CHAPTER 8

FORMULATION OF AN ALGORITHM FOR THE PREDICTION OF BLOOD GLUCOSE WITH CUSTOMIZED ANFIS

8.1 INTRODUCTION

Vast metabolic biodiversity of diabetic population and limited knowledge on the complex human physiological process of glucose metabolism makes it difficult to achieve 100% accurate predictions. Since the human glucose metabolism is being influenced by many interdependent factors, a specialized system which is related to the human brain would be needed for the prediction of glucose concentration in blood plasma. This part of work addresses the self tuning capacities of fuzzy systems and computational power of neural networks in predicting the BG concentration from continuous glucose monitoring sensor data time series. This chapter explains about the development of algorithm for the prediction of near future glucose concentrations in blood plasma with customized ANFIS. Performance comparison of the proposed model with previously proposed methods has also been made in this chapter.

8.2 CUSTOMIZATION OF ANFIS

The objective of the present work module is to explore the benefits of the neuro fuzzy systems with an ANFIS for forecasting of glucose concentration in blood plasma, some time slots ahead. The features \{M, V, S, K, A\} extracted from the incoming CGM sensor data, as explained in Feature based Neural Network (FNN) were applied here also. In the current work, ANFIS, usage has been customized in two aspects.
i) First in the learning algorithm of ANFIS, i.e., customized error propagation in backward pass.

ii) Second in finding the optimized minimum number of features for predicting future glucose levels.

Each input was represented in terms of three linguistic terms {Low, Normal, and High}. More terms like very low and very high could have been introduced. However, to have predictions for shorter time periods, less number of linguistic terms would be sufficient. Since the prediction model is dealing with a physiological signal, glucose concentration in blood which was continually varying, the Gaussian function would be the suitable choice for generating the membership values (Atsalakis & Valavanis 2009b).
The training of ANFIS required specifications of the initial FIS, data partitioning and data clustering methods. The FIS consisted of a rule base i.e, a database which defines the membership functions and a reasoning mechanism. The default grid partition and subtractive clustering methods are being adopted for the current research work with ANFIS. If the FIS has been designed based on the past known behavior of the target system, then it could be expected to be able to reproduce the behavior of the target system.

Differential Evolution is an optimization method capable of handling non differentiable, non linear and multimodal objective functions (Price et al 2005, Jin & Branke 2005). The population of candidate solutions is maintained by combining the existing ones by simple operations and keeping whichever candidate solution has the best score or fitness value for the optimization problem. In other words, new trial vectors are produced by adding the weighted difference vector between two population members to a third member. The concept of DE has been depicted in Figure 8.2 the detail of which has been given in proposed algorithm in section 8.4.

**Figure 8.2.** Differential Evolution applied to contributing factors to generate Trial vectors for Optimized Parameters
8.2.1 Forward Pass

The process of converting the crisp values into linguistic terms (membership values) is known as fuzzification. As per the reasons mentioned earlier, Gaussian membership functions were used. In the forward pass, before starting the learning process, the values of premise parameters of the membership functions \{C_{ij}, \sigma_{ij}\} had to be initialized. The values were set in such a way that, the centers of membership functions were equally spaced along the range of each input variable. Usually, the center parameter and the support parameter would be initialized respectively with mean and variance values of inputs (Khan et al 2010, Mitra & Hayashi 2000). Human expertise could be used in determining the parameters of membership functions and later, refined with regression and other optimization techniques (Hong & Chen 1999, Sugeno & Kang 1998). When the values of premise parameters were fixed, then the overall output could be expressed as a linear combination of the consequent parameters.

For the ANFIS model with a two input system, in which each input has been represented in terms of three membership functions, the consequent parameters \(p_1, q_1, r_1, p_2, q_2, r_2, \ldots, p_3, q_3 and r_3\) were estimated as per procedure of Jang (1997). The output in forward pass is given by

\[
f = \frac{w_1}{w_1 + w_2 + \cdots + w_9} f_1 + \cdots + \frac{w_9}{w_1 + w_2 + \cdots + w_9} f_9
\]  

which was linear with the consequent parameters \(p_1, q_1, r_1, p_2, q_2, r_2, \ldots, p_3, q_3 and r_3\).

i.e., \[f = X \theta\] (8.2)

where, \(X\) was the matrix of inputs given by,

\[X = [\bar{w}_1 x_1 + \bar{w}_2 x_2 + \bar{w}_3 + \bar{w}_4 x_1 + \bar{w}_5 x_2 + \bar{w}_6 + \cdots + \]

where, \( T \) was the transpose. If the matrix was invertible, then

\[
\theta = [p_1, q_1, r_1, p_2, q_2, r_2, \ldots, p_9, q_9 \text{ and } r_9]^T
\]  
\[(8.4)\]

By LS technique, the set of unknown parameters could be estimated as,

\[
\theta = (X^TX)^{-1}X^T f
\]  
\[(8.5)\]

This is the parameter estimation in the forward pass.

