{N}\left(\frac{\mathbf{E}-\widetilde{\mathbf{E}}}{\mathbf{E}}\right)^{2} \qquad \mathbf{(3.8)}$$

The objective of the SSE is thus to determine the values of the cell potential, slope and conditional selectivity coefficient which minimises the SSE over the calibration solutions described in table 3.1.

To this end the parameters of the simplified Nikolskii-Eisenman expressions were encoded as genes in a simple binary format (see section 1.6 for a discussion of the encoding of parameters on genes and reference 5 for a discussion of different encoding methods). For the purposes of the GA, the fitness of an individual chromosome in the population is inversely related to the magnitude of the SSE (see equation **3.8**) such that the larger the value of the SSE , the less fit the individual is and hence the lower the chances are of it reappearing in the population in the next generation.

While GAs are very robust techniques they also have their limitations, namely poor search precision (6) and premature convergence (5). Poor search precision is used to describe the variation in the final solutions about a global optimum produced on repeating a GA, an effect which is most likely to be due to the random processes occurring within the GA itself. This is less likely to be a problem with strong methods such as simplex optimisation, which are driven by more deterministic rules.

Premature convergence occurs when the finite population used in a GA becomes dominated by a particularly fit individual from an earlier generation by the process of selective reproduction. In this case, the diversity of the population is reduced within a few iterations to such an extent, that crossover can no longer function as a means of searching new areas of the response surface, because the crossover between two chromosomes will have no effect if the chromosomes involved are identical. As such mutation becomes the only way of investigating new areas of the search space. In these cases the GA has effectively become trapped in a local minimum because it has converged to a rapidly to allow for searching of the entire response surface with acceptable resolution.

In order to develop a GA for determining the response parameters (from the simplified Nikolskii-Eisenman expression) for an array of ISEs a number of issues were studied, these include :

(i) An investigation of the effects of linear prescaling, population size and percentage crosspool on the performance of the GA.

(ii) A study of crossover and mutation processes occurring during the operation of a GA with a population of 75 chromosomes, employing a linear prescaling constant of 1.5 and allowing 80% of the population to cross.

Studies (iii) and (iv) were performed to investigate the use of alternative scaling and selection schemes as a means of reducing the problem of premature convergence in the GA.

(iii) A study of the use of rank prescaling

(iv) A study of the use of stochastic remainder sampling and reduction of premature convergence by preventing incest

(v) A study of the use of a simplex technique post-hybridised with the GA at different stages in its operation. The purpose of this study was to address the difficulty of the poor search precision of the GA.

(vii) Having studied different aspects of the use of GAs relevant to the problem of developing a GA for multivariate calibration, it was then necessary to directly apply the adapted GA to the task of optimising the response parameters of the ammonium, sodium, potassium and calcium ISEs using the responses of the ISEs (described in table 3.2) to the solutions (described in table 3.1).

(viii) The values of the response parameters determined by the GA for the four ISEs were compared with the values for the same response parameters described in the literature.

(ix) The performance of the GA was compared with the performance of a simplex technique on the same set of responses of the electrodes.

(x) The dynamics of the search for the response parameters of the sodium ISE.

3.2 *Experimental Details*

The data used for this study were provide courtesy of Mr. F.J. Sáez de Viteri, School of Chemical Sciences, Dublin City University. The data were acquired from an array of electrodes selective for ammonium, sodium, potassium and calcium which were employed in a flow injection system. For a fuller information on the use of sensor arrays see reference 7. The compositions of these calibration solutions are depicted in table 3.1. Table 3.2 depicts the potential (mV) reponses of the ISEs to these solutions. Fuller details of the construction of the electrodes and the flow injection analysis system can be found in (1). The experimental data were acquired via an Analog devices RTI-815 data acquisition card fitted inside an IBM 386 compatible PC. Data acquisition and processing software were written in Microsoft QuickBasic and Visual Basic. The calibration solutions were prepared according to a two level, four factor, partial factorial experimental design, developed in a fashion to evoke a significant response from the electrodes to their interferents in comparison to that of the corresponding primary ions.

Solution	$[NH_4^+]$ (M)	$[Na^+]$ (M)	$[K^+]$ (M)	$[Ca^{2+}]$ (M)	Solution	$[NH_4^+]$ (M)	$[Na^+]$ (M)	$[K^+]$ (M)	$[Ca^{2+}]$ (M)
1	10^{-2}	$9x10^{-3}$	$3x10^{-3}$	$8x10^{-3}$	17	$8x10^{-3}$	$8x10^{-3}$	10^{-2}	$8x10^{-3}$
2	10^{-2}	$3x10^{-3}$	$5x10^{-4}$	$9x10^{-4}$	18	$9x10^{-4}$	$6x10^{-4}$	10^{-2}	$5x10^{-3}$
3	10^{-2}	10^{-3}	$5x10^{-3}$	$7x10^{-4}$	19	$4x10^{-3}$	$8x10^{-4}$	10^{-2}	$7x10^{-4}$
4	10^{-2}	$5x10^{-4}$	$6x10^{-4}$	$3x10^{-3}$	20	$8x10^{-4}$	$3x10^{-3}$	10^{-2}	$9x10^{-3}$
5	10^{-4}	$7x10^{-3}$	$4x10^{-3}$	$4x10^{-4}$	21	$2x10^{-3}$	$9x10^{-4}$	10^{-4}	$9x10^{-3}$
6	10^{-4}	$5x10^{-3}$	$8x10^{-4}$	$2x10^{-3}$	22	$3x10^{-4}$	$4x10^{-3}$	10^{-4}	$3x10^{-3}$
7	10^{-4}	$2x10^{-4}$	$3x10^{-3}$	$6x10^{-3}$	23	10^{-3}	$5x10^{-3}$	10^{-4}	$9x10^{-4}$
8	10^{-4}	$3x10^{-4}$	$2x10^{-4}$	$2x10^{-4}$	24	$2x10^{-4}$	$4x10^{-4}$	10^{-4}	$3x10^{-4}$
9	$9x10^{-3}$	10^{-2}	$9x10^{-3}$	$7x10^{-3}$	25	$8x10^{-3}$	$7x10^{-3}$	$8x10^{-4}$	10^{-2}
10	$9x10^{-4}$	10^{-2}	$6x10^{-3}$	$6x10^{-4}$	26	$5x10^{-3}$	$7x10^{-4}$	$8x10^{-3}$	10^{-2}
11	$4x10^{-4}$	10^{-2}	$7x10^{-4}$	$4x10^{-3}$	27	$3x10^{-4}$	10^{-3}	$4x10^{-4}$	10^{-2}
12	$3x10^{-3}$	10^{-2}	$9x10^{-4}$	$8x10^{-4}$	28	$7x10^{-4}$	10^{-4}	10^{-3}	10^{-2}
13	$5x10^{-4}$	10^{-4}	$2x10^{-3}$	$5x10^{-3}$	29	$6x10^{-3}$	$6x10^{-3}$	$7x10^{-4}$	10^{-4}
14	$6x10^{-3}$	10^{-4}	$7x10^{-3}$	10^{-3}	30	$3x10^{-3}$	$8x10^{-4}$	$3x10^{-3}$	10^{-4}
15	$7x10^{-3}$	10^{-4}	$4x10^{-4}$	$6x10^{-3}$	31	$8x10^{-4}$	$2x10^{-3}$	$2x10^{-3}$	10^{-4}
16	2x10-4	10^{-4}	3x10-4	5x10-4	32	$6x10^{-4}$	$5x10^{-4}$	10^{-4}	10^{-4}

Table 3.1 - Concentrations of ammonium, sodium, potassium and calcium in the calibration solutions (the activities corresponding to these concentrations can be found in table 3.1 in the tables appendix).

Solution	NH_4^+ ISE Potential (mV)	Na^+ ISE Potential (mV)	K^+ ISE Potential (mV)	Ca^{2+} ISE Potential (mV)	Solution	NH_4^+ ISE Potential (mV)	Na^+ ISE Potential (mV)	K^+ ISE Potential (mV)	Ca^{2+} ISE Potential (mV)
1	130.3	141.2	142.7	66.7	17	130.8	142.4	164.6	69.5
2	131.7	118.3	117.2	45.5	18	105.7	101.7	167	62.8
3	132.3	108.2	153.8	40.1	19	126.3	99.7	161.2	38.7
4	133.7	79.3	121.5	57	20	106	124.8	163.8	33.8
5	83.2	149.2	149.5	33.7	21	102	85.5	92.7	61.5
6	59.9	135.9	113	49.2	22	67.8	119.6	70	51.8
7	78.2	78.6	141.1	58.6	23	92.1	125.6	80.7	43
8	48.2	68.9	81.5	23.1	24	58.5	58.5	68.5	28.1
9	130.1	144.1	157.5	59.2	25	131	129.4	122.3	63.8
10	98.9	142.6	154.3	41.6	26	126.1	100.3	160.6	63.4
11	72.5	143.5	107.2	59	27	65.7	88.3	93.6	63.6
12	111.9	145.9	115.9	34.9	28	81.1	58.6	117.9	64.4
13	85.7	64.9	131.7	51.4	29	123.7	148.2	119	20.6
14	131.4	84.9	157.3	40	30	114.2	98.5	145	21.1
15	128.9	49.9	112.9	59.3	31	90.7	119.2	134	22
16	63.6	42.8	97.4	35.6	32	78.8	85.6	76.4	21.5

Table 3.2 - Responses of the ammonium, sodium, potassium and calcium ISE array to the solutions described in table 3.1.

The genetic algorithm, modified simplex (8a) and Gauss Jordan elimination (8b) software were written in Borland Turbo C++ and appear as **[GENCA5.C]**, **[SIMPLEX.C]** and **[INVERT.C]** respectively in the software appendix.

3.3 Results:

3.3.1 Modifications to the SGA

3.3.1.1 Effects of linear prescaling and elitism during crossover

A SGA was first studied in which chromosomes produced offspring in the next generation according to the ratio of the error of the individual chromosome to the average error of the population.

$$\mathbf{Offspring}_i = \frac{\frac{1}{P}\sum_{i=1}^{P}\mathbf{SSE}_i}{\mathbf{SSE}_i} \qquad \mathbf{(3.9)}$$

In which **Offspring$_i$** denotes the number of offspring that will be produced by individual **i** in the next generation, **SSE$_i$** denotes the error of individual **i** and **P** denotes the total number of individuals in the population.

An attempt was made to reduce premature convergence by placing an upper limit on the number of offspring which could be produced by the chromosome with the lowest error. The errors of the chromosomes in the current population were linearly prescaled according to two criteria.

In which **Offspring$_i$** denotes the number of offspring that will be produced by individual **i** in the next generation, **SSE$_i$** denotes the error of individual **i** and **P** denotes the total number of individuals in the population.

An attempt was made to reduce premature convergence by placing an upper limit on the number of offspring which could be produced by the chromosome with the lowest

error. The errors of the chromosomes in the current population were linearly prescaled according to two criteria.

(i) The ratio between the average error of the population and the error of the chromosome which most closely models the behaviour of an ISE remains constant during the entire operation of the GA at a value α. This value α also represents an upper limit on the number of offspring which could be produced by fitness proportional reproduction.

(II) The average fitness of the population is unaffected by the transform. This ensures that a chromosome whose error is close to the average error of the entire population contributes one offspring to the population in the next generation.

Appendix 8 describes the mathematics for the scaling which fulfils these two requirements. This form of scaling eliminates the problems of negatively scaled fitness values which can occur in the procedure described by Goldberg (9) which involves scaling according to the ratio of the performance of the most fit individual to the average performance of the population. Negatively scaled values can occur during the later stages of the operation of a GA (in which case the average population performance is very close to that of the most fit individual in the population) when an individual chromosome in the population has a performance much worse than either the most fit individual in the population or the population average fitness.

In this study however, scaling and reproduction are performed with respect to the ratio of the average population error to the lowest error in the population. The scaling procedure places a lower positive limit on the ratio of individual error to the average population error that prevents any negatively scaled values occurring.

The effect of such linear prescaling is to reduce the differences in the errors of individuals during the early stages of the search and hence prevent domination of the population by particularly fit individuals leading to premature convergence. However during the later stages of the search (at which point there is not a great deal of variation between the errors of different members of the population) prescaling serves to increase the differences in the errors of chromosomes, similarly reducing convergence by increasing the degree of competition between the chromosomes for reproduction in the next generation. Crossing was achieved by sorting the current population according to the SSE of each chromosome. The population was then split such that a certain percentage of the population would be used to generate a crosspool. Chromosomes within this crosspool, were chosen at random as mates and were crossed at random along their bitstrings. The resulting chromosomes were then resubstituted back into the population in place of chromosomes with a higher error. GAs commonly use populations of size 50-500 chromosomes (5). Smaller populations of 25, 50 and 75 chromosomes were used to study the effect of varying the upper limit on the number of offspring produced by the most fit chromosome (chromosome with the lowest error) in the range of 1.5 to 2.0 and the variation of the size of the crosspool (the number of low error chromosomes used for crossing and substitution in place of chromosomes with high error) as 40%, 56% and 80% of the population.

There are a number of different metrics for discussing the performance of a GA, but for the purposes of this study two performance measures will be used, namely Grefenstette's off-line performance and the SSE from the most fit individual in the population.

Grefenstette (10) defined the offline performance of a GA as

$$\mathbf{x}_{\mathbf{e}}^{*}(\mathbf{s}) = \frac{\mathbf{1}}{\mathbf{T}}\sum_{t=1}^{T}\mathbf{f}_{\mathbf{e}}^{*}(\mathbf{t}) \qquad \mathbf{(3.10)}$$

in which $\mathbf{x}_{\mathbf{e}}^{*}(\mathbf{s})$ refers to the off-line performance of a particular configuration of a GA denoted by **s** on a particular problem domain denoted by **e** and $\mathbf{f}_{\mathbf{e}}^{*}(\mathbf{t})$ refers to the best objective function value on iteration **t**. This expression indicates that the off-line performance is a running average of the best performance values to a particular iteration of the GA. In this study, the offline performance will describe a running average of the SSE from the calibration solutions to the ammonium ISE obtained after averaging the results from three repetitions of the GA configuration involved. This indicates that a GA with a low final offline performance described the response of the ISE to the calibration solutions better than a GA with a high final offline performance because the first GA has a lower calibration error than the second. Table 3.3 depicts the results obtained from studies of the variation of GA configuration with respect to the number of chromosomes used for a population, the percentage of those chromosomes which were allowed to cross

Population	Prescaling Factor α	Crosspool Size (% of population)	Average Offline Performance	Change in Offline Performance	Premature Convergence
25	1.7	56	122.38	4.041	19:26
25	1.7	80	137.68	54.87	35:29
25	2.0	40	123.5	77.679	29
25	2.0	56	138.86	49.31	61:30
25	2.0	80	108.27	12.369	19:51
50	1.5	56	87.59	12.025	-
50	1.5	80	87.135	7.87	36
50	1.7	40	104.28	57.49	-
50	1.7	56	104.135	12.016	13:20
50	1.7	80	87.16	36.575	44:71
50	2.0	40	91.305	37.08	-
50	2.0	56	98.2	12.39	27
50	2.0	80	112.58	38.83	73
75	1.5	40	87.271	99.977	-
75	1.5	56	89.126	9.91	92
75	1.5	80	79.268	9.02	57:61
75	1.7	40	116.04	45.663	-
75	1.7	56	99.79	3.43	30
75	1.7	80	87.52	3.786	9
75	2.0	40	108.74	16.8	-
75	2.0	56	86.634	10.47	52
75	2.0	80	119.99	21.68	58

Table 3.3 - Average final offline performance, average changes in the offline performance and number of iterations before premature convergence for three repetitions of different GA configurations with respect to population size, prescaling constant for reproduction and percentage of the population allowed to cross

and the size of the prescaling constant (limiting the number of offspring that the most fit individual in the population could produce in the next generation). This investigation was performed by using three repetitions of a particular GA configuration (the random number generator being seeded with different values for each repetition).The different GA configurations are evaluated in terms of the average of the offline performances achieved at the end of the GA for the three repetitions of a particular configuration. The average change in offline performance from the start of the operation of a GA to its finish for the three repetitions is also examined to consider the effect of the different initial offline performances on the final offline performances achieved by a GA. If a repetition of a particular GA converged before the designated termination criterion of 100 iterations, the number of iterations at which it converged is described in the column entitled premature convergence. From the table it can be seen that for the very small population of 25 chromosomes the GA converged prematurely at least once and the offline performance was quite high at convergence. This suggested that the GA was not performing a very efficient search of the space and was converging before a good solution could be found, irrespective of the degree of prescaling used or the percentage of the population used for a crosspool. It can also be seen that in the cases of larger populations that increasing the size of the crosspool increased the chances of premature convergence. This is probably caused by the reintroduction of particular chromosomes into the population by unproductive reproduction. This is the term given to a crossing procedure performed between two homogeneous mates, the results of which cross are identical to the original mates, which in this form of GA are substituted back into the population.

The effect of increasing the size of the crosspool appears to be complicated by the degree of prescaling employed by the GA. In the population of 50 chromosomes with prescaling of 1.7, increasing the size of the crosspool from 50-80% appears to improve the performance of the GA in terms of increasing the number of iterations performed by the GA before premature convergence and also in terms of the final offline performance of the GA. This was probably caused by increasing the chances of exploring the search space, produced by increasing the chances of recombination. However within the same population but with a prescaling constant of 2.0, increasing the crosspool from 56 to 80% may have had the effect of reducing premature convergence (although that was only observed in one example of three repetitions of the GA), it did not appear to improve the final performance of the GA. This ambiguity seems to be demonstrated again in the population of 75 chromosomes which clearly shows that a low degree of prescaling and a small crosspool produce the most dramatic improvements in the offline performance of the GA, even if its final offline performance may not be as good as other configurations of the GA (whose populations were most likely initiated with better candidates for the calibration parameters of the electrode in question).

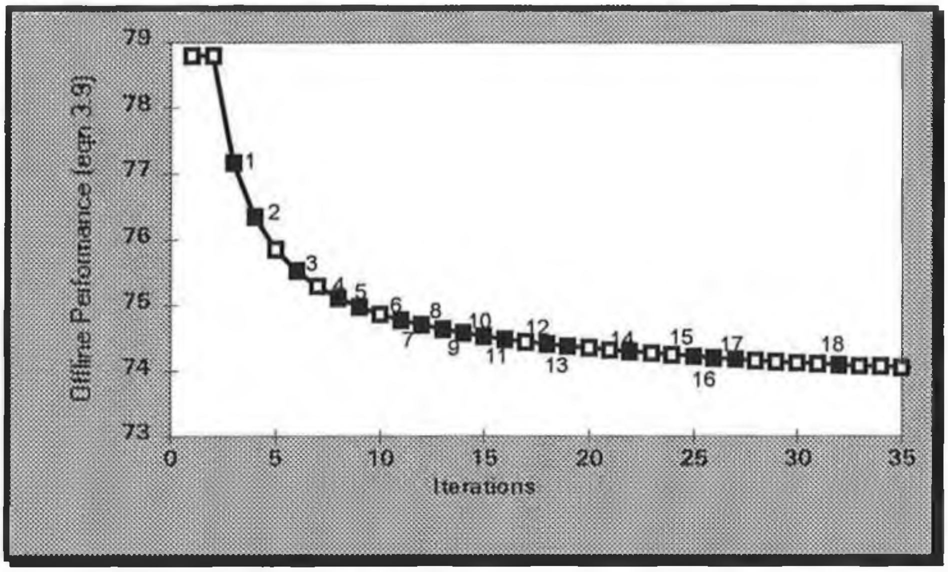


Figure 3.6- Progress of A GA with a population of 75 chromosomes allowing 80% of the population to cross and using a prescaling constant of 1.5 to control the number of offspring produced in an iteration by the most fit individual the population

Point	Operation	Number of Different Chromosomes
1	Mutation	2 (7), 1 (42), 4 (1), 5 (1), 3 (1), 7 (1)
2	Crossover	2 (4), 1 (4), 3 (3), 4 (1), 5 (1), 8 (2), 9 (1), 20 (1)
3	Crossover	1 (11), 2 (4), 3 (4), 4 (1), 9 (1), 27 (1)
4	Crossover	1 (7), 4 (1), 8 (2), 9 (1), 10 (1), 16 (1) 13 (1)
5	Mutation	1 (11), 5 (1), 7 (1), 8 (3),13 (1),15 (1)
6	Mutation	1 (4), 2 (2), 3 (2), 12 (1),25 (1),24 (1)
7	Mutation	1 (3), 4 (1), 7 (1), 61 (1)
8	Crossover	1 (5), 2 (1), 5 (1), 6 (1), 20 (1), 37 (1)
9	Crossover	1 (9), 4 (1),5 (1), 6 (1), 8 (1), 10 (1), 14 (1), 19 (1)
10	Mutation	1 (3),2 (1), 13 (1), 20 (1), 37 (1)
11	Crossover	1 (4), 3 (1), 6 (1), 62 (1)
12	Mutation	1 (3), 3(1), 7 (2),13 (1),18 (1), 24 (1)
13	Mutation	1 (1),2 (1), 14 (1), 27 (1), 31 (1)
14	Mutation	1 (3), 10 (1), 11 (1), 51 (1)
15	Mutation	1 (1), 74 (1)
16	Mutation	1 (1), 2 (1), 72 (1)
17	Mutation	1 (2), 3(1), 14 (1), 56 (1)
18	Mutation	1 (2),73 (1)

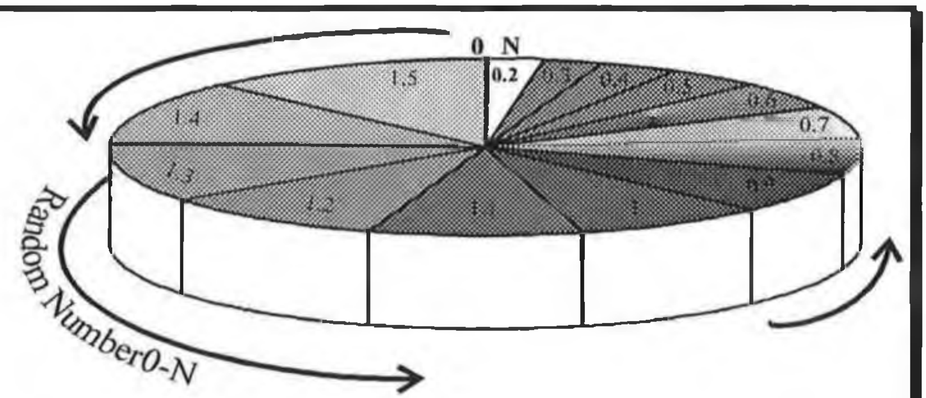
Table 3.4- The number of copies of particular chromosomes present in the population depicted in figure 3.6 at the different stages of the GA described by the black coloured data points with matching indices in figure 3.6. The number in brackets describes the incidence of the multiple copies in the population (i.e. 2 (2) indicates that there two different sets of chromosomes with two copies each in the population)

Table 3.4 and figure 3.6 depicts the off-line performance of the GA with 75 chromosomes with prescaling of 1.5 (to allow the chromosome which best described the response of the ammonium ISE [lowest SSE] to have a probability of producing offspring in the next iteration of the GA 1.5 times greater than a chromosome whose SSE was close to the average SSE of the current population) and 20% of the most fit chromosomes in the population allowed to cross. The averaging effect of the off-line performance smoothes out the sudden decreases in the error of the most fit chromosome which is produced by random beneficial changes in its genetic makeup arising from crossing or mutation. It can be seen that in the early stages of searching, as the diversity of the population is high, that crossover is predominantly responsible for improvements in the performance of the most fit organism in the population. But as the diversity in the population is reduced in the later stages of searching, mutation becomes the more predominant searching mechanism (since the chances of crossing occurring between identical chromosomes increase and hence the chances of crossing acting as a productive means of searching for a solution decrease) and the search prematurely converges to a solution after 60 iterations, without examining the search space sufficiently well to find an adequate solution.

