Incorporation of Fuzzy Classification Properties into Backpropagation Learning Algorithm

Manish Sarkar and B. Yegnanarayana
Department of Computer Science & Engineering
Indian Institute of Technology, Madras - 600 036, INDIA
{manish@bronto.iitm, yegna@iitm}.ernet.in

Abstract

Most of the real life classification problems have ill defined, imprecise or fuzzy class boundaries. Feedforward neural networks with conventional backpropagation learning algorithm are not tailored to these kinds of classification problems. Hence, in this paper, feedforward neural networks, that use fuzzy objective functions in the backpropagation learning algorithm, are investigated. A learning algorithm is proposed that minimizes an error term, which takes care of fuzziness in classification from the point of view of possibilistic approach. Since the proposed algorithm has possibilistic classification ability, it can encompass different backpropagation learning algorithms based on crisp and constrained fuzzy classification. The efficacy of the proposed scheme is demonstrated on a vowel classification problem.

1. Introduction

Nowadays feedforward neural networks (FFNN) based on backpropagation (BP) learning algorithm [2] are used extensively for classification. However, a major drawback of BP algorithm is that it assigns each input pattern exactly to one of the output classes, assuming well-defined class boundaries. In real life situations, however, boundaries between classes may be overlapping. There can be some data points which do not completely belong to a single class, but partially belong to other classes too. This limits the applicability of BP algorithm on real life problems. In order to alleviate this drawback, fuzzy set based classification approach, inside the basic framework of BP algorithm has been recently investigated. Several interesting feedforward neuro-fuzzy systems have been proposed [6], [7], and they cover a wide range of applications.

This research work proposes a method of embedding fuzzy classification properties into conventional BP learning algorithm of feedforward neural networks (FFNN). Input is assumed to be crisp for these FFNNs, only the classification is fuzzy. Since the classification is fuzzy, an input pattern may not necessarily belong to one class; rather it may belong to more than one class with different degrees of belongingness. Consequently, unlike the conventional BP, the number of target classes corresponding to each input training pattern may be more than one. The aim of the proposed learning algorithm during training is to minimize an error term, henceforth termed as *fuzzy mean square error*. The fuzzy mean square error is defined as the overall weighted sum of square error between the actual network output and all possible target outputs, where the weight signifies the level of belongingness of the pattern into the corresponding target class. If a new input is presented to the neural network after training, it yields the output as class membership values of the input pattern. We also propose another formulation of the learning algorithm by considering a network that tries to minimize an alternative error term, called *fuzzy cross entropy*, which is a fuzzy counterpart of crisp cross entropy [2]. Although the learning algorithm for fuzzy mean square error and fuzzy cross entropy differ, the basic philosophy of introducing the concept of fuzzy classification into the crisp error measure is same.

The proposed learning algorithm is derived such that the sum total of membership values of a particular pattern to all the classes need not necessarily be equal to one. This implies that the membership assignment is not *constrained fuzzy* [4]; on the other hand, it is *possibilistic* [4]. This kind of property is desirable to signify the ignorance or different levels of evidence, which are well discussed in belief theory [9] and possibility theory [3]. In case of constrained fuzzy membership assignment, i.e., when sum total of membership values of an input pattern to all the classes is one, we show that the attractive learning algorithm, given by Pal and Mitra [6], is equivalent to the proposed algorithm. In addition to it, when the classification is crisp, the proposed learning algorithm boils down to the conventional BP algorithm. Thus, it turns out that the possibilistic approach of the proposed algorithm leads it to encompass both constrained fuzzy classification and crisp classification. Another aspect of the proposed learning algorithm is that it has the scope for controlling the amount of fuzziness that is involved in the classification process.

2. Background of Fuzzy Classification

A C-class classification problem for a set of input data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P\}$ is basically an assignment of membership values $\mu_c(\mathbf{x}_p)$ on each $\mathbf{x}_p \in \mathbf{X}$, $\forall c =$

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.

