EFFECTS OF RANDOMIZING PATTERNS AND TRAINING UNEQUALLY REPRESENTED CLASSES FOR ARTIFICIAL NEURAL NETWORKS

Youngsup Kim[$] and Tommy L. Coleman[#]
[$]GIS Institute, School of Computer Science and Electronic Engineering
Handong University, Pohang, Korea, 791-708
[#]Center for Hydrology, Soil Climatology, and Remote Sensing (HSCaRS)
Alabama A&M University, Normal AL 35762, USA

ABSTRACT

Artificial neural networks (ANN) have been successfully used for classifying remotely sensed imagery. However, ANN still is not the preferable choice for classification over the conventional classification methodology such as the maximum likelihood classifier commonly used in the industry production environment. This can be attributed to the ANN characteristic built-in stochastic process that creates difficulties in dealing with unequally represented training classes, and its training performance speed.

In this paper we examined some practical aspects of training classes when using a back propagation neural network model for remotely sensed imagery. During the classification process of remotely sensed imagery, representative training patterns for each class are collected by polygons or by using a region-growing methodology over the imagery. The number of collected training patterns for each class may vary from several pixels to thousands. This unequally populated training data may cause the significant problems some neural network empirical models such as back-propagation have experienced.

We investigate the effects of training over- or under- represented training patterns in classes and propose the pattern repopulation algorithm, and an adaptive alpha adjustment (AAA) algorithm to handle unequally represented classes. We also show the performance improvement when input patterns are presented in random fashion during the back-propagation training.

1.0 INTRODUCTION

Multispectral classification is one of the most often used methods to extract information from remotely sensed imagery. The classification of remote sensed imagery typically involves the collection of spectral samples to discriminate a fixed set of classes in the imagery. These samples are often referred to as training sites in traditional remote sensing or input patterns when using a neural network. The samples usually represent relatively homogeneous examples of known land-cover types. Based on the samples collected from the imagery, conventional (hard) classification algorithms determine a discriminant function or extract spectral and spatial characteristics (means, standard deviation, covariance matrices, etc.) to perform the classification.

However, while conventional classification usually uses a parametric method, artificial neural networks (ANN) are nonparametric models that have been successfully used for classifying remotely sensed imagery(Frizzelle, 2001). However, ANN still is not the preferable choice for classification over the conventional classification methodology such as the maximum likelihood classifier commonly used in the industry production environment. This can be attributed to the ANN characteristic built-in stochastic process that creates difficulties in dealing with unequally represented training classes, and its training performance speed.

We examined some practical aspects of training classes when using a back propagation neural network model for remotely sensed imagery. After a brief introduction of the neural network and back-propagation, we show the performance improvement when input patterns are presented in random fashion during the back-propagation training. Afterwards the pattern repopulation algorithm, and an adaptive alpha adjustment (AAA) algorithm are presented to handle unequally represented classes.

2.0 BACK-PROPAGATION

The most popular class of the artificial neural network applied is a supervised multilayer perceptrons with back-propagation as a learning algorithm. Figure 1 shows a typical back-propagation neural network with one input layer, one hidden layer and one output layer. In this 3-tier network, all nodes in a given layer are connected to each node in the subsequent layer. Each connection has an associated weight that can be excitatory or inhibitory. The development of the back-propagation learning algorithm for determining weights has made these networks the most popular among researchers and users of neural networks (Jain, 1996). The back-propagation algorithm is a gradient descent method to minimize the total squared error of the output computed by the network. After training or once determining weights, it can be used to classify the new input patterns. Simple feeding of input patterns to the network will produce output signals for each class in the feed-forward fashion. Therefore, it can be very fast and use a massive parallel computing power.

