

11-3-2006

Fuzzy Neural Network Models For Multispectral Image Analysis

Arun D. Kulkarni

University of Texas at Tyler, akulkarni@uttyler.edu

Sara McCaslin

Follow this and additional works at: http://scholarworks.uttyler.edu/compsci_fac



Part of the [Computer Sciences Commons](#)

Recommended Citation

Kulkarni, Arun D. and McCaslin, Sara, "Fuzzy Neural Network Models For Multispectral Image Analysis" (2006). *Computer Science Faculty Publications and Presentations*. Paper 5.

<http://hdl.handle.net/10950/334>

This Article is brought to you for free and open access by the School of Technology (Computer Science & Technology) at Scholar Works at UT Tyler. It has been accepted for inclusion in Computer Science Faculty Publications and Presentations by an authorized administrator of Scholar Works at UT Tyler. For more information, please contact tbianchi@uttyler.edu.

Fuzzy Neural Network Models For Multispectral Image Analysis

ARUN KULKARNI and SARA MCCASLIN
Computer Science Department
The University of Texas at Tyler, Tyler, TX 75799, USA

Abstract: Fuzzy neural networks (FNNs) provide a new approach for classification of multispectral data and to extract and optimize classification rules. Neural networks deal with issues on a numeric level, whereas fuzzy logic deals with them on a semantic or linguistic level. FNNs synthesize fuzzy logic and neural networks. Recently, there has been growing interest in the research community not only to understand how FNNs arrive at particular decisions but how to decode information stored in the form of connection strengths in the network. In this paper, we propose fuzzy neural network models for classification of pixels in multispectral images and to extract fuzzy classification rules. During the training phase, the connection strengths are updated. After training, classification rules are extracted by backtracking along the weighted paths through the FNN. The extracted rules are then optimized using a fuzzy associative memory (FAM) bank. The data mining system described above is useful in many practical applications such as mapping, monitoring and managing our planet's resources and health, climate change impacts and assessments, environmental change detection and military reconnaissance.

Keywords: Fuzzy Neural Networks, Multispectral Image Analysis, Rule Extraction, Remote Sensing

1. Introduction

Conventional statistical classification techniques such as the discriminant functions, nearest neighbor classifier, maximum likelihood classifier, and various clustering techniques have been widely used in remote sensing for multispectral image analysis [1,2,3]. Neural networks provide a powerful and reasonable alternative to conventional classifiers. Neural networks are preferred for classification because of their parallel processing capabilities as well as learning and decision making abilities. Neural networks with learning algorithms such as the back-propagation are used as supervised classifiers, and self organizing networks with learning algorithms such as the competitive learning and Kohonen's feature maps are used as unsupervised classifiers. Many neural-network based decision systems have been used for

multispectral image analysis. Neural networks provide algorithms for problems such as optimization, classification, and clustering, whereas fuzzy logic is a tool for representing and utilizing data and information that possesses non-statistical uncertainty. Fuzzy logic methods often deal with issues such as reasoning on a semantic or linguistic level [4]. There are many ways to synthesize neural networks and fuzzy logic. Fuzzy neural networks provide greater representation power, higher processing speed, and are more robust than conventional neural networks. Lin and Lee [5] proposed a neural network based fuzzy inference system that maps crisp inputs to crisp outputs. Jang [6] has proposed architecture called Adaptive Network-based Fuzzy Inference System (ANFIS) that combines a fuzzy system and a neural network system. The model consists of five

layers. Pal and Mitra [7] have developed a fuzzy neural network model using a back-propagation learning algorithm. They have used the model to classify Indian Telugu language vowel sounds and to extract inference rules. Kulkarni [8] suggested fuzzy-neural network models for supervised and unsupervised classification. The model consists of three layers. The first layer is an input layer. The second layer is used for fuzzification wherein input feature values are mapped to membership functions. The last layer implements fuzzy inference rules. Until recently, FNNs have been used as “black boxes”: they can learn from training samples and successfully classify data samples, but they do not readily provide the user with any information as to how the network reached the decision. Recently, there has been growing interest in the research community not only to understand how FNNs arrive at particular decisions but how to decode information stored in the form of connection strengths in the network. They also describe techniques for extracting fuzzy rules from neuro-fuzzy systems. Mitra et al. [9] have proposed two methods for rule generation. In the first method, they have treated the network as a black box and using the training set input and the network output to generate a set of if-then rules. The second method is based on the backtracking algorithm. Wang and Mendel [10] developed a five-step algorithm for directly extracting the rules from a training data set. Kulkarni and McCaslin [11] have used a fuzzy neural network for classification of pixels in a multispectral image and have extracted classification rules using the backtracking algorithm.

