To the Graduate Council:

I am submitting herewith a thesis written by Nithyananda G. Narayana entitled "Neural networks for parameter identification in compartment models". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Laurence F. Miller

Laurence F. Miller, Major Professor

We have read this thesis and
recommend its acceptance

Peter J. Jones

Accepted for the Council

Associate Vice Chancellor and
Dean of the Graduate School

NEURAL NETWORKS FOR PARAMETER IDENTIFICATION IN COMPARTMENT MODELS

A

Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Nithyananda G. Narayana

August, 1995

DEDICATION

This Thesis is dedicated to my parents

Mr. G. N. Narayana and

Mrs. Mukundamma Narayana

who are a constant source of inspiration in my life.

Acknowledgements

I express my deepest and sincere thanks to my Major Professor Dr. Laurence F. Miller for being my Advisor. I am deeply indebted to him for his guidance, encouragement and patience while preparing this thesis. I also thank him for his confidence in me.

I would like to thank Mr. Roy P. Osborne, RSO, Radiological safety Department for providing financial assistance, advice and hands on experience in the Radiation Safety procedures.

I would also thank Dr. Tom Kerlin, Head of the Department for encouraging me do my higher studies in Nuclear Engineering at The University of Tennessee. I also thank Dr. Peter Groer and Dr. Perez for being on my committee.

I also thank my friend Mr. Masod Naghadolfeizi and my roommate Subu for their valuable friendship and advise.

I am deeply indebted to my parents, brother, sister and uncle (Narayanaswamy Krishnappa) whose support, patience and untiring love are matchless.

ABSTRACT

In health physics and nuclear medicine, numerous situations arise in which it is required to model the transport of radionuclides in biological systems. Compartment models and Patlak graphical analysis methods have been employed for quantification of these parameters. This thesis evaluates the application of artificial neural network models to identify model parameters in such applications.

Neural network models were developed for predicting various model parameters. A trial and error procedure was employed to obtain a suitable number of hidden nodes and hidden layers to achieve appropriate network structure for health physics and nuclear medicine applications. Computer algorithms were written to generate training and testing data for these network models. After the networks were trained, their performance was evaluated on unseen patterns (not used in training) of synthetic data by comparison of results from the artificial neural network models with those obtained from traditional methods.

The neural network models trained with synthetic data were also tested with experimental data obtained from Positron Emission Tomography (PET) scans. In addition, several neural network models were developed and trained using different mathematical features of the synthetic data. Then, the

performance of these network models were evaluated by testing with unseen patterns (not used in training) of the synthetic data.

The results show that the network models predicted the model parameters with more than 95% accuracy. Myocardial blood flow (MBF) estimated by neural network models correlated well with MBF obtained by Patlak method (correlation (r) = 0.97, slope = 1.0055). MBF obtained by neural network models agreed well with MBF obtained by those specified by the compartment models (r = 0.98, slope = 1.01). Good agreement of the MBF estimates from neural network models were also observed for data containing small amounts of noise (r = 0.97, slope 1.0092).

Results from these studies demonstrate that neural network models are reliable and suitable tools for identification and estimation of various parameters in compartmental models. In addition, the neural network models can capture the nonlinear characteristics of data more efficiently and in less complicated manner than traditional techniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Statement of the problem

In health physics and nuclear medicine, numerous situations arise in which it is required to model the transport of radionuclides in biological systems. The internal exposure and movement of radionuclides is monitored based on the measurement of radionuclides in the whole body, specific organs and excreta. Biokinetic models that describe the behavior of material in the body or in a specific organ are used to establish a relationship between the levels in the body or organs, and they are used to estimate intake and the resulting dose.

In nuclear medicine, imaging techniques are used to determine the movement of activity in the organ, to identify the kinetics of metabolic processes, and to verify transport properties in the myocardium. This data also permits one to calculate activity as a function of time at a given location.

In this study, neural network models are developed for predicting model parameters like transfer rate constants, initial amounts of radioactive materials, and other important diagnostic indicators. Although computer algorithms for solving compartment models and parameter identification are valid, they are often less robust relative to noise and complexity than neural

networks. Neural network models can be developed to identify important model parameters for which nonlinear minimization algorithms are computationally inefficient.

Backpropagation neural networks are utilized to estimate the translocation rates between compartments and to determine initial amounts of radioactive material present in a specified compartment. A neural network model is also employed to calculate regional myocardial blood flow in nuclear medicine applications. The neural network models are developed and trained on synthetic data obtained from the compartment model, and they are used to predict model parameters for Positron Emission Tomography data.

1.2 Overview of the methodology

Neural networks are studied for parameter identification with training data obtained from two, three and four-compartment models, and a computer program for solving general compartmental models was written to provide training and testing information. Simulations were run for times ranging from several minutes to about thirty years. In some nuclear medicine applications, effective half-life of radionuclides in the body may be few minutes; however, in some health physics applications, radioactivity may be in the body for many years.

Parametric values that determine model response were selected randomly over specific ranges. Results from these simulations provide the training and

testing information for the neural network models. In this study, The Neuralware Software package [1] was used to develop the neural network models.

Several neural network models were developed and trained using different mathematical features of the data. Synthetic data for estimating the regional myocardial blood flow(RMBF) in two-compartment model was generated from the algorithm[2] shown in Appendix B. The networks were also tested for perturbation and sensitivity by introducing noise in the system, and then the network outputs were observed for possible changes in comparison with the desired outputs.

1.3 Organization of the thesis

Chapter 2 describes the organization of artificial neural networks (ANN), advantages over other data processing techniques, and it presents a historical perspective. It also describes the topology of backpropagation neural networks.

Chapter 3 describes the algorithm developed for solving compartmental models for internal dosimetry applications. Evaluation of the exponential of a squared matrix is also presented in this chapter.

Chapter 4 focuses on the development of a two-compartment model for a typical human heart. It also describes the method of estimating the regional

myocardial blood flow (RMBF) using the compartment model, and it presents an analytical solution to the model for generating synthetic data for training the artificial neural networks (ANN).

Chapter 5 is devoted to the development of neural network models for parameter identification in compartment models employed in internal dosimetry and nuclear medicine applications. It discusses the training procedures and results obtained from the neural network models. It also compares the results obtained from the neural network models with those obtained from other methods.

Conclusions and recommendations for future work are outlined in Chapter 6.

CHAPTER 2

NEURAL NETWORK TOPOLOGY

2.1 Introduction

The structure of an artificial neural network (ANN) is inspired by the network of the brain and nervous system. Artificial neural networks (ANN) are computational systems formed by a number of nonlinear information processing units that are highly interconnected and operate in parallel. They do not require detailed programming to break a problem down into a form suitable for parallel processing. Instead, neural networks are trained by exposing them to specified data sets of information, which enables its self-organizing and learning capabilities to produce the desired goal. Artificial neural networks generally encompass a wide variety of applications. Notable examples are pattern and speech recognition, data compression and signal processing. ANN's provide a suitable alternative to computing methods which may not perform very well for complex nonlinear systems.

Neural networks differ from other data processing techniques with a high connectivity between units and the distributed manner in which computation is achieved. Each unit, known as a processing element (PE), is analogous to biological neurons, capable of performing elementary mathematical operations and nonlinear transformations. The PE's operate in a massive

parallel distributed processing architecture. These processing elements are arranged in layers, and they are connected by weights to resemble the structure of neurons in the cerebral cortex portion of the brain, which is illustrated in Figure 2.1[3].

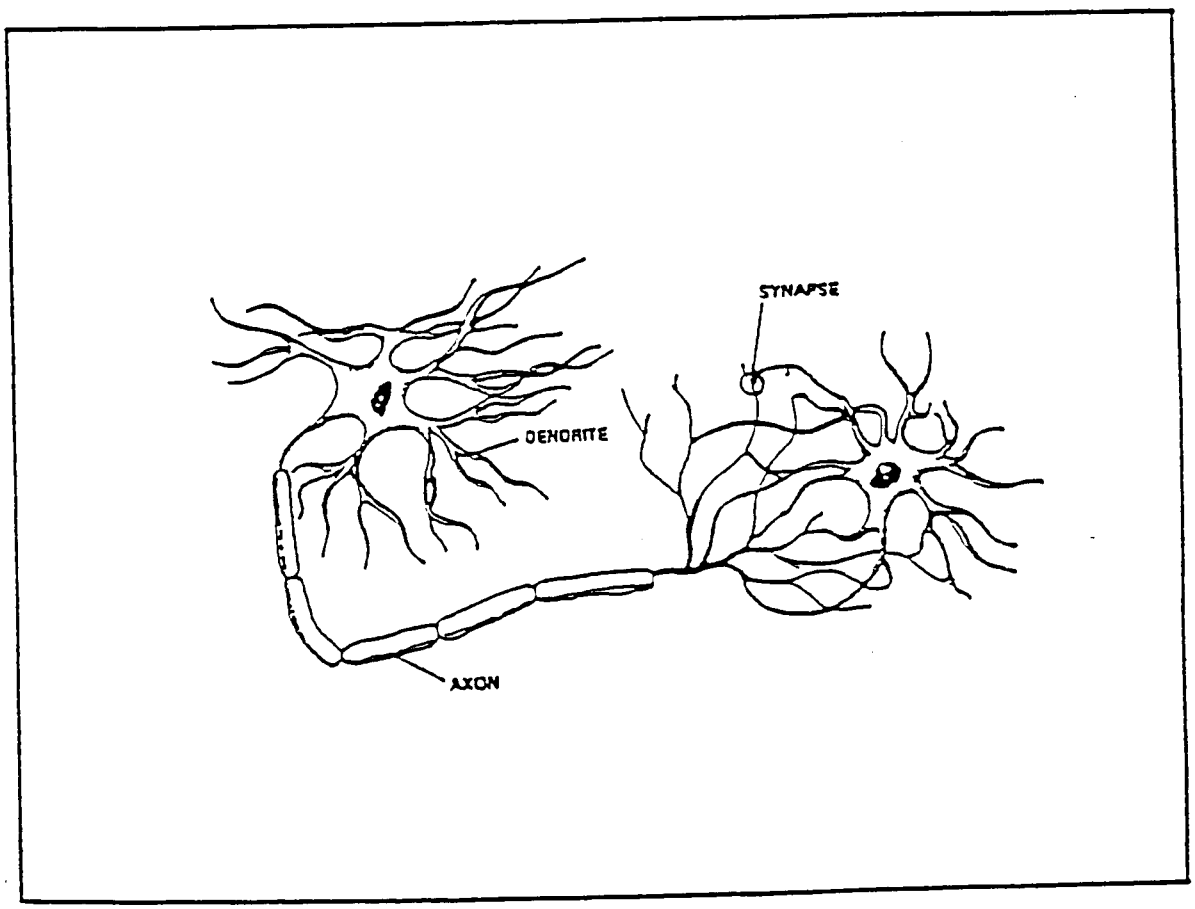


Figure 2.1 Biological Neuron[3].

Neural network models attempt to mimic or emulate the very basic functions of biological nervous systems. In designing a neural network, there are many features of the neurode that must be selected. The connection strengths are simulated using a number called weight. A negative weight has an inhibitory effect and positive weight has a excitatory effect on the information flowing between weights. This design has the ability to process information in parallel since information flows continuously and simultaneously. One notable advantage of neural networks is the speed of computation. Another advantage is the robustness. Since there are many nodes with only local connections and since the information is spread over a large section of the system, damage to a few nodes need not impair overall performance of the system.

Perhaps the most important characteristic of neural networks is their ability to model processes and systems from actual data. The neural network is supplied with data and then "trained" to mimic the input-output relationships of the process or system. This is called the learning phase. During this phase, the connection weights associated with each processing elements (PE) are adjusted by a learning algorithm. The training time depends on the complexity of the application, the network model and the training algorithm. The next phase is the recall phase, where performance of the network is examined and evaluated by presenting inputs of unseen patterns (not used in the training) to the network. Several trials on both the network structure and the amount of training time may be required to obtain a suitable network for a given application.

Neural networks may also be used to interpolate and extrapolate effectively from learned data, and more importantly, to model processes from actual process data.

2.2 Historical perspective

The history of neural networks began forty eight years ago with the work of a neurobiologist, Warren Mcculloch and a statistician, Walter Pitts, which was followed by Frank Rosenblatt, Minsky, Hebb, Widrow and others in the 1950's and 1960's[1]. In 1957, Rosenblatt's perceptron sparked a great amount of research interest in neural computing. However, research in neurocomputing came virtually to a halt in 1969 due to a paper by Minsky and Papert that proved perceptrons cannot classify inputs that are visually nonlocal. The funding agencies shifted their investment to other artificial intelligence areas, which started to a achieve significant successes.

Other researchers like Kohonen, Grossberg, Rumelhart, Specht, Hecht-Nielsen and Hopfield continued their work in neural computing despite Minsky and Paperts book[1]. With the developement of new theories and algorithms and with the availability of faster and parallel hardware, neural computing was reinvigorated. In 1982, with John Hopfield's enthusiasm and clarity of the first paper presentation since the 1960's prompted researchers to become interested again in the fascinating field of neural computing. Also, there was a realization that the solution of certain problems requires a significant amount of computational power, and the recognition that

traditional sequential computing, however fast, was inadequate for the solution of some problems.

2.3 Backpropagation neural network

Backpropagation is a systematic method for training multiple layer artificial neural networks. A network is presented with a set of input vectors together with the corresponding desired output patterns. At each step, the algorithm compares the actual output with the desired output and calculates the error. The error is then used to update the weights in the output layer in the desired direction. The errors are then backpropagated to the hidden layers to update the remaining weights, hence the name backpropagation. The training method quantifies the contribution from individual units to the global error, and it adjusts weights between connected processing elements so that the difference between the actual output and the desired output is minimized in a least-squares sense.

Backpropagation networks operate either in autoassociative or heteroassociative mode depending upon the objective of the network. For an autoassociative network, the input vector and the output vector are identical. A heteroassociative network learns associations between different input and output vector pairs. The basic backpropagation neural network (BPN) algorithm is presented by Rumelhart and McClelland [3] and is discussed here only briefly.

A typical backpropagation network has an input layer, an output layer and atleast one hidden layer. The input layer acts as a buffer or fanout for the inputs to the system. It receives information (input vectors) and transmits the information to the hidden layers through weighted connections. The hidden layers are used by the network to develop its own internal representations of the input patterns which represent the nonlinear characteristics of the data. Backpropagation trains the hidden layers by propagating the adjusted error back through the network, layer by layer, adjusting the weight of each layer as it goes. The topology of the network is as shown in Figure 2.2.

The input layer receives the input to the system and passes it to the first hidden layer. The first hidden layer sums the weighted information from the input layer, modifies it using a logistic function (discussed later), and propagates the modified information forward to another hidden layer or to the output layer. The processing at each hidden layer is similar, and the information eventually reaches the output layer where it is compared with the vector of desired outputs.

In the output layer, the overall error is calculated. The error is the difference between the ideal (desired) output and the actual output generated by the network. This error is propagated back through the layers, and the weights are adjusted at each layer. This training process is repeated for each associated input/output pair until the error is reduced to a predefined level, or for a specific number of iterations, according to the criteria for termination.

