Towards Neural Network Model for Insulin/Glucose in Diabetics-II

Raed Abu Zitar
College of Information Technology
Philadelphia University
Jordan
rzitar@philadelphia.edu.jo

Abdulkareem Al-Jabali
College of Engineering
Al-Isra University
Jordan

Keywords: Levenberg-Marquardt Neural Network, Polynomial Networks, Diabetics, Insulin.

Received: March 18, 2004

In this work we extending our investigations for a general neural network model that resembles the interactions between glucose concentration levels and amount of insulin injected in the bodies of diabetics. We use real data for 70 different patients of diabetics and build on it our model. Two types of neural networks (NN's) are experimented in building that model; the first type is called the Levenberg-Marquardt (LM) training algorithm of multilayer feed forward neural network (NN), the other one is based on Polynomial Network (PN's). We do comparisons between the two models based on their performance. The design stages mainly consist of training, testing, and validation. A linear regression between the output of the multi-layer feed forward neural network trained by LM algorithm (abbreviated by LM NN) and the actual outputs shows that the LM NN is a better model. The PN's have proved to be good static "mappers", but their performance is degraded when used in modelling a dynamical system. The LM NN based model still proved that it can potentially be used to build a theoretical general regulator controller for insulin injections and, hence, can reflect an idea about the types and amounts of insulin required for patients.

Povzetek: Na osnovi podatkov o 70 pacientih je razvit nevronski model za razmerna med insulinom in glukozo.

1 Introduction

Diabetes is a disease in which the body cannot properly use the energy it gets from food. Normally, most of the food we eat is broken down or digested into sugar or glucose. Glucose provides the body's cells with the energy they need. Insulin, a hormone produced in the pancreas, helps the glucose get inside the cells where the glucose is burned for energy. In diabetes the body cannot make enough insulin or is resistant to the insulin it makes. As a result, your blood glucose can become much higher than usual. A normal fasting blood glucose range is about 65 -110. When your blood sugar is 126 or higher after fasting for eight hours, the diagnosis of diabetes is made.

It is a widespread chronic illness that accounts for a large part of the health care budget. It affects approximately one hundred million people world wide [1] and may lead to a variety of vascular, neurological or metabolic complications.

Diabetes and complications associated with it can be viewed as a partial or total failure of one or more intrinsic therapeutic feedback loops. In a healthy person the relationship established between glucose level and insulin secretion is an effective feedback control loop. Increased blood glucose level (the controlled variable) results in the production of the hormone insulin by the pancreas (the controller). This insulin reduces blood glucose from its elevated level. Diabetic patient has not this inter-relationship or it does not work as it does in healthy people.

In practice, the full picture is more complex and the diabetic patient needs to be regarded as a multi-input/multi-output physiological system which contains several controllable and measurable variables as well as other factors which are not directly observable. The patient's diet (the carbohydrate content of which will directly elevate blood glucose level), hormones (gastrointestinal, glucagons, …etc), the physical effort exerted, the amount of insulin delivered, and other factors [2] can be considered to be control variables which need to be adjusted in order to maintain homeostasis within the human organism. Obviously, the manipulation of all variables that affect the dynamics of diabetes is cumbersome.

1.1 Mathematical Models of Glucose/Insulin Dynamics

Mathematical models have provided one mean of understanding diabetes dynamics. There are various models based on glucose and insulin distributions, and those models have been used to explain glucose /insulin interaction . All these models are valid under certain conditions and assumptions [3]-[9]. These models represent a range of approaches, including linear [2],[3], nonlinear [4],[5], probabilistic [6], compartmental [7], non-compartmental [8], and parametric models [9]. Although these models may be useful in a research setting, they all have limitations in predicting blood glucose in real-time clinical situations because of the inherent requirement of frequently updated information about the models' variables like glucose loads and insulin availability. For example, glucose challenges to the body, such as those resulting from a meal, are important glucose sources in models, but are not conveniently measurable and must instead be considered as unknown disturbances. As another example, the timing and amount of subcutaneous insulin injections are known to the patient, but the resulting vascular availability of insulin is often variable, depending on factors such as the insulin dose and delivery site. Since frequent insulin determinations are not practical for routine management, only estimates of vascular insulin concentrations can be incorporated in models when applied in an actual clinical setting. In the absence of accurate, frequently updated information about glucose loads and insulin concentration, these conventional models can only be marginally effective in real time at reliably predicting future blood glucose values [10]. Given this situation, if continuous or very frequent blood glucose monitoring is available, recent and past glucose values may be exploited as an alternative to the use of conventional models to describe blood glucose dynamics.

