Decision Support System for Heart Disease Diagnosis Using Neural Network

Niti Guru*
Anil Dahiya**
Navin Rajpal***

THESE days the Artificial Neural Networks (ANNs) have been widely used as tool for solving many decisions modelling problems. The various capabilities and properties of ANN like Non-parametric, Non-linearity, Input-Output Mapping, Adaptivity make it a better alternative for solving massively parallel distributed structure and complex task in comparison of statistical techniques, where rigid assumption are made about the model. Artificial Neural Network being non-parametric, makes no assumption about the distribution of the data and thus capable of "letting the data speak for itself". As a result, they are a natural choice for modeling complex medical problems when large databases of relevant medical information are available. In this paper we use the Neural Network for prediction of Heart disease, Blood Pressure and Sugar. A set of experiments was performed on a sample database of 78 patients' records, 13 input variables (Age, Blood Pressure, Angiography's report etc.) are used for training and testing of the Neural Network. We have suggested supervised network for diagnosis of heart diseases and trained it using Back Propagation algorithm. The System is trained for 78 patients' records. On the basis of this trained data, when unknown data is entered by Doctor, the system will find that unknown data from the trained data and generate list of possible diseases from which patient can suffer. Error made by human being can be avoided in this system; hence the system is more reliable and helps the doctor to take correct decision.

Keywords: *Decision Support System, Neural Network, Artificial Neural Network.*

Introduction

Neural networks are being applied to an increasing large number of real world problems. Their primary advantage is that they can solve problems that are too complex for conventional technologies; problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be defined. In algorithmic approach, the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known, the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. (Neural Networks) In general, neural networks are well suited to problems that people are good at solving, but for which computers generally are not. These problems include pattern recognition and forecasting – which requires the recognition of trends in data. In case of neural network, even for imprecise inputs the network is able to retrieve the desired output or the data that is closest to the desired output. Considering the successful applicability of neural networks in many areas, an endeavour to assess their performance for data retrieval forms the basis for this research work.

Neural Network

(Haykin, Simon) Neural Networks are an information processing technique based on the way biological nervous systems, such as the brain, process information. The fundamental concept of neural networks is the structure of the information processing system. Composed of a large number of highly interconnected processing elements or neurons, a neural network system uses the human-like technique of learning by example to resolve problems. The neural network is configured for a specific application, such as data classification or pattern recognition, through a learning process called training.

Network layers

- (Neural Networks) The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. Figure1 displays the three layers viz., input, hidden, output layer.

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

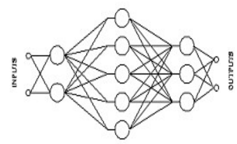

Figure 1: A Three Layer Back Propagation ANN Network

Transfer Function

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- **Linear:** For linear units, the output activity is proportional to the total weighted output.
- **Threshold:** For threshold unit, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.
- **Sigmoid:** For sigmoid units, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than linear or threshold units, but all three must be considered rough approximations.

All learning methods used for adaptive neural networks can be classified into two major categories:

a) Supervised learning, which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimisation of error between the desired and computed unit values. The aim is to determine a set of weights, which minimises the error. According to Scholtes (1991), One well-known method, which is common to many learning paradigms, is the Least Mean Square (LMS) convergence.

b) Unsupervised learning uses no external teacher and is based upon only local information. (Neural Networks) It is also referred to as self-organisation, in the sense that it self-organises data presented to the network and detects their emergent collective properties. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line. We have suggested supervised network for diagnosis of heart diseases and trained it using Back Propagation Algorithm. Back Propagation Algorithm has emerged as the standard algorithm for training of networks under supervised form of learning. The algorithm derives its name from the fact that the partial derivatives of the performance measure with respect to the free parameters (synaptic weights and biases) of the network are determined by back propagating the error signals through the network. In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. (Tsoukalas, 1997) This process requires that the neural network compute the error derivative of the weights. In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the error derivative.