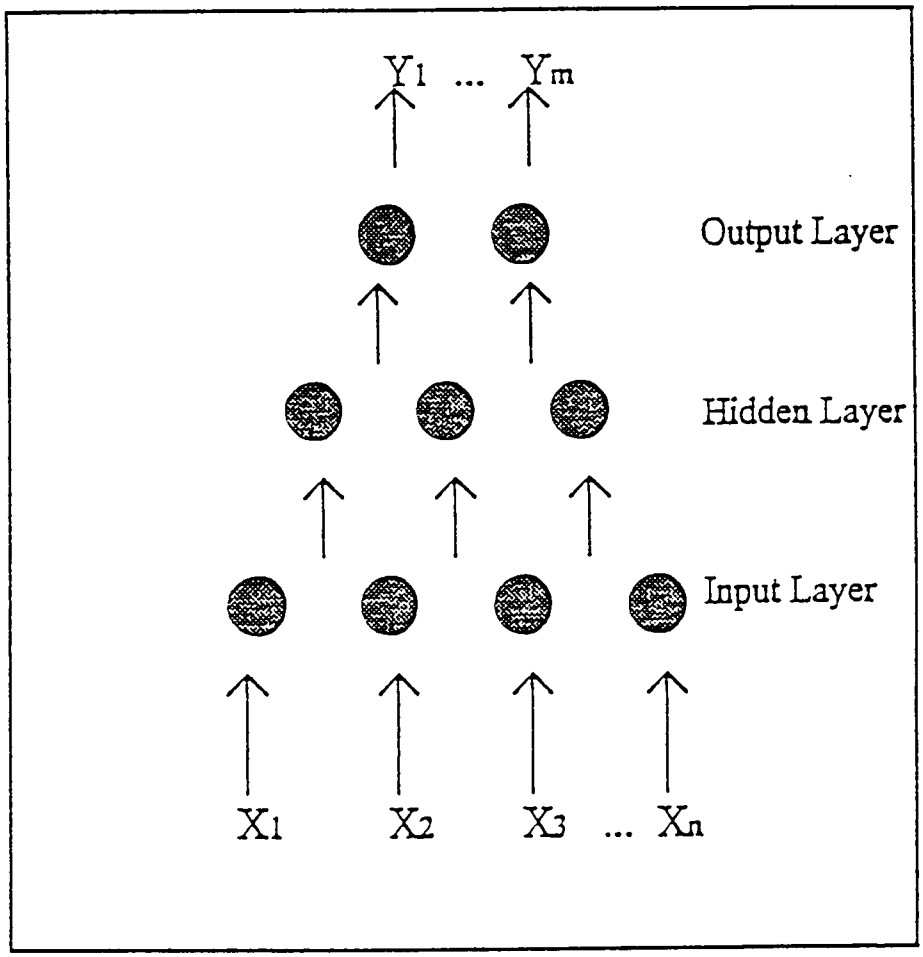


Figure 2.2 Topology of Backpropagation network

During training, the information flows forward from the input layer to the output layer, and the error is propagated backwards through the weights to calculate the weight adjustments. Training involves the modification of the weights until the error is minimized.

The logistic function acts as an automatic scaling factor and gain control. Another name given to this function is transfer function, and it is referred to by that name widely used in the literature. The function provides small gain for large signals to prevent saturation and high gain for small signals to prevent attenuation. The logistic function used most commonly is the sigmoid defined by,

$$F(x) = \frac{1}{1+e^{-\alpha x}}, \qquad \alpha > 0 \qquad 1.1$$

where x is the input to the function, and α is the shaping parameter. In theory, the algorithm only requires that the function be bounded and everywhere differentiable. Values of F(x) range from zero for large negative values of x to one for large positive values of x. Other logistic functions are the hyperbolic tangent and the arc tangent where the values vary between -1 and 1 as x changes from a large negative value to a large positive value.

In the fully connected backpropagation model, every processing element in a layer is connected to all neurons in the next higher layer with no lateral connections. A BPN processing element calculates the scalar product of each input vector and its corresponding connection weights, it then filters this product using the logistic function of Equation 1.1 as shown in Figure 2.3.

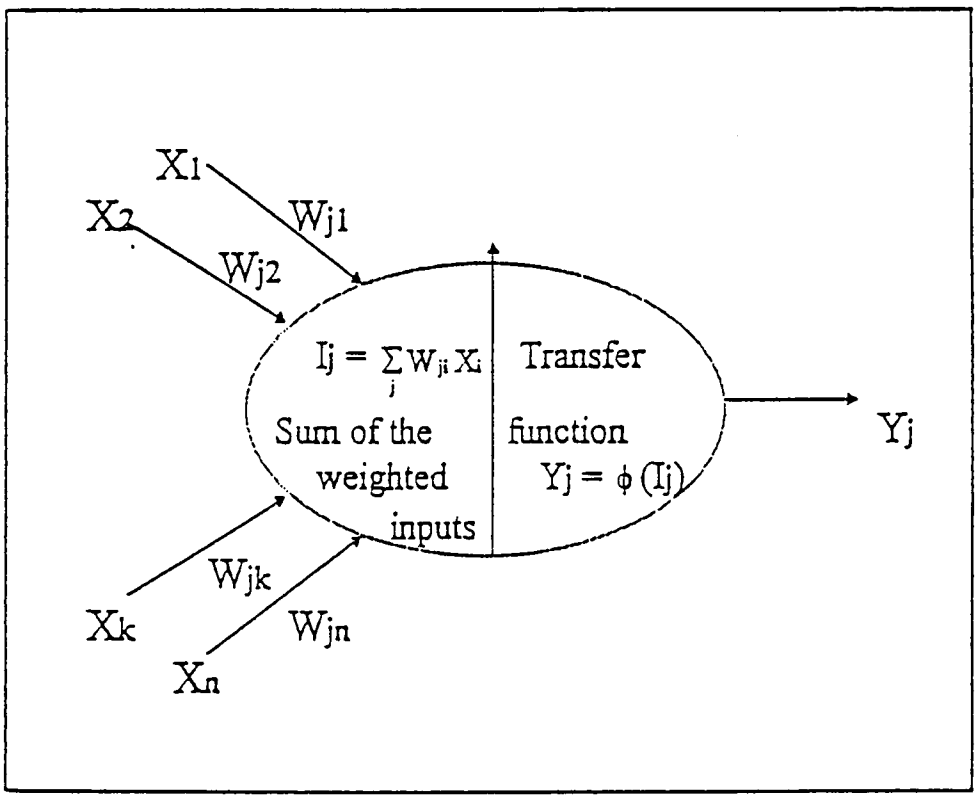


Figure 2.3 Artificial Neuron

CHAPTER 3

COMPARTMENT MODEL

3.1 Introduction

Mathematical models are used to describe the physiological and biochemical information of a process in terms of mathematical relationships for various process variables. Once the modeling objective has been defined, a compartmental representation of the system can be developed. The compartments are denoted by labeled boxes linked together by arrows which indicate the translocation pathways between compartments. A compartment is defined as a space in which the tracer is distributed uniformly. The tracer kinetics are usually described by a set of linear, first order, ordinary differential equations. This is due to the fact that in many of the natural processes, the tracer movement can be characterized by a single exponential function of time which is sufficiently simple and accurate for many applications.

3.2 Methodology

In solving any compartment model, the first step is to number all the compartments from 1 to n. This is schematically shown in Figure 3.1 for a typical compartment model. The model to be solved can then be represented

entirely by a n x n rate matrix (if it is linear). Each element R(x, y) of the matrix contains the rate constants for transfer from the xth to the yth compartment, while reserving the diagonal elements R(x, x) for the initial amounts in each compartment x. Figure 3.2 illustrates the rate matrix corresponding to the compartment model in Figure 3.1.

The rate matrix contains a complete description of the model and its initial conditions. Hence, an algorithm can be written which acts directly to give the amount of activity for any compartment as a function of time for a given set of model parameters.

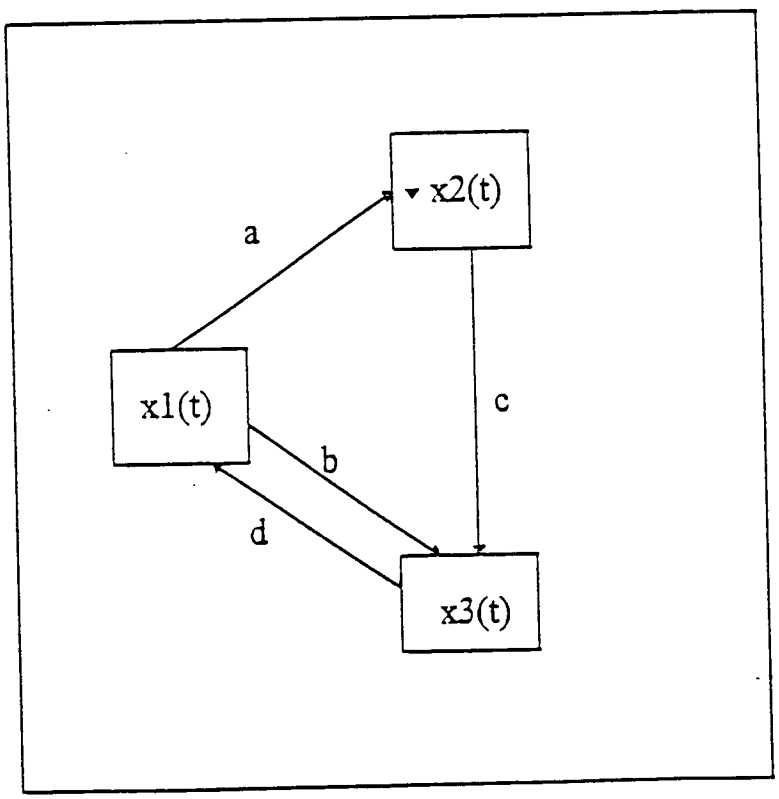


Figure 3.1 A compartment model showing the amount of activity in each of compartment, and the transfer rate constants a, b, c, d between compartments.

x1(0)	a	b
	x2(0)	c
d		x3(0)

Figure 3.2 Rate matrix [R] corresponding to the compartment model in Figure 3.1.

-(a+b)		d
a	-c	
b	c	-d

Figure 3.3 Matrix [A] obtained from rate matrix [R] of Figure 3.2 corresponding to the compartment model in Figure 3.1.

3.3 The algorithm

3.3.1 Solution to the model

The algorithm operates by first transforming the rate matrix [R] into a new matrix [A] by replacing each diagonal element of [R] with the negative of the sum of each of the row elements and then by taking its transpose[4]. This helps one to solve the general system of differential equations describing the model by using the matrix [A] directly. Consider the matrix [A] and the rate matrix [R] having values a_{xy} and r_{xy} respectively, then

$$a_{xy} = r_{yx} \text{ for } x \neq y,$$

and

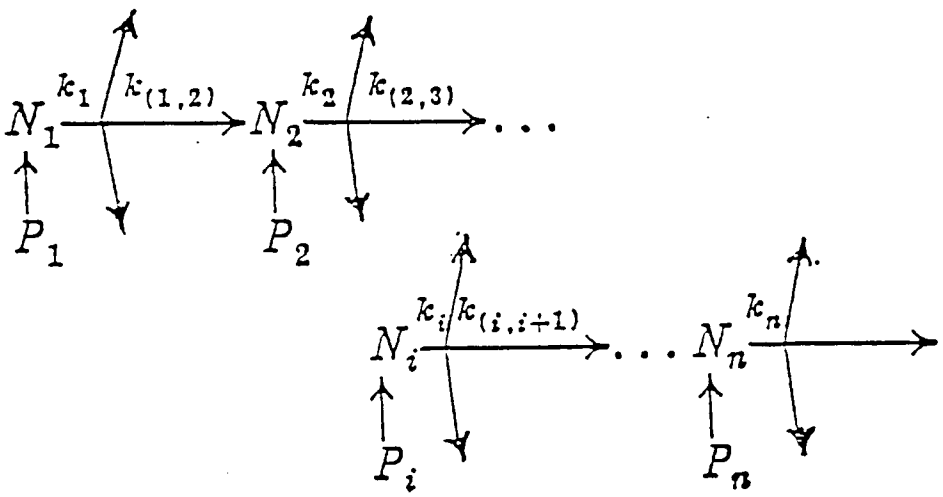
$$a_{xx} = -\sum_{y=1}^{N} r_{yx} = -K_x .$$

Each diagonal element, a_{xx}, has a negative value of the total rate constant, K_x, which equals the instantaneous fraction of the content of the xth compartment removed per unit time by all the pathways other than the radioactive decay. The diagonal elements a_{xx} of [A] are reduced by the decay constant, λ, to treat the radioactive decay at any instantaneous fractional rate. The matrix [A], for a specific compartment model, is shown in Figure 3.3. Once the matrix [A] is known, it can be used to solve the model. If the initial amounts i_x (0) in each compartment x are contained in a column vector i(0), it is shown later in this section, that the amount in compartment x at any time t is given by

$$i_x(t) = e^{[A]t} i_x(0) \qquad 3.1$$

where $e^{[A]}$ is the exponential of the matrix [A].

General equations have been obtained for the serial transformation by radioactive decay of each member of a radioactive series as described by the Bateman equation[5]. The simple general equation can be applied to any system obeying linear first order kinetics. Consider the serial transformation by any linear first order process of each member in the series:



where

N_i = quantity of the ith species present at a particular time,

k_i = total removal constant for the ith species (i. e the instantaneous fraction of the ith species destroyed per unit time by all linear first order removal processes),

$k_{(i,i+1)}$ = partial removal constant for the ith species (i.e the instantaneous fraction of the ith species transformed per unit time to the (i+1)th species), and related to the branching fraction, $f_{(i,i+1)}$:

$k_{(i,i+1)} = f_{(i,i+1)}k_i$, and

P_i = constant independent rate of production of the ith species.

The differential equations for the instantaneous time rates of change in the quantities of each member of the chain are:

$$\frac{dN_1}{dt} = P_1 - k_1 N_1$$

$$\frac{dN_2}{dt} = P_2 + k_{(1,2)} N_1 - k_2 N_2$$

.

.

.

$$\frac{dN_i}{dt} = P_i + k_{(i-1)} N_{(i-1)} - k_i N_i$$

.

.

$$\frac{dN_n}{dt} = P_n + k_{(n-1,n)} N_{(n-1)} - k_n N_n$$

These equations may be solved by standard methods to obtain the quantity of any member of the series.

3.3.2 Derivation of matrix solution for recyling models

In a compartmental model, the rate of change in the amount of material is generally determined by two competing processes: the increase from

pathways entering and the decrease from pathways leaving the compartment[4]. If the element of the rate matrix [R] has the numerical value r_{xy}, which represents the instantaneous fraction of the content of compartment x translocated per unit time to compartment y, then the the system can be described by a set of first-order differential equations which can be written as

$$\frac{d i_x}{dt} = \sum_{y=1}^{N} r_{yx} i_y - i_x \sum_{y=1}^{N} r_{xy} \qquad \text{for} \quad x \neq y \qquad 3.2$$

where i_x and i_y are considered to be functions of time representing the contents of compartments x and y respectively.

The matrix [A] can then be defined with elements given by

$$a_{xy} = r_{yx}, \text{ for } x = 1 \text{ to } N, \; y = 1 \text{ to } N \text{ and } x \neq y$$

$$a_{xx} = -\sum_{y=1}^{N} r_{xy} \quad \text{for } x = 1 \text{ to } N.$$

Substituting a_{xy} and a_{xx} into equation (2) the following relationship results.

$$\frac{di_x}{dt} = \sum_{y=1}^{N} a_{xy} i_y . \qquad 3.3$$

If the N values of i_y are considered as a column vector **I**, then the right-hand side of this equation is equivalent to a matrix multiplication of [A] by **I**, then $\frac{dI}{dt} = [A]I$ can be solved by the rules of the matrix algebra to arrive at

$$\mathbf{I} = e^{[A]t}\mathbf{I}(o), \qquad 3.4$$

where I(o) is the column vector of initial amounts in each compartment.

