# Towards Neural Network Model for Insulin/Glucose in Diabetics

*Raed Abu Zitar*
College of Information Technology
Philadelphia University
Jordan

**Abstract:** *In this work we look 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 Radial Basis Function (RBF) neural network. 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. This model can be potentially 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.*

**Keywords:** *Levenberg-Marquardt Neural Network, Radial Basis Function Neural Network, Diabetics, Insulin.*

***Received:*** November 21, 2003 | ***Revised:*** February 13, 2004 | ***Accepted:*** April 20, 2004

## 1. Introduction

Diabetes 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 Model

Feed-forward *neural networks* have been extensively used to solve many kinds of problems. It is being applied in a wide range of areas covering subjects such as prediction of temporal series, structure prediction of proteins, and speech recognition. One of the fundamental properties making these *networks* useful is their capability to learn from data. Through synaptic modifications, the neural network is capable of obtaining a new structure of internal connections that is appropriate for solving a determined task. In this work, we use two different types of neural networks; the LM NN model and the radial basis function NN model. 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 ]. This technique is known for fast convergence toward "closest" local minimum and can escape shallow local minima. On other hand, the Radial Basis Function Neural Network (RBF NN) has also a feed forward structure consisting of a single hidden layer of ***J*** locally tuned units which are fully interconnected to an output layer ***L*** linear units . Each hidden unit output is obtained by calculating the "closeness" of the input to a multi-dimensional point (Mj) associated to every neuron (unit) ***J*** . Here, each neuron has its unique multi-dimensional point. The output of the hidden unit (neuron) resulting for an input **x** is given by:

$$Z_j(\mathbf{x})=K(\|\mathbf{x}-\mathbf{M}_j\|/\sigma^2_j) \qquad (1)$$

where K is strictly positive radial symmetric function (kernel) with a unique maximum at its center Mj and which drops off rapidly to zero away from the center. The parameter $\sigma_j$ is the width of the receptive field in the input space for unit J. This implies that the Zj has an appreciable value only when the "distance" $\|\mathbf{x}-\mathbf{Mj}\|$ is smaller than the width $\sigma_j$ . Given an input vector x, the output of the RBF network is the L-dimensional activity vector whose *l*th component is given by

$$Y_l(x)=\sum_{j=1}^{L} w_{1j}Z_j(x) \qquad (2)$$

where $w_{1j}$ is the weight connecting the hidden layer with the output layer, see Figure (1) please.

The LM NN model, see Figure (2) please, is a regular back propagation model but with merits added to accelerate the learning process as all shown in [11,12].

## 3. Simulations with Neural Networks

In our simulations, we used a set of data for 70 different patients [7]. 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 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.

### 3.1 Simulations with RBF NN Model

The RBF model we explained earlier is used to model the data of the 70 patients. This model architecture has one neuron at the output layer. 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 RBF NN number of hidden neurons that we settled on is equal to the size of the training set which happened to be 1600 points (excluding points used for testing and validation). The parameters that govern training here are the numbers of hidden units, the location of them and their associated widths. Generally speaking, there is no formal method for specifying the required number of hidden units. The width we used was very small (around 0.005). The center of the RBF's is equal to the coordinates of the training vector. This in order to achieve smoother interpolation of points. However, this resulted in the high number of neurons at hidden layer. Training of RBF NN is based on gradient descent methods [15 ], [16 ]. The learning rule adjusts the weights that connect the hidden units with the output neuron. As shown in Figure (3), the training process was slow at the beginning then went down quickly at last stages. Although, the training error is very low and close to preset goal, the linear regression for the testing and validation of data show that the RBF network could not capture well the PGL/NGL dynamics. The performance at testing and validation points is very unacceptable, see Figures (4), (5), (6) and (7). It is clear from those figures that the RBF network has good performance only for the group of data used in training. There is no interpolation neither extrapolation abilities demonstrated by the linear regression figures. 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, RBF NN's are only good "memorizers" since it captured well the training data only as shown in Figure(3).