(Fahlman, 1988) The algorithm computes each error derivative by first computing the rate at which the error changes as the activity level of a unit is changed. For output units, the error is simply the difference between the actual and the desired output. To compute the error for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the errors of those output units and add the products. This sum equals the error for the chosen hidden unit. After calculating all the errors in the hidden layer just before the output layer, we can compute in like fashion the errors for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. (Fahlman, 1988) Once the error has been computed for a unit, it is straightforward to compute the error derivative for each incoming connection of the unit. The error derivative is the product of the error and the activity through the incoming connection.

The experiment is based on single input layer, hidden layer, output layer consist of 13 neurons respectively.

Figure 2 showing Multilayer Feed Forward back propagation network. The subscripts I, H, O, V, W denote input, hidden and output neurons, Weight at Hidden Layer, Weight at Output Layer respectively

I_1: Input of input Layer 1

$O_{I1:}$ Output of input Layer 1

$V_{11:}$ Weight between inputs Neuron 1 to Hidden Neuron 1

$W_{11:}$ Weight between hidden Neuron 1 to output Neuron 1

$I_{H1:}$ Input of Hidden Layer 1

$O_{H1:}$ Output of Hidden Layer 1

Io1: Input of Output Layer 1

$Oo_{1:}$ Output of Output Layer 1

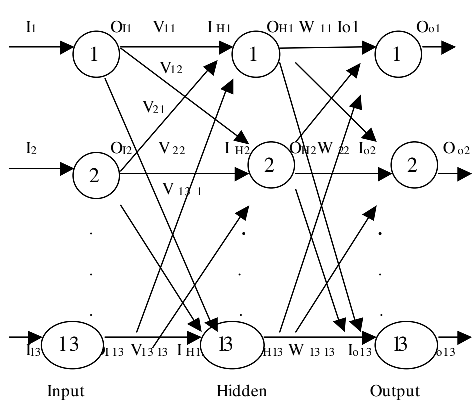

Figure 2: Multilayer feed forward back propagation network

Input layer computation
Equation (1) defines linear activation function; the output of the input layer is input layer. Taking one set of data

$$\{O\}_{I}=\{I\}_{I}. \quad (1)$$

(Rajasekaran, Vinayalakshmi) the hidden neurons are connected by synapses to input neurons and V_{ij} is the weight of the arc between ith input neurons to jth hidden neuron. The input to the hidden neuron is the weighted sum of output of the input neuron to get I_{HK} (K=1 to 13) as

Equation (2) defines Input for Hidden Neuron 1

$$I_{H1} =[V]_{I1} O_{I1} +V_{21} O_{12+.....}+V_{13\,1}O_{I13}. \quad (2)$$

Hidden layer computation
Equation (3) defines the sigmoidal function or Squashed –S function, the output of hidden neuron is given by

$$O_{HK=} 1/(1+e^{-I_{HK}}) \quad (3)$$

K= (1 to 13)

$I_{HK:}$ Input to Hidden Neuron

Equation (4) defines the Input to the output neurons; it is the weighted sum of the outputs of the hidden neurons.

To get I_{O1}

$$I_{O1=}W_{11} O_{H1} +W_{21} O_{H2+.....}+ W_{13} O_{H13}. \quad (4)$$

Output layer computation
Equation (5) defines the Sigmoidal function, the output of the hidden neuron is given by

$$O\, o_{K=} 1/(1+e^{-I_{o1}}). \quad (5)$$

K= (1 to 13)

Where Ook is the output of the Kth output neuron, Iok is the input of the Kth output neuron

Calculation of Error
(Zhang and Moris, 1998) considering any K^{th} output neuron and for the training example we have calculated the output 'O' for which the target output 'T' is known

Equation (6) defines the error norm in output for the Kth output neuron

$$E_r^1 =(1/2)e_r^{\,2} =(1/2)(T-O)^2. \quad (6)$$

Where $E_r^{1} =(1/2)$ second norm of the error in the K^{th} neuron (e^r) for the given training pattern. The square of the error is considered since irrespective of whether error is positive or negative, we consider only absolute values.

Experiment

The System is trained for 78 patients records. On the basis of this trained data, when unknown data is entered by Doctor, the system will find that unknown data from the trained data and generate list of possible diseases from which patient can suffer. In Figure 3 we initialise the input, weights at input, hidden and output layers, and desired output, threshold value, error is a difference between network output and desired output for the training set. Equation (6) is used to calculate error, use (1), (2), (3), (4), (5) to calculate input/output at input layer, hidden layer, output layer and reduce the error by adjusting weights till we achieve tolerance value that is equal to desired output or less than that.