3.3.3 Evaluation of the exponential

The exponential of a square matrix can be solved by various methods. The approach shown below evaluates the exponential of [A] by the series expansion of $e^{[A]}$. Maclaurin series is often used to express various mathematical functions including the exponential function. It is defined for e^x as

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots \qquad 3.5$$

Similarly the exponential of a matrix A is

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = I + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots\ldots \qquad 3.6$$

This method is not often recommended since an accurate representation of $e^{[A]}$ requires a large number of terms in the series expansion. However, this can be remedied by using the following procedure:

- Reduce all the elements of [A] by a factor of 2^n. Fast convergence can be achieved by choosing n such that no element exceeds 0.2.
- Evaluate the reduced matrix $[A]/2^n$ by series expansion.

- Multiply the reduced matrix by itself n times recursively. This, reduces the number of multiplication's needed from 2^n to n. That is,

$$e^{[A]} = [e^{[A]/2^n}]^{2^n} \qquad 3.7$$

A comparison of series expansion and modified series expansion method for evaluating $e^{[A]}$ is as shown in Table 3.1[4].

The right-hand columns of Table 3.1 give :

(a) the factors by which [A] is reduced (2^n),

(b) the number of terms (k) required for convergence, and

(c) the total number of matrix multiplication's (n + k - 2) needed to evaluate $e^{[A]t}$ at various times for the above test model.

From Table 3.1 it can be seen that the modified series expansion method requires fewer matrix multiplication's than the series expansion method and hence is much faster for all values of t. The reduction factor 2^n is proportional to t, and the number of matrix multiplication's at large t is approximately equal to n. Hence when t increases exponentially, the time taken to solve the model by modified method increases linearly. Thus, $e^{[A]}$, can be evaluated rapidly for very large values of time t. In addition, the rounding errors are also minimal.

Table 3.1 A comparison of the two methods for calculating exp[A]t [4].

	Method				
	Series expansion		Modified series expansion		
Time t	No. of terms	No. of matrix multiplications	Reduction factor (2^n)	No. of terms (k)	No. of matrix multiplications (n + k - 2)
0.01	4	2	1	4	2
0.02	4	2	1	4	2
0.05	6	4	1	6	4
0.1	6	4	1	6	4
0.2	8	6	8	4	5
0.5	13	11	16	4	6
1	19	17	16	6	8
2	31	29	32	6	9
5	62	60	128	5	10
10	97	95	256	5	11
20	156	154	512	5	12
50	•	•	1024	6	14
100	•	•	2048	6	15
200	•	•	4096	6	16
500	•	•	8192	6	17
1000	•	•	16384	6	18
2000	•	•	32768	6	19
5000	•	•	131072	5	20

3.4 Description of the algorithm

The following steps describe the algorithm used for solving the computer model given in Section 3.2.

- Initialization

  <> Number compartments in the model from 1 to n

  <> Enter number of compartments in the model

  <> Enter decay constant lambda

  <> Enter rate matrix elements R_{xy}

- Calculation of [A]

  <> Transpose a_{xy} (= r_{yx})

  <> Calculate the diagonals a_{xx}

  <> Multiply [A] by t

- Reduce [A]

  <> Calculate the reduction factor (IZ)

  <> Reduce [A]

- Calculate exp [A]

◇ Series expansion of exp[A]

◇ Test for convergence

- Compute Results

◇ Calculate i_x (t) using the equation (1)

◇ Print results.

CHAPTER 4

TWO-COMPARTMENT MODEL FOR THE HEART

4.1 Model development

The physiological and biochemical information about a process in a biological system can be incorporated into a mathematical model to provide a mathematical description of the process under study. A compartment model consists of two or more compartments, interconnected by arrows to indicate the exchange of materials between compartments. The compartment model can be approximately modeled by a collection of ordinary differential equations. Each equation describes the time rate of change of amount of material in a particular compartment.

In this chapter, a two-compartment model is employed to describe the tracer dynamics of NH_3 ammonia in the heart for estimating the regional MBF(myocardial blood flow). One of these compartments represents the trapped NH_3 space and the other represents the freely diffusible NH_3 volume. The inputs for the model are radioactivity in the region of interest of the heart muscle and radioactivity concentration in the blood. The radioactivity in the region of interest in the heart is measured with PET(positron emission tomography) as a function of time, and the radioactivity concentration in the blood can be measured by analyzing samples, by using a blood pool in the heart and by a scintillation probe.

4.2 Estimation of myocardial blood flow from a two-compartment model

A two-compartment model shown in Figure 4.1 is used to describe the tracer dynamics in the heart tissue. The model consists of vascular space $Q_1(t)$ (counts/pixel/min) representing ^{13}N-ammonia activity in the freely diffusible compartment and $Q_2(t)$ (counts/pixel/min) representing the trapped ^{13}N-ammonia activity within the heart tissue. The activities in each compartment as a function of time are $Q_1(t)$ and $Q_2(t)$. The transfer rate constants between the two compartments are K_1(ml/min/g) and K_2(ml/min/g). The term V_1 (ml/g) and V_2 (ml/g) represents the distribution volume of free ^{13}N-ammonia in each compartment respectively. The arterial input function is represented by $C_a(t)$ and SPF is the spillover fraction of activity from blood to myocardium.

The rate of radioactivity change in the freely diffusible compartment and in the trapped compartment can be approximated by first order differential equations as follows:

$$\frac{dQ_1(t)}{dt} = -\frac{k_1 + F}{V_1}Q_1(t) + \frac{k_2}{V_2}Q_2(t) + F * C_a(t), \text{ and} \qquad 4.1$$

$$\frac{dQ_2(t)}{dt} = \frac{Q_1(t)}{V_1}k_1 - \frac{Q_2(t)}{V_2}k_2, \qquad 4.2$$

where F is the mycordial blood flow (MBF) in ml/min/g.

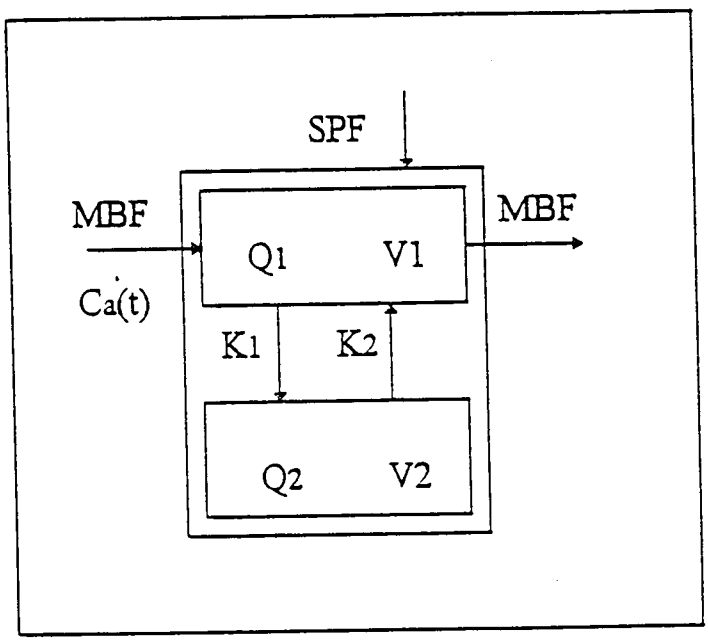


Figure 4.1. A two-compartment model representing the tracer dynamics in the myocardium [19].

Myocardial blood flow (MBF) can be determined by solving the differential equations in the two-compartment model and by fitting to the myocardial tissue time-activity curves. The total activity Q(t) is measured and is the sum of the activity of freely diffusible ^{13}N-ammonia $Q_1(t)$, ^{13}N-metabolites in the trapped pool $Q_2(t)$ and the product of spillover factor (SPF) times the arterial blood radioactivity AB(t). In particular,

$$Q(t) = Q_1(t) + Q_2(t) + SPF * AB(t). \qquad 4.3$$

In practice, Q(t) is obtained from a Positron Emission Tomography (PET) scanner. The relationship between K_1 and F during the model fitting can be described as

$$K_1 = F[1.65e^{(1.25/F)} - 1]. \qquad 4.4$$

This relationship can be derived by equating the extraction fraction (E_m) from the two-compartment model,

$$E_m = \frac{K_1}{K_1 + F}, \qquad 4.5$$

Substituting for K_1 in Equation 4.5, gives

$$E_m = 1 - 0.607e^{(-1.25/F)} \qquad 4.6$$

4.3 Estimation of Regional Myocardial Blood Flow using Patlak Graphical Analysis

The Patlak graphical analysis method is used to estimate myocardial blood flow since it is computationally straight forward and requires only linear regression[6]. Integrating Equations 4.1 and 4.2, and assuming K_2 to be negligible, results in

$$Q_1(t) = -\frac{K_1 + F}{V}\int_0^t Q_1(t)dt + F\int_0^t C_a(t)dt, \qquad 4.7$$

and

$$Q_2(t) = \frac{K_1}{V}\int_0^t Q_1(t)dt. \qquad 4.8$$

By substituting $Q_1(t)$ and $Q_2(t)$ into equation 4.3, expressing the time integral of $Q_1(t)$ in terms of $C_a(t)$, and using the relationship that $Q_1(t) = F*V(K_1+F)*C_a(t)$ at large t, one obtains the following Patlak equation:

$$\frac{Q(t)}{C_a(T)} = K\frac{\int_0^t C_a(\tau)d\tau}{C_a(t)} + \frac{F^2}{F + K_1^2} + SPF\frac{AB(t)}{C_a(t)}. \qquad 4.9$$

Note that K is the slope of the straight portion of the Patlak plot, and is given by

$$K = F\frac{K1}{F + K1} \qquad 4.10$$

By using Equations 4.5 and 4.6, Equation 4.10 can be expressed as:

$$K = F^{*}E_{m}. \qquad 4.11$$

Then

$$K = F[1 - 0.607e^{1.25/F}]. \qquad 4.12$$

Thus, myocardial blood flow (MBF) can be easily calculated, by assuming that $K_2 = 0$, and by estimating the slope, K, of the straight portion of the Patlak plot which consists of $Q(t)/C_a(t)$ (vertical (y) axis) versus $\int_0^t C_a(\tau)d\tau / C_a(t)$ (horizontal (x) axis).

4.4 Solution to the model for synthetic data generation

Synthetic data are generated by solving the differential equations of the model to test its sensitivity for different parameters. The arterial input function $C_a(t)$, is assumed to be a gamma variate function, $C_a(t) = ate^{-\beta t}$ in which 'a' varies from 0.1 to 0.6 and β is 0.1[7]. The following describes the mathematical derivation of a two-compartment model which is taken from reference [2]. Minor modifications to the model were implemented for this application. The model differential equations are,

$$\frac{dQ_1(t)}{dt} = -\frac{F + k_1}{V_1}Q_1(t) + F^{*}C_a(t) \qquad 4.13$$

$$\frac{dQ_2(t)}{dt} = \frac{k_1}{v_1} Q_1(t). \qquad 4.14$$

if K_2 is assumed to be zero. Substituting $C_a(t) = ate^{-\beta t}$, Equation 4.13 becomes

$$\frac{dQ_1(t)}{dt} + \frac{MBF + k_1}{v_1} Q_1(t) = Fat\, e^{-\beta t}. \qquad 4.15$$

Substituting $k = (F+k_1)/v_1$, in the above Equation 4.15, one obtains

$$\frac{dQ_1(t)}{dt} + kQ_1(t) = Fate^{-\beta t}. \qquad 4.16$$

$$e^{kt} Q_1(t) = \int e^{kt} Fate^{-\beta t} dt + C_1$$

$$= \frac{Fa}{(k-\beta)} \int t de^{(k-\beta)} + C_1$$

$$= \frac{Fa}{(k-\beta)} [te^{(k-\beta)t} - \frac{1}{(k-\beta)} e^{(k-\beta)t}] + C_1 \qquad 4.17$$

$$Q_1(t) = \frac{Fa}{(k-\beta)} te^{-\beta t} - \frac{Fa}{(k-\beta)^2} e^{-\beta t} + C_1 e^{-kt}. \qquad 4.18$$

The constant C_1 is obtained by using the initial condition $Q_1(0) = 0$ in Equation 4.18, hence

$$C_1 = \frac{Fa}{(k-\beta)^2}. \qquad 4.19$$

Substituting the value of C_1 in the Equation 4.18, one obtains

$$Q_1(t) = \frac{Fa}{(k-\beta)}[te^{-\beta t} - \frac{1}{(k-\beta)}e^{-\beta t} + \frac{1}{(k-\beta)}e^{-kt}]. \qquad 4.20$$

Substituting $C = k_1/v_1$ and integrating both sides of the Equation 4.13 results in

$$Q_2(t) = \int C\,Q_1(t)dt + C_2$$

$$= \frac{FaC}{(k-\beta)}[-\frac{1}{\beta}(te^{-\beta t} + \frac{1}{\beta}e^{-\beta t}) + \frac{1}{(k-\beta)\beta}e^{-\beta t} - \frac{1}{(k-\beta)k}e^{-kt}] + C_2. \qquad 4.21$$

The constant, C_2 can be determined by using the initial condition, $Q_2(0) = 0$, and is given by the following equation,

$$C_2 = -\frac{FaC}{(k-\beta)}[-\frac{1}{\beta^2} + \frac{1}{(k-\beta)\beta} - \frac{1}{(k-\beta)k}]. \qquad 4.22$$

Substituting the value of C_2 in equation 4.21, gives

$$Q_2(t) = \frac{FaC}{(k-\beta)}[-\frac{1}{\beta}(te^{-\beta t} + \frac{1}{\beta}e^{-\beta t}) + \frac{1}{(k-\beta)\beta}e^{-\beta t} - \frac{1}{(k-\beta)k}e^{-kt} \quad 4.23$$
$$-(-\frac{1}{\beta^2} + \frac{1}{(k-\beta)\beta} - \frac{1}{(k-\beta)k})].$$

Then the synthetic data representing the activity of the PET scan region of interest is given by

$$Q(t) = Q_1(t) + Q_2(t) + SPF * C_a(t). \quad 4.24$$

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

Artificial neural networks are applied as non-parametric models to biological systems, which are traditionally represented as compartment models. In this study, neural network models are developed to identify parameters of two, three and four compartment models. The model parameters are namely translocation rates, initial amount of radioactive materials and regional MBF.

Neural network models were trained with a backpropagation agorithm using the Norm-cum rule and the hyperbolic tangent transfer function[1]. The structure of backpropagation network model consists of an input layer, an output layer and one or more hidden layers. Depending on the complexity of the application, it may require more than one hidden layer and numerous hidden nodes. In this research, some neural network models were trained with two hidden layers. A trial and error procedure is employed to obtain the suitable number of hidden layers and hidden layer nodes to produce satisfactory results for a given application.

The neural network models were trained with simulated data obtained from the algorithms developed in Chapter 3 and Chapter 4 for solving compartment models. The backprop builder from Neuralworks Professional II / plus software package was used to build the neural network models.

5.2 Two-compartment model

Several network models were developed and studied for a two-compartment model. They are discussed in sections below.

5.2.1 Network based on synthetic data

The network for identifying the regional myocardial blood flow consisted of 12 nodes in the input layer, 10 nodes in the first hidden layer and 6 nodes in the second hidden layer. The output layer has only one node. The synthetic data for training the network were obtained from the algorithm described in Chapter 4. Approximately 150 patterns of data were presented. The training process continued until the RMS error between the desired and predicted output was reduced to 0.03. This required approximately 21,000 iteration cycles. The network was then tested with 20 patterns of data different from the ones used in training where the MBF was known. The results showed that network was able to predict the regional MBF with more than 96% accuracy. Of the 20 patterns that were used for testing, the model predicted 18 patterns with an accuracy of 97%. The other 2 patterns were different from the desired

output only by 8% error. The results are shown in Figure 5.1. It can be observed that it has a good correlation between both estimates.