The features of data that can be used for such studies are sometimes based on individual blood glucose values from a patient or a group of patients, while in many other studies statistical averages of repeated challenges for a given patient a or group of patients are used. Furthermore, blood glucose is sampled frequently enough to capture a detailed record of excursions. The monitoring period for a given individual is extended over a long time period (several weeks). Full information about external factors such as meals, insulin injections and the type, exercise, etc.. that cause blood glucose perturbations is also recorded.

2 The Neural Based Models

Feed forward neural networks have been used extensively to solve many kinds of problems, being applied in a wide range of areas covering subjects such as prediction of temporal series, structure prediction of proteins, and speech recognition [7]. One of the fundamental properties making these networks useful is its capacity to learn from examples. Through synaptic modifications algorithms, the network is capable of obtaining a new structure of internal connections that is appropriate for solving a determined task.

The general underlying theory of the whole learning process is poorly understood. There are few general results, especially concerning generalization. One particular point of interest is the selection of a concise subset of examples from the whole training set as a way of improving generalization ability. This problem has also been referred to as "active learning" or "query-based learning" by many authors. In a broad sense, these terms refer to any form of learning in which the learning algorithm has some control over the inputs used for the training.

In this work, we use two different types of neural networks; the LM NN model and the polynomial network model (PN's). In a previous work, we studied the modeling through Radial Basis Function Networks [?]. We showed that LM NN had more success than Radial Basis Networks. We related that to capability of LM NN training algorithm which is an advanced version of back propagation algorithm to capture the dynamics of the control surface the associates the patient state variables with the output which is the amount of insulin injected. Although, both of them are feed forward types of neural networks, they fundamentally differ in the way training is implemented. LM NN model is a feed forward model consisting of two layers. Its learning strategy starts with incremental error back propagation algorithm and gradually switches to conjugate gradient-based back propagation for the final convergence phase [11].

In the other hand, PN's technique is known for fast convergence toward "closest" local minimum and can escape shallow local minima. We may consider the problem of finding the proper amount of insulin as an identification problem which involves finding the best matching class given a list of target classes (and their models obtained in the training phase) In general, the training data for each class consists of a set of previous state variables of the possible input vectors that come from the history of the patient(s). In our case, each observation is represented by a single vector containing four previous values of the patient state variables which are present glucose level, previous glucose level, meals, exercises, short term insulin, medium term insulin, and long term insulin, we will come to the details of the simulations later .

Now, for each class, i, we have a set of N_i training observations represented by the N_i feature vectors $[\mathbf{x}_{i,1} \quad \mathbf{x}_{i,2} \quad \cdots \quad \mathbf{x}_{i,N_i}]^t$ Identification requires the decision between multiple hypotheses, Hi. Given an observation feature vector x, the Bayes decision rule [7] for this problem is

$$i^{opt} = \arg\max_i p(H_i \mid \mathbf{x}) \tag{1}$$

A common method for solving equation (1) is to approximate an ideal output on a set of training data with a network. That is, if $\{f_i(x)\}$ are discriminant functions [8], then we train $f_i(x)$ to an ideal output of 1 on all in-class observation feature vectors and 0 on all out-of-class

observation feature vectors. If f_i is optimized for mean-squared
error over all possible functions such that

$$f_i^{opt} = \arg\min_{f_i} E_{\mathbf{x},H}\left\{|f_i(x) - y_i(\mathbf{x},H)|^2\right\}, \quad (2)$$

The solution entails that:

$$f_i^{opt} = p(H_i \mid \mathbf{x})$$

In equation (2), $E_{x,\,H}$ is the expectation operator over the joint distribution of x and all hypotheses, and $yi(\mathbf{x},H)$ is the ideal output for H_i. Thus, the least squares optimization problem gives the functions necessary for the hypothesis test in equation(1). If the discriminant function in (9) is allowed to vary only over a given class (in our case polynomials with a limited degree), then the optimization problem of equation (9) gives an approximation of the a posteriori probabilities[8]. Using the resulting polynomial approximation in equation (8) thus gives an approximation to the ideal Bayes rule. The basic embodiment of a K_{th} order polynomial network consists of several parts. In the training phase, the elements of each training feature vector, x = [x_1, x_2 ..., x_M], are combined with multipliers to form a set of basis functions, *p(x).* The elements of *p(x)* are the monomials of the form:

$$\prod_{j=1}^{M} x_j^{k_j}, \text{ where } k_j \text{ is a positive integer, and } 0 \le \sum_{j=1}^{M} k_j \le K \quad (3)$$

