Using Artificial Neural Networks Models for Predicting Pancreas Behavior

Bassam Abdul-Rahman Abed

Supervised BY
DR. Rashid Al-Zubaidy

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Deanship of Academic Research and Graduate Studies
Philadelphia University

June 2008

To My Family ...

To My Friends...

I thank Allah so much for finishing this work and I pray for him that he will give me what I deserve. Whatever, I will be thankful and satisfied.

I would like to thank all my friends who are always ready to offer their help and encouragements.

I would like to dedicate this work to my parents,
Who respected my decisions, helped me, supported me and gave me the trust and love. I don’t show them the love which they deserve, so thanks and I love you.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Associative Prof. Rashid Al-Zubaidy for his guidance and help. Also I would like to thank Assistant Prof. Ramzi Alsona'. I would like to thank the Philadelphia university administration for their support and help in the past two years. Specially, President of Philadelphia University, Dean of Scientific Research, Dean of IT Faculty, and Chief of Computer Science Division.

I would like to thank every professor who taught me in the past two years. I would like to thank Prof. Raed Abu zaiter and assistant Prof. Issa Ish'habat for their help.

Abstract

Artificial Neural Network (ANN) is currently a 'hot' research area in medicine and this work is based on predicting the behavior of an organ of a human body called pancreas by using neural networks. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Their ability to learn by example makes them very flexible and powerful. So we have data set from 102 of three different groups of subjects. From data set examples, neural networks were learned by using two algorithms, Radial Basis Function (RBF) and General regression Neural Network (GRNN). After learning, a simulation (testing) for learned data has been made and a comparison between both algorithms has been done subject to the learning performance. Furthermore a comparison between RBF and GRNN algorithms was done based on the ability of both networks to generalize input data not seen before.

All simulations were made by MATLAB 7 neural network toolbox and the results showed that RBF and GRNN are good function approximators. It was noticeable that RBF and GRNN were fast training algorithms even GRNN was faster. It was apparent that neural network was a good choice for predicting a behavior of nonlinear and complex systems such as pancreas. Also the results showed that the performance of RBF in learning was better than GRNN, and the ability of GRNN in generalization or testing was better than RBF.

Table of Contents

Chapter 1 Introduction

Chapter 2 A literature review of neural network algorithms used for regulating glucose level

Chapter 3 The Pancreas And It‘s Endocrine System

Chapter 4 Artificial neural networks algorithms for function approximation

Chapter 5 design and implementation

Chapter 6 Results

Chapter 7 Conclusions and Future Work

Reference

List of Figures

List of Tables

List of Abbreviations

NO	ABBREVIATION	STANDS FOR
1	ANN	Artificial Neural Network
2	BMI	Body Mass Index
3	FFNN	Feed Forward Neural Network
4	GRNN	General Regression Neural Network
5	IGT	Impaired Glucose Tolerance
6	LM	Levenberg- Marquardt
7	MLP	Multilayer Perceptron
8	NGT	Normal Glucose Tolerance
9	PID	Pima-Indian Dataset
10	PN	Polynomial Network
11	RBF	Radial Basis Function
12	RPROP	Resilient Propagation
13	T2DM	Type 2 Diabetes Mellitus

حزيران 2008

Philadelphia University
June, 2008

Successfully defended and approved on ……………….

<u>Examination Committee   signature</u>

DR. …………………….., Chairman.

Academic Rank …………………………

DR. ……………………….., Member.

Academic Rank …………………………

DR. ……………………….., Member.

Academic Rank …………………………

DR. ……………….., External Member.

Academic Rank …………………………

جامعة فيلادلفيا
نموذج تفويض

انا بسام عبدالرحمن عابد أفوض جامعة فيلادلفيا بتزويد نسخ من رسالتي للمكتبات أو
المؤسسات أو الهيئات او الاشخاص عند طلبها .

التوقيع :
التاريخ:

Philadelphia University
Authorization Form

I, Bassam Abdul-Rahman Abed, authorize Philadelphia University to supply copies of my thesis to libraries or establishment or individuals upon request.

Signature :
Date

Chapter 1

Introduction

1.1 Motivation

Diabetes is a disease that is characterized by an elevated blood Glucose level. Diabetes can be caused by a reduction of the production of Insulin by the pancreas (type I diabetes) or by the Insulin being less effective at moving Glucose out of the blood stream and into cells that need it (type II diabetes). A blood Glucose level that is elevated for a long period of time can result in vascular, neurological or metabolic complications, such as kidney failure, blindness and an increased chance of heart attacks [26].

Diabetes is a widespread chronic illness that accounts for a large part of the health care budget. Diabetes affects approximately 20% of Jordanian people according to statistics made by Jordanian diabetes national center [27]. In USA about 20.8 million people, approximately 7.0% of the population of USA have diabetes according to statistics made in 2005. Diabetes was the sixth leading cause of death listed on U.S. death certificates in 2002 [26].

The function of the pancreas which secretes hormones that regulate blood Glucose levels is very important to prevent diseases such as diabetes. So to study the pancreas model and apply it to the neural network models is strong motivation for this thesis.

Neural networks are widely used in the field of health care for a range of different purposes. Also in the field of diabetic treatment neural networks can be found, for example for diagnosing diabetes, selecting a proper treatment based on a number of variables or simply for supporting the decision making process of a physician. But neural networks are becoming more common in diabetic management as well. Insulin pumps for example can be equipped with an Insulin dosage controller that is driven by a model based on a neural network.

Another motivation for this work is to study some of Artificial Neural Network (ANN) algorithms and to evaluate their application on nonlinear and complex system such as pancreas, and to compare performances of theses algorithms with respect to pancreas.

1.2 Objectives

The primary objective of this research is to study artificial neural network algorithms for function approximation. More specifically, the objective of the work in this thesis can be summarized as:

- To study the models of the radial basis function neural network (RBFNN) and general regression neural network (GRNN) and apply them to predict the behavior of the pancreas, GRNN and RBF are known as good function approximators, fast training algorithms, and capable of modeling any nonlinear and complex systems efficiently with no need for a priori knowledge about the function of the system.

- To evaluate the performance of neural network in complex and nonlinear systems such as pancreas.
- To compare the performance of radial basis function and general regression neural network with respect to pancreas model.
- Neural network algorithms, RBF and GRNN, will be applied to three different models according to the types of subjects in the training set used in this study, these groups are: a group of patients with Type 2 Diabetes Mellitus (T2DM), a group of people with Normal Glucose tolerance (NGT), and a group of people with Impaired Glucose Tolerance (IGT).
- Learning and testing the neural network algorithms will be taken place in the data presented in this study. And the neural network algorithms will be compared subject to learning and testing.

1.3 Major Contributions of the thesis

Resembling the interactions between Glucose concentration levels and amount of Insulin injected in the bodies of diabetics [24], classifying a patient as diabetic or not subject to risk factors such as age, gender, family history of diabetes, body mass index (BMI)… etc [23], modeling Insulin dosage controller in artificial pancreas, modeling the blood Glucose metabolism of a diabetic [22], diagnosing diabetes [8] , modeling of the Glucose-Insulin dynamics of diabetic [12] , and prediction of patients with acute pancreatitis [9] were made by different neural network algorithms in past. Also comparisons between the different neural network models have been made, such as multilayer perceptron (MLP), feedforward backpropagation (FFBP), radial basis function neural network (RBFNN), general regression neural network (GRNN), Polynomial neural network (PNN) and others.

Major contribution of this thesis can be summarized as follows:

Studying neural network models and apply them to predict the behavior of the pancreas is the main contribution of this study. There has been much publicity about the ability of artificial neural networks to learn and generalize, Learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data while generalization (i.e., response of the network to input data not seen before) is equivalent to the use of this multidimensional surface to interpolate the test data, so the neural network will be learned to approximate a function equivalent to the target function which is pancreas, when new data comes from outside, this neural network must has the ability to interpolate or test data. RBF networks are applied to many kinds of problems include function approximation, data classification, prediction, and data clustering. GRNN is used for function approximation (regression).

In this study we will use both GRNN and RBFNN to approximate the function of pancreas. Radial basis function networks are non-parametric models. By non-parametric models, it means that there is no a priori knowledge about the function that is to be used to fit the training set. And this is applied to pancreas; we don’t have a priori knowledge about the relations between the inputs to pancreas and the output.

Another contribution of this study is to do comparison between radial basis function and general regression neural networks, in this work a comparison will be made not just for the results but intensive study will be made according to function approximation capability and generalization ability for each algorithm.

1.4 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 provides a literature review of using neural network algorithms to regulate the Glucose levels in the blood, many algorithms were used such as multilayer perceptron (MLP), Feedforward Backpropagation (FFBP), radial basis function neural network (RBFNN), general regression neural network (GRNN), Polynomial neural network (PNN).

Chapter 3 provides a brief about the pancreas and how the pancreas is working.

In chapter 4, artificial neural networks is briefly introduced, then radial basis function neural network is introduced and compared with other neural network algorithms, finally general regression neural network is introduced.

Chapter 5 presents designing the RBF and GRNN models for pancreas, the architecture of both networks will be introduced, values of neural network parameters such as number of neurons and spread values are determined, method of designing the models is presented, and implementing the neural network algorithms, radial basis and general regression neural algorithms to model the pancreas will be presented.

In chapter 6 results are shown, and a comparison between the two models RBF and GRNN based on their performance is made. Finally, Conclusion and future work are summarized in Chapter 7.

Chapter Two

A literature review of neural network algorithms used for regulating Glucose

2.1 Classification patients with pancreas's diseases using ANN

Performance of the RBF neural network was compared with Multilayer Perceptron (MLP) network model and the classical logistic regression on diabetes database [23]. The risk factors considered for analysis are age, gender, family history of diabetes, body mass index (BMI), total cholesterol level (TC), triglycerides (TG), low density lipids (LDL) and high-density lipids (HDL). The efficiency of the constructed models was evaluated by comparing the sensitivity, specificity and overall correct predictions for datasets. The results indicate that the RBF network has a better performance than other models. The sensitivity and specificity of both neural network models had a better predictive power compared to logistic regression. Even when compared on an external dataset, the neural network models performed better than the logistic regression. When comparing, RBF and MLP network models, [23] found that RBF had better performance in testing and external datasets. [23] Indicated the good predictive capabilities of RBF neural network. Also the time taken by RBF is less than that of MLP in the application.

The performance of General Regression Neural Network (GRNN) was examined on the Pima Indian Diabetes (PID) data set in [8]. The performance of the standard Multilayer Perceptron (MLP) and radial basis function (RBF) feed forward neural networks were also examined for the comparison. All patients in PID data set in [8] were Pima-Indian women at least 21 years old. The binary response variable takes the values '0' or '1', where '1' means a positive test for diabetes and '0' is a negative test for diabetes. There were eight clinical findings: 1) Number of times pregnant. 2) Plasma Glucose concentration a 2 hours in an oral Glucose tolerance test. 3) Diastolic blood pressure. 4) Triceps skin fold thickness. 5) 2-Hour serum Insulin. 6) Body mass index. 7) Diabetes pedigree function. 8) Age. The performances of the MLP, RBF and GRNN structures in are given in Table (2.1).

Table (2.1) The performances of the MLP, RBF and GRNN in this study.

ALGORITHMS	TRAINING SET	TEST DATA	mean total correct prediction
MLP	88.19 %	77.08 %	85.41%
RBF	100 %	68.23 %	92.06%
GRNN	82.99 %	80.21 %	82.29%

The performance of RBF was worse than the MLP for data testing. The best result achieved on the test data was the one using the GRNN structure (80.21%). Results showed that, general regression neural network (GRNN) could be a good and practical choice to classify a medical data.

Evaluation of the ability of an artificial neural network (ANN) that uses radiologic and laboratory data to predict the outcome in patients with acute pancreatitis was studied by [9]. The ANN was trained and tested by using a round-robin technique, and the performance of the ANN was compared with that of linear discriminate analysis and Ranson and Balthazar grading systems. A back-propagation ANN was developed with the six most statistically significant parameters (blood pressure, extent of inflammation, fluid aspiration, serum creatinine level, serum calcium level, and the presence of concurrent severe illness). Performance of the ANN was statistically significantly better than the Ranson and Balthazar. The six findings described earlier were the six input nodes, and the positive or negative patient outcome was the output node.

2.2 Predicting blood Glucose levels using ANN

Predicting the time course of blood Glucose levels from the complex interaction of Glucose counterregulatory (Glucose rising) hormones and Insulin was demonstrated by feedforward neural network in study by [14]. The system consisted of the activation of hormones such as Glucagon, catecholamines (Epinephrine, Norepinephrine), Growth Hormone (GH), and Cortisol upon low concentrations of blood Glucose (hypoglycemia). Weights were updated using Resilient Propagation (RProp), the time course of blood Glucose levels demonstrated the best predictability using RProp as the training algorithm compared with Backpropagation algorithm.

2.3 Modeling Glucose-Insulin by ANN

Two types of neural networks (NN's) were experimented in building the model of the interactions between Glucose concentration levels and amount of Insulin injected in the bodies of diabetics. Comparisons between Levenberg- Marquardt (LM) training algorithm of multilayer feed forward neural network (FFNN) and Polynomial Network (PN's) were done. The components of a training vector were the present Glucose level PGL , short term Insulin STI , midterm Insulin MTI, time period, and meal. The single output of the model has a target of the next Glucose level NGL. PN's model could not learn to predict correctly the next values of Glucose levels (NGL). PN's are only good "mappers". The results demonstrated the ability of the Levenberg-Marquardt (LM) NN to model the whole set of data [24].

2.4 Modeling Glucose metabolism by ANN.

The application of neural networks to modeling the blood Glucose metabolism of a diabetic was studied. In particular, recurrent neural networks and time series convolution neural networks were considered, and then compared to linear models and to nonlinear compartment models [22].