Based on these results it was determined that variation of the prescaling constant and the percentage of the most fit chromosomes involved in crossing, slowed down premature convergence but did not reduce the problem satisfactorily.

3.3.1.2 Rank Scaling and Roulette Wheel Selection

In the SGA, the number of progeny an individual will produce is related to the fitness of the individual determined from the objective function. In the case of this study the lower the SSE, the more progeny an individual would produce. The number of offspring of each individual is determined by calculating the fitness of the individual relative to the average fitness of all the individuals of the population. This direct relationship between an individual's fitness and the number of its offspring while being a central part of an SGA is also a reason for its susceptibility to premature convergence. If there is a particularly fit individual in a population of otherwise very unfit individuals the ratio of the individual's fitness to the average fitness of the population does not place any limitations on the number of offspring it can produce, other than that it must be less than or equal to the size of the population itself. Linear prescaling attempts to limit the number of offspring a particular individual can produce for the next generation but it does not fully tackle the problem of "super-fit" individuals. An alternative approach to this problem described by Baker (11) allocates the number of offspring to an individual according to the rank of its performance rather than its magnitude. In this procedure the chromosomes in the population are sorted and assigned a rank according to their errors. An individual with a low error would be given a higher rank than an individual with a high error according to this scheme. The number of offspring that could be produced by an individual according to its rank was prescaled such that the area under the linear transform was equal to the population size, and such that the highest ranking individual would produce a certain number of offspring according to a prescaling constant. In this modification of the SGA, individuals are selected to reappear in the next generation by a roulette wheel selection procedure. Roulette wheel selection is a random process which is depicted in figure 3.7. The procedure simulates a roulette wheel whose slots are weighted according to the number of offspring an individual should produce. A random number is generated which may or may not fit within one of the particular slots in the wheel and as such determines which individual will produce one offspring in the next generation, the actual procedure was implemented in the manner described by Goldberg (9).



N= Number of chromosomes in the population
(N=12 in this case)

Figure 3.7- Diagrammatic representation of roulette wheel selection, the slots in the wheel are weighted according to the number of offspring which an individual should produce [the chromosome with the lowest SSE is weighted to 1.5 in this diagram]. A random number is generated in the range 0-number of chromosomes in the population. The chromosome into whose slot the random number lies will produce one offspring..

This approach was investigated for a population of 75 chromosomes with rank prescaling to 1.5 and 1.7.

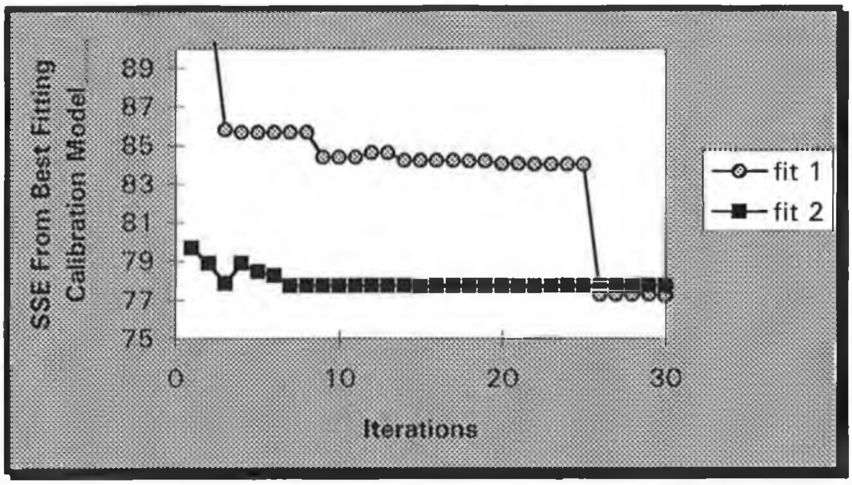


Figure 3.8 - Variation of SSE of best fitting calibration model in a population of 75 chromosomes being scaled for reproduction according to their ranks with a prescaling constant of 1.5 and selected for reproduction using the roulette wheel approach

Figure 3.8 depicts the change in SSE of the most fit individual in the population during the operation of the GA. The curve is not as smooth as the offline performance curve in figure 3.6 because there is no averaging effect being used. There are sporadic dramatic decreases in error with randomly occurring beneficial changes in the genetic makeup of the chromosome with the lowest error. The GA was repeated twice and it can be seen that the performance of the GA during the repetition is marked by sporadic decrease in error as would be expected, however in some cases the reduction in error is followed by return of the error to its previous value. This is probably caused by a chromosome with a low error being genetically modified to produce a lower error, which instead of reappearing in the next generation is instead lost, returning the lowest error of the system to its previous state. This apparently random loss of the most fit individual in the population also occurred in the GA with prescaling to 1.7.

It can also be seen that the two repetitions of the GA do not achieve the same SSE, reflecting the poor search precision of the GA.

3.3.1.3 Stochastic Reminder Sampling Without Replacement And Reduction Of Premature Convergence By Prevention Of Incest

Roulette wheel selection is a stochastic process with a high degree of variance between the expected number of progeny and the actual number which occur in the next generation (as seen in the possibility of losing the most fit individual from the current population). To reduce this problem, an alternative selection procedure known as stochastic remainder sampling without replacement was implemented in the manner described by Goldberg (9). In this procedure the expected number of offspring are calculated according to rank as before. Individual chromosomes are assigned offspring according to the integer part of their expected number of offspring. The fractional parts of the expected number values are treated as probabilities. Each chromosome in turn will use the fractional parts of their expected number of offspring in weighted coin tosses in order to determine whether they will have another offspring in the next generation (e.g. a chromosome with an expected number of offspring of 1.7 will definitely produce one offspring and will have a 0.7 chance of producing another offspring). This process continues until the population is full.

The crossover operator was targeted next as a part of the GA within which modifications could be made to prevent the convergence of the GA to sub-optimal solutions. The modification was developed to increase the occurrence of productive crossover. The modification involved the use of a rejection loop, whereby two chromosomes were selected randomly from the current population as mates for the crossover operator. The two chromosomes were compared and if they were identical then the second chromosome from the pair would be rejected and a new chromosome would be chosen from the population at random, until the two chromosomes were no longer identical. The second modification was developed to prevent crossovers between related but non-identical chromosomes. If a particular chromosome mated with its parent, there would be little change in the genotype of either chromosome, reducing the exploration efficiency of the crossover operation. The modification to the GA involved the storage of the genetic profile of each parent of a chromosome produced by crossing and a rejection loop similar to the one described for ensuring two identical chromosomes did not mate. When a chromosome was chosen for crossing in the next generation, its mate was compared with the genetic profile of its parents. If the two genetic profiles were identical, the chromosome was rejected and another chromosome was selected. This modification, which while being similar to that of Eshelmann and Schaffer (12), does not explicitly establish a variable threshold between the crossing chromosomes, comparing their similarities on the basis of their Hamming distance. Rather, the threshold is implicitly established by the parental relationships between the chromosomes.

It can be seen from figure 3.9, that for a population of 300 chromosomes and a crosspool which was composed of 80 % of the total population, rank selection reduced the chances of premature convergence and increased the chances of finding optimal solutions to the optimisation problems, but incest prevention during crossing did not result in any major changes in the performance of the GA. Since the prevention of incest during crossing added a large overhead to the programs memory requirements it was decided not to pursue this approach any further.

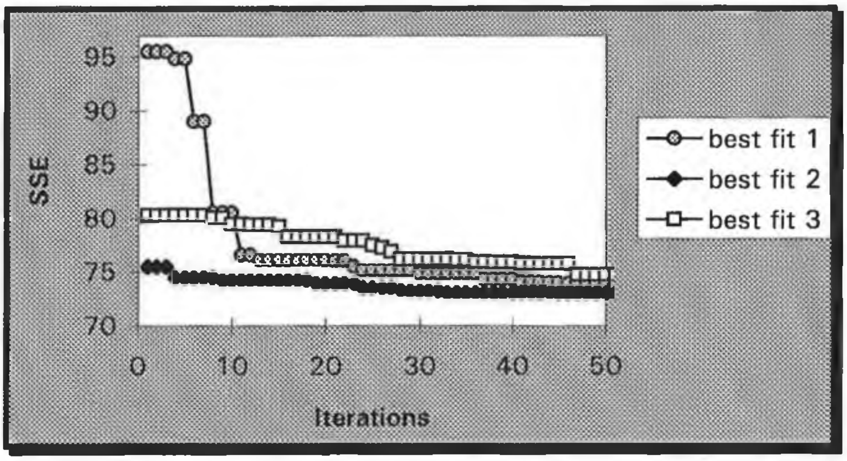


Figure 3.9 - Variation of SSE of best fitting calibration models in a population of 300 chromosomes (crosspool of 240 chromosomes) prescaled for reproduction according to their ranks and selected for reproduction using the stochastic remainder sampling approach

3.3.1.4 Post-Hybridisation Of GA With Simplex Optimisation

A GA was written to switch to a simplex method of optimisation by using the different six most fit individuals in the population at that time to generate the initial vertices of the simplex. The switch to the simplex was triggered when the difference between the average fitness of the population and the fitness of the most fit individual had fallen below a certain threshold, indicating that the diversity of the population was being reduced to a level where the efficiency of the GA for searching would be affected. The magnitude of the threshold was determined by letting a GA calculate through a number of iterations until it converged on a particular set of parameters for the Nikolskii-

Eisenman expression for the ammonium ISE. The threshold was then varied in units of a similar magnitude as the SSE of the solution to which the preliminary GA had converged. If the threshold has a low value then the population of the GA had a low diversity (almost converged to a single set of calibration parameters for the ammonium ISE), this tends to happen in the later stages of the operation of a GA as the population gradually fills with a chromosome which best solves the particular task set to the GA. As such the magnitude of the threshold would determine at what stage in the GA there would be a switch to the simplex. A large threshold would indicate that there was a switch to a simplex in an earlier stage of the GA than a small threshold. A study was performed on the effect of variation of the threshold level in different sized populations on the SSE of the chromosome which best described the response of the electrode

Table 3.5 depicts the results obtained by repeating a GA three times over different population sizes and at different switching thresholds. From table 3.5 it can be seen that the changes in average SSE from the best fitting calibration models are very small. The uncertainty expressed in the standard deviations of the SSEs are larger than the cahnegs in the average SSE. As such, the changes in the average SSE are not significant relative to the uncertainty in its determination. Therefore this data suggests that varying the size of the populatiom and the threshold for transfer to a simplex have no significant effect on the SSE determined by the GA. Although the results of this study do not any significant changes in the average SSE it might have been expected that with smaller populations a transfer to a simplex method in the early stages of the GA (before the diversity of the population had decreased significantly) would be beneficial because it would bypass the problem of premature convergence in the population.

Population Size	Threshold	Average SSE From The Best Fitting Calibration Models	Standard Deviation of SSE From The Best Fitting Calibration Models
50	10	73.661	1.459
50	5	75.489	1.604
50	1	77.885	3.708
100	10	78.483	5.0205
100	5	78.4249	4.079
100	1	74.744	2.2606
200	10	76.3395	5.0279
200	5	79.6375	9.377
200	1	76.0098	1.63
300	10	72.9397	0.10986
300	5	74.473	2.2828
300	1	79.488	5.245

Table 3.5 - Results from post-hybridisation of GA with simplex repeated three times using the GA configuration discussed in the text

This approach to post-hybridising the simplex technique with a GA was not pursued in the further studies because it was observed that while the solutions from the most fit individuals in the population were different, the genes which were most significant in determining the fitness of these individuals were very similar. Since the vertices of the simplex were taken from the six most fit individuals of the population, some of the dimensions of the vertices were effectively set as constants, limiting the movement of the simplex in the search space (Nelder and Mead (13) noted that if one variable of a simplex is set to a constant on the vertices of the simplex, the simplex is constrained to search in one dimension less).

3.3.2 Application to an array of ISEs used in FIA regime

The previous study showed no significant difference in average SSE achieved with differnt population sizes when transferring to a simplex with the differnt six most fit individuals in a current population. This method of post-hybridising the simplex with the GA was abandoned for the reasons discussed above. Instead a GA was reperated six times, the calibration parameters obtained from the chromosomes which most closely modelled the response of the ISE from each repetition of the GA were then employed as vertices of a simplex optimisation procedure used to refine the eventual calibration model.

The results from the investigations discussed in sections 3.3.1.1-3.3.1.3 suggested a particular configuration for the GA to be used for the determination of the optimal calibration parameters of the array of ISEs used in this study. This configuration involved the use of a population of 300 chromosomes which were selected for reproduction by stochastic remainder sampling without replacement according to their ranks. Single point crossover was performed with a crossover probability of 0.65 and mutation was performed by random bit inversion with a probability inversely related to the size of the population i.e. 0.003.

The data used in this application were the potentials (described in table 3.2) measured by a series of ISEs to a series of solutions whose concentrations are described in table 3.1. Details of the experimental system used are described in (1). The activities of the ammonium, sodium, potassium and calcium ions were calculated from the concentrations of the ions in the calibration solutions by means of the Davies expression (14).

$$\log\gamma_i = -0.5z_i^2\left(\frac{\sqrt{I}}{1+\sqrt{I}} - 0.2I\right) \qquad (3.11)$$

The activities of the ions in these solutions were determined by linearisation of the respective Nikolskii-Eisenman expression followed by Gauss-Jordan elimination of the selectivity coefficient matrix.

ISE	E° (mV)	S (mV)	$K^{*}_{jNH_4^+}$	$K^{*}_{jNa^+}$	$K^{*}_{jK^+}$	$K^{*}_{jCa^{2+}}$
NH_4^+	100-300	50-60	-	10^{-5}-0.01	10^{-3}-1.0	10^{-5}-1.0^{-3}
Na^+	100-1000	10-100	10^{-3}-1.0	-	10^{-3}-1.0	10^{-5}-1.0^{-3}
K^+	200-300	50-60	5x10^{-4}-0.5	5x10^{-6}-0.5	-	10^{-4}x10^{-2}
Ca^{2+}	100-300	20-30	10^{-7}-0.05	10^{-7}-0.1	10^{-4}-0.5	-

Table 3.6 - Ranges within which the search is confined for each calibration parameter

Table 3.6 describes the ranges within which each calibration model parameter was encoded for each ISE. The ranges for the parameters were initially set more broadly (e.g. for the potassium ISE 100<**E**<1000, 10<**S**<100, 0.001<$K^{*}_{K^+NH_4^+}$<1.0, 0.001<$K^{*}_{K^+Na^+}$<1.0, 0.00001<$K^{*}_{K^+Ca^{2+}}$<0.001). A GA was operated within these initial ranges, as the general values of the calibration parameters obtained from the most fit individuals in the population became more clear, the initial ranges for the calibration parameters were refined into those described in table 3.6.

ISE =j	SSE	Ej (mV)	Sj (mV)	$K^{pot}_{jNH_4^+}$ (x10^{-3})	$K^{pot}_{jNa^+}$ (x10^{-3})	$K^{pot}_{jK^+}$ (x10^{-3})	$K^{pot}_{jCa^{2+}}$ (x10^{-3})
NH_4^+	0.021 (8.3)	226.007 (0.5)	45.526 (0.8)	-	0.816 (37.6)	186.4 (2.1)	6.226 (31.4)
Na^+	0.027 (9.0)	257.875 (0.7)	55.02 (1.1)	0.635 (96.9)	-	107.757 (1.8)	15.656 (8.6)
K^+	0.045 (18.0)	271.386 (1.4)	51.952 (2.5)	154.9 (12.3)	0.17 (15.0)	-	4.853 (8.7)
Ca^{2+}	0.099 (49.4)	122.2 (5.7)	24.354 (7.896)	0.136 (236.4)	1.843 (60.2)	5.306 (56.5)	-

Table 3.7 - Calibration parameters determined for each ISE after six repetitions of the GA. The terms in brackets refer to the percentage relative standard deviations(%RSDs) of the associated parameters).

Table (3.7) describes the calibration parameters obtained from the most fit individual in a population after the termination of a GA. Since the GA was repeated six times for each electrode, the calibration parameters are represented as their average values obtained from the repetitions accompanied by their percentage relative standard deviations (%RSDs). It can be seen that the %RSDs of the calibration parameters generally tend to increase as the magnitude of the calibration parameters decrease (e.g. for the ammonium ISE, %RSD $\boldsymbol{K}^{*}_{NH_4^+Ca^{2+}}$ ~60 times %RSD **Ej**, for the sodium ISE %RSD $\boldsymbol{K}^{*}_{Na^+Ca^{2+}}$ ~ 13 times %RSD **Ej**).

This is to be expected since larger magnitude parameters such as the slope or cell potential would make a much larger contribution to the SSE term which is being minimised by the GA. The use of narrower initial ranges for such parameters, tends to offset the possibility that these parameters would become rapidly fixed within a population to effectively become constants, allowing the parameters of lower significance to the SSE term to be determined more precisely because of the diversity in the population of genes encoding these parameters.

The results obtained from the ammonium and sodium ISE are interesting since the initial ranges of E were much bigger than those for **S**, yet **E** was determined more precisely than **S**. The %RSDs of the calibration parameters of the potassium selective electrode do not appear to vary as much as those of the other ISEs. This is most likely due to difficulties establishing the range for the search for the calibration parameters. This problem would be more clearly demonstrated by the operation of the simplex technique which does not have its search space confined by the encoding procedure in the same way as the GA. It can also be seen that the determination of the calibration parameters tends to have a large range for the calcium ISE, with this effect being particularly noticeable for $K^{*}_{Ca^{2+}NH_4^+}$. This problem may have occurred because the ISE being selective for the divalent ion Ca^{2+}, would produce a lower magnitude signal to its primary ion than an ISE selective for a monovalent ion would have to the same activity of its primary ion (i.e. has a lower slope than an ISE selective for a monovalent ion). As such the sensitivity of the calcium ISE to the calibration solutions would be reduced relative to the other ISEs.

If the models with the smallest SSE from the repeated runs of the GA are used to determine correlations between predicted activity and actual sample activity for all the solutions used to calibrate all the electrodes a correlation coefficient of 0.9800 was found for the NH_4^+ ISE, 0.9391 for the Na^+ ISE, 0.9810 for the K^+ ISE and 0.8703 for the Ca^{2+} ISE.

ISE j=	Ej	Sj	$K^{pot}_{jNH_4^+}$ (x10^{-3})	$K^{pot}_{jNa^+}$ (x10^{-3})	$K^{pot}_{jK^+}$ (x10^{-3})	$K^{pot}_{jCa^{2+}}$ (x10^{-3})	SSE	Correlation
NH_4^+	225.776	45.443	-	0.936	184.9	6.465	0.019	0.980
Na^+	257.19	54.19	<0.001	-	106.89	17.226	0.022	0.939
K^+	270.155	51.396	143.04	<0.001	-	5.36	0.037	0.981
Ca^{2+}	119.081	23.293	<0.001	0.897	1.672	-	0.028	0.870

Table 3.8 - Calibration parameters determined for each ISE after simplex optimisation based on results obtained from the GA

Table 3.8 depicts the parameters obtained for the calibration models for each electrode following simplex optimisation based on the results of the GAs. Comparing table 3.7 with table 3.8 it can be seen that there is not a great deal of change between the best fit calibration models from the GAs and the results from the simplex optimisation.

The largest improvement in SSE s seen for the calcium electrode with large changes in the least precisely determined calibration parameter namely $K^{*}_{Ca^{2+}NH_4^+}$, due to difficulties encountered with the GA encoding the range of the search space for this parameter, similar to the difficulties encountered for the potassium selective electrode described earlier.

Figures 3.10 a, b, c and d depict the spread of the predicted activities against the known sample activities for the four ISEs. They show a fairly even spread about the hashed diagonal line depicting the ideal case of the predicted activity equalling the known sample activity. Some outliers have been labelled in the figures in an attempt to relate these outliers to patterns in their associated solution compositions.