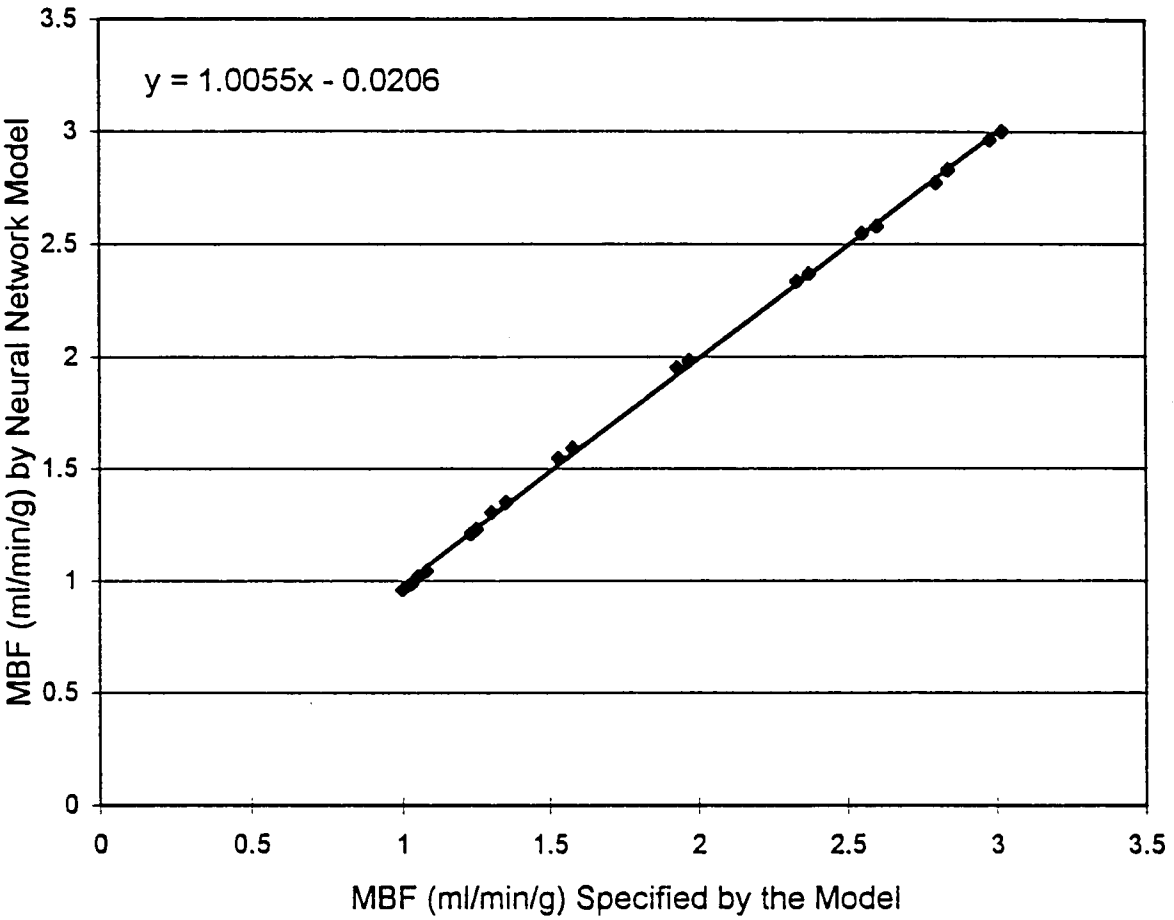


Figure 5.1. Comparison of regional MBF estimated by neural network model and that specified by the model. The regressional lines and correlation coefficients were y = -0.0206 + 1.0055x, r = 0.99.

The network was then tested with 18 patterns of data obtained from Positron Emission Tomography(PET) scans. The PET scan data are usually in binary format. A computer program included in Appendix H[2] was used to convert the binary data to readable format. The data were then normalized prior to using it for training the network. The neural network predicted the regional MBF with more than 96% accuracy. The results are shown in Figure 5.2. From the Figures 5.1 and 5.2, MBF estimated by the neural network model correlates well with the measured blood flow values from two-compartment model and the positron emission tomography scans. The linear regression line demonstrates a strong relationship between both estimates.

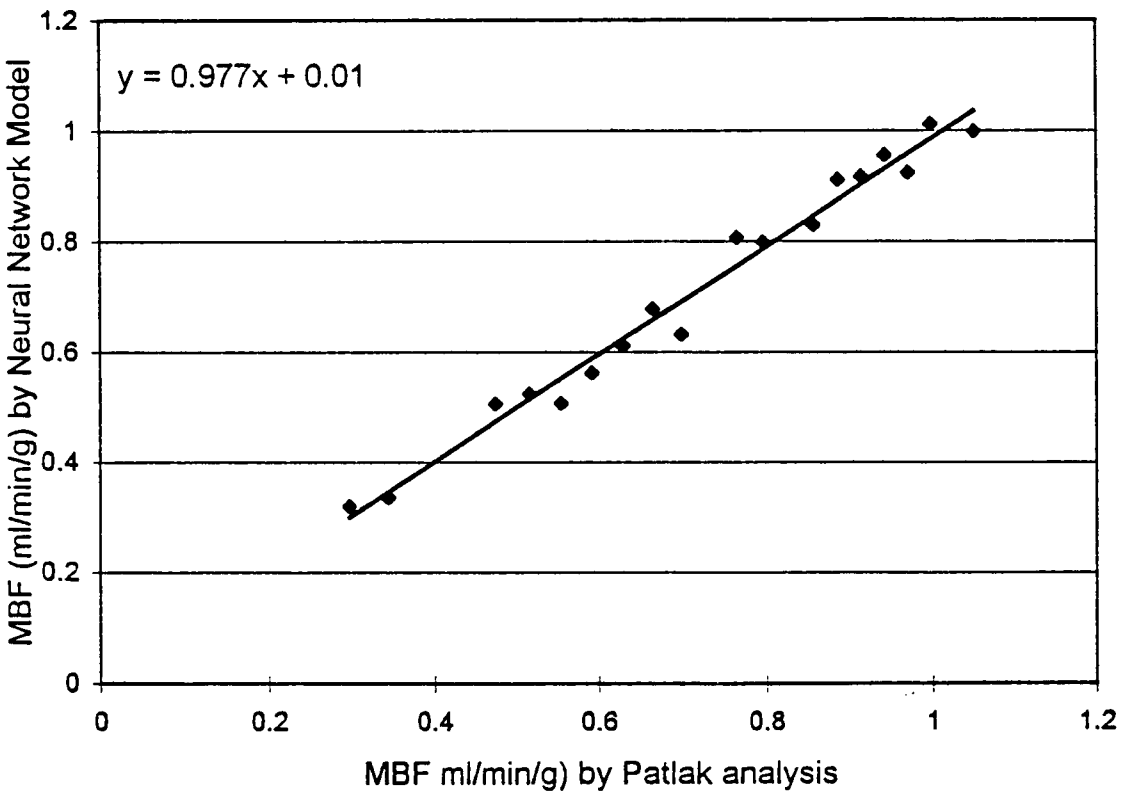


Figure 5.2. Comparison of regional MBF estimated by Patlak graphical analysis method and neural network model for Positron Emission Tomography data. The regressional lines and correlation coefficients were $y = 0.01 + 0.977x$, $r = 0.96$.

5.2.2 Network based on noisy data

The performance of the model was also evaluated with noise contaminated data since the data normally contains some level of measurement noise. This was accomplished by adding 2% noise to the original testing patterns. The noise was generated by using a random number generator with a Gaussion distribution whose mean and variance are 0.0 and 1.0 respectively. The following relationship was used to add the noise to the data.

$$X_n(t) = X(t)[1 + \alpha n(t)]$$

where

$X_n(t)$ = noisy data

$X(t)$ = simulated data

$n(t)$ = noise data

α = noise percentage (fraction).

Similar calculations were performed for 4% and 6% respectively. The computer program for generating noise in the data is given in Appendix E and the results are shown in Figure 5.3. A good correlation of regional MBF estimates over a wide range can be observed with correlation r = 0.97. Thus, one can conclude that the neural network models are not very sensitive to small percentage of noise present in the data.

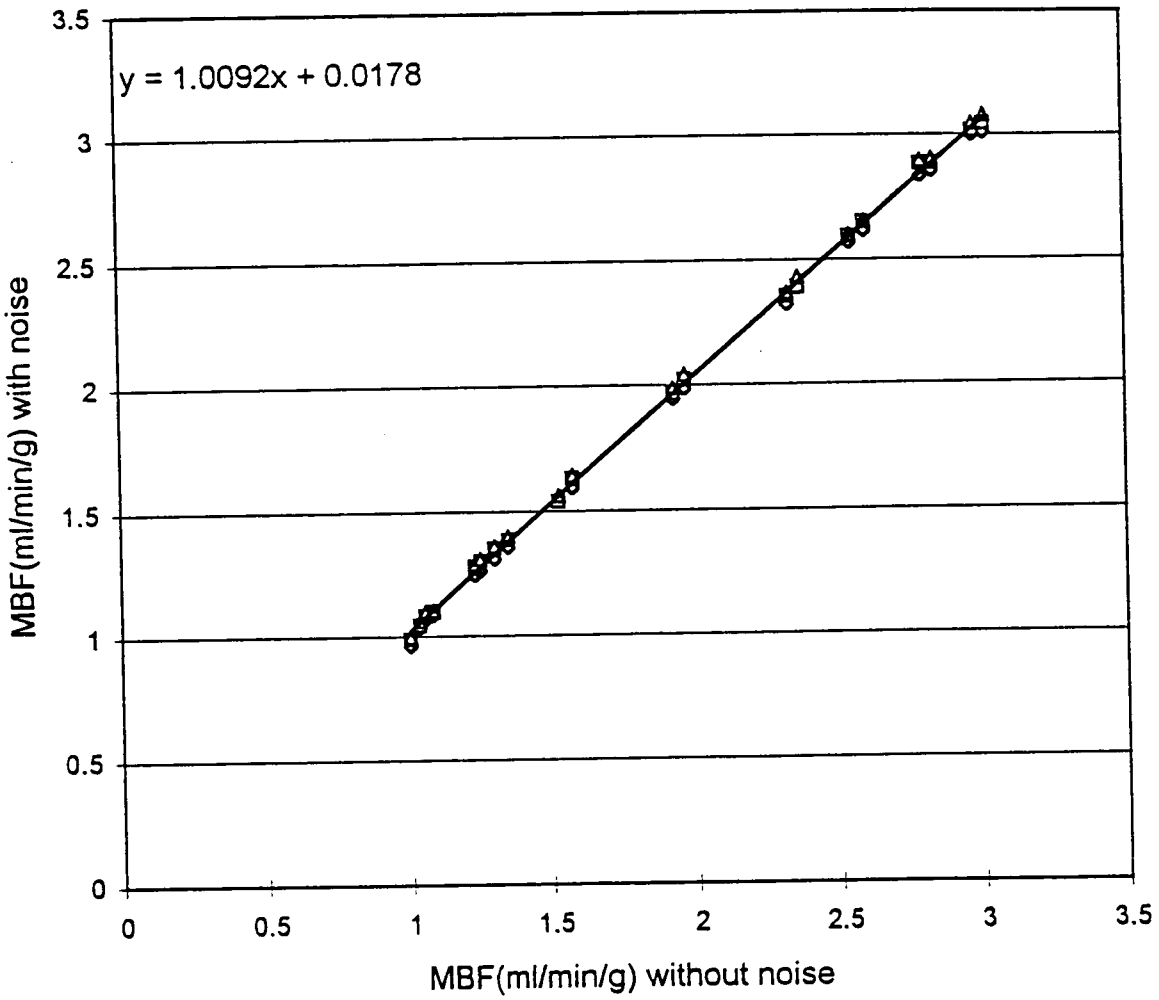


Figure 5.3. Comparison of regional MBF without noise and with 2%, 4% and 6% noise in the data. The regressional lines and correlation coefficients were $y = 0.0178 + 1.0092x$, $r = 0.97$.

5.2.3 Network based on moments of data

Another network model was developed to identify the regional MBF by training the neural network with moments about the mean of the data. The algorithm to obtain moments of the data is included in Appendix C. The network consisted of 5 nodes in the input layer, 3 nodes in the hidden layer and one in the output layer. The training data consisting of moments were presented to the network and trained until the error was reduced to 0.05. This required 48,000 training iterations. The network was then tested with 20 patterns of data different from the ones used for training. The predicted results were in the the error by approximately 12%. Figure 5.4 shows the results of this study. It can be observed that the correlation of MBF was r = 0.98. An advantage of this approach is that it can be used for arbitrary length scan times and scan methods since it is independent of time.

5.2.4 Network based on numerical integration of the data

Numerical integration is the process of calculating the value of a definite integral $\int_a^b f(x)dx$ from the tabulated values of f(x). This is done by using the trapezoidal rule which is explained in Appendix G. A network model was developed to identify regional MBF by training the network with the data obtained by trapezoidal rule method. The reason for using this method was to eliminate the time dependent factor in the input. The algorithm is included in the Appendix F. The network model consisted of 11 nodes in the input layer, 8 nodes in the first hidden layer and 6 nodes in the second hidden layer.

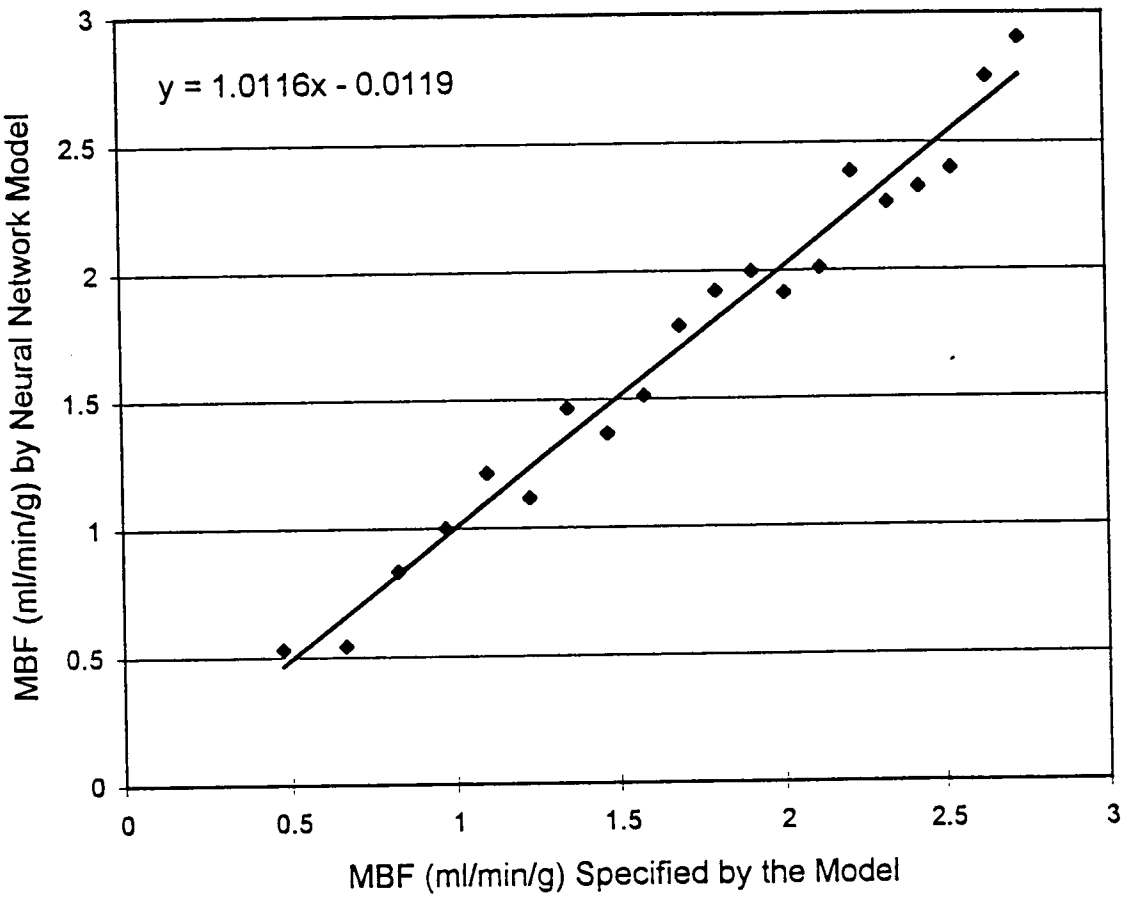


Figure 5.4. Comparison of regional MBF estimated by moments about the mean of the data and neural network model. The regressional lines and correlation coefficents were $y = -0.0119 + 1.0116x$, $r = 0.98$.

