Fuzzy Neural Network Models For Clustering

A. D. Kulkarni and V. K. Muniganti

Computer Science Department
The University of Texas at Tyler, Tyler, TX 75701

ABSTARCT

Conventional clustering techniques group input feature vectors in crisp classes. However, in practice the input feature vector may belong to more than one class. This is specially the case for pixels in a satellite image. Often, the area represented by a pixel may belong to more than one category, i. e. half the area may belong to water and remaining half may belong to land. In the situation such as this, we would like to use partial memberships. Fuzzy sets allow partial memberships. In this paper we have suggested an algorithm for clustering that combines neural networks with fuzzy logic. We have used the model successfully to analyze satellite images. The results are presented in the paper.

Introduction

Unsupervised classification techniques deal with learning without training samples. It is a common phenomenon that features belonging to the same class tend to form groups or clusters in the feature space. Many conventional clustering algorithms such as the K-means, isodata are available in practice (Jain and Dubes, 1988; Ball and Hall, 1966). These iterative algorithms use some similarity measure as the clustering criterion to optimize the objective function. The most commonly used similarity measure is the Euclidean distance in the feature space. Similarity measures such as the normalized correlation, Minkowsky metric, Mahalonbic metric, and Hamming distance are also used. In all of these algorithms, an input sample is assigned to a class on the basis that it is *closer* to patterns of that class. In classical or crisp clustering algorithms, a sample is assigned to one and only one cluster. Often in practice it is desirable to allow partial memberships so that a sample can be assigned to more than one class with a degree of belief that the sample belongs to each class. Application of fuzzy set theory to classical clustering algorithms has resulted in a number of algorithms.

Self organizing neural networks with learning paradigms like competitive learning, adaptive resonance theory, and Kohonen's self organizing maps are examples of unsupervised learning (Kulkarni, 1994). These algorithms are similar to classical crisp clustering algorithms. Recently, many models that combine neural networks and fuzzy logic techniques have been suggested. These include adaptive fuzzy c-shell clustering, fuzzy min-max neural networks, fuzzy-ARTMAP, and adaptive fuzzy leader clustering (Hall et al., 1992; Carpenter et al., 1992; Newton et al., 1992). Neural networks provide algorithms for learning, classification, and optimization where as fuzzy logic often deals with issues such as reasoning on a high (semantic or linguistic) level. Consequently, the two technologies complement each other (Bezdek, 1993). There are number of ways to synthesize fuzzy logic with neural networks. The simplest way is to use fuzzy membership functions to preprocess input data and/or to use partial membership information to post process data (Lin and Lee, 1991; Kulkarni, et al., 1995; Mitra and Pal, 1994). An alternative approach is to use a neural network model with fuzzy signals and fuzzy weights. In this paper we have used the first approach. Here, we have suggested an algorithm which is similar to the fuzzy adaptive leader clustering algorithm. We have used partial memberships to update weights in the neural network model. We can also use fuzzy membership functions to pre-process input data. As an illustration, the model is used to analyze multispectral satellite data.

ANN Model for Unsupervised Learning

A model for a fuzzy competitive learning is shown in Figure 1. The model consists of two layers: the input layer and output layer. To start with the number of clusters is set to zero. The first input vector is set as a prototype for the first cluster. With each input vector, the unit in the recognition layer with the highest weighted sum, is declared as the winner. If the input vector satisfies the distance criterion the input vector is assigned to that cluster and the weights are updated using partial membership values. If no unit satisfies the distance criterion, a new cluster with the input vector as the centroid is formed. The process is described in the steps shown below.

Let x_i represents the ith input vector, such that

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (1)$$

where n represents the number of features.

Step 1: To start with the number of clusters M is assumed to be zero. Let x_1 be the first input vector. Assume the first input vector as the first prototype for the first cluster and assign the weights accordingly, i. e. $v_1 = x_1$, where v_1 represents weights connecting units in layer L_1 to unit i in layer L_2, in Figure 1.

Step 2: Let x_j be the next input vector. Find the winner unit in layer L_2 using the minimum distance (or the maximum dot product) as the criterion.