### 8.2.2 Customized Backward Pass

In the backward pass of ANFIS, the consequent parameters \( \{p_i, q_i, r_i\} \) have been kept fixed and the premise parameters \( \{C_{i,j}, \sigma_{i,j}\} \) would be obtained by the GD algorithm traditionally. In the backward pass, the chain rule procedure is the common method used in error back propagation algorithms and has also been applied for ANFIS. This method includes many steps of partial differentiations, which makes the process cumbersome. Particle Swarm Optimization (PSO) method has also been tried in literature to estimate the premise parameters (Pousinho et al 2012). In the proposed work, the chain rule in GD algorithm has been replaced with a series of multiplications of contributing factors from each layer which are optimized with DE algorithm. These optimized signal vectors were propagated backward to update the premise or antecedent parameters by the customized algorithm as described below.

Let the error,

\[
E = \frac{1}{2} (d_k - f_k)^2 ; \quad k = 1, 2, \ldots, n
\]  
\[(8.7)\]
Where, $d_k$ be the actual target output and $f_k$ would be the predicted output at $k$ th iteration. The error signal vector at layer 4 was specified by

$$
\delta_{04,k} = \frac{1}{2} \left[ ( d_k - F_k ) \times \alpha_k \right]
$$

(8.8)

Where, $\alpha_k$ was the learning rate parameter obtained from the linear combination of inputs $x_1, x_2$.

\[ a_k = g_1 \times x_1 + g_2 \times x_2 \]  

(8.9)

$g_1$, $g_2$ were the coefficients estimated through LS procedure. The contributions obtained from each layer were as follows.

From layer 5 : $F_k$, the output value

From layer 4 : $\delta_{04,k}$, the error signal vector

From layer 3 : $\bar{w}_l$, the firing value

From layer 1 : $\frac{\mu_{x_1}}{c_{l,k}}$, $\frac{\mu_{x_1}}{\sigma_{l,k}}$ contributions of membership values with respect to center parameter and spread parameter respectively.

The contributing factors ( $F_k$, $\delta_{04,k}$, $\bar{w}_l$, $\frac{\mu_{x_1}}{c_{l,k}}$, $\frac{\mu_{x_1}}{\sigma_{l,k}}$ ) from the ANFIS layers were optimized with DE algorithm mentioned above and applied for updating the premise parameters by the iterative mechanism until it satisfy the criteria of minimum RMSE between the predicted value and measured value of BG. The update equations are as follows :

\[
c_{i,k}(n + 1) = c_{i,k}(n) - F_k \times \delta_{04,k} \times \bar{w}_l \times \frac{\mu_{x_1}}{c_{i,k}}
\]

(8.10)

\[
\sigma_{i,k}(n + 1) = \sigma_{i,k}(n) - F_k \times \delta_{04,k} \times \bar{w}_l \times \frac{\mu_{x_1}}{\sigma_{i,k}}
\]

(8.11)

Here, the center parameter $c_i$ and the spread parameter $\sigma_i$ were updated with error signal vector at layer 4 and the contributions from each layer in the backward direction to find the optimum value of antecedent parameters with less number of iterations.
The chain rule is normally applied for differentiating compositions of functions. The chain rule has also been used as a formula for computing the derivative of the composition of two or more functions. The utility of the chain rule is that it turns a complicated derivative into several easy derivatives (Hamdan 2013). The simplest generalization of the chain rule to higher dimensions uses the total derivative. The total derivative is a linear transformation that captures how the function changes in all directions. Because the total derivative is a linear transformation, the functions appearing in the formula can be rewritten as matrices. The matrix corresponding to a total derivative is called a Jacobian matrix, and the composite of two derivatives corresponds to the product of their Jacobian matrices (Schalkoff 1997). This leads to a lengthy process. Hence, instead of applying chain rule, the contributing factors from each layer were extracted and their product was used for updating the premise parameters.