The data set consisted of the protocol of a male type I diabetic patient over a period of 63 days. During that time period, times and dosages of Insulin injections (basal Insulin and normal Insulin), the times and amounts of food intake (fast, intermediate, and slow carbohydrates), the times and durations of exercise (regular or intense) and the blood Glucose level was the output node. A Feedforward Multi-layer Perceptron (MLP) was used in a recurrent fashion since previous predictions were used as inputs. Experiments showed that the inclusion of a proper error model improves performance considerably.

In combination with the linear error model the recurrent neural network is a powerful model for blood Glucose prediction and gave best results and outperformed both a compartment model and the time series convolution neural network approach.

Chapter 3

The Pancreas And It‘s Endocrine System

3.1 Pancreas

The pancreas is a gland organ in the digestive and endocrine systems of human. It is both exocrine (secreting pancreatic juice containing digestive enzymes) and endocrine (producing several important hormones, including Insulin, Glucagon). In humans, the pancreas is a 15-25 cm (6-10 inch). It weighs between 65g - 75g. One of the organs behind the abdominal cavity, it is located posterior to the stomach and in close association with the duodenum. And it is often described as having mainly three regions: a head, body and tail. See figure (3.1) and (3.2) respectively [29].

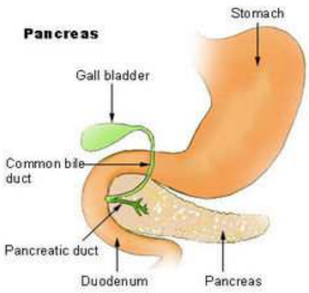

Figure 3.1 location of the pancreas in human body

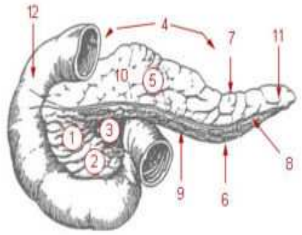

Figure 3.2 regions of pancreas

1: Head of pancreas
4: Body of pancreas, 11: Tail of pancreas

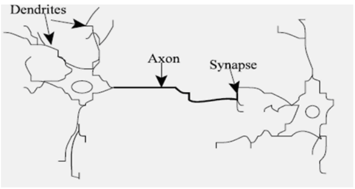

Figure 3.3: biological neuron

Here we can find similarity between the shape of the pancreas and the biological neuron see figure (3.3) above, the neuron is a simple processing unit that receives and combines signals from many other neurons through filamentary input paths, the dendrites. Dendrites are connected with the main body of the nerve cell, the soma. And the neuron ends with axon (the tail) [15].

3.2 Endocrine pancreas

Under a microscope, when properly stained, it is easy to distinguish two different tissue types in the pancreas. As seen in table (3.1). These regions correspond to the main pancreatic functions [29]:

Table 3.1 regions of pancreas correspond to the main pancreatic functions

Appearance	Region	Function
light staining circles (islets of Langerhans)	Endocrine pancreas	secretes hormones that regulate blood Glucose levels
darker surrounding tissue	Exocrine pancreas	produces enzymes that break down digestible foods

Our work will be using the artificial neural networks models for endocrine pancreas. Because the main function of islets of langerhans is regulating the level of Glucose in the blood. There are four main types of cells in the islets of Langerhans. They are relatively difficult to distinguish using standard staining techniques, but they can be classified by their secretion as shown in table (3.2):

Table 3.2 types of cells in the islets of langerhans classified by their secretion

Name of cells	Endocrine product	% of islet cells	Representative function
beta cells	Insulin and Amylin	50-80%	lower blood sugar
alpha cells	Glucagon	15-20%	Raise blood sugar
delta cells	Somatostatin	3-10%	inhibit endocrine pancreas
PP cells	Pancreatic polypeptide	1%	inhibit exocrine pancreas

The endocrine system is an information signaling system much like the nervous system. However, the nervous system uses nerves to conduct information, whereas the endocrine system uses blood vessels as information channels. Glands located in many regions of the body release into the bloodstream specific chemical messengers called hormones, which regulate the many and varied functions of an organism [29].

Feedback is both a mechanism, process and signal that is looped back to control a system within itself. This loop is called the feedback loop. A control system usually has input and output to the system; when the output of the system is fed back into the system as part of its input, it is called the "feedback”. Feedback and regulation are self related. The negative feedback helps to maintain stability in a system in spite of external changes. It is related to homeostasis. Negative feedback (shortened to NFB) is a type of feedback in which the system responds in an opposite direction to the perturbation. It is a process of feeding back to the input a part of a system's output, so as to reverse the direction of change of the output. This tends to keep the output from changing, so it is stabilizing and attempts to maintain constant conditions. This often results in homeostasis (in biology) such that the system will return to its original set point automatically. Examples of this are numerous, from the regulating of body temperature, to the regulating of blood Glucose levels. The disruption of negative feedback can lead to undesirable results: in the case of blood Glucose levels, if negative feedback fails, the Glucose levels in the blood may begin to rise dramatically, thus resulting in Diabetes [25]. The principle of negative feedback control is illustrated by the diagram in figure (3.4) below:

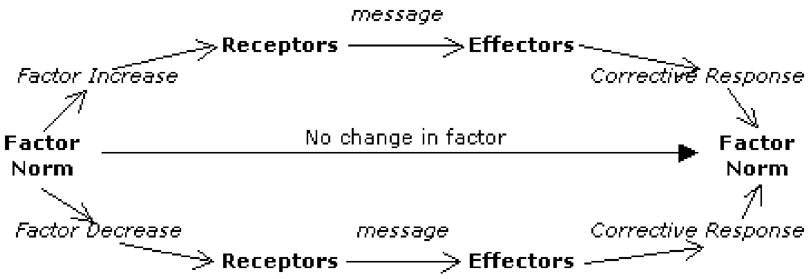

Figure 3.4 mechanism of negative feedback

The pancreas also works in the same mechanism of the negative feedback. The receptors of the pancreas are responsible for monitoring Glucose levels in the blood. Two types of cells release two different hormones from the pancreas, Insulin and Glucagon, these hormones target the liver, one or the other depending on the Glucose concentration. In cases where Glucose levels increase, less Glucagon and more Insulin is released by the pancreas and targets the liver. In cases where Glucose levels decrease, less Insulin and more Glucagon is released by the pancreas and targets the liver (see figure 3.5).

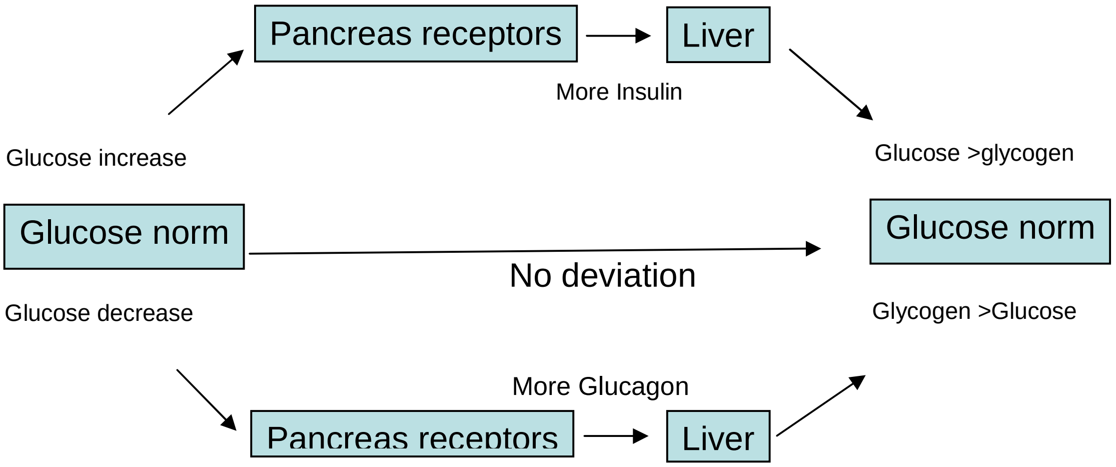

Figure 3.5 mechanism of regulating blood sugar

3.3 Pancreas features

After studying pancreas, and how pancreas is working, we conclude that these features are the most important features in pancreas, and we will use them as input and output to pancreas, and next is a summary for these features.

1. **Glucose:**

Glucose is a simple sugar that serves as the main source of energy for the body. The carbohydrates we eat are broken down into Glucose (and a few other simple sugars), absorbed by the small intestine, and circulated throughout the body. Most of the body's cells require Glucose for energy production; brain and nervous system cells not only rely on Glucose for energy, they can only function when Glucose levels in the blood remain above a certain level [28].

2. Insulin and C-peptide:

The synthesis of Insulin begins at the translation of the Insulin gene, which resides on chromosome 11. This primary translation product is called preproInsulin, once the preproInsulin reaches the endoplasmic reticulum, proInsulin is created. ProInsulin consists of three domains: an amino-terminal B chain, a carboxyl-terminal A chain, and a connecting peptide in the middle known as the C-peptide. Within the endoplasmic reticulum, proInsulin is exposed to several specific peptidases that remove the C-peptide and generate the mature and active form of Insulin [28].

3. Glucagon:

pancreatic hormone released by alpha cells of islands of langerhans which is secreted to tell the liver to turn some glycogen back into Glucose when blood Glucose level is below the normal level, leads in raising the blood Glucose levels [27].

4. Glucagon-like peptide-1 (GLP-1):

Glucagon-Like Peptide-1 (GLP-1) is derived from the transcription product of the proGlucagon gene. The major source of GLP-1 in the body is the intestinal L cell that secretes GLP-1 as a gut hormone [29]. The known physiological functions of GLP-1 include:

- Increases Insulin secretion from the pancreas in a Glucose-dependent manner.
- Decreases Glucagon secretion from the pancreas.

5. Glucose-dependent Insulinotropic Peptide (GIP):

Glucose-dependent Insulinotropic Peptide (GIP) is a member of the secretin family of hormones. GIP, along with Glucagon-like peptide-1 (GLP-1), belong to a class of molecules referred to as incretins. [19] It is now believed that the function of GIP is to induce Insulin secretion. Incretins are a type of gastrointestinal hormone that cause an increase in the amount of

Insulin released from the beta cells of the islets of Langerhans after eating, even before blood Glucose levels become elevated. As expected, they also inhibit Glucagon release from the alpha cells of the Islets of Langerhans. The two main candidate molecules that fulfill criteria for an incretin are Glucagon-like peptide-1 (GLP-1) and. Glucose-dependent Insuliniotropic Peptide (GIP) [6].

Figure (3.6) summarize how the endocrine pancreas and incretins hormones regulating the Glucose level.

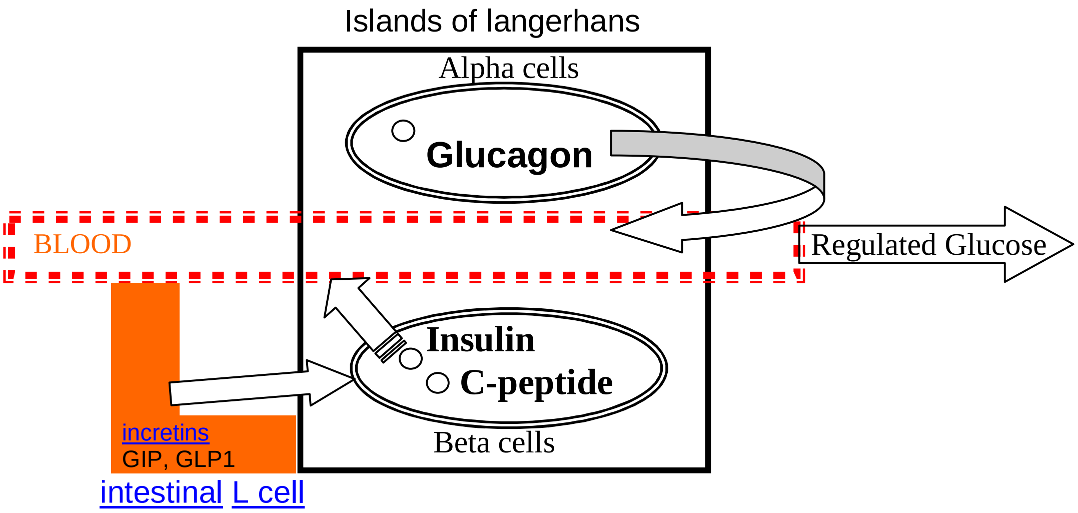

Figure 3.6 how blood Glucose is regulated by pancreas

As seen in the figure 3.6, pancreas has endocrine system called islands of Langerhans, they consist two types of cells called Alpha cells and Beta cells. Alpha cells are producing Glucagon. Beta cells are producing insulin and C-peptide. They have been discussed before. Other hormones are secreted by Intestinal L cell called GIP and GLP-1. so we will consider these hormones as input features for pancreas. And these hormones together will regulate the level of Glucose, so the output of the pancreas will be regulated Glucose.

Chapter 4

Artificial Neural Networks Algorithms for Approximating Function

4.1 Artificial Neural Networks

One type of network sees the nodes as ‘artificial neurons’. These are called artificial neural networks (ANNs). An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the *axon*. This signal might be sent to another synapse, and might activate other neurons. The complexity of real neurons is highly abstracted when modeling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron. Another function (which may be the identity) computes the output of the artificial neuron (sometimes in dependence of a certain threshold) [15]. ANNs combine artificial neurons in order to process information see figure (4.1).

