Proceedings of the National Students' Convention; NSC-2010
National Students' Convention, February 27, 2010
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

Diabetes Detection Using Artificial Neural Networks & Back-Propagation Algorithm

Prashant Kumar , Subin Khullar , Harmeet Singh and Rohit Wadhwa
Department of Information Technology, Bharati Vidyapeeth's College of Engineering, A-4 Paschim Vihar,N.D-63.
E-mail: prashant.talkin@gmail.com, subin.khullar@gmail.com, harmeet2harry@gmail.com, rohit.wadhwa10787@gmail.com

ABSTRACT
This is a project aimed at predicting the chances of diabetes in a person, that whether or not is he/she prone to it. We have used certain parameters namely: number of pregnancies, glucose, BP, skin fold, insulin, body mass index, pedigree and age. The database of 768 patients with these 8 parameters each was taken from National Institute of Diabetes and Digestive and Kidney Diseases. Using neural network feed forward prediction model in conjunction with back propagation algorithm, and given training data set, we predicted whether a subject was likely to have diabetes.

INTRODUCTION
The aim of our project was to use certain key parameters to predict whether a person was suffering or likely to suffer from diabetes. These parameters are:

- Number of pregnancies
- Blood Glucose Level
- BP (Systolic)
- Skin Fold
- Insulin
- Body Mass Index
- Pedigree
- Age

We used a 3 layer neural network with one hidden layer and customizable number of hidden layer neurons with a customization option for the prediction function used. We tried out approximating using 3 functions:

- Binary Sigmoidal

  $f(x)=1/(1+e^{-x})$

- Bipolar Sigmoidal

  $f(x)=2/(1+e^{-x}) - 1$

- Hyperbolic Tangential

  $f(x)=(e^{x} - e^{-x})/ (e^{x} + e^{-x})$

The more the number of hidden layer neurons the more accurate was the result, but it took more computation time. So, a decent trade off value of 20 was chosen. Experimental results showed that binary sigmoidal function was the most accurate of all.

ANN (ARTIFICIAL NEURAL NETWORK) USING BACK–PROPAGATION ALGORITHM
An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

IMPORTANCE OF NEURAL NETWORKS
Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.
2. **Self-Organisation:** An ANN can create its own organisation or representation of the information it receives during learning time.
3. **Real Time Operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. **Fault Tolerance via Redundant Information Coding:** Partial destruction of a network leads to the corresponding

degradation of performance. However, some network capabilities may be retained even with major network damage.

APPLICATIONS OF NEURAL NETWORKS

Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

1. Sales forecasting
2. Customer research
3. Risk management
4. Industrial process control
5. Data validation
6. Target marketing
7. Medicine & Research

ALGORITHM

We used ANN with one hidden layer of 20 neurons (as described above) & adjusted the weights and biases of the links for each level using back propagation algorithm. The weights & biases were randomly initialized in the beginning. There was only one outer layer neuron whose result could be ≈ 1(diabetes) or ≈ 0(no diabetes).

We formally define the algorithm used as:

ANN_BACKPROP(inp_par[])

1. {
2. Initialize nodes and layers to form the neural network.
3. Initialize biases, weights and learning rate.
4. Obtain training samples and their expected output values.
5. While(terminating condition is not true) //it could be a specified no: of iterations or error rate to prevent infinite loop.
6. {
7. Initialize each input layer 'i' node with normalized inp_par[i]
8. For each hidden layer node 'j'
9. {
10. Calculate input to node as value of "sum" given by:

    Sum=∑wt[i][j]*inp_val[i], where wt[i][j] is the weight associated with the link (i,j). Here, 'i' is input layer node and 'j' is the current hidden layer node.

11. Calculate output from node 'j' according to the prediction function used + the 'bias' value for the layer.
12. }
13. For each output layer node 'j'
14. {
15. Calculate input to node as value of "sum" given by:

    Sum=∑wt[i][j]*val[i], where wt[i][j] is the weight associated with the link (i,j). Here, 'i' is hidden layer node and 'j' is the current output layer node.

16. Calculate output from node 'j' according to the prediction function used + the 'bias' value for the layer.
17. }
18. Calculate error for each node 'j' of output layer as:

    Err_out[j]=(Expected output – Obtained output)*Obtained output*(1 – Obtained output).

