Back Propagation Neural Network for Classification of IRS-1D Satellite Images

E. Hosseini Aria, J. Amini, M.R.Saradjian

Department of geomantics, Faculty of Engineering, Tehran University, Iran - aria@engineer.com , jamini@.ut.ac.ir , mrsaradji@yahoo.com , Jamebozorg G., National Cartographic Center (NCC), Tehran, Iran

Fax: (0098) 21 -8008837

KEY WORDS: Back-propagation, Neural network, Classification, IRS-1D image, Feature, Probability distribution,

ABSTRACT:

The suitability of Back Propagation Neural Network (BPNN) for classification of remote sensing images is explored in this paper. An approach that consists of three steps to classify IRS-1D images is proposed. In the first step, features are extracted from the first-order histogram measures. The next step is feature classification based on BPNN, and in the finally step the results are compared with the maximum likelihood classification (MLC) method. The statistical features in this paper are based on the first-order distribution measure: mean, standard-deviation, skew-ness, kurtosis, energy, and entropy. The network contains 3 layers. The extracted features are fed to input layer that consists of 18 neurons. The back propagation neural network was trained on six classes of the IRS-1D image base on known features and the trained network was used to classify the entire image. The method of this paper is tested for regions of Iran. The IRS-1D 8-bits bands 2, 3 and 4 of LISS-Ш sensor were fused with pan data to construct an image with 5.8 m spatial resolution. Experimental results show that BPNN method is more accurate than MLC and is more sensitive to training sites.

1.INTRODUCTION

Based on biological theory of human brain, artificial neural networks (NN) are models that attempt to parallel and simulate the functionality and decision-making processes of the human brain. In general, a neural network is referred to as mathematical models of theorized mind and brain activity. Neural network features corresponding to the synapses, neuron, and axons of the brain are input weights. Processing Elements (PE) is the analogs to the human brain's biological neuron. A processing element has many input paths, analogous to brain dendrites. The information transferred along these paths is combined by one of a variety of mathematical functions. The result of these combined inputs is some level internal activity (I) for the receiving PE. The combined input contained within the PE is modified by the transfer function (f) before being passed to other connected PEs whose input paths are usually weighted(W) by the perceived synaptic strength of neural connections.

Neural networks have been applied in many applications such as: automotive, aerospace, banking, medical, robotics, electronic, and transportation. An other application of NN is in remote sensing for classification of images. Many methods of classification have been already proposed. Bendiktsson et al.(1990) compared neural network and statistical approaches to classify multi-spectral data. They noted that conventional multivariate classification methods cannot be use in processing multi-source spatial data because of their often different distribution properties and measurement scales. Heermann and Khazenie(1992) compared NN with classical statistical techniques. They concluded that the back propagation network could be easily modified to accommodate more features or to include spatial and temporal information. Bischof et al (1992) included texture information in the NN process and concluded that neural network were able to integrate other sources of knowledge and use then in classification. Hepner et al (1990) compared the use of NN back propagation with maximum likelihood method for classification. The result showed that a single training per-class neural network classification was comparable to a four-training site per-class in conventional classification. Ritter and Hepner (1990) used feed-forward neural network model for classification. The results showed that the neural network had the ability to distinguish small linear pattern, which were apparent on the TM image.

In this paper, artificial neural network for classification of IRS-1D data has been implemented. The back propagation algorithm is applied for classification of the images. A good method for training is an important problem in the classification of IRS data with neural network. TrainLM method has been applied on using back propagation neural networks algorithm on IRS images. Specifications of the training method are discussed in section 2, the study area and experimental results are explored in section 3 and the conclusions are presented in section 4.

2-THE MODEL OF NEURAL NETWORK

First, this section presents the architecture of the back propagation algorithm. Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple layer network and non linear differentiable transfer function. Input vectors and corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined in this study. Networks with biases, a sigmoid layer and a linear output layer are capable of approximating any function with a finite number of discontinuities. The back propagation algorithm consists of two paths; forward path and backward path. Forward path contain creating a feed forward network, initializing weight, simulation and training the network. The network weights and biases are updated in backward path. (Rumelhart,1986) A single layer network with 4 inputs is shown in figure 1.