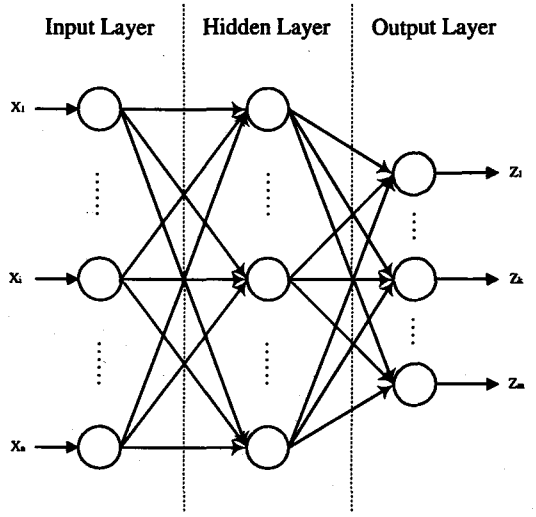

Figure 1. A Typical Back-Propagation Neural Network With One Hidden Layer

The basic steps for training a network using back-propagation in on-line or pattern-mode learning (Jain, 1996; Reed, 1998) are as shown;

Step 1: Initialize the weights to small random values.
Step 2: Feed forward a training pattern through the network.
Step 3: Compare the outputs with the desired values.
Step 4: Calculate the derivatives ($\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$) of the error with respect to the weights. A learning rate η is a small positive constant.
Step 5: Adjust the weights to minimize the error

$$w(t+1) = w(t) - w_{ij} \quad (1)$$

Step 6: Go to step 2 and repeat for the next pattern until the error is acceptably small or time is exhausted.

An alternative to pattern-mode is batch-mode learning in updating weights. In batch-mode all patterns are evaluated to obtain the derivative in Step 4. These weights are summed for an epoch (or one iteration for all patterns) and used to update the weights once. Even though this batch-mode learning closely approximates gradient descent, it requires many epochs to update the weights that reduce the error by a small amount.

3.0 ALGORITHMS PROPOSED AND EXPERIMENTS

3.1. THE NEURAL NETWORK ARCHITECTURE AND DATA SETS USED

The neural network architecture used for the experiment had three layers, with three neurons in the input layer (one for each spectral band), four or five neurons in the output layer (one for each class) and eight or ten neurons in the hidden layer. The number of neurons in the hidden layer was set empirically to two times that of the output neurons (or classes). First, a computer program artificially generates those class patterns as an ideal case. This data set is called COMP data set through out this paper. In this COMP data set, five classes have an unequally represented number of patterns such as 10, 100, 450, 700, and 1000, respectively. The meaning of "unequally represented classes" is explained in the following sections. Additionally, two training data sets called LAND_1 and LAND_2 were extracted from the LANDSAT multispectral imagery data. The LAND_1 data set has 1253 patterns with four classes that have 22, 85, 122 and 1024 patterns, respectively. The LAND_2 data set has 2649 patterns with five classes that have 1151, 842, 477, 143, and 36 patterns, respectively. A fixed learning rate of 0.25 is used for all classes. The momentum of the network is set to 0 for simplicity. The resubstitution (or R method) and the holdout (H method) were used for training and testing (Toussaint, 1974).

3.2 TRAINING PATTERNS RANDOMIZATION PROPOSED

The training patterns presented to the network in a random, constantly changing order avoid cyclic effects that could prevent it from reaching the convergence (minimum) in the learning process of the back-propagation. As shown in Figure 2, the randomization of training patterns helps the convergence of the back-propagation learning algorithm. Therefore, the randomization was applied for the rest of the experiments throughout this paper. The proposed algorithm of the training patterns randomization is summarized as follows:

Step 1: Read all training patterns into an array data structure (list) in memory.
Step 2: Select a pattern from the minority classes in the list in sequence.
Step 3: Select another pattern from majority classes in the list at random.
Step 4: Swap two classes locations with each other.
Step 5: Repeat Step 3 and 4 for all minority classes patterns.
Step 6: Start the neural network training using back-propagation.
Step 7: Optionally, repeat the randomization after every epoch.