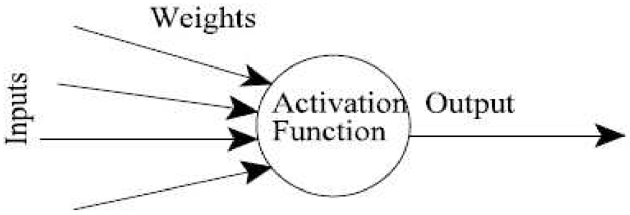

Figure 4.1 an artificial neuron

The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training. The number of types of ANNs and their uses is very high. The differences in them might be the functions, the accepted values, , the learning algorithms and the topology (figure 4.2) the artificial neuron in figure (a) where θ is the activation function and X1 to X4 are the input vectors and w1 to w4 are the weights and o is the output, in figure (b) the multilayered artificial neural network. [15].

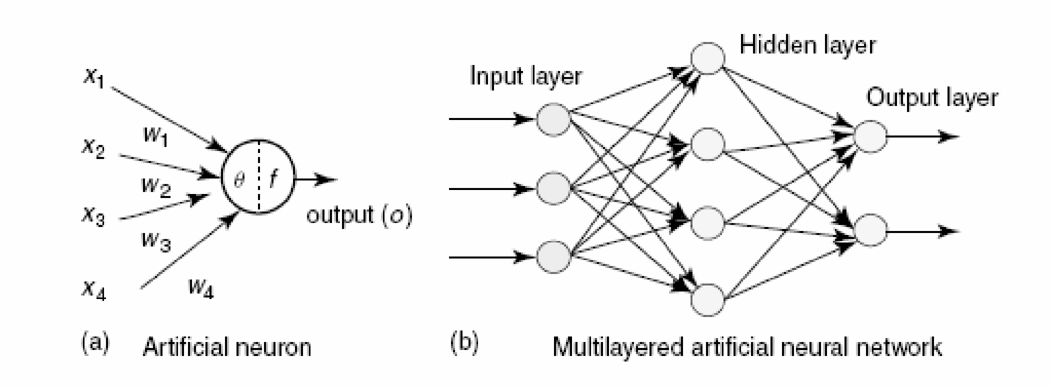

Figure 4.2 Architecture of an artificial neuron and a multilayered neural network

The learning in neural networks may be classified mainly into two sorts. These are supervised learning and unsupervised learning. In supervised learning, an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done, and the errors or discrepancies between the desired and actual response for each node in the output layer are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The term supervised originates from the fact that the desired signals on individual output nodes are provided by an external teacher [15].

The best-known examples of this technique occur in the backpropagation algorithm, the delta rule, and the perceptron rule. In unsupervised learning (or self-organization), a (output) unit is trained to respond to clusters of pattern within the input. In this paradigm, the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather, the system must develop its own representation of the input stimuli [15].

4.2. Radial basis function neural network

The radial basis function neural networks are originally motivated by the locally tuned response in biological neurons. Their activation functions of the hidden neurons are determined by the distances between the input vector and prototype vectors. RBF neural network has gained much popularity in the past decades due to its fast training and its universal approximation capability with local responses, which can approximate any continuous functions with arbitrary precisions. Usually the RBF network contains three layers: the input layer, the hidden layer and the output layer. Input attributes are fed into the hidden layer linearly with a unit weight through the input layer. The hidden units provide a nonlinear transformation by a set of radial basis functions that constitutes the basis for the input when they are mapped into the space of the hidden neurons. The output layer of RBF network only computes the linear combination of the outputs from the hidden layers [17].

There are many types of radial basis functions such as Thin-Plate-Spline function, Multiquadric function, Inverse Multiquadric function, Gaussian function. Among these types of radial basis functions, Gaussian function is the most widely used in RBF networks. It is found to be capable of making an accurate global mapping with refined local details. Compared with other radial basis functions, Gaussian function has several advantages. First, the values of Gaussian function decrease monotonically with the growth of distance from the center, which makes the Gaussian function local in its response. It is more plausible from the biological point of view, because the response is finite. Besides, both the position and shape of Gaussian function is more flexible to adjust compared with other radial basis functions [17].

Gaussian transfer function gives a response that drops off rapidly as the distance between the hidden unit and the input vector increases and is symmetrical about the radial axis hence the name Radial Basis Function. The rate with which the response drops is determined by the “spread” of the hidden unit. The challenge of designing an RBF network lies in properly placing hidden layer neurons and choosing an optimal value for the spread constant such that the entire input space of interest is covered with minimum overlap. These decisions are usually made empirically, rather than through automatic training methods [7]. In [11] the learned hypothesis is a function of the form Showed in equation (4.1).

$$\hat{f}(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x)) \qquad (4.1)$$

Where each x_u is an instance from X and where the kernel function $K_u(d\ (x_u,\ x))$ is defined so that it decreases as the distance $d\ (x_u,\ x))$ increases. Here k is a user provided constant that specifies the number of kernel functions to be included. Even though $f^{\wedge}(x)$ is a global approximation to $f\ (x)$, the contribution from each of the $K_u(d\ (x_u,\ x))$ terms is localized to a region nearby the point x_u. It is common to choose each function $d\ (x_u,\ x))$ to be a Gaussian function centered at the point x_u with some variance $\sigma^2{}_u$ as shown in equation (4.2)

$$K_u(d(x_u, x)) = e^{-\frac{1}{2\sigma_u^2} d^2(x_u, x)} \qquad (4.2)$$

The functional form of Equation (1) can approximate any function with arbitrarily small error, provided a sufficiently large number k of such Gaussian kernels and provided the width σ^2 of each kernel can be separately specified. The function given by Equation (4.1) can be viewed as describing a two layer network where the first layer of units computes the values of the various $K_u(d\ (x_u,\ x))$ and where the second layer computes a linear combination of these first-layer unit values. An example radial basis function (RBF) network is illustrated in Figure (4.3).

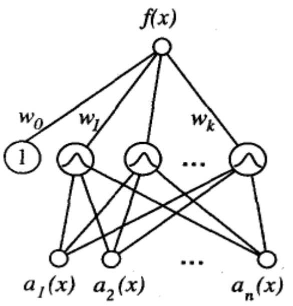

Figure 4.3 A radial basis function network. Each hidden unit produces an activation determined by a Gaussian function centered at some instance x_u. Therefore, its activation will be close to zero unless the input x is near x_u. The output unit produces a linear combination of the hidden unit activations. Although the network shown here has just one output, multiple output units can also be included.

Given a set of training examples of the target function, RBF networks are typically trained in a two-stage process. First, the number k of hidden units is determined and each hidden unit *u* is defined by choosing the values of x_u and $\sigma^2{}_u$ that define its kernel function $K_u(d\ (x_u, x)$. Second, the weights w_u are trained to maximize the fit of the network to the training data, using the global error criterion given by Equation (4.3):

$$E \equiv \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2 \qquad (4.3)$$

Because the kernel functions are held fixed during this second stage, the linear weight values w_u can be trained very efficiently [11].

Several alternative methods have been proposed for choosing an appropriate number of hidden units or, equivalently, kernel functions. One approach is to allocate a Gaussian kernel function for each training example $(x_i,\ f\ (x_i))$, centering this Gaussian at the point $x_{i.}$ Each of these kernels may be assigned the same width $\sigma^2{}_u$. Given this approach, the RBF network learns a global approximation to the target function in which each training example $(x_i,\ f\ (x_i))$ can influence the value of $f^{\wedge}$ only in the neighborhood of x_i**.** One advantage of this choice of kernel functions is that it allows the RBF network to fit the training data exactly. That is, for any set of *m* training examples the weights $w_0 \ldots w_m$, for combining the Gaussian kernel functions can be set so that $f^{\wedge}(\mathrm{x_i}) = f(\mathrm{x_i})$ for each training example $(x_i,\ f\ (x_i))$ [11].

A second approach is to choose a set of kernel functions that is smaller than the number of training examples. This approach can be much more efficient than the first approach, especially when the number of training examples is large. The set of kernel functions may be distributed with centers spaced uniformly throughout the instance space X. Alternatively, we may wish to distribute the centers nonuniformly, especially if the instances themselves are found to be distributed nonuniformly over X. In this later case, we can pick kernel function centers by randomly selecting a subset of the training instances, thereby sampling the underlying distribution of instances [11].

To train Gaussian RBF network, four types of parameters need to be searched. They are the number of hidden neurons, the positions of the centers of all the hidden neurons, the widths of all the hidden neurons and the weights that connect the hidden neurons and the output neurons. There are many algorithms being proposed for the training of Gaussian RBF networks and they can

summarized into a few categories according to their schemes to get these four parameters. The optimal weights between the hidden layer and the output layer can be obtained by many methods. One is the regulation method which determines the weights by matrix computation, another is gradient descent based method, least mean square is one of the mostly often used gradient descent methods, and Linear Least Square method to obtain the optimal weights between the hidden layer and the output layer. There are several ways to determine the center position and the width of the hidden neurons. One way is to select the centers of the hidden neurons randomly from input patterns. The width is set to a predefined value. In some other methods, the center and width for each hidden neuron are obtained using various clustering methods, such as learning vector quantization, the k-means clustering. The determination of the number of hidden neurons is the trickiest scheme of the training algorithm. Too small a number of hidden neurons does not allow for the reduction of the error to a satisfactory low level, while too high a number of hidden neurons destroys the generalization ability and leading to the problem of overfitting. However for most of the supervised learning algorithms, the number of hidden neurons can only be obtained by trial and error. Therefore, a sequential learning method, which can adjust the number of neurons dynamically during the training, is more appropriate [17].

One disadvantage of radial basis function is that the cost of classifying new instances can be high. This is due to the fact that nearly all computation takes place at classification time rather than when the training examples are first encountered. Therefore, techniques for efficiently indexing training examples are a significant practical issue in reducing the computation required at query time [11]. Radial basis function performs well when many training data are available [18].

Radial basis function networks provide a global approximation to the target function, represented by a linear combination of many local kernel functions. The value for any given kernel function is non-negligible only when the input x falls into the region defined by its particular center and width. Thus, the network can be viewed as a smooth linear combination of many local approximations to the target function. One key advantage to RBF networks is that they can be trained much more efficiently than feedforward networks trained with backpropagation follows from the fact that the input layer and the output layer of an RBF are trained separately [11].

4.3 General Regression Neural Network

GRNN is a memory-based network that provides estimates of continues variables and converges to the underlying (linear and nonlinear) regression surface. This GRNN is a one pass learning algorithm with a highly parallel structure. Even with sparse data in a multidimensional measurement space, the algorithm form can be used for any regression problem in which an assumption of linearity is not justified. The parallel network form should find use in applications such as learning the dynamics of a plant model for prediction or control [16].

Extensive efforts have been devoted to developing techniques of linear time-invariant systems. The linear identification is based on measured input and output values of the system, identification for non linear systems is also based on measured input and output values, but it is more difficult [16].

If the variables to be estimated are future values, then the procedure is a predictor. If the variable(s) to be estimated relate output variables to input variables, then the procedure can be used to model the process or system. Once the system has been modeled, a control function can be defined. If the procedure is taught samples of a control function, it can estimate the entire control function, and it becomes a controller [16].

The regression of a dependent variable, Y, on independent variable, X, is the computation of the most probable value of Y for each value of X based on a finite number of possibly noisy measurements of X and the associated values of Y. The values of X and Y are usually vectors. In system identification, the dependent variable, Y, is the system output and independent variable, X, is the system input. In order to implement system identification, it is usually necessary to assume some functional form with unknown parameters αi. The values of the parameters are chosen to make the bet fit to the observed data. In the case of linear regression, for example, the output, is assumed to be a linear function of the input, and the unknown parameters αi , are linear coefficients. The approach presented in [16] uses a method that frees it from the necessity of assuming a specific functional form. Rather, it allows the appropriate form to be expressed as a probability density function (pdf) that is empirically determined from the observed data. Thus, the approach is not

limited to any particular form and requires no prior knowledge of the appropriate form. In [16] the joint pdf will be estimated from examples using nonparametric estimators. The resulting regression equation can be implemented in a parallel, neural-network-like structure. Since the parameters of the structure are determined directly from examples rather than iteratively, the structure “learn” and can begin to generalize immediately. To the extent that the network is implemented in parallel hardware, it also can estimate vales of Y for any new value of X in the short time determined by the propagation time through four layers of a network [16].

4.3.1 General Regression

Assume that f(x,y) represents the known joint continues probability density function of a vector random variable, x, and a scalar random variable, y. Let X be a particular measured value of the random variable x. the conditional mean of y given X (also called the regression of y on X) is given by:

$$E[y|\boldsymbol{X}] = \frac{\int_{-\infty}^{\infty} y f(\boldsymbol{X}, y)\, dy}{\int_{-\infty}^{\infty} f(\boldsymbol{X}, y)\, dy}. \qquad (4.4)$$

When the density f(x,y) is not known, it must usually be estimated from a sample of observation of x, and y. [16] used the class of consistent estimators, these estimators are a good choice for estimation the probability density function f, if it can be assumed that the underlying density is continues and that the first partial derivatives of the function evaluated at any x are small [16].

Defining the scalar function $D^2_{i,}$

$$D_i^2 = (\boldsymbol{X} - \boldsymbol{X}^i)^T(\boldsymbol{X} - \boldsymbol{X}^i) \qquad (4.5)$$

$$\hat{Y}(\boldsymbol{X}) = \frac{\sum_{i=1}^{n} Y^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}. \qquad (4.6)$$

Where Y^i is a sample value, n is a number of sample observations. Ŷ (X) is a desired conditional mean, X =is a particular measured value if random variable x, X^i is a Sample value [16].

Last equation which involves summations over observations, it is directly applicable to problems involving numerical data. The estimate Ŷ(X) can be visualized as a weighted average of all the observed values, Y_i, where each observed value is weighted exponentially according to its Euclidean distance from X. when the smoothing parameter σ is made large, the estimated density is forced to be smooth and in the limit becomes a multivariate Gaussian with covariance $\sigma^2 I$. on the other hand, a smaller value of σ allows the estimated density to assume non-Gaussian shapes, but with the hazard that wild points may have to great an effect on the estimate. As σ becomes very large, Ŷ(X) assumes the value of the sample mean of the observed Y^i, and as σ goes to 0, Ŷ(X) assumes the value of the Y^i associated with the observation closest to X. for intermediate values of σ, all values of Y^i are taken into account, but those corresponding to points closer to X are given heavier weight [16].