In the case of the activities predicted by the ammonium ISE in figure 3.10 a, the data points labelled NH_4^1, NH_4^2 and NH_4^3 demonstrate a predicted ammonium activity lower than its known activity. In each of the solutions associated with these data points the concentration of one or more of the interferents is much higher than the concentration of the ammonium ion. In the case of the data point labelled NH_4^1 (which refers to solution 28 in table 3.1) the concentration of the calcium ion is approximately 14 times higher than the concentration of the ammonium ion. Similarly, for the data point NH_4^2 (which refers to solution 18 in table 3.1) the concentration of the potassium ion is approximately 11 times higher than the concentration of the ammonium ion. For the data point NH_4^3 (which refers to solution 10 in table 3.1) the concentration of sodium is approximately 11 times higher than the concentration of the ammonium ion. This suggests that the model may be overcompensating for the influence of the interferences on the response of the electrode, and as a consequence, underestimating the activity of the primary ion. At higher activities of the ammonium ion there can be seen a cluster in

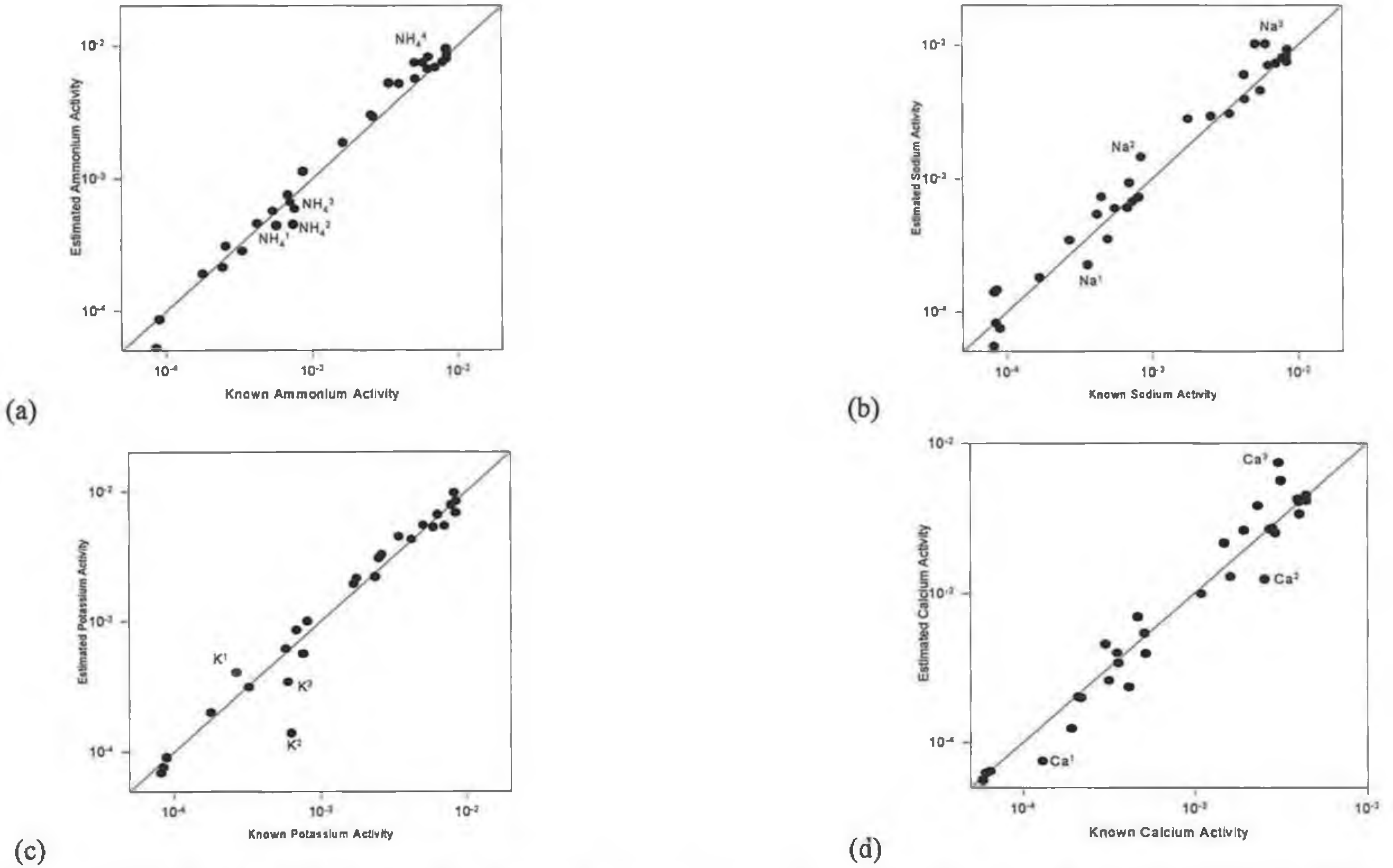


Figure 3.10 -Correlation between predicted and solution activity for the entire calibration set by the (a) ammonium ISE (b) sodium ISE (c)potassium ISE and (d) calcium ISE. The text on the graphs (e.g. Ca^1, Na^1 etc.) are indices which will be used to refer to the associated outliers in the textual discussion.

which the predicted activity of the ammonium ion is higher than its known activity. The data point labelled NH_4^4 is a member of this cluster. This data point refers to solution 25 in table 3.1. It can be seen from the table that the concentrations of the sodium, potassium and calcium ions are either lower than, or very similar to, the concentration of the ammonium ion in this solution. In this case the contribution of the interferents to the response of the ammonium ISE would be insignificant relative to the contribution from the primary ion and most likely could not be modelled properly. A similar case is found for the data point labelled Na^1 in figure 3.10b, this data point refers to solution 24 in table 3.1. It can be seen from table 3.1 that the concentrations of the ammonium, potassium and calcium are smaller than the concentration of the sodium in this solution. In this case however, the predicted sodium activity is smaller than its known activity. At higher sodium activities outliers can be seen in which the predicted activity is larger than the known sodium activity. The data point labelled Na^2 refers to solution 3 in table 3.1. It can be seen from table 3.1 that the concentration of ammonium is 10 times larger than the concentration of sodium in this solution. The data point labelled Na^3 which refers to solution 5 in table 3.1 has a composition which contrasts with that of the solution referred to by data point Na^2. In this case, it can be seen from the table that the concentrations of the interferents are lower than the concentration of the sodium. One of the most obvious outliers exists for the potassium ISE (labelled as K^2 in figure 3.10 c), refers to solution 25 in table 3.1. It can be seen from the table that the concentration of calcium is approximately 12 times larger than the concentration of potassium for this solution. While the data point labelled K^3 (which refers to solution 29 in table 3.1) does not demonstrate as large a difference between the predicted and known activity of potassium (distance between the data point and the hashed diagonal, in the direction parallel to the y-axis) it can be seen that the concentration of ammonium and sodium are approximately 8.5 times higher than the concentration of potassium. In the case of the data point labelled K^1 (which refers to solution 16 in table 3.1), the predicted activity of the potassium ion is higher than its actual activity. It can be seen from table 3.1 that the ammonium, sodium and calcium ions are of lower or very similar concentration to the potassium ion. In the case of the predictions of calcium activity depicted in figure 3.10d it can be seen that the data points Ca^1 and Ca^2 are distinctive outliers in which the predicted calcium activity is lower than the known calcium activity. The data points Ca^1 and Ca^2 refer to the predicted activity of the calcium in to solution 8 and solution 13 in table 3.1 respectively. In both cases it can be seen that the concentration of ammonium, sodium and potassium are either lower or very similar to the concentration of the calcium ion. The data point Ca^3 is a member of a cluster of data points which describe a higher predicted calcium activity than its known activity. The data point Ca^3 refers to solution 17 in table 3.1, from which it can be seen that concentrations of the ammonium, sodium and potassium ions are very similar to the concentration of the calcium ion.

In summary, it can be seen that the outliers labelled in figures 3.10 a, b, c and d referred to solutions in which two main problems existed, namely;

(I) the concentration of the interferents for a particular ISE were much higher than the primary ion concentration. In this case the calibration models were not capable of decoupling the large contributions to the response of the electrodes arising from the interferents from the contribution arising from the primary ion.

(II) The concentration of the interferents for a particular ISE were either lower or of similar magnitude to the concentration of the primary ion. In this case the contribution of the interferents to the response of the ISE would be insignificant relative to the contribution from the primary ion, as such the activities of the interferents could not be modelled correctly.

3.4. Comparison Of Calibration Parameters With Those Determined By Other Procedures

Table 3.9 depicts the response parameters calculated for the four ISEs by the GA-simplex method compared with values for the same parameters obtained by (i) a simplex method. applied by Sáez de Viteri et al to the same set of concentration and ISE response data as was used for this study (1) (referred to as simplex method 1 in the table) (ii) a separate solution method described by Davies et. al. (referred to by the reference number in the table) (15), (iii) a simplex method applied to data acquired from an FIA system in a different concentration range by Forster et al. (3) (referred to as simplex method 2 in the table), (iv) a simplex method applied to an array of dip-type ISEs by Forster et al. (2) (referred to as simplex method 3 in the table), (v) mixed and separate solution (referred to by the reference number in the table) methods described by Svehla et al. (16) and by (vi) similar traditional manual methods described by Forster et al. (2). Simplex method 1 provides the most valid comparison with the Ha-simplex method since the other techniques mentioned involve the use of different experimental approaches (eg. separate solution method) and/or different concentration ranges (eg. simplex method 2). It can be seen from the table that for the ammonium and sodium selective electrodes a sub-Nernstian slope was determined by the GA approach and by all the other literature sources. It can also be seen that for the ammonium ISE that the selectivity coefficient against sodium is the lowest valued selectivity coefficient of the ISE determined by GA-simplex and simplex method 1 approaches. The value for this selectivity coefficient determined by the simplex method 1 is larger than that determined by the GA-simplex approach and is comparable with the value determined by the separate solutions method (reference 15). However the percentage relative standard deviation (%RSD) for this parameter determined by the simplex method 1 is quite high indicating that the parameter was not very precisely determined by this approach. A similar effect is seen for the selectivity coefficient of the sodium ISE against ammonium, in which the value for the selectivity coefficient determined by the simplex method 1 is larger than the value calculated by the GA-simplex approach and comparable to the value calculated by the separate solution method 2. The mixed solution method determines a larger value again for this selectivity coefficient. The %RSD of this parameter determined by this approach is the highest for all the selectivity coefficients of this electrode. It can be seen from the table that there is a close agreement between the values calculated for the selectivity coefficient of the ammonium ISE against potassium by the GA-simplex approach, simplex method 1 and separate solutions method . This is reflected in a %RSD which has a smaller value than the %RSDs of the other selectivity coefficients calculated by simplex method 1. There is not as close an agreement between the values of the selectivity coefficient of the ammonium ISE against calcium determined by the GA-simplex, simplex methods 1 and separate solution method 1. This in turn is reflected by a large %RSD calculated for this parameter by simplex method 1. The GA-simplex method and simplex method 1 determine a high value for the selectivity coefficient of the sodium ISE against potassium relative to the values calculated for this parameter by the other methods depicted in the table. This suggests that the high value determined for this parameter is less a feature of the optimisation procedure used as much as a feature of the experimental data itself. There can be seen a close agreement between the value of the selectivity coefficient of the sodium ISE against calcium calculated by the GA-simplex approach and simplex method 1 but an interestingly wide variation between the values determined for this parameter over all the other methods. It is very difficult to draw conclusions from this because the differences in the selectivity coefficients may be due to differences in the ways the experiments were performed (e.g., concentration ranges for the measurements) (1).

It can be seen that the GA simplex and simplex method 1 approaches determine a sub-Nernstian slope for the potassium ISE, whereas slightly higher slopes are obtained by the other methods, indicating that this is likely to be a feature of the experimental data itself rather than the optimisation procedure employed. This, however, may be a feature of the experimental data itself rather than the optimisation procedure employed. It can be seen that there is a close agreement between the values calculated for the selectivity coefficient of the potassium ISE against ammonia calculated by the GA-simplex and simplex method 1 approaches. The same cannot be said for the selectivity coefficient of the ISE against sodium. The value of the selectivity coefficient calculated by simplex method 1 is comparable with the values calculated for this selectivity coefficient by simplex method 2 and the simplex method in a steady state system. However this parameter has a very high %RSD calculated by the simplex method 1. The value of the selectivity coefficient of the potassium ISE against calcium calculated by the GA-simplex approach is much smaller than the value calculated by the simplex approach and is comparable to the value calculated for this variable by the other multivariate approaches. However the selectivity coefficient calculated by the simplex method 1 has a very large %RSD, suggesting that the determination of this parameter by the simplex method 1 is rather imprecise.

A great deal of variation can be observed between the calibration parameters determined for the calcium ISE using the different techniques, even between the same technique applied to the same data (as judged by the %RSDs of the parameters determined by simplex method 1). It is difficult to draw any conclusions from these data because of the imprecision of the simplex method 1 (although it could be said that the selectivity coefficient of the ISE against sodium have some agreement) and because the selectivity coefficients involved in the Nikolskii-Eisenman expression for this electrode (i.e. $z_i \neq z_j$) are a function of the interfering ion activity (1).

ISE	Response Parameter	GA-Simplex Method	Simplex Method 1 [1]	Separate Solutions Method references [15] and [16]	Simplex Method 2 [3]	Simplex Method 3 [2]	Mixed Solution Method [16]	Manual Methods [2]
NH_4^+	E	225.78	226.9 (0.2)	* [15]				
	S	45.44	45.6 (0.4)	47.0 [15]				
	$K^{pot}_{NH_4^+Na^+}$	$9.36x10^{-4}$	$4.7x10^{-3}$ (51.1)	$6.35x10^{-3}$[15]				
	$K^{pot}_{NH_4^+K^+}$	0.185	0.173 (3.4)	0.15 [15]				
	$K^{pot}_{NH_4^+Ca^{2+}}$	$6.46x10^{-3}$	$5.2x10^{-3}$ (69.2)	$3.27x10^{-3}$ [15]				
Na^+	E	257.19	261.87 (0.9)	* [16]	348.2	-72.3	*	
	S	54.19	54.16 (1.6)	* [16]	55.9	54.3	54.2	
	$K^{pot}_{Na^+NH_4^+}$	$<1x10^{-6}$	$2.0x10^{-3}$ (45.0)	$1.58x10^{-3}$ [16]	*	*	$1x10^{-2}$	
	$K^{pot}_{Na^+K^+}$	$1.07x10^{-1}$	0.103 (1.6)	$3.16x10^{-3}$ [16]	$2.13x10^{-2}$	$2.03x10^{-2}$	$3.98x10^{-3}$	
	$K^{pot}_{Na^+Ca^{2+}}$	$1.72x10^{-2}$	$1.46x10^{-2}$ (15.1)	$5.01x10^{-4}$ [16]	$8.0x10^{-3}$	$1.79x10^{-2}$	$3.16x10^{-4}$	
K^+	E	270.15	269.28 (0.3)		154.2	57.8		*
	S	51.39	52.21 (0.4)		55.2	54.1		53.8
	$K^{pot}_{K^+NH_4^+}$	$1.43x10^{-1}$	$1.33x10^{-1}$ (7.1)		*	*		*
	$K^{pot}_{K^+Na^+}$	$<1.0x10^{-6}$	$1.3x10^{-3}$ (130.8)		$5.5x10^{-3}$	$4.2x10^{-3}$		$4.1x10^{-4}$
	$K^{pot}_{K^+Ca^{2+}}$	$5.36x10^{-3}$	$1.27x10^{-2}$ (128.3)		$6.2x10^{-3}$	$5.12x10^{-3}$		$1.6x10^{-4}$
Ca^{2+}	E	119.08	125.89 (2.7)		326.8	50.4		*
	S	23.29	25.01 (3.9)		27.8	26.8		27.2
	$K^{pot}_{Ca^{2+}NH_4^+}$	$<1x10^{-6}$	$2.0x10^{-4}$ (50.0)		*	*		*
	$K^{pot}_{Ca^{2+}Na^+}$	$8.97x10^{-4}$	$1.5x10^{-3}$ (60.0)		$4.3x10^{-3}$	$1.0x10^{-5}$		$2.0x10^{-5}$
	$K^{pot}_{Ca^{2+}K^+}$	$1.67x10^{-3}$	$7.5x10^{-3}$ (54.7)		$5.1x10^{-3}$	$1.7x10^{-5}$		$2.8x10^{-5}$

Table 3.9 - Calibration parameters determined for the four ISEs by the GA-simplex approach compared with values determined for the same parameters by other methods (The terms in brackets beside the parameters determined by simplex method 1, refer to the %RSDs of these parameters, the terms in square brackets in the separate solutions parameters refer to the references from which the parameters come)

3.5 *Comparison Of Genetic Algorithm With Simplex Technique*

ISE =j	SSE	E_j (mV)	S_j	$K^*_{jNH_4^+}$	$K^*_{jNa^+}$	$K^*_{jK^+}$	$K^*_{jCa^{2+}}$
			(mV)	($x10^{-3}$)	($x10^{-3}$)	($x10^{-3}$)	($x10^{-3}$)
NH_4^+	0.021	225.846	45.502	-	2.09	185.13	4.19
	(24.27)	(0.22)	(0.34)		(96.34)	(2.63)	(93.41)
Na^+	0.048	256.241	55.419	4.419	-	109.1	5.679
	(23.61)	(0.31)	(0.56)	(50.33)		(3.59)	(102.63)
K^+	0.038	270.506	51.523	142.63	0.453	-	5.322
	(13.28)	(0.19)	(0.43)	(2.45)	(238.5)		(6.48)
Ca^{2+}	0.032	119.37	23.378	0.264	0.823	1.674	-
	(22.11)	(0.31	(0.49)	(187.46)	(40.34)	(12.12)	

Table 3.10 - Calibration parameters determined by six repetitions of the simplex method applied to data composed of the responses of the four ISEs described earlier to the calibration solutions described in table 3.1(i.e. the same data set as was used by the GA). The terms in brackets refer to the percentage relative standard deviation (%RSD) of each parameter.

Table 3.10 depicts the calibration parameters determined for the four ISEs by a simplex optimisation procedure, using the same data set as was used for the GA-simplex technique. The simplex used in this approach was initialised randomly in the ranges for the calibration parameters developed for the GA. The simplex was repeated six times, as such table 3.10 describes the average value of each calibration parameter accompanied by its %RSD.

It can be seen that the %RSD of a calibration parameter tends to decrease as the magnitude of the parameter increases, in a similar fashion to that observed for the GA. It can also be seen that the %RSDs of the selectivity coefficients of the potassium ISE vary more than those obtained using the GA approach depicted in table 3.7. The calibration parameters determined by the simplex technique are very similar to those obtained by the

GA even for the calcium ISE. The largest difference between the calibration parameters occurs for the selectivity coefficients, in particular those describing the selectivity of the sodium ISE against ammonium, which is approximately seven times larger when determined by the simplex approach than when determined by the GA approach. This selectivity coefficient however, is the least precisely determined of the selectivity coefficients of the ISE using the GA approach.

ISE = j	SSE	E$_j$ (mV)	S$_j$ (mV)	$K^*_{jNH_4^+}$ (x10^{-3})	$K^*_{jNa^+}$ (x10^{-3})	$K^*_{jK^+}$ (x10^{-3})	$K^*_{jCa^{2+}}$ (x10^{-3})
NH_4^+	0.108	5.66	6.28	-	2.32x10^{-2}	6.48x10^{-1}	2.49x10^{-1}
Na^+	6.21x10^{-2}	6.10	3.62	7.66x10^{-2}	-	2.35x10^{-1}	5.32x10^{-2}
K^+	2.63	48.52	34.35	29.46	5.56x10^{-4}	-	1.48
Ca^{2+}	0.46	342.59	273.69	0.42	11.14	218.81	-

Table 3.11 - Ratios of the variances of the calibration parameters determined by GA and simplex approaches to the same data set composed of the responses of the four ISEs described earlier to the solutions of composition described in table 3.1

Table 3.11 depicts the ratios of the variances of the calibration parameters determined by means of the GA and simplex approaches. Using an f-test at a 95% significance level ($f_{5,5}^{0.05} = \mathbf{5.05}$ **(17)**) it was determined that the variance of the SSE determined by the simplex approach was significantly larger than that determined by the GA for the sodium and ammonium ISEs. The variances with which the cell potentials are determined are significantly greater using the GA approach as opposed to the simplex approach. This is also observed for the slopes of the ammonium, potassium and calcium ISEs. Since the SSEs of the ammonium, potassium and calcium ISEs determined by the GA-simplex approach were not significantly larger than those determined by the simplex approach and the estimates for the slopes and cell potentials of these electrodes determined by the former approach were significantly larger than those determined by the latter approach, it suggests that the precision with which the selectivity coefficients are determined plays a part in the variances of the calibration errors. The variances with which the smallest magnitude selectivity coefficients are determined are significantly greater using the simplex technique than using the GA technique. However in the case of the calcium ISE where there is not a great deal of difference between the selectivity coefficients, the variance with which the selectivity coefficients determined by the GA are significantly greater than those determined by the simplex.

Sáez de Viteri et al. (1) discussed how variation of the value of $\boldsymbol{K}_{ij}^{*}$ for an ammonium ISE affected its calibration error when the rest of the Nikolskii-Eisneman parameters were held constant at their optimum values determined previously by a simplex method. It was found that if $\boldsymbol{K}_{ij}^{*}$ remained less than 10^{-2} that variation of its value did not influence the calibration error very much. In general, variation of the value of $\boldsymbol{K}_{ij}^{*}$ had little effect on the calibration error unless the parameter was grossly overestimated. As such, it is not surprising that $\boldsymbol{K}_{ij}^{*}$**s** are the least precisely determined parameters in the Nikolskii-Eisenman expressions for each electrode by both GA and simplex approaches. These results suggest, however, that the simplex approach more precisely determines parameters which contribute most to the SSE. Conversely the GA determines parameters which contribute less to the SSE than the simplex approach. As such, it suggests the advantages to be had from hybridisation of the simplex and GA approaches.

