Volume 5, No. 8, Nov-Dec 2014

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Validation of a Neural Network Based Leaf Classification Algorithm

C. S. Sumathi
Assistant Professor,
School of Post Graduate Studies,
Tamil Nadu Agricultural University,
Coimbatore- 641 003, Tamil Nadu, India

A.V. Senthil Kumar
Director,
Dept. of PG & Research in Computer Applications,
Hindusthan College of Arts and Science,
Coimbatore- 641 028, Tamil Nadu, India

Abstract: Plants are living organism which belongs to vegetable kingdom that can live both on land and water. More than 300,000 species of plants exists on earth. In order to effectively conserve and save the genetic resources, plants are to be identified and leaf shape plays a significant role in plant classification. In this paper, since identifying the relevant feature is of vital importance the features are extracted using information gain based feature selection method. A feed forward neural networks with different learning methods viz., Levenberg-Marquardt learning, Incremental Backpropagation learning and Batch Back propogation learning automate the leaf recongnition for plant classification. Comparison shows that information gain helps select features that show good improvement on feed forward neural network (normalized cubic spline) with batch back propogation classifier algorithm out performs with an accuracy of 95.56%.

***Key words*:** Information gain, Levenberg-Marquardt, Incremental Backpropagation, Batch Back propogation, Feed Forward Neural Network

I. INTRODUCTION

Feed forward neural network (NCS) [1] automates the leaf recognition for plant classification. The classification accuracy of the Feed forward neural network (NCS) is compared with Radial Basis Function (RBF), Classification and Regression Tree (CART) and Multilayer Layer Perceptron (MLP). Correlation base feature selection is used for selecting features. The features extracted are trained using 10 fold cross validation and tested with CART, RBF, MLP classifiers and Feed forward neural network (NCS). The output from Feed forward neural network (NCS) for a nine species problem is found satisfactory with better accuracy and recall.

In 1994 an algorithm that adds and deletes units/layers was proposed [11]. The algorithm generates an intelligent and test procedure, explores various alternatives and selects the one promising feature. It is a relatively new constructive for Real Valued Examples (CARVE), proposed in 1998. It uses convex hull methods for the determination of network weights. The algorithm proceeds with an empty hidden layer into which thresholds units are added one at a time until the layer is complete. Cross Validation for Feed forward Neural Network is used in Construction for adding units to a single hidden layered network. The network with maximum hidden units is accepted if the total accuracy on training and cross validation samples is more than that of the previous network.

A. Feature Selection:

Information gain is a well known feature selection method. It is a reasonable objective to use for selecting feature. Using information gain will help to reduce the noise which is due to irrelevant features for influencing classifier. Information gain (IG) measures amount of information in bits about class prediction, when the only information available is presence of a feature and corresponding class distribution [9]. Information gain measure selects test attribute at every node in the tree [8]. The attribute with highest information gain (greatest entropy reduction) is selected as test attribute for current node. This attribute minimizes information needed to classify samples in resulting partitions. Entropy, measures amount of disorder or uncertainty in systems. In classification setting, higher entropy (more disorder) corresponds to sample of mixed label collection. Lower entropy corresponds to a case where there are pure partitions. In information theory[2], sample D's entropy is defined as follows:

$$H(D) = -\sum_{i=1}^{k} P(c_i \mid D)\log_2 P(c_i \mid D)$$

Where $P(c_i \mid D)$ is probability of a data point in D being labeled with class c_i, and k is number of classes. $P(c_i \mid D)$ is estimated directly from the data as follows:

$$P(c_i \mid D) = \frac{|\{x_j \in D \mid x_j \text{ has label } y_j = c_i\}|}{|D|}$$

Also weighted entropy of a decision/split are defined as follows:

$$H(D_L, D_R) = \frac{|D_L|}{|D|}H(D_L) + \frac{|D_R|}{|D|}H(D_R)$$

Where D was partitioned into D_L and D_R due to split decision. Finally, , information gain for a given split is defined as::

$$Gain(D, D_L, D_R) = H(D) - H(D_L, D_R)$$

In other words, Gain is anticipated entropy reduction caused by knowing an attribute's value.