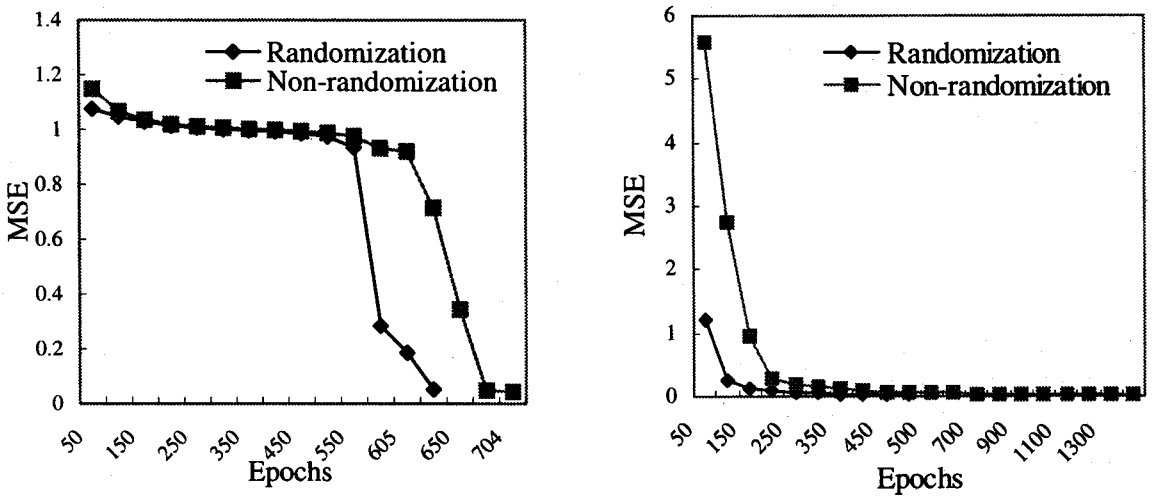

Figure 2. Randomization Performance For COMP Data Set(Left) and LAND_1 Data Set(Right)

3.3 ALGORITHMS FOR UNEQUALLY REPRESENTED CLASSES

During the classification process of remotely sensed imagery, representative training patterns for each class are collected by drawing polygons or by using a region-growing methodology over the imagery. The number of collected training patterns for each class may vary from several pixels to thousands. These unequally populated training patterns may cause the significant problems some empirical models of neural networks such as back-propagation. Because the back-propagation algorithm assumes that all training patterns are equally important during learning process. Some algorithms to overcome an under- or over-populated input pattern problem have been presented (DeRouin, 1991; Raymond, 1990). A Dynamic Training Set was proposed that dynamically increases or decreases the number of patterns in the training based on the mean squared error (Raymond, 1990). A learning rate adjustment algorithm has also been proposed known as Alpha Adjustment Training (DeRouin, 1991). In this algorithm, the learning rate adjustment is made for each training class depending on the number of training patterns rather than using one learning rate for all the patterns of classes.

3.4 THE TRAINING PATTERN REPOPULATION

In the remotely sensed image classification environments, it is not necessarily true that under-represented class patterns are less important than others and vice versa. Overly populated training pattern classes (or majority classes) usually have many duplicated patterns since they were likely to be collected from a large homogenous land cover area. Under-represented training pattern classes (or minority classes) might be important, but were collected from smaller land cover areas. The training sets with minority and majority classes is called an unequally trained data set rather than a poorly trained set. Even if all training patterns are collected evenly for each class, it could be a poorly trained pattern set if a class has many duplicated patterns. The proposed algorithm is both to reduce the population of the over-represented class training patterns and to duplicate that of the under-represented ones in random. The algorithm is summarized as shown below:

Step 1: Read all the training patterns into an array data structure in memory.

Step 2: Compute an average number of training patterns of the data set.
Step 3: Identify the number of patterns in each class that exceed or fall short
Step 4: Depopulate the training patterns from majority classes at random.
Step 5: Populate the training patterns by duplicating patterns in minority classes.
Step 6: Start the neural network training procedure.

This repopulation algorithm keeps the total number of the training patterns the same before or after repopulating the training patterns. Figure 3 shows that the training pattern repopulation is very effective to have a significant performance increase. This algorithm is not expensive computationally since it is preprocessing before the major iteration of the back-propagation algorithm. In terms of programming, it is simple and cost-effective in data structure since the total number of training data can be kept the same as before. However, the over-represented class may have a slight chance of not reaching 100% classification level when the original data is fed back the network. It is reasonable since only a subset of the original training patterns are used.