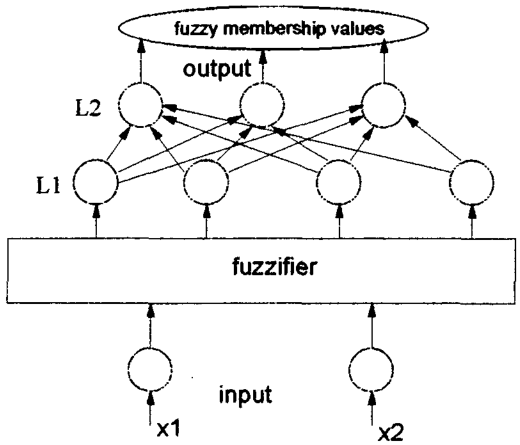

Figure 1. Neural network model for fuzzy competitive learning

$$d^2_{min} = \min_{i \in M} \| x_j - v_i \|^2 \qquad (2)$$

where

$$\| x_j - v_i \|^2 = \sum_{k=1}^{M} (x_{jk} - v_{ik})^2 \qquad (3)$$

and M represents the number of clusters.

Step 3: If the winner unit does not satisfy the distance criterion given in Equation (4), then create a new cluster and make its prototype vector be equal to x_j.

$$R_i = \| x_j - v_i \| / [\, 1/N_i \sum_{k=1}^{N_i} \| x_k - v_i \| \,] < \tau \qquad (4)$$

where $k=1,2,...N_i$ are number of samples in class i.

Step 4: Otherwise, update the winner cluster prototype associated with y_i by calculating the new centroid and membership values using Equations (5) and (6).

$$\mu_{ij} = \{ [\ 1/\ \| xj - vi \|^2\ ]^{1/m-1} \} / \{ \sum_{k=1}^{M} [\, 1/ \| x_j - v_k \|^2 \,]^{1/m-1} \} \qquad (5)$$

where $1 \le i \le M$, and $1 \le j \le N$.

$$v_i = [\ 1 / \sum_{j=1}^{N_i} \mu_{ij} \,]\ [\ \sum_{j=1}^{N_i} (\mu_{ij})^{\mu} x_j] \qquad (6)$$

where $1 \le i \le M$.

Step 5. If no more input samples then stop, else go to Step 2.

We can also use a fuzzifier at the input side. A fuzzifier block shown in Figure 1, can be implemented using two layers of neural units. A model, with the blown upfuzzifier block , is shown in Figure 2. The layers in the model are described below.

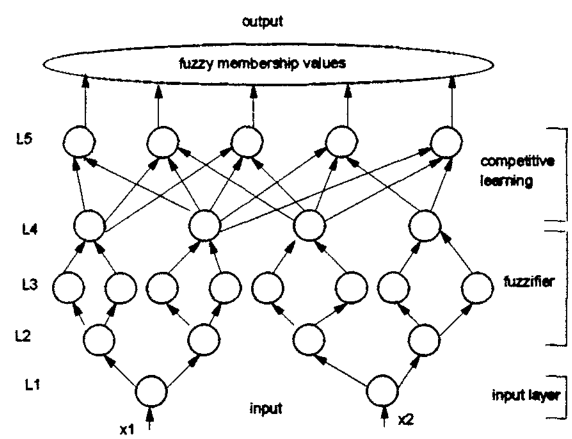

Figure 2. Neural network model for fuzzy competitive learning with fuzzified inputs

Layer L_1. The number of units in this layer is equal to the number of input features. Units in this layer correspond to the input features and they just transmit the input vector to the next layer. The net-input and the activation function for this layer is given by

$$net_i = x_i$$
$$out_i = net_i \qquad (7)$$

where net_i indicates the net-input for unit i and out_i represents the output of unit i.

Layers L_2, L_3, and L_4. These layers implement the membership function. Units in layer L_2 represent the linguistic term variables. In this model we have used five term variables {*very-low, low, medium, high, very-high*} for each input feature value. Hence, the number of units in layer L_2 is five times the number of units in layer L_1. The net-input and activation functions for units in these layers are chosen so as to implement the triangular or π-shaped membership functions. The net-input and output for units in layer L_2 for the triangular membership functions, are given by

$$net_i = x_i / \rho_i$$
$$out_i = net_i \tag{8}$$

The net-input and output for units in layer L_2 for π-shaped

$$neti = x_i$$
$$outi = net_i \tag{9}$$

Each of the units in layer L_2 is connected to two units in layer L_3. Each pair of units in layer L_3 represents the left and right sides of a triangular shaped membership function. The weights connecting these units are +1 and -1. The net-input and output for each of the two units in layer L_3 are given by