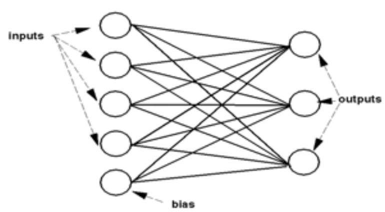
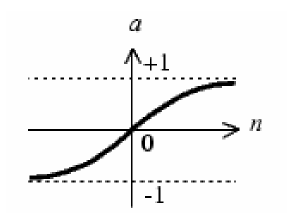


Figure 1. Single layer network

Feed forward networks often have one or more hidden layers of sigmoid neurons followed by output layer of linear neurons. Multiple layers of neurons with non linear transfer functions allow the network to learn non linear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1 (Figure 2).



$$a = sig(n)$$

Figure 2. Sigmoid transfer function

On the other hand, if we want to constrain the outputs of network between 0 and 1, then the output layer should use a log-sigmoid transfer function.

Before training a feed forward network, the weight and biases must be initialized. Once the network weights and biases have been initialized, the network is ready for training. We used random numbers around zero to initialize weights and biases in the network. The training process requires a set of proper inputs and targets as outputs. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function for feed forward networks is mean square errors, the average squared errors between the network outputs and the target output.

2.1.Model of training

There are several training algorithms for feed forward networks. All these algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called back propagation, which involves performing computational backwards through the network.

The simplest implementation of back propagation learning updates the network weights and biases in the direction in which the performance function decreases more rapidly. An iteration of this algorithm can be written

$$X_{k+1} = X_k - \alpha_k g_k \qquad \textbf{(1)}$$

where X_k = a vector of current weights and biases

g_k = the current gradient

α_k = learning rate

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all inputs are applied to the network before the weights are updated.

In this paper, we have used Levinberg-Marquardt training (trainLM) (Hagan, Menhaj, 1994). It's faster and more accurate than the standard back propagation algorithm for training. It can converge from ten to one hundred times faster than the standard algorithm using delta rules. This algorithm operates in the batch mode and is invoked using train.

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed forward networks), then the Hessian matrix can be approximated as

$$H = J^T J \qquad \textbf{(2)}$$

and the gradient can be computed as

$$g = J^T e \qquad \textbf{(3)}$$

where J = the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases

e = a vector of network errors

The Jacobian matrix can be computed through a standard back propagation technique that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - \left[J^T J + \mu I\right]^{-1} J^T e \qquad \textbf{(4)}$$

When the scalar μ is zero, it is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

3. STUDY AREAS AND EXPRIMENTAL RESULTS

The flow diagram of this paper is shown in figure 3. This diagram consists of 3 steps: feature extraction, network design, training and classification.

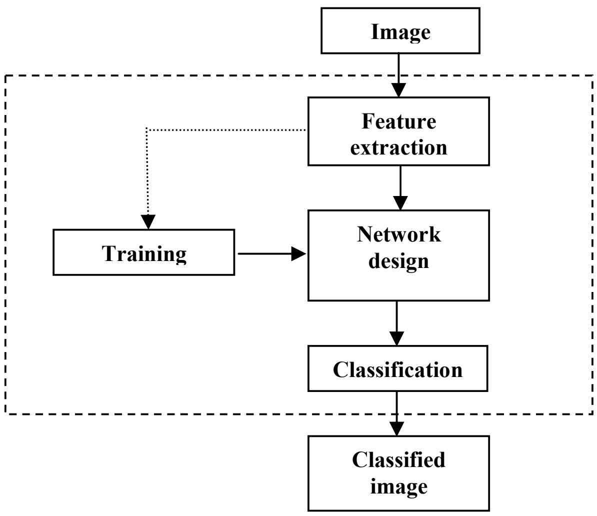


Figure 3: Flowchart for land cover classification using ANN

3.1. Feature extraction

Classification of multispectral remote sensing data may be considered as mapping, F, from a multidimensional gray value space into a discrete vector space of feature classes given by

$$F:[a,b,c,d,...]^{M*N} \rightarrow \{A,B,C,...\}^{M*N} \qquad \textbf{(5)}$$

where: a, b, c, d,…= gray value of the pixel in different spectral bands

A, B, C, … = the feature classes

M*N = is total number of pixels in the image, in any of the spectral bands.

The most basics of all image features are some measures of image amplitude in term of luminance, spectral value, etc. One of the simplest ways to extract texture features in an image is use the first-order probability distribution of the amplitude of the quantized image. They are generally easy to compute and largely heuristic. The first order histogram estimate of $p(b)$ is simply

$$p(b) = \frac{N(b)}{M} \qquad \textbf{(6)}$$

where b = a gray level in an image

M= represent the total number of pixels in a neighborhood window centered about an expected pixel

N(b) = the number of pixels of gray value b in the same window that $0 \le b \le L-1$

Then the following measures have been extracted by using first order probability distribution.