II. MATERIAL AND METHODS

A. Feed Forward Neural Network (NCS):

Neural Network is information processing paradigm inspired by the way biological nervous systems process information. The novel construction of the information processing system is the input element of this concept[3]. It is composed of a large number of highly interconnected processing elements called neurones working in accord to solve specific problems. They learn by example and build for a specific application, like pattern recognition or data classification, through a learning process. In biological systems learning involves adjustments to the synaptic connections that exist between the neurones.

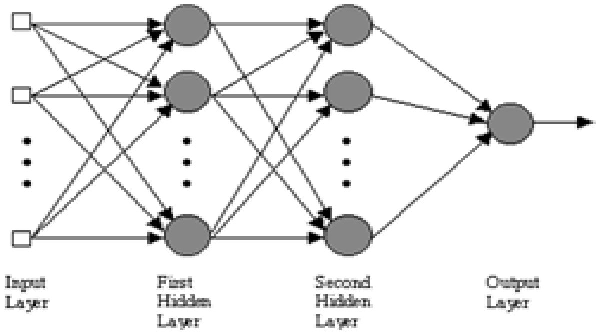

Figure 1 Multilayer perceptron

The signals from input to output travel one way only in Feed Forward NNs. Hence, they tend to be straight forward networks that associate inputs with outputs and broadly used in pattern recognition. The inputs fed on the input layer, propagates through the network in forward direction through the hidden layers to give an output. Output signal is calculated using weights, bias and activation function. The neural network is trained using backpropagation rule by backpropagating the errors and changing weights of nodes. The error is the difference between the outputs obtained and desired output. The following are the algorithms used for calculating various parameters involved in training a neural network [1].

The sum input for a given neuron is given by:

$$s_k = \sum_j w_{jk} y_j + \theta_k$$

Where s_k is the total or effective input for unit k, w_{jk} the weight of the connection, y_k is current activation and θ_k is the bias.

Activation function A_f takes the input and current activation and by learning gets the new activation value:

$$y_k \ \ t \ = A_f \ \ y_k \ \ t-1 \ .s_k \ \ t-1$$

The value of activation functions are generally limited to 0, 1 using a threshold function. The sigmoid is smooth, differentiable, nonlinear, and saturating, thus, is the most widely used activation function[5]. Most commonly used is sigmoid function.

$$y_k = A \ \ s_k \ = \frac{1}{1+e^{-s_k}}$$

The spline-based NN is built using generalized sigmoidal (GS) neuron, which contains adaptive parametric spline activation function. The spline activation is easy to adapt and implement. It also retains squashing property of the sigmoid and smoothing characteristics. MLP built using spline activation function are universal approximators and have smaller structural complexity.

The spline activation function reproduces the shape of whole cubic spline along the directions specified by wj, j=1,..,n [7].

$$\varphi \ \ w_j x \ \ \sum_{i=1}^{N} c_i \left| w_j x - \alpha_{ij} \right|^3$$

$f \ \ x$ can be written as:

$$f \ \ x \ = \sum_{j=1}^{n} \mu_j \varphi_j \ \ w_j x$$

μ_j and w_j are found using backpropagation, thus optimal set of parameters and coordinates are found. The tracts in the spline are described by combination of coefficients. Local spline basis functions controlled by only 4 coefficients are used to represent the activation function. Catmull-Rom cubic spline [6] is used and its ith tract is expressed as:

$$F_i \ \ u \ = \begin{bmatrix} F_{x,i} \ \ u \\ F_{y,i} \ \ u \end{bmatrix} = \frac{1}{2} \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$$

Feed Forward Neural Network (NCS) with various learning algorithms

a. Feed Forward Neural Network (NCS) with Levenberg–Marquardt based learning:

The Levenberg-Marquardt (LM) algorithm is the most widely used optimization algorithm. It provides a numerical solution to the problem of minimizing a function. The solving the equation is given by:

(JtJ + λI)δ = JtE (1)

Where J is the Jacobian matrix for the system, λ is the Levenberg's damping factor, δ is the weight update vector that we want to find and E is the error vector containing the output errors for each input vector used on training the network. The δ tell us by how much we should change our network weights to achieve a (possibly) better solution.

