RESEARCH ARTICLE OPEN ACCESS

Neural Network for Pattern Classification

RashmiAmardeep[1] and Dr.K ThippeSwamy[2]
[1]*Research Scholar, Sri Siddhartha Academy of Higher Education, Tumkur, India*
[2]*Visvesvaraya Technological University, PG Regional Center Mysore, India*

ABSTRACT
Neural Networks (NN) are the most active research and application in the field of pattern classification, using training and testing data to build a model. However, the success of the networks is highly dependent on the performance of the training process and hence the training algorithm. Many training algorithms have been proposed so far to improve the performance of neural networks. The paper gives the study of multilayer perceptron, radial basis function network, and probabilistic neural network. The algorithms are tested on three different datasets and comparisons in terms of accuracy with test data are mentioned in the paper.
Keywords: Multilayer Perceptron, Radial Basis Function Network, Probabilistic Neural Network

Date of Submission: 09-12-2017 Date of acceptance: 28-12-2017

I. INTRODUCTION

ANN is a powerful tool for data analysis and is a non-linear mapping structure. ANN identifies and learns correlated patterns between input datasets and corresponding target value. After training ANN can be used to predict input data. ANN is being recognized in the area of classification and predictions. There are various types of ANN like multilayer perceptron, Radial Basis function, Probabilistic Neural Network and are been discussed. The graphical representation of supervised learning where the weights are adjusted is given in figure 1.

Figure 1: A learning cycle in ANN Model

The phases of ANN are training, validation and testing. The training set is used to train the network. The validation set is used to validate the network, to adjust network design parameters. The test set is used to test the generalization performance of the selected design of neural network.

Advantages of NN are as follows:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self Organization: An NN can create its own organization or representation of the information it receives during learning rate.
- Real time operation: NN computation may be carried out in parallel and special hardware devices are being designed and manufactured which take advantages of this capability.
- 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

II. NEURAL NETWORK CLASSIFIERS

Listed below are few of the neural networks:
1) Multilayer perceptron
2) Radial basis function network
3) Probabilistic neural network.

2.1Multilayer Perceptron

Multilayer perceptron (MLP) are the most common form of a multilayer feed forward network. Figure 2 shows an MLP which has three types of layers: an input layer, an output layer and a hidden layer. Neurons in input layer act as buffers for distributing the input signals X_i (i=1, 2 …n) to neurons in the hidden layer. Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons.

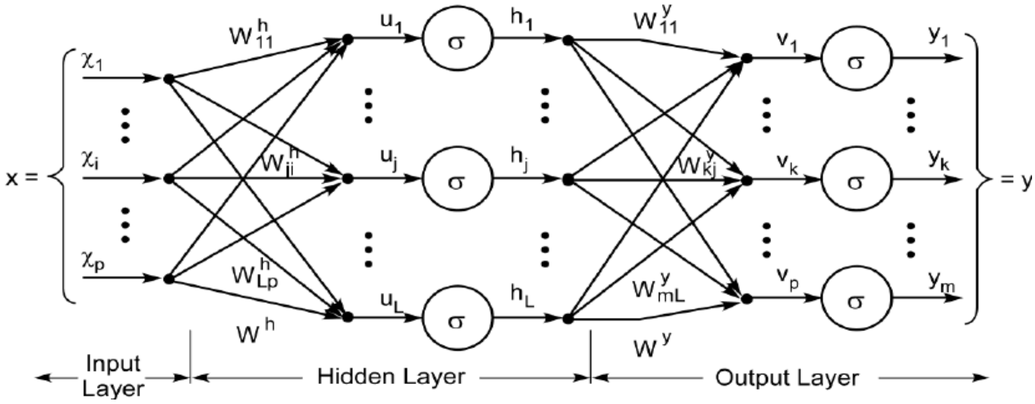

Figure 2: Multilayer Perceptron Network

Each neuron *j* (Figure 2) in the hidden layer sums up its input signals X_i after weighting them with the strengths of the respective connections w_{ji} from the input layer and computes its output Y_j as a function *f* of the sum.