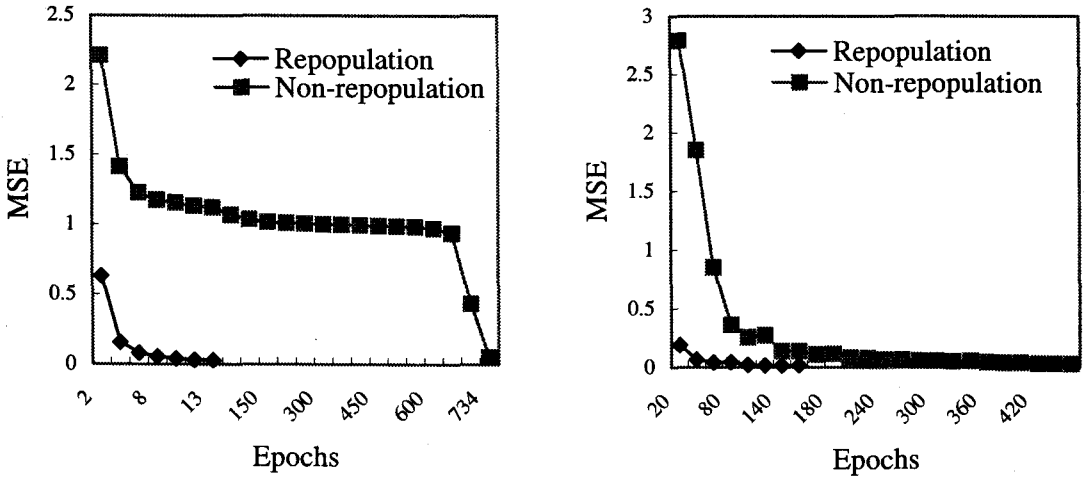

Figure 3. The Training Pattern Repopulation For COMP and LAND_1 Data Sets.

3.4 ADAPTIVE ALPHA ADJUSTMENT (AAA) ALGORITHM

The alpha (learning rate) adjustment training adjusts the learning rate such that the new rate is reciprocal in some extent to the number of training patterns. It can be expressed as shown below (DeRouin, 1991):

$$\alpha_x = \alpha[(1-\frac{n_x}{m})\frac{N}{N-1}]^p \qquad (2)$$

where, m is the total number of training patterns that have been presented, n_x is the total number of class x training patterns that have been presented to the network, and N is the number of output classes. The factor of $N/(N-1)$ normalizes the learning rate so that, in the case of equal representation of classes, the attention factor approaches 1. The exponent p determines the degree of emphasis to the attention given to minority classes, but lowers the attention from majority classes at the same time.

DeRuouin's learning rate adjustment method provided a simple and computationally inexpensive method to train the net with unequally under represented class patterns. This algorithm has shown a poor classification performance for over-represented classes (or majority classes) when the value of exponent p goes higher. The poorly trained patterns can be categorized as one or a mixed form of over-represented, under-represented, and redundantly/duplicated represented training patterns. The learning rate adjustment algorithm based on the population of patterns suffered from the real world data set that is less predictable and less well-behaved (DeRouin, 1991). Unfortunately, a good method has not been known to determine the optimal value of the exponent p.

The proposed Adaptive Alpha Adjustment (AAA) algorithm considers that the learning rate becomes a function of the population of training patterns as well as the mean square errors. The new adaptive learning rate is calculated as shown below:

$$\alpha_{xx} = \alpha_x[(1 - \frac{MSE_x}{\sum_i MSE_i})\frac{N}{N-1}]^{-1} \tag{3}$$

where, α_x is the learning rate for class x from the alpha adjustment algorithm, α_{xx} is a new learning rate based on the mean square errors. With $p = 1$ in the equation (2), the equation (3) can be simplified as follows;

$$\alpha_{xx} = \frac{1 - \frac{n_x}{m}}{1 - \frac{MSE_x}{\sum_i MSE_i}} \tag{4}$$

The proposed algorithm is summarized as follows:

Step 1: Set the learning rate based on alpha-adjustment training for the first epoch.
Step 2: Start the neural network back-propagation procedure.
Step 3: Iterate all the input patterns for the back-propagation (one epoch).
Step 4: Compute Mean Square Error (MSE) for each class.
Step 5: Adjust the learning rate by Equation (4)
Step 6: Repeat Step 3 through 5 until the stopping conditions are met.