19. Calculate error for each node 'j' of hidden layer as:

    a. Calculate "sum" as:

       Sum=∑Err_out[i]*wt[i][j], where wt[i][j] is the weight associated with the link (i,j) between hidden layer node 'j' and output layer node 'i'.

    b. Err_hid[j]=(Obtained output)*(1 – Obtained output)*sum

20. // Adjust weights and biases according to the errors calculated by back propagation For each weight wt[i][j] between input and hidden layer

    a. Wt[i][j]=wt[i][j]+lr*Err_hid[j]*(Obtained output at j), where 'lr' is the learning rate

21. For each weight wt[i][j] between hidden and output layer

    a. Wt[i][j]=wt[i][j]+lr*Err_out[j]*(Obtained output at j), where 'lr' is the learning rate

22. Bias_hid=bias_hid+lr*RMS(Err_hid[j]), where Err_hid[j] is the error of a hidden layer node j.

23. Bias_out=bias_out+lr*RMS(Err_out[j]), where Err_out[j] is the error of a output layer node j.

24. }

25. .//Training is complete, display results in user console. Start predicting new samples. Accept inp_par[] from the user and perform steps 7 – 16.

26. The resulting pattern obtained from the output of all the output layer nodes is compared with a pattern sequence to detect a match and the respective class of final output is displayed to the user. The accuracy of prediction will largely depend on the data set given and the parameters entered. It will largely depend on the statistical correlation of the values of the parameters and the respective final output class.

27. }

RESULTS OBTAINED

We trained our neural network with 200 samples from the database and tested it with 50 samples. We obtained specificity of 82.14% & sensitivity of 88.8%. We obtained RMS error rate of 0.019% for the 50 samples tested. (Here we calculated error in terms of percentage error of output from output expected i.e. 0 or 1.) On an average it took about 1275 iterations to converge to the result with learning rate of 0.1 and momentum of 0.9, which was found out to be an optimal value for binary sigmoidal function. The entire project was made in NetBeans IDE 6.8 using java swings API (using JApplet for GUI).

ACKNOWLEDGEMENT

Our sincere thanks to Prof. Dr. N.D. Kaushika for suggesting the problem and for constant expert guidance throughout the project.

FUTURE SCOPE

Future scope of this project lies in the use of supplementary approximation algorithms like genetic algorithm, ant algorithm and fuzzy logic in conjunction with artificial neural networks to get even more accurate results with fewer number of iterations.

CONCLUSION

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture.Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.Perhaps the most exciting aspect of neural networks is the possibility that some day 'conscious' networks might be produced. There is a number of scientists arguing that consciousness is a 'mechanical' property and that 'conscious' neural networks are a realistic possibility.

REFERENCES & BIBLIOGRAPHY

1. An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition.
2. Neural Networks at Pacific Northwest National Laboratory http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html
3. Artificial Neural Networks in Medicine http://www.emsl.pnl.gov:2080/docs/cie/techbrief/NN.techbrief.ht
4. Industrial Applications of Neural Networks (research reports Esprit, I.F.Croall, J.P.Mason)
5. A Novel Approach to Modelling and Diagnosing the Cardiovascular System http://www.emsl.pnl.gov:2080/docs/cie/neural/papers2/keller.wcnn95.abs.html
6. Electronic Noses for Telemedicine http://www.emsl.pnl.gov:2080/docs/cie/neural/papers2/keller.ccc95.abs.html
   An Introduction to Computing with Neural Nets (Richard P. Lipmann, IEEE ASSP Magazine, April 1987)
7. Pattern Recognition of Pathology Images http://kopernik-eth.npac.syr.edu:1200/Task4/pattern.html
8. Developments in autonomous vehicle navigation. Stefan Neuber, Jos Nijhuis, Lambert Spaanenburg. Institut fur Mikroelektronik Stuttgart, Allmandring 30A, 7000 Stuttgart-80.
9. Klimasauskas, CC. (1989). The 1989 Neuro Computing Bibliography. Hammerstrom, D. (1986). A Connectionist/Neural Network Bibliography.
10. DARPA Neural Network Study (October, 1987-February, 1989). MIT Lincoln Lab.
11. Neural Networks, Eric Davalo and Patrick Naim.
12. Assimov, I (1984, 1950), Robot, Ballatine, New York.
13. Learning internal representations by error propagation by Rumelhart, Hinton and Williams (1986).
14. Alkon, D.L 1989, Memory Storage and Neural Systems, Scientific American, July, 42-50.