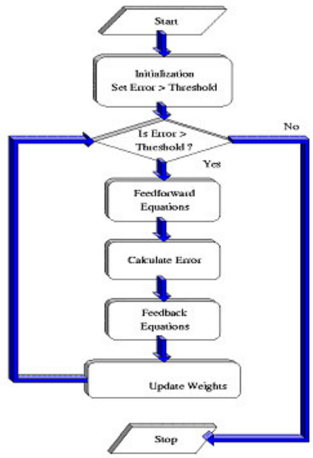

Figure: 3 Flowchart showing Back Propagation Algorithm

The experiment was done using a program written in Met lab for back propagation algorithm.

We started with initialising the weights of the network in random order, which the network modified in the process of reducing the error vector. Training is performed for 78 patient's record, after training simulation was done

Experiments Results

When the system is trained for the record of 78 patients, the success rate for imprecise inputs to retrieve the desired output or the data that is closest to the desired output is 100%

We are considering input for 3 patients, where value for 13 parameters are taken, and values are normalized.

See Table 1

Taking normalized values of 13 inputs

Case1

0.35 0.85 0.88 0.55 0.75 1.13 1.02 0.25 0 0 0 0 0 0.2

Lipid profile is out of range, so it's a case of heart disease, Expected output: 1

Case2

0.62 0.93 1.1 0.96 1.27 0.95 0.70 0.25 0 0 0 0 0 0.21

B.P Disysytolic is out of range, so it's a case of B.P,

Expected output: 0.3

Case 3

0.54 0.85 0.73 1.01 1.37 0.92 0.72 0.25 0 0 0 0 0 0.23

Sugar is out of range, so it's a case of Sugar,

Expected output: 0. 5

Discussion and Conclusion

Error made by human being can be avoided in this system; hence the system is more reliable and helps the doctor to take correct decision.

These improvements could mean increased efficiency, a greater level of accountability and greater visibility of tasks carried out by the various users present in the Disease Diagnosis system.

We have tried to explore the possibility of using Neural Network as an Indexing Function and the results obtained show that the network has immense potential of being used as an indexing function.

We therefore conclude that ANN is a fast alternative to classical statistical techniques for prediction and modelling of experimental data.

It is believed that neural networks will have extensive application to biomedical problems in the next few years.

The benefit of using ANNs is that they are not affected by factors such as fatigue, working conditions and emotional state. Of course, the system doesn't eliminate the need for doctors since a human expert is more reliable. Neural networks will never replace human experts but they can help in screening and can be used by experts to double-check their diagnosis.

References

Fahlman, S.E. (1988a) "*An Empirical Study of Learning Speed in Back Propagation Networks*", Carneige Mellon Report, No CMU-Cs, p.88-162.

Fahlman, S.E. (1988b), "*Learning Variations of Back Propagation – An Empirical Study*", Proc. of the 1988, Connectionist Models Summer School, Morgan Kaufmann San Mater,CA.

Haykin, Simon "*Neural Network – A Comprehensive Foundation*", Second Edition, Low Price Edition, p.1-203.

Neural Networks, http://www.pcai.com/web/ai_info/neural_nets.html

Neural Networks, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

Rajasekaran, S. and Vijayalakshmi Pai, G.A. "*Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications*" Eastern Economy Edition, PHI, p.1-59.

Scholtes, J.C. (1991) "*Neural nets and their relevance for information retrieval*", ITLI Prepublication CL University of Amsterdam Institute for Language Logic and Information Department of Computational Linguistics October, p.50-92.

Tsoukalas, L.H. and Uhrig, R.E. (1997) "*Fuzzy and Neural Approaches in Engineering*", John Wiley & Sons, USA.Zhang, J. and Moris, A.J. (1998) "*A Sequential Learning Approach for Single Hidden Layer Neural Networks*", Neural Networks, Vol.11, No.1, p.65-80.