The output layer has one node. One hundred patterns of data were presented to the network for training. The training was achieved after 35,000 iterations with an error of 0.03. The network was examined with 25 unseen patterns of data not used in the training. The results show that the predicted regional MBF has an accuracy of 97%. The error was less than 3%. It can be seen from Figure 5.5, that there is a good correlation between the neural network model estimates and that obtained by trapezoidal rule method. The network was also tested by perturbing the test inputs. The results indicate that the predicted output is not greatly affected by perturbation in the input patterns. This is graphically shown in Figure 5.6.

5.2.5 Network based on Patlak Data

Neural network models were also developed to identify the regional MBF by training the network with the data obtained from the Patlak graphical analysis method. It is described in Chapter 4. The network model consisted of 10 nodes in the input layer, 8 nodes in the first hidden layer and 6 nodes in the second hidden layer. The output layer had one node. Approximately 120 patterns of data were presented to the network for training. The training was achieved after 40,000 iteration cycles with an error less than 4%. The network was examined with 25 unseen patterns of data not used in the training. The results show that the predicted regional MBF has an accuracy of more than 96%. The slope of the linear regression line of the MBF estimates obtained by the Neural network model and Patlak graphical analysis was

about 1.0. Figure 5.7 shows excellent correlation of the MBF estimates obtained by the Patlak graphical analysis and the Neural network model.

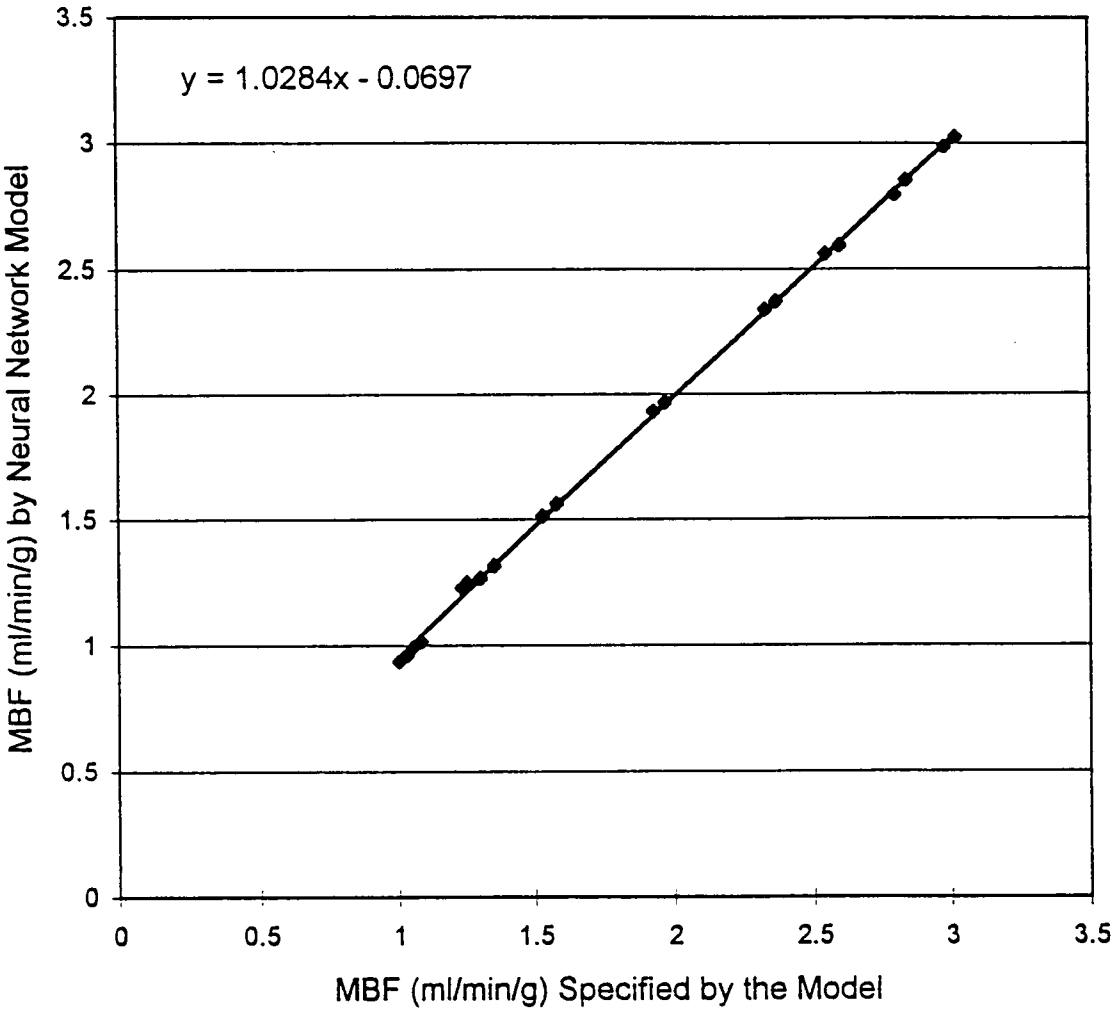


Figure 5.5. Comparison of regional MBF estimated with data obtained by trapezoidal rule method and neural network model. The regressional lines and correlation coefficients were y = -0.0697 + 1.0284, r = 0.98.

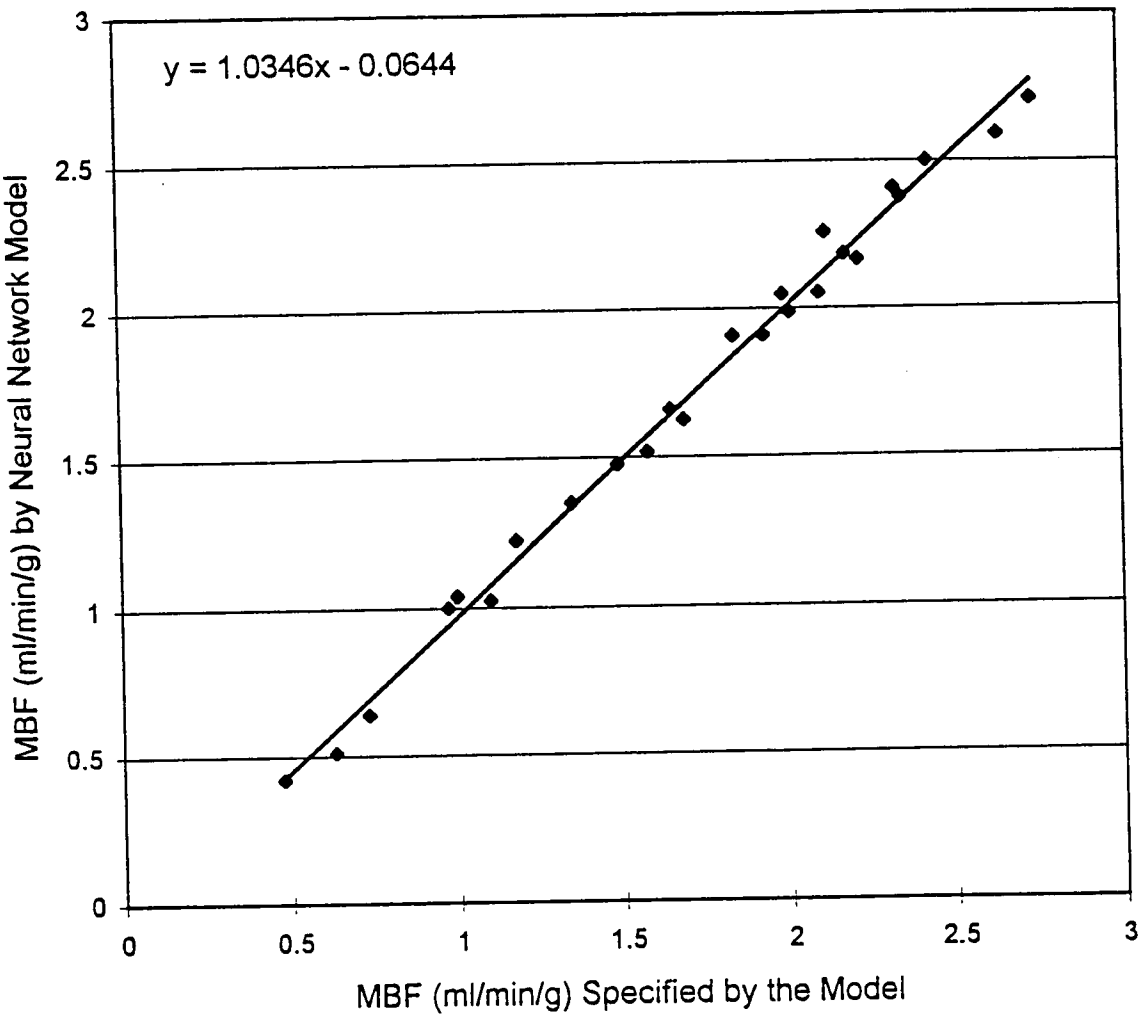


Figure 5.6. Comparison of regional MBF estimated by trapezoidal rule with perturbed data and neural network model. The regressional lines and correlation coefficients were y = -0.0644+1.0346x, r = 0.95.

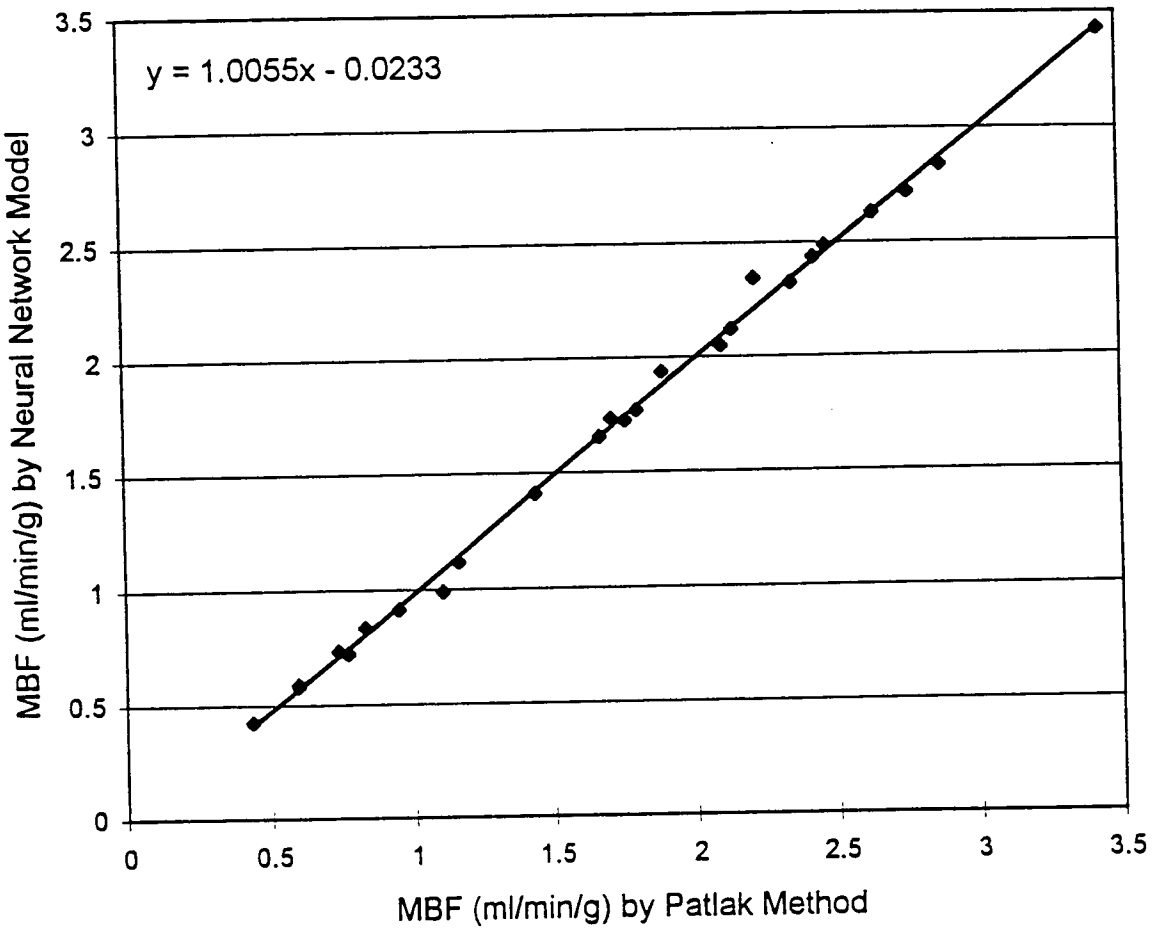


Figure 5.7. Comparison of regional MBF estimated by Patlak method and by Neural network model. The regressional lines and correlation coefficients were y = -0.0233 + 1.0055x, r = 0.97.

5.2.6 Discussion

Several techniques based on Artificial neural network methodology are suitable for parameter identification in compartment models used in nuclear medicine applications. This includes the development of several neural network models to predict regional myocardial blood flow and other parameters of importance.

The performance of the models was examined with unseen synthetic data, data obtained from PET scans and noisy data. The network models predicted regional MBF with less than 3% error. This study indicates that the Neural network methodology is an efficient and accurate approach for quantifying regional myocardial blood flow. The excellent correlation of the MBF estimates obtained by the Neural network model with Patlak graphical analysis and the two-compartment model indicates that they can produce good estimates of regional MBF. The results may be obtained with very little computational effort and results are insensitive to small percentage of noise present in the data.

5.3 Three-compartment model

A typical schematic diagram of the neural net model for a three-compartment model is shown in Figure 5.8. It includes an output layer, input layer and an hidden layer. This structure was used to construct two neural

network models, one for identifying the initial amount of material in each compartment and the other for identifying the output after time 't'.