When the model equations are available, one can, in principle, compute the derivative from these equations by directly differentiating the equations and performing the necessary algebraic manipulations. This approach results in an additional set of equations that could be solved in a coupled manner, or in decoupled manner. For lower dimensional linear models, these equations could be solved directly along with the model equations to estimate the derivatives. However, for nonlinear functions, the derivatives are generally more complicated. Further, if the original model has ‘n’ outputs and ‘m’ model parameters, in order to estimate the partial derivatives of all the outputs with respect to all the parameters, the number of equations, required would be maximum (Stuart & Peter 2003). Hence, this approach requires a considerable amount of effort for generating all the necessary model equations. The partial derivatives, by definition, could be approximated by writing proportional value of the function and the constant of proportionality itself is considered as a parameter. Hence, the proposed
customized back propagation would be successful in optimizing the premise parameters.

Since the learning rate was adapted based on the features of input signal, optimum decision of premise / antecedent parameters could be accomplished. The effect of error was varied in updating the peak and support parameters of membership function, so as to have appropriate approximations in the parameter values. Thus, the simple series of multiplications have accomplished the task of complex chain rule procedure for optimized estimation of premise / antecedent parameters. Thereby the adaptive learning rate would avoid the local minima and thereby overcoming the slow convergence problem of GD algorithm.

Hence in the current research work, the prediction of glucose concentration has been carried out with traditional method LS in the forward path and proposed DE based customized method in the backward pass. The performances in terms of RMSE were almost similar and with even reduced training time in the proposed method.

8.3 INPUT SELECTION

The structure of ANFIS model would be implemented based on,

i. a first order Sugeno fuzzy model, so that the consequent part of fuzzy IF-THEN rules would be a linear equation.

ii. the T-norm operator at layer 2 performing the AND operation i.e., the algebraic product.

iii. the type of membership functions.

The five features i.e., the standard first four statistical moments (Mean, Variance, Skewness and Kurtosis) of time varying signal along with approximate entropy have been extracted to track the nonlinear variation of the BG time series. If the ANFIS model had been constructed in such a way to process all the five input features, the system could have been more complex
and computational time would also be of increasing. By grid partitioning procedure, if the ANFIS model had been with five inputs and each input with 3 membership functions, then it would require the implementation of $3^5 = 243$ rules, and a total of $243 \times 6 = 1458$ parameters have to be estimated. Hence, the system would have become highly complicated and time consuming. To address this issue, the input selection procedure has been adopted as prescribed by Jang (1996), to identify the potential inputs that have high priority in tracking the nonlinear BG dynamics.

As mentioned earlier in the previous section, ANFIS employs hybrid learning method that combines the least squares and back propagation algorithm, making use of error signal vectors with respect to modifiable parameters. The least squares method is the major driving force that leads to fast training, while back propagation serves to slow changes in the underlying membership functions. Because of this feature, ANFIS could generate satisfactory results right after the first epoch of training that is only after the application of the LS method (Lin & Lin 1997, Denai et al 2004). Owing to the computational efficiency of the LS method, ANFIS models could be constructed with various combinations of inputs. The input selection procedure is based on the heuristic assumption that the model with smallest RMSE value after one epoch of training has the greater potential of providing minimum RMSE values after extended training.

8.4 PROPOSED ALGORITHM FOR THE CUSTOMIZATION OF ANFIS

Since the CGM data are non linear in nature during the sudden rise or fall of blood glucose levels, the proposed algorithm can track the physiological changes in an efficient manner.

Step 1: Apply moving window to CGM data and extract the features

$$R_k = \{M_k, V_k, S_k, K_k, A_k\}$$ as in FNN method.
Step 2: Selection of dominant features for the prediction model as follows:

- No. of dominant features to be selected ‘n’ should be greater than 1 and less than or equal to N/2, where N is equal to the total number of input features i.e., 5.
- Hence the possible number of features ‘n’ are 2 and 3.
- Form $N^C_n$ number of ANFIS models

Step 3: For each $N^C_n$ ANFIS model, apply one iteration with forward and backward training.