3.6 Dynamics Of The Optimisation Procedure

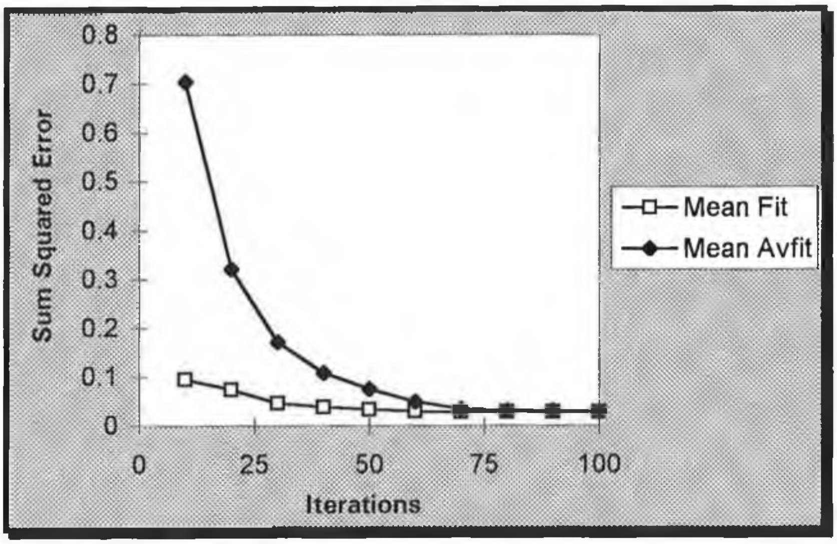


Figure 3.11 -Variation of SSE of best fitting calibration model (Mean Fit) and average population SSE (Mean Avfit) in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times

Figure (3.11) depicts the changes in the average values of the SSE of the calibration models which most closely model the response of the sodium ISE, obtained from the six repetitions of the GA. It also depicts the changes in the average SSE for the entire population averaged over the six repetitions. It can be seen that the SSE from the best fitting calibration model rapidly decreases as the GA progresses and improved very little after 80 iterations of the GA. It can also be seen that the average SSE of the population decreases as the GA progresses (as the population becomes filled with chromosomes encoding calibration expressions which model the response of the electrode increasingly well). The average SSE of the population approaches that of the individual which most closely models the response of the electrode in about 100 iterations. This suggests that when the GA finds an individual which encodes a good calibration model that the GA rapidly fills the rest of the population with chromosomes encoding very similar calibration models.

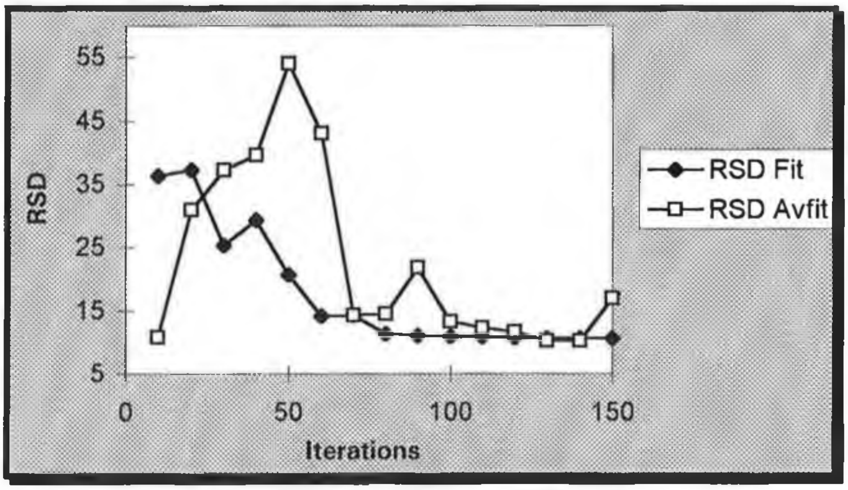


Figure 3.12 - Variation Of %RSD of best fitting calibration model(%RSD Fit) and average population SSE (%RSD Avfit) in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times

Figure (3.12) depicts the changes in the %RSDs of the error terms whose averages are depicted in figure (3.11). It can be clearly seen that the %RSDs of the errors of calibration models that most closely model the response of the electrode rapidly decrease as the GA progresses and then slow after about 80 iterations (i.e. after the average SSE of the best calibration model slows). This suggests that the populations repetitively started from different positions in the search space initially had very different models best representing the response of the electrode, but the GAs rapidly converged to a similar expression, suggesting that perhaps a global optimum had been found in the search space. The changes in the %RSD values of the average SSE of the population are more difficult to follow. It appears that in the periods of maximum decrease in %RSD of the SSE of the best calibration models in the population that the %RSDs of the average SSE of the population are increasing. This suggests that while the GAs may be finding very similar calibration models that best represent the response of the electrode that the rest of the populations are not all moving towards this model at the same rate. Some populations are becoming increasingly dominated by the particular model, while others are not. By the time the %RSD of the best calibration model has stabilised the %RSD of the average SSE of the populations has decreased again. Thereafter, while there is small change in the %RSD of the best calibration model, there is gradual decrease in the

%RSD of the average SSE of the population with sporadic increase probably due to random mutation or crossing processes occurring within the population.

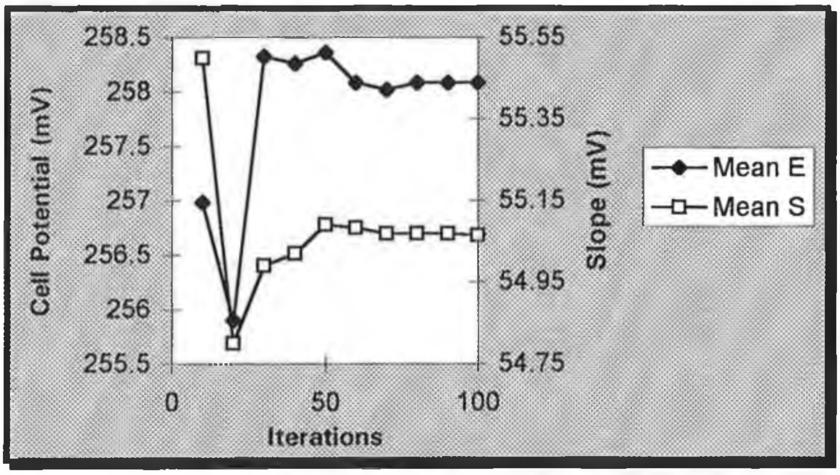


Figure 3.13 - Variation of standard cell potential (Mean E) and slope (Mean S) of best fitting calibration model in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times

Figure (3.13) depicts the changes in the values of the standard cell potential and slope parameters obtained from the chromosome which most closely models the response of the sodium ISE, averaged over the six repetitions of the GA. It can be seen that the first 30 iterations of the GA demonstrate the most dramatic changes in these calibration parameters whose final values are virtually established by about 80 iterations.

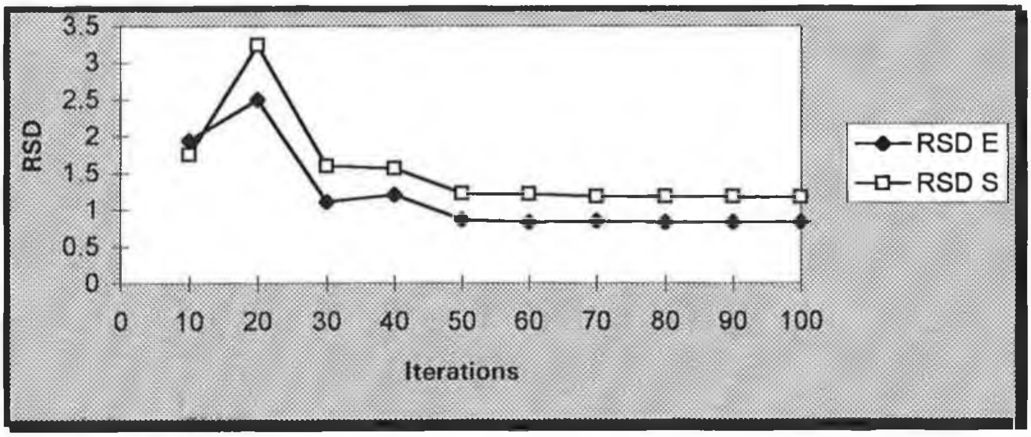


Figure 3.16 - Variation of %RSD of standard cell potential (%RSD E) and slope (%RSD S) of best fitting calibration model in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times

The %RSDs of these parameters are depicted in figure (3.14) which spike during the period of most change in the average values of the slopes and cell potentials. The %RSDs of these parameters also stabilise rapidly although slightly earlier than the average values do. This suggests that the GA converges to particular values of the slope and cell potential at the same time as it has converged to a solution in the response space and that the precision with which these parameters are determined does not change as the GA further progresses.

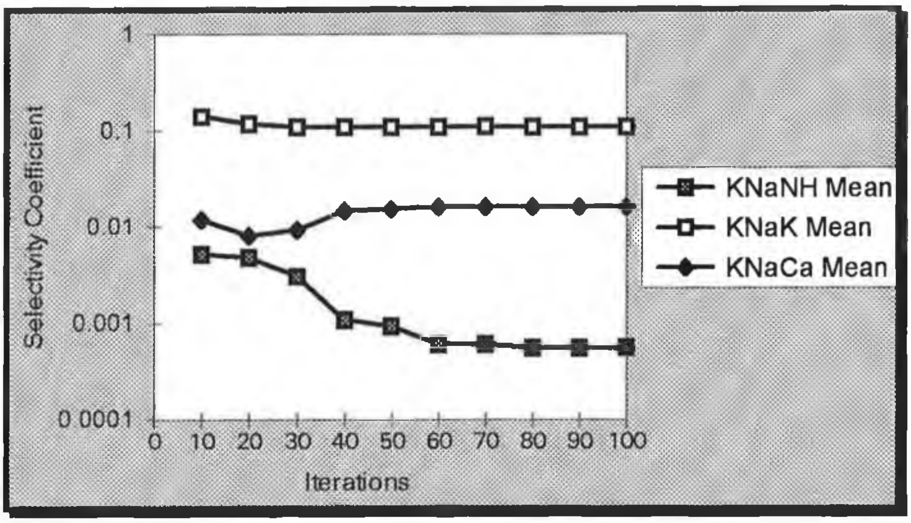


Figure 3.15 - Variation of selectivity coefficients of best fitting calibration model in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times. KNaNH, KNaK and KNaCa refer to the selectivity coefficients of the sodium electrode against ammonium, potassium and calcium respectively

Figure (3.15) depicts the changes in the selectivity coefficients of the sodium ISE against ammonium, potassium and calcium obtained from the calibration model which most closely describes the response of the electrode as the GA progressed, averaged over six repetitions of the GA. It can be seen that the selectivity coefficients of the ISE against ammonium which has the lowest magnitude of all the three selectivity coefficients in the finally determined calibration model undergoes a decrease of approximately one order of magnitude in the first 80 iterations of the GA at which point it stabilises and undergoes a smaller decrease in the later stages of the GA. The selectivity coefficient of the ISE against calcium which is the next highest to the selectivity coefficient of the electrode against ammonium undergoes less dramatic change than the previously discussed selectivity coefficient and stabilises by about 60 iterations. The selectivity coefficient of the electrode against potassium which has the highest value of the selectivity coefficients in the final model converges very rapidly within about 30 iterations of the GA.

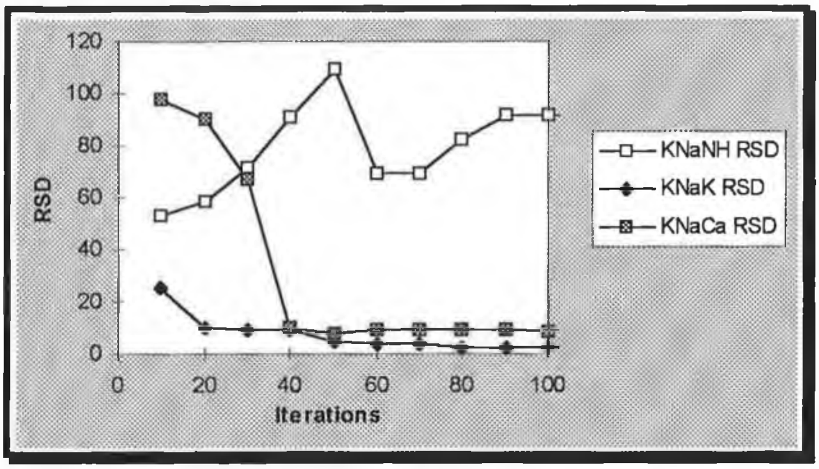


Figure 3.16 - Variation of %RSDs of the selectivity coefficients of the best fitting calibration model in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE repeated six times. KNaNH RSD ,KNaK RSD and KNaCa RSD refers to the %RSD of the selectivity coefficient of the sodium ISE against ammonium, potassium and calcium respectively.

Figure (3.16) depicts the changes in the %RSDs of the values of the selectivity coefficients depicted in figure (3.15). It can be seen that the %RSD of the selectivity coefficient of the ISE against ammonium increases as the GA progresses and generally stabilises within 100 iterations. The %RSD of the selectivity coefficient of the ISE calcium undergoes rapid decrease and stabilisation by about 80 iterations and the selectivity coefficient against potassium also undergoes rapid decrease and stabilisation by about 80 iterations. This suggests that while multiple repetitions of a GA converge to solutions involving similar values of the calibration parameters, such as E^O and S which are more likely to make a significant contribution to the SSE, which the GA is minimising, that a parameters such as the selectivity coefficient of the lowest magnitude is least precisely determined and the optimisation process of the GA may actually decrease the precision with which such parameters are determined.

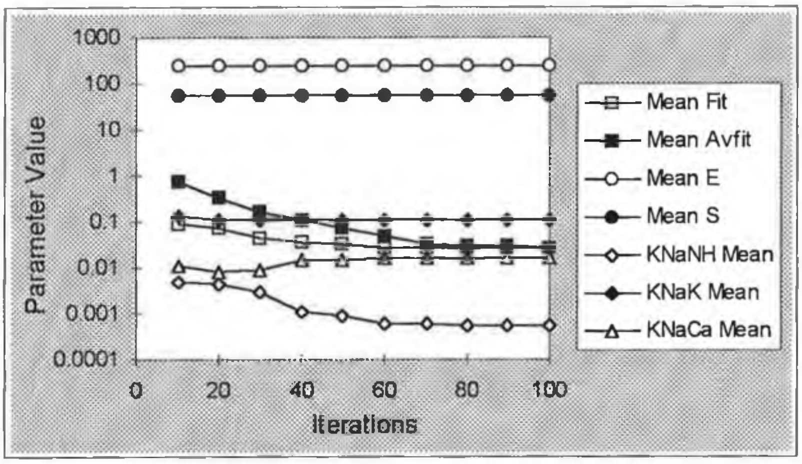


Figure 3.17 Variation of the response parameters of the best fitting calibration model in a population of 300 chromosomes with rank prescaling and stochastic remainder sampling applied to the sodium ISE six times. The terms used in the graph are as described in figures 3.11-3.16.

Figure 3.17 summarises the effects seen in figures 3.11, 3.13 and 3.15. It can be clearly seen that the slope and standard cell potentials undergo very little change during the progress of the GA. This is likely to be due to a rapid initial search for these parameters followed by filling of the population with the values for these parameters, thereby effectively reducing the dimensionality of the search space for the GA. The selectivity coefficient of the ISE against ammonia undergoes a slight change until about 20 iterations . In contrast the less significant selectivity coefficients (e.g. the selectivity coefficient of the sodium ISE against ammonium and calcium) undergo change until much later stages of the GA as the population is less likely to fill with the values of genes which make a small contribution to the fitness of a given chromosome.

3.7 Conclusion

Genetic algorithms were investigated as a means of optimising a least squares error parameter for non-linear calibration of an array of ion-selective electrodes. A variety of different modifications to the simple GA were investigated with a view to reducing the effects of premature convergence and poor search precision inherent to the GA technique. Simple linear prescaling was considered to slow down the process of diversity reduction within a population but not sufficient to reduce the effect. Rank selection reduced the effect of diversity reduction within the chromosomes of a population although the population still tended to converge more quickly on some genes relative to others depending on their significance in determining the fitness of the organism. Roulette wheel selection was considered, but found to be unsatisfactory with low degrees of prescaling because the stochastic nature if the selection process led to the possibility if losing the most fit individual within a population during evolution, remainder stochastic sampling was found to be a more useful means of selection in this respect because there was less chance of losing the most fit individual in the population. Prevention of incest during crossing as investigated as a means of reducing the reintroduction of genetic material into a population, but was found not to make any large contribution to the final result produced by the GA and because of the increased complexity of the program produced by having to keep account of chromosomes and their parents was not continued in further work. The GAs developed were then used for calibration, very high degrees of correlation were found between predicted activities of samples and their actual activities.

3.8 Bibliography

1. F.J. Sáez deViteri and D.Diamond, **Analyst** 119 (1994) 749-758.
2. R.J. Forster, F.Regan and D.Diamond, **Anal. Chem.** 63 (1991) 876-882.
3. R.J. Forster and D. Diamond, **Anal. Chem.** 64 (1992) 1721-1728.
4. D. Betteridge, A.P. Wade and A.G. Howard, **Talanta** 32 (8B) (1985) 709-722.
5. C.B. Lucasius, Towards Genetic Algorithm Methodology in Chemometrics, Ph. D. thesis, Katholieke Universiteit Nijmegen, Netherlands, 1993.
6. A.P. De Weijer, C.B. Lucasius, L. Buydens, G. Kateman, H.M. Heuvel and H. Mannee, **Anal. Chem**. 66 (1994), 23-31.
7. D. Diamond, **Electroanalysis,** in press.
8. W.H. Press, S.A. Teukolsky, B.P. Flannery and W.T. Vetterling, **Numerical Recipes in C : The Art of Scientific Computing,** Cambridge University Press, 1991.
   (a) Modified simplex p.305-309
   (b) Gauss Jordan elimination p.32-37
9. D.E. Goldberg, **Genetic Algorithms in Search, Optimization and Machine Learning,** Addison-Wesley, Reading, MA., 1989.
10. J. J. Grefenstette, **IEEE Trans. Syst. Man Cybern.** SMC-16 (1) (1986) 122-128.
11. J.E. Baker in J.J. Grefenstette (Ed.), **Adaptive Selection Methods for Genetic Algorithms,** Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum, NJ, 1985, p. 101-111.
12. L.J. Eshelman and J.D. Schaffer in J.D. Schaffer (Ed.), ***Preventing Premature Convergence in Genetic Algorithms by Preventing Incest,*** Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA, 1991. p. 115-122.
13. J.A. Nelder and R. Mead. ***Computer J.***, 7 (1965) 308
14. A.L. Horvath, **Handbook of Aqueous Electrolyte Solutions: Physical Properties, Estimation and Correlation Methods**, p.213, Ellis Horwood Ltd., Chichester, W. Sussex, England, 1985
15. O. Glyn Davies, G.J. Moody and J.D.R. Thomas. Analyst, 113 (1988), 497-503

16. K. Cunningham, G. Svehla, S.J. Harris and M.A. M^{c}Kervey. Analyst, 118 (1993) 341-345

17. E. Mansfield, **Statistics for Business and Economics: Methods and Applications ,2nd Edition**, p. A22, W.W. Norton and Co., U.S., 1983

3.9 Glossary:

Chromosome: A data structure which holds a string of parameters for the problem being solved by the GA

Fitness: A value assigned to a chromosome which reflects how well the chromosome solves the task of interest. A fitness function is used to map the parameters encoded on a chromosome to a fitness value. In this study the fitness is inversely related to the SSE of a particular calibration model encoded by a chromosome.

Gene: A subsection of a chromosome which (usually) encodes the value of a single task parameter.

Generation: An iteration of a GA

Genetic Drift: Changes in gene numbers in a population over many generations resulting from chance rather than reproduction

Hamming Distance :Number of vector components in which two vectors differ. i.e. the Hamming distance between two binary numbers means the number of bits that are different in the numbers

Schema: A pattern of gene values in a chromosomes which may include "don't care" states, i.e. for chromosomes encoded in a binary alphabet (2 characters) each schema may be represented by a string of the same length as the chromosomes but encoded in an alphabet of 3 characters (1, 0, *)

Chapter 4 Determination of Stability Constants Using Genetic Algorithms

Abstract:

A genetic algorithm (GA)-simplex hybrid approach has been developed for the determination of stability constants using calorimetric and polarographic data obtained from literature sources. The polarographic data were obtained from studies of cadmium chloride and lead with the crown ether dicyclohexyl-18-crown-6. The calorimetric data were obtained from a study of a two step addition reaction of $Hg(CN)_2$ with thiourea. The stability constants obtained using the GA-simplex hybrid approach compare favourably with the values quoted in the literature.

4.1 Introduction

This section describes the definition of stability constants and different experimental methods used for their determination, focusing specifically on polarographic and calorimetric techniques, data from which were used in this study. Problems associated with existing methods of calculating stability constants and the advantages posed by genetic algorithm approaches in these contexts are discussed. Finally the design of the genetic algorithm devised to determine stability constants are discussed placing particular emphasis on the genetic representation of the stability constants and associated equilibria and the implementation of the objective function.

4.1.1 Stability Constants

A stability constant is a thermodynamic quantity which refers to the formation of a complex $M_iL_jH_k$ in which M is usually a metal ion, L refers to a ligand which can be a simple anion or a neutral molecule of variable complexity and H refers to a proton which may be involved in the reaction (1) (2).