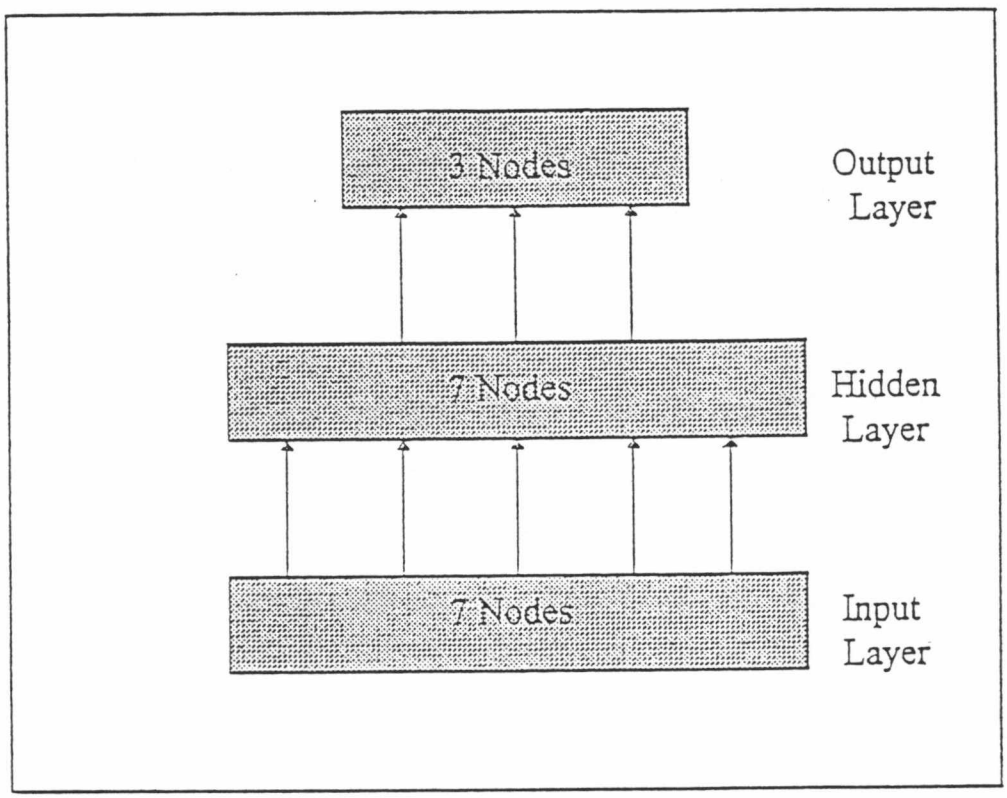


Figure 5.8 Network model representing a three-compartment model

5.3.1 Network for identifying initial amount of radioactive material present in the compartments

The network for identifying the initial amount consisted of 7 nodes in the input layer with one representing the time 't', three for translocation rates between compartments and three for specified amount of material after time 't'. The hidden layer consisted of 7 nodes. The output layer was composed of 3 nodes each representing the initial amount of material that was present in each of the three compartments. Approximately 70 patterns of data were presented to the network for training. The network was trained individually until the maximum error for each pattern was reduced to 0.02. This required 40,000 training iteration cycle. The network was then tested with 20 patterns of data different from the ones used in the training. The results indicated that the predicted initial amount of material in each compartment has an accuracy of more than 90%. As shown in Figure 5.9, initial amount of radioactive material estimated by neural network model correlates well with those obtained by compartment model method.

5.3.2 Network for identifying specified amounts of radioactive material left in the compartments after time 't'

The network for identifying the specified amounts included 7 nodes in input layer with one node representing the time t, three for translocation rates and three for initial amount of material. The hidden layer was composed of 7 nodes. The output layer consisted of 3 nodes each representing the final

amount of material that will be present after time 't' in each of the compartments. Approximately the same number of patterns were presented to the network as in the previous case, until the maximum error reduced to 0.01. Similar to the previous case, the number of training cycles was 40,000 iterations. The network was examined with unseen patterns, to evaluate its performance. The predicted specified amount of material in each of the 3 compartments at time 't' had an error less than 10%. As can be seen from Figure 5.10, they have a correlation coefficient of r = 0.86.

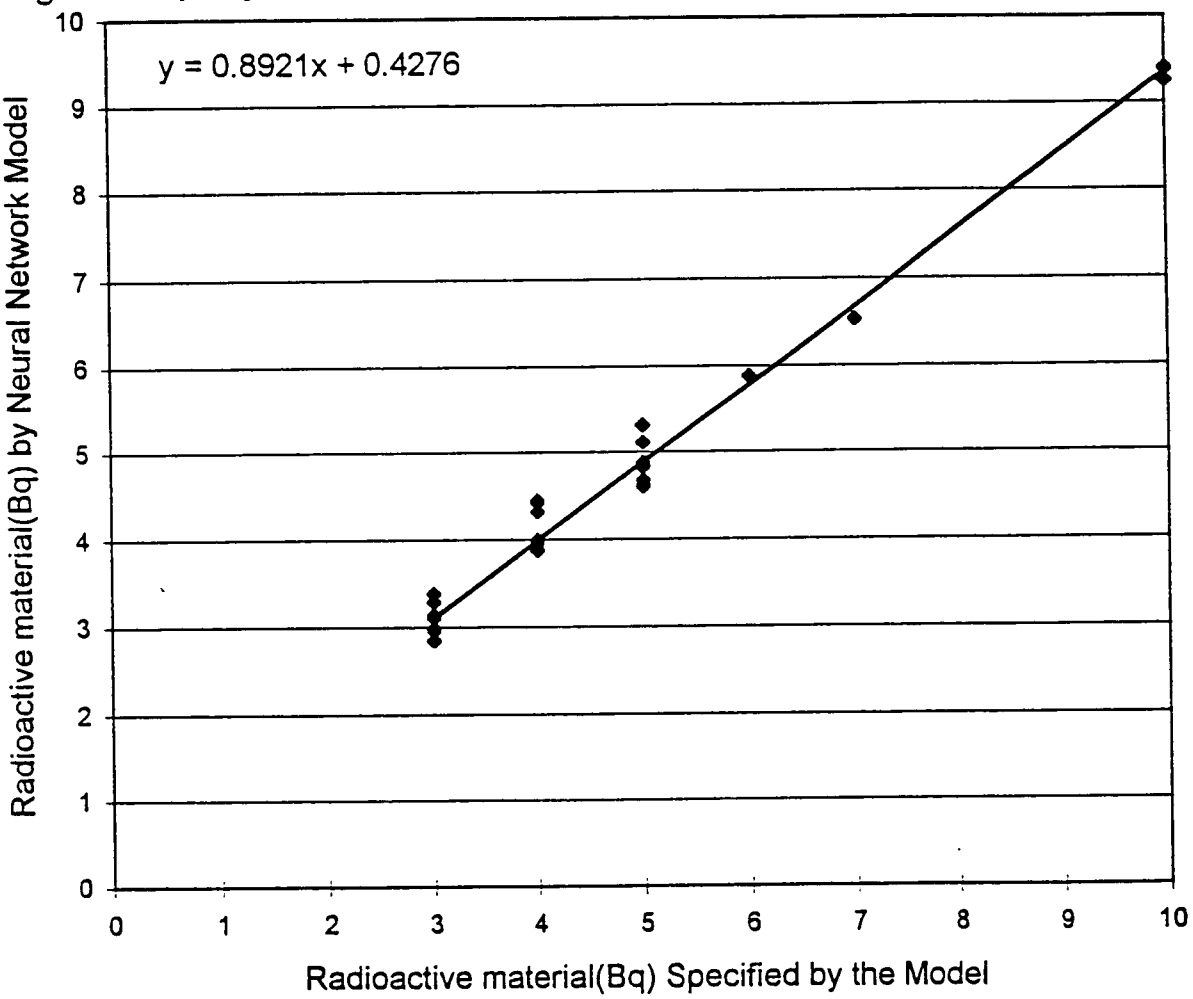


Figure 5.9. Comparison of initial amounts of radioactive material(Bq) estimated by neural network model and that specified by the model. The regression fit and correlation fit coefficients were y = 0.4276 + 0.8921x, r = 0.87.

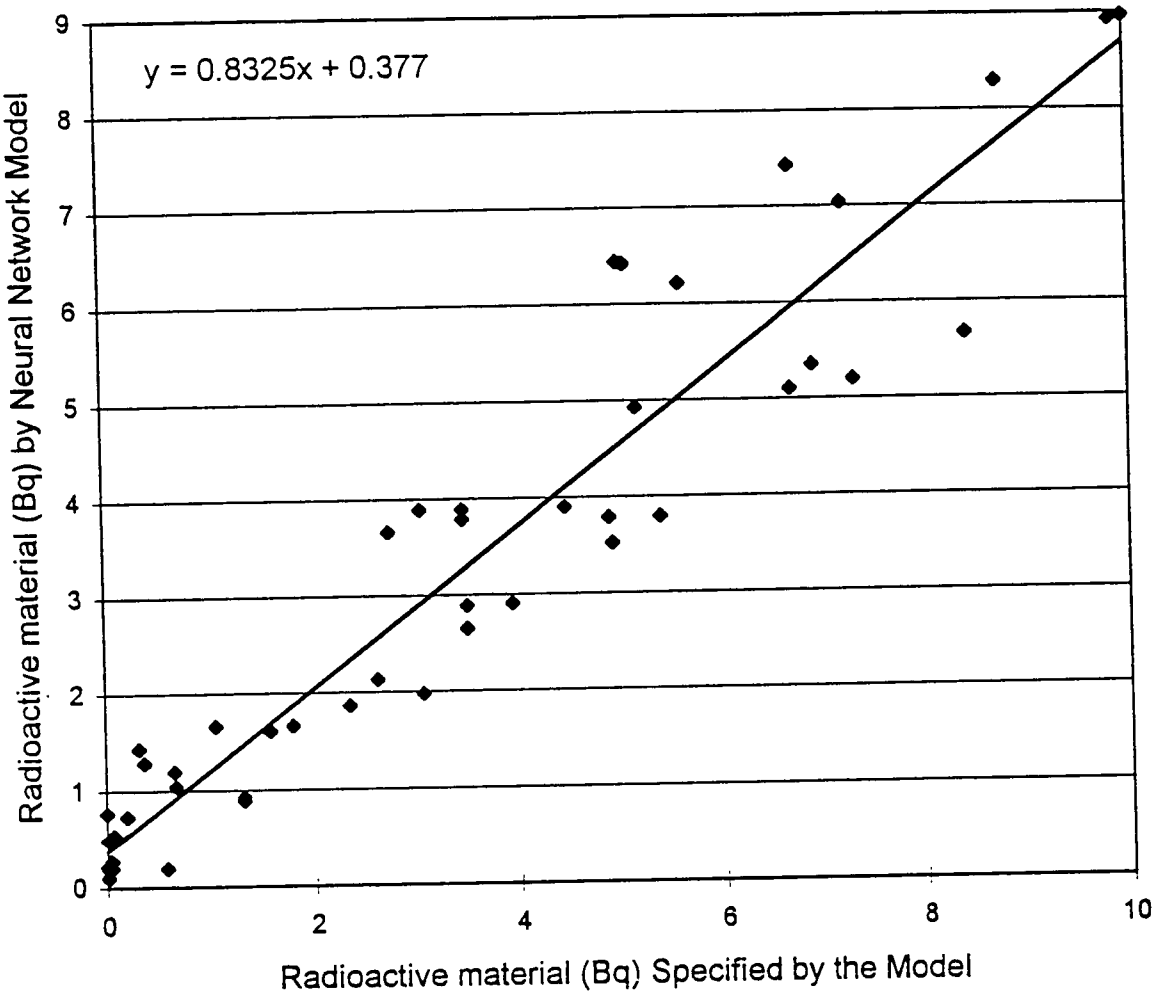


Figure 5.10. Comparison of specified amounts of radioactive material (Bq) left in each compartment estimated by Neural network model and that specified by the model. The regression fit and correlation coefficients were $y = 0.377 + 0.8325x$, $r = 0.84$.

5.4 Four-compartment model

Figure 5.11 shows a typical neural network model analogous to a four-compartment model employed in biological system. The model structure is composed of 3 layers, namely input layer, output layer and a hidden layer. This was used to construct two neural network models to identify various parameters in the compartment model.

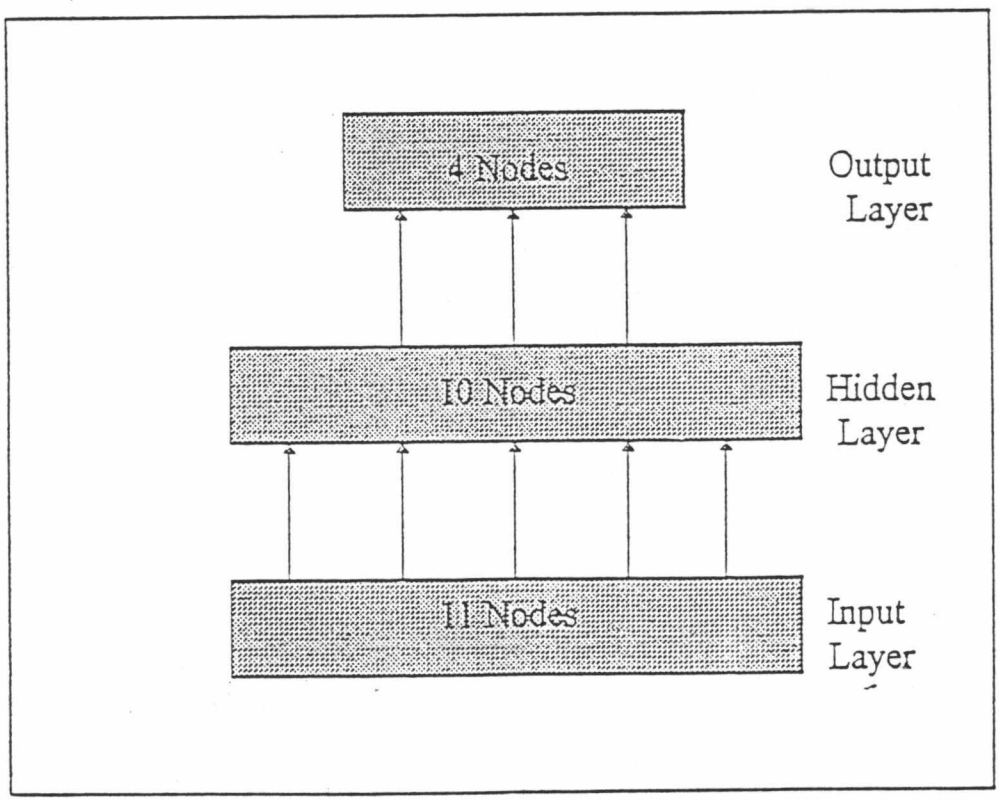


Figure 5.11 Network model representing a four-compartment model.

5.4.1 Network for identifying initial amount of radioactive material present in the compartments

The network for identifying the initial amount consisted of 11 nodes in the input layer with one node representing time 't', six for translocation rates between the compartments and four for specified amounts of material after time 't'. The hidden layer included 10 nodes. The output layer consisted of 4 nodes each representing the initial amount of material that was present in each of the 3 compartments. Approximately 85 patterns of data were presented to the network for training. The network was then tested with 25 patterns of data different from the ones used in training. The network predicted the initial amount of material in each of the compartment with more than 90% accuracy. Figure 5.12 shows the linear regression fit to the data.

5.4.2 Network for identifying the specified amount of radioactive material left in the compartments after time 't'

The network for identifying the specified amount consisted of 11 nodes in the input layer with one node representing time 't', six for translocation rates and four for initial amount of material. The hidden layer consisted of 10 nodes. The output layer consisted of 4 nodes each representing the final amounts of material that will be present after time t in each of the compartments. About the same number of patterns were presented as in the previous case until the maximum error was reduced to 0.01. Network was trained for 44,000 iterations and tested with new patterns. The network

predicted the final amount of material in each of the compartments with an error less than 10%. The regression lines and correlation coefficients were y = 0.4738 + 0.8299x, r = 0.86, as seen from the Figure 5.13.