- Fix the premise parameters $\{C_{ij}, \sigma_{ij}\}$ in random and find consequent parameters $\{p_i, q_i, r_i\}$ by LS procedure.
- With the consequent parameters, find the premise parameters by the custom method using the factors contributing from layers while travelling backwards.
- The so obtained factors are optimized with DE method to find the optimum values for premise parameters as follows.

i) Get the contributing factors from layers 5, 4, 3 and 1. Let the factors be a set $X = \{x_1, x_2, x_3, x_4\};$ (8.12)

ii) For each agent $x_i$ in the population, do:

   - Pick 3 distinct agents $x_i, x_j, x_k$ from the population in random.
   - Pick a random index $R = \{1, 2, ... n\}$ (8.13)
     where the highest value of $n$ is the dimensionality of the problem to be optimized.
   - Compute agents’ potentially new position $Z = \{z_1, z_2, ... z_n\}$
     i.e., Mutant vectors by iterating over each $i \in \{1, 2, ..., n\}$ as follows:
     i. Pick $r_i \sim U(0,1)$ uniformly from the open range (0,1).
     ii. $Z_i = x_i + F^*(x_j - x_k)$ (8.15)
If \((i = R)\) or \((r_i < CR)\) let \(w_i = z_i\) else \(w_i = x_i\). Where \(F \in (0,2)\) is the factor or the differential weight and \(CR\) is the cross over Probability.

If \((f(Z) < f(X))\) then replace the agent in the population with the improved candidate solution, i.e., \(x = z\) in the population.

iii) Pick the agent from the population that has the lowest fitness and return it as the best found candidate solution.

iv) Updation of premise parameters by the iterative mechanism until it satisfy the criteria of min RMSE between the predicted value and measured value of BG.

\[
c_{i,k}(n + 1) = c_{i,k}(n) - w_1 * w_2 * w_3 * w_4 \quad (8.16)
\]

\[
\sigma_{i,k}(n + 1) = \sigma_{i,k}(n) - w_1 * w_2 * w_3 * w_4 \quad (8.17)
\]

**Step 4:** Store and Analyze the results of training and testing phases. Find the model with minimum error in training and testing phase.

**Step 5:** Fine tuning of final ANFIS model for prediction.

**Step 6:** Go for prediction of BG.

## 8.5 IMPLEMENTATION OF ANFIS PREDICTION MODEL

The CGM time series was represented as a moving window of length \(n\) equals 6. The above said features \(R_k = \{M_k, V_k, S_k, K_k, A_k\}\) were extracted from each window and applied for training the ANFIS models. The data sets have to be framed separately for training each of the ten models first with three input combinations and in second step with two input combinations. Each data set consisted of input combinational features and the corresponding output for training. In all the cases, every input has been
represented in terms of three membership functions, which were of type gaussian. The gaussian function, which has a spread parameter that controls the behavior of the function, is the most preferred function.

Initially, ANFIS models of three input combinations were endeavored. \( C_3^5 = 10 \) ANFIS models have been constructed and the corresponding RMSE values in training and testing phase were observed. Then \( C_2^5 = 10 \) ANFIS models have been constructed with two input combinations, and their training and testing results were obtained. The performance of three inputs and two inputs combination ANFIS models are listed in Tables 8.1 and 8.2 respectively.

**Table 8.1 Performance of combination of three input features**

<table>
<thead>
<tr>
<th>ANFIS Model No.</th>
<th>Input Combinations</th>
<th>RMSE in mg/dL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training Phase</td>
</tr>
<tr>
<td>1</td>
<td>Mean, Variance, Skewness</td>
<td>3.25</td>
</tr>
<tr>
<td>2</td>
<td>Mean, Skewness, Kurtosis</td>
<td>2.43</td>
</tr>
<tr>
<td>3</td>
<td>Mean, Kurtosis, ApEn</td>
<td>3.67</td>
</tr>
<tr>
<td>4</td>
<td>Variance, Skewness, Kurtosis</td>
<td>4.26</td>
</tr>
<tr>
<td>5</td>
<td>Variance, Kurtosis, ApEn</td>
<td>4.66</td>
</tr>
<tr>
<td>6</td>
<td>Variance, ApEn, Mean</td>
<td>4.85</td>
</tr>
<tr>
<td>7</td>
<td>Skewness, Kurtosis, ApEn</td>
<td>7.12</td>
</tr>
<tr>
<td>8</td>
<td>Skewness, ApEn, Mean</td>
<td>6.16</td>
</tr>
<tr>
<td>9</td>
<td>Kurtosis, Mean, Variance</td>
<td>8.65</td>
</tr>
<tr>
<td>10</td>
<td>ApEn, Variance, Skewness</td>
<td>9.34</td>
</tr>
</tbody>
</table>
Figure 8.3 Performance of three input ANFIS models
(M-Mean, V-Variance, S-Skewness, K – Kurtosis, A-ApEn)