Mathematically, the stability constant describes the equilibrium

$$\mathbf{iM^{m+}+jL^{l-}+kH^{+} \rightleftharpoons \left(M_iL_jH_k\right)^{im+k-jl}} \qquad \mathbf{(4.1)}$$

as follows

$$\boldsymbol{\beta^{o}_{ijk} = \frac{\left\{\left(M_iL_jH_K\right)^{im+k-jl}\right\}}{\{M^{m+}\}^{i}\{L^{l-}\}^{j}\{H^{+}\}^{k}}} \qquad \mathbf{(4.2)}$$

where the terms in curly brackets refer to the activities of the species involved in the equilibrium. Since many experimental techniques such as spectroscopy and calorimetry have responses which are dependent on the concentrations of different species rather than their activities, a stoichiometric stability constant may be calculated as follows :

$$\boldsymbol{\beta_{ijk} = \frac{\left[\left(M_iL_jH_k\right)^{im+k-jl}\right]}{[M^{m+}]^{i}[L^{l-}]^{j}[H^{+}]^{k}}} \qquad \mathbf{(4.3)}$$

The terms in brackets in expression **(4.3)** refer to the free concentrations of the species involved in the equilibrium as opposed to the total concentrations of the species which are experimental variables of the investigation in question.

If the experiments are performed in media of constant ionic strength, so that the activity coefficients of the species involved do not vary during the experiment then the stoichiometric stability constant $\boldsymbol{\beta_{ijk}}$ can be converted to the thermodynamic stability constant $\boldsymbol{\beta^{o}_{ijk}}$ by multiplying it by a constant corresponding to the ratio of the activity coefficients of the metal, ligand and complexed species.

$$\beta^{o}_{ijk} = \beta_{ijk} \frac{\gamma_{\left(M_i L_j H_k\right)^{im+k-jl}}}{\left(\gamma_{M^{m+}}\right)^{i} \left(\gamma_{L^{l-}}\right)^{j} \left(\gamma_{H^{+}}\right)^{k}} \qquad (4.4)$$

Otherwise, the stoichiometric coefficient should be corrected for changes during the experiment in the activity coefficients of the species involved in the equilibria.

The activity coefficients **(γ)** of the species involved in this contribution may be calculated by means of the Davies expression (3).

$$\log \gamma_i = -0.5 z_i^2 \left(\frac{\sqrt{I}}{1+\sqrt{I}} - 0.2I \right) \qquad (4.5)$$

In which γ_i is the activity coefficient of species **i**, z_i is the charge of species **i** and **I** is the ionic strength of the medium under investigation.

There are a number of experimental techniques used for the determination of stability constants including potentiometry (4), polarography (5), NMR spectroscopy (6) and calorimetry (7). Stability constants can be calculated from the experimental data by a number of means of graphical or numerical methods (8) (9). The numerical approaches to the determination of stability constants normally involve the minimisation of a sum of squares or weighted sum of squares parameter. This is achieved by a number of different methods including the Gauss method used by the program SCOGS for pH titration data (10) and the pit mapping technique used by the program DALSFEK for spectroscopic data (11).

A major problem associated with many of these approaches arises from the numerical difficulties posed by high dimensional search spaces associated with increasingly complex equilibrium models. This is especially true for experimental methods which require the determination of additional parameters such as heats of formation in calorimetry or molar extinction coefficients in spectroscopy, thereby increasing the dimension of the search space. Because of the multi-dimensional nature of the data space involved, the GA was investigated as a means of solving the minimisation problem, because it is known from literature (12) that these algorithms are very suited to dealing such tasks.

Laouenan and Suet (13) discussed the inability of many existing programs to deal with different types of experimental data. However the GA described in this paper is readily adaptable to data from different experimental methods to produce new objective functions.

4.1.2 Polarographic Methods

The stability constants of electroactive species which undergo reversible reduction can be determined polarographically by the method of DeFord and Hume (14). Changes in the half-wave potential and diffusion current of the metal on addition of ligand can be related to the concentration of the metal by means of a term **Fo** which describes the ratio between the total concentration of metal (in all its complexed and uncomplexed forms) $\mathbf{C_M}$ and the concentration of the free metal **[M]** described by equation **(4.6)**

$$\mathbf{Fo_{(calc)} = \frac{C_M}{[M]}} \qquad \mathbf{(4.6)}$$

The term Fo can also be related to the experimental variables of the study as described by equation **(4.7)**

$$\mathbf{Fo_{(expt)} = exp\left(\frac{nF}{RT}\left[E_{1/2F} - E_{1/2C}\right]\right) + ln\left(\frac{I_{dF}}{I_{dC}}\right)} \qquad \mathbf{(4.7)}$$

where **F** refers to the Faraday constant, **R** refers to the gas constant , **T** refers to temperature in Kelvin and **n** refers to the number of electrons involved in the electrochemical process. $\mathbf{E_{1/2F}}$ and $\mathbf{E_{1/2C}}$ refer to the half wave potentials of the metal ion in the absence of ligand and in the presence of ligand, respectively, and $\mathbf{I_{dF}}$ and $\mathbf{I_{dC}}$ refer to the diffusion currents of the metal ion in the absence of ligand and in the presence of ligand, respectively.

The concentration of free metal ion is related to the total concentration of metal ion by the mass balance expressions involving the stability constants for the system.

$$C_M = \sum_0^i \sum_0^j i\beta_{ij}[M]^i[L]^j \qquad (4.8)$$

$$C_L = \sum_0^i \sum_0^j j\beta_{ij}[M]^i[L]^j \qquad (4.9)$$

The stability constants can thus be determined by a process of minimising the sum of squared errors **(SSE)** (the term error refers to the difference between the experimentally derived value of **Fo(expt)** and the value calculated from the free metal concentrations **Fo(calc)**).

$$\begin{aligned} \mathrm{SSE} &= \sum_{i=1}^{\mathrm{numdata}} \left(\left[Fo_{(\mathrm{expt})} \right]_i - \left[Fo_{(\mathrm{calc})} \right]_i \right)^2 \\ &= \sum_{i=1}^{\mathrm{numdata}} \left(\exp\left[\frac{nF}{RT} \left\{ E_{1/2F} - \left(E_{1/2C} \right)_i \right\} \right] + \ln\left[\frac{I_{dF}}{\left(I_{dC} \right)_i} \right] - \frac{\left(C_M \right)_i}{[M]_i} \right)^2 \end{aligned} \qquad (4.10)$$

The term **numdata** refers to the number of measurements performed in the experiment and the index **i** refers to the measurement **i** in a particular experiment. Leggett's program POLAG (15) which performs the least squares calculations by means of a Gauss-Newton approach, is one of the more advanced software packages for determining stability constants from polarographic data.

4.1.3 Calorimetric Determination Of Stability Constants

Calorimetric titration can be used for the determination of stability constants by the measurement of the heat evolved upon addition of a ligand solution to a metal solution (16). When the heat measurements are corrected for dilution effects, changes in heat capacity and other factors external to the complexation reaction itself, the resulting corrected heat **Qc** can be described by

$$\mathbf{Qc}_{(\mathrm{calc})} = \sum_{i=1}^{N} \Delta\mathbf{H}_i \mathbf{mol}_i \qquad \mathbf{(4.11)}$$

where **N** refers to the number of species formed, $\Delta\mathbf{H}_i$ refers to the heat of formation of a species **i**, $\mathbf{mol}_i$ refers to the number of moles of species **i** present in the reaction vessel and can be calculated from the stability constants and the total concentrations of metal and ligand used for the experiment. (17).

This indicates that in addition to determining the stability constants for the system, one must also determine the heats of formation of these complexes, thereby effectively doubling the dimension of the search space. These parameters are generally determined by minimising the sum of squares of the difference between the predicted value of heat produced in the system (by means of stability constants and heats of formation) and the experimentally calculated values as depicted in expression **(4.12)**.

$$\mathbf{SSE} = \sum_{i=1}^{\mathrm{numdata}} \left(\mathbf{Qc}_{(\exp t)} - \mathbf{Qc}_{(\mathrm{calc})}\right)^2 = \sum_{i=1}^{\mathrm{numdata}} \left(\mathbf{Qc}_{(\exp t)} - \sum_{j=1}^{N} \Delta\mathbf{H}_j \mathbf{mol}_j\right) \qquad \mathbf{(4.12)}$$

4.1.4 Genetic Algorithms

A description of the theory behind the operation of genetic algorithms is found in chapter one of this thesis and a description of its practical implementation is described in chapter 3.

The representation of an optimisation problem associated with a GA appeared suitable for designing a system which was capable of determining both the nature of the equilibrium model and the value of the stability constants involved in this equilibrium. In many software systems, the operator can determine the relative suitability of different equilibrium models by selecting different models manually for optimisation and comparing the relevant statistics associated with each model. In this GA approach a range of different models will be evaluated in parallel removing the need for manual comparison.

4.1.5 Genetic Representation Of The Stability Constant Determination Problem

Since the aim of this study was to optimise equilibrium models and stability constants, a variable length chromosome format was decided upon for the population. Each chromosome was represented as a structure which encoded the number of different metals and ligands involved in the candidate equilibrium model. The structures also described the stoichiometric numbers of the metals and ligands in the candidate equilibrium model. Figure 4.1 graphically depicts the chromosome structure.

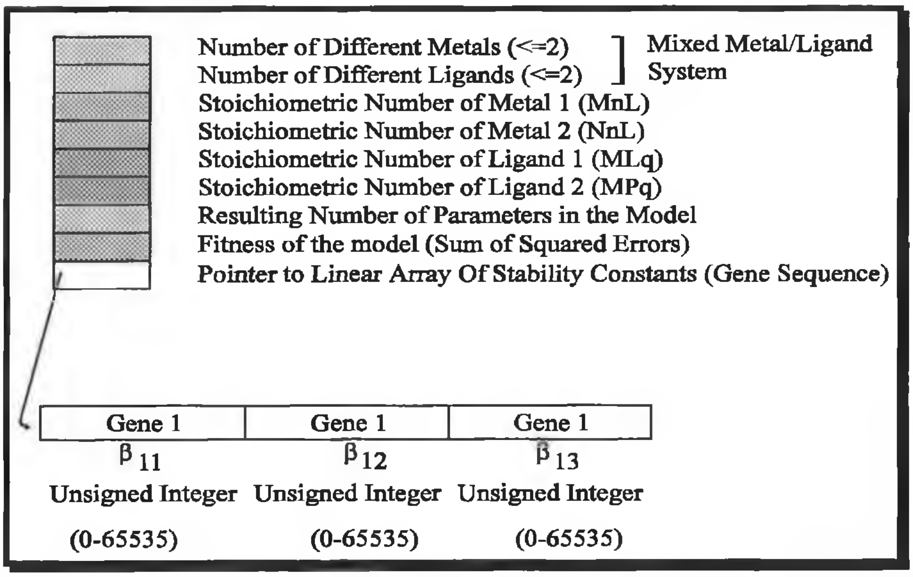

Number of Different Metals (<=2)
Number of Different Ligands (<=2)
Mixed Metal/Ligand System
Stoichiometric Number of Metal 1 (MnL)
Stoichiometric Number of Metal 2 (NnL)
Stoichiometric Number of Ligand 1 (MLq)
Stoichiometric Number of Ligand 2 (MPq)
Resulting Number of Parameters in the Model
Fitness of the model (Sum of Squared Errors)
Pointer to Linear Array Of Stability Constants (Gene Sequence)
Gene 1
Gene 1
Gene 1
β_{11}
β_{12}
β_{13}
Unsigned Integer
Unsigned Integer
Unsigned Integer
(0-65535)
(0-65535)
(0-65535)

Figure 4.1 - Genetic representation of stability constant model in a chromosome structure

This representation scheme was used to build up a series of equilibria for the lowest order complex ML to the highest order complex specified by the chromosome. This scheme was then used to determine the number of stability constants needed to determine the equilibria encoded by the chromosome and hence the total number of parameters needed to be optimised (depending on whether the experimental data used was derived from a polarographic or calorimetric analysis). The value of each parameter was represented as an integer scaled in the range 0- (2^{16}-1). The resulting string of parameters for each model were dynamically allocated and reallocated according to the breeding processes involved in the GA.

The actual configuration for the GAs used for this study is depicted in table 4.1.

Population:	200-300 chromosomes
Scaling:	Rank Scaling (18)
Rank Scaling Upper Limit	1.1
Selection	Stochastic Remainder (19)
Crossover	Single Point (19)
Crossover Restriction	Chromosome mates must encode the same number of parameters Hamming distance restriction (20)
Cross Rate	0.65
Mutation	Single Point
Mutation Rate	1.0/(Population Size)

Table 4.1- General configuration of Genetic Algorithms used for this study.

A problem with the implementation of GAs is the low precision with which a solution is determined (12), in order to improve on the end solution determined by the GA, the GA was post-hybridised with a simplex technique. In this case, members of the population which fit the experimental data well were used to fill the vertices of the simplex.

4.1.6 Implementation of Objective Function

In the previous sections discussing polarography and calorimetry, it was shown that stability constants determined by these techniques were calculated by means of optimising a sum of squared errors expression given by equations **(4.10)** and **(4.12)** respectively. It can be seen that both expression **(4.10)** and **(4.12)** involve the equilibrium concentrations of the species described by the equilibrium model. The equilibrium concentrations of the species can be calculated from the total metal and ligand concentrations (which are experimental variables) and from the proposed equilibrium model and stability constants, by means of the mass balance expressions **(4.8)** and **(4.9)**.

As such, it can be seen that the calculation of a fitness measure for a chromosome is an indirect procedure which involves using the chromosome encoded model and experimental data describing total metal and ligand concentrations to calculate the equilibrium concentrations of the species described by the model. This is achieved by a separate optimisation routine which minimises the difference between the total metal and ligand concentrations predicted from the mass balance expressions and the total metal and ligand concentrations used experimentally. These equilibrium concentrations are then used to predict a value of Fo or corrected heat value respectively for polarographic data or calorimetric data. These Fo or heat values are then compared with the experimentally determined values for these parameters to generate a fitness measure (which is inversely related to the sum squared error calculated from equations **4.10** and **4.12**) for the chromosome.

Figures 4.2 and 4.3 schematically depict the processes occurring during the evaluation of the objective function for data from polarographic and calorimetric experiments

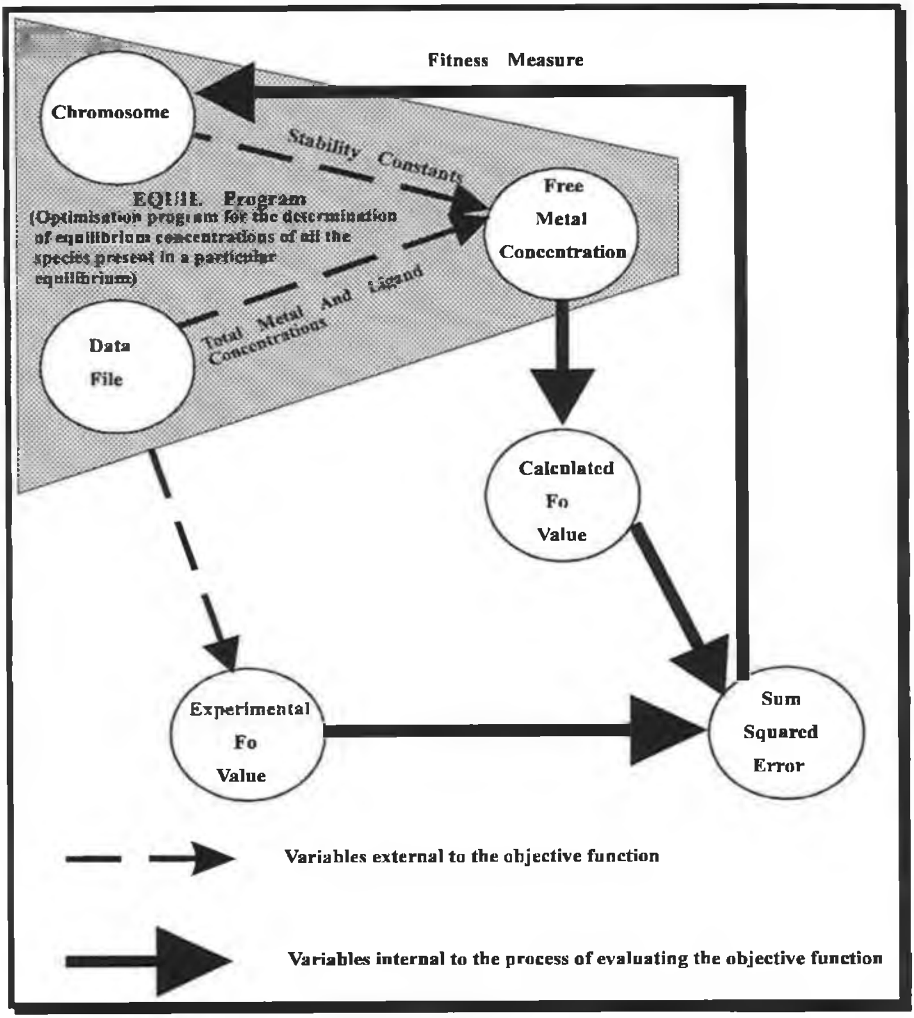

Fitness Measure
Chromosome
Stability Constants
EQUIL Program
(Optimisation program for the determination of equilibrium concentrations of all the species present in a particular equilibrium)
Free Metal Concentration
Total Metal And Ligand Concentrations
Data File
Calculated Fo Value
Experimental Fo Value
Sum Squared Error
Variables external to the objective function
Variables internal to the process of evaluating the objective function

Figure 4.2- Schematic representation of the evaluation of the objective function for polarographic data

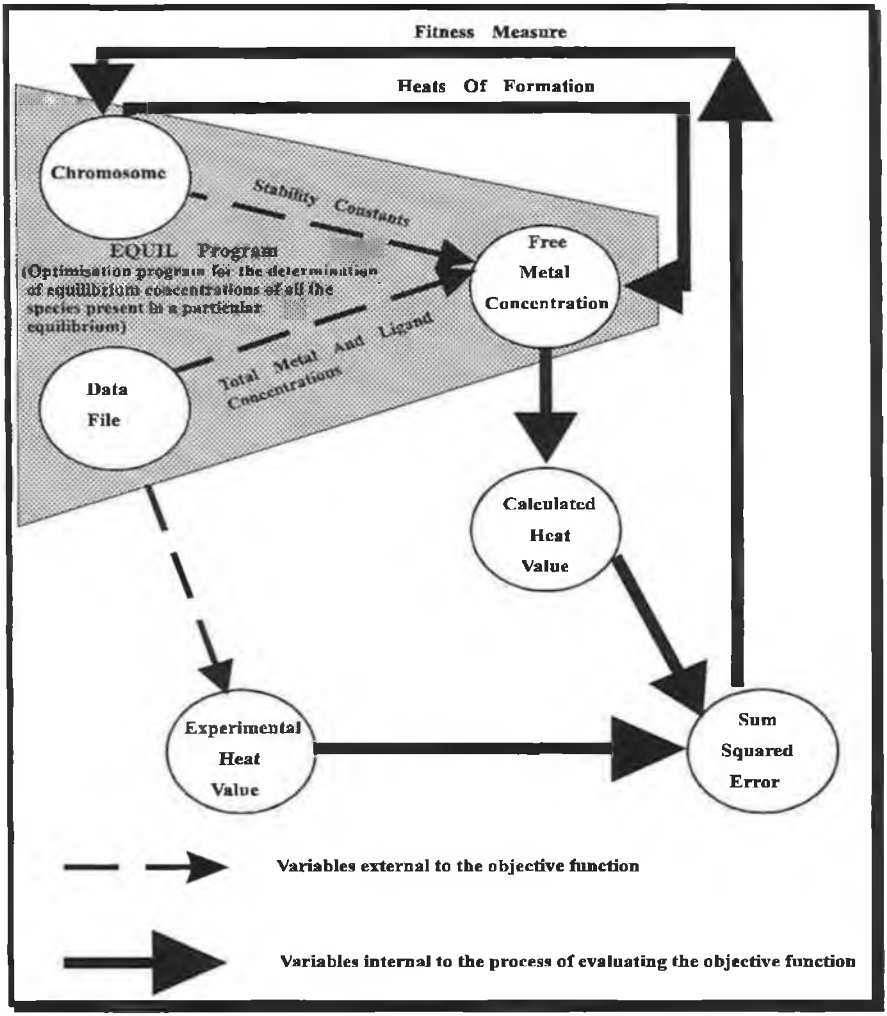

Fitness Measure
Heats Of Formation
Chromosome
Stability Constants
EQUIL Program
(Optimisation program for the determination of equilibrium concentrations of all the species present in a particular equilibrium)
Free Metal Concentration
Data File
Total Metal And Ligand Concentrations
Calculated Heat Value
Experimental Heat Value
Sum Squared Error
Variables external to the objective function
Variables internal to the process of evaluating the objective function

Figure 4.3- Schematic representation of the evaluation of the objective function for calorimetric data

4.1.7 Parameter Standard Deviations

A major advantage of computational methods of determination of stability constants over graphical methods is the ability to estimate the precisions of the model parameters determined by the optimisation procedure.

The principle of the GA approach involves the minimisation of a sum of squared errors value, calculated as the difference between an experimental observation and a corresponding value calculated from the equilibrium model as follows :

$$\mathbf{SSE} = \sum_{i=1}^{m} \mathbf{W}_i \left[\mathbf{y}_{i_{(\exp t)}} - \mathbf{f}\left(\alpha_1 \ldots \alpha_n, \mathbf{x}_{i,1} \ldots \mathbf{x}_{ik}\right)\right]^2 \qquad \mathbf{(4.13)}$$

Where $\mathbf{W_i}$ represents the weighting of each experimental point (assumed to be unity in this case), **m** represents the number of observations, $\mathbf{y_{i(exp\,t)}}$ represents the experimental observation (**fo** for polarography and heat for calorimetry) and $\mathbf{f}\left(\alpha_1 \ldots \alpha_n, \mathbf{x}_{i,1} \ldots \mathbf{x}_{ik}\right)$ represents the function calculating a value corresponding to the experimental value. In this case **f** is a function of $\mathbf{x_l}$**(l=1 to k)** independent variables (e.g. total metal and ligand concentrations) and $\boldsymbol{\alpha_j}$ **(j=1 to n)** parameters (e.g. stability constants, heats of formation).