The JtJ matrix can also be known as the approximated Hessian. The λ damping factor is adjusted at each iteration, and guides the optimization process. If reduction of E is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss–Newton algorithm, whereas if iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient descent direction. Variations of the algorithm may include different values for v, one for decreasing λ and other for increasing it [4]. Advantages of LMA includes, the learning capability of the LMA is learned to be superior, it has rapid convergence and fastest among the other training algorithms and suitable for medium size datasets

b. Feed Forward Neural Network (NCS) with Incremental Backpropagation algorithm based learning:

The incremental back propagation updates the weights iteratively after each process. After presentation of a training vector and computing the weight changes, they are introduced immediately to adapt the weights, without accumulating all changes until the end of the epoch. This sometimes referred as stochastic training. This has the ability to escape from entrapment at poor local minima on the error surface. This can happen as the shape of the error landscape changes with each process. Because of this, the instantaneous error may become of such magnitude that it allows the algorithm to jump out of the current landscape basin. Theoretically however, it is impossible to precisely investigate this behavior. Hence, the performance of the incremental backpropagation is laborious to analyze.

c. Feed Forward Neural Network (NCS) with Batch Back propagation algorithm based learning:

In batch training, before learning takes place all patterns are presented to the network. Several passes are made through the training data in every case. Here, all the training patterns are presented and their corresponding weight updates are added and then are the actual weights in the network updated. This method is iterated until criteria are met [10]. The weights are updated only after all patterns have been presented and hence randomly selecting the patterns are avoided. Subsequently each weight change made during continuous training will reduce the error for that particular instance, but can decrease or increase the error on the training set. Hence, batch training produces better accuracy.

Batch learning proceeds as follows:
begin initialize nH, w, criterion q, h, r ¬ 0
do r ¬ r + 1 (increment epoch)
m ¬ 0 ; Dwji ¬ 0; Dwkj ¬ 0
do m ¬ m + 1
xm¬select pattern
Dwji¬ Dwji + $\eta x_i \delta_j$; Dwkj¬ Dwkj + $\eta y_j \delta_k$
until m=n
wji¬ wji +Dwji; wkj¬ wkj +Dwkj
until $\|\nabla J(\mathbf{w})\| < \theta$
return w end

III. RESULTS AND DISCUSSION

Plant leaves from nine species were selected with 15 samples of each species. Image of the plant leaves used is shown in Fig. 2. The information gain based features were extracted using MatLab and classified using Feed Forward Neural Network (NCS) with various learning method *viz.*, Levenberg–Marquardt, Incremental Back propagation, Batch Back propagation.

Figure 2 Sample image of plant leaves

The classification accuracy obtained is tabulated is given in table 1 and in table 2 tabulates the precision, recall and fMeasure for various algorithms.

Table 1. Classification Accuracy

Technique Used	Classification accuracy
Feed Forward Neural Network (NCS) with Levenberg–Marquardt learning	92.60%
Feed Forward Neural Network (NCS) with Incremental Backpropagation learning	94.07%
Feed Forward Neural Network (NCS) with Batch Backpropagation learning	95.56%

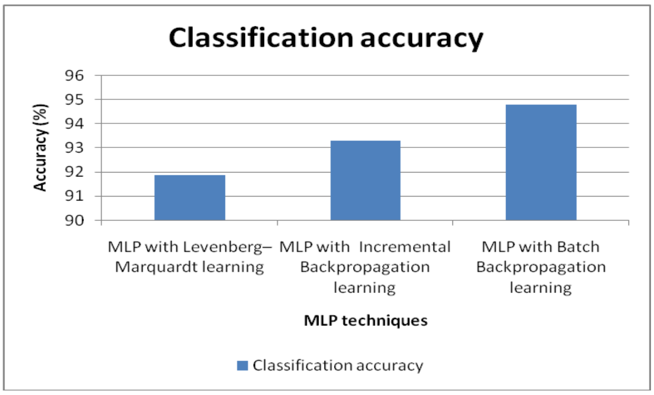

Figure 3 Classification Accuracy

From table 1 and Fig. 3 it is observed that the classification accuracy is achieved for different techniques. Feed Forward Neural Network (NCS) with Batch Back Propagation learning achieves better accuracy as 2.96% than Feed Forward Neural Network (NCS) with Levenberg-Marquardt learning and as 1.49% than Feed Forward Neural Network (NCS) with Incremental Back Propagation Learning.