Mean:

$$S_M = \bar{b} = \sum_{b=0}^{L-1} bP(b)$$

Standard deviation:

$$S_D = \sigma_b = \left[\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^2 P(b)\right]^{1/2}$$

Skew-ness:

$$S_S = \frac{1}{\sigma_b^3}\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^3 P(b)$$

Kurtosis:

$$S_K = \frac{1}{\sigma_b^4}\sum_{b=0}^{L-1}\left(b-\bar{b}\right)^4 P(b) - 3$$

Energy:

$$S_N = \sum_{b=0}^{L-1}\left[P(b)\right]^2$$

Entropy:

$$S_E = -\sum_{b=0}^{L-1} P(b)\log_2\{P(b)\}$$

The first two features are the mean and standard deviation of pixel intensities within the image window.

Next, to get information on the shape of the distribution of intensity values within window, the skewness and kurtosis are determined. The skewness, characterizes the degree of asymmetry of the intensity distribution around the mean intensity. If skewness is negative, the data spread out more to the left of the mean than to the right. If skewness is positive, the data are spread out more to the right.

The kurtosis, measure the relative peakness or faltness of the intensity distribution relative to the normal distribution. The kurtosis of the normal distribution is 3. Distribution that are more outlier-prone than the normal distribution have kurtosis greater than 3; Distribution that are less outlier-prone have kurtosis less than 3.

Finally, the energy and entropy is determined. The energy, is useful to examine the power content (repeated transitions) in a certain frequency band. Entropy is a common concept in many fields, mainly in signal processing (Coifman1992).

3.2 Network design

In this paper, a three – layer network is developed. An input vector and the corresponding desired output are considered first. The input is propagated forward through the network to compute the output vector. The output vector is compared with the desired output, and the errors are determined. The errors are then propagated back through the network from the output to input layer. The process is repeated until the errors being minimized. The input layer of network contains 18 neurons, corresponding to 3 bands of IRS-1D image (six features for per pixel in each band). The output layer contains 6 neurons corresponding to 6 pre-defined land cover categories in the classification.

When designing a neural network, one crucial and difficult to determined parameter is the number of neurons in the hidden layers (Bischof et al., 1992). The hidden layer is responsible for internal representation of the data and the information transformation input and output layers. If there are too few neurons in the hidden layer, the network may not contain sufficient degrees of freedom to form a representation. If too many neurons are defined, the network might become over-trained (Heermann et al., 1999). Therefore, an optimum design for the number of neurons in the hidden layer is requied. In this research, we used one hidden layer with a number of different neurons to determine the suitable network. Table 1 shows the error of network for four cases.

Class	No. of points	Error (percent)			
		(A) n=5	(B) n=10	(C) n=15	(D) n=20
1	910	3.84	0.66	0.00	1.87
2	696	2.87	8.19	0.72	3.30
3	272	2.20	0.00	0.94	0.74
4	1242	42.75	33.90	2.98	96.94
5	547	2.38	4.02	6.03	13.35
6	290	4.48	2.07	0.00	0.00
Average		9.75	8.14	1.78	19.37

Table 1. Four networks with different neurons in hidden layer. 'n' is the number of neurons in hidden layer.

As seen in table 1, a network with 15 neurons in hidden layer have the minimum error, so it is the best case for designing network in this case. It is also as fast training as the other ones. The training diagram for each network is shown in figure 6, with mean square error of 0.01. Figure 6 shows the number of repetition to receiving the error, in A, B, C, and D.

3.3 The comparison between BPNN and MLC

The proposed method has been tested on IRS-1D 8-bits bands 2, 3 and 4 of LISS-III sensor fused with PAN data to construct an image with 5,8m spatial resolution of northern regions of Iran.

In Maximum Likelihood Classification (MLC) method 6 training sites has been selected for classification. The results were compared with a neural network with 15 neurons in hidden layer. The comparison results are:

The overall accuracy in MLC method is 75.00% and in BPNN method is 85.19%.

- In MLC method, the unclassified region is very small and is about 2.5 percent of the whole classified image. But in BPNN method, the unclassified region is about 30 percent of the classified image. This is because of: the higher accuracy of N.N and as a result, it is more sensitive to the training sites. Also, since a 3*3 window moves for feature extraction on the image, the feature on the edges aren't classified.

4. CONCLUSIONS

-In comparison of back propagation neural network method with maximum likelihood method in classification on IRS-1D image, the BPNN presented to be more accurate than MLC method. The overall accuracy in MLC method is 75.00% and in BPNN method is 85.19%.

- The BPNN method showed that if the accuracy of NN is higher, the sensitivity to training sites is increased.

- However, in BPNN method, if the training sites are not sufficient or they aren't selected from the whole image, unclassified regions are more when compared with unclassified regions in MLC method.