Once the training feature vectors are expanded into their polynomial basis terms, the polynomial network is trained to approximate an ideal output using mean-squared error as the objective criterion. The polynomial expansion of the i*th* class feature vectors are denoted by:

$$\mathbf{M}_i = [\mathbf{p}(\mathbf{x}_{i,1}) \quad \mathbf{p}(\mathbf{x}_{i,2}) \quad \cdots \quad \mathbf{p}(\mathbf{x}_{i,N_i})]^t$$

The global matrix for all C classes is obtained by concatenating all the individual Mi matrices such that:

$$\mathbf{M} = [\mathbf{M}_1 \quad \mathbf{M}_2 \quad \cdots \quad \mathbf{M}_C]^t$$

The training problem reduces to finding an optimum set of weights, w, that minimizes the distance (in this case the in the L2 sense) between the ideal outputs and a linear combination of the polynomial expansion of the training data such that:

$$\mathbf{w}_i^{opt} = \arg\min_{\mathbf{w}} \|\mathbf{M}\mathbf{w} - \mathbf{o}_i\|_2$$

where $\boldsymbol{o_i}$ represents the ideal output comprised of the column vector whose entries are N_i ones in the rows where the *ith* class's data is located in M and zeros otherwise. The weights (identification models) $\boldsymbol{w_i^{op}}$ can be obtained explicitly (non iteratively) by applying the normal equations method [9] such as

$$\mathbf{M}^t\mathbf{M}\mathbf{w}_i^{opt} = \mathbf{M}^t\mathbf{o}_i$$

If we define:

$$\mathbf{R}_j = \mathbf{M}_j^t\mathbf{M}_j, \quad \mathbf{R} = \sum_{j=1}^{C}\mathbf{R}_j, \text{ and } \mathbf{m} = \mathbf{M}_i^t\mathbf{1}$$

this will yield:

$$\mathbf{w}_i^{opt} = \mathbf{R}^{-1}\mathbf{m} \quad (4)$$

In the recognition stage when an unknown feature vector, x, is presented to all C polynomial networks, the vector is expanded into its polynomial terms ***p(x)*** (similar to what was done in the training phase) and its class, c, is determined such that

$$c = \arg\max_i \mathbf{w}_i^{opt}\mathbf{p}(\mathbf{x}) \quad (5)$$

2.1 Simulations with Neural Networks

In our simulations, we used a set of data for 70 different patients. Sample of the data used is shown in Table (1). The terms; STI stands for short term insulin , MTI for midterm insulin, LTI for long term insulin. In the columns for exercise and meal, "1" stands for "yes" and "0" stands for "no". The terms PGL stands for present glucose level and NGL stands for next glucose level. The period of time is the minutes between two consecutive measurements of the glucose level in blood. However, we normalized data before training ending up with 0 mean and unity standard deviation. We did spectral component analysis and eliminated all components less than 0.1% of the variations. The components of a training vector in our data were the PGL, STI , MTI, time period, and meal. We eliminated the all "1" exercise input, the all "0" postprandial input, and the all "0" LTI input. These inputs have no effect since they do not contribute to the variation of the output as they are always kept constant to a single value. The single output of our model has a target of the NGL. This NGL is measured after the given time period of time. We had data for more than 70 patients with total of more than 30,000 samples of input/target training pairs. The training process itself is equivalent to a nonlinear regression process between the normalized inputs (spectral components) and the normalized targets. When training is complete, the output of the neural network is un-normalized in a reverse process for the principal components normalization stage that was implemented before training. The un-normalized data is then passed

Table. 1: Sample of patients data used for modelling

PGL mg/dL	STI U	MTI U	LTI U	Exercise	Meal	Postprandial	Time period (minutes)	NGL mg/dL
100	9	13	0	1	0	0	478	119
119	7	0	0	1	1	0	343	123
123	0	0	0	1	1	0	524	216
216	12	13	0	1	1	0	561	211
211	7	0	0	1	1	0	869	257
257	11	13	0	1	0	0	600	129
129	7	0	0	1	1	0	867	239
239	14	14	0	1	1	0	558	129
129	0	0	0	1	1	0	299	340

through a linear regression stage. The linear regression is implemented between the un-normalized outputs of the neural network and the actual targets taken from the data files (NGL). The linear regression reflects the degree of accuracy and correctness of the neural network predictions.