When the underlying parent distribution is not known, it is not possible to compute an optimum σ for a given number of observations n. it is therefore necessary to find σ on an empirical basis. This can be done easily when the density estimate is being used in a regression equation because there is a natural criterion that can be used for evaluating each value of σ, namely, the mean squared error between Y^j and the estimate Ŷ (X^j). For this purpose, the estimate in last equation must be modified so that the jth element in the summation is eliminated. Thus each $\hat{Y}(X^j)$ is based on inference from all the observations except the actual observed value at X^j . This procedure is used to avoid artificial minimum error as σ → 0 that results when the estimated density is allowed to fit the observed data points. Overfitting of the data is also present in the least-squares estimation of linear regression surfaces, but there it is not as severe because the linear regression equation has only p+1 degrees of freedom. If n>>p, the phenomenon of overfitting is commonly ignored. Y and Ŷ can be vector variables instead of scalars. In this case, each component of the vector Y would be estimated in the same way and from the same observations (X, Y), except that Y is now augmented by observations of each component [16].

4.3.2 Neural Network Implementation

The architecture of a basic GRNN model has four layers: input, pattern, summation and output layer as shown in figure (4.4). Each layer of processing units is assigned with a specific computational function when nonlinear regression is performed. The first layer of processing units termed input neurons, are responsible for reception of information. There is a unique input neuron for each predictor variable in the input vector *X*. No processing of data is conducted at the input neurons. The input neurons then present the data to the second layer of processing units called pattern neurons. A pattern neuron is used to combine and process the data in a systematic fashion such that the relationship between the input and the proper response is “memorized.” Hence, the number of pattern neurons is equal to the number of cases in the training set.

A typical pattern neuron i obtains the data from the input neurons and computes an output y using the transfer function of:

$$\theta_i = e^{\frac{-(X-U_i)'(X-U_i)}{2\sigma^2}} \tag{4.7}$$

Where X is the input vector of predictor variables to GRNN, U_i is the specific training vector represented by pattern neuron i, and σ is the smoothing parameter. The outputs of the pattern neurons are then forwarded to the third layer of processing units, summation neurons, where the outputs from all pattern neurons are augmented. Technically, there are two types of summations, simple arithmetic summations and weighted summations, performed in the summation neurons. In GRNN topology, there are separate processing units to carry out the simple arithmetic summations and the weighted summations. Equations (4.8) and (4.9) express the mathematical operations performed by the ‘simple’ summation neuron and the ‘weighted’ summation neuron, respectively.

$$S_s = \sum_i \theta_i \qquad (4.8)$$

$$S_w = \sum_i w_i\theta_i \qquad (4.9)$$

The sums calculated by the summation neurons are subsequently sent to the fourth layer of processing unit, the output neuron. The output neuron then performs the following division to obtain the GRNN regression output *y* as shown in equation (4.10) below:

$$y = \frac{S_w}{S_s} \qquad (4.10)$$

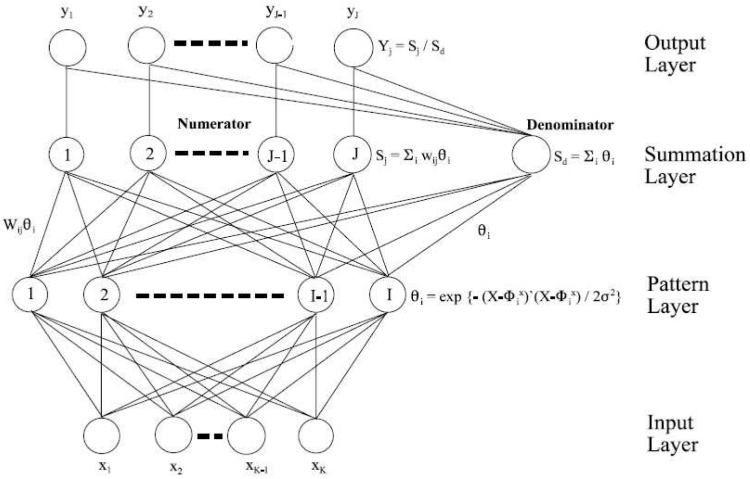

Figure 4.4 The GRNN Block diagram

The advantages of GRNN relative to other nonlinear regression technique are as follows [16]:

1. The network “learns” in one pass through the data and can generalize from examples as soon as they are stored.
2. The estimate converges to the conditional mean regression surfaces as more and more examples are observed; yet, as indicated in th examples, it forms very reasonable regression surfaces based on only a few samples.
3. The estimate is bounded by the minimum and maximum of the observations.
4. The estimate cannot converge to poor solutions corresponding to local minima of the error criterion (as sometimes happens with iterative techniques).
5. A software simulation is easy to write and use.
6. The network can provide a mapping from one set of sample points to another. If the mapping is one to one, an inverse mapping can easily be generated from the same sample points.
7. The clustering version of GRNN limits the numbers of nodes and (optionally) provides a mechanism for forgetting old data.

The main disadvantage of GRNN (without clustering) relative to other techniques as that it requires substantial computation to evaluate new points. There are several ways to overcome this disadvantage. One is to use to clustering versions of GRNN. Another is to take advantage of the inherent parallel structure of this network and design semiconductor chips to do the computation. The two in combination provide high throughput and rapid adaptation [16].

GRNN suffers badly from the curse of dimensionality. GRNN cannot ignore irrelevant inputs without major modifications to the basic algorithm. So GRNN is not likely to be the top choice if you have more than 5 or 6 nonredundant inputs [2].

To summarize, The GRNN is principally a normalized RBF network for which a hidden unit is centered at every training sample. The RBF units of the GRNN architecture are generally characterized by the Gaussian kernels. The hidden layer to output layer weights are just the target values, so that the output is simply a weighted average of the target values of training cases close to the given input case. The GRNN is a universal approximator for smooth functions, so it should be able to solve any smooth function approximation problem provided enough data are given.

Chapter 5

Design and Implementation

5.1 Introduction

In this chapter we will try to answer the following questions: What kind of data will be used to train and test the neural network? What are the artificial neural network (ANN) algorithms will be used in this study? What is the best neural network architecture? Which factors or features are most important in pancreas so we can use them as input data? What is the output of the neural network? What are the optimal learning parameters? What is the method is used to design the neural network?

5.2 Data Used For the Study

5.2.1 Types of Subjects

In this study three types of subjects will be used, the first group is Type 2 Diabetic mellitus (T2DM), Normal Glucose Tolerance (NGT), and Impaired Glucose Tolerance (IGT). And next is a brief description for each one of them.

1. Type 2 Diabetic Mellitus (T2DM):

Type 2 Diabetes was previously called Non–Insulin-Dependent Diabetes Mellitus (NIDDM). Type 2 diabetes accounts for about 90% to 95% of all diagnosed cases of diabetes. It usually begins as Insulin resistance, a disorder in which the cells do not use Insulin properly. As the need for Insulin rises, the pancreas gradually loses its ability to produce it, then the patient is diagnosed as T2DM [26]. In this study The T2DM group is consisting of 54 T2DM patients with a mean of diabetic duration of 4.9 ± 5.6 years (yr). There are 44 males and 10 females with diabetes. In this study the mean age of this group is 55.9 ± 8.0 yr and this group has body mass index (BMI) mean of 30.2 ±5.3 (kg/m2).

2. Normal Glucose Tolerance (NGT):

This group is the normal peoples who don’t have diabetes or high Glucose level, their pancreas is operating normally, and it produces Insulin and Glucagon in normal way as we have discussed before. This group is consisting of 33 peoples; there are 27 males and 6 females. The average age of this group is 56.2 ± 9.1 and the average of BMI is 29.6 ± 6.2.

3. Impaired Glucose Tolerance (IGT):

Its some kind of Pre-diabetes, People with pre-diabetes have higher than normal blood Glucose levels, but that are not high enough to diagnose diabetes. People with pre-diabetes have normal to high levels of Insulin, which their body cannot use efficiently (called Insulin resistance). If not treated, pre-diabetes will eventually turn into type 2 diabetes [26]. This group is consisting of 15 peoples; there are 12 males and 3 females. The mean age of this group is 55.3 ± 6.8 and this group has body mass index (BMI) mean of 35.0 ± 5.3.

Table 5.1 Summary of the characteristics of the T2DM, NGT, and IGT group

CHARACTERISTICS	T2DM	NGT	IGT
No. (Male / Female)	54 (44/10)	33 (27/6)	15 (12/3)
Age (yr)	55.9 ± 8.0	56.2 ± 9.1	55.3 ± 6.8
BMI (kg/m2)	30.2 ± 5.3	29.6 ± 6.2	35.0 ± 5.3
Fasting PG (mmol/liter)	11.7 ± 4.0	5.9 ± 0.6	6.2 ± 0.6
Fasting plasma Insulin (pmol/liter)	48 ± 5.0	40 ± 4.0	78 ± 15.0
Fasting plasma C peptide (pmol/liter)	778 ± 48	667 ± 42	999 ± 134
Fasting plasma Glucagon (pmol/liter)	13.0 ± 0.8	8.4 ± 0.9	11.2 ± 1.6
Fasting plasma GIP (pmol/liter)	12.7 ± 1.5	8.6 ± 0.7	9.8 ± 1.2
Fasting plasma GLP-1 (pmol/liter)	6.6 ± 0.5	4.9 ± 0.4	4.9 ± 0.4

Summary of the characteristics of the T2DM, NGT, and IGT groups is given in table, number of subjects, gender, age, BMI, the mean of fasting Glucose, Insulin, C-peptide, Glucagon, GIP, and GLP-1 are shown in table (5.1).

The T2DM group was recruited from the diabetes out-clinic, whereas the groups with NGT or IGT classified after an oral Glucose tolerance test according to the World Health Organization (WHO) criteria of 1985, responded to an advertisement in a local newspaper None had a history of bowel disease, alcohol abuse, or, for the NGT/IGT subjects, diabetes among first degree relatives. According to the patients' medical records, they had normal Serum Creatinine, normal hepatic function, and no Albuminuria. The study was approved by the ethical committee for Copenhagen and Frederiksberg Municipalities and was conducted according to the principles of the Helsinki Declaration [1].

5.2.2 Procedure

After 3 days of discontinued antidiabetic medication and an overnight fast for 10 hours, the subjects consumed a mixed breakfast meal containing 2250 kJ (41.8% fat, 40.7% carbohydrate, and 17.5% protein; fiber content, 6.7 g). The meal was served with coffee or tea and ingested within 10–15 minutes. Blood was sampled from a needle in a forearm vein before the start and during the next 4 hours and was distributed into fluoride tubes for analysis of Plasma Glucose (PG) and into EDTA/Aprotinin tubes for analysis of plasma concentrations of GLP-1, GIP, Glucagon, Insulin, and C-peptide [1].

In this study, the data were taken subject to time. The measurements were taken initially at time = 0. Time = 0 is the fasting time which has been measured before eating the meal. After that the measures have been taken of 10 minutes period till 60 minutes, then a period of 30 minutes to 240 minutes (4 hours).

The mean of each attributes from fasting time (time =0) to time= 240 for T2DM, NGT, and IGT subjects are shown in tables (5.2), (5.3), and (5.4) respectively. As seen in these tables that the levels of the hormones and the Glucose are started to increase after 20minutes. It can be justified that the meal was finished after 10 to 15 mints for all subjects. So it's logically that these levels will be increased because the Glucose will be increased after the meal. But these levels have different values from one group to another. For example, lets take how Insulin was changed from time = 20 to time = 30 for all groups, Insulin for T2DM at time = 20 was 88 and increased to130 at time =30. For NGT, Insulin was 170 at time = 20 and increased to 230 at time = 30. At last, IGT Insulin was

202.03 and increased to 322.82. We can notice from this example that the significant differences of the Insulin levels among the three groups of subjects, and it can be justified by the function of the pancreas, if it was operating normally or not.

Also these levels at most times dropped off after 60 or 90 minutes, it can be justified by the secreting of the Insulin at high levels will continue till 90 minutes then started to drop off, of course the amount of insulin secreted into blood will be subject to levels of glucose for normal peoples. So we expect from the neural network to learn this system well unless the nature of the data which have been experimented for each patient have no reasonable noise.

In this study, the input vector of the neural network for both RBF and GRNN is representing a single patient or subject and the elements of this vector will represent the hormones which have been stated before and also the output element which is the Glucose level for single patient. So what we are going to do in this study is to use the neural network models to draw a function that will find the relations between the input vector and the output vector without a prior knowledge about this function.

Table 5.2 sample of T2DM group with average values of the input and output at all times

TIMES	GLUCAGON	INSULIN	C-PEPTIDE	GIP	GLP-1	GLUCOSE
T=0	13.00	48	781	12.81	6.63	11.7
T=10	14.80	72	906	19.21	7.29	11.58
T=20	18.00	88	968	48.11	9.47	12.20
T=30	22.45	130	1103	70.78	12.67	12.90
T=40	22.45	152	1228	78.67	13.77	13.70
T=50	21	163	1283	79.67	14.10	14.50
T=60	19.86	176	1378	81.99	13.99	14.99
T=90	15.89	185	1503	72.18	11.57	14.98
T=120	14.70	165	1503	59.70	10.91	14.59
T=150	13.13	127	1378	54.15	10.03	13.70
T=180	11.94	103	1315	47.34	9.236	13.30
T=210	10.53	84	1256	38.26	8.27	12.50
T=240	10.5	80	1195	30.72	8.49	12.00

Table 5.3 sample of NGT group with average values of the input and output at all times

TIMES	GLUCAGON	INSULIN	C-PEPT	GIP	GLP-1	GLUCOSE
T=0	8.41	40	667	8.61	4.90	5.91
T=10	11.0	70	792	15.11	5.67	5.91
T=20	13.0	170	1230	68.61	10.44	6.55
T=30	13.51	230	1730	91.74	13.80	7.07
T=40	12.51	235	2046	94.34	15.66	7.47
T=50	12.0	270	2290	96.94	16.10	7.47
T=60	10.31	270	2375	94.37	17.18	7.06
T=90	9.11	184	2250	89.48	17.44	6.24
T=120	9.51	130	1812	81.59	15.8	5.83
T=150	9.0	80	1562	65.81	13.38	5.61
T=180	8.51	60	1300	52.03	10.88	5.44
T=210	8.01	40	1060	36.55	9.34	5.42
T=240	8.0	35	1000	27.66	8.24	5.41

Table 5.4 sample of IGT group with average values of the input and output at all times.