2. Methodology

We describe a fuzzy-neural network model and the rule extraction algorithm. The model consists of three layers. The layers and the rule extraction algorithm are described in this section. The rule extraction algorithm can be generalized to models with multiple layers.

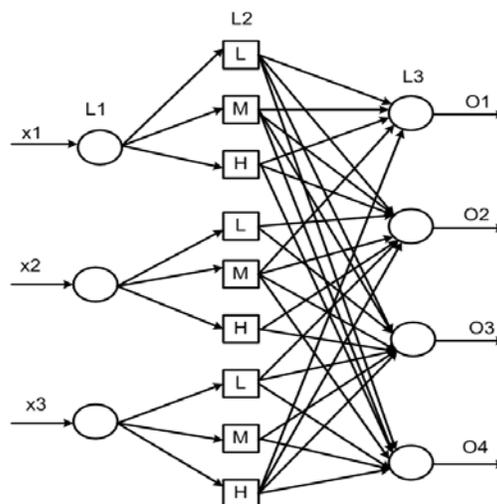


Figure 1. Three layer fuzzy neural network

2.1 Fuzzy-Neural Network Model

A three-layer fuzzy perceptron model is shown in Figure 1. The first layer is an input layer. The second layer is used for fuzzification wherein input feature values are mapped to membership values, and the last layer implements the fuzzy inference engine. We have chosen Gaussian membership functions. However, membership functions of other shapes can be used such as the triangular or bell-shaped functions. Initially, membership functions are determined using the mean and standard deviation of input variables. Subsequently, during learning, these functions are updated. Layers L₂ and L₃ represent a two-layer feed-forward network. The connection strengths connecting these layers encode fuzzy rules used in decision-making. In order to encode decision rules, we have used a gradient descent search technique. The algorithm minimizes the mean squared error between the desired output and the actual output. Layers in the model are described below. Layer L₁. The number of units in this layer is equal to the number of input features. Units in this layer correspond to input features, and they just transmit the input vector to the next layer. The output for *i*th unit is given by

$$o_i = x_i \quad (1)$$

where x_i indicates the input for unit i .

Layer L₂. This layer implements membership functions. We have used five term sets $\{very-low, low, medium, high, very-high\}$ for each input feature value. The number of units in layer L₂ is equal to the number of term sets times the number of units in L₁. The net-input and activation function for units are chosen so as to implement Gaussian membership functions, which are given by

$$f(x, \sigma, m) = \exp\left\{-\frac{(x-m)^2}{2\sigma^2}\right\} \quad (2)$$

where m represents the mean value and σ represents the standard deviation for a given membership functions. The net-input and output for units in L₂ are given by

$$\begin{aligned} net_i &= x_i \\ out_k &= f(net_i, \sigma_{ij}, m_{ij}) \end{aligned} \quad (3)$$

where $k = i \times j$, and out_k represents the output corresponds to the j th membership function that corresponds to the input x_i .

Layers L₂ and L₃. These layers implement the inference engine. Layers L₂ and L₃ represent a simple two-layer feed-forward network. Layer L₂ serves as the input layer, and L₃ represents the output layer. The number of units in the output layer is equal to the number of output classes. The net-input and output for units in L₃ are given by

$$net_j = \sum_{i=1}^n out_i w_{ij} \quad (4)$$

$$out_i = \frac{1}{1 + \exp\left\{-\left(net_i + \phi\right)\right\}} \quad (5)$$

where out_i is the output of unit i , and ϕ is a constant. Initially, weights between layers L₂ and L₃ are chosen randomly, and

subsequently updated during learning. The membership functions are initially determined based on the minimum and maximum values for input features. The algorithm minimizes the mean squared error between the desired and the actual outputs. The model learns in two phases. During the first phase of learning the weights between layers L₂ and L₃ are updated and during the second phase membership function parameters are updated to minimize the mean squared error. Once the learning is completed the model can be used to classify any unknown input sample.

2.2 Rule Generation and Optimization

Our rule generation method combines the backtracking rule extraction technique with the fuzzy associative bank technique for rule reduction and optimization. Figure 2 illustrates the process to extract and reduce the number of rules. The input to the rule extraction algorithm is a set of weight matrices of the trained neural network and training data samples. In these models, if-then rules are not explicitly represented in the knowledge base; they are generated by the inference system from the connection weight matrices. In the decision making phase, the network has already made the decision. We take a subset of the currently known information to justify the decision. The next step in rule generation is backtracking. The output of a backtracking algorithm is a collection of rules, many of which may be redundant and/or conflicting. These rules are then presented to a FAM bank, where redundant and conflicting rules are discarded using the measure of a degree of significance of the rule. The final output of a rule generation system is a set of non-redundant classification rules extracted from a sample data set. The three major components of this process are training the fuzzy neural network, extracting rules, and optimizing the rule set.