<table>
<thead>
<tr>
<th>ANFIS Model No.</th>
<th>Input Combinations</th>
<th>RMSE in mg/dL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training Error</td>
</tr>
<tr>
<td>1</td>
<td>Mean, Variance</td>
<td>2.4</td>
</tr>
<tr>
<td>2</td>
<td>Mean, Skewness</td>
<td>1.15</td>
</tr>
<tr>
<td>3</td>
<td>Mean, Kurtosis</td>
<td>2.54</td>
</tr>
<tr>
<td>4</td>
<td>Mean, ApEn</td>
<td>2.89</td>
</tr>
<tr>
<td>5</td>
<td>Variance, Skewness</td>
<td>3.6</td>
</tr>
<tr>
<td>6</td>
<td>Variance, Kurtosis</td>
<td>4.71</td>
</tr>
<tr>
<td>7</td>
<td>Variance, ApEn</td>
<td>3.92</td>
</tr>
<tr>
<td>8</td>
<td>Skewness, Kurtosis</td>
<td>4.67</td>
</tr>
<tr>
<td>9</td>
<td>Skewness, ApEn</td>
<td>5.83</td>
</tr>
<tr>
<td>10</td>
<td>Kurtosis, ApEn</td>
<td>6.11</td>
</tr>
</tbody>
</table>
Figure 8.4 Performance of two input ANFIS models

The concept used here is that training many ANFIS models for a single epoch is simpler and more efficient than training a single ANFIS model for many iterations. The training and testing error values of three input and two input combinations were given respectively in Figures 8.3 and 8.4. On comparing the results, it was found that the RMSE values of two input ANFIS models were lesser than that of three input combinations. Hence, out of the five extracted features, two dominant features have to be selected as inputs for the final ANFIS model. By analyzing the results of two input ANFIS models, it has been made clear that the model with mean and skewness as inputs resulted in a minimum RMSE value of 1.025 mg/dL in the training phase. Since skewness is a measure of distributional symmetry and conceptually, it describes the nature of variation in the signal of interest, skewness along with mean had played a good role in predicting the BG values. Therefore, the
combination of mean and skewness associate well with each other in tracking the BG dynamics. The selection process facilitated the removal of noise or irrelevant inputs, the interdependent inputs and has made the model more concise and also condensed the processing time. Hence, this Mean-Skew ANFIS model was selected for the final prediction process after extended training.

8.5.1 Final ANFIS Model

The data set for the final ANFIS model was arranged in such a way that it had three columns and 800 rows. The first and second columns contained the mean and skewness values respectively from the moving windows and the third column contained the corresponding output. 50% of the data sets were allotted for training and remaining 50% for testing.

As per the input selection process, mean and skewness were selected as the highly prioritized, potential inputs in representing the dynamics of BG. These input ranges were partitioned with grid partitioning, and each input was represented in terms of three membership values. The fuzzified values of mean and skewness were denoted respectively as \(\{ML, MN, MH\}\) and \(\{SL, SN, SH\}\) to specify the low, normal and high ranges of features. The respective values of mean and skewness are given in Table 8.3.

<table>
<thead>
<tr>
<th>Input</th>
<th>Crisp Values</th>
<th>Linguistic Term</th>
<th>Fuzzy Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>50 to 150 mg/dL</td>
<td>Low</td>
<td>ML</td>
</tr>
<tr>
<td></td>
<td>100 to 200 mg/dL</td>
<td>Normal</td>
<td>MN</td>
</tr>
<tr>
<td></td>
<td>150 to 250 mg/dL</td>
<td>High</td>
<td>MH</td>
</tr>
<tr>
<td>Skewness</td>
<td>-2 to 0</td>
<td>Low</td>
<td>SL</td>
</tr>
<tr>
<td></td>
<td>-1 to 1</td>
<td>Normal</td>
<td>SN</td>
</tr>
<tr>
<td></td>
<td>0 to 2</td>
<td>High</td>
<td>SH</td>
</tr>
</tbody>
</table>

Table 8.3 Fuzzification of inputs
The membership value distribution for mean and skewness are shown in Figure 8.5.

![Membership distribution of Mean (Input 1) and Skewness (Input 2)](image)

**Figure 8.5  Membership distribution of Mean (Input 1) and Skewness (Input 2)**

The FIS is generally formed with the set of IF-THEN rules. Since the proposed model had two inputs, each with three membership values, the ANFIS would have $3 \times 3$ nine rules. The rules consist of combination of premise parameters by AND product and the corresponding consequent part in linear combination. The consequent parameters are fixed through LS procedure whereas the premise parameters which contribute in membership values are obtained by the customized error propagation method explained above.