$1, 2, \ldots, C,\ \forall p = 1, 2, \ldots, P$. If the membership values are crisp, then $\mathbf{X}$ is partitioned into C subgroups during the classification process. In fuzzy context, C partitions of $\mathbf{X}$ are set of values $\{\mu_c(\mathbf{x}_p)\}$, that can be conveniently arranged on a $C \times P$ matrix $\mathbf{U} = [\mu_c(\mathbf{x}_p)]$. Based on the characteristic of $\mathbf{U}$, classification can be of three types as follows [4]:

1. *Crisp classification*:

$$\mathbf{M}_{hc} = \Big\{\mathbf{U} \in \Re^{CN} \mid \mu_c(\mathbf{x}_p) \in \{0,\ 1\}\ \forall c, \forall p; \sum_{c=1}^{C} \mu_c(\mathbf{x}_p) = 1;\ 0 < \sum_{p=1}^{P} \mu_c(\mathbf{x}_p) < P\ \forall c\Big\} \tag{1-a}$$

2. *Constrained Fuzzy Classification*:

$$\mathbf{M}_{fc} = \Big\{\mathbf{U} \in \Re^{CN} \mid \mu_c(\mathbf{x}_p) \in [0,\ 1]\ \forall c, \forall p; \sum_{c=1}^{C} \mu_c(\mathbf{x}_p) = 1;\ 0 < \sum_{p=1}^{P} \mu_c(\mathbf{x}_p) < P\ \forall c\Big\} \tag{1-b}$$

3. *Possibilistic Classification*:

$$\mathbf{M}_{pc} = \Big\{\mathbf{U} \in \Re^{CN} \mid \mu_c(\mathbf{x}_p) \in [0,\ 1]\ \forall c, \forall p; 0 < \sum_{p=1}^{P} \mu_c(\mathbf{x}_p) < P\ \forall c\Big\} \tag{1-c}$$

From the relations (1-a), (1-b) and (1-c), it is obvious that $M_{hc} \subset M_{fc} \subset M_{pc}$. We will see later that our proposed learning algorithm is based on possibilistic classification, and hence as a natural consequence, various BP algorithms based on constrained fuzzy and crisp classification become particular cases of the proposed algorithm.

Next part of the discussion describes how to determine the membership value of each pattern. The membership of the pth pattern to class c is defined as [5]

$$\mu_c(\mathbf{x}_p) = \frac{1}{1 + (\frac{z_{pc}}{F_d})^{F_e}} \tag{2}$$

where z_{pc} is the weighted distance and, the positive constants F_d and F_e are the denominational and exponentional fuzzy generators controlling the amount of fuzziness in this class-membership set. The weighted distance is discussed in detail later. Obviously, $\mu_c(\mathbf{x}_p)$ lies in the interval [0, 1]. Specifically, higher the distance of a pattern from a class, the lower is its membership value to that class. In particular when the distance is zero, membership value is one (maximum) and, on the other hand, when the distance is infinite, membership value is zero (minimum). The method of calculating weighted distance is as follows:

Let the N-dimensional vectors $\mathbf{m}_c$ and σ_c denote the mean and standard deviation, respectively, of the set of training data for the cth class. The weighted distance of a training pattern $\mathbf{x}_p = [x_{p1}, x_{p2}, \ldots, x_{pN}]^T$ from the cth class is defined as [5]

$$z_{pc} = \sqrt{\sum_{i=1}^{N} \left[\frac{x_{pi} - m_{ci}}{\sigma_{ci}}\right]^2}\ \forall c = 1, \ldots, C \tag{3}$$

The weight $\frac{1}{\sigma_{ci}}$ is used to take care of the variance of the classes so that a feature with higher variance has less weight (significance) in characterizing a class. Note that here $\sum_c \mu_c(\mathbf{x}_p)$ need not be equal to one.