For a neural network model shown in Figure 1, in order to extract classification rules, we start with layer L₃. For every node in layer L₃ that has output value greater than the active node value (i.e.,

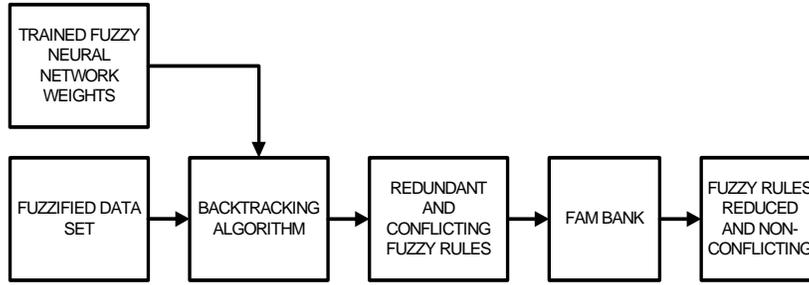


Figure 2. Block diagram for rule generation and optimization

$o_i \geq 0.5$), a path is traced to the input features. After selecting a neuron in the output layer, we select those neurons j in the layer L_2 that have positive impact on the conclusion at output neuron j . The activity level z_{ij} of any link is calculated as the product of the weight w_{ij} between node i and j and the output o_j of node j in layer L_2 ; and a path backward from that node was considered only if the activity level is greater than the user set active link threshold value.

$$z_{ij} = w_{ij}o_j \quad (6)$$

If z_{ij} is greater than the active link threshold, the feature and the membership function involved are recorded. These form the antecedent parts of the fuzzy rules. After all paths back to layer L_1 have been investigated as described, rules encompassing all possible combinations of features and membership functions recorded are produced. Since there are many data pairs, and each pair generates one or more rules, it is highly probable that there will be some conflicting rules, i. e., rules may have the same IF part but different THEN part. One way to resolve this conflict is to assign a degree of significance to each rule generated from the data pair, and to accept only the rule from a conflict group that has the maximum degree of significance. We define the degree of significance of rule as

$$D = \mu_0\mu_1\dots\mu_n o_i \quad (7)$$

where D is the degree of significance of the rule, and μ_i represents the degree of membership for feature i . For example, consider a rule “if x_1 is low and x_2 is medium the class is ω_3 ”. The degree of significance is given by

$$D = \mu_{x_1}^{low} \mu_{x_2}^{medium} o_3 \quad (8)$$

The extracted rules are then placed in the FAM bank. Figure 3 shows a FAM bank with two input feature system that uses five membership functions. The ‘1’ in a cell indicates the existence of a rule. For example a cell in the upper right hand corner is a rule that corresponds to the antecedent

			X1		
			M	H	VH
VL					1
L		1			
X2 M				1	
H	1			1	
VH	1				

Figure 3. Fuzzy associative memory

“If x_1 very-high and x_2 very-low”.

For a fuzzy-neural network model with four layers the rule extraction process is more complex because of existence of a hidden layer. In this model the path with the maximum impact from the output neuron to the input features is traced using the Equation (9).

$$z_{ij} = \max \left\{ \max_o (w_{ij}o_j) + \max_h (w_{ij}o_j) \right\} \quad (9)$$

For both networks, the FAM bank rule reduction methodology is the same. First the degree of significance file is normalized such that the degree is replaced by its percentage of the maximum degree found. In order to map a rule to a cell of the FAM bank the first step is to determine whether each feature is represented in the rule. If so, then the corresponding cell to that combination of features is checked. The rule is added if either there is no rule present in the cell or the degree of significance of the rule under consideration is greater than the degree of significance of the rule already present in the cell. It is obvious that this method will eliminate redundant and/or conflicting rules by recording only rules with the highest degree of significance. If one or more features are absent in the antecedent part, then an entire row or column may have to be investigated.