If the mean BG level was in the low range and had acquired the membership value $ML$ and skewness had a value in low range and its
corresponding membership value was \( SL \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{ML,SL\} and the consequent parameter values \{\( p_1, q_1, r_1 \)\}.

If the mean BG level was in the low range and had acquired the membership value \( ML \) and skewness had a value in normal range and its corresponding membership value was \( SN \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{ML,SN\} and the consequent parameter values \{\( p_2, q_2, r_2 \)\}.

If the mean BG level was in the low range and had acquired the membership value \( ML \) and skewness had a value in high range and its corresponding membership value was \( SH \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{ML,SH\} and the consequent parameter values \{\( p_3, q_3, r_3 \)\}.

If the mean BG level was in the normal range and had acquired the membership value \( MN \) and skewness had a value in low range and its corresponding membership value was \( SL \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{MN,SL\} and the consequent parameter values \{\( p_4, q_4, r_4 \)\}.

If the mean BG level was in the normal range and had acquired the membership value \( ML \) and skewness had a value in high range and its corresponding membership value was \( SH \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{MN,SH\} and the consequent parameter values \{\( p_5, q_5, r_5 \)\}.

If the mean BG level was in the normal range and had acquired the membership value \( MN \) and skewness had a value in normal range and its corresponding membership value was \( SN \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \{MN,SN\} and the consequent parameter values \{\( p_6, q_6, r_6 \)\}.

If the mean BG level was in the high range and had acquired the membership value \( MH \) and skewness had a value in low range and its
corresponding membership value was \( SL \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \( \{MH,SL\} \) and the consequent parameter values \( \{p_7, q_7, r_7\} \).

If the mean BG level was in the high range and had acquired the membership value \( MH \) and skewness had a value in normal range and its corresponding membership value was \( SN \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \( \{MH,SN\} \) and the consequent parameter values \( \{p_8, q_8, r_8\} \).

If the mean BG level was in the high range and had acquired the membership value \( MH \) and skewness had a value in high range and its corresponding membership value was \( SH \), then the output of the ANFIS model would be computed as the linear combination of fuzzy values \( \{MH,SH\} \) and the consequent parameter values \( \{p_9, q_9, r_9\} \). As a summary the rules are expressed as follows.

1. IF \( Mean = ML \) AND \( Skewness = SL \), THEN \( f_1 = p_1ML + q_1SL + r_1 \).
2. IF \( Mean = ML \) AND \( Skewness = SN \), THEN \( f_2 = p_2ML + q_2SN + r_2 \).
3. IF \( Mean = ML \) AND \( Skewness = SH \), THEN \( f_3 = p_3ML + q_3SH + r_3 \).
4. IF \( Mean = MN \) AND \( Skewness = SL \), THEN \( f_4 = p_4MN + q_4SL + r_4 \).
5. IF \( Mean = MN \) AND \( Skewness = SN \), THEN \( f_5 = p_5MN + q_5SN + r_5 \).
6. IF \( Mean = MN \) AND \( Skewness = SH \), THEN \( f_6 = p_6MN + q_6SH + r_6 \).
7. IF \( Mean = MH \) AND \( Skewness = SL \), THEN \( f_7 = p_7MH + q_7SL + r_7 \).
8. IF \( Mean = MH \) AND \( Skewness = SN \), THEN \( f_8 = p_8MH + q_8SN + r_8 \).
9. IF \( Mean = MH \) AND \( Skewness = SH \), THEN \( f_9 = p_9MH + q_9SH + r_9 \).

The structure of the final ANFIS model used for tracking the BG dynamics is shown in Figure 8.6. The structure of ANFIS implied the formation of consistent rule base, which would be essential for the knowledge interpretation of a neural fuzzy system.
Figure 8.6  Structure of the final ANFIS model

The sample values of premise parameters \( \{C_{ij}, \sigma_{ij}\} \) at the initial stage and at some iterations have been shown in Table 8.4. And the sample values of contributing factors’ values during the customized back propagation process are given in Table 8.5. The equations for updation of premise parameters are obtained through the customized equations as given in the proposed algorithm.
### Table 8.4 Premise Parameters

