



International Journal of Research Publications

Classification Analysis using Radial Basis Function Neural Network and Back Propagation Neural Network

"Sabai Phyu" *

"University of Computer Studies, Yangon, Myanmar"

Abstract

In the age of digital era, machine learning is much more influence and popular rather than case based methods. Among the machine learning research area neural network is very popular for its classification accuracy and learning rate on the complex environment. The important remark on the neural network classification is to use the big volume of data size for training and its data types. The fundamental working concept of neural networks is learning and training of the data computing system. Neural Network is composed of a bulk number of interconnected computing elements. This paper mentions that the properties of the two type of neural networks: Radial Basis Function (RBF) and Back Propagation (BP) neural networks are analyzed and compared based on mean square error, accuracy and nature of datasets concerning with attribute types. There are 15 datasets in this paper are used from UCI machine learning repository.

© 2018 Published by IJRP.ORG. Selection and/or peer-review under responsibility of International Journal of Research Publications (IJRP.ORG)

Radial Basis Function Neural Network, Feed-forward neural network; Back Propagation Neural Network; machine learning.

* Corresponding author. Tel.: +95 9 5351617.
 E-mail address: sabaiphyu@ucsy.edu.mm



International Journal of Research Publications

1. Introduction

There are many available technologies for data mining area, including Artificial Neural Networks, Regression, and Decision Trees, etc. Neural networks are one of the most popular machines learning method in data mining research area due to their black box nature. An Artificial Neural Network (ANN) is a kind of mathematical or computing model based on an emulation of biological neural system. That is composed of a number of connected elements (neurons). The neuron is a basic processing unit that receives two forms of input one is inside of the network and the rest one is from outside of the network applies a local transformation to that input, and produces a single output signal which passes onto other neuron and/or outside the network. The input value is modified by associated value with the connection strength, or weight. The local transformation within the network is referred to as the activation function usually called sigmoid function.

NN advantages are that they can adapt to new scenarios, they are fault tolerant and can deal with noisy data. Time to train NN is probably identified as biggest disadvantage. They also require very large sample sets to train model efficiently. The system is intended to compare and analyse the radial basis and back propagation neural networks according to the nature of datasets (types of attribute values), accuracy measure with mean square error.

The paper organization is from section 1 to section 5. Section 2 is the system related works. In section 3, it introduces the background theory of the

system. Next part is Section 4 and it shows the system implementation and experimental results. The last is Section 5. It includes conclusion part.

2. Related Works

In this section, the work in the literature related to this system is described.

Tuba K [10] showed that in the literature, various algorithms are proposed for training RBF networks, such as the gradient descent (GD) algorithm [5] and Kalman Filtering (KF) [9]. These two algorithms are derivative based and have some weaknesses such as converging to a local minima and time-consuming process of finding the optimal gradient. Because of these limitations, several global optimization methods have been used for training RBF networks for different science and engineering problems such as genetic algorithms (GA) [1], the particle swarm optimization (PSO) algorithm, the artificial immune system (AIS) algorithm [2] and the differential evolution (DE) algorithm [14].

Another method for training multi-layered Feed-forward ANNs is Weight-elimination algorithm that automatically derives the appropriate topology and therefore avoids also the problems with overfitting [12]. Genetic algorithms have been used to train the weights of neural networks [8] and to find the architecture of neural networks [13]. There are also Bayesian methods in existence which attempt to train neural networks [11].

3. Background Theory

3.1. Artificial Neural Network

A neural network sometimes called

Artificial Neural Network. It has a set of connected input/output units. Each connection has an associated weight. The weight adjusting is made during the learning phase, by using and adjusting the weight it can be able to predict or classify the correct class type or cluster type of the input samples. [6]

3.2. Multilayer Feed-forward Networks

Normally neural network is composed of the layers such as input, output and hidden. A multilayer feed-forward neural network has perceptrons interconnection. In the interconnection the calculations flow and data in a single direction, from the input layer to the outputs. The source computation nodes of the input layer supply respective elements of the activation pattern (input vector), which originate the input signals applied to the neurons (computation nodes) in the second layer. The output signals of the second layer are used as inputs to the third layer and so on for the rest of the network. The set of the output signals of the neurons in the output (final) layer of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input layer. The neural network in is said to be fully connected that every node in each layer is connected to all of other nodes in adjacent forward layer.[3]