TIMES	GLUCAGON	INSULIN	C-PEPTIDE	GIP	GLP-1	GLUCOSE
T=0	11.20	78	990	9.80	4.927	6.27
T=10	14.30	131.54	1099	17.76	6.80	6.28
T=20	17.60	202.03	1220	53.39	11.58	6.75
T=30	17.90	322.82	1721	82.75	13.51	7.16
T=40	17.60	383.50	1834	86.33	13.47	7.40
T=50	15.80	378.67	2046	84.97	14.7	7.47
T=60	14.90	412.66	2243	93.42	14.9	7.4
T=90	12.40	401.61	2363	85.22	14.1	7.30
T=120	11.70	289.03	2241	71.28	13.48	6.7
T=150	11.30	195.73	2050	60.23	11.55	6.4
T=180	11.40	166.13	1721	52.67	9.94	6.22
T=210	11.00	100.10	1221	35.82	9.813	5.90
T=240	11.00	80	1184	27.53	8.25	5.80

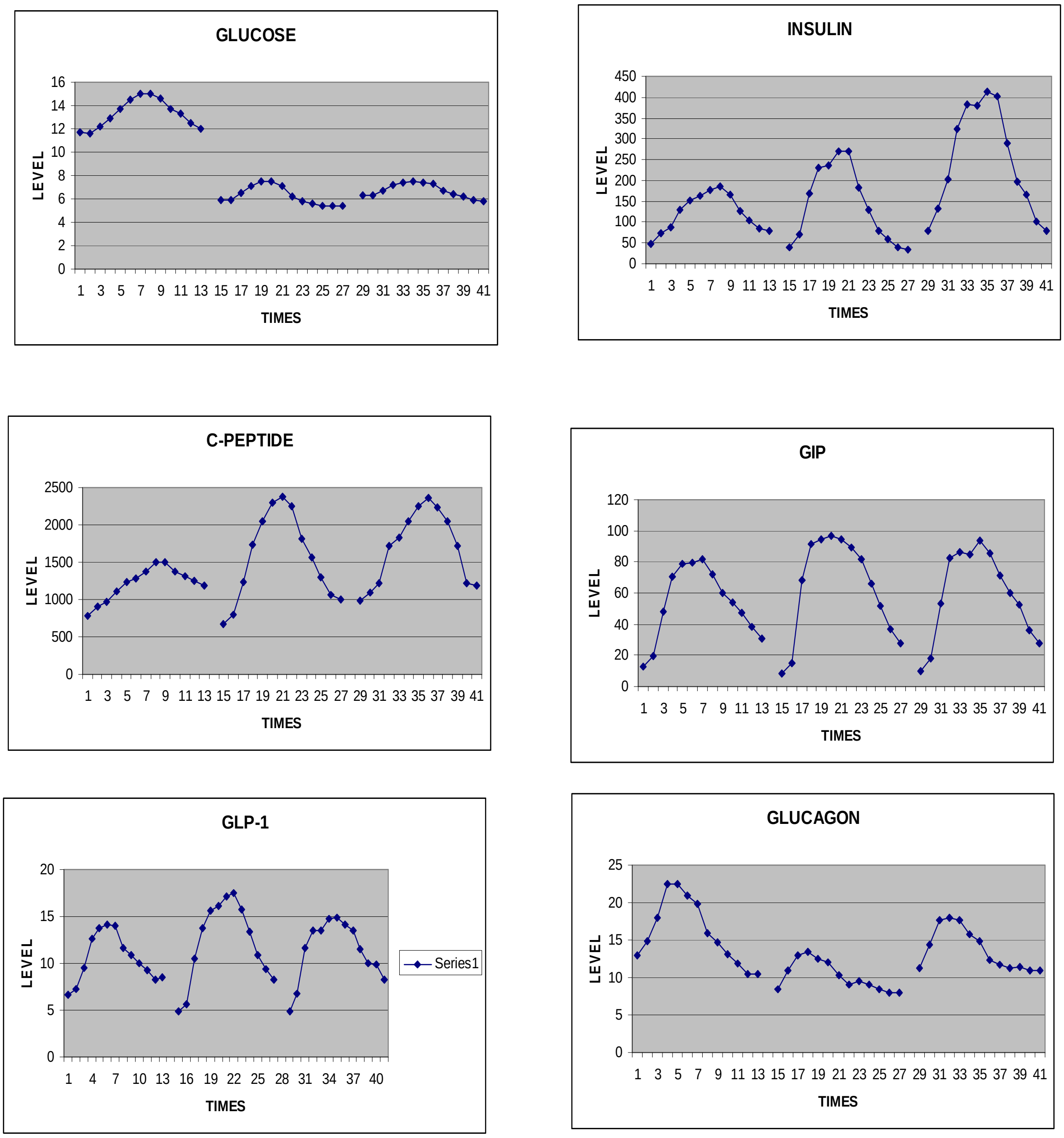

Figure 5.1 the representation of the Insulin, c-peptide, GIP, GLP-1, Glucagon, and Glucose averages for T2DM, NGT, and IGT groups at all times.

In figures (5.1), the x-axis represents the groups and the times, from numbers 1-13 which represent T2DM (1 represents T2DM at time = 0, 2 represents T2DM at time =10, and so on till 13 which represents T2DM at time = 240). In x-axes, numbers from 14 to 27 are representing NGT group from time = 0 to time = 240. Finally, numbers from 28 to 41 are representing IGT group from time = 0 to time = 240. The y-axis is representing the levels of the Insulin, c-peptide, GIP, GLP-1, Glucagon, and Glucose.

As shown in figure (5.7), the shape of the curve of Insulin, C-peptide, GIP, and GLP-1 functions are almost similar for all subjects T2DM, NGT, and IGT. And for Glucagon and Glucose they are almost similar too. This can be justified by their functions, GIP and GLP-1 are stimulating the Insulin levels secreted by pancreas, and c-peptide is secreted in the blood by pancreas almost in the same rate of Insulin. While the main function of the hormone Glucagon is to increase the levels of Glucose in the blood to maintain the normal level of Glucose.

The level of Insulin of T2DM is below the level of Insulin of NGT and IGT even the Glucose level of the T2DM is so high. And this is the main reason for diabetes in this group. And the levels of Insulin in IGT are too high compared to NGT and T2DM. As seen from the figure (5.1), the levels of c-peptide are so high compared to other values and it will give us a strong motivation to make normalization to the c-peptide.

5.3 Methods: MATLAB7

The simulations were released by using MATLAB 7, Neural Network Toolbox. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran [28].

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software. MATLAB has evolved over a period of years with input from many users. MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow learning and applying specialized technology [28].

5.4 Design of the neural Network

5.4.1 Network Architecture

RBFNN Usually contains three layers: the input layer, the hidden layer and the output layer. For GRNN, The architecture of a basic GRNN model has four layers: input, pattern, summation and output layer. For both RBFNN and GRNN, the input layer doesn’t process it just passes the input to the next layer, the hidden layer for RBFNN or the pattern layer for GRNN.

1. The Input and Output Nodes:

Neural network learning provides a robust approach to approximating real-valued, discrete-valued, and vector-valued target functions. Artificial neural networks are among the most effective learning methods currently known. The target function to be learned is defined over instances that can be described by a vector of predefined features. These input attributes may be highly correlated or independent of one another. Input values can be any real values [11]. So after studying how the pancreas works, input must be described as the predefined features of the pancreas, Insulin, c-peptide, Glucagon, and GIP, and GLP1 will be considered as input nodes to the neural network as illustrated in figure (5.2).

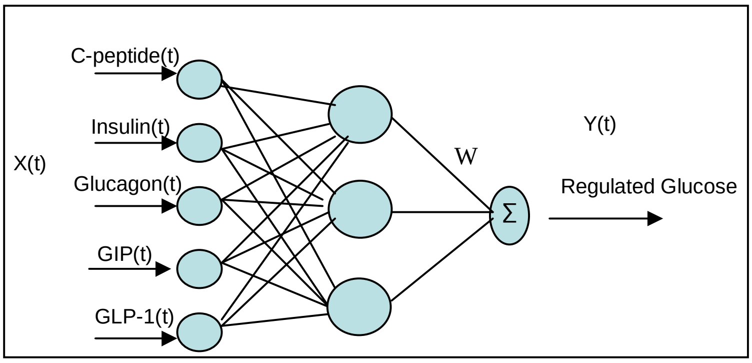

Figure 5.2 neural network architecture; with 5 input nodes and one output node and one hidden layer

2. The Hidden Layer for RBF:

The RBF neural network architecture considered for the application was a single hidden layer with Gaussian RBF. As seen in figure (5.3) the neuron model of radial basis function, receives input vectors from input layer then it calculates the distance between this input vector with weight vector, then a dot product is done between neuron bias and the result of the distance to generate a vector will be used in Gaussian function and the output layer is linear layer.

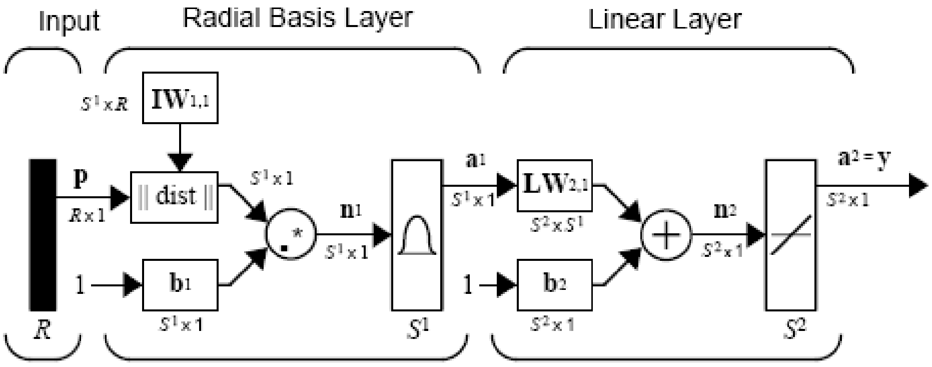

Figure 5.3 architecture of RBF

Where:

R: number of elements in input vector.

P: is a single vector that will be fed to the hidden layer from input layer.

W: weight vector.

‖dist‖: Euclidian distance between its weight vector W and the input vector P.

b1: is bias.

X: is the dot product between ‖dist‖ and bias.

n: is the result from dot product.

∩ . : is the radial basis function, Gaussian.

a: is the output from this neuron to next layer which is the output layer

3. The Number of Neurons in Hidden Layer for RBF:

We have discussed in the previous chapter the methods in which the number of neuron must be decided, so there are two methods : first one is to choose the number of neurons equal to the number of input vectors exist in the training data, suppose the number of neurons in neural network is Q, in this case, Q=P. This function can produce a network with zero error on training vectors. The drawback to this method is that it produces a network with as many hidden neurons as there are input vectors. For this reason, it does not return an acceptable solution when many input vectors are needed to properly define a network, as is typically the case.

The second method is to choose number of neurons less than the number of input vectors. This method also creates a radial basis network one neuron at a time, but we may specify an error goal, so neuron will be added to the network until the sum-squared error falls beneath the error goal or maximum number of neurons has been reached. At each iteration, the input vector that result in lowering the network error the most is used to create a neuron. The error of the new network is checked, and if low enough learning is finished. Otherwise the next neuron is added. This procedure is repeated until the error goal is met, or the maximum number of neurons is reached. In this study, the second method will be used.

Table 5.5 number of neurons and input vectors for each RBF network in this study

NEURAL NETWORK	GOAL	INPUT VECTORS	EPOCHS OR NEURON
NGT t = *	0.002	33	25
T2DM t = *	0.002	54	25
IGT t = *	0.002	15	<15
NGT all times	0.002	429	400
T2DM all times	0.002	702	700
IGT all times	0.002	195	175
NGT and T2DM	0.002	1131	1125
Pancreas	0.002	1326	1300
Pancreas testing	0.002	1117	1100

Note: t = * means at each time from time = 0, 10, 20, 30, 40, 50, 60, 90, 120, 150, 180, 210, and time = 240. As shown from the table (5.5) that number of neurons at all networks was less than number of input vectors, and the error goal was the same in all networks 0.002.

4. Choosing the Spread Parameter

Choosing the spread parameter of the radial basis functions determines the width of an area in the input space, to which each basis function responds. The spread parameters are often known as the smoothing parameters. The only condition we have to meet is to make sure that SPREAD is large enough so that the active input regions of the neurons overlap enough so that several neurons always have fairly large outputs at any given moment. This makes the network function smoother and results in better generalization for new input vectors occurring between input vectors used in the design. (However, SPREAD should not be so large that each neuron is effectively responding in the same, large, area of the input space.)