<table>
<thead>
<tr>
<th>Parameters of First Input (Mean)</th>
<th>Parameters of Second Input (Skewness)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Parameter</td>
<td>Spread Parameter</td>
</tr>
<tr>
<td>$C_{A,1}$</td>
<td>$C_{A,2}$</td>
</tr>
<tr>
<td>Initial Values</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>95</td>
</tr>
<tr>
<td>79.6</td>
<td>92.8</td>
</tr>
<tr>
<td>82.3</td>
<td>97.5</td>
</tr>
<tr>
<td>83.8</td>
<td>98.6</td>
</tr>
</tbody>
</table>

### Table 8.5 Contributing factors in custom ANFIS

<table>
<thead>
<tr>
<th>Initial Values</th>
<th>Sample Iterated Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_k$</td>
<td>$\phi_k$</td>
</tr>
<tr>
<td>82.2048</td>
<td>3.8976</td>
</tr>
<tr>
<td>82.3120</td>
<td>3.3440</td>
</tr>
<tr>
<td>83.1269</td>
<td>0.9366</td>
</tr>
<tr>
<td>82.2474</td>
<td>1.4413</td>
</tr>
</tbody>
</table>
The update equations for the premise parameters i.e., the center and spread parameters of the two inputs are given in Equations from (8.18) to (8.21).

\[
C_{A,k}(n + 1) = C_{A,k}(n) - F_k \times \delta_{04,k} \times \bar{W}_i \times \frac{\mu_{yi}}{\mu_{yi}} \\
\sigma_{A,k}(n + 1) = \sigma_{A,k}(n) - F_k \times \delta_{04,k} \times \bar{W}_i \times \frac{\mu_{yi}}{\mu_{yi}} \\
C_{B,k}(n + 1) = C_{B,k}(n) - F_k \times \delta_{04,k} \times \bar{W}_i \times \frac{\mu_{yi}}{\mu_{yi}} \\
\sigma_{B,k}(n + 1) = \sigma_{B,k}(n) - F_k \times \delta_{04,k} \times \bar{W}_i \times \frac{\mu_{yi}}{\mu_{yi}}
\]

8.6 PERFORMANCE ANALYSIS OF THE CUSTOMIZED ANFIS MODEL

The predictive capability of the final ANFIS model was analyzed in the PHs of 30 and 60 minutes and compared with that of the earlier work; the FNN for Prediction of BG dynamics. Figures 8.7 and 8.8 portray the predictive capability of the proposed method along with the prediction by the FNN method, for a sample CGM time series, in the PH of 30 and 60 minutes respectively. The decreasing level of skewness could be associated with the increasing level of mean suggesting that, larger mean CGM data would tend to reduce skewness of CGM distribution. However, total CGM seemed to have a much larger buffering capacity than mean CGM, thus resulting in lower/less two way correlations. The input selection had resulted in the reduction of training data dimension, which enabled the application of grid partitioning for ANFIS modeling, promoting better data visualization and the scope of ANFIS.
Figure 8.7 Comparison of ANFIS and FNN in 30 minutes PH

Figure 8.8 Comparison of ANFIS and FNN in 60 minutes PH
In the initial stages, for capturing the dynamics of input signal, the ANFIS prediction model had a higher lag values with an average lag of 9.5 and 15.6 minutes respectively in 30 and 60 minutes PHs. After extended training and parameter estimation, the system could track the nonlinear dynamics of the BG data more accurately than the FNN model with an average lag of 4.5 and 6.9 minutes PH respectively. It could be observed from the Figures 8.7 and 8.8 that, the performance of ANFIS was faster in tracking the BG dynamics, which might be due to the advanced adaptation of the learning rate parameter in back propagation. The prediction performance of Final ANFIS when applied to a two days data set has been shown in the Figures 8.9 and 8.10 for 30 and 60 minutes PH respectively.

![Performance of ANFIS-30 minutes PH: Larger Data Set](image)

**Figure 8.9** Performance of ANFIS for a two days\(^6\) data set in 30 minutes PH
Table 8.6 has given the details of prediction error in terms of RMSE in mg/dL and SD. It was observed that, as PH increased, the amount of error was also elevated. However, the error range of ANFIS method was lesser than that of neural network method. The percentage improvement in prediction accuracy for 30 minutes PH was around 65, for 45 minutes PH, around 55 and for 60 minutes PH, it was around 46. The efficiency in prediction decreased with the increase in PH. Still, it seemed appreciable to have an average improvement of 46%, even in the longer PH of 60 minutes.
In 30 minutes PH, the range of RMSE for ANFIS method was 0.5 to 4.7 mg/dL and the average RMSE was 2.4 mg/dL, whereas the FNN provided RMSE values in the range of 1.8 to 11.7 mg/dL and an average RMSE of 4.95 mg/dL. For 60 minutes PH, the ANFIS resulted in the error range of 2.6 to 12.8 with mean RMSE of 8.8 mg/dL. FNN had the range of 11.8 to 23.9 mg/dL with a mean RMSE of 9.26 mg/dL.