3. Proposed Algorithm

Let, the training set in a C class problem consists of vector pairs $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_P, \mathbf{y}_P)\}$, where $\mathbf{x}_p \in \Re^N$ refers to the pth input pattern and $\mathbf{y}_p \in \{\mathbf{t}_c \mid c = 1, 2, \ldots, C;\ \mathbf{t}_c \in \Re^C\}$ refers to the target output of the network corresponding to this input. Specifically, if $\mathbf{x}_p$ is from the kth class, then $\mathbf{y}_p = \mathbf{t}_k$, where $t_{kk} = 1$ and $t_{ck} = 0\ \forall c,\ c \neq k$. The network used here is a multilayer feedforward network which can have several hidden layers. Without loss of generality, number of the hidden layers can be assumed to be one with H hidden nodes. When an input pattern $\mathbf{x}_p$ is applied at the input layer of the network, the output of the jth hidden unit is $o^h_{pj} = f^h_j(\text{net}^h_{pj}) = \frac{1}{1+\exp(-\text{net}^h_{pj})}$ where $\text{net}^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j$. Here w^h_{ji} is the weight of the link from the ith node of the input layer to the jth node of the hidden layer, and θ^h_j is the bias term for the jth hidden node. Similarly, the equations for the kth output node is $o^o_{pk} = f^o_k(\text{net}^o_{pk}) = \frac{1}{1+\exp(-\text{net}^o_{pk})}$ where $\text{net}^o_{pk} = \sum_{j=1}^{H} w^o_{kj} f^h_j(\text{net}^h_{pj}) + \theta^o_k$. The h and o superscripts refer to quantities in the hidden and output layer, respectively.

3.1. Training Algorithm

The adaptive parameters of the FFNN consist of all the weights and bias terms. The sole purpose of the training phase is to determine the optimum setting of the weights and bias terms so as to minimize the difference between the network output and the target output. This difference is called the training error of the network. The error measure can be *fuzzy mean square error* which is basically a fuzzy counterpart of the mean square error [2] used in the conventional BP algorithm.

In the conventional BP algorithm, the mean square error for the pth input pattern is defined as $E_p = \frac{1}{2}\sum_{k=1}^{C}(t_{pk} - o^o_{pk})^2$. However, use of E_p as an error term

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.

is justified when each input pattern belongs to only one class. But in fuzzy classification, the input pattern may belong to more than one class with different degrees of belongingness. It implies that the target value of an input pattern may be more than one. In other words, each input can have all possible target values with different membership values (certain membership values may also be zero). Through training the network attempts to reach those target values weighted by different membership values. In other words, the problem of training can also be conceptually viewed as a fuzzy constraint satisfaction problem. Here, the constraint is that each input pattern should belong to a particular class, and the associated membership value signifies upto what extent this constraint should be satisfied. In the training phase, the task of the proposed network is to adapt the parameters such that these constraints are resolved optimally. Mathematically, for the pth input pattern the constraints can be expressed as the fuzzy mean square error term, which is defined as

$$E_p^f = \frac{1}{2}\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck} - o_{pk}^o)^2 \tag{4}$$

Here, the index of μ, i.e., $q \in [0, \infty)$ controls the amount of fuzziness present into the classification. Different values of q signifies upto what extent, the constraints should be satisfied. When $q = 0$, each input pattern tries to attain all the target outputs with equal importance, and ultimately the network learns the mean of all the class centers. When the value of q is greater then one, the constraint associated with high membership value gets more importance to be resolved. When q tends to be infinity, only the input pattern that belongs to a class completely, i.e. with membership one, is learned. In fact, it can be proved that E_p^f decreases strictly as q increases in $[1, \infty]$ for $0 < \mu_c(\mathrm{x}_p) < 1\ \forall c$. As a result, for $0 < \mu_c(\mathrm{x}_p) < 1\ \forall c$, the minimization of E_p^f becomes trivial at $q = \infty$ as E_p^f attains its minimum value which is zero. On the other hand, when q is less than one, the constraint associated with high membership value gets less importance to be resolved. Thus, q controls the extent of membership sharing among the fuzzy classes. This can be good; on the other hand, one must choose q to actually implement it. In the current work q is assumed to be one. The role of q here is quite similar to the index of fuzziness in concentration and dilation operators found in *fuzzy hedge* [3], and index of fuzziness in Fuzzy C-Means clustering algorithm [1].