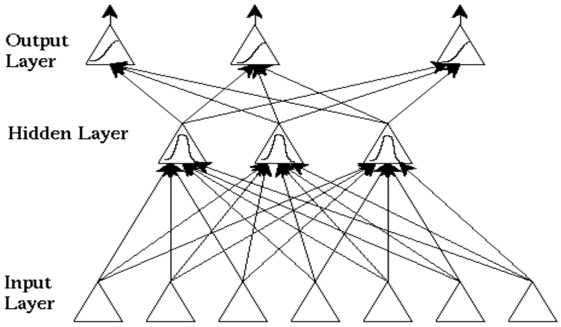

**Figure 1:** A scheme of feed forward Neural Network (NN) using Radial Basis Functions (RBF) at the hidden layer.

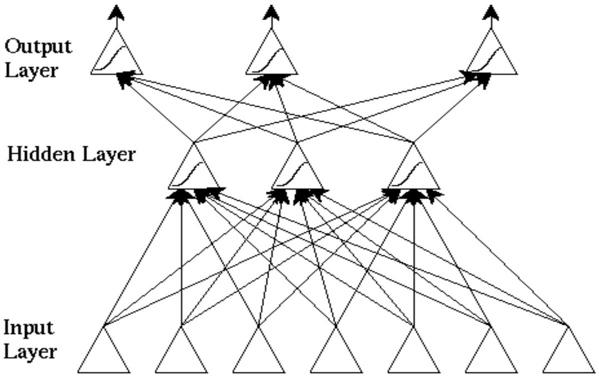

**Figure2.** A scheme of feed forward Neural Network (NN) that could be used with Levenberg-Marquardt (LM) training algorithm.

### 3.2 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 (8) 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(8) as training error goes down, the testing and validation error also goes down. Figure (9) 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.

## 4. Conclusions and Discussions

RBF networks have been applied with success to function approximation problems [17 ]. That was what gave us motivation to use RBF networks in modeling the glucose/insulin dynamics. However, on difficult approximation problems, RBF need additional stages other than the ones we used in this work. A stage for assigning the centers of the radial basis function is needed, other than the uniform distribution of centers according to the training vectors that we used. Clustering techniques [ 18] can be used to find regions where data is concentrated and, therefore, use narrower width functions. RBF NN also requires much more data than the ones used in our work to achieve similar performance to that the LM NN model has achieved. The nature of the RBF networks is “local”. By this we mean that the network only responds for particular inputs that are within the radial basis function active region. Values out of the “specific radial basis function region” will not stimulate that function. Therefore, RBF networks are not that sensitive to history or previous inputs. However, preprocessing is crucial to set up the RBF NN. We do admit that more advanced techniques to estimate the centers of the RBF’s and their spreads would probably upgrade the RBF NN’s performance. 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 RBF NN. 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 RBF NN when the data is cheap or plentiful like in adaptive control or some signal processing applications [19]. RBF networks have the advantage of being fast in training especially when number of radial basis functions needed is small. As explained earlier, RBF NN has a single stage of adjusting the weights if the centers of the radial basis functions is assigned. 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 RBF NN 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 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 dependent on a local response of certain neurons (the hidden neuron with the RBF that responds primarily to specific inputs ) and the other (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.

Future work will include enhancing the estimation techniques for the parameters of the RBF NN, in addition to 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.

## References

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

2. Prank, Klaus; Jurgens, Clemens; et al ;” 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, 1998.

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

4. Lehmann ED, Hermanyi I, Deutsch T. Retrospective validation of a physiological model of glucose-insulin interaction in type 1 diabetes mellitus. *Med Eng Phys* 16:193-202, 1994 , *Trans. Blamed Eng* 41:116-124, 1994 .

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

6. Andreassen S, Benn JJ, Hovorka R, Olesen KG, Carson ER: 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, 1994 .

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

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

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

10. Bremer, Troy; Gough, David A. “Is blood glucose predictable from previous values? “ *Diabetes*, Vol. 48 Issue 3 ;1999.