The principle of propagation of errors described by Deming (21) can be used to determine the standard errors of the function parameters $\boldsymbol{\alpha_j}$ by means of the linear term in the Taylors expansion of the function **f**. The variance of each function parameter can be determined from the diagonal elements of the matrix $\mathbf{B\sigma^2}$. **B** is calculated as $\mathbf{(J^TWJ)^{-1}}$ where **J** is the Jacobian matrix of the function depicted as follows

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \alpha_1} & \cdots & \frac{\partial f_1}{\partial \alpha_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_m}{\partial \alpha_1} & \cdots & \frac{\partial f_m}{\partial \alpha_n} \end{bmatrix} \qquad \textbf{(4.14)}$$

σ^2 represents the variance between the experimental observation and the calculated value. The partial derivatives contained in the Jacobian were calculated numerically in this study by means of Ridder's method of polynomial extrapolation (22a) and a singular value decomposition (22b) approach was used to invert the matrix **B**.

For further details of this approach, the reader is directed to references (23), (24) and (25).

The dataflow diagram for the GA and the associated error handling software is depicted in figure 4.4.

4.2 Experimental

The GA software was written in ANSI style C running on an Atari ATW-800 transputer workstation using the Helios operating system. The equilibrium concentrations of the different species present in a given equilibrium were calculated by means of the program **EQUIL** (26) which was merged with the GA. The GA software appears in the software appendix as the program **GASTAB.C** and the simplex optimisation routine appears as the program **GASIMP.C**, the standard deviation software appears as the program **STDEV.C**.

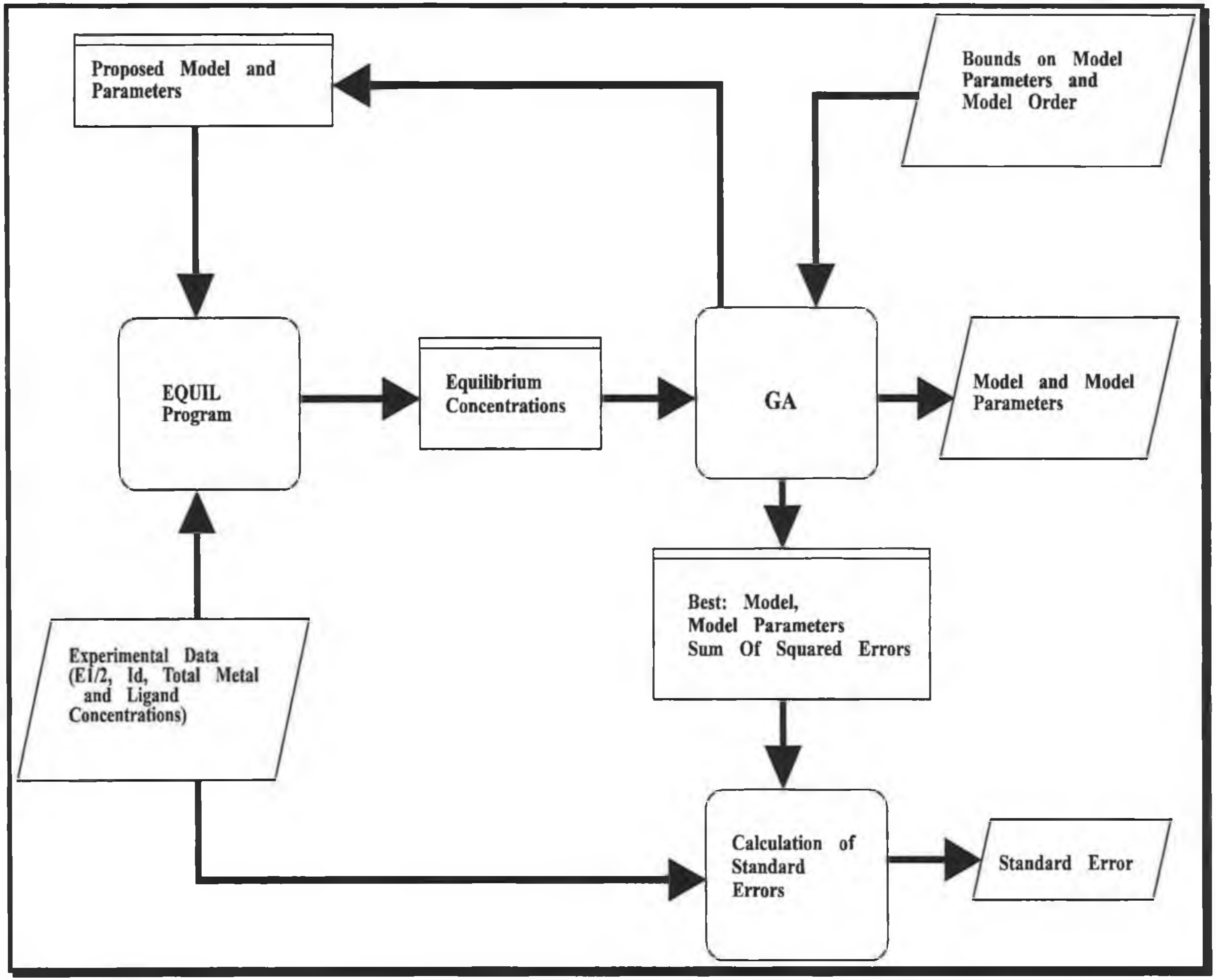

Proposed Model and Parameters
Bounds on Model Parameters and Model Order
EQUIL Program
Equilibrium Concentrations
GA
Model and Model Parameters
Best: Model, Model Parameters Sum Of Squared Errors
Experimental Data (E1/2, Id, Total Metal and Ligand Concentrations)
Calculation of Standard Errors
Standard Error

Figure 4.4 Dataflow diagram for the GA and associated error handling software

4.3 Results

4.3.1. Determination Of Stability Constants From Polarographic Experiments

4.3.1.1 Complexation Of Cadmium Chloride in a Perchlorate Medium

The ability of the GA to determine stability constants from data obtained from polarographic experiments was tested initially with data published by Heath and Hefter (27) concerning a cadmium (II) chloride system studied by means of differential pulse polarography. It involved the measurement of the peak potential and peak current from the polarographic reduction of cadmium (II) in 1.0 M sodium perchlorate with varying concentrations of sodium chloride. The total concentration of cadmium(II) was $4.0\text{x}10^{-5}$ M and the chloride concentration varied over the range 0.05 to 1.00 M. The raw data used for this study is described in table 4.2.

$[Cl^{-}]_T$ (M)	$-E_p$ (V vs. Ag/AgCl)	I_p (μA)
0	0.5885	7.10
0.05	0.5989	7.25
0.1	0.6057	7.37
0.2	0.6151	7.37
0.4	0.6277	7.37
0.6	0.6365	7.40
0.80	0.6435	7.40
1.00	0.6493	7.40

Table 4.2 Polarographic data used for the determination of the stability constants of the cadmium (II) chloride system. (E_p and I_p represent the peak potential and peak current respectively, $[\ ]_T$ represents the total concentration of the species in the square brackets)

In order to determine the stability constants from these data the original DeFord and Hume expressions **(4.7)** for data obtained from linear potential sweep (d.c) polarography were altered for the differential pulse (d.p.) technique, by replacing the half-wave potential and diffusion currents with the peak potential and peak current respectively.

Three different complexes are expected to exist, namely $CdCl^+$, $CdCl_2$ and $CdCl_3^-$ in which the stability constants, β_{11}, β_{12} and β_{13} describe the equilibria depicted in figure 4.5.

$$Cd^{2+} + Cl^- \overset{\beta_{11}}{\rightleftharpoons} CdCl^+$$

$$Cd^{2+} + 2Cl^- \overset{\beta_{12}}{\rightleftharpoons} CdCl_2$$

$$Cd^{2+} + 3Cl^- \overset{\beta_{13}}{\rightleftharpoons} CdCl_3^-$$

Figure 4.5 Complexation equilibria for cadmium chloride in a perchlorate medium

These data were also studied by Leggett (15) with the POLAG program and by Laouenan and Suet (13)with the MICMAC program. The estimates of the stability constants provided by these different studies are depicted in table 4.3.

	Literature Values		
Parameters	Heath and Hefter (27)	Leggett **(15)**	Laouenan and Suet (13)
$Log\beta_{11}$	1.352	1.329	1.328
$Log\beta_{12}$	1.748	1.736	1.735
$Log\beta_{13}$	1.544	1.514	1.511

Table 4.3 Values of the stability constants for cadmium chloride complexation determined previously from literature sources based on data acquired from a differential pulse polarography experiment performed by Heath and Hefter

It can be seen from table 4.3 that there are some differences between the values of the stability constants calculated by Heath and Hefter (27) and the other studies. It can also be seen that there is much closer agreement between the values of the stability constants determined by Leggett (15) and Laouenean and Suet (13). This was most likely to have been caused by Heath and Hefter's (27) use of a graphical method to calculate the stability constants from their experimental data whereas Leggett (15) and Laouenan and Suet (13) used more sophisticated numerical techniques.

A GA with a configuration similar to that in table 4.1 was used for this study except for employing a population containing 200 chromosomes and a rank prescaling constant of 1.3 and was allowed to iterate thorough 100 cycles before termination. The population of chromosomes was initialised to search for a suitable model for the complexation of the cadmium chloride from a range of possible models.

- $M+L \rightleftharpoons ML$
- $M+2L \rightleftharpoons ML_2$
- $M+3L \rightleftharpoons ML_3$

The GA was also initialised to search for stability constants in the range $\mathbf{0 \leq \log\beta \leq 2}$. The resulting GA was repeated five times and in each case converged to the correct model for the cadmium chloride complexation (in terms of the stoichiometric coefficients of the metal and ligand involved in the equilibria). Table 4.4 depicts the statistics concerning the values of the stability constants determined from the repetitions of the GA.

Parameters	Mean Value Of The Parameter Determined After Repetition Of The GA	Standard Deviation Of The Parameter Determined After Repetition Of The GA
$Log\beta_{11}$	1.301	0.025
$Log\beta_{12}$	1.804	0.059
$Log\beta_{13}$	1.290	0.224

Table 4.4 The mean values of the stability constants of cadmium chloride complexation determined from the differential pulse polarographic data of Heath and Hefter (27) by a GA with 200 chromosomes in its population configured as described in table 4.1 (except for a rank prescaling constant of 1.3) to search for stability constants whose values lay in the range $\mathbf{0 \leq \log\beta \leq 2}$.

These results show that the first stability constant predicted by the GA agrees more closely with that predicted by the literature sources than the other stability constants of the model. It can also be seen that the first stability constant was more precisely determined on repetition of the GA than the other stability constants. The stability constants determined from the repetitions of the GA were used to provide the vertices of a simplex to refine the model determined by the GA and the standard deviations of the parameters in the refined model were then determined by means of the techniques discussed in section 4.1.7. The resulting parameters and their standard deviations are depicted in table 4.5.

Parameters	Parameter Value After Refinement Of A GA Model Using A Simplex Method	Standard Deviation Of The Parameters Determined From Simplex Refined Model
Logβ_{11}	1.322	0.003
Logβ_{12}	1.742	0.008
Logβ_{13}	1.502	0.015

Table 4.5 - The values and standard deviations of the stability constants of cadmium chloride complexation determined from the differential pulse polarographic data of Heath and Hefter (27) following simplex refinement of the stability constants determined by a GA with 100 chromosomes in its population configured as described in table 4.1 (except for a rank prescaling constant of 1.3) to search for stability constants whose values lay in the range $\mathbf{0 \leq \log\beta \leq 2}$**.**

It can be seen from table 4.5 that there is a particular improvement in the agreement between the literature quoted values and GA-simplex determined values of the stability constants encoding higher order complexes. It can also be seen that the standard deviation of the stability constant describing the $CdCl_3^-$ has the largest value. This may be explained by examining the distribution plot for the cadmium complexes depicted in figure 4.6, where it can be seen that the $CdCl_3^-$ complex remains at a very low concentration relative to the other cadmium containing species until the later stages of the titration (i.e. chloride concentrations >0.6 M) and as such it contributes very little to the estimate of the total concentration of cadmium as determined from expression **4.8** which is in turn used for the calculation of an fo value (see expression **4.6**) which is used in the objective function in the minimisation process of the GA. Since the $CdCl_3^-$ complex contributes little to the variable used in the objective function until the later stages of the titration whereas the $CdCl^+$ and $CdCl_2$ complexes contribute more to the total cadmium concentration estimate (except for the $CdCl^+$ complex at the chloride concentrations > 0.8 M) for the duration of the experiment it is more likely that the stability constants associated with the $CdCl^+$ and $CdCl_2$ complexes will be determined more precisely than the stability constant associated with the $CdCl_3^-$ complex.

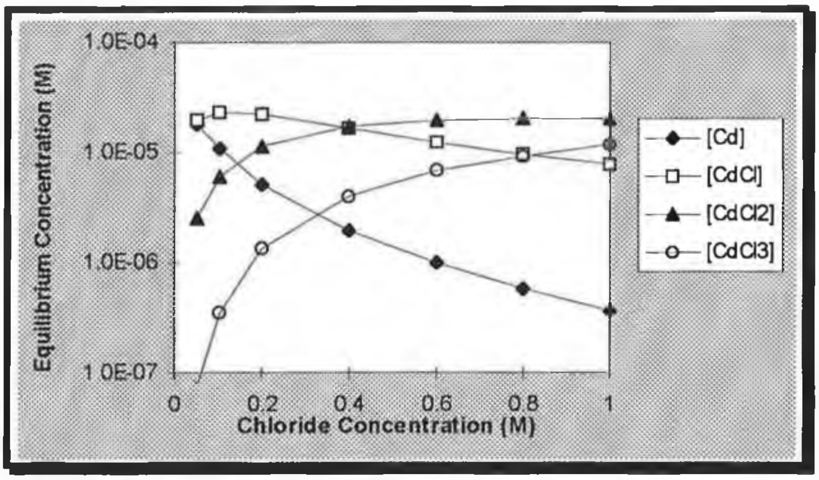


Figure 4.6 Distribution of cadmium between its free and complexed forms as a function of chloride concentration for a total cadmium concentration of 4.0x10^{-5} M as calculated from the stability constants determined by the GA-simplex hybrid (the values of these stability constants are depicted in table 4.4)

4.3.1.2 Complexation Of Lead By the Crown Ether Dicyclohexyl-18-Crown-6 In 0.1M Methanol.

This study concerned the determination of the equilibrium model and stability constants for the complexation of lead by dicyclohexyl-18-crown-6 in methanol from sampled d.c. polarography data originally investigated by Chen et al. (28) The total concentration of lead(II) varied from 0.5mM to 0.455 mM and the concentration of the crown ether varied from 0.98mM to 4.5mM.

The raw experimental data for this study are depicted in table 4.6

$[\text{Ligand}]_T$ (10^{-3} M)	$[\text{Lead (II)}]_T$ (10^{-3} M)	$E_{1/2}$ (V)	I_d (μA)
-	0.500	-0.3038	1.290
0.980	0.490	-0.4300	1.211
1.456	0.485	-0.4377	1.217
1.923	0.480	-0.4412	1.211
2.381	0.476	-0.4442	1.202
2.830	0.472	-0.4470	1.201
3.704	0.463	-0.4511	1.181
4.128	0.459	-0.4532	1.181
4.546	0.455	-0.4551	1.178

Table 4.6 - Polarographic data for the system Pb^{2+}/ dicyclohexyl-18-crown-6/ 0.1M TBAP/ methanol ($[\ ]_T$ represents the total concentration of the species in the square brackets)

The study by Chen et al. involved the use of the POLAG program which suggested that two species could exist described by the equilibria depicted in figure 4.7

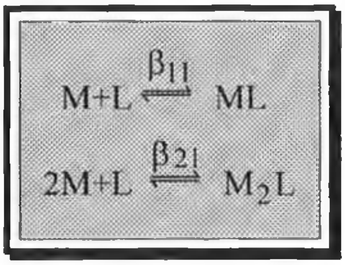

$M+L \overset{\beta_{11}}{\rightleftharpoons} ML$
$2M+L \overset{\beta_{21}}{\rightleftharpoons} M_2L$

Figure 4.7 Complexation equilibrium models for lead by dicyclohexyl-18-crown-6 in methanol

The values of the stability constants determined by the POLAG program for the experimental data were $\log\beta_{11}=7.46\pm0.01$ and $\log\beta_{21}=14.2\pm0.1$.

The initial GA used for this study had a population of 100 chromosomes and a crosspool of 30 chromosomes. The population of chromosomes was initialised to search for a suitable model for the complexation of the lead by the crown ether from a range of possible models

* $M+L \rightleftharpoons ML$

* $M+2L \rightleftharpoons ML_2$

* $2M+L \rightleftharpoons M_2L$

* $2M+2L \rightleftharpoons M_2L_2$

The GA was also initialised to search for stability constants in the range $\mathbf{7 \leq \log\beta \leq 18}$.

Initial studies with the GA yielded , $2M+L \rightleftharpoons M_2L$ and $2M+2L \rightleftharpoons M_2L_2$ models with stability constants β_{11} and β_{12} (when the model encoded an ML_2 complex) in the approximate range of 7.3 to 7.5 and β_{21} and β_{22} (when the model encoded an M_2L_2 complex) in the approximate range 13.7 to 15.7. Based on this initial study the range of models to be searched by the GA was reduced to those encoding $2M+L \rightleftharpoons M_2L$ and $2M+2L \rightleftharpoons M_2L_2$ equilibria and the stability constant range to be searched was reduced to $\mathbf{7 \leq \log\beta \leq 16}$. The newly configured GA yielded models encoding $2M+L \rightleftharpoons M_2L$ and $2M+2L \rightleftharpoons M_2L_2$ equilibria with very little difference in their SSE. As such it was decided to use members from the population yielding the simpler equilibrium model to construct the vertices of a simplex for further refinement of the model.

Table 4.7 depicts the values of the stability constants determined for the model encoding the $2M+L \rightleftharpoons M_2L$ equilibrium determined by the GA and GA-simplex hybrid.

Optimisation Technique	β_{11}	β_{21}	SSE
GA	7.47	14.04	$1.0\text{x}10^{-4}$
GA-simplex hybrid	7.459 (0.08)	14.147 (0.66)	$8.4\text{x}10^{-5}$
POLAG program (28)	7.46 (0.01)	14.3 (0.1)	$1.9\text{x}10^{-6}$

Table 4.7 Values of the stability constants determined for the complexation of lead(II) by dicyclohexyl-18-crown-6 in 0.1M methanol by a GA configured as described in the text a GA-simplex hybrid and the POLAG program. The terms in brackets beside the stability constants determined by the GA-simplex hybrid refer to the standard deviations of the parameters determined in the manner described in section 4.1.7.

It can be seen from table 4.7 that while the POLAG program finds a better fitting model than the GA-simplex hybrid, that there is a close agreement between the stability constants determined by the POLAG program and the GA-simplex hybrid.

4.3.2 Determination of stability constants from calorimetric experiments

The ability of the GA to determine stability constants from calorimetric data was tested with data published by Eatough, Izatt and Christensen (29) concerning the two step addition reaction of $Hg(CN)_2$ with thiourea whose resulting equilibria can be described by the models depicted in figure 4.8

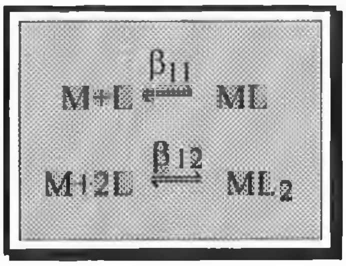


Figure 4.8 Complexation equilibrium models for $Hg(CN)_2$ with thiourea

The experiment involved the measurement of the heat released on the titration of 0.09999L of 0.03010 F $Hg(CN)_2$ with 1.422 F thiourea. The raw experimental data used for this study is depicted in table 4.8.

Time (Sec)	Volume Of Added Ligand (ml)	$[\text{Thiourea}]_T$ (M)	$[\text{Hg(CN)}_2]_T$ (M)	Corrected Heat Qc (Cal)
54.06	0.307493	0.00436	0.030008	-0.44
114.06	0.648773	0.009167	0.029906	-0.91
174.06	0.990053	0.013942	0.029805	-1.39
234.06	1.331333	0.018685	0.029704	-1.85
294.06	1.672613	0.023396	0.029605	-2.31
354.06	2.013893	0.028075	0.029506	-2.76
414.06	2.355173	0.032723	0.029407	-3.2
474.06	2.696453	0.03734	0.02931	-3.61
534.06	3.037733	0.041927	0.029213	-4.02
594.06	3.379013	0.046484	0.029116	-4.42
654.06	3.720293	0.05101	0.02902	-4.81
714.06	4.061573	0.055507	0.028925	-5.18
774.06	4.402853	0.059974	0.028831	-5.54
834.06	4.744133	0.064412	0.028737	-5.89
894.06	5.085413	0.068822	0.028643	-6.23
954.06	5.426693	0.073202	0.02855	-6.56
1014.06	5.767973	0.077555	0.028458	-6.88
1074.06	6.109253	0.08188	0.028367	-7.18

Table 4.8 Calorimetric data for the two step addition reaction of thiourea to $Hg(CN)_2$. The titrant delivery rate used for calculation of the volume of thiourea added, was 0.005688 ml/sec.