Next we derive learning law for the network following the same method as followed in the conventional BP algorithm [2]. Here we assume that the weight updation, Δw, takes place after the presentation of each input pattern. Assuming the use of same learning-rate parameter η for all the weight changes made in the network, the weight changes applied to the weights w_{kj} and w_{ji} are calculated, respectively, in accordance to the gradient-descent rules: $\Delta w_{kj}^o = -\eta\frac{\partial E_p^f}{\partial w_{kj}^o}$ and $\Delta w_{ji}^h = -\eta\frac{\partial E_p^f}{\partial w_{ji}^h}$.

From appendix A we can write,

$$\begin{aligned}\Delta w_{kj}^o &= \eta\left[\mu_k^q(\mathrm{x}_p) - \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right] \\ &\quad o_{pk}^o(1 - o_{pk}^o)o_{pj}^h &(5)\\ &= \eta\delta_{pk}^o o_{pj}^h &(6)\end{aligned}$$

where $\delta_{pk}^o = \left[\mu_k^q(\mathrm{x}_p) - \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right]o_{pk}^o(1 - o_{pk}^o)$.

Again, from appendix B,

$$\begin{aligned}\Delta w_{ji}^h &= -\eta \acute{f}_j^h(\mathrm{net}_{pj}^h)x_{pi}\sum_{k=1}^{C}\Big[\mu_k^q(\mathrm{x}_p) - \\ &\quad \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\Big]o_{pk}^o(1 - o_{pk}^o)w_{kj}^o &(7)\\ &= \eta \acute{f}_j^h(\mathrm{net}_{pj}^h)x_{pi}\sum_{k=1}^{C}\delta_{pk}^o w_{kj}^o &(8)\\ &= \eta\delta_{pj}^h x_{pi} &(9)\end{aligned}$$

where $\delta_{pj}^h = \acute{f}_j^h(\mathrm{net}_{pj}^h)\sum_{k=1}^{C}\delta_{pk}^o w_{kj}^o$.

Now, we generalize other error measure, i.e., cross entropy for the pth input pattern, which is defined as follows:

$$\mathcal{H}_p = \sum_{k=1}^{C}\left(t_{pk}\ln\left(\frac{t_{pk}}{o_{pk}^o}\right) + (1 - t_{pk})\ln\left(\frac{1 - t_{pk}}{1 - o_{pk}^o}\right)\right) \tag{10}$$

Since t_{pk} is either zero or one, we can write the above definition as

$$\mathcal{H}_p = -\sum_{k=1}^{C}\left(t_{pk}\ln(o_{pk}^o) + (1 - t_{pk})\ln(1 - o_{pk}^o)\right) \tag{11}$$

Following the same logic, as we used to justify the use of fuzzy mean square error in place of mean square error, we can generalize $\mathcal{H}_p$ to its fuzzy counterpart, called fuzzy cross entropy, which is defined as

$$\begin{aligned}\mathcal{H}_p^f &= -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)\Bigg[\sum_{k=1}^{C}\big(t_{ck}\ln(o_{pk}^o) + (1 - t_{ck}) \\ &\quad \ln(1 - o_{pk}^o)\big)\Bigg] &(12)\end{aligned}$$

It can be proved that $\mathcal{H}_p^f$ decreases strictly to zero as q increases in $[1, \infty]$ for $0 < \mu_c(\mathrm{x}_p) < 1\ \forall c$. Here q controls the amount of fuzziness in a similar way as it does in (4).