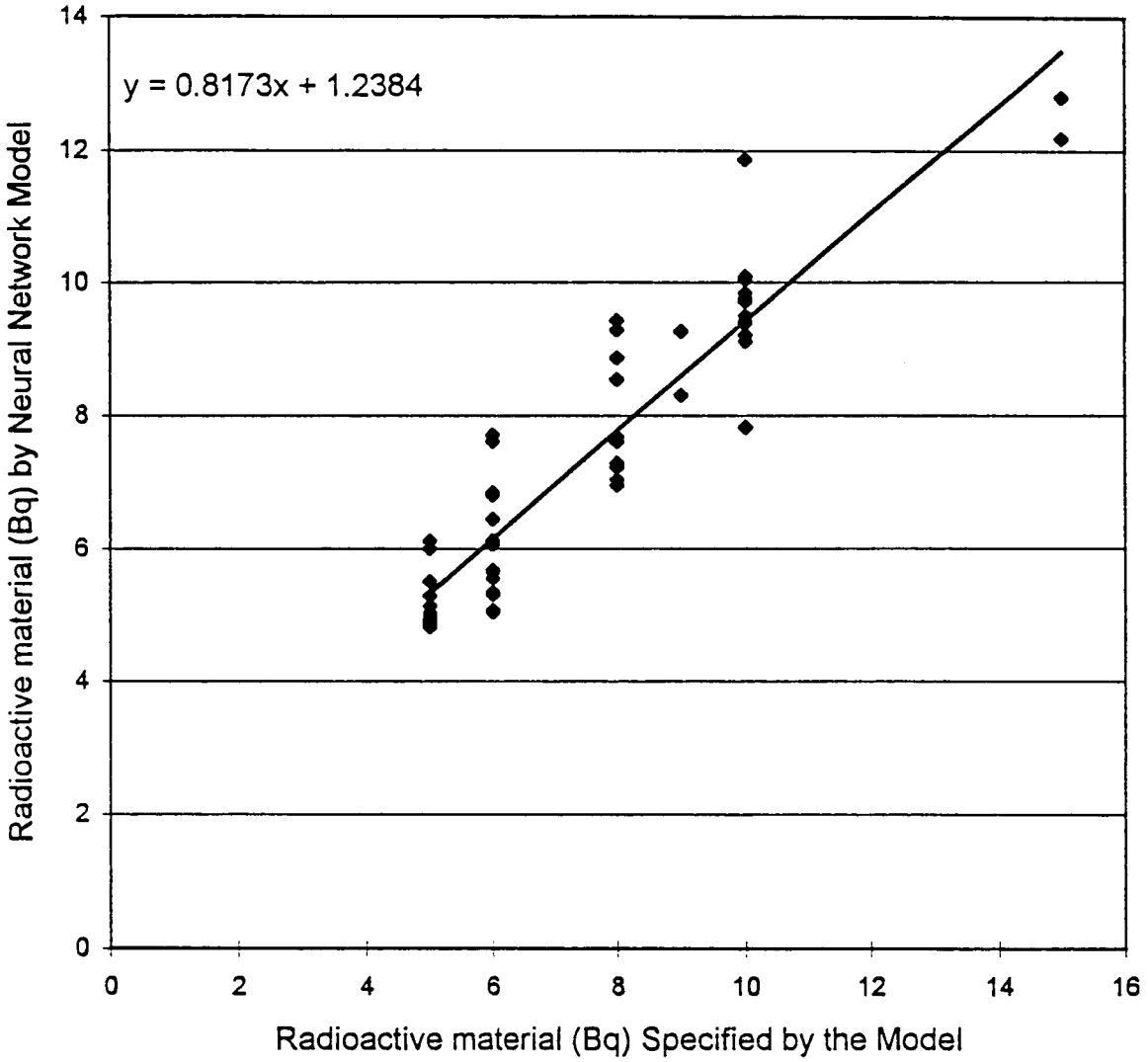


Figure 5.12. Comparison of initial amounts of radioactive material (Bq)in each compartments estimated by neural network model and that specified by the model.The regressional lines and correlation coefficients were y = 1.2384 + 0.8173x, r = 0.87.

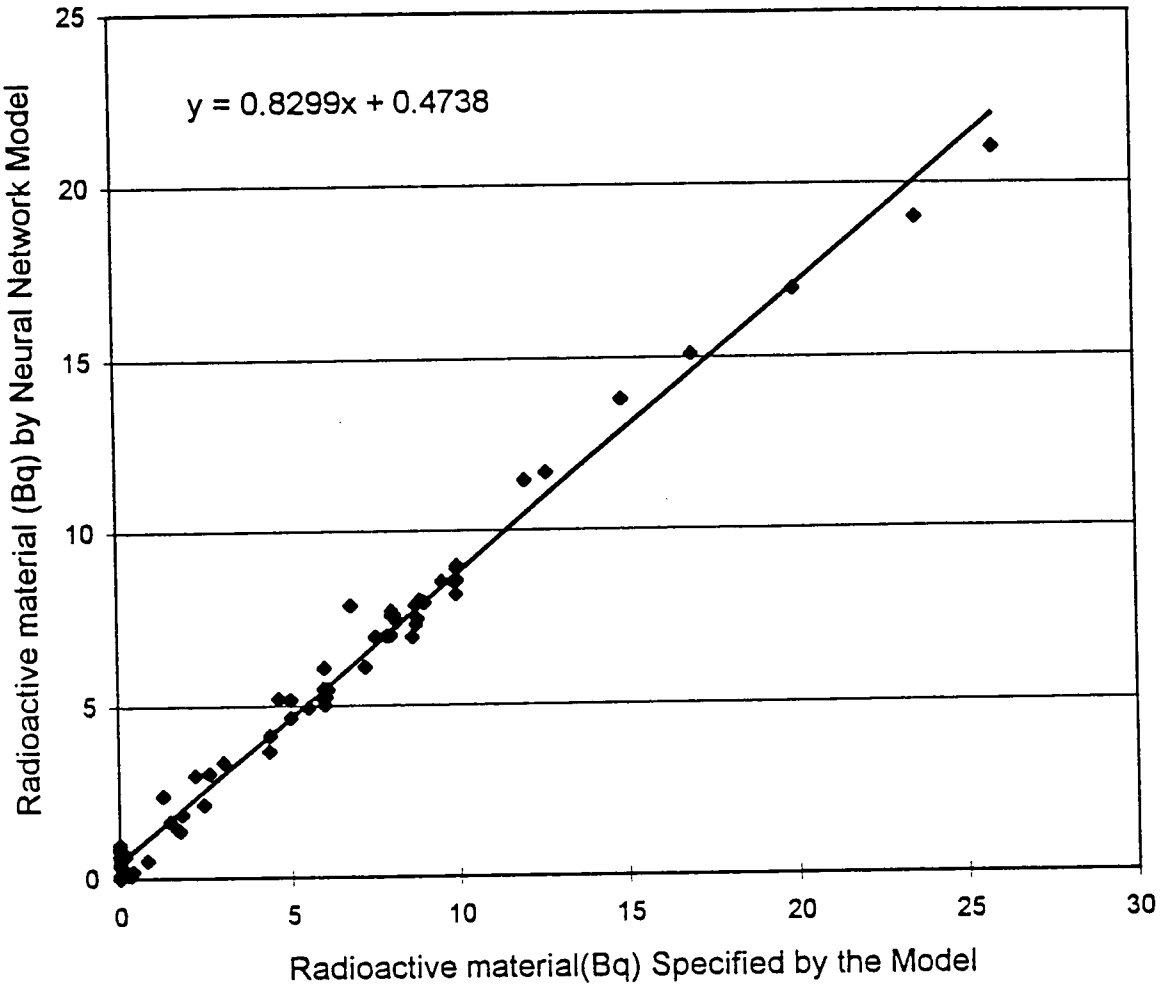


Figure 5.13. Comparison of specified amounts of radioactive material (Bq) left in each compartment estimated by neural network model and that specified by the model. The regression fit and correlation coefficients were $y = 0.4738 + 0.8299x$, $r = 0.86$.

5.4.3 Discussion

A methodology has been presented for identifying the model parameters in a compartment model. Two different networks were developed for identifying different model parameters, one for identifying the initial amount of material in each compartment and the other for predicting the specified amount of material in each compartment. The network model was constructed and tested for both three- compartment and four-compartment models.

The study of these results show that network is able to identify the initial amount of material in each compartment with a good accuracy. For the case of identifying the final amount of material in each compartment, the predicted values were different from the desired values with an error less than 10%. This was due to the fact that the range of values in the input data field was relatively high.

In conclusion, it can be said that the neural network models are able to accurately identify various parameters of interest, in biological systems used in internal dosimetry applications. On the whole, the important application of the neural network is that it can be used to identify model parameters like the initial amounts of material that was present in each compartment.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

A neural network methodology was employed in this research study to develop non-parametric models, analogous to traditional compartment models, for biological systems. The primary function of neural network models was to identify parameters of the compartment models used in internal dosimetry and nuclear medicine applications. These parameters include initial amounts of radioactive material present in the compartments given the other parameters of the compartment model, and translocation rate constants between compartments.

To generate training and testing data for the ANN models, a computer algorithm described in Chapter 3, was first developed to solve the compartment models for dosimetry applications. The performance of the models were then evaluated by comparing the results obtained from the ANN models and with traditional compartment models. It was found that the accuracy of ANN results were approximately 95% of those obtained from computer algorithm.

Similarly, two additional models were developed for parameter identification in nuclear medicine applications. The networks were trained with the synthetic data generated for the compartment model, described in

Chapter 4 to estimate regional myocardial blood flow. Data obtained from positron emission tomography (PET) scans were used to test the performance of the ANN models. The results show that the error between the desired output and those obtained from the models was less than 3%. It was also found that the network models were insensitive to noise present in the data. This indicates that the ANN models are able to perform better or atleast as good as compartmental models for such applications.

From these findings, it can be said that the neural network methodology is able to identify the parameters of compartment models with a good accuracy, even with noise contaminated data. In addition, the neural network models can capture the nonlinear characteristics of data more efficiently and in a less complicated manner than traditional techniques. Results from these studies demonstrate that ANN models are reliable and suitable tools for identification and estimation of various parameters in compartmental models employed in biological systems.

As future work to this study, it would be interesting to examine the applications of artificial neural network models for

1. Parameter identification in kinetic models describing the transport of radionuclides in the environment.
2. To generate parametric images of regional myocardial blood flow in a two-compartment model used to describe tracer dynamics in the heart tissue.

LIST OF REFERENCES

LIST OF REFERENCES

1. NEURALWARE INC., Neural Computing, Manuals for Neuralworks Professional II and Neuralworks Explorer Software, Vol.1, Pittsburgh, PA, 1990.

2. MILLER LF, Personal Communications, 1992.

3. ISRAEL E. ALGUINDIGUE, "A methodology for the analysis and monitoring of vibration data using artificial neural network systems", May 1993.

4. A. BIRCHALL, A. C. JAMES, "A microcomputer algorithm for solving first-order compartmental models involving recycling", Health physics, v.56 pp.857-868, 1988.

5. K. W. SKRABLE, C. FRENCH, G. CHABOT, A. MAJOR, "A general equation for the kinetics of linear first order phenomena and suggested application", Health Physics v.27 pp. 155-157, 1974.

6. SMITH GT, HUANG SC, NEINABAR CA, KRIVOKAPICH J, SCHWAIGER M, SCHELBERT HR, "Noninvasive quantification of regional myocardial blood flow with N-13 ammonia and dynamic PET", Journal of Nuclear Medicine, Vol. 29, pp 950, 1988.

7. W. S. MCCULLOCH, W. PITTS, "A Logical Calculus of the Ideas Imminent in Nervous activity", Bulletin of Mathematical Biophysics, 5, pp.115-133, 1943.

8. DARPA, Neural Networks study, AFCEA International Press, Fairfax, Virginia, 1988.

9. D. O. HEBB, The Organization of Behavior, John Wiley & Sons, New York, 1949.

10. R. P. LIPPMANN, "An Introduction to Computing with Neural Networks', IEEE ASSP Magazine, April 1987.

11. R. ROSENBLATT, Principles of Neurodynamics, Spartan Books, New York, 1949.

12. D. E. RUMELHART, J. L. MCCLELLAND, Parallel Distributed Processing, MIT Press, Cambridge, MD, 1986.

13. A. BIRCHALL, "A microcomputer algorithm for solving compartmental models involving radionuclide transformations", Health Physics. V.50 pp. 389-397, 1986.

14. R. BRONSON, "Matrix Methods", Academic Press, 1990.

15. G. J. BORSE, "Fortran 77 and numerical methods for engineers", PWS Engineering, c1985.

16. SAMUEL L. MARATECK, "Fortran 77", Academic Press, c1985.

17. JOHN SHELLY, "Essentials of Fortran 77", Chichester: New York, J.Wiley, c1985.

18. WU Y, "Measurement of regional myocardial blood flow by compartment modeling", December 1990.

19. YONG CHOI, SUNG-CHENG HUANG, RANDALL A. HAWKINS, WILLIAM G. KUHLE, MAGNUS DAHLBOM, CARL K. HOH, JOHANES CZERNIN, MICHAEL E.PHELPS AND HENRICH R. SCHELBERT, "A simplified method for quantification of myocardial blood flow using nitrogen-13-ammonia and dynamic PET". The Journal Nuclear Medicine, vol. 34, No. 3, March 1993.

20. PATLAK CS, BLASBERG RG, FENSTERMACHER JD, "Graphical evaluation of blood-to-brain transfer constants from multiple-time uptake data", Journal of Cerebral Blood Flow Metabolism, 1983: v.3 : 1-7.

21. MILLER LF, SMITH GT, WU Y, UHRIG RE, "Evaluation of Neural Networks for Parameter Identification from Positron Emission

Tomography Scans", Transactions of the American Nuclear Society, November 11-16, 1990.

