DICTA2002: Digital Image Computing Techniques and Applications, 21-22 January 2002, Melbourne, Australia.

Image Segmentation Employing Neural Networks and Multivariate Analysis

Yueh-Hsiang Liao and Gwong C. Sun
CECS Department, University of Louisville
Louisville, KY 40292, U.S.A.
E-mail: liao@louisville.edu , gcsun001@louisville.edu

Abstract

This paper addresses a neural network model employing multivariate analysis to implement the image segmentation task. Image segmentation is the process of partitioning an image into related regions or sections based on the characteristics. Each pixel of the segmented image belongs to one of the predefined segment classes. In order to classify the pixels of the image, a neural network is applied. The proposed neural network model contains a single hidden layer with a small number of nodes determined by the desired percentage of variation in the selected principal components obtained from a principal component analysis of the training sample features. The weights between the input layer and the hidden layer are computed directly using principal component analysis. The weights between the hidden layer and the output layer are computed by applying the delta rule iteratively in the training procedure.

The significance of this research is that the proposed model considerably reduces number of hidden nodes and training time, compared with the neural network model trained by backpropagation algorithm, and still maintains the expected accuracy for the classification results.

1. Introduction

Image segmentation techniques have been developed for many years; there are various methods and algorithms to deal with the image segmentation tasks. Increasing the accuracy and efficiency of the image segmentation are also the concerns of the designs and algorithms in the image segmentation. The main goal of our proposed image segmentation model is trying to meet these requirements mentioned above.

Our proposed model is a neural network based model employing the statistical multivariate analysis techniques. There are one hidden layer and one output layer in the proposed model, the output of the hidden layer are independent normally distributed variates. The details of the design of our proposed model and the results of simulation and practical application are going to be elaborated in the following paragraphs.

2. The structure of proposed pattern recognizer

The structure diagram of the proposed neural network model [1][2] for image segmentation is shown in Figure 1. In the following paragraphs, the details of the structure and the training method of the proposed model are going to be depicted.

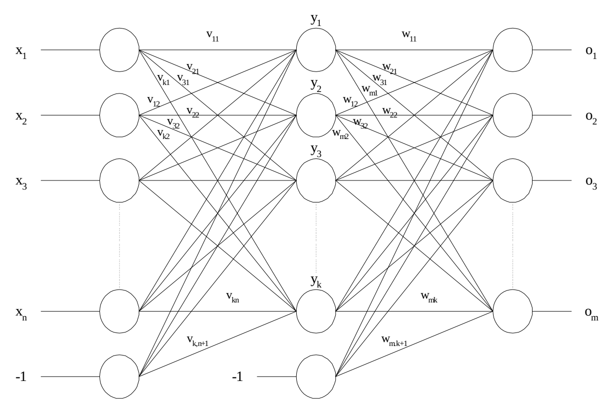

Figure 1. Structure diagram of neural network model

The proposed neural network model [3], which is shown in Figure 1 above, is composed of input layer, one hidden layer and output layer. We assume this image segmentation method is classification-based, the information of a pixel and its neighbor pixels is used to classify each pixel into one of the predefined segment classes. Therefore, each pixel and its neighbor pixels are treated as a pattern feature vector. There are n+1 input nodes representing n pattern features of interest and one bias node with input -1. There are k+1 hidden nodes representing the most k significant principal components y_1, y_2,…, y_k and one bias node with input -1, where k is not greater than the number of pattern feature n. The number of nodes of the hidden layer is decided by the percentage of the variation in the selected principal components obtained from a principal component analysis of the training sample feature vectors. The details of finding the hidden nodes are going to be described in the later paragraphs. There are m output nodes, which are m pattern categories or pattern classes, indicating the pattern classes recognized by the proposed neural network model.

The weights between the input layer and hidden layer are obtained by applying principal component analysis. In order to describe the method to find the weights between the input layer and hidden layer, we will brief the mathematical background of principal component analysis.

3. Mathematical background review

Assume the random variables x_1, x_2,…, x_n of interest have a certain multivariate normal distribution with mean vector m and covariance matrix å. It is possible to find a set of new variables, y_1, y_2,…, y_n, called principal components, which are linear combinations of the original random variables, but they are uncorrelated [4].

We start by considering a set of sample population consisting of M n-dimensional pattern feature vectors [5]. The form of feature vector is:

$$X = (x_1, x_2, ..., x_n)^T \tag{1}$$

The mean vector of the sample population is calculated by

$$m = \frac{1}{M}\sum_{i=1}^{M} X_i \tag{2}$$

The covariance matrix å can be calculated by

$$\Sigma = \frac{1}{M-1}\sum_{i=1}^{M}(X_i - m)(X_i - m)^T \tag{3}$$

In many applications, the variables x_1, x_2,…, x_n are represented in more than one different unit of measurement, thus the set of normalized variates z_1, z_2,…, z_n should be applied. The normalized variates are

$$z_i = \frac{x_i - m_i}{s_i} \tag{4}$$

where μ_i is the i^th element of the mean vector m and σ_i is the square root of the i^th diagonal element of the covariance matrix å.