Following the same method as we used in case of fuzzy mean square error, here we obtain the following

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.

learning equations (see appendix C and D).

$$\Delta w^o_{kj} = \eta \frac{\partial \mathcal{H}^f_p}{\partial w^o_{kj}} = \eta \left[\mu^q_k(\mathrm{x}_p) - \sum_{c=1}^{C} \mu^q_c(\mathrm{x}_p) o^o_{pk} \right] o^h_{pj} \tag{13}$$

$$\Delta w^h_{ji} = \eta \frac{\partial \mathcal{H}^f_p}{\partial w^h_{ji}} = \eta \acute{f}(\mathrm{net}^h_{pj}) x_{pi} \sum_{k=1}^{C} \left[\mu^q_k(\mathrm{x}_p) - \sum_{c=1}^{C} \mu^q_c(\mathrm{x}_p) o^o_{pk} \right] w^o_{kj} \tag{14}$$

Therefore, it turns out that, by introducing fuzzy concepts in the usual BP error measures, we can obtain a large class of learning algorithms. Although, the exact formulation of the learning equations for fuzzy mean square error and fuzzy cross entropy may be different, the underlying concept of introduction of fuzziness into the usual error measures is same.

Now we illustrate the following particular cases of the proposed learning algorithm.

1. *Crisp Classification*: In case of crisp classification only one component of $\mu^q_c(\mathrm{x}_p)\ \forall c = 1, \ldots, C$, is one and the remaining components are zero. Thus, the expression for E^f_p boils down to $E^f_p = -\frac{1}{2} \sum_{k=1}^{C} \sum_{c=1}^{C} (t_{cp} - o_{pk})^2$, which is the mean square error term found in the conventional BP algorithm. Consequently, in a crisp case the learning algorithm based on mean square error and fuzzy mean square error become identical. This can be easily verified by making membership assignments in (6) and (9) crisp.

   Similarly, in case of crisp classification fuzzy cross entropy boils down to the conventional cross entropy term, and consequently, learning equations for cross entropy and fuzzy cross entropy become same.

2. *Constrained Fuzzy Classification*: When $\sum_c \mu_c(\mathrm{x}_p) = 1\ \forall p$ and $q = 1$, the learning equations (6) and (9) achieve simpler forms as follows:

$$\Delta w^o_{kj} = \eta \delta^o_{pk} o^h_{pj} \tag{15}$$

$$\Delta w^h_{ji} = \eta \delta^h_{pj} x_{pi} \tag{16}$$

   where $\delta^o_{pk} = \left[\mu_k(\mathrm{x}_p) - o^o_{pk}\right] o^o_{pk}(1 - o^o_{pk})$ and $\delta^h_{pj} = \acute{f}^h_j(\mathrm{net}^h_{pj}) \sum_{k=1}^{C} \delta^o_{pk} w^o_{kj}$. In this particular situation, we can note down that this version of the proposed algorithm is equivalent to the learning algorithm proposed by Pal *et al.* in [6]. It is also important to note that we are not considering Pal *et al.*'s algorithm with fuzzy linguistic input; rather we are considering it with crisp input. In the future correspondence, the proposed algorithm will be extended to take care of fuzzy linguistic input.

   For $\sum_c \mu_c(\mathrm{x}_p) = 1\ \forall p$ and $q = 1$, the learning equations (13) and (14) based on fuzzy entropy can be simplified as

$$\Delta w^o_{kj} = \eta \left[\mu_k(\mathrm{x}_p) - o^o_{pk}\right] o^h_{pj} \tag{17}$$

$$\Delta w^h_{ji} = \eta \acute{f}(\mathrm{net}^h_{pj}) x_{pi} \sum_{k=1}^{C} \left[\mu_k(\mathrm{x}_p) - o^h_{pj}\right] w^o_{kj} \tag{18}$$

   This particular case of the learning algorithm is derivable from a variant of Pal *et al.*'s cross entropy [6], i.e., $\mathcal{H}^{\mathrm{pal}}_p = (\mu_k(\mathrm{x}_p) \ln(o^o_{pk}) + (1 - \mu_k(\mathrm{x}_p)) \ln(1 - o^o_{pk}))$. This result is quite obvious as the definition of fuzzy cross entropy boils down to Pal *et al.*'s cross entropy when $q = 1$ and $\sum_c \mu_c(\mathrm{x}_p) = 1\ \forall p$. This claim can be proved from appendix E.