11. Hassoun M. H. “ Fundamentals of Artificial Neural Networks”, MIT Press, Cambridge, Mass.,1995.

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

13. T. Kohonen, "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, 1993. 1996.

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

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

16. Kosko, B. ed. “Neural Networks for Signal Processing.” , Prentice-Hall, 1991.

17. Broomhead, D.S., and Lowe, D. Multivariate function interpolation and adaptive Networks, *Complex Systems*, 2, 321-355.

18. Kohonen, T. “Self-Organization and Associative Memory,” Springer-Verlag *Series in Information Sciences* 8, 1983.

19. Lee, Y. “Handwritten digit recognition using k-nearest neighbor, radial-basis functions, and back-propagation neural networks, *Neural Computation*, 3(3), 440-449, 1991.

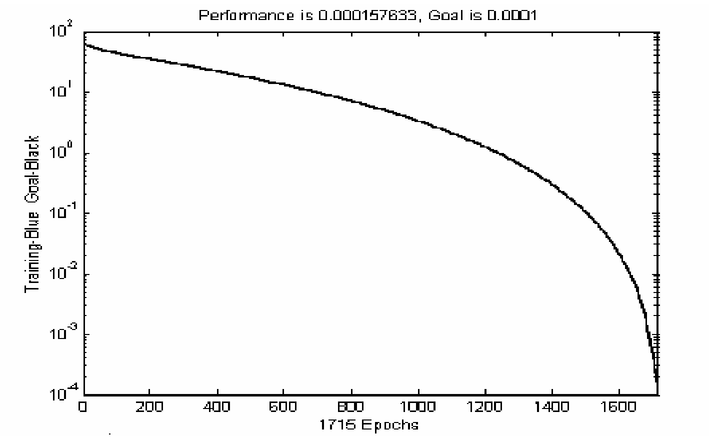

**Figure 3:** The error versus learning epochs for RBF neural network.

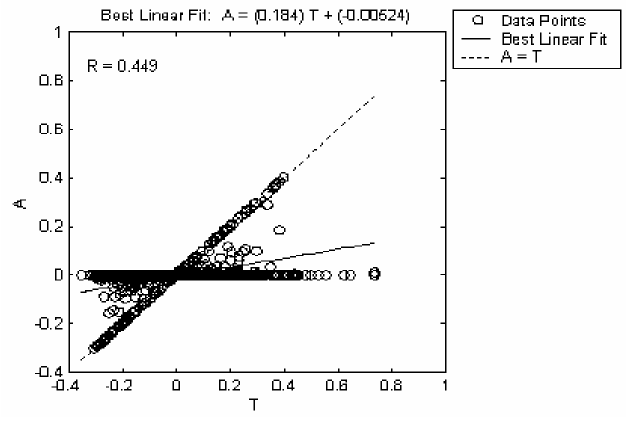

