Softcomputing method for diagnosing diabetes patients

Smt Lanka Pratibha1*, Smt Y.S. Vijaya Lakshmi1, M N Vamsi Thalatam2

1Department of Computer Science, GVP Degree College (A), Visakhapatnam.
2Department of CSE, School of Engineering, GVP College for Degree & PG Courses, Visakhapatnam

*Corresponding author: E.Mail: pratibha.lanka@hotmail.com

ABSTRACT

Diagnosing diabetes through soft computing approaches is one of the prominent research areas in the present computational and computer science engineering discipline. In the present paper we are trying to give the preliminary decision about the state of the diabetes problem based on the information provided by the patient. The back propagation neural network algorithm is used in this paper to classify whether the patient is normal or suffering with type 2 diabetes mellitus.

Key Words: Algorithm, Back propagation Neural Network, Diabetes, Soft-computing

I. INTRODUCTION

Soft Computing is an amalgamation of methodologies designed to model and enable solution to real world problems like classification and prediction. The aim of the soft computing is to achieve close resemblance with human like decision making by exploiting the tolerance for uncertainty and approximate reasoning. Many of the soft computing methods are existing like, evolutionary computing, artificial neural network and fuzzy logic and their hybrid combinations. In this paper we consider the one of the prominent artificial neural network (ANN) method called back propagation algorithm for classification of patient’s status of diabetes complication. From the decades of research classification is emerged as one of the important decision making tool. Since ANN is an alternative method for classification, we used this algorithm to classify whether the patient is normal or suffering with type 2 diabetes mellitus. We consider the PIMA Indian data set for this approach.

Diabetes Mellitus: Diabetes mellitus (DM) is characterized by hyperglycemia resulting from defects in insulin secretion and/or action. The chronic hyperglycemia of DM can lead to long-term dysfunction, damage or failure of various tissues and organs such as the eyes, kidneys, heart, vascular tissue, and nerves. The DM is classified into two types, based on the primary mode of onset and patho biology of the disease process. The younger onset insulin dependent diabetes mellitus (IDDM) is called type 1 diabetes, and the older onset insulin not dependent diabetes mellitus (NIDDM) is also called type 2 diabetes. In type 1 diabetes mellitus, the development of DM is due to autoimmune destruction of pancreatic β cells with consequent insulin deficiency. These patients are dependent on insulin that needs to be administered from external sources. The patients with type 2 diabetes mellitus are not insulin deficient but show peripheral insulin resistance and consequent hyperinsulinemia. But, over a period of time, these patients also become insulin deficient due to the exhaustion of pancreatic β cells and hence, may eventually become insulin dependent [Undurti N. Das, Wiley-Blackwell 2010].

About the data: The data set used in this study is PIMA Indian women population of Phoenix and Arizona. The data set is a continuous study since 1965 by the National Institute of Diabetes and Kidney diseases.

Eight variables are used to classify the diabetes status of PIMA Indian women. The variables are number of times pregnant (NTP), plasma glucose concentration (PGC), diastolic blood pressure (DBP), Skin fold thickness (SFT), 2-hour serum insulin (SI), body mass index (BMI), diabetes pedigree function (DPF), Age (AGE). A sample data set is shown in table (1). Based on these eight variables the status of diabetes i.e. class 0 (no diabetes) class (suffering with diabetes) is predicted using artificial neural network method called Back Propagation algorithm (BPA).

Artificial Neural Network (ANN): Artificial Neural Network could be defining as an interconnected of simple processing element whose functionality is based on the biological neuron. Simple neuron (Figure 1) introduced by McCulloch and Pitts in 1940s, consists of input layer, activation function, and output layer [Wan Hussain Wan Ishak, 2002]. Input layer receive input signal from external environment (or other neuron). Activation function is the neuron internal states that calculates and sum the input signals. The signals are then transmitted to output layer. The input layer, activation function and output layer in artificial neuron are similar to the function of dendrites, soma and axon in biological neuron.

About Training the Network: Training the network is time consuming, it usually learns after several epochs, depending on how large the network is. Thus, large network required more training time compared to the smaller one. Basically, the network is trained for several epochs and stopped after reaching the maximum epoch. For the same reason minimum error tolerance is used provided that the differences between network output and known outcome are less than the specified value [Pofahl, W. E., Walczak, S. M., Rhone, E., and Izenberg, S. D, 1998]. We could also stop the training after the
network meets certain stopping criteria. During training the network might learn too much. This problem is referred to as over fitting. Over fitting is a critical problem in most all standard NNs architecture. Furthermore, NNs and other AI machine learning models are prone to over fitting [Lawrence, S., Giles, C. L., and Tsoi, 1997]. One of the solutions is early stopping [Sarle, W., 1995], but this approach need more critical intention as this problem is harder than expected [Lawrence, S., Giles, C. L., and Tsoi, 1997]. The stopping criterion is also another issue to consider in preventing over fitting. Hence, for this problem during training, validation set is used instead of training data set. After a few epochs the network is tested with the validation data. The training is stopped as soon as the error on validation set increases rapidly higher than the last time it was checked [Prechelt, L.1998]. Figure 2 shows that the training should stop at time t when validation error starts to increase.