The training data were accessed as follows; for every consecutive four training points, the first and third point are used for training, the second point is used for testing, and the fourth point is used for validation. Then, the process is repeated for the whole set of data. Of course, during testing and validation there is no learning (training), only nonlinear regression through the neural network followed by a linear regression stage between targets and un-normalized outputs to measure accuracy of prediction.

It should be mentioned here that what is being done in this work is some kind of system identification [13], [14]. Our ultimate goal is to find some general parameters that govern the behavior of the glucose levels in diabetics. When some quantity of medication is investigated its crucial to search for a general theoretical model that can be used to help in testing the effect of that medication. Models such as the ones we present here can be used in giving a theoretical hint about the effect of the insulin in diabetics. These models can be further used in building insulin controllers that automatically insert the proper amount of insulin and work as regulator control for a required level of glucose in blood.

2.2 Simulations with Polynomial Network Model

The PN model we explained earlier is used to model the data of the 70 patients. This model architecture has one neuron at the output layer, see Figure.1. The number of neurons (units) at hidden layer starts with one, then two , and goes up as long as the error values did not reach the given criteria. The PN model (which could be considered as special type of neural networks) number of neurons at the output layer equal to the discrete ranges of insulin injections. The inputs are polynomialized , as we will see later, and then are treated as feature vectors that require classification to the right level of output class. This process is equivalent to a nonlinear layer in standard neural networks. The output layer only contains the weights that are associated with each class. Each neuron at the output layer is associated with a class as target. The error is calculated as the sum of squares between the output and the target divided over the sum of target values (in order to give percentage as shown in Table. 2.

The model we have here could not learn to predict correctly the next values of glucose levels (NGL). As a result of the previous experiments, PN's are only good "mappers" , as evident from Table. 2. results.

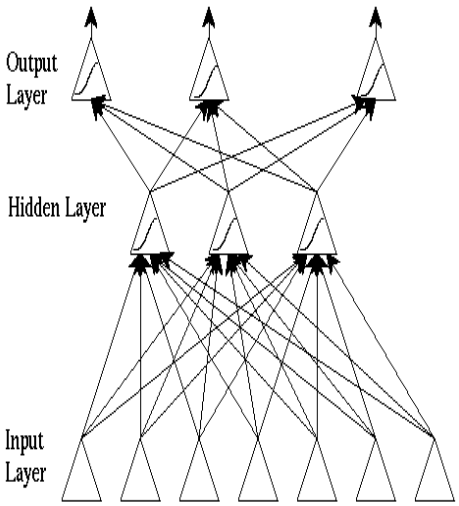

Figure. 1. A scheme of feed forward Neural Network (NN) that could be used with Levenberg-Marquardt (LM) training algorithm or even PN's under some assumptions.

2.3 Simulations with the Levenberg-Marquardt (LM) NN Model

In this model, we used 5 hidden units and one output unit. Adaptive parameters are used in calculating adjustments in weights and biases [15], [16]. Error back propagation algorithm in conjunction with Levenberg-Marquardt (LM) optimization [11] is used. This usually results in fast but memory consuming training. Figure. 2. shows graphs for training, testing, and validation. The training data is prepared in a manner similar to the previous method. The testing and the validation points in the graph are done by passing the inputs through the neural network only without any modifications for weights. The mean square error , which is the performance criteria, is calculated according to the difference between the target and the output of the neural network. It is clear from Figure. 2. as training error goes down, the testing and validation error also goes down.

Table2: Sample of PN's error rates.

group of observations	training error	testing error
1	14.6%	20.6%
2	16.7%	23.1%
3	21.1%	26.7%
4	22.3%	24.7%
5	12.5%	16.6%
6	10.5%	15.7%
7	11.56%	14.5%
8	16.8%	19.8%
9	21.6%	30.3%

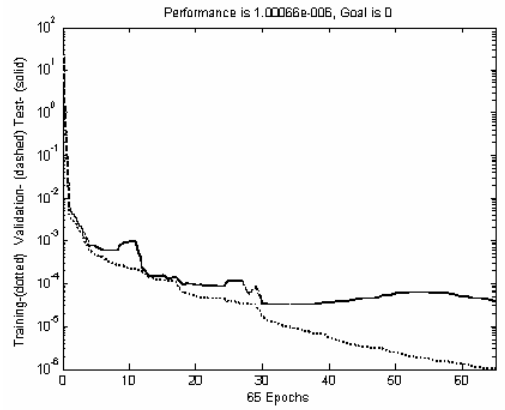

Figure. 2: The error versus Training/Validation/Testing epochs for Levenberg-Marquardt neural network.