**Table 8.6 Sample results for ANFIS prediction**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>PH = 30 Minutes</th>
<th></th>
<th>PH = 60 Minutes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE (mg/dL)</td>
<td>Time Lag (Min)</td>
<td>RMSE (mg/dL)</td>
<td>Time Lag (Min)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upward Trend</td>
<td></td>
<td>Upward Trend</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Downward Trend</td>
<td></td>
<td>Downward Trend</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>4.5</td>
<td>5.3</td>
<td>7.5</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>1.7</td>
<td>2.9</td>
<td>4.6</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>3.9</td>
<td>4.2</td>
<td>5.9</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>2.6</td>
<td>1.5</td>
<td>2.9</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>2.3</td>
<td>3.8</td>
<td>6.9</td>
</tr>
<tr>
<td>Data Set 2</td>
<td>3.6</td>
<td>4.9</td>
<td>6.8</td>
<td>9.3</td>
</tr>
<tr>
<td></td>
<td>3.1</td>
<td>3.7</td>
<td>4.3</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>4.7</td>
<td>5.3</td>
<td>7.7</td>
<td>10.1</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>2.8</td>
<td>4.1</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>2.6</td>
<td>3.5</td>
<td>4.6</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>1.5</td>
<td>2.8</td>
<td>5.2</td>
</tr>
<tr>
<td>Data Set 3</td>
<td>2.4</td>
<td>3.1</td>
<td>4.7</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>4.5</td>
<td>6.1</td>
<td>8.9</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>1.9</td>
<td>3.3</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>2.4</td>
<td>4.5</td>
<td>6.3</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>3.8</td>
<td>5.8</td>
<td>7.1</td>
</tr>
</tbody>
</table>
The performance of the final ANFIS model was evaluated with the traditional GDBP algorithm and with the customized BP algorithm. The parameters considered for the assessment are the average number of iterations in the training phase, average MSE and the percentage of correct predictions in the testing phase. The stopping criterion for the network training has been fixed as the MSE of 0.01. The results of comparison were shown in Table 8.7.

Table 8.7 Performance comparison of ANFIS with GDBP and Custom BP algorithm

<table>
<thead>
<tr>
<th></th>
<th>Average number of Iterations in Training</th>
<th>Average MSE</th>
<th>% of Correct Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANFIS with Traditional GD BP</td>
<td>212</td>
<td>0.4651</td>
<td>65.0</td>
</tr>
<tr>
<td>ANFIS with Customized BP</td>
<td>157</td>
<td>0.0728</td>
<td>96.1</td>
</tr>
</tbody>
</table>

The Prediction accuracy has greatly been improved by the proposed ANFIS method compared with FNN, as depicted clearly in the Figures 8.7 and 8.8 for PH = 30 and 60 minutes respectively. The hybrid learning procedure of ANFIS has facilitated this enhancement. The LS method in the forward pass and the custom made propagation of errors in the backward pass formulated the ANFIS to capture the nonlinear dynamics of blood glucose time series.
The improvement in accuracies of 65% and 46% has been obtained in PHs of 30 and 60 minutes respectively by ANFIS. If the training had been carried out with a vast exploration of time series data, the performance would still have been better. From the Figure 8.11, it could be understood that an average of 65.3% and 45.6% of improvement in prediction was obtained through ANFIS model in 30 and 60 minutes of PH respectively. The results have shown that the predictive capability of ANFIS model was higher than the earlier Neural Network model. An overall improvement of 55.45% in prediction accuracy has been obtained with the proposed method.

The predictive capabilities of the models have also been analyzed in hypo/hyperglycemic ranges and quite an amount of improvement was observed with FNN and ANFIS models. The computational ability of NN and
self tuning capacity of fuzzy systems enabled ANFIS to track the physiological variations of BG optimally than FNN, even in the 60 minutes PH.

8.7 SUMMARY

This chapter had given the details about the development of customized algorithm for prediction of BG with an ANFIS. Prediction of BG with well trained and tested ANFIS would be more accurate than any other prediction models. If implemented after vigorous study, it would be much helpful for the diabetic society for avoiding complications. The assessment of the proposed prediction models in terms of RMSE, Time Lag and CEGA have been made in the following chapter.