II. BACK PROPAGATION ALGORITHM

To bypass the linear classification problem, we can construct multilayer networks. Back propagation neural network is a multi-layer (ML) neural network as shown in the below figure. The network shown in fig (3) is 2-3-3 configuration network. As shown in the figure 3, the network consists of one input layer, one hidden layer and one output layer. For the problem we considered in this paper there are eight input neurons in the input layer and only one output neuron in the output layer.

Learning Procedure of BP: Randomly assign weights (between 0-1). Present inputs from training data, propagate to outputs, compute outputs O, adjust weights according to the delta rule, back propagating the errors. The weights will be nudged closer so that the network learns to give the desired output. Repeat; stop when no errors, or enough epochs completed.

Pseudo Code BP Algorithm:

Initialise the weights
Repeat
For each training pattern
“Train on that pattern”
End
Until the error is acceptably low

The above code utilized the following equations.

\[
\Delta w_k = cI_k (T_j - O_j) f' (Activation Function) \\
\Delta w_k = cI_k (T_j - O_j) f'(sum)(1-f'(sum))
\]

For the Output unit k, f(sum)=O(k). For the output units, this is:

\[
\Delta w_{j,k} = cH_j (T_k - O_k) O_k (1-O_k)
\]

For the Hidden units (skipping some math), this is:

\[
\Delta w_{i,j} = cH_j (1-H_j) I_i \sum_k (T_k - O_k) O_k (1-O_k) w_{j,k}
\]

This structure is shown in figure (4).
Figure 3. A ML network with 2-3-3 configuration

Figure 4. The three layered network structure

III. RESULTS AND DISCUSSIONS

The neural network used in this paper is p-q-1 structure. There are p inputs (in this problem p=8) and 1 output (here class 0 or class 1). Q is the hidden layer, here only one hidden layer is considered. For training we have used first 80% of the data and for testing purpose we used the remaining 20% of the PIMA Indian data set. Better results are observed by classifying whether a patient is suffering from diabetes or not having any diabetes.

IV. CONCLUSION

In the present research of soft computing many approaches are used for classification problems. But Back propagation neural network has its own importance in simple classification problem like this. It is very easy to understand the concept and implemented for any sort of classification problems. For better results different hybrid models can be proposed in future course of action.

REFERENCES

Wan Hussain Wan Ishak, “Notes on Neural Networks Learning and raining”, www.generation5.org/content/2004/NNTrLr.asp

PIMA Indian Sample Data: (Table 1)

<table>
<thead>
<tr>
<th>NTP</th>
<th>PGC</th>
<th>DBP</th>
<th>SFT</th>
<th>SI</th>
<th>BMI</th>
<th>DPF</th>
<th>AGE</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.00</td>
<td>134.00</td>
<td>74.00</td>
<td>33.00</td>
<td>60.00</td>
<td>25.90</td>
<td>.46</td>
<td>81.00</td>
<td>.00</td>
</tr>
<tr>
<td>10.00</td>
<td>101.00</td>
<td>76.00</td>
<td>48.00</td>
<td>180.00</td>
<td>32.90</td>
<td>.17</td>
<td>63.00</td>
<td>.00</td>
</tr>
<tr>
<td>7.00</td>
<td>142.00</td>
<td>60.00</td>
<td>33.00</td>
<td>190.00</td>
<td>28.80</td>
<td>.69</td>
<td>61.00</td>
<td>.00</td>
</tr>
<tr>
<td>8.00</td>
<td>181.00</td>
<td>68.00</td>
<td>36.00</td>
<td>495.00</td>
<td>30.10</td>
<td>.62</td>
<td>60.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
<td>189.00</td>
<td>60.00</td>
<td>23.00</td>
<td>846.00</td>
<td>30.10</td>
<td>.40</td>
<td>59.00</td>
<td>1.00</td>
</tr>
<tr>
<td>.00</td>
<td>173.00</td>
<td>78.00</td>
<td>32.00</td>
<td>265.00</td>
<td>46.50</td>
<td>1.16</td>
<td>58.00</td>
<td>.00</td>
</tr>
<tr>
<td>8.00</td>
<td>176.00</td>
<td>90.00</td>
<td>34.00</td>
<td>300.00</td>
<td>33.70</td>
<td>.47</td>
<td>58.00</td>
<td>1.00</td>
</tr>
<tr>
<td>12.00</td>
<td>140.00</td>
<td>82.00</td>
<td>43.00</td>
<td>325.00</td>
<td>39.20</td>
<td>.53</td>
<td>58.00</td>
<td>1.00</td>
</tr>
<tr>
<td>7.00</td>
<td>195.00</td>
<td>70.00</td>
<td>33.00</td>
<td>145.00</td>
<td>25.10</td>
<td>.16</td>
<td>55.00</td>
<td>1.00</td>
</tr>
<tr>
<td>7.00</td>
<td>150.00</td>
<td>78.00</td>
<td>29.00</td>
<td>126.00</td>
<td>35.20</td>
<td>.69</td>
<td>54.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>