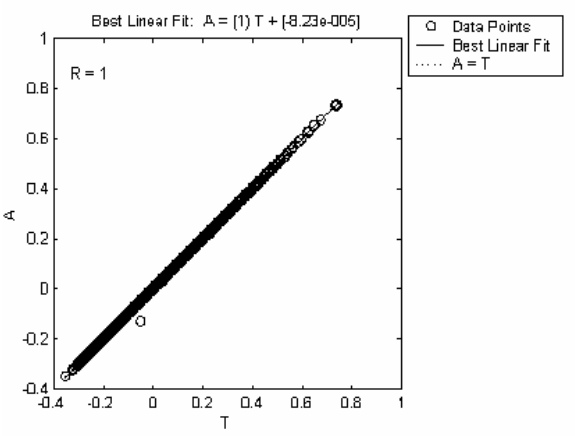

Figure 3. Linear regression of LM NN simulations

Figure. 3. shows a linear regression for the whole set of data. Although, around half of the data is only used in the training, the linear regression for the whole set of data is excellent. Also, note that, the linear regression is an outside process used only to map the normalized output of the neural network with the actual target data. However, the whole process of testing and validation is based on non linear regression. Neural networks are highly nonlinear by nature. The results demonstrate the ability of this type of networks to model the whole set of data. The neural network, here, could capture, identify, and generalize the insulin/glucose dynamics for the samples of the 70 patients with high accuracy. The normalization process for the raw inputs/targets has great effect on preparing the data to be suitable for the training. Without this normalization training the neural networks would have been very slow.

3 Conclusions and Discussions

RBF networks and Back propagation Feed forward networks have been applied with success to function approximation problems [17]. However, PN's were mainly used for pattern classification and recognition problems [20],[21], In this work, we propose using this type of networks in , a more like, regulator control problem. The idea as a whole is a decision making problem, in which a group of previous observations for the patients state variable are used to approximate a decision value for the amount of Insulin required. As shown in Table. 2., It is clear that testing error rates and even training error rates are higher than the acceptable. Again, LM NN is proved to be superior over PN's for this type of problems. The PN's even gave worse results when compared with Radial Basis Function Networks of our previous work [20]. The solutions derived by PN's come from numerical solutions of the polynomialized inputs matrices. Therefore, there is not much flexibility and adaptation in this method to catch up the severe nonlinearity and time dependency of this problem. The weakness could be in the mapping process in which the inputs were polynomialized. This process is similar to the kernel functions used by RBF's, however it is still less flexible in shape and works much more in general scale than the very local kernel functions used by the RBF network. The LM NN, on the other hand, adjusts all the parameters of the network at every training sample and hence, all parameters of the network contribute to the generation of the output concurrently. This would give LM NN more ability to create a global fit for data. Moreover, this collective behavior reduces the size of the network to much smaller size than that for PN. As a result, it is more advantageous to use LM NN when data is "expensive" (i.e. not abundant) and when data is complex. While it is advised to use PN's (or RBF networks) when the data is cheap or plentiful like in adaptive control or some signal processing applications [19]. PN's have the advantage of being fast in training especially when number of classes is small. As explained earlier, PN's has a "single shut" solution. It may suffer from ill-conditioned cases if the matrices were not singular, but this is rare to happen since inputs vary significantly. LM NN training process is more complicated and time consuming.

If we try to relate the results we have with the nature of data we are dealing with, it is fair to conclude that the nature of data we have is not an PN type of data. The target for training, which is the NLG, is not only a function of current state of patient and of the amount and type of insulin she/he just has, but also it is dependent on previous states of the patient and on previous medications she/he already has. The LM NN model is a successful method to identify and capture those dynamics. Some other techniques for modeling are based some on conceptual mathematical modeling followed by standard numerical optimization to approximate the model parameters (least squares method for example). However, in this paper we are more interested in Artificial Intelligence-based models and, in particular, in Neural Networks (NN's). Moreover, we presented two techniques of NN, one is more successful in classifying features. While (LM NN) has a more global strategy at which all hidden neurons participate in generating the output for some input or stimuli. As a matter of fact, NN proved to be a potentially good modeling tool for such type of problems, and that is the bottom line for this work. But we do not advice to use PN's in modeling of dynamical problems due to the limited success we had. We have decided even not to apply linear regression for

the outputs and the targets in the case of PN's due to the apparent unsuccessfulness.