The table 1 shows the performance of those algorithms mentioned in the paper in terms of the number of epochs. The training pattern repopulation algorithm outperformed the other algorithms discussed. The alpha adjustment algorithm with a higher value of the exponent p value tends to hurt the convergence of the majority class significantly. The reason is that the algorithm sets the learning rate too low for the majority class training patterns. The newly proposed AAA algorithm showed better performance than the alpha adjustment algorithm in many cases.

Randomization	OFF				
Confidence	97%				
Algorithm	REPOP	AAA	Alpha Adjustment		
			p=1	p=3	p=6
COMP	42	87	515	265	111
LAND_1	223	636	727	925	*
LAND_2	1217	1007	1263	1385	1917

Randomization	ON				
Confidence	97%				
Algorithm	REPOP	AAA	Alpha Adjustment		
			p=1	p=3	p=6
COMP	6	31	490	234	83
LAND_1	50	144	140	1114	*
LAND_2	64	114	220	151	201
Confidence	99%				
Algorithm	REPOP	AAA	Alpha Adjustment		
			p=1	p=3	p=6
COMP	12	31	492	232	89
LAND_1	92	316	320	2249	*
LAND_2	93	228	384	276	373
Confidence	100%				
Algorithm	REPOP	AAA	Alpha Adjustment		
			p=1	p=3	p=6
COMP	70	102	493	238	485
LAND_1	179	2360	2397	*	*
LAND_2	*	*	*	*	*

Table 1. The Performance Comparisons in terms of the Number of Epochs. The * mark indicates that it did not converge until 3000 epochs. The confidence is computed by (number of patterns converged)/(total number of patterns) * 100. REPOP stands for the training pattern repopulation algorithm.

4.0 CONCLUSION

When using the back propagation to find the weights of a neural network, the randomization of training patterns can improved significantly its convergence performance. It is true for both evenly and unequally populated training data set. The alpha adjustment algorithm with a higher value of the exponent p sets the learning rate too low for the majority class training patterns. The newly proposed adaptive alpha adjustment (AAA) algorithm usually showed better performance than the other cases. The algorithms proposed should be tested more with real world data sets that have noise and less separability between class patterns in order to verify their validity. The training pattern repopulation algorithm simply outperformed the other algorithms.

5.0 ACKNOWLEDGMENT

The authors wish to acknowledge the contribution of ZI Imaging Cooperation in Huntsville, Alabama for their support and for providing the image processing software that made

this study easier. Particular acknowledgement must be made to Mr. Orrin Long in ZI Imaging Cooperation who informed us of the need for this research. Special thanks should go to my fellow researchers, GooHo Kown and HwaJung Lee, for their contributions of testing and proving the algorithms.

This work has been supported by the Center for Hydrology, Soil Climatology, and Remote Sensing (HSCaRS) and the Agricultural Experiment Station, Alabama A and M University. The preparation of this technical manuscript was supported in part by NASA Grant No. NCC8-140. Any use of trade product or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government. This paper was submitted to present at the Seventh International Conference on Remote Sensing for Marine and Coastal Environments, Miami, Florida, 20-22 May 2002.

6.0 REFERENCES

- Anil K. Jain, Jianchang Mao, K.M. Mohiuddin, Artificial Neural Networks: A tutorial IEEE Computer Magazine, pp 31 – 44, March 1996
- DeRouin, E., and Brown, J. Fausett, L., Neural Network Training on Unequally Represented Classes, Intelligent Engineering Systems Through Artificial Neural Networks. New York: ASME Press, pp. 135-141, 1991
- Frizzelle, B. G., Moody, A., Mapping, Continuous Distributions of Land Cover: A Comparison of Maximum-Likelihood Estimation and Artificial Neural Networks, PE & RS, Vol. 67, No.6, pp.693-705, June 2001
- G. T. Toussaint, Bibliography on Estimation of Misclassification, IEEE Transactions on Information Theory, IT-20, 1974
- L. Fausett, Fundamentals of Neural Networks, Readings, Prentice Hall.
- R.D. Reed, R. J. Marks II, Neural Smithing, Reading, MIT, 1998
- Raymond K.M. Cheung, Irving Lustig, Alain L. Kornhauser, Relative Effectiveness of Training Set Patterns for Back Propagation, IJCNN, 1, pp.673-678, 1990