The covariance matrix of the normalized variables z_1, z_2,…, z_n is **C** which is the correlation matrix of original variables x_1, x_2,…, x_n. The i^th principal component y_i is defined as

$$y_i = \sum_{j=1}^{n} l_{ij} z_j \tag{5}$$

which is the i^th largest portion of total variance. In order to find the principal components, the eigenvalues λ's and eigenvectors l's needed to be computed, we will have that λ_i is the i^th largest eigenvalue of **C** and l_i is the eigenvector associated with λ_i. Thus the linear compound y_i, the i^th principal component of the original variables x_1, x_2,…, x_n, can be found, the first k largest principal components plus the bias node with the value of -1 are the nodes of the hidden layer. The weights between the input and hidden layers, **v**'s, are obtained from the principal component process as

$$v_{ij} = \frac{l_{ij}}{s_j}\ ,\ i = 1,2,...,k\ \ and\ \ j = 1,2,...,n \tag{6}$$

and

$$v_{i,n+1} = \sum_{j=1}^{n}\left(l_{ij}\frac{m_j}{s_j}\right)\ ,\ i = 1,2,...,k \tag{7}$$

where l_{ij} is the j^{th} element in the i^{th} eigenvector associated with the i^{th} largest eigenvalue of the correlation matrix **C**. There are only small number of principal components going to be applied to the nodes of the hidden layer, the number of principal components applied to the hidden layer is determined by the percentage of variation of the first k principal components to total variance. The set of the first k significant principal components, which are based on the predefined level of significance, are computed from the matrix **C** and the weighs **v**'s are plugged in.

The output nodes are computed by the activation

$$f(net_i) = \frac{1}{1+e^{-net_i}} \tag{8}$$

where net_i is

$$net_i = \sum_{j=1}^{k} w_{ij} y_j - w_{i,k+1} \tag{9}$$

The weights between the hidden layer and output layer are computed by employing the delta rule iteratively as

$$w_{ij}(t+1) = w_{ij}(t) + h\left(d_i - o_i(t)\right)o_i(t)\left(1 - o_i(t)\right)y_i \tag{10}$$

where the η is the learning rate, $o_i(t)$ is the output of the i^{th} node in the output layer at the training iteration t, and d_i is the desired output which is 1 if the training pattern belongs to the class i and 0, otherwise. The training process will be done when the total error less than the predefined error threshold or the number of the training iterations exceeds the preset maximum iterations.

4. Simulation results

In this computer simulation, multi-normal distributed feature patterns with different number of pattern features are trained in the fully connected perceptron model with one hidden layer by traditional backpropagation (BP) algorithm [6] and principal component analysis (PCA) method. The comparison results are discussed in the following paragraphs.

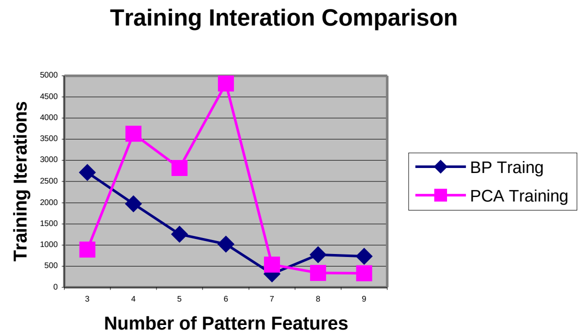

Figure 2. Training iteration comparison

The comparison of training iterations for two different training methods is shown in Figure 2 above. The training iterations of training by PCA method are obviously higher than that of training by BP algorithm between the number of pattern features 3 and 7, the training iterations of PCA training decreasing steeply from the number of pattern features 7. We can see the training iterations of PCA training are apparently lower than that of the BP training in the higher number of pattern features. The PCA training reduces the training iterations considerably in the higher dimensional pattern feature input.

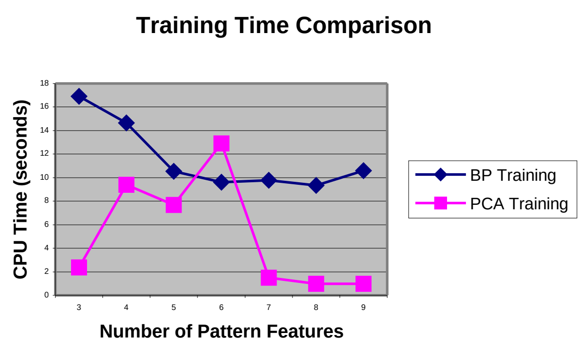

Figure 3. Training time comparison

The comparison of training time between two training methods is shown in Figure 3 above. The training time of PCA training is apparently shorter compared with that of BP training in almost all number of pattern features. The training time is especially shorter when the number of pattern features is larger. This diagram demonstrates the computation time improvement and efficiency of the PCA training compared with that of the BP training.