$$\begin{aligned} net1_i &= (m_i - x_i) / r_i \\ out1_i &= 1 - net1_i \quad \text{for } m_i - r_i \leq x_i \leq m_i \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{10}$$

where $net1_i$ and $out1_i$ represent the net-input and the output of the unit that corresponds to the left side of the triangular membership function, and i represents the ith input feature. Similarly the net-input and output of the unit that corresponds to the right side of a triangular membership function are given by

$$\begin{aligned} net2_i &= (x_i - m_i) / r_i \\ out2_i &= 1 - net2_i \quad \text{for } m_i < x_i \leq m_i + r_i \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{11}$$

In the case of p-shaped functions Equations (10) and (11) are replaced by Equations (12) and (13), respectively. The π-shaped functions are commonly used membership functions (Cox, 1994).

$$\begin{aligned} net1_i &= x_i \\ out1_i &= S(net1_i,\ m_i - b_i,\ m_i - b_i/2,\ m_i) \\ &\qquad \text{for } m_i - b_i \leq net_i < m_i \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{12}$$

where $net1_i$ and $out1_i$ represent the net-input and the output of the unit that corresponds to the left half of the p-shaped membership function., and i represents the ith input feature. Similarly the net-input and output of the unit that corresponds to the right half of the p-shaped membership function is given by

$$\begin{aligned} net2_i &= x_i \\ out2_i &= 1 - S(net2_i,\ m_i,\ m_i + b_i/2;\ m_i + b) \\ &\qquad \text{for } m_i < net2_i \leq m_i + b \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{13}$$

Equations (12) and (13) represent two sides of the π-curve. Each unit in layer L_4 combines the outputs of the corresponding two units in L_3. The outputs of units in layer L_4 represent the membership values.

Layers L_4 and L_5: These layers implement the competitive learning algorithm described in steps 1 through 6. Units in Layer 6 represent output classes or clusters of samples.

Computer Simulation

We have developed software to simulate the neural network model with the learning algorithm described above. As an illustration, we have used the model for multispectral satellite image analysis. The application deals with recognition of pixels in a satellite image. In remote sensing measured signals expressed as the function of a wave length are often referred to as the "spectral signature" of the object on which measurements are made. In principle spectral signatures are unique. Therefore it is possible to identify an object from its spectral signature. We have used data from Thematic Mapper (TM) sensor (scene # y4018116055) obtained on January 1983. The scene represents Mississippi river bottom land and is of the size 512 scans x 152 pixels. With TM, images are obtained in seven spectral bands. Multispecral image acquisition is depicted in Figure 3. The scene was analyzed using a simple competitive learning algorithm as well as the competitive learning with fuzzy partial memberships algorithm. The output images are shown in Figures 4 and 5, respectively. In our analysis we used only four spectral bands, as these bands showed maximum variance and contained most of the information. The results obtained with both the algorithms are summarized in Table I and II, respectively.

Discussions and Conclusions

In this paper, we have proposed a neural fuzzy system which uses the learning algorithm similar to the adaptive fuzzy leader clustering algorithm (Newton, et al., 1992). The architecture learns and adapts online, such that it is not necessary to have a priori knowledge of all data samples or the number of clusters present in the data. However, the choice of threshold τ is critical and requires some a priori knowledge of separation of clusters in the feature space. Learning here is match based ensuring stable and consistent learning. The output is a crisp classification. The simple competitive learning algorithm assigns input samples to one and only one cluster. However, in practice samples are not always pure, i. e. a sample may belong to more than one class. This is often true with pixels with high spatial resolution. The model with fuzzy partial memberships is more appropriate for such data. It can seen from the output images that outputs obtained with the fuzzy competitive learning algorithm are more homogeneous than the corresponding output images with the simple competitive learning algorithm. It is also possible to use fuzzified inputs. The fuzzification process is a nonlinear mapping which results

Threshold	Number of Clusters	Number of Samples	Prototypes of Clusters f1	f2	f3	f4
0.250	9	158756	0.1878	0.2148	0.2148	0.4066
		48114	0.5974	0.3066	0.0649	0.0309
		18655	0.3159	0.2574	0.2937	0.1328
		6313	0.4807	0.2749	0.1694	0.0748
		29291	0.1885	0.2879	0.3709	0.1525
		168	0.2216	0.1892	0.4717	0.1173
		8	0.1808	0.1380	0.0629	0.6181
		4	0.3771	0.2369	0.0481	0.3378
		835	0.3899	0.2905	0.2174	0.1021
0.345	4	185314	0.2010	0.2314	0.3888	0.1785
		56076	0.5788	0.3022	0.0811	0.0378
		20744	0.1766	0.2057	0.4217	0.1957
		10	0.2183	0.1493	0.0614	0.5707