The stability constants calculated from the resulting data by Eatough et al. (29) had the values $\log\beta_{11}$=2.074, $\log\beta_{12}$=2.644 and the heats of formation for the resulting complexes had the values ΔH for the ML complex =-1.279 kcal/mole and ΔH for the ML_2 complex = -9.957 kcal/mole. The GA for this study was initialised to search for stability constants in the range $\mathbf{0 \leq \log\beta \leq 3}$ and heat of formation values in the range $\mathbf{-1.0kcal/mole \leq \Delta H \leq -10.0kcal/mole}$. Repetitions of the GA yielded either ML_2 or ML_3 models with little difference in their SSE values but widely varying ΔH values. Members of the population from the GA which converged to the simplest model (i.e. ML_2) were used to construct the vertices of a simplex for refinement of the model. The results of this study are depicted in table 4.9.

	Literature	GA	GA-simplex hybrid
$\mathrm{Log}\beta_{11}$	2.074	1.411	2.124 (0.132)
$\mathrm{Log}\beta_{12}$	2.644	2.186	2.687 (0.105)
ΔH(ML) kcal/mole	-1.279	-2.303	-1.248 (0.08)
ΔH(ML_2) kcal/mole	-9.957	-7.231	-10.113 (0.57)
SSE	0.00018	0.000316	0.000215

Table 4.9 Comparison of stability constants and heats of formation determined from a calorimetric study of the two step addition reaction of thiourea with $Hg(CN)_2$ from reference 29 by GA and GA-simplex hybrid with the values for these variables determined in the reference. The terms in brackets beside the stability constants determined by the GA-simplex hybrid refer to the standard deviations of the parameters determined in the manner described in section 4.1.7.

As can be seen from the table there is an obvious difference between the parameters of the models predicted by the different techniques with relatively large standard deviations for the stability constant and the heat of formation for the ML_2 complex. However on examination of the correlation matrix for the stability constants and heats of formation determined by the GA-simplex hybrid as depicted in table 4.10, it can be seen that the first stability constant is very highly correlated with the second stability constant. In addition the first heat of formation of the ML complex and the second stability constant are also highly correlated with the heat of formation for the ML complex.

	$\mathrm{Log}\beta_{11}$	$\mathrm{Log}\beta_{12}$	ΔH(ML)	ΔH(ML_2)
$\mathrm{Log}\beta_{11}$	1.000	0.987	0.999	-0.878
$\mathrm{Log}\beta_{12}$	0.987	1.000	0.990	-0.790
ΔH(ML)	0.999	0.990	1.000	-0.868
ΔH(ML_2)	-0.878	-0.790	-0.868	1.000

Table 4.7 Correlation matrix for the stability constants and heats of formation for the two-step addition reaction of $Hg(CN)_2$ with thiourea

This high degree of correlation between some of the variables may have caused some of the problems encountered by the GA by making it difficult to find a unique solution for the modelling problem in the highly correlated variables. In addition to this it can be seen from figure 4.9 that the ML_2 complex has a very low concentration relative to the concentration of the ML complex until a thiourea concentration of about 0.03 M is reached. As such it may have been more difficult to estimate the heat contribution from the ML_2 complex as precisely as the heat contribution from the ML complex as is demonstrated by the larger standard deviation of the former in table 4.9.

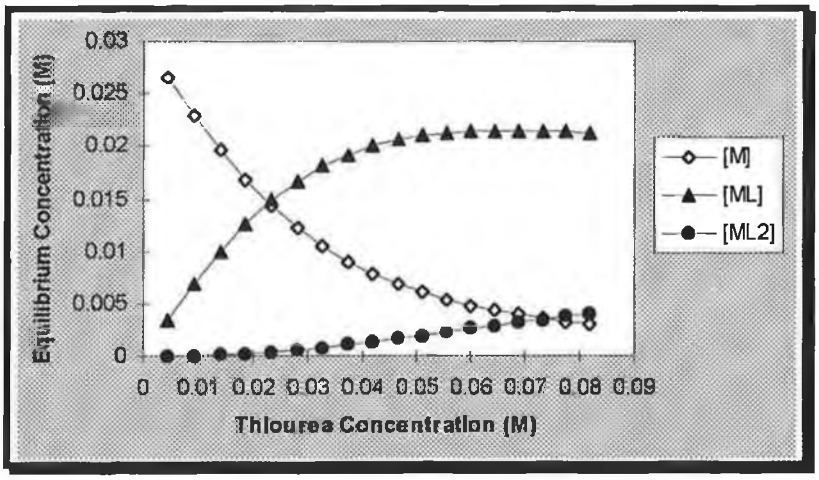


Figure 4.9 Distribution of $Hg(CN)_2$ between its free and complexed forms as a function of thiourea concentration for a total $Hg(CN)_2$ concentration varying between 0.03M and 0.028M as calculated from the stability constants determined by the GA-simplex hybrid (the values of these stability constants are depicted in table 4.6)

4.4 Conclusion

It can be seen from the preceding discussions that the technique of variable chromosomes length genetic algorithms hybridised with the simplex technique have the potential for wide application in equilibrium studies. The ease with which the objective function can be redefined indicates that this approach could be used for data from a broad range of experimental techniques and lens itself to studies involving data from combined techniques. The variable chromosome length GA itself could be improved by the use of an f-test to compare to increase competition between the different models encoded on the chromosomes. This might be achieved by developing f-distributions for each chromosome to fuzzify their objective functions. In addition it would be of interest to develop methods of dynamically rescaling the search space as the GA progressed, to improve the resolution of the search and reduce the possibility of premature convergence.

4.5 Bibliography

1. H.S. Rossotti, **Talanta**, (21) 1974, 809-829.
2. D. PH. Zollinger, **Computerized Methods For The Determination Of Stability Constants**, Ph.D. Thesis, University of Twente, 1986.
3. A.L. Horvath, **Handbook of Aqueous Electrolyte Solutions: Physical Properties, Estimation and Correlation Methods**, p.213, Ellis Horwood Ltd., Chichester, W. Sussex, England, 1985
4. G. Michaux and J. Reisse, **J. Am. Chem. Soc.,** (104) 1982, 6895-6899
5. C. Boudon, F. Peter and M. Gross, **J. Electroanal. Chem. Interfacial Electrochem.**, (117) 1981, 65-86
6. H.S. Gold and M.R. Rice, **Talanta**, (29) 1982, 637-640
7. P.U. Frueh, J.T. Clerc and W. Simon, **Helv. Chim. Acta**, (54) 1971, 1445-1450
8. F.Gaizer, **Co-ordination Chemistry Reviews**, (27) 1979, 195-222.
9. F.J.C. Rossotti, H.S. Rossotti and R.J. Whewell, **J. Inorg. Nucl. Chem.** (33) 1971, 2051-2065.
10. I.G. Sayce, **Talanta** (15), 1968, 1397-1411.
11. R.M. Alcock, F.R. Hartley and D.E. Rogers, **J. Chem. Soc. Dalton Transactions**, 1978, 115-123.
12. C.B. Lucasius, **Towards Genetic Algorithm Methodology in Chemometrics**, Ph.D. thesis, Katholieke Universiteit Nijmegen, Netherlands, 1993
13. A. Laouenan and E. Suet, **Talanta** (32) 4, 1985, 245-255.
14. D.D. DeFord and D.N. Hume, **J. Am. Chem. Soc.** (73) 1951, 5321-5323.
15. D.J. Leggett, **Talanta**, (27) 1980, 787-793
16. J.J. Christensen, J. Ruckman, D.J. Eatough and R.M. Izatt, **Thermochim. Acta** (3) 1972, 203-218.

17. D.J. Eatough, J.J. Christensen and R.M. Izatt, **Thermochim. Acta** (3) 1972, 219-232.

18. J.E. Baker, **Adaptive Selection Methods For Genetic Algorithms** in J.J. Grefenstette (editor), **Proceedings of the First International Conference on Genetic Algorithms**, p.101, Lawrence Erlbaum Associates, Hillsdale N.J. 1985.

19. D.E. Goldberg, **Genetic Algorithms in Search, Optimization and Machine Learning**, Addison-Wesley, Reading MA, 1989.

20. L.J. Eshelman and J.D. Schaffer, **Preventing Premature Convergence By Preventing Incest**, in R.K. Belew and L.B. Booker (editors), **Proceedings of the Fourth International Conference on Genetic Algorithms,** p.115. Morgan Kauffman, San Mateo CA, 1991.

21. W.Edwards Deming, **Statistical Adjustment of Data**, Dover Publications Inc., N.Y. 1964.

22. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, **Numerical Recipes in C** (2^{nd} Edition), Cambridge University Press, 1992. **(a)**-Numerical Differentiation 186-189, **(b)**-Singular Value Decomposition 59-70.

23. W.E. Wentworth, **Journal of Chemical Education,** (42) 2 1965, 96-103.

24 E. Casassus, R.Tauler and M. Filella, **Anal. Chim. Acta,** (191) 1986, 399-411.

25. D.L.Massart, B.G.M. Vandeginste, S.N. Deming, Y.Michotte and L. Kaufman, **Chemometrics:A Textbook**, Elsevier Science Publishers B.V. The Netherlands 1988.

26. M.Bos and H.Q.J. Meershoek, **Anal. Chim. Acta,** (61) 1972, 50-64.

27. G.A. Heath and G. Hefter, **J. Electroanal. Chem**., (84) 1977, 295-302;

28. L. Chen, M.Bos, P.D.J. Grootenhuis, A. Christenhusz, E. Hoogendam, D.N. Reinhoudt and W.E. Van Der Linden, **Anal Chim Acta,** (201) 1987, 117-125.

29. D.J. Eatough, R.M. Izatt and J.J. Christensen, **Thermochimica Acta,** (3) 1972, 233-246

Chapter 5. Discussion

The study of the application of artificial neural networks (ANNs) and genetic algorithms (GAs) to chemical problems is a new area of research. The number of publications in this area has dramatically increased in the last two or three years (see table 5.1) as different types of networks, different training algorithms and different modifications to the GA are being investigated in the context of solving chemical problems.

	Number of Publications	
Year	Neural Networks Keyword	Genetic Algorithms Keyword
1993	40	7
1992	17	1
1991	7	1

Table 5.1 - The number of publications with the keywords neural networks and genetic algorithms from the years 1991-1993 recorded in the Royal Society of Chemistry Analytical Abstracts Database (CD-ROM) 1980-1994, SilverPlatter Version 3.11 (as of September 1994)

While the ANN and GA techniques have been used in other disciplines for a longer time, it is only recently that the results of these studies have been presented in formats which make accessible reading for those of a lesser technical knowledge of mathematics. From my own personal experience, I found that my initial studies of these techniques frustrated by the mass of new terminology and new concepts which I had to ingest before obtaining the vaguest idea of where to start.

While software packages are becoming increasingly available which enable people to learn about ANNs and GAs more quickly (because they don't have to expend the time in developing their own software) these packages, like most packages which employ more conventional statistical analysis, should be approached with a degree of caution. ANNs and GAs appear deceptively simple, and with such software it might appear that their use is merely a question of throwing data at the technique and awaiting the "correct answer" to be "pop-out".

ANNs and GAs should not be considered as "magic bullets" which solve every problem nor should they be used blindly as "black-boxes". This thesis is not written as a "how-to" guide to ANNs and GAs, rather, it is a chronicle of my studies from an initial condition little or no knowledge to a stage of hopefully a little more enlightenment. In retrospect, many of the experiments described in this thesis were not designed in the best possible fashion and have some important failings which will be discussed later. Unfortunately, research is not a linear process from a state of complete ignorance to understanding.

Another aspect of this discussion should include a consideration of the reasons for a chemist using GAs and ANNs. These techniques are extremely alluring and it is very easy to lose sight of the chemical problem which provided the initial reason for using the GA or ANN. Personally, my chemical knowledge provided a contextual background understanding of the chemical problem of interest. However, that knowledge existed more in terms of an awareness of the nature of the available information and the required information from the GA or ANN. However, while this understanding helped in designing the GA or ANN to solve the chemical problem. the emphasis tended to lie more in the design of the GA or ANN than in the underlying chemistry. In retrospect, this may have not been the best approach to take, it would probably have been better to have more control over the quantity and quality of the data used for the studies. It would also have been desirable to have a clearer view of the applications of these techniques to "real-life" situations.

In discussing some of the features of using GAs and ANNs, it is hoped that it will help to reduce the amount of time and effort that a novice would spend in learning about

GAs and ANNs before getting to the stage of being able to apply them to anything chemical. This is not a listing of different approaches to use, more of general features to be aware of when using these techniques. It should be recognised however that the interpretations of these features and the approaches taken in this thesis are specific for the problems in question and undoubtedly will not be generally applicable to every problem.

As discussed earlier, GAs and ANNs should not be used blindly to solve chemical problems. There may be other methods of approaching these problems. While GAs and ANNs are very powerful techniques, they also have their drawbacks. For ANNs, large data sets should be available for training (it is difficult to quantify the word "large" because it depends on the problem in hand, however for noisy data it is recommended that there should be approximately ten times as many training patterns as weights in a network (1a)). However this is often not practical because of a limited availability of data. Hence, in retrospect, the small number of patterns used in the ANNs study in chapter 2, relative to the number of weights in the networks studied, was a major shortcoming in this work and limits the scope of its conclusions. For both ANNs and GAs long computational times may be involved (the computational time will vary depending on the particular problem, the software implementation and the hardware used, the ANN study in chapter 2 would run for about an hour on a SPARC workstation and the GA used in chapter 4 modelling an M_2L system would run for about 3 to 4 hours on a 66 MHz 486). These two requirements may be prohibitive for the application of interest. Another consideration is that it requires a certain degree of skill to use GAs and ANNs, although this is also true of alternative numerical or statistical techniques.

One of the major roles of a chemometrician using ANNs and GAs is identifying in very clear terms the nature of the chemical problem to be solved. This involves defining the available or input data and the desired or output data. For the ANN studies in this thesis, the input data were the potential (mV) responses of a series of ISEs and the output data were the classifications of the presence/absence of a particular cation. For the GA study of mulitvariate calibration described in chapter 3, the input data were the potential (mV) responses of a series of ISEs and the activities of the series of cations which evoked the responses. The output data in this study were the standard cell potentials, slopes and selectivity coefficients of the ISEs. For the GA study of stability constant determination, the input data were half-wave potentials, diffusion currents and metal and ligand concentrations for polarographic experiments and corrected heat and metal and ligand concentrations for calorimetric experiments. The output data were equilibrium models and stability constants. For the calorimetric experiment, the heats of formation of the complexes involved in the equilibria were additional output data. The definition of the input and output data may not always be clear and requires some understanding of the underlying chemical problem itself.

Given that the input and output data have been decided upon, the next hurdle involves deciding on how to present the data. Part of the problem involves deciding on how much of the input data is actually needed for the task in question. Wavelet transforms (1b), PCA or a range of other techniques could be considered as potential initial screens for reducing the dimensionality of a set of input data. In retrospect, the FIA patterns could have been presented to the network in chapter 2 with a smaller number of inputs, using the techniques just discussed. The large number of inputs (240) used for these patterns meant that there was also a large number of connection weights in the network (e.g. 13423 connections for a network with 55 neurons in the hidden layer and 3 output neurons). The number of patterns available (70) (based on the eight independent experimentally acquired patterns) for training was inadequate for the task of training a network of this size and the network was clearly overdetermined. Another presentation issue refers to the scaling of data, especially if the input and output data refer to variables of widely differing ranges. In this type of situation it is preferable to scale each variable separately to prevent the effect of one variable dominating the others. In chapter 2, the input patterns were scaled in the range 0 to 1 to match the output range of a sigmoid, which turned out to be a rather fortuitous. Bos (1c) has discussed how inputs should be scaled to a similar magnitude to the activations of the neurons and Annema (2) has shown how unipolar inputs (inputs scaled in a range covering only one sign e.g. 0 to 1 instead of +0.5 to -0.5) would result in faster learning by reducing the amount of time spent in temporary minima.

For GAs, variables can be represented in a binary or floating point form on a chromosome. The work described in this thesis has focused on binary representation. A number of things need to be considered when using this form of representation. These considerations include how many bits should be used to represent the variables. Another consideration refers to the search range specified for the variable. These two considerations will determine the resolution with which a variable can be determined and the number of bits used to represent the variable will also play a part in determining the memory requirements of the program (which is a consideration when allocating and/or reallocating memory to a population during the running of a GA). In the work described in this thesis, variables were represented by 16 bit integers . This was decided upon in an attempt to achieve a compromise between the minimum of 8 bit characters and 64 bit double precision floating point data types defined under the ANSI C standard. As mentioned above the search range specified for a variable also determines the resolution with which a variable is evaluated. A problem with this arises if the optimal value for a particular parameter lies outside the search range. If this occurs, the GA will not be able to find the optimal value for the parameter. If the parameter is a major contributor to SSE (1/fitness) of the particular problem, the SSE will remain high (the term "high" itself depends on the problem of interest, as is the question of what is acceptably and unacceptably high, but for analytical purposes a difference between a predicted and true value of less than 10% is desirable).

In cases where there is little information available about the likely range for a particular variable, a GA with a broad range for the variable (covering maybe 100s or 1000s of orders of magnitude depending on the likely magnitude of the variable itself) might be used to provide an initial search for the variable. The results of this initial GA could then be used to narrow the search range for another GA, thereby enabling a more refined search for the variable. Another approach might be to narrow the search range dynamically as the GA progresses. Although this approach has not been used in this thesis it could be expected that there might be difficulties with its implementation. These difficulties would arise in deciding at what point in the operation of the GA, and at what rate the search range is decreased. This might be problematic because there would always be the risk, that a particular area could be removed from the search for a particular variable. This removed area might later prove beneficial in combination with other values for the other variables encoded on the chromosome.

Having decided upon the nature and presentation of the input and output data to a ANN, the next consideration involves the choice of the training and testing sets from this data and deciding whether a cross validation set is needed. The training and testing sets should be chosen in a fashion which attempts to maintain a uniform distribution of the patterns (in a classification problem) or values of the variables (in function estimation), in order to prevent the model formed by the network from being biased. The training and testing sets should also be constructed considering that networks interpolate better than they extrapolate.

The next question for the user refers to the topology of the network. The number of input units and output neurons are determined by the dimensionality of the input and output data as presented to the network. However, the number of hidden layers and the number of units within those layers is entirely up to the user. The number of hidden layers to use is a contentious issue. Lippmann (3) showed that two hidden layers could potentially solve almost any pattern classification task. The literature concerning the theory behind these issues, especially in relation to function approximation, tends to be mathematically intense (and contradictory when dealing with the Kolmogorov theorem (4,5)). If possible a single hidden layer should be used, for the simplistic reason of reducing both the computational time of the training algorithm and the complexity of the network.

The question of how many neurons to use in a hidden layer is also controversial. It is very much dependent on the problem in hand. However the underlying thought in many attempts to choose the optimal number of neurons in the hidden layer assumes that if too few neurons are used for the application, the network will not be able to form a representation of the problem. Similarly, it is thought that if too many neurons are used in the hidden layer, the generalisation ability of the network will be deteriorated. The quantification for the term "too few neurons" can be based on the observation that a network will not be able to produce the desired outputs to the patterns in the training set. This occurs because the network will not be able to form a representation of the mapping sampled by the training set. Similarly the quantification for the term "too many neurons" might be based on an increase in the MSE on the test set (with increasing number of neurons in the hidden layer) because of generalisation difficulties associated with overfitting.

Figures 5.1 is a grossly exaggerated simplification of what might be seen when varying the number of neurons in the hidden layer, according to this viewpoint. In practice plots like this are not likely to be seen.

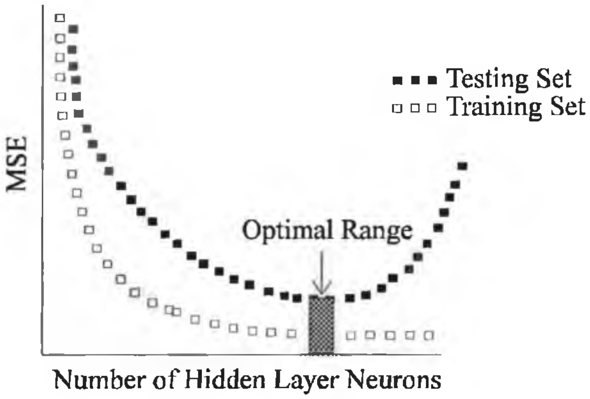


Figure 5.1 *Simplification of the possible variation of MSE of a training and a testing set with variation in the number of neurons in the hidden layer of an ANN*

Using this approach, the optimal number of neurons in the hidden layer could be taken as the minimum number of neurons which enables a network to form a mapping with an adequate MSE on the training set. The range of hidden layer neurons which could be chosen according to this criterion is depicted as the optimal range in figure 5.1. According to this view, the optimal choice of hidden layer neurons is also the number of neurons which minimises the MSE on the training set.

This approach takes an extremely simplistic view of the performance of a network with the number of neurons in the hidden layer. In practice, the behaviour of networks in such studies are complicated by changes in the initialisation of the weight space. As such this approach of manually optimising a network topology has doubtful use. However there are a number of techniques concerned with automated optimisation of the number of neurons in the hidden layer. These can be divided into two major classes, namely growing and pruning techniques. Growing techniques are based on the idea of taking a small network e.g. one neuron in the hidden layer and progressively adding more neurons to the hidden layer. Hines et al (6) compare three growing algorithms namely RCE, Alpaydin and RV and found the Alpaydin algorithm the most efficient in terms of learning speed and number of neurons in the hidden layer. Pruning techniques take oversized networks and progressively remove hidden layer neurons or connections which make little contribution to the overall output of the network. Reed (7) in his survey of pruning techniques discusses the problems associated with these techniques, these include the difficulty in deciding when to stop pruning and also the problem of possible correlations between the weights or neurons being deleted. While these techniques have not been used in the studies described in this thesis, they would appear a potentially useful area of future research.