$Y_i = F\ (\sum W_{ij}\ X_i\ )$ ----- (1)

F can be a simple threshold function or a sigmoidal, hyperbolic tangent or radial basis function as given in figure 3, a mesh layered perceptron network

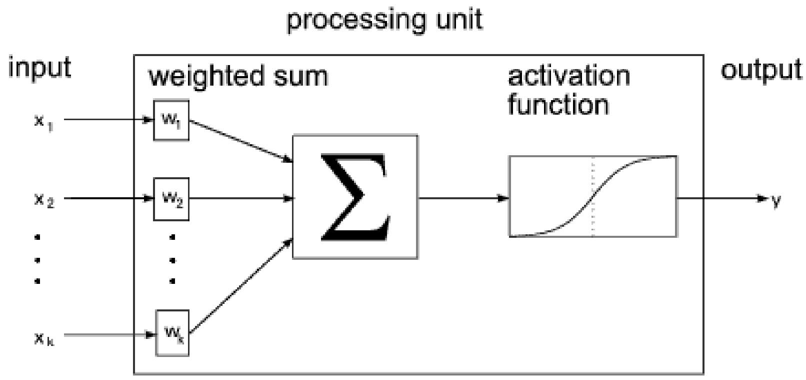

Figure 3: Perceptron Process

The output of neurons in the output layer is computed similarly. The back-propagation algorithm, a gradient descent algorithm, is the most commonly adopted MLP training algorithm.

It gives the change Δw_{ji} the weight of a connection between neurons i and j as follows:

$\Delta\ W_{ij} = \eta\ \delta_j\ X_i$ -----(2)

where η is a parameter called the learning rate and δ_j is a factor depending on whether neuron j is an input neuron or a hidden neuron. For output neurons,

$\delta_j = (\partial f / \partial net_j)(y_j^{(t)} - y_j)$ ----(3)

and for hidden neurons

$\delta_j = (\partial f / \partial\ net_j)(\Sigma_q\ w_{jq}\delta_q)$ ---- (4)

In Eq. (3), net_j is the total weighted sum of input signals to neurons j and $y_j(t)$ is the target output for neuron j. As there are no target outputs for hidden neurons, in Eq. (4), the difference between the target and actual output of a hidden neurons j is replaced by the weighted sum of the δq terms already obtained for neurons q connected to the output of j. The process begins with the output layer, the δ term is computed for neurons in all layers and weight updates determined for all connections, iteratively. The weight updating process can happen after the presentation of each training pattern (pattern-based training) or after the presentation of the whole set of training patterns (batch training). Training epoch is completed when all training patterns have been presented once to the MLP. A commonly adopted method to speed up the training is to add a "momentum" term to Eq.(5) which effectively lets the previous weight change influence the new weight change:

$\Delta w_{ij}\ (i + 1) = \eta\ \delta_j\ X_i + \mu\ \Delta w_{ij}(i)$ ----- (5)

where Δw_{ij} (i + 1) and Δw_{ij}(i) are weight changes in epochs (i + 1) and (i), respectively, and μ is "momentum" coefficient (Jayawardena & Fernando, 1998).

2.2 Radial Basis Function Network

Radial basis function (RBF) network is a feed-forward network. Radial basis function networks consist of two layers: a hidden radial basis layer of S^1 neurons, and an output linear layer of S^2 neurons. Each radial basis layer neuron's weighted input is the distance between the input vector and its weight vector. Each radial basis layer neuron's net input is the element-by-element product of its weighted input with its bias. Each neuron's output is its net input passed through radial basis transfer function.

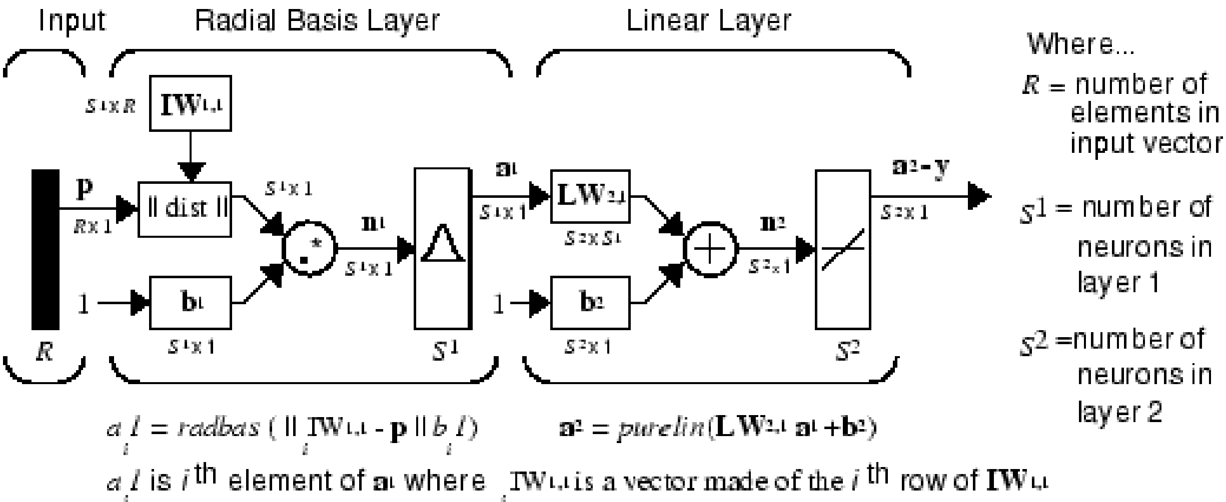

Figure 4: Radial Basis Function Network

Radial basis function network is created iteratively one neuron at a time. Neurons are added to the network until the sum-squared error falls beneath an error goal or a maximum number of neurons have been reached. Design parameter of radial basis function network is spread of radial basis transfer function.