-In this research, features have been extracted using first-order probability distribution. These features extracted in this way are more suitable for images that have similar textures.

- Levinberg-Marquardt training is faster than the general delta rule and it needs less input pattern to training than the other one.

REFRENCES

Benedikson, J. A., P.H. Swian,, and O.K Erosy, (1990), "Neural network approaches versus statistical methods in classification of multisource remote sensing data". IEEE Transactions on Geoscience and Remote sensing, Vol. 28.

Bischof, H., .Schneider, and A.J.Pinz (1992) "Multispectral classification of Landsat-images using neural network," IEEE Trans. Geosci. Remote sensing, vol. GE-30,no. 3,pp.482-490.

Chen Y.Q., (1995), "Novel Techniques for Image Texture Classification". PH.D. thesis, University of Southampton

Coifman, R.R.; M.V Wickerhauser (1992), "Entropy-based algorithms for best basis selection," IEEE Trans. on Inf. Theory, vol. 38, 2, pp. 713-718.

Hagan, M. T., and M. Menhaj,(1994) "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993.

Heerman P.D. and N.Khazenie, (1992), "Classification of multispedtral remote sensing data using a bach propagation neural network," IEEE Trans, Geosci. Remote Sensing, vol.GE_30,no.1,pp.81-88.

Hepner, G. F., Logan, T., Ritter, N. and Brayant, N., (1990), Artificial neural network classification using a minimal training set: comparison to conventional supervised

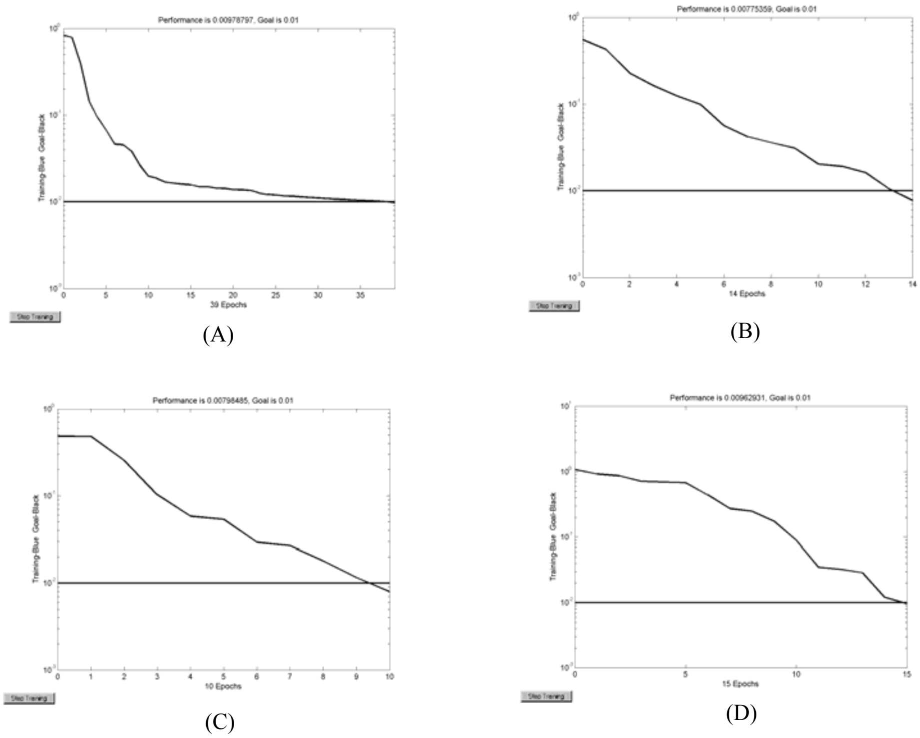


Figure 6. Training diagrams for 4 cases of hidden layer and number of iterations; A: 5 neurons in hidden layer (39 epochs); B: 10 neurons in hidden layer (14 epochs) C: 15 neuron in hidden layer (10 epochs); D: 20 neurons in hidden layer (15 epochs)

classification. Photogrammetric Engineering and Remote Sensing, 56, 469-473

Mohanty K.K., and T.J. Majumdar (1996), " An artificial neural network (ANN) based software package for classification of remotely sensed data". Computers & Geosciences Vol. 22, No.1, pp. 81-87

Pratt,W. K (2001), "Digital Image Processing”. Los Altos, California

Rumelhart, D.E, G.E. Hinton, R.J. Williams,(1986) "Learning internal representations by error propagation," D. Rumelhart and J. McClelland, editors. Parallel Data Processing, Vol.1, Chapter 8, the M.I.T. Press, Cambridge, pp. 318-362.