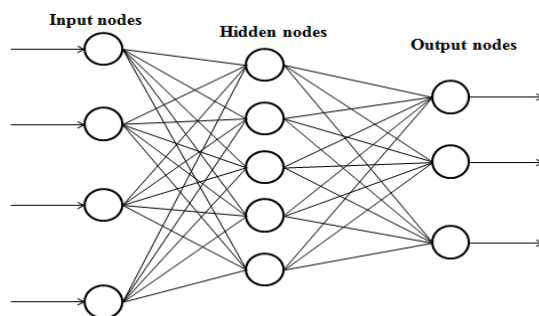


Fig 1. feed-forward neural network architecture

3.3. Back-propagation Neural Network

Back-propagation Neural Network is a form of supervised learning type. This algorithm consists of two steps. First, a training input pattern is presented

to the network input layer. The network propagates the input pattern of one layer to next layer. It will be done repeatedly until the output is generated. If there is different between the actual output and the desired output, it is called error. An error is calculated and then propagated backward through the network from the output layer to the input layer. The weights are modified while the error is propagated. [3]

3.4. Back Propagation Algorithm

Input: training sample, samples; learning rate, l;

Output: A neural network trained (to classify the samples).

Method:

1. initialize all weights and biases in network;
2. while terminating condition is not satisfied {
3. for each training sample x in samples {
4. for each hidden or output layer unit j {
5. $I_j = \sum_i W_{ij} O_i + \theta_j$; // compute the net input of unit j with respect to the layer, i
6. $O_j = 1 / (1 + e^{-I_j})$;
7. for each unit j in the output layer
8. $Err_j = O_j(1 - O_j)(T_j - O_j)$;
9. for each unit j in the hidden layers, from the last to the hidden layer
10. $Err_j = O_j(1 - O_j) \sum_k Err_k W_{jk}$; //compute the error with respect to the next higher layer, k
11. for each weight W_{ij} in network {
12. $W_{ij} = W_{ij} + (l) Err_j O_i$;
13. $W_{ij} = W_{ij} + \Delta W_{ij}$;
14. for each bias θ_j in network
15. $\Delta \theta_j = (l) Err_j$;
16. $\theta_j = \theta_j + \Delta \theta_j$;
17. }

3.4. Radial Basis Function Neural Network

The Radial Basis Function Neural Network (RBFNN) is viewed as a “3-layer network” where the input vector is the first layer, the second “hidden” layer is the RBF neurons, and the third layer is the output layer. The entire input vector is shown to each of the RBF neurons. Each RBF neuron stores a “prototype” vector. Each RBF neuron compares the input vector to its prototype, and outputs a value

between 0 and 1 which is a measure of similarity. The neuron's response value is also called its "activation" value. The output of the network consists of a set of nodes, one per category. Each output node computes an output by multiply the neuron's activation with weight values.[7]

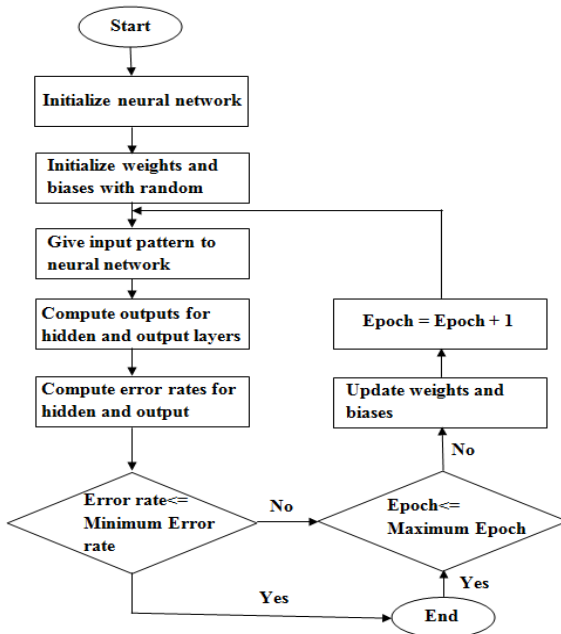


Fig 2. Training Neural Network with Back Propagation

3.5. Radial Basis Function Algorithm

Input: The training samples, samples; the learning rate, l ;

Output: A neural network trained to classify the samples.