**Figure 4:** The Testing output/target linear regression results for RBF network.

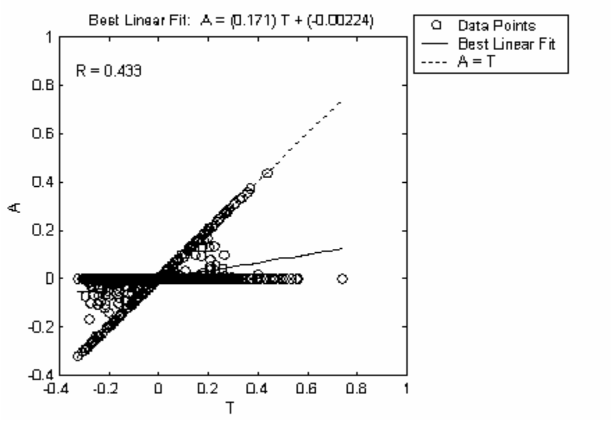

**Figure 5:** The training output /target linear regression results for RBF neural network.

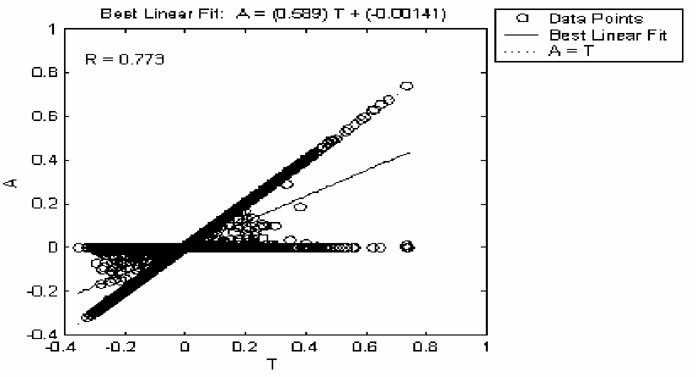

**Figure 6:** The validation output/target linear regression results for the RBF neural network.

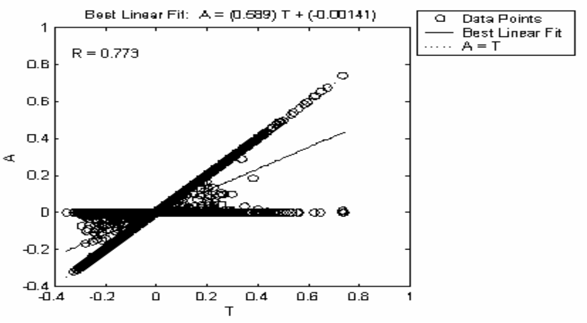

**Figure 7:** The total data set output/target linear regression results for the RBF neural network.

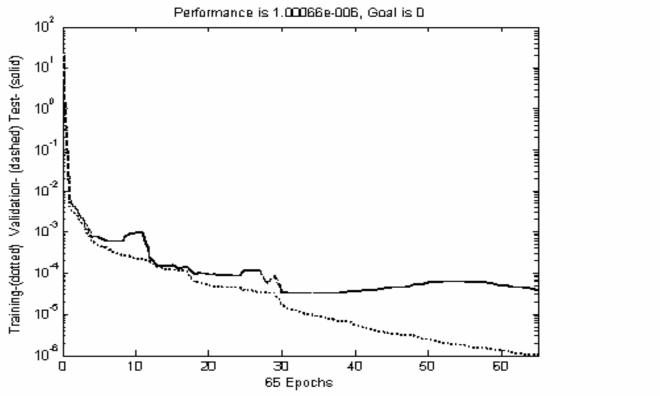

**Figure 8:** The error versus Training/Validation/Testing epochs for Levenberg-Marquardt neural network.

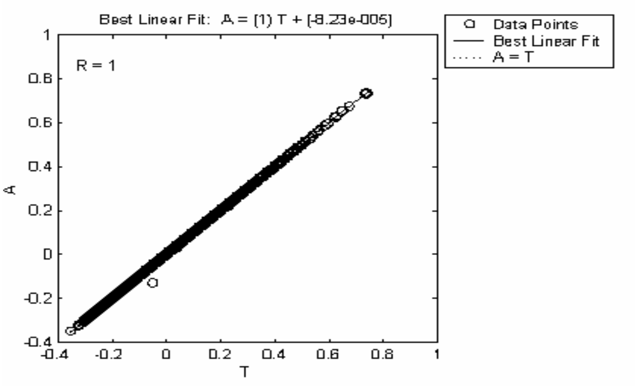

**Figure 9:** The output /target linear regression results for Levenberg-Marquardt neural network.

| PGL<br>mg/dL | STI<br>U | MTI<br>U | LTI<br>U | Exercise | Meal | Postprandial | Time period<br>(minutes ) | NGL<br>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 |

**Table 1:** Sample of patients data used during the modeling.