3. Results and Conclusion



Figure 4. Chernobyl scene (original scene)

We have developed software to simulate fuzzy neural network models and to generate classification rules. We used fuzzy neural network models as a supervised classifier to classify pixels in a multispectral image and to generate fuzzy rules. We analyzed Landsat-4 Thematic Mapper data. We analyzed two Landsat scenes. The first scene represents nuclear plant in Chernobyl. The scene was classified using the fuzzy

perceptron. Each pixel was represented by a vector of five reflectance values. We used reflectance values in bands 2, 3, 4, 5, and 7 as input features because these bands showed the maximum variance. We used five linguistic term sets {*very-low*, *low*, *medium*, *high*, *very-high*} to represent a reflectance value of a pixel. During the training phase the networks were trained using training set data. We selected five training areas, each of the size 10x10 pixels that represented five classes: matured crop, harvested field, vegetation, built-up area, and water. Each class was represented by a small homogeneous region. The training set areas were chosen interactively by selecting homogenous regions from the raw image displayed on the computer terminal. The target output vectors for five classes were defined as (1, 0, 0, 0, 0), (0, 1, 0, 0, 0), and (0, 0, 1, 0, 0). Only a small fraction (500 pixels) of the entire data set (256134 pixels) was used as training samples. We have considered a portion of the scene and it is of the size 512 columns and 512 rows. The original scene is shown in Figure 4. The spectral signatures for five classes are shown in Figure 5. The optimum rule set was defined as a rule set with comparatively fewer rules and over all accuracy above 90%. The optimum rule set was obtained with link threshold value of 0.4 and rule threshold value of 0.3. The optimum rule that was generated with the trained fuzzy perceptron models is shown below:

R1: If band-2 is *low* and band-3 is *low* and band-4 is *low* and band-5 is *very-low* and band-7 is *very-low* then *class* is *water*.

R2 If band-2 is *low* and band-3 is *low* and band-4 is *medium* band-5 is *medium* and band-7 is *low* then *class* is *matured-crop*.

R3: If band-2 is *medium* and band-3 is *medium* and band-4 is *high* and band-5 is *very-high* and band-7 is *medium* then *class* is *vegetation*.

R4: If band-2 is *medium* and band-3 is *high* and band-4 is *medium* and band-5 is *high* and band-7 is *medium* then *class* is *built-up area*.

R5: If band-2 is *high* and band-3 is *very-high* and band-4 is *very-high* and band-5 is

very-low and band-7 is very-low then *class* is *harvested-field*.

In order to evaluate a generated rule set, a fuzzy inference system with the generated rule set as a knowledge base was built. The classification accuracy of the fuzzy inference system depends on the quality of the generated rule set. Our experiments show that fuzzy neural networks provides a valuable data mining tool for satellite image analysis and knowledge extraction.

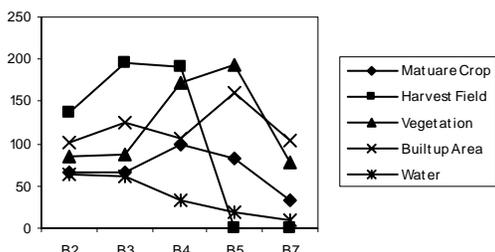


Figure 5. Spectral signatures for Chernobyl Scene

References

[1] G. G. Wilkinson, "Results and Implications of a study of fifteen years of satellite image classification experiments", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, 2005, pp 433-440.

[2] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, John Wiley, Hoboken, NJ, 2003.

[3] J. A. Richards, "Analysis of remotely sensed data: The formative decade and the future", *IEEE Transactions on*

Geoscience and Remote Sensing, vol. 43, no. 3, 2005, pp 422-432.

[4] J. C. Bezdek, "Editorial-fuzzy models-What are they and why", *IEEE Trans. on Fuzzy Systems*, vol. 1, 1993, pp. 1-5.

[5] C. T. Lin, and C. S. Lee George, "Neural network based fuzzy logic control and decision system", *IEEE Trans. on Computers*, vol. 40, 1991, pp. 1320-1336.

[6] J. S. R. Jang, "ANFIS Adaptive network based fuzzy inference systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, 1993, pp. 665-685.

[7] S. K. Pal, and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification", *IEEE Trans. on Neural Networks*, vol. 3, 1992, pp. 683-697.

[8] A. D. Kulkarni, *Computer Vision and Fuzzy-Neural Systems*, Prentice Hall PTR, Upper Saddle River, NJ, 2001.

[9] S. Mitra, R. K. De, and S. K Pal, "Knowledge-based fuzzy MLP for classification and rule generation", *IEEE Trans. on Neural Networks*, vol. 8, no. 6, 1997, pp. 1338-1350.

[10] L. Wang, and J. M. Mendel, "Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 6, 1992, pp. 1414-1427.

[11] A. D. Kulkarni, and S. McCaslin, S., "Knowledge discovery from multispectral satellite images", *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, 2004, pp 246-250.