Radial basis networks can be designed with newrb. The transfer function for a radial basis neuron is:

$Radbas(n) = e^{-n2}$ ----- (6)

Radial basis function network consists of three layers as in Fig 4. The input layer has neurons with a linear function that simply feed the input signals to the hidden layer. Moreover, the connections between the input and hidden layer are not weighted. The hidden neurons are processing units that perform the radial basis function. Each unit is mathematically defined as:

$\varphi j\ (x) = \varphi(\| x - xj \|)$, j = 1 ,2, …, n ---- (7)

The j^{th} input data point x_j denotes the center of the radial basis function, and the vector x is the pattern applied to input layer. Selecting the basis function is not crucial to the performance of the network the most common being the Gaussian basis function which is used in this study. It is defined as:

$\varphi_j\ (x) = \exp\ (-1/2\ \sigma\ 2\ j\ (\| x - x_j \| 2))$, j = 1 ,2, …, n --- (8)

The output neuron is a summing unit to produce the output as a weighted sum of the hidden layer outputs as shown by:

$F(x) = \Sigma w_j \varphi_j\ (x)$, j = 1 ,2, …, n ----(9)

2.3 Probabilistic Neural Network

Probabilistic neural network is a multilayered feed-forward network with four layers: Input layer, Pattern layer, Summation layer and Output layer. It is derived from RBF .When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a competitive output layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes.

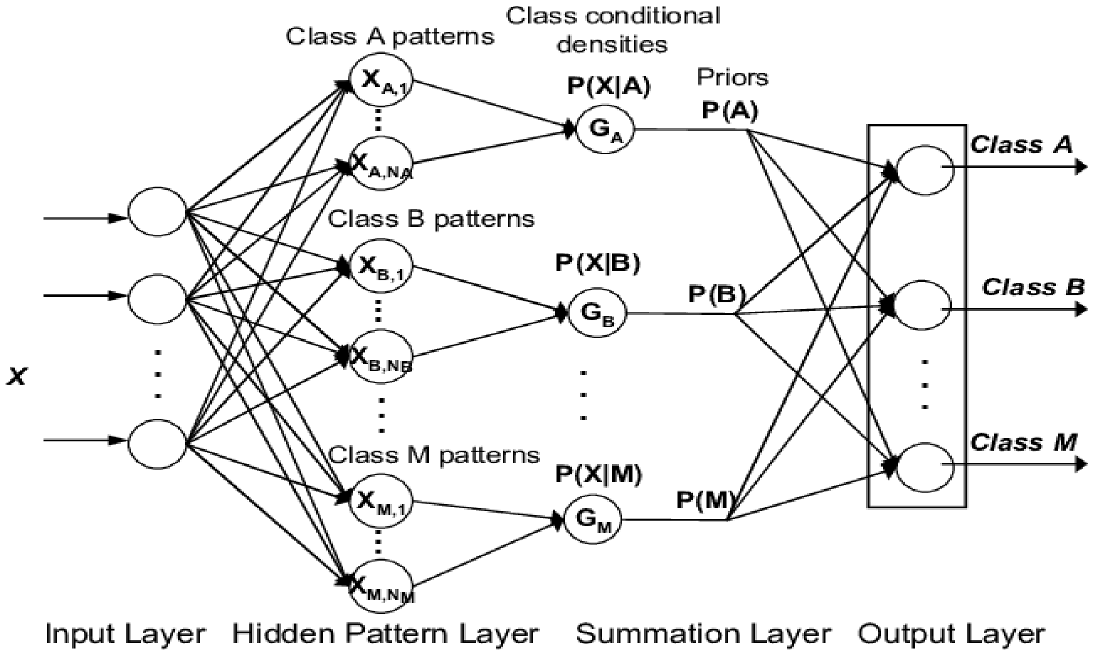

Figure 5: Probabilistic Neural Network

In PNN weights are not "trained" but assigned. Existing weights will never be alternated but only new vectors are inserted into weight matrices when training. So it can be used in real-time. Since the training and running procedure can be implemented by matrix manipulation, the speed of PNN is very fast. Training samples can be added or removed without extensive retraining.