Having decided upon (a) the nature and presentation of the input and output data to a ANN, (b) the composition of the training and testing sets and (c) the possible topology of the network, the next stage to consider is the actual training of the network. In this context it is necessary to consider the type of training algorithm being used, the values of any adjustable parameters such as momentum and learning rate and the criterion by which training will be terminated.

The backpropagation algorithm is based on a gradient descent approach (through the space formed by the network weights) to minimise the difference between the desired and actual output of the network. The number of iterations required to reach a minimum in the weight space with this form of optimisation may be very large (the term “large” depends on the problem in hand but can mean anything from hundreds to thousands of iterations for chemical problems). A particular area of research attempting to reduce the number of iterations required for optimisation use information concerning the second derivative of the error surface. Van Der Smagt (8) made a comparison of different optimisation techniques for this purpose and found that a much shorter training time was achieved using the second derivative of the error function, although the standard back-propagation algorithm was less likely to get stuck in local minima. While training algorithms which use information concerning the second derivative of the error surface have not been used in the studies described in this thesis, these techniques would appear to be potentially useful for future research. An example of these techniques is the scaled conjugate gradient approach used by Bos (1d). Another advantage of this approach discussed by Bos (1e) is that it does not need any user setting of parameters such as learning rate and momentum. In the standard back-propagation algorithm however, these two parameters need to be set by the user. The values of these parameters help to determine how fast the optimisation algorithm finds an optimum (however it would also be dependent on the weight space and as such on the topology of the network and the initialisation of the weights). The meaning of these parameters have already been discussed in terms of the back-propagation algorithm in the introduction, however little attention has been given to the practicalities of the choice of their values. There aren't any fixed rules or fixed values for these parameters which will work for every problem. Instead, it is up to the user to try different values for themselves keeping a couple of things in mind such as

* the behaviour observed may be due to different initialisations of the weights

* the effect of the momentum is likely to be interdependent on the effect of the learning rate. This can be seen because the momentum adds a fraction of the previous weight change to the present weight change, but the previous weight change was dependent on the learning rate and momentum.

* the values of the learning rate and momentum should also be considered in terms of the transfer function or basis function between the different layers. A sigmoid transfer function bounds the output of a neuron between 0 and 1. A linear transfer function does not have this bounding. Smits et al. (9) recommend that the learning rate in this case should be kept below 0.1 to prevent divergence or oscillation

The investigations described in this thesis solely used the sigmoid transfer function. Based on studies in this work (see table 2.3) it was found that learning rates of 0.25-0.75 and momentum values of 0.5 to 0.8 were preferable for the particular application. It should be emphasised again that these values were specific for the application in question and should not be considered as being generally applicable to every application. Given the considerations discussed above, the approach of experimenting with different values for the learning rate and momentum, should probably involve looking for gross rather than specific features. A simple way of observing the training behaviour of a network involves plotting the MSE on the training set as a function of the number of iterations of the algorithm. Ideally, one might expect to see a very rapid initial decrease in the MSE on the training set followed by a much slower decrease in the MSE in the later stages of training (see figure 5.2).

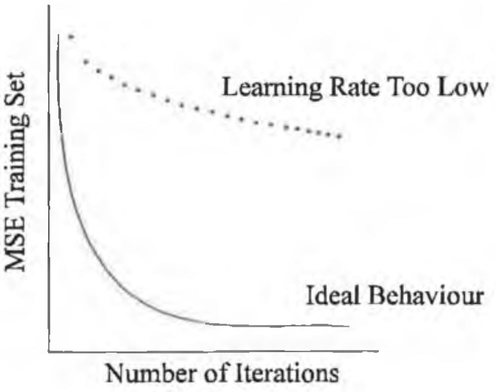


Figure 5.2-Simplified example of ideal training behaviour of a network and training behaviour when the learning rate is too low. This is depicted by changes in the MSE of the training set with the number of iterations of a backpropagation algorithm

If the learning rate was too small for the particular application it would show up as a very slow decrease in the MSE with the number of iterations of the back-propagation algorithm (as can be seen in figure 5.2).

Similarly if the learning rate were too high for the particular application it might be expected to see severe oscillation on the MSE versus iteration number plot (see figure 5.3).

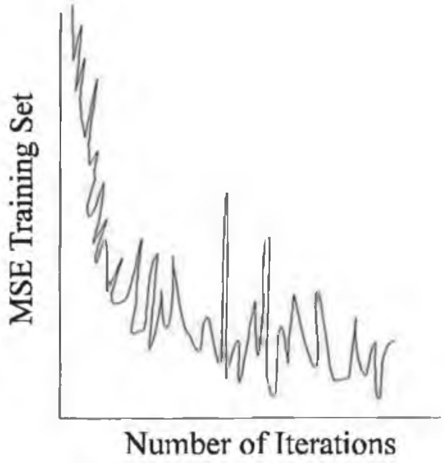


Figure 5.3- Simplified example of training behaviour of a network with a high learning rate as seen by oscillation on the variation of the MSE of the training set with the number of iterations of a backpropagation algorithm

Between these two extremes, optimising the learning rate and momentum is likely to be very difficult.

The decision of when to terminate training can be based on a number of criteria including termination after a fixed number of iterations, or after a predefined MSE has been achieved. Both these criteria are likely to be very dependent on the problem itself and as such rather subjective. Bos (1f) has discussed a termination criteria based on studying the rate of change of the MSE during the later stages of training (when the MSE is changing more slowly than the initial stages of training). If the rate of change of MSE does not increase with further iterations of the training algorithm, then it is assumed that the network is approaching a minimum and that the network training can be terminated with little risk.

Before training it is also worth considering how to evaluate the performance of the network. It is unlikely that the behaviour of networks can be deterministically proven and/or predicted on anything other than toy problems. The best that could be achieved would be to test the performance of the network as rigorously as possible and then to quote the performance of the network under the conditions of the testing. There are two different ways of looking at the performance of a network namely performance in terms of the training set and performance of the network on the testing set. It is desired of a network that it be able to learn the representation of the mapping sampled by the training set and be able to generalise to the testing set. In this context it is worth discussing the problem of overtraining. This can be observed by an examination of a plot of the MSE on the training set and the MSE on the testing set during training. In cases of overtraining one might expect to see the MSE of the training and testing sets to initially both decrease, at some point however, the MSE on the testing set will start to increase while the MSE of the training set will continue to decrease (see figure 5.4)

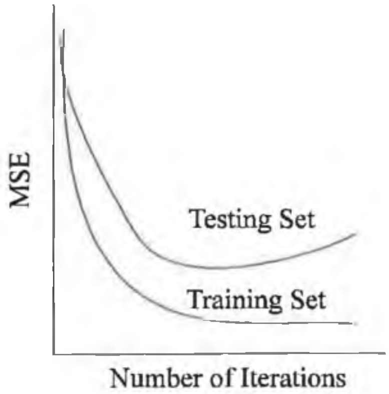


Figure 5.4 - Graphic depiction of the behaviour of a network demonstrating overtraining, in the later stages of training the MSE of the testing set increases as the MSE on the training set continues to decrease

Definitive reasons for how or why overfitting happens are generally not readily accessible to those of limited mathematical understanding, as the theory behind generalisation tends to be mathematically intense. Instead it is likely that the initial decrease in the MSE of the training and testing sets is due to the formation of a crude model which is appropriate for both sets. After a while however, this coarse model takes on specific attributes of the training (e.g. noise) that doesn't exist in the same form in the test set. Apart from the causes of overfitting, the question of how to alleviate it is also controversial. One approach taken is to stop training if an increase is seen in the MSE of the testing set while there is a decrease in the MSE of the training set. A problem with this approach is that it implicitly establishes the testing set into the training regime. This problem can be dealt with by using a cross-validation set to cross-check the performance of a network trained in this fashion.

Chapter 2 of this thesis dealt with the use of ANNs for pattern classification purposes. In this case the final result of the network is related to the network output by means of a threshold. If the output of the threshold is above a certain threshold, then the particular feature or species is deemed as being present, if the output of the network is below a certain threshold the feature or species is deemed absent. The region between these two thresholds is a region of uncertainty. The larger the difference between the two thresholds, the tighter are the classification criteria. For example, let the possible output range of a neuron be 0 to 1, in this case an output of 0 indicates that the species/feature is absent and an output of 1 indicates that the species/feature is present. If the lower threshold is set to a value of 0.1 and the upper threshold to 0.9, then the patterns can only be classified definitively as being present or absent within an output range of 0.1 for each conclusion. Similarly if the lower threshold is 0.2 and the upper threshold is 0.8, then the patterns can be definitively classified within the broader output range of 0.2 for each conclusion. As such, given a set of patterns and network outputs the thresholds themselves can be altered, to automatically optimise the number of correct classifications.

Having discussed ANNs in some length, it is time to focus attention on the genetic algorithm (GA). The issue of clearly identifying a problem and the relevant input and output data has already been addressed, as has the issue of representing the problem on the chromosome. The next thing to consider is how large a population to use. Studies described in this thesis mainly used populations of between 100 and 500 chromosomes. With smaller populations, particularly fit[1] individuals can rapidly dominate the population. For instance if an individual can produce 2 offspring in the next iteration of a GA, then within 7 iterations it will fill a population of 100 chromosomes. This is of course a gross simplification, because it doesn't take into account the disruptive effects of crossover and mutation, but it gives a feel for how quickly an individual can fill a population.

The reasons for not using a population of more than 500 chromosomes relate to simple practicalities, the computational time involved in evaluating each chromosome can be prohibitive.

Having settled on a particular problem representation, one of the biggest problems in the implementation of GAs involves maintaining diversity within a population without disrupting potentially useful solutions. In the investigations described in this the selection and crossover operators were the most deeply studied. There are a broad range of genetic operators which have not investigated in this thesis, Lucasius thesis (10) gives a good overview of these operators and their use.

[1] The fitness of an individual refers to how well a chromosome performs the required task of the particular application. In cases of calibration the fitness of the individual refers to how well the model encoded by the chromosome fits the experimetal data and is quantified by the reciprocal of the SSE of the model

Of the approaches used in this thesis, scaling of one form or another was found to be essential to reduce the chances of particular individuals dominating the population. Linear prescaling limits the number of offspring an individual can produce in the next iteration of a GA. As such, it limits the effects of individuals which represent solutions tens or hundreds of times better than the rest of the population (a "super-fit" individual). For instance, without prescaling, an individual who was 100 times more fit than any other individual in the population, would be 100 times more likely to produce offspring in the next iteration of the GA, and would rapidly fill a population of 100 chromosomes. With linear prescaling however, the number of offspring it could produce relative to the average fitness of the population would be limited. While linear prescaling removes the problem of a super-fit individual (as would be seen in an absolute sense without prescaling), it preserves the relative differences in the fitnesses of the chromosomes which gave rise to the super-fit individual.

With this in mind, rank prescaling was found to be particularly effective, because it scales individuals with respect to their rank (when sorted according to their fitnesses), and as such doesn't consider the relative differences in the fitnesses. The approach taken to rank prescaling did not exactly follow the form described by Baker (11). Instead, a modified form of rank prescaling was used in which the most fit individual in the population (rank 1) would have its fitness scaled according to a user-defined upper-limit and the individual with middle ranking (rank =P/2 where P= the number of chromosomes in the population), would have its fitness scaled to 1.0. The reasoning behind this is the same as that used for linear fitness prescaling but in this case the ranks are scaled. Baker found that an upper limit of 1.1 was necessary to prevent premature convergence. In initial studies with the modified rank prescaling approach, it was found that there was too much disruption happening with upper limits of this magnitude (i.e. potentially useful solutions to a problem were being lost during the operation of the GA). Instead, after a degree of experimenting manually with different upper limits, it was found that upper limits of between 1.3 and 1.5 provided a better balance between reducing premature convergence and reducing disruption. The selection scheme used in these studies was stochastic remainder sampling without replacement, in this case it is guaranteed that an individual with a fitness scaled to 1.3 or 1.5 will produce one offspring in the next iteration of the GA, with a 0.3 or 0.5 chance of it producing another offspring. Using an upper limit of 1.3 to 1.5 on the rank prescaling, is a cautious attempt to reduce the certainty of an individual doubling itself every iteration of the GA, whilst providing some degree of incentive for propagating more fit individuals in the population.

Of the selection schemes available, roulette wheel selection and stochastic remainder selection were used in the studies described in this thesis. Stochastic remainder selection was found to be preferable because if the most fit individual in the population has its fitness prescaled to a value greater than or equal to 1.0, then it is guaranteed that the individual will reappear at least once in the next iteration of the GA. This is something which cannot be said of roulette wheel selection. While it is probablistically more likely that the most fit individual will reappear in the population in the next iteration of the GA, there is no deterministic way of saying it will happen. As such I feel that cautious intervention by the user is justified.

In the studies described in this thesis, the single point crossover operator was used when performing crossover. There are a wide variety of other operators as discussed by Lucasius (10) and certainly uniform crossover looks particularly useful in alleviating the problems experienced with highly correlated parameters. An example of this problem can be seen in the calorimetric based determination of stability constants in section 4.3.2.

Depending on the type of problem involved, the user may need to consider the scheme by which individuals are selected as mates for crossing. In the studies described in this thesis, mates were not chosen from the population for two reasons. The first reason was to reduce the chances of crossover producing chromosomes identical to the original mates (unproductive crossover). The second reason was to prevent the production of chromosomes which were inappropriate for the chemical requirements of the application. Both of these considerations were handled by means of rejection loops. In the implementations of rejection loops used in this thesis, a chromosome is randomly chosen from the population to act as the first mate for the crossover, a second chromosome which is also chosen randomly from the population, is rejected as a prospective second mate if it fails to fulfil certain conditions. Chromosomes are repeatedly chosen from the population until one of them fulfils the conditions, at which point, it is used as the second mate in the crossover with the chromosome chosen as the first mate. The process of random selection and rejection of chromosomes is continued until a fixed number of chromosomes have been crossed, or until a fixed number of chromosomes have been successively rejected as second mates.

There are a number of points to consider in this rejection loop. The first consideration refers to the requirements which have to be satisfied before a chromosome is used as a second mate in crossing. These requirements check for similarities between the prospective mates and also checks the history of mates which have already been selected for crossover (in the same iteration of the GA). The rationale behind these rules is to attempt to make crossover as productive as possible, by ensuring that the same individuals are not repetitively chosen for crossover and also to ensure that those chromosomes which are chosen, will produce different chromosomes when crossed. For the study in chapter 3, chromosomes which were identical were rejected as mates. In chapter 4, a Hamming distance threshold was used to reject mates. Since chapter 4 dealt with variable length chromosomes, the threshold was set to be dependent on the length of the chromosomes, such that chromosomes which differed in less than 10% of their bit positions were rejected. This 10% was chosen as an arbitrary value, there is a lot of scope for optimising it or for dynamically changing it during the operation of the GA. The second requirement of the rejection loop was specific to the study described in chapter 4. Since variable length chromosomes were being used, it was possible to cross chromosomes which encoded different numbers of parameters and different chemical equilibria. This would be valid in the sense of the GA, but would have no chemical meaning. In this sense, it is necessary to identify the chemical features and requirements of the problem of interest and to adapt the genetic operators according to those needs. To solve this problem, chromosomes were rejected as crossover mates if they did not encode the same number of parameters. This rejection rule could also be more refined.

The next issue refers to how many cycles of the rejection loop to use before terminating the selection of mates for crossover. In the studies described in chapter 4, this was chosen to be 100 times the required number of mates. This ensures that the number of attempts to find a second mate are dependent on the number of required mates. This is based on the assumption, that the larger the number of chromosomes required to mate, the more difficult it is going to be to find enough mates which satisfy the conditions of the rejection loop. As such, there should be more chances to select the second mate to fulfil the conditions.

The third issue refers to how many chromosomes should be allowed to cross. The work described in this thesis did not allow the entire population to cross. The probability of crossing was also not directly determined from the fitness of the individual. Instead, during the selection stage of the GA, the previous population was completely replaced with the population determined by the selection strategy. As such, the probability of crossing according to fitness, was implicitly established by the distribution of the individuals in the newly replaced population. In this case a percentage of the population called the crosspool was chosen for crossing. The crosspool was filled with chromosomes chosen at random from the entire population, according to the conditions specified by the rejection loop. The original logic for this approach was to ensure that the most fit individuals in the population were not disrupted by crossover. This was achieved by ensuring that only the least fit chromosomes (in the population) could be overwritten by the crossed chromosomes (in the crosspool). However a balance had to be achieved, because limiting the percentage of the population which could be used for crossover also limits the scope of the crossover operator to explore new regions of the search space. By following the history of the population through several iterations it was found that using crosspools of 80% of a population of 25 chromosomes caused problems because of non-productive crossover (before rejection rules had been developed). As the GA developed in complexity (with rejection loops) and larger populations were used this was less problematic.

In the backpropagation algorithm the learning rate and momentum were parameters whose values had to be set by the user. In GAs the upper limit for fitness prescaling (if prescaling is being used), the crossrate and the mutation rate are parameters whose values also need to be set by the user. Fitness prescaling and the reasoning behind the choice of a particular value for the upper limit have already been discussed. The crossrate describes the probability that crossing will occur in a given iteration of the GA.

De Jong's studies described by Goldberg (12) suggested a crossrate of 0.6. In the studies described in this thesis, a value of 0.7 tended to be used, as a medium between guaranteeing that crossover will happen (crossrate = 1.0) and having an equal probability of crossover happening/not happening (crossrate = 0.5). The mutation rate is used to determine whether a chromosome in the population will undergo mutation during a given iteration of the GA. In the studies described in this thesis, the mutation rate was set to be the reciprocal of the population size. The rationale behind this approach comes from the idea that if the population is small, then it is easier for an individual which is fitter than the rest of the population to fill the population within a few iterations of the GA. As such, the random effect of mutation could be used to alleviate this problem, with larger sized populations it is more difficult for an individual to completely fill the population, while at the same time it is desirable to minimise the disruptive effect of the mutation operator.

This discussion has hopefully given some general idea of some of the features to look out for, when using GAs and ANNs. Table 5.2 summarises the ideas I have discussed in this document. There are many other techniques and features specific to those techniques which I haven't discussed for greater details of these techniques the thesis by Bos (1) and Lucasius (10) are recommended.

There is a great deal of scope for future research into the applications of ANNs and GAs. In the course of this discussion, some techniques which could be particularly interesting have been discussed. The GA and ANN approaches described in this thesis have not been optimised for their particular tasks and there is a lot of room for future research in their optimisation. In a broader sense, there still does not exist a standard methodology for designing and using GAs and ANNs, this area still requires a degree of skill and experience on the part of the user.

ANNs	GAs
* Identify the chemical problem and the nature of the input and output data	* Identify the chemical problem and the nature of the input and output data
* Consider the representation of the input and output data * Dimensionally Reduction * Scaling	* Consider the representation of the input and output data * Scaling the search range for a variable * Number of bits used to represent the variable
* Choosing the training and testing sets * Cross-validation set	* Choosing the data on which the GA optimises
* Topology of the network * Number of hidden layers * Number of neurons in the hidden layer * Pruning and growing methods	* Size of the population
* Training algorithm * Standard back-propagation * Scaled conjugate gradient	* Fitness/rank prescaling * Selection scheme * Rejection loop for crossover
* Learning rate/momentum	* Crossrate and mutation rate
* Overtraining	
* Validation	

Table 5.2 Summary of the subjects dealt with in this discussion document

An area of future research which will be of particular interest, will lie in the areas of automated development of these techniques. To do this however, a more clear definition of the features and problems associated with the use of GAs and ANNs will need to be developed, as will a clearer understanding of the contributing factors to these features.

Bibliography

1. A. Bos, **Artificial Neural Networks as a Tool in Chemometrics**, Ph. D. Thesis, University of Twente, Enschede, 1993, (a) p.25, (b) p.128-131, (c) p.142, (d) p.19-21, (e) p.103, (f) p.143-144,
2. Anne-Johan Annema, **Analysis, Modelling and Implementation of Analog Integrated Neural Networks**, Ph. D. Thesis, University of Twente, Enschede, The Netherlands, 1994
3. R.P. Lippmann, **IEEE ASSP Magazine**, 1987 April, 4-22
4. R. Hecht-Nielsen, **Theory of the backpropagation neural network** chapter III 3, p.65-93 in **Neural Networks for Perception Volume 2, Computation, Learning and Architectures**, H. Wechsler (Editors), Academic Press Inc 1991
5. F. Girosi and T. Poggio, **Neural Computation**, 1, 1989, 465-469
6. E.L. Hines, C.C. Gianna and J.W. Gardner, Chapter 9, **Neural Network Based Electronic Nose Using Constructive Algorithms**, p.135-154, **Techniques and Application of Neural Networks** (Editors) M. Taylor and P. Lisboa, Ellis Horwood, U.K. 1993
7. R. Reed, **IEEE Transactions on Neural Networks**, 4, 1993, 740-747
8. P. P. Van Der Smagt, **Neural Networks**, 7, 1994, 1-11
9. J.R.M. Smits, W.J. Melssen, L.M.C. Buydens and G. Kateman, **Chemometrics and Intelligent Laboratory Systems**, 22, 1994, 165-189
10. C.B. Lucasius, **Towards Genetic Algorithm Methodology in Chemometrics**, Ph. D. Thesis, Katholieke Universiteit Nijmegen, The Netherlands, 1993.
11. J.E. Baker, **Adaptive Selection Methods for Genetic Algorithms**, p. 101-111 in the **First International Conference on Genetic Algorithms**, (Editor) J.J. Grefenstette, Lawrence Erlbaum Associates, Hillsdale N.J., 1985
12. D.E. Goldberg, **Genetic Algorithms in Search, Optimisation and Machine Learning**, Wesley Publishing Company Inc. USA, 1989