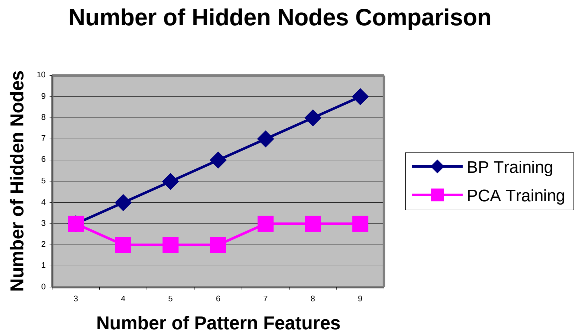

Figure 4. Number of hidden nodes comparison

The comparison of the number of the hidden nodes between two training methods is shown in Figure 4 above. The number of the hidden nodes in the hidden layer of the PCA training classifier is smaller than that of BP training classifier. The number of hidden nodes in the PCA training classifier is comparatively small when the higher dimensional input feature patterns are applied in the training procedure.

5. Image segmentation result

An aerial image is employed to demonstrate the classification result of PCA training classifier. There are two 256x256 pixel images, the original aerial image and the segmented aerial image, are illustrated in the Figure 5 and Figure 6 respectively.

Figure 5. Original aerial image

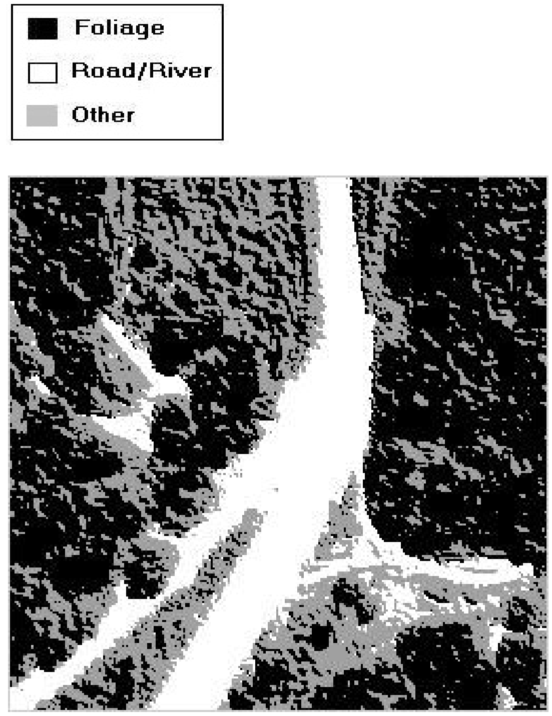

Figure 6. Segmented aerial image

There are three predefined segment classes, the foliage class, road or river class, and other class, in the original aerial image. We assume all of the pixels of the image are multi-normal distributed; they belong to all of those predefined segment classes individually. Before doing the training procedure, several training sample patterns for each segment class needed to be selected from the image. Each pattern vector consists of 9 pixels, 8 neighbor pixels and itself, center pixel. We need to utilize those sample patterns to perform the PCA training for the pattern classifier. After the training procedure is done, the well-trained pattern classifier is going to be employed to classify the original image into three predefined segment classes, which are marked by three different gray level pixels. Every pixel and its 8 neighbor pixels are treated as an input pattern feature vector into the well-trained classifier; therefore, the center pixel is classified into one of those three segment classes. We need to repeat this process from left to right and top to down till all of the pixels of the image are classified into the segment classes they belong to. The final segmented image is shown in the Figure 6. We can easily observe the distribution of three segment classes by different gray level pixels. By comparing the original image and the classified image, the proposed model has done the expected task of image segmentation.

6. Conclusion

In this paper we presented a method for training a neural network model to implement the image segmentation. This is a classification-based image segmentation model. The computation time, the number of training iterations, the number of hidden nodes and the efficiency by applying this training method are much improved, especially obvious in the higher dimensional pattern feature inputs. This method also maintains the same accuracy of classification compared with that of the traditional backpropagation training. The pattern classifier trained by the proposed method cannot only be applied in the image segmentation, but also other multivariate normal distributed data set. The experimental results look promising in the image segmentation field. We are going to find some other applications applicable for this research and keep looking for better ways to improve this research.

7. References

[1] Earl Gose, Richard Johnsonbaugh, and Steve Jost, "Pattern Recognition and Image Analysis," Upper Saddle River, NJ: Prentice Hall, 1996, pp. 227-258

[2] Sing-Tze Bow, "Pattern Recognition and Image Preprocessing," New York: Marcel Dekker, Inc., 1992, pp. 153-164

[3] David H. Kil and Frances B. Shin, "Pattern Recognition and Prediction with Applications to Signal Characterization," Woodbury, New York: AIP Press, 1996, pp.119-142

[4] William R. Dillon and Matthew Goldstein, "Multivariate Analysis Methods and Applications," John Wiley & Sons, Inc., 1984, pp. 23-52

[5] Donald F. Morrison, "Multivariate Statistical Methods," New York: McGraw-Hill, Inc.,1967, pp. 75-94, pp. 221-256

[6] N. K. Bose and P. Liang, "Neural Network Fundamentals with Graphics, Algorithms, and Applications," McGraw-Hill, Inc., 1996, pp.162-169