Average of the pdf's for the "n" samples in the population is:

$1/ n\sigma \sum W(x- x_k /\sigma)$---(10)

The flowchart is as follows:

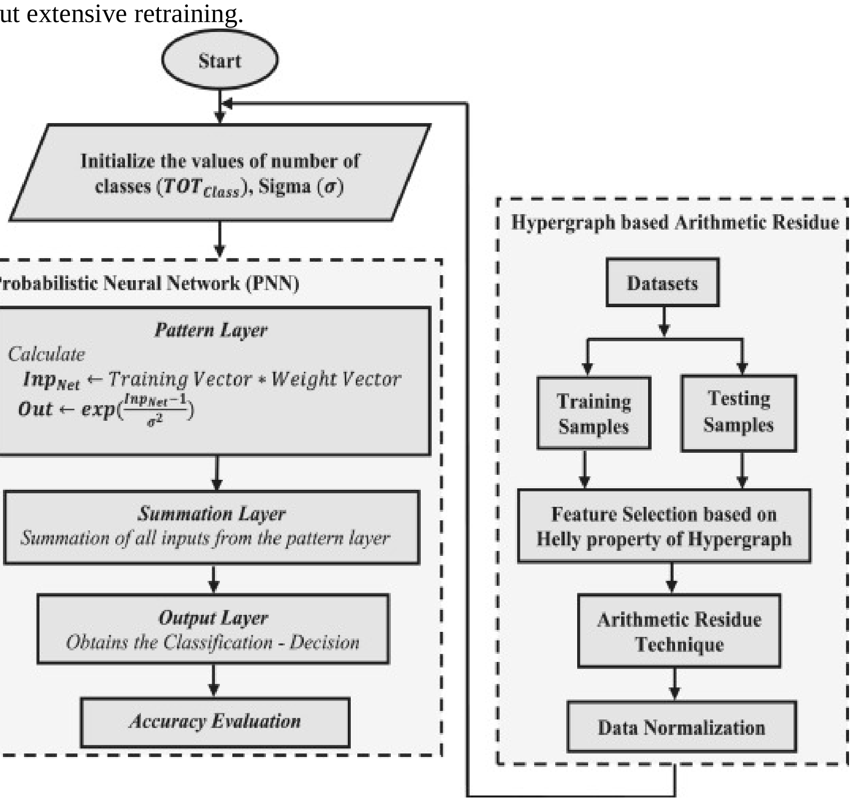

Figure 6: Flowchart of PNN

III. RESULTS

In the following table the values of correct classification function obtained by supplying different sets of input data into the chosen models of neural networks and processing their outputs are presented. The datasets present in UCI are classified. From these datasets 70 % of instances are used for training and 30 % for testing are performed. Learning iterations is set to 1000.

Table1: Classification result of the methods based on artificial neural network

Sl no	Data sets	Function	Classification Accuracy comparison on test data
1	Breast Cancer data 699 samples 2 classes 10 attributes	1.MLP 2.RBF 3.PNN	94 % 96% 95%
2	Iris Data Sets 150 samples 3 classes 4 attributes	1.MLP 2.RBF 3.PNN	96% 99% 95%
3	Wine Data set 178 samples 3 classes 13 attributes	1.MLP 2.RBF 3.PNN	95 % 99% 97%

Figure 7: Comparison with overall Accuracy

IV. CONCLUSIONS

This paper introduces a neural network approach for Classification performance of all three investigated types of neural networks is acceptable. Radial basis function network exhibits better generalization performance then multilayer perceptron and probabilistic neural network. Small number of inputs effect crucially on the generalization performance of neural network classifier. The obtained results confirm the superiority of the RBF technique over the MLP technique in most of the cases with deterministic coefficients near 98 %.

REFERENCES

[1]. Guoqiang Peter Zhang , "Neural Networks for Classification: A Survey ",IEEE transactions on systems, man, and cybernetics , Vol.30, Issue.4, 2000

[2]. [Zak98] Anthony Zaknich, Artificial Neural Networks: An Introductory Course. [Online]. http://www.maths.uwa.edu.au/~rkealley/ann_all/ann_all.html (as of June 6, 2002).

[3]. J. Du, D. Huang, X. Wang, and X. Gu, "Shape recognition based

[4]. on radial basis probabilistic neural network and application to plant species identification," in Proceedings of2005 International Symposium of Neural Networks, ser. LNCS 3497. Springer, 2005.

[5]. D. F. Specht, "Probabilistic neural networks," Neural Networks, vol. 3,1990

[6]. Matlab neural network toolbox documentation. MathWorks. Inc. [Online]. available: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/radial10.html#8378

[7]. Cowell, R., Lauritzen, S., Spiegelhater, D., & David, P. (2003). Probabilistic networks and expert systems. New York, NY: Springer.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

RashmiAmardeep "Neural Network for Pattern Classification." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 12, 2017, pp. 40-44.