Future work will include designing neural based controllers to regulate the level of glucose in blood based on those NN plant models. We hope that these neural network based techniques will add a little knowledge toward the understanding of insulin/glucose dynamics.

4 References

[1] E.Carson et al, , 1992 "A Spectrum of Approaches for Controlling Diabetes", IEEE Transactions on Control Systems, No.12.

[2] Pank, Klaus;Jurgens, Clemens; et al; , 1998 "Predictive neural networks for learning the time course of blood glucose levels from the complex Interaction of counter regulatory hormones" Neural Computation, Vol. 10 Issue 4.

[3] Bergman RN, Ider YZ, Bowden CR, Cobelli C, 1979: Quantitative estimation of insulin sensitivity. Am. J. Physiology 236:E667-E677.

[4] Lehmann ED, Hermanyi I, Deutsch T. , 1994 Retrospective validation of a physiological model of glucose-insulin interaction in type 1 diabetes mellitus. Med Eng Phys16:193-202, 1994 , Trans. Blamed Eng 41:116-124.

[5] Naylor JS, Hodel AS, Albisser AM, Evers JH, Strickland JH, Schumacher DA, 1997: Comparison of parameterized models for computer-based estimation of diabetic patient glucose response. Med. Inform. 22:21-34.

[6] Bremer, Troy; Gough, David A. ;1999 "Is blood glucose predictable from previous values? " Diabetes, Vol. 48 Issue 3.

[7] Hassoun M. H. ,1995 " Fundamentals of Artificial Neural Networks", MIT Press, Cambridge, Mass.

[8] Sturis J, Polonsky KS, Mosekilde E, Van Cauter E, 1991: Computer model for mechanisms underlying ultra oscillations of insulin and glucose. Am. J. Physiology 260:E801-E809.

[9] Andreassen S, Benn JJ, Hovorka R, Olesen KG, Carson ER, 1994: A probabilistic approach to glucose prediction and insulin dose adjustment: description of a metabolic model and pilot evaluation study. Computer Methods Programs Biomedical 41:153-163.

[10] Ferrannini E, Smith JD, Cobelli C; Toffolo G, Pilo A, DeFronzo RA, 1985: Effect of insulin on the distribution and disposition of glucose in man. J. Clinical Invest 76:357-364.

[11] Cobelli C, Toffolo G, Ferrannini E, 1996: A model of glucose kinetics and their control by insulin, compartmental and non compartmental approaches. Math Biosciences 71:291-316.

[12] Rumelhart, D. E., McClelland J. L., and the PDP Research Group, 1986. "Parallel Distributed Processing: Exploration in the Microstructure of Cognition." Vol.1. MIT Press, Cambridge, Mass.

[13] T. Kohonen, , 1993, "Self-organizing maps: Optimization approaches, in Artificial Neural Networks" ,T. Kohonen, K.Makisara, O.Simula, and J.Kanga, eds., pp..1147-1156. IEEE, New York.

[14] B. Eisenstein and R.Vaccaro, , 1982"Feature Extraction by System Identification," IEEE Trans. on Systems, Man, and Cybernetics, vol.SMC-12, No. 1, pp.42-50.

[15] Jefferies, C. , 1991" Code Recognition and set selection with neural networks" Boston, Birkhauser.

[16] Kosko, B. ed. , 1991 "Neural Networks for Signal Processing." , Prentice-Hall.

[17] Broomhead, D.S., and Lowe, D. Multivariate Function Interpolation and Adaptive Networks, Complex Systems, 2, 321-355.

[18] Kohonen, T. , 1983 "Self-Organization and Associative Memory," Springer-Verlag Series in Information Sciences 8.

[19] Lee, Y. , 1991 "Handwritten Digit recognition Using k- nearest Neighbor, Radial-basis Functions, and Back-propagation Neural Networks, Neural Computation, 3(3), 440-449.

[20] Abu Zitar, RA, 2004, " Towards Neural Networks Model for Insulin/Glucose in Diabetics," accepted at *the International Journal of Computers and Information Systems*.

[21] K. Fukunaga, 1990, Introduction to Statistical Pattern Recognition. Academic Press.