Method:

1. Initialize all weights and biases in network;
2. While terminating condition is not satisfied {
3. for each training sample x in samples {
4. for hidden unit j {
5. $\Theta_j(x) = \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right)$ //compute the gaussian function of unit j using the mean, standard deviation and centroids.
6. For each output k {
7. $I_k = \sum_j W_{kj} \Theta_j + \theta_k$; // compute the net input of unit j with respect to the layer, i
8. $O_k = 1 / (1 + e^{-I_k})$; // compute the output of each unit k
9. for each unit j in the output layer
10. $Err_k = O_k(1 - O_k)(T_k - O_k)$;
11. for each unit j in the hidden layer
12. $Err_j = O_j(1 - O_j) \sum_k Err_k W_{jk}$;
13. for each weight W_{ij} in network {
14. $W_{ij} = W_{ij} + (l) Err_j O_i$;
15. $W_{ij} = W_{ij} + \Delta W_{ij}$;
16. for each bias θ_j in network
17. $\Delta \theta_j = (l) Err_j$;
18. $\theta_j = \theta_j + \Delta \theta_j$;
19. }

4. Network Structure for datasets

1. The number of input attributes of the datasets should be the best for the number of input layer nodes.
2. According to the rule-of-thumb rule, the number of hidden layer nodes is the number of output layer nodes plus 2/3 of the size of input layer nodes.
3. The class labels or outputs of the datasets should be the number of output layer nodes. [2]

The network structures for each dataset are shown in Table 1.

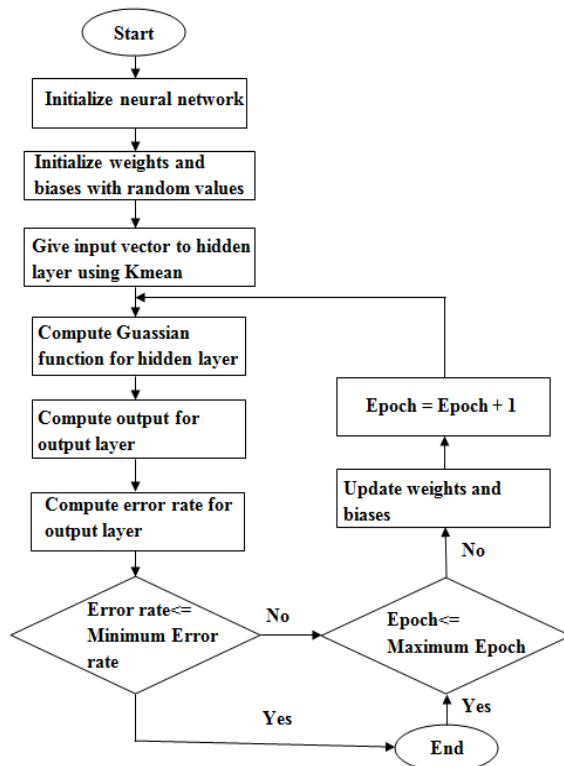


Fig 3. Training Neural Network with Radial Basis Function

TABLE I

INPUT, HIDDEN, AND OUTPUT NODES NEEDED TO BUILD NETWORK ARCHITECTURES FOR EACH DATASETS

Datasets	Input nodes	Hidden nodes	Output nodes
Iris Plant	4	5	3
Breast Cancer	9	8	2
Car Evaluation	6	8	4
Wine	13	12	3
Ecoli	8	12	8
ISOLET	617	482	26
Molecular Biology	61	48	3
Mushroom	22	18	2
Letter Recognition	16	31	26
Nursery	8	10	5
Statlog (Image	19	20	7