Table I. Output with a simple competitive learning algorithm

Threshold	Number of Clusters	Number of samples	Prototypes of Clusters			
5.80	9	14661	0.1631	0.3224	0.3703	0.1441
		53046	0.2004	0.2418	0.3851	0.1726
		112705	0.1796	0.2043	0.4173	0.1986
		20114	0.2754	0.2512	0.3261	0.1472
		50212	0.5987	0.3062	0.0642	0.0306
		11086	0.4015	0.2699	0.2267	0.1016
		267	0.2581	0.1884	0.4539	0.0994
		9	0.1699	0.1319	0.0684	0.6297
		44	0.2471	0.2665	0.2383	0.2478
7.38	3	206165	0.1944	0.2270	0.3961	0.1823
		55969	0.5876	0.3038	0.0737	0.0347
		10	0.1911	0.0140	0.0631	0.6047

Table II. Output with competitive learning with fuzzy memberships

Figure 4a. Classified Output, Competitive Learning, $\tau = 0.250$

Figure 4b. Classified Output, Competitive Learning, $\tau = 0.345$

Figure 5a. Clssified output clustering with fuzzy memberships, $\tau = 5.80$

Figure 5b. Classified output, clustering with fuzzy memberships, $\tau = 7.38$

in increased dimensions; however, it also increases separability of clusters in the feature space. Also with fuzzified inputs it is possible to interpret decision rules in terms of linguistic variables. We are in the process of investigating the effect of fuzzification of input variables on separability of clusters in the feature space. We are also investigating the possibility of extrcating ligustic rules for classification.

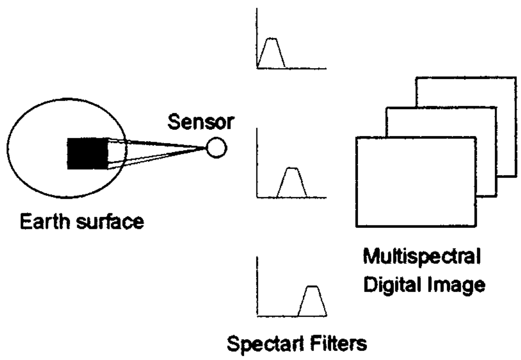

Figure 3. Multispectral Image

References

Ball, G. H., and Hall, D. J. (1966). ISODATA: An iterative method of multivariate data analysis and pattern classification. *Proceedings of International Communication Conference*, Philadelphia, PA.

Bezdek, J. C. (1993). Fuzzy models- what are they and why? *IEEE Transactions on Fuzzy Systems*, vol 1, pp 1-5.

Carpenter, G. A., and et al. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, vol 3, pp 698-713.

Cox, E. (1994). *The fuzzy systems handbbok.* Academic Press, Cambridge, MA.

Hall, L. O. and et al. (1992). A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Transactions on Neural Networks*, vol 3, pp 672-682.

Jain, A. K., and Dubes, R. C. (1988). *Algorithms for clustering data.* Prentice-Hall Englewoood Cliffs, NJ.

Kulkarni, A. D. (1994). *Artificial neural networks for image understanding.* Van Nostrand Reinhold, New York.

Kulkarni, A. D., Giridhar, G. B., and Coca, P. (1995). Neural network based fuzzy logic decision systems for multispectral image analysis. *Neural Parallel & Scientific Computations*, vol 3, pp 205-218.

Lin, C. T., Lee George C. S. (1991). Neural network based fuzzy logic control and decision systems. *IEEE Transactions on Computers*, vol 40, pp 1320-1336.

Mitra, S., and Pal, S. K. (1994). Logical operation based fuzzy MLP for classification and rule generation. *Neural Networks*, vol 7, pp 353-374.

Newton, S. C., Pemmaraju, S., and Mitra, S. (1992). Adaptive fuzzy leader clustering of complex data sets in pattern recognition. *IEEE Transactions on Neural Networks*, vol 3, pp 794-800.