Thus, being possibilistic in nature, the proposed algorithm encapsulates various BP algorithms based on crisp as well as constrained fuzzy classification.

3.2. Testing

The network learns the fuzzy boundaries between the different classes after training. In this stage, a separate set of test patterns is given as inputs to the network. Generated outputs are class memberships corresponding to the respective test inputs.

4. Results and Discussion

We consider the task of vowel recognition [8] to demonstrate the efficiency of the proposed scheme. For our study we consider the vowels 'a', 'e', 'i', 'o' and 'u'. The data required for training is collected from vowel part of utterances of consonant vowel pairs of three different speakers. First three formants are used as features. They are extracted from the utterances by taking the LPC [8] and finding the frequencies at which the spectrum reaches peaks. We use these extracted features to constitute a training set of 800 examples. Here our objective is to employ different learning techniques on this training set, and compare their classification performances on a different test set that consists of 1000 samples.

Initially we use Bayes [2] classifier for multivariate normal patterns with *a priori* probabilities $p_i = \frac{P_i}{P}$, where P_i denotes [5] the number of patterns in the ith class and P is the total number of training patterns. The covariance matrix for each class is determined from the training patterns of that particular class. Classification performance of the Bayes classifier on the test set is shown in the second column of Table 1.

Next we use different BP learning algorithms based on the variants of mean square error and cross entropy to train a feedforward neural network with 3 input nodes, 5 hidden nodes and 5 output nodes. The target

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.

Table 1. Results of vowel classification using formant data for different types of classification algorithm.

		Mean square error			Cross entropy		
Class	Bayes classifier	Conventional BP	Pal *et al.*'s algorithm	Proposed algorithm	Conventional BP	Pal *et al.*'s algorithm	Proposed algorithm
'a'	84.89%	80.36%	87.09%	90.27%	83.18%	88.86%	92.76%
'e'	82.87%	80.59%	85.66%	86.62%	82.86%	86.95%	85.92%
'i'	87.34%	82.31%	87.21%	89.41%	82.23%	87.78%	91.68%
'o'	80.29%	86.89%	83.75%	86.76%	86.76%	83.69%	86.18%
'u'	84.75%	87.91%	92.73%	93.89%	88.98%	91.73%	94.87%
overall	85.05%	83.61%	87.28%	89.39%	84.80%	88.01%	90.28%

patterns are 5 dimensional vectors containing 1 in one location and 0 in all others. Here, we adopt the strategy of picking the output node with highest activation value as the output class corresponding to an input. For all the three learning algorithms, convergence is achieved within 3000 iterations. In all the cases, learning rate is decreased over iterations as the training error becomes smaller and smaller.

Here we illustrate the results of different BP algorithms based on the variants of mean square error. Classification efficiency of the network trained with the conventional BP algorithm, Pal *et al.*'s algorithm and proposed algorithm are demonstrated in the third, fourth and fifth columns of the Table 1, respectively. The values of F_e, F_d and q are chosen as 2, 5 and 1, respectively. In Table 1, we can observe better classification performance of the proposed method compared to the other methods on the same test set.

Now we illustrate the classification efficiency of the network with different BP learning algorithms based on the variants of cross entropy. Sixth, seventh and eighth columns of Table 1 shows the classification efficiency of the conventional BP algorithm, Pal *et al.*'s algorithm and the proposed algorithm on the test set. Here also the proposed method performs better than the other methods. This improvement takes place because of considering the fuzziness involved in classification from possibilistic angle.