Segmentation)			
Statlog (Shuttle)	9	12	7
Poker Hand	10	15	10
Pen-Based Recognition of Handwritten Digits	16	19	10
Internet Advertisements	1558	1170	2

TABLE II

INPUT, HIDDEN, AND OUTPUT NODES NEEDED TO BUILD NETWORK ARCHITECTURES FOR EACH DATASETS

Datasets	#of Attribute	#of Instances	Attribute Type	# of Class Label
Iris Plant	4	150	Real	3
Breast Cancer	9	286	Categorical	2
Car Evaluation	6	1728	Categorical	4
Wine	13	178	Integer, Real	3
Ecoli	8	336	Real	8
ISOLET	617	7797	Real	26
Molecular Biology	61	3190	Categorical	3
Mushroom	22	8124	Categorical	2
Letter Recognition	16	20000	Integer	26
Nursery	8	12960	Categorical	5
Statlog (Image Segmentation)	19	2310	Real	7
Statlog (Shuttle)	9	58000	Integer	7
Poker Hand	10	1025010	Categorical, Integer	10
Pen-Based Recognition of Handwritten Digits	16	10992	Integer	10
Internet Advertisements	1558	3279	Categorical, Integer, Real	2

5. Advantages of Neural Network

1. NN provides high tolerance of noisy data (ability to classify patterns on which they have not been trained).
2. NN can be used when you may have little knowledge of the relationships between attributes and classes. They are well -suited for continuous-valued inputs and outputs.
3. NN can work successfully on real world data including handwritten character recognition, pathology, and laboratory medicine and training a computer to pronounce English text.
4. NN technique can be used to speed up the computation process.
5. Several techniques have been developed for extraction of rules from trained neural networks.
6. NN is very useful for classification and prediction.

6. System Implementation

6.1. Overview of the System

In this system, the two methods of neural networks, back-propagation neural network and radial basis function neural network are compared and analysed. The 15 datasets are used. Which are received from repository of UCI machine learning. The system allows the data to train the neural network with back propagation and radial basis function algorithm. Neural networks trained with both algorithms and store the weight values respectively. And then used the test dataset for the testing phase.

After getting the result from the test datasets with both algorithms, the radial basis function has more advantages. The radial basis function test result has proven to be more accurate in evaluation datasets which have only discrete attributes, than back propagation. And then convergence time is faster in all 15 datasets.

6.2. System Architecture and Design

The architecture and design of the system is shown in Fig 4. There are two parts in the system. They are train and test. Each dataset splits into training and testing sets using hold out method,

usually one third for testing and the rest for training. In training, the user trains 2/3 of the datasets with back propagation and radial basis function networks and save the networks. In testing, the user tests the remaining 1/3 of datasets using saved networks.

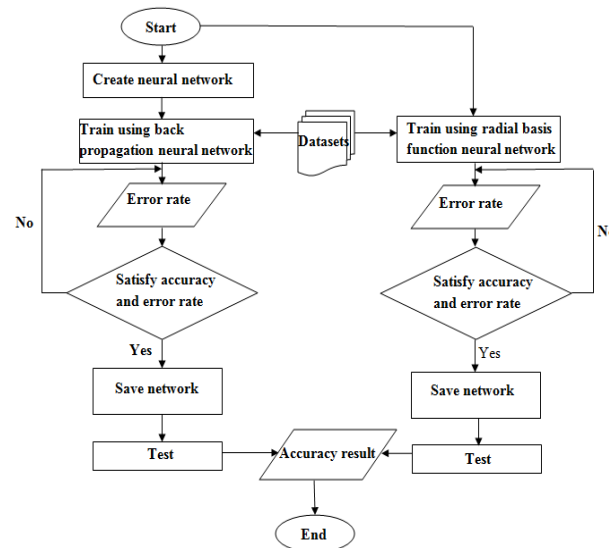


Fig 4. System Flow Diagram

6.3 Experimental Results

The system compared about back propagation and radial basis function neural networks by using different learning rates based on the 15 datasets. The highest accuracy results for back propagation and radial basis function neural networks for each dataset are shown in Table 2.