APPENDICES

APPENDIX A

PROGRAM TO RUN A COMPARTMENT MODEL

This program can be applied to all first order compartment models. It is explained in detail in Chapter 3. When the program is executed, it asks for initial and final time. The time is usually specified in days. The program then gives the initial and final amounts of radioactive material in each compartment.

```
c       Program to solve a compartment model

        implicit real*8 (a-h, o-z)
        dimension r(50,50), a(50,50), sum(50,50), term(50,50), b(50,50)
        xt(50), xo(50), q(50,100)

        n = 4
        lam = alog(2.0)/5.85e5
        terr = 1e-10
        r(1,1) = 20
        r(1,2) = 4.85e-1
        r(2,3) = 2.67e-2
        r(3,2) = 2.95e-3
        r(2,4) = 3.83e-2
        r(3,4) = 2.92e-1
        r(2,2) = 12
```

```
      r(3,3) = 10
      r(4,4) = 0
      print*, 'Enter initial time'

      read(*,*)ti
      print*, 'Enter the step size for the time'
      read(*,*)step
      print*, 'Enter the final time'
      read(*,*)tf
      t = ti
17    d = lam*t
      do i = 1,n
      do j = 1,n
      sum(i,j) = 0.0
      a(i,j) = r(j,i)
      enddo
      enddo
      do i = 1, n
      a(i,i) = -lam
      do 34 k = 1, n
         if (k eq i) then
         goto 34
         else
         a(i,i) = a(i,i) - r(i,k)
         endif
```

```
34      continue
        enddo

        do i = 1, n
        xo(i) = r(i,i)
        sum(i,i) = 1
        term(i,i) = 1
        enddo
        amax = 0
        do i = 1,n
           if (a(i,i).lt.amax) then
           amax = a(i,i)
        enddo

        do iz = 0, 10000
        ak1 = 2**float(iz)
        ak2 = -amax/ak1
              if(ak2.lt.0.2) then
              goto 240
              endif
        enddo

240     continue
        do i = 1,n
        do j = 1,n
```

```
a(i,j) = a(i,j)/ak1
enddo
enddo

do 56 ir = 1, 10000
do j = 1,n
do i = 1,n
b(i,j) = 0.0
do k = 1,n
b(i,j) = b(i,j) + term(i,k)*a(k,j)
enddo
enddo
enddo

do i = 1,n
do j = 1,n
term (i,j) = b(i,j)/ir
sum(i,j) = sum(i,j) + term(i,j)
enddo
enddo

do 31 i = 1,n
do 32 j = 1,n
if (sum(i,j).eq.0.0) then
goto 32
```

```
      endif
      qw = term(i,j)/sum(i,j)
      if (qw.gt.terr) then
      goto 56
      endif

32    enddo
31    enddo
      goto 57

56    continue
57    print*, 'Satisfactory convergence achieved'

      do 23 id = 1, iz
      do i = 1,n
      do j = 1,n
      a(i,j) = 0.0
      do k = 1,n
      a(i,j) = a(i,j) + sum(i,k)*sum(k,j)
      enddo
      enddo
      enddo

      do j = 1,n
      sum(i,j) = a(i,j)
```

```
        enddo
        enddo
23      continue

        do i = 1,n
        xt(i) = 0.0
        do k = 1,n
        xt(i) = xt(i) + sum(i,k)*xo(k)
        enddo
        enddo

        print*, 'Comp    initial    amount    Time'
        print 125, 'Comp, initial, amount, t'
125     format (io, f3.2, f3.9, f3.8, e8.2)

        sumx = 0
        do i = 1, n
        print*, i, xo(i), xt(i), t
        sumxo =  sumxo + xo(i)
        enddo

        sum1 = 0
        do i = 1,n
        sum1 = sum1 + xt(i)
        enddo
```

```
      if (d.gt.1000) then
      resum = 0
      else
      sumt = sumxo*exp(-d)
      resum = abs(sum1 - sumt)/sumt
      endif
      t  = t*step
      if (t.gt.tf) goto 42
      goto 17
42    stop
      end
```

APPENDIX B

PROGRAM FOR SYNTHETIC DATA GENERATION

This program generates synthetic data of PET scans for training the Neural network models [19]. The algorithm is described in detail in Chapter 4.

```
      dimension y(50)
c     This program generates positron emission tomography synthetic data
c       a for the following range of parameters
c       f: 0.1 to 6.0
c       a: 0.1 to 0.3  (used 0.2 for the first test case)
c       b: 0.1 to 0.2  (used 0.14 for the first test case)
c       spf: 0.1 to 1.0
        tmax = 120.0
        tdly  = 0.20
        npts = 12
        a = 0.45
        b = 0.2
        v1 = 0.8
        nfp = 60
        spf = 0.4
```

```
      icall = 1
      f = 7.0 / 60.0
      call qtot (f, a, b, spf, v1, tdly, c2scl, dt, tmax, npts, icall, y)
      f=0.20
      do 20 k = 1, 60
      f = f + 0.15
      p = f * (1 - 0.607 * exp (-1.25/f))
      spf=0.2
c     do 10 l = 1, 1
      do 10 l = 1, 1
      spf = spf + 0.1
      b = 0.12
      do 10 m = 1, 1
       b = b + 0.15
       icall = icall + 1
       fm = f / 60.0
       call  qtot(fm, a, b, spf, v1, tdly, c2scl, dt, tmax, npts, icall, y)
       fscl = (alog10(1.0+f)) ** 0.5
       fscl = f / 7.0
c      write (7, 1000) (y(j), j=1, npts)
```

```
      write (7, 1500) (y(j), j = 1, npts), p
c     write(7,2000) fscl
10    continue
20    continue
c1000 format (' ','(',12 (1x, e10.3))
1500  format (' ','(',12 (1x, e10.3), 1x, e10.3,' )')
c2000 format (' ',2(1x, e10.3,) ')')
2000  format (' ',1x,e10.3,' )')
      stop
      end
      subroutine
      qtot(f, a, b, spf, v1, tdly, c2scl, dt, tmax, npts, icall, y)
      dimension y(1)
      x1 = (1.65*exp(1.25/(60.0*f))-1.0)*f
      c = x1/v1
      x = (f+x1)/v1
c     calculation of q1 follows
      dt = tmax/npts
      tstrt = dt/2.0
```

```
      do 100 k = 1, npts
      tr = tstrt + (k - 1)*dt
      t = tr - tdly
      y(k)=0.0
      if (t.lt.0.0) then
      go to 100
      endif
      cx = f*a/(x - b)
      t1 = t*exp(- b * t)
      t2 = exp(- b * t)/(x - b)
      t3 = exp(- x * t)/(x - b)
      q1 = cx* (t1 - t2 + t3)
c     calculation of q2 follows
      cx = f * a * c/(x - b)
      t1 = (t * exp(- b * t) + exp(- b*  t)/b)/b
      t2 = exp(- b * t)/(b * (x - b))
      t3 = exp(-x * t)/(x * (x - b))
      c2 = - cx * (- 1.0/(b ** 2) + 1.0/(b * (x - b)) - 1.0/(x * (x - b)))
      if (icall. eq. 1)then
      c2scl = c2
```

```
      print *,  ' The value of c2scl is:      ', c2scl
      endif
      q2 = cx * (t2 - t1 - t3) + c2
c     the following  is for the input function
      ca = a * t * exp(- b * t)
c     the measured output is given by
      y(k) = q1 + q2 + spf * ca
      y(k) = y(k)/c2scl
100   continue
      return
      end
```

APPENDIX C

PROGRAM TO CALCULATE MOMENTS ABOUT THE MEAN OF THE DATA

```
      real*8 q(20), qm(10), qb, sum, sum1, t, t1

      open (5, file = 'sum.dat', status = 'old')
      open (6, file = 'out.dat', status = 'new')

      t = 10.0
      t1 = 120.0
      sum = 0.0
      sum1 = 0.0

      do k = 1, 120
      read (5, *) ( q(j), j = 1, 12)

      do j = 1, 12
      sum = sum + q(j)
      enddo

      qb = sum/(j-1)
      sum = 0.0

      do i = 1, 6
      do j = 1, 12
      sum1 = sum1 + ((1.0 - (q(j)/qb))**i) * t
      enddo

      qm(i)  = sum1/t1
      sum1 = 0.0
      enddo

      write (6, 10) (qm(i), i = 1, 5)
      enddo
10    format (3x, 5(f12.6))
      end
```

APPENDIX D

PROGRAM TO ADD PERTURBATION TO THE DATA

```
      real*8 q(20), qm(20)

      open (5, file = 'nick1.dat', status = 'old')
      open (7, file = 'nic.dat', status = 'new')


      do k = 1, 20
      read (5, *) (q(j), j = 1, 12)

      do j = 1, 12
      qm(j) = q(j) * 0.02 + q(j)
      enddo

      write (7, 10) (qm(i), i = 1, 12)
      enddo
10    format (2x, 12(f10.5))
      end
```

APPENDIX E

PROGRAM TO ADD NOISE IN THE DATA

```
      real q(20), qmn(20), qm(20), sum, Ga, y, y1, percent

      open (20, file = 'test.dat', status = 'old')
      open (21, file = 'out.dat', status = 'new')

      sum = 0.0
      t = 10.0
      ii = 999999999

      print*, 'Enter the amount of Noise Percent'
      read *, percent

      do k = 1, 20
      read (20, *) (q(j), j = 1, 12)

      do j = 1, 12
      sum = sum + q(j)
      enddo

      qm(i) = abs(sum * t)
      sum = 0.0
      y = ran(i)
      Ga = sqrt(-2.0 * alog(y)) * cos(6.28318 * y1)
      write (6, *) y, Ga
      y1 = y
      qmn(i) = qm(i) * (1 + percent * Ga/100)
      write (21, 10) (qmn(i), i = 1, 12)
      enddo
10    format (2x, 12(f10.5))
      end
```

APPENDIX F

PROGRAM TO NORMALIZE THE DATA

```
      real*8 q(20), qm(10), qb, sum, sum1

      open (5, file = 'test.dat', status = 'old')
      open (6, file = 'out.dat', status = 'new')

      t = 10.0
      t1 = 1.0
      sum = 0.0
      sum1 = 0.0

      do k = 1, 20
      read (5, *) (q(j), j = 1, 12)

      do j = 1, 12
      sum = sum + q(j)
      enddo

      qb = sum * t
      sum = 0.0

      do i = 1, 12
      do j = 1, 12
      sum1 = q(j)/qb
      enddo

      qm(i) = abs(sum1 * t1)
      sum1 = 0.0
      enddo

      write (6, 10) (qm(i), i = 1, 12)
      enddo
10    format (2x, 12(f8.5))
      end
```

APPENDIX G

Trapezoidal rule method:

Numerical integration is the process of calculating the value of definite integral $\int_a^b f(x)dx$ from the tabulated values of f(x). Suppose the values of y = f(x) tabulated for x = a, a + h, ... a + nh are $y_0, y_1, ...y_n$ respectively, then

$$\int_a^b f(x)dx = \int_a^{a+h} f(x)dx + \int_{a+h}^{a+2h} f(x)dx + ... \int_{a+(n-1)h}^{b=a+nh} f(x)dx .....(A)$$

Now $\int_a^{a+h} f(x)dx = h\int_0^1 f(a+rh)dr$ where x = a + rh, dx = hdr.

$$= \int_0^1 [y_o + r\Delta y_o + \frac{r(r-1)}{2!}\Delta^2 y_o + ...]dr$$

$$= [y_o + \frac{1}{2}\Delta y_o + \frac{1}{3.2!}\Delta^2 y_o + ...]$$

$$= h[y_o + \frac{1}{2}(y_1 - y_o)], \text{ to the first order in } \Delta.$$

$$= \frac{h}{2}(y_o - y_1) \text{ approximately.}$$

Similarly, $\int_{a+h}^{a+2h} f(x)dx = \frac{h}{2}(y_1 + y_2)$ nearly, and so on.

Finally,

$$\int_{a+(n-1)h}^{a+nh} f(x)dx = \frac{h}{2}(y_{n-1} + y_n) \text{ nearly.}$$

Substituting all these values in (A) we get

$$\int_a^b f(x)dx = h[\frac{1}{2}(y_o + y_n) + (y_1 + y_2 + ... + y_{n-1})]$$

which is known as the trapezoidal rule.

APPENDIX H

PROGRAM TO READ BINARY DATA FILE AND CONVERT IT TO ASCII READABLE FORMAT

This program reads the binary form of data in the files specified and converts it into ascii readable format. This program was employed for reading the PET scan data which is usually in the binary form.

```
c       integer*1 logx(400000), nl, nr
        integer*2 logx(200000),nl, nr
        integer*2 n_out(20)
        integer*2 l_out(20)
        integer*2 id2l, id2r, idc, id2x, id2y
        integer imx(20), isha(20,20), ishl(20,20), ishc(20,20)
        dimension a_out(20), dmx(20), dvscl(20)
        character*14 fn_out, fn_in, fn_scl, fn_ckx
        write(*,'(A\)')' Enter input file name '
        read(*,'(A14)')fn_in
        write(*,'(A\)')' Enter file name of scaling data '
        read(*,'(A14)')fn_scl
        read(*,'(A14)')fn_ckx
        n_shft=0
```

```
      write(*,'(A\)')' Enter first and last row to print '
      read(*,*)ir1, ir2
      write(*,'(A\)')' Enter first and last column to print '
      read(*,*)ic1, ic2
      ip1=1
      ip2=12
      open(unit = 1, file = fn_in, form = 'binary')
      read(1, end = 10)logx
10    continue
      write(*,'(A\)')' Enter output file name '
      read(*,'(A14)') fn_out
      open(unit = 2, file = fn_out, form = 'formatted')
      open(unit = 3, file = fn_scl, form = 'formatted')
      open(unit = 4, file = fn_ckx, form = 'formatted')
      read(3,*)(dmx(k),  k = 1,12)
      write(4, 1712)(dmx(k), k=1,12)
1712  format('  dmx(-) = ',12(1x,e10.3))
      call scl_fac(logx, imx, dmx, dvscl, isha, ishc, ish)
      write(4,1714)(imx(k), k=1,12)
1714  format(' imx(-) = ',12 (1x,i5))
```

```
      write(4,1716)(dvscl(k), k=1,12)
1716  format('  dvscl(-) = ',12(1x,e10.3))
      npix = 20
      ioff = 128*128
c     ioff = ioff*2
      npix = 20
      iprt = 0
      irx = ir1
      do 20 ir = ir1, ir2
      icx = ic1
      do 20 ic = ic1, ic2
c     ioff = (ir - 1)*256 + ic*2-1
      ixxf = (ir - 1) * 128 + ic
      ipx = ip1
      do 15 k = ip1, ip2
      iprt = iprt + 1
      isel = ixxf + (k - 1) * ioff
      il = isel + n_shft
      nr = logx(il + 1)
```

```
nl = logx(il)
id2l = 0
id2r = 0
id2l = nl
id2r = nr
id2x = 0
id2x = isha(id2l, 8)
idc = ior(id2x, id2r)
n_out(iprt) = idc
n_out(iprt) = nl
n_out(iprt) = float(nl) * 256.0/32768.0
id2y = ishl(id2l, 8)
idc = ioer(id2x, id2r)
idc = 256 * nl + nr
idc = ishc(nl, 8)
l_out(iprt) = idc
rprxt = float(idc)
a_out(iprt) = rprxt * dvxcl(k)
if(iprt.eq.12)then
write(2, 1400)(a_out(l),l=1,12)
```

```
      loc = il-7
      irx = ir
      icx = ic
      ipx = k
      iprt = 0
      endif
15    continue
20    continue
1000 format(' ',12(1x,I6))
1200 format(' loc ',3i4,' = '8(1x,i6))
1100 format(' ')
1400 format(' ( ',12(1x,e11.4),' )',)
      stop
      end
      subroutine scl_fac(logx, imx, dmx, dvscl, ishl, isha, ishc)
      integer * 2 logx(1)
      integer imx(1)
      dimension dmx(1), dvscl(1)
      ioff = 128 * 128
```

```
      do 20 j = 1,12
      imx(j)=0
      do 10 k = 1, ioff
      loc = k + (j - 1) * ioff
      ivx = logx(loc)
      if (ivx. gt. imx(j))then
      imx(j) = ivx
      endif
10    continue
20    continue
      do 30 l = 1,12
      dvscl(l) = dmx(l) / float(imx(l))
30    continue
      return
      end
```

Vita

Nithyananda G. Narayana was born in Bangalore, India, on the September 10, 1967. He attended elementary and high school in his native state, Karnataka, and graduated from high school in May 1986. He represented his school in cricket.

In 1986, he enrolled in Bangalore Institute of Technology, Bangalore, India. In July 1990, he received the degree of Bachelor of Engineering with a major in Instrumentation/Electronic Engineering. During October 1990 to October 1991, he worked as a Graduate Engineer Trainee in KEB, a state government electricity board, India. In the spring of 1992, the author enrolled in the Masters degree program in Nuclear Engineering at the University of Tennessee, Knoxville. The degree was awarded in August 1995 with specialization in Radiation Protection Engineering.