Acknowledgement

This work was carried out while Manish Sarkar held a Dr. K. S. Krishnan Research Fellowship from Department of Atomic Energy, Government of India.

Appendix

A. The expression for $\frac{\partial E_p^f}{\partial w_{kj}^o}$ can be derived as,

$$\frac{\partial E_p^f}{\partial w_{kj}^o} = -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)\frac{\partial f_k^o(\mathrm{net}_{pk}^o)}{\partial(\mathrm{net}_{pk}^o)}\frac{\partial(\mathrm{net}_{pk}^o)}{\partial w_{kj}^o} \qquad \text{(A-1)}$$

$$= -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)o_{pk}^o(1-o_{pk}^o)o_{pj}^h \qquad \text{(A-2)}$$

$$\text{So, } \frac{\partial E_p^f}{\partial w_{kj}^o} = -\left[\mu_k^q(\mathrm{x}_p)(t_{kk}-o_{pk}^o)+\sum_{c=1_{c\neq k}}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)\right]o_{pk}^o(1-o_{pk}^o)o_{pj}^h \qquad \text{(A-3)}$$

Since $t_{kk}=1$, and $t_{ck}=0\ \forall c\neq k$,

$$\frac{\partial E_p^f}{\partial w_{kj}^o} = -\left[\mu_k^q(\mathrm{x}_p)(1-o_{pk}^o)-\sum_{c=1_{c\neq k}}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right]o_{pk}^o(1-o_{pk}^o)o_{pj}^h \qquad \text{(A-4)}$$

$$= -\left[\mu_k^q(\mathrm{x}_p)-\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right]o_{pk}^o(1-o_{pk}^o)o_{pj}^h \qquad \text{(A-5)}$$

B. The expression for $\frac{\partial E_p^f}{\partial w_{kj}^h}$ can be found as follows:

$$\frac{\partial E_p^f}{\partial w_{ji}^h} = -\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)\frac{\partial f_k^o(\mathrm{net}_{pk}^o)}{\partial(\mathrm{net}_{pk}^o)}\frac{\partial(\mathrm{net}_{pk}^o)}{\partial o_{pj}^o}\frac{\partial o_{pj}^o}{\partial(\mathrm{net}_{pj}^h)}\frac{\partial(\mathrm{net}_{pj}^h)}{\partial w_{ji}^h} \qquad \text{(B-1)}$$

$$= -\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)o_{pk}^o(1-o_{pk}^o)w_{kj}^o f_j'^h(\mathrm{net}_{pj}^h)x_{pi} \qquad \text{(B-2)}$$

$$= -f_j'^h(\mathrm{net}_{pj}^h)x_{pi}\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)o_{pk}^o(1-o_{pk}^o)w_{kj}^o \qquad \text{(B-3)}$$

Following the steps involved while deriving (A-5) from (A-2), we can write

$$\mu_k^q(\mathrm{x}_p)-\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o \equiv \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)o_{pk}^o(1-o_{pk}^o) \qquad \text{(B-4)}$$

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.

Hence,

$$\frac{\partial E_p^f}{\partial w_{ji}^h} = -f_j^h(\text{net}_{pj}^h)x_{pi}\sum_{k=1}^{C}\left[\mu_k^q(\mathrm{x}_p) - \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right. \\ \left. o_{pk}^o(1-o_{pk}^o)w_{kj}^o \right] \quad \text{(B-5)}$$

C. The find the value of $\frac{\partial \mathcal{H}_p^f}{\partial w_{kj}^o}$, we differentiate (12) with respect to w_{kj}^o as follows:

$$\frac{\partial \mathcal{H}_p^f}{\partial w_{kj}^o} = -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)\left(\frac{t_{ck}}{o_{pk}^o} - \frac{1-t_{ck}}{1-o_{pk}^o}\right)\acute{f}(\text{net}_{pk}^o)o_{pj}^h \quad \text{(C-1)}$$

$$= -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)\frac{(t_{ck}-o_{pk}^o)}{o_{pk}^o(1-o_{pk}^o)}o_{pk}^o(1-o_{pk}^o)o_{pj}^h \quad \text{(C-2)}$$

$$= -\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)o_{pj}^h \quad \text{(C-3)}$$