Table 5.6 spread evaluation (sum squared error (SSE) or performance) in each RBF network,
Where spread = 1, spread = 0.5 and spread 100 are compared

TIME	T2DM PERFORMANCE		IGT PERFORMANCE			NGT PERFORMANCE	
	spread 0.5	spread 1			Spread 100	spread 1	spread 0.5
T=0	0.00698372	0.0077404			0.646561	0.00538381	0.00352886
T=10	0.00700856	0.00550006			0.646561	0.0044842	5.10575
T=20	0.0040293	0.00409547			0.457039	0.0193842	0.0177089
T=30	0.0129406	0.0100435			0.393806	0.00415088	0.0044737
T=40	0.00250796	0.00267576			0.355645	0.00547706	0.0091768
T=50	0.00895158	0.00535531			0.371396	0.00454526	4.70773
T=60	0.006495	0.00768203			0.331815	0.0156191	0.00788685
T=90	0.00454809	0.00449228			0.26509	0.00641297	4.68084
T=120	0.00193609	0.00862235			0.435124	0.0107673	0.0067996
T=150	0.0129238	0.00985514			0.44144	0.01164	0.00336936
T=180	0.0104064	0.00681251			0.62687	0.0071359	0.00378768
T=210	0.0079573	0.00466609			0.636005	0.0114052	0.0101668
T=240	0.00696427	0.00803166			0.674025	0.00174056	0.00300833

Spread = 1 was chosen in all times for T2DM and NGT, because performance of the network was better. The sum squared error (SSE) with networks using the spread 1 was less than networks using spread 0.5 in the most of networks except for T2DM at time =0, 10, 40, 50, 90, 120, 150, 210. But the differences in performance error was so minor almost the same.

For NGT, all networks at all time, the performance of spread 1 was better except at time = 120. For IGT the spread 1 and 0.5 are giving bad performance comparing with networks using spread 100. So spread =100 was chosen in IGT networks at all times because it's the best performance. Of course much more spreads were tried, but the best were the spreads which are shown in the table (5.6) and table (5.7). Choosing the spreads was by using trial and error.

Table 5.7 evaluation of spreads 1 and 0.5 for all RBF networks of NGT, IGT, and T2DM at all times subject to SSE performance

RBF ALL TIMES	PERFORMANCE SPREAD = 0.5	PERFORMANCE SPREAD =1
NGT	0.0132038	0.00215727
T2DM	0.0929451	0.192126
IGT	0.00698776	0.00963103
NGT+T2DM	0.23613	0.2643
Pancreas	0.2866	0.272524
Pancreas-test	0.257464	

.

In the table (5.7), the spread = 1 of NGT all times was better performance than a network with spread=0.5. And for T2DM spread = 0.5 was better than spread = 1. For IGT, the network using the spread = 0.5 was the best performance. For pancreas, the network using spread=1 was better than network using spread = 0.5. For GRNN, the hidden layer is similar to the radial basis network, the optimum spread value was found by trial- and –error, and it was spread of value 0.1.

5. The architecture for the GRNN

The architecture for the GRNN is shown below in figure (5.4). It is similar to the radial basis network, but has a slightly different second layer.

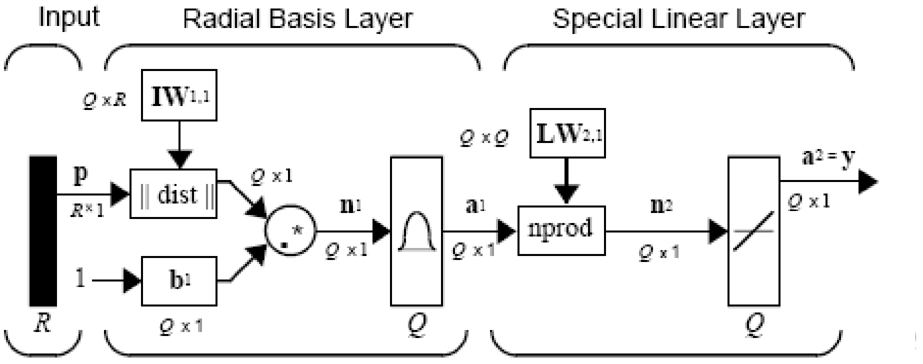

Figure 5.4 The architecture for the GRNN

Here the nprod is a norm product produces number of elements in vector n2. Each element is the dot product of a row of Weight (which W is set to target T) and the input vector a1 (output of the pattern layer), all normalized by the sum of the elements of a1 [4].

5.4.2Performance Criteria

This study is interested in comparing the performance of two learning algorithms RBF and GRNN. What is an appropriate test for comparing learning algorithms? And how can we determine whether an observed difference between the algorithms is statistically significant?

For training, After learning each network by RBF and GRNN for the same learning data set, we simulated or tested the same training set, then we subtracted the target value from the output value for each vector, and this represents the error of the network, then we compared the RBF and GRNN networks by taking the percentage of the samples which passed the test. For example, suppose that RBF has performance of 90%, this means that 90 % of the training data set have passed the test, and the error was below 0.002. And this is the training performance.

For testing, we have trained some samples of data set. Then we have made the simulation on different data rather than the trained data. But from the same data set of this study, and we made the same comparison between target and output. And we found the error. And compared GRNN and RBF based on this error. And this is the testing performance.

5.4.3 Testing

From data set of pancreas which consists of 1326 vectors, we have chosen 1117 for training set, and the rest for testing, we applied both RBF and GRNN on these data to evaluate their performance, For RBF, the network consisted of 1100 neurons and a spread of 1, and goal was 0.002. And for GRNN, a spread of 0.1 was chosen for the network.

5.4.4 Design the Neural Networks

To summarize:

- RBF neural network and GRNN will be applied to the group of subjects of NGT, T2DM, and IGT.

- Standard RBF algorithm was used with Gaussian activation function and the number of neurons was chosen automatically by MATLAB function “newrb” with no need for trial and error strategy as we have done for spread.

- Input nodes are: Insulin, Glucagon, C-peptide, GIP, and GLP-1. And output node is Glucose.

- In our study we have many models, the reason for applying the neural network to many data sets with different noise, number of vectors, type of subjects, time, and nature. We have constructed neural network for each time of the 4 hours we have explained before. we have a network for NGT at t=0, and a network for NGT at time = 10,…, and a network NGT at time= 240, means that we have 13 networks for NGT, 13 networks for T2DM, and 13 networks for IGT.

Chapter 6

RESULTS

The training and simulation were released by using MATLAB 7, Neural Network Toolbox. Two different neural network structures, which are radial basis function (RBF) and general regression neural network (GRNN) were applied to T2DM patients, NGT peoples, and IGT persons. There were 54 T2DM patients, 33 NGT persons, and 15 IGT persons in this study. After an overnight fast for 10 hours, the subjects consumed a mixed breakfast meal, and the data were taken from the subjects: Insulin, C-peptide, Glucagon, GIP, GLP-1, and Glucose. these data were taken from the subjects at time = 0 which is the fasting one, then every 10 minutes period to an hour, and 30 minutes period to 4 hours.

In this study, both RBF neural network and GRNN were applied to:

1. T2DM from t = 0 to t = 240.
2. NGT from t =0 to t =240.
3. IGT from t = 0 to t = 240.
4. T2DM all times together.
5. NGT all times together.
6. IGT all times together.
7. NGT and T2DM together at all times.
8. NGT, T2DM, and IGT together at all times.
9. NGT, T2DM, and IGT data set was divided into training set and test set.

The number of neurons was different from neural network to another, as discussed in the previous chapter. The error goal for the radial basis function networks was 0.002. For GRNN and RBF, the optimum spread values were found by trial-and-error and used for training and test data. For GRNN and RBF, spread value of 0.1 and 1 respectively. For IGT all times RBF network, the spread was 0.5 and a spread of 100 for IGT from t = 0 to t-240.

Table 6.1 performance of GRNN and RBF for NGT, IGT, and T2DM at each time

TIMES	T2DM (%)		NGT (%)		IGT (%)	
	RBF	GRNN	RBF	GRNN	RBF	GRNN
T=0	93	60	88	80	84	100
T=10	97	67	90	82	60	77
T=20	96	65	87	70	50	65
T=30	96	58	90	80	43	60
T=40	97	65	100	75	47	62
T=50	90	63	97	76	40	66
T=60	95	64	85	74	44	59
T=90	98	75	90	78	50	66
T=120	96	80	100	80	55	68
T=150	98	56	100	77	65	72
T=180	97	75	95	84	80	84
T=210	98	71	90	80	66	70
T=240	97	78	90	85	63	65

The results of this study are summarized as follows:

- As shown in the table (6.1), for T2DM the performance of RBF was better than GRNN at all times, RBF was slightly better than GRNN for NGT, and for IGT, RBF was worse than GRNN at all times, RBF was worse for IGT and slightly better than GRNN for NGT because RBF neural network is not performing well when there are no enough data, thus Radial basis function performs well when many training data are available. GRNN performance was not a good choice in T2DM. It maybe because one of many reasons, but for sure the main reason is the nature of data of T2DM. GRNN is suffering badly from the curse of dimensionality. GRNN cannot ignore irrelevant inputs without major modifications to the basic algorithm. So GRNN is not likely to be the top choice if we have more than 5 or 6 nonredundant inputs.

Table 6.2 performance of RBF and GRNN for NGT, IGT, and T2DM at all times

ALL TIMES	RBF	GRNN	INPUT
T2DM	80%	50%	702
NGT	99	70	429
IGT	97	85	195
NGT + T2DM	88.1%	63%	1131
NGT + T2DM + IGT	90%	66%	1326

- In the table (6.2), RBF performance for NGT, T2DM and IGT at all times was above 88.1, this is can be justified because there are many training data are available to train the neural network. Again GRNN was not performing well in T2DM for the same reason; maybe there are more than 6 nonredundant inputs.

- For GRNN IGT at all times the performance was the best = 85. It can be justified, if we looked again to the input data for IGT, see figure (6.1). We can see that the maximum value of insulin was so high at most times in IGT compared to other values such as GIP, GLP-1, Glucagon, and Glucose.

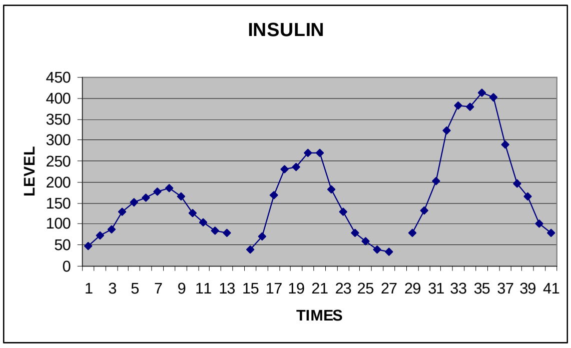

Figure 6.1 Insulin level at all time for T2DM, NGT, and IGT.

We justify that GRNN performed so well in IGT for these reasons:

1. Choosing an optimal spread of 0.1 made the performance optimal (See table 6.3).
2. Normalized product in layer 2 of GRNN is normalization, so it gave better results than RBF. When insulin was so high, the normalization was needed.
3. GRNN is performing better than RBF when there is not many training set as discussed in chapter 3.

4. There was no nonredundancy vector in IGT training data.

Table 6.3 comparing different spread performance of IGT all times for GRNN.

GRNN	SPREAD= 0.1	SPREAD=1	SPREAD=2	SPREAD =3	SPREAD=9
IGT all times	100%	98%	90%	60%	10%

As shown in table (5.3) above, we notice that when we chose a spread of 0.1, it gave us performance of 100%, and every time the spread gets larger, the performance gets down.

- As shown in table (6.4), the performance of the RBF was better than GRNN in training, 84% for RBF and 72% for GRNN. But when the testing is compared, we can see that the GRNN is better than RBF in testing, RBFNN performed so badly 45%, and GRNN performed well.

Table 6.4 Performance of the testing data for RBF and GRNN.

TEST DATA	TRAINING	TESTING	SPREAD
RBFNN	84%	45%	1
GRNN	72%	80%	0.1

- As can be seen in figures (6.2) and (6.3), the RBF for T2DM at all times and GRNN for T2DM all times respectively, the following figures show the response of neural network. Inputs with targets have been trained to approximate the function, and then the targets with output vectors were plotted, The target values (which are the neural network response) are shown by circles, the output measurements are given by the ‘+’ symbols, clearly for RBF there is still an error (when the ‘+’ symbols are not in the center of the circle) but is not significant as in GRNN. Since the given data for T2DM is noisy. We justified that GRNN was performing badly by the nature of data itself, choosing optimal spread by trial and error, or existing of more than 6 nonredundancy vectors in T2DM as [2].

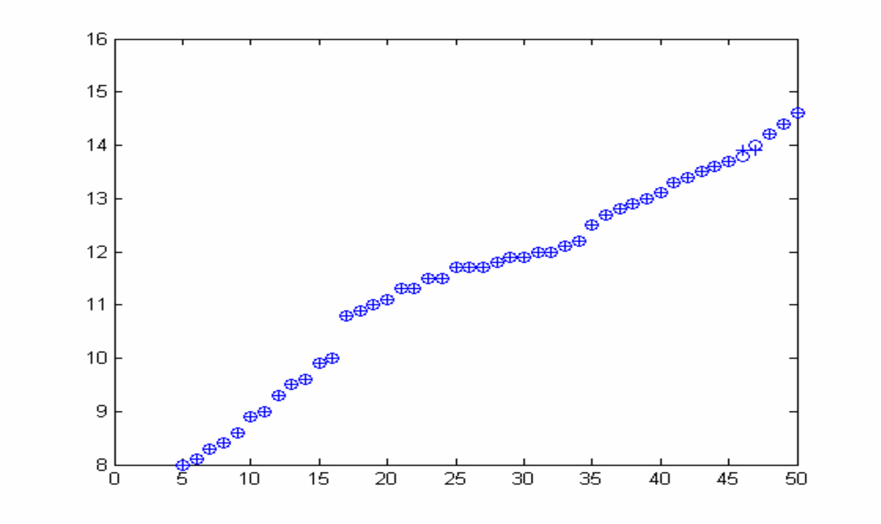

Figure 6.2 training T2DM by RBF, while O is the target and + is the output.
X-axis is the subjects and y-axis is the value of glucose from 8 to 16.