Table 2 Precision, recall and f Measure

Technique Used	Precision	Recall	f Measure
Feed Forward Neural Network (NCS)with Levenberg–Marquardt learning	0.93	0.925	0.924
Feed Forward Neural Network (NCS) with Incremental Back propagation learning	0.946	0.94	0.94
Feed Forward Neural Network (NCS) with Batch Back propagation learning	0.959	0.955	0.955

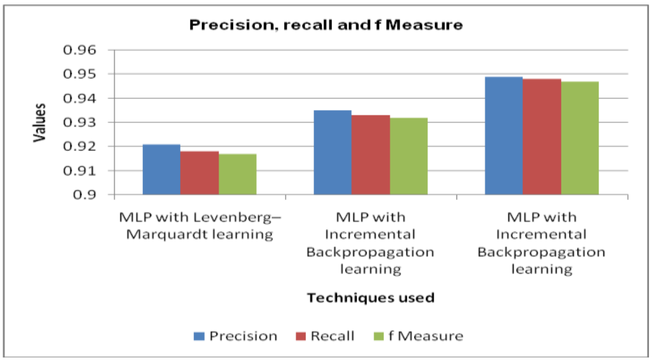

Figure 4 Precision, recall and f Measure

From table 2 and Fig. 4 it is observed that the Precision Recall and fmeasures are achieved for different techniques. Feed Forward Neural Network (NCS) with Batch Back Propagation learning achieves better precision of 0.959, better Recall of 0.955 and good fmeasures of 0.955.

IV. CONCLUSION

In this study, a feed forward neural network is used to automate the leaf recognition for plant classification. The information gain based features were extracted and classified using Feed Forward Neural Network (NCS) with various learning method such as Levenberg–Marquardt, Batch Back propagation, Incremental Back propagation. Plant leaves from nine species were selected with 15 samples of each species. Experimental results show that the proposed Feed Forward Neural Network (NCS) with Batch Back Propagation learning method achieved a better performance of classification accuracy, precision, recall and fmeasures when compared to other learning methods. The classification accuracy achieved by the proposed method is 95.56% which is the best accuracy when compared to other learning methods.

V. REFERENCES

[1] C. S. Sumathi and A. V. Senthil Kumar, Plant Leaf Classification Using Soft Computing Techniques, IJFCC 2013 Vol.2(3): 196-199 ISSN: 2010-3751

[2] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. Wiley- Inter science, 1991.

[3] http://www.mu-sigma.com/analytics/thought_leadership/cafe-cerebral-neural-network.html

[4] http://crsouza.blogspot.in/2009/11/neural-network-learning-by-levenberg_18.html

[5] http://www.byclb.com/TR/Tutorials/neural_networks/ch10_1.htm

[6] S. Guarnieri, F. Piazza, and A. Uncini, “Multilayer neural networks with adaptive spline-based activation functions,” in Proc. Int. Neural Network Soc. Annu. Meet. WCNN, Washington, DC, 1995, pp. I695– I699.

[7] L. Vecci, F. Piazza, and A. Uncini, “Learning and approximation capabilities of adaptive spline activation neural networks,” Neural Networks, vol. 11, no. 2, pp. 259–270, Mar. 19

[8] Barros, R. C., Basgalupp, M. P., de Carvalho, A. C., &Freitas, A. A. (2011, July). Towards the automatic design of decision tree induction algorithms. In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (pp. 567-574). ACM.

[9] Cover, T. M., & Thomas, J. A. (2012). Elements of information theory. John Wiley & Sons.

[10] Azhagusundari, B., and Thanamani, A. S. Feature selection based on information gain. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN, 2278-3075.

[11] Nabhan, T. M. and A. Y. Zomaya, “Toward generating neural network structures for function approximation”, Neural Networks, Vol. 7, No. 1, pp. 89–99, 1994.