TABLE II

THE HIGHEST ACCURACY (%) BETWEEN BACK PROPAGATION AND RADIAL BASIS FUNCTION NEURAL NETWORKS BASED ON THESE 15 DATASETS

Datasets	Accuracy(%) with BPNN	Accuracy(%) with RBFNN
Iris Plant	96.00	100.00
Breast Cancer	66.10	96.61
Car Evaluation	62.84	87.32
Wine	81.52	89.13
Ecoli	88.43	91.87
ISOLET	83.40	89.56
Molecular Biology	71.22	94.55

Mushroom	73.41	95.67
Letter Recognition	79.33	89.45
Nursery	74.56	97.89
Statlog (Image Segmentation)	76.26	88.47
Statlog (Shuttle)	75.34	83.67
Poker Hand	73.60	85.34
Pen-Based Recognition of Handwritten Digits	78.88	89.28
Internet Advertisements	75.36	89.78

7. Conclusion

In this paper, the back propagation and radial basis function neural network are compared based on the 15 datasets. In these datasets, breast cancer, car evaluation, Molecular Biology, Mushroom and Nursery datasets have categorical (nominal) attributes only. According to the experimental results above, radial basis function neural network also high accuracy with datasets consist of discrete attributes (e.g. nominal, ordinal) than back propagation neural network. And also radial basis function neural network is faster training time because it has only one layer for calculating error (output layer to hidden layer). Radial basis function is more reliable on the nature of datasets, high accuracy and faster training time than back propagation neural networks according to the above reasons.

References

- [1]. Barreto, A.M.S., Barbosa H.J.C., Ebecken N.F.F. "Growing Compact RBF Networks Using a Genetic Algorithm". Proceedings of the 7th Brazilian Symposium on Neural Networks; Recife, Brazil. 2002; pp. 61–66.
- [2]. De Castro L.N., von Zuben F.J. "An Immunological Approach to Initialize Centers of Radial Basis Function Neural Networks". Proceedings of Brazilian Conference on Neural Networks; Rio de Janeiro, Brazil. 2001; pp. 79–84.
- [3]. Hashim, H., M.A. Haron; "Artificial Neural Networks based Algorithm for Identifying Engine Oil Parameters", 2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks.
- [4]. Heaton J, "Introduction to Neural Networks for Java, Second Edition".
- [5]. Karayiannis, N.B. "Reformulated radial basis neural networks trained by gradient descent". IEEE Trans. Neural Network. 1999; 3:2230–2235. [PubMed]
- [6]. Paras, Sanjay Mathur, Avinash Kumar, and Mahesh Chandra, "A Feature Based Neural Network Model".[NNN]
- [7]. Ramya, S., Dr. Radha N, Research Scholar, Department of Computer Science, "Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms".
- [8]. Siddique, M. N. H. and Tokhi, M. O. (2001), "Training Neural Networks: Backpropagation vs. Genetic Algorithms", IEEE International Joint Conference on Neural Networks, Vol. 4, pp. 2673-2678.
- [9]. Simon, H, "A Comprehensive Foundation of Neural Network", Second Edition, Prentice Hall, Inc, 1999.
- [10]. Tuba, K. and Erkan, B., "A Comparison of RBF Network Training Algorithms for Inertial Sensor based Terrain Classification", Sensors (Basel). 2009; 9 (8): 6312-6329
- [11]. Vivarelli, F. & Williams, C. (2001). "Comparing Bayesian neural network algorithms for classifying segmented outdoor images". Neural Networks 14: 427-437.
- [12]. Weigend, A. S., Rumelhart, D. E., & Huberman, B. A. (1991). "Generalization by weight-elimination with application to forecasting". In: R. P. Lippmann, J. Moody, & D. S. Touretzky (eds.), Advances in Neural Information Processing Systems 3, San Mateo, CA: Morgan Kaufmann.
- [13]. Yen, G. G. and Lu, H. (2000), "Hierarchical genetic algorithm based neural network design, In: IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks", pp. 168-175.
- [14]. Yu, B., He, X. "Training Radial Basis Function Networks with Differential Evolution." Proceedings of IEEE International Conference on Granular Computing; Atlanta, GA, USA. 2006; pp. 369–372.