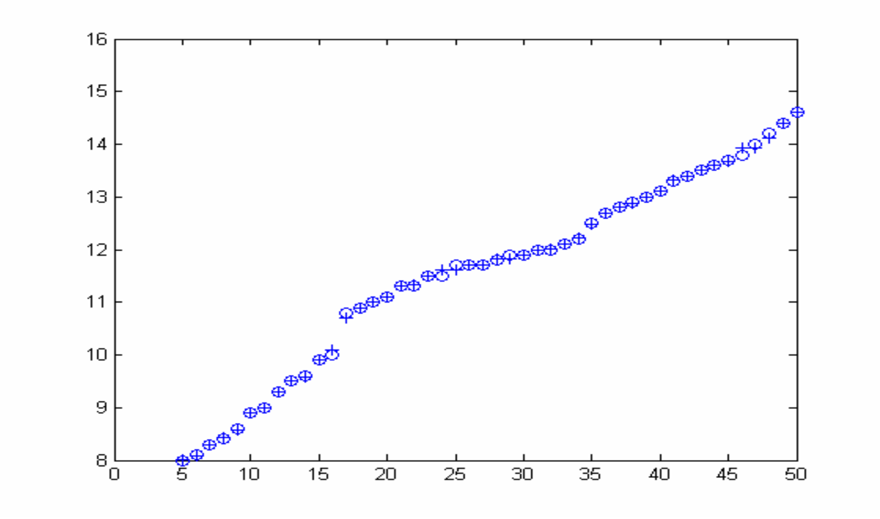

Figure 6.3 training T2DM by GRNN, while O is the target and + is the output.
X-axis is the subjects and y-axis is the value of glucose from 8 to 16.

- In the following figures (6.4) and (6.5), the RBF for NGT and T2DM at all time and GRNN for NGT and T2DM at all times, respectively. After the training has been done to approximate the function, we have plotted the target vectors by ‘O’ and the response of the network by ‘+’, the x-axis represents only 600 subjects of 720 subjects, y-axis represents the Glucose (output) from levels 8 to 10, also clearly we can notice that RBF has better results than GRNN. (errors are noticeable by not placing the ‘+’ in the center of the ‘O’).

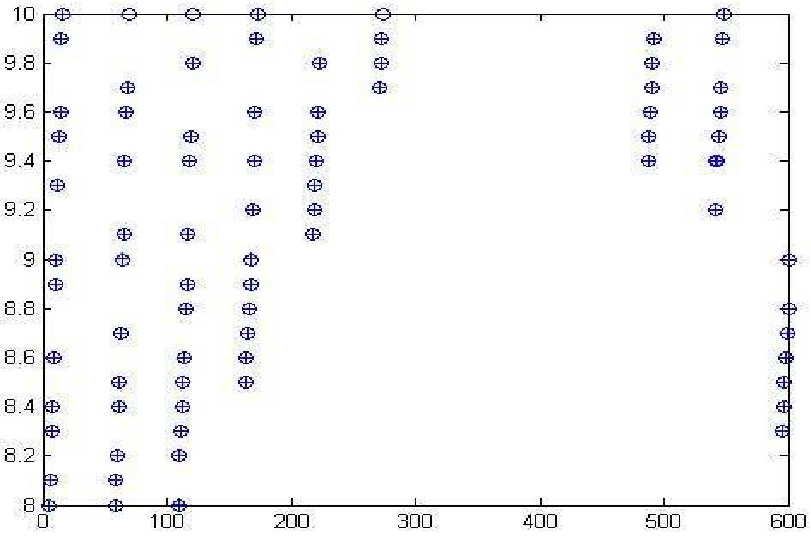

Figure 6.4 training NGT and T2DM by RBF, while O is the target and + is the output. X-axis is the subjects and y-axis is the value of glucose from 8 to 10.

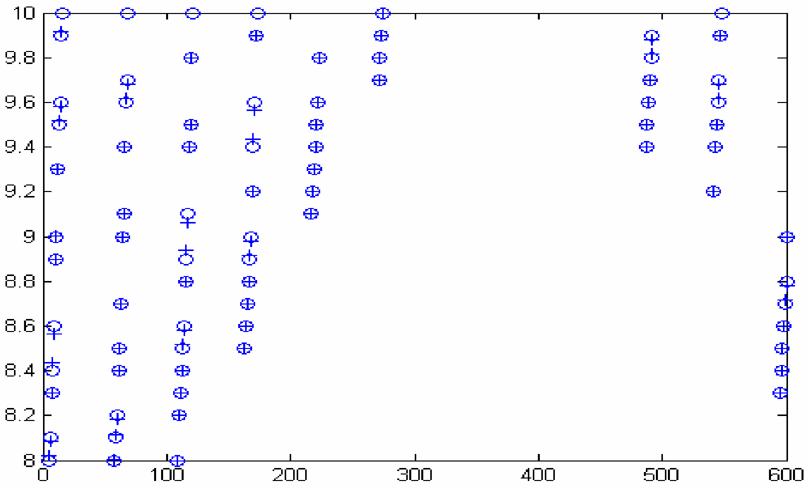

Figure 6.5 training NGT and T2DM by GRNN, while O is the target and + is the output. X-axis is the subjects and y-axis is the value of glucose from 8 to 10.

- Finally, figures (6.6) and (6.7) plotted the targets and output vectors for pancreas RBF and pancreas GRNN. X-axis for only 600 subjects and y-axis in figure (5.6) represents Glucose levels.

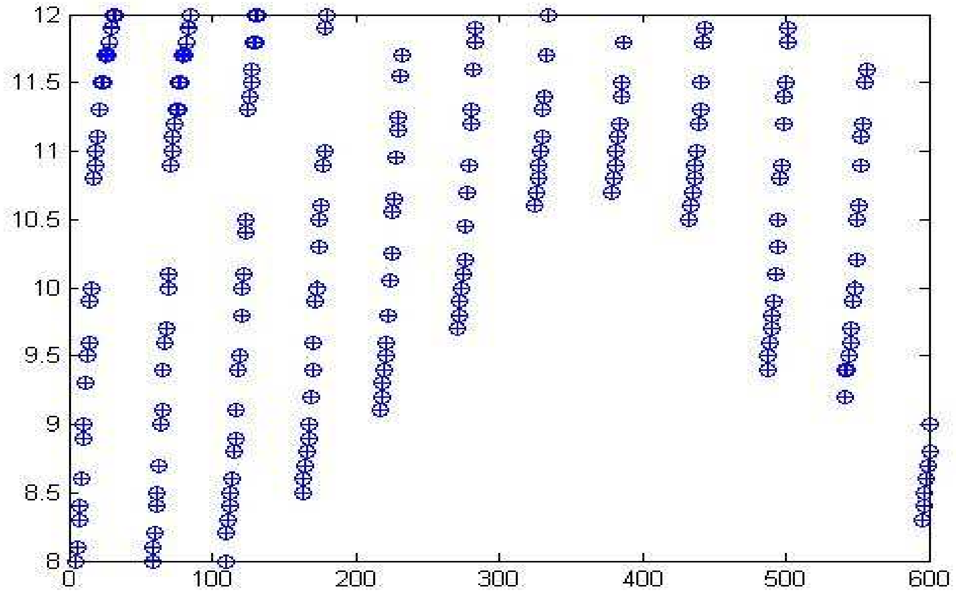

Figure 6.6 training Pancreas by RBF, while O is the target and + is the output.
X-axis is the subjects and y-axis is the value of glucose from 8 to 12.

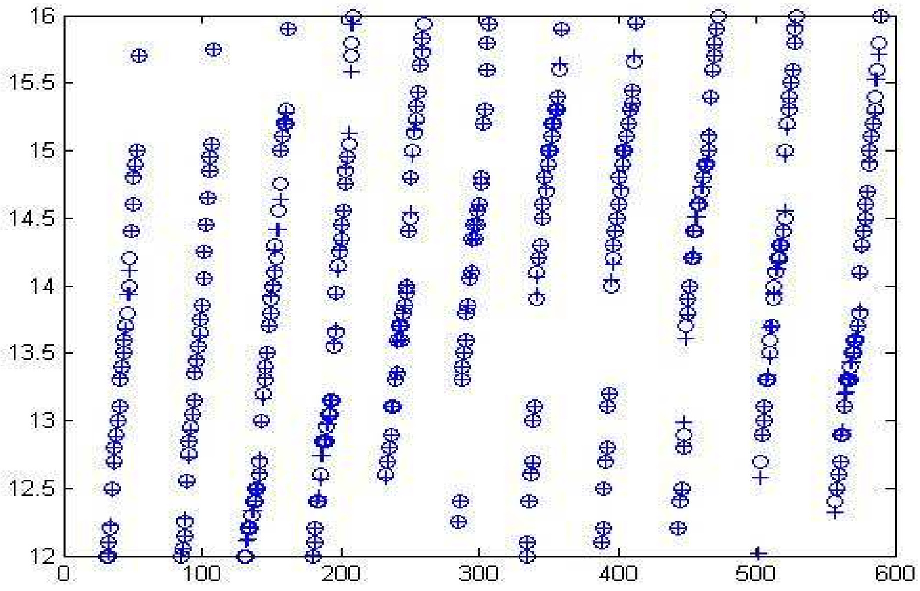

Figure 6.7 training Pancreas by GRNN, while O is the target and + is the output.
X-axis is the subjects and y-axis is the value of glucose from 12 to 16.

Table 6.5 converging to error goal according to spread and number of neurons

NEURONS	PERFORMANCE SPREAD = 1	PERFORMANCE SPREAD =0.5	NEURONS	PERFORMANCE SPREAD = 1	PERFORMANCE SPREAD =0.5
1125	0.294827	0.293214	650	4.0657	61.4987
1100	0.299909	0.323221	600	6.71092	109.912
1075	0.342415	0.377774	550	**9.05474**	229.48
1050	0.45427	0.421479	500	13.4859	424.015
1025	0.46734	0.509057	400	55.9734	672.987
1000	0.517112	0.679457	300	262.504	1140.77
950	0.671112	1.07487	200	1365.25	2128.56
900	0.798151	1.5419	100	3532.73	7208.07
850	0.935965	2.36178	75	5108.76	8582.32
800	1.35625	**8.30855**	50	7841.35	10327.6
750	1.80475	23.3575	25	11351.3	13306.2
700	3.11798	38.9404			

- In the table (6.5) shows how the radial basis function converges to the goal, when the RBF network reached 25 neurons and spread of 1, the performance was 11351.3. And for the RBF network of spread 0.5, the performance was 13306.2. RBF network of 100 neurons and a spread of 1, the performance was 3532.73, and RBF network of spread = 0.5, the performance was 7208.07. RBF network of 500 neurons and a spread of 1, the performance was 13.4859, for RBF network of spread = 0.5 it was 424.015. We can notice that the RBF network of spread = 1 has improved significantly in the first 550 neurons then it started to converge slowly, and the RBF network of spread = 0.5 started to fit badly then it improved after 800 neurons, finally, the RBF network of spread 1 converged to error 0.294827, while the RBF network of spread = 0.5 showed better at last with error 0.293214 both networks consisted of 1125 neurons.

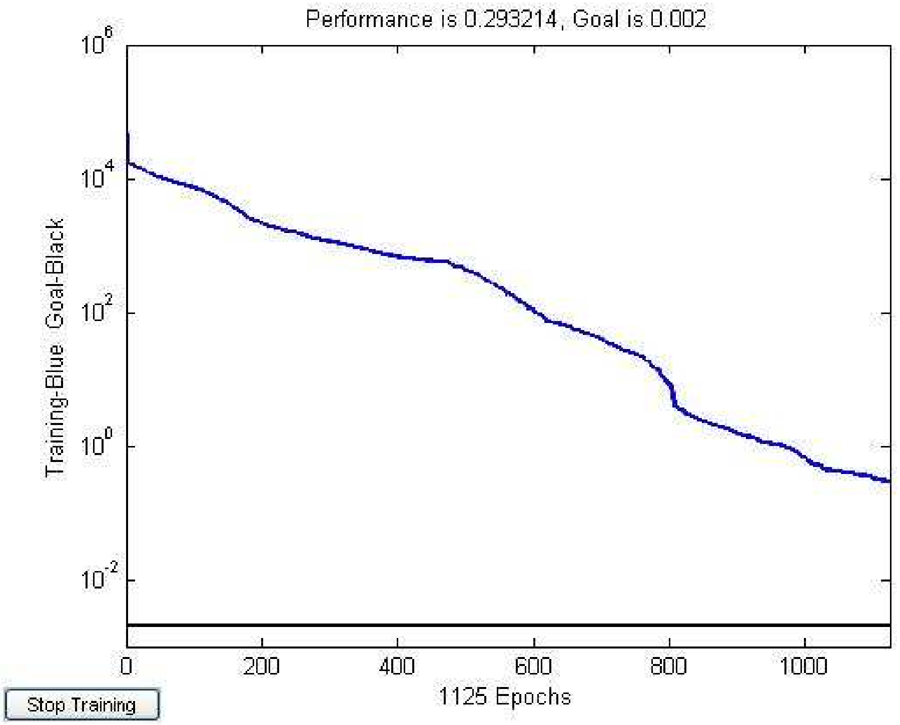

Figure 6.8 convergence curve of a RBF network for NGT and T2DM consisted of 1125 neurons and a Spread of 0.5

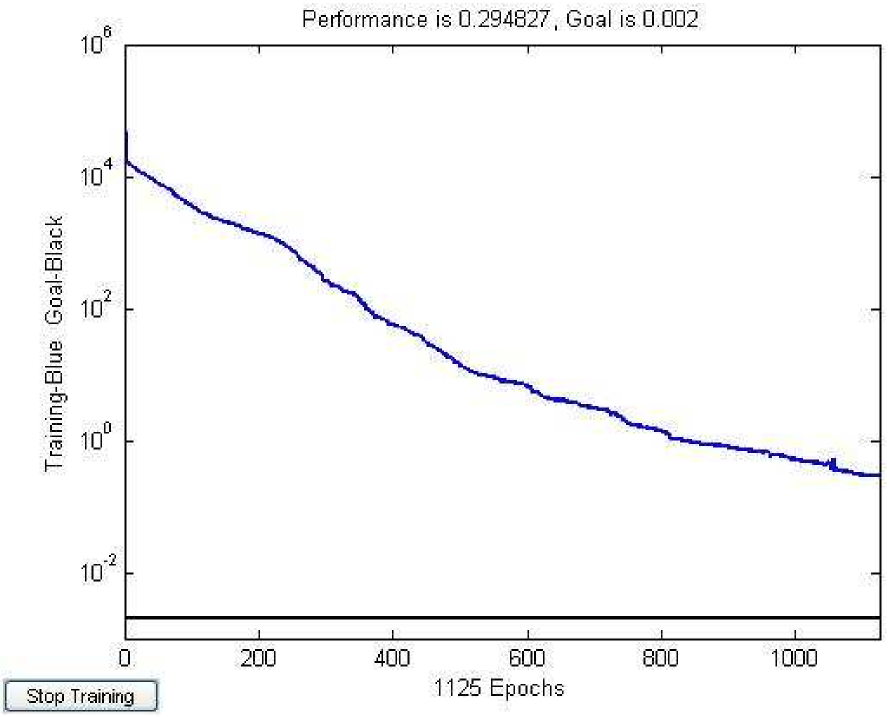

Figure 6.9 convergence curve of a RBF network for NGT and T2DM consisted of 1125 neurons and a Spread of 1

- Convergence curves of both networks are shown in figures (6.8) and (6.9), From the convergence curve in figures (6.8) and (6.9) we can deduce that the fit of RBF network of spread = 0.5 is quite bad compared to the RBF network of spread = 1 in the beginning. Then at the end they converged to approximately the same performance

- Two things are apparent now. First, with all neural network problems we face, the question of determining the reasonable, if not optimum, size of the network (to make the size of the network bigger, increase the number of iterations, neurons, or epochs) is so important. This brings in also more network parameters, such as the optimal smoothing parameter or spread. The other thing, which could be done, is to improve the training algorithm performance or even change the algorithm.

- Number of input vectors of each network will be the same as number of subjects in each group NGT, T2DM, and IGT. Input vectors were 33, 54, and 15 respectively. Then we made a network for all times to each group, means that we have a network for NGT at all times, T2DM at all times, and IGT at all times, the number of input vectors were 429, 702, and 195 respectively. Also we made another network for NGT and T2DM at all times, the number of input vectors was 1125. Finally, we made a network for all subjects at all times, we will call it Pancreas. The number of input vectors was 1326. See table (4.7).

- Number of neurons was different to each RBF network we have constructed, but the number of neurons was less than the number of input vectors for every network. The number of neurons for NGT at any time, T2DM at any time, IGT at any time, NGT all times, T2DM all times , IGT all times, NGT and T2DM , Pancreas, and Pancreas testing were 25, 25, <15, 400, 700, 175, 1125, 1300, and 1100 respectively.

- Spread was 0.1 for GRNN and for RBF was 1 except for IGT at any times, spread of 100 was chosen, and for IGT all times, a spread of 0.5 was chosen.

- Sum-squared error (SSE) was chosen as error performance algorithm. Goal error 0f 0.002 was chosen for RBF.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Two different neural network structures, which are radial basis function (RBF) and general regression neural network (GRNN) were applied to medical data for three different groups, 54 type 2 diabetes patients, 33 normal glucose tolerance persons, and 15 peoples having impaired glucose tolerance. This study indicates the good function approximation capabilities of RBF neural network and general regression neural networks. When comparing RBF and GRNN models, we find that RBF is better than GRNN in training and GRNN was better in testing.

Recall that the performance of RBF was better than the GRNN for all spread values tried but in some cases such as IGT networks, GRNN was better than RBF, we relate these results to the nature of data we have, and choosing the optimal spread, the only technique was used to choose spread values in this study was trial-and error. Although the RBF neural networks training algorithm gave the best result for the training data, the most important result should be considered with the test data. The best result achieved on the test data is the one using the GRNN structure (85 %).

In this study RBF performed so badly in testing the data, this is due to the fact that nearly all computation takes place at classification time rather than when the training examples are first encountered. Therefore, techniques for efficiently indexing training examples are a significant practical issue in reducing the computation required at query time [11].

In this study, it was apparent that GRNN and RBF are: fast training, modeling of non-linear functions, good function approximators, and good performance in noisy environments given enough data. Although GRNN was faster than RBF, GRNN was better than RBF in testing data (data for testing were taken from the data set itself but excluded from the training), and RBF was better than GRNN in training when there was enough data available. GRNN was performing badly in T2DM networks, we justified that by existing more than 6 nonredundant inputs [2]. The limitation of the RBF neural network is that it is more sensitive to dimensionality, has greater difficulties when there is not enough training data, and a new training for the neural network is needed to approximate new data and this is neither very economic nor practical procedure.

Unfortunately, we still have some limitations and drawbacks in this study, we determined the optimal parameters (spread and centers) more or less by trial-and-error. A more sophisticated method to determine the optimal parameter values is desirable and it improves the performance considerably. A potential limitation of this study is that training and testing were performed with a single data set and that there were no results presented based on the external/similar independent dataset.

.

We conclude that the result is quite satisfactory and it reflects the nature of both algorithms, RBF and GRNN. Results showed the advantages and disadvantages of both algorithms. Although results would still require improvements by adapting automatic techniques choosing optimal parameters, applying the neural network models to external data set, and filtering the data for redundancy, clustered samples and outliers.

7.2 Recommendations for Future work

The main drawback we have faced in our design is the generalization problem of RBF, and we need to be improved for GRNN. So, in the future, we are going to upgrade our model by applying amendments to the standard RBF.

Many proposals have been made to solve the problem of overfitting or generalization. My recommendation is to make the normalization and treatment of data in three steps, first step, before the training by using pre-treatment of data, such as eliminating the outliers, or to test the standard deviation before the training, second step, using normalized RBF, which will use the normalization in hidden layer. And at last, after testing, to find an algorithm to remove the data which has significant or even small errors and to make new training set and to repeat the steps 1 and 2 again, and to test the data. So in this case we have two training algorithms, means that when new data come and needed to be tested or generalized, we need to use method of trial and error or other methods to check which algorithm must be used for the data to be generalized.

REFERENCES

[1] Britt, M., Nielsen, T., Damholt, M.B., Madsbad, S., Hilsted, L.M., Hughes, T.E., Michelsen, B.K., And Holst, J.J. “Determinants Of The Impaired Secretion Of Glucagon- Like Peptide-1 In Type 2 Diabetic Patients”, The Journal Of Clinical Endocrinology & Metabolism (2001).

[2] Caudill, M. "GRNN and Bear It," AI Expert. (1993)

[3] Chen, S., Cowan, C.F.N. and Grant, P.M. “Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks”, IEEE Transactions on Neural Networks. (1991).

[4] Demuth, H., Beale M. “Neural Network Toolbox for use with MATLAB®”. (2000)

[5] Fausett, L. “Fundamentals of Neural Networks”, Prentice Hall, USA. (1994)

[6] Schuit, F.C., Huypens, P., Heimberg, H., and Pipeleers, D.G. ”Glucose Sensing in Pancreatic b-Cells A Model for the Study of Other Glucose-Regulated Cells in Gut, Pancreas, and Hypothalamus”. Diabetes, Vol. 50, (2001).

[7] Hawickhorst, B.A., Zahorian, S.A., “A Comparison of Three Neural Network Architectures for Automatic Speech Recognition”, Old Dominion University, Norfolk, (2000).

[8] Kayaer, K., and Yildirim, T. “Medical Diagnosis on Pima Indian Diabetes Using General Regression Neural Networks”. Turkey

[9] Keogan, M.T, Lo, J.Y., Freed, K. S., Raptopoulos, V., Blake, S., Kamel, I.R., Weisinger, K., Rosen , M.P., Nelson, R.C. “Outcome Analysis of Patients with Acute Pancreatitis by Using an Artificial Neural Network1”, Academic Radiology, Vol 9, No 4, April (2002).

[10] Kok, P. ” Predicting blood glucose levels of diabetics using artificial neural networks” Research Assignment for the Master of Science program Computer Science, Delft University of Technology. (2004)

[11] Mitchell,T. M. “Machine learning”, book: chapter 8 : instance based learning, radial basis function, page 238.

[12] Mougiakakou, S.G; Prountzou, A.; Iliopoulou, D; Nikita, K.S.L Vazeou, A.; Bartsocas, C.S.,’’ Neural Network Based Glucose-Insulin Metabolism Models For Children With Type1 Diabetes,’’ Annual International Conference Of The IEEE, (2006)

[13] M. P. Wand and M. C. Jones: ‘Kernel smoothing’; London. (1995)

[14] Prank, K., Jurgens, C., Muhlen, A.V, Brbant, G. ,“Predictive Neural Networks For Learning The Time Course Of Blood Glucoe Levels From The Complex Intercation Of Counterregulatory Hormones “, Nannover, Germany. Neural computation (1998).

[15] Rojas, R. “Neural Networks: A Systematic Introduction” Springer, Berlin. (1996)

[16] Specht, D. F. “A general regression neural network” IEEE Transactions on Neural Networks. (1991)

[17] Runxuan, Z. “Efficient Sequential and Batch Learning Artificial Neural Network Methods for Classification Problems” A thesis submitted to the Nanyang Technological University in fulfillment of the requirements for the degree of Doctor of Philosophy. (2005)

[18] Stamatios V. Kartalopoulos, “understanding neural networks and fuzzy logic,basic concepts and applications” , IEEE Neural Network Council, Sponsor.

[19] Thorens, B. "Glucagon-like peptide-1 and control of insulin secretion". Diabète & métabolisme. (1995)

[20] T. Poggio and F. Girosi, “Regularization algorithms for learning that are equivalent to multilayer networks,” Science. (1990)

[21] Trajanoski, Z., “Neural Predictive Controller for Insulin Delivery Using the Subcutaneous Route”, IEEE Transactions on Biomedical Engineering, Vol. 45, No. 9, September (1998)

[22] Tresp, V., Briegel, T. and Moody, J. ”Neural Network Models for the Blood Glucose Metabolism of a Diabetic”, IEEE Transactions on Neural Networks. (1999)

[23] Venkatesan, P., and Anitha, S. “Application of a radial basis function neural network for diagnosis of diabetes mellitus” India. Current Science, Vol. 91, No. 9, 10 November (2006).

[24] Zitar, R.A. “Towards Neural Network Model for Insulin/Glucose in Diabetics-II”.
Philadelphia University/ Jordan. Informatica (2005).

[25] http:// www.Biologyonline.org. (DEC 2007)

[26] http:// www.wikipedia.com. (DEC 2007)

[27] http://www.ncbi.nlm.nih.gov/books. (DEC 2007)

[28] http://www.glucagon.com. (Jan 2008)

[29] http:// www.diabetes.org, “American Diabetes Association”. (Jan 2008)

[30] http:// www.ncd.org.jo, “Jordan Diabetes National Center” (Dec 2007).

تعتبر الشبكات العصبونية (Neural Networks) حاليا من احد مواضيع الهامه في البحث العلمي الطبي وهذا العمل يقوم على التنبؤ باستخدام الشبكات العصبونية لسلوك عضو من اعضاء الانسان يدعى البنكرياس.

ان الشبكات العصبونية والتي لها القدرة الكبيرة على ايجاد نتائج من بيانات معقدة وغير دقيقة وتستطيع استخراج نماذج وايجاد نزعات والتي تكون معقدة جدا ايجادها من قبل الانسان او اساليب الحواسيب الاخرى .
وقدرة الشبكات العصبونية للتعلم من الامثلة الموجودة تجعلها قوية ومرنة.

في هذا البحث لدينا بيانات من 102 شخص ومن هذه البيانات تم تعليم الشبكة العصبونية باستخدام خوارزميات (Radial Basis Functions) و (General Regression Neural Networks) وبعد التعلم تم عمل فحص للبيانات المتعلمة وتم المقارنة بين الخوارزمتين حسب كفاءة التعلم وكذلك تم عمل مقارنة بين الخوارزمتين بالنسبة للتعميم على بيانات جديدة لم تدخل مرحلة التعلم.

كل العمليات التشبيهية تم عملها في (MatLab 7) والنتائج كانت جيدة لكلا الخوارزميات بالنسبة للتنبؤ وكانت سريعة التعلم مع ان(General Regresion Neural Network) كانت اسرع. اظهرت النتائج ان الشبكات العصبونية خيار جيد في انظمة غير خطية ومعقدة مثل نظام البنكرياس واظهرت النتائج أن كفاءة (Radial basis function) في التعلم كانت احسن من (General Regerssion Neural Networks) وعملية التعميم والفحص كانت احسن بالنسبة ل (General Regression Neural Networks).

استخدام الشبكات العصبونية في التنبؤ في سلوك البنكرياس

من قبل
بسام عبدالرحمن عابد

باشراف
د. رشيد الزبيدي

قدمت هذه الرسالة استكمالا لمتطلبات
الحصول على درجة الماجستير في علم الحاسوب

عمادة البحث العلمي والدراسات العليا
جامعة فيلادلفيا