Hence, using identity (B-4),

$$\frac{\partial \mathcal{H}_p^f}{\partial w_{kj}^o} = -\left[\mu_k^q(\mathrm{x}_p) - \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right]o_{pj}^h \quad \text{(C-4)}$$

D. The value of $\frac{\partial \mathcal{H}_p^f}{\partial w_{ij}^h}$ can be calculated as,

$$\frac{\partial \mathcal{H}_p^f}{\partial w_{ij}^h} = -\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)\left(\frac{t_{ck}}{o_{pk}^o} - \frac{1-t_{ck}}{1-o_{pk}^o}\right) \\ \acute{f}(\text{net}_{pk}^o)w_{kj}^o\acute{f}(\text{net}_{pj}^h)x_{pi} \quad \text{(D-1)}$$

$$= -\sum_{c=1}^{C}\sum_{k=1}^{C}\mu_c^q(\mathrm{x}_p)(t_{ck}-o_{pk}^o)w_{kj}^o\acute{f}(\text{net}_{pj}^h)x_{pi} \quad \text{(D-2)}$$

Applying identity (B-4),

$$\frac{\partial \mathcal{H}_p^f}{\partial w_{kj}^o} = -\acute{f}(\text{net}_{pj}^h)x_{pi}\sum_{k=1}^{C}\left[\mu_k^q(\mathrm{x}_p) - \sum_{c=1}^{C}\mu_c^q(\mathrm{x}_p)o_{pk}^o\right]w_{kj}^o \quad \text{(D-3)}$$

E. When $q = 1$ and $k = 1$, the equivalence of $\mathcal{H}_p^f$ and Pal *et al.*'s entropy can be established by the following steps.

$$\mathcal{H}_p^f = -\sum_{c=1}^{C}\mu_c(\mathrm{x}_p)\left[\sum_{k=1}^{C}\left(t_{ck}\ln(o_{pk}^o) + (1-t_{ck})\right.\right. \\ \left.\left. \ln(1-o_{pk}^o)\right)\right] \quad \text{(E-1)}$$

$$= -\sum_{k=1}^{C}\left[\sum_{c=1}^{C}\mu_c(\mathrm{x}_p)\left(t_{ck}\ln(o_{pk}^o) + (1-t_{ck})\right.\right. \\ \left.\left. \ln(1-o_{pk}^o)\right)\right] \quad \text{(E-2)}$$

$$= -\sum_{k=1}^{C}\left[\mu_k(\mathrm{x}_p)\left(t_{kk}\ln(o_{pk}^o) + (1-t_{kk})\right.\right. \\ \left. \ln(1-o_{pk}^o)\right) + \sum_{c=1_{c\neq k}}^{C}\mu_c(\mathrm{x}_p)\left(t_{ck}\ln(o_{pk}^o)\right. \\ \left.\left. +(1-t_{ck})\ln(1-o_{pk}^o)\right)\right] \quad \text{(E-3)}$$

Since $t_{kk} = 1$ and $t_{ck} = 0 \; \forall c \neq k$,

$$\mathcal{H}_p^f = -\sum_{k=1}^{C}\left[\mu_k(\mathrm{x}_p)\ln(o_{pk}^o) + \sum_{c=1_{c\neq k}}^{C}\mu_c(\mathrm{x}_p)\ln(1-o_{pk}^o)\right] \quad \text{(E-4)}$$

In case of constrained fuzzy approach $\mu_k(\mathrm{x}_p) + \sum_{c=1_{c\neq k}}^{C}\mu_c(\mathrm{x}_p) = 1$, and hence,

$$\mathcal{H}_p^f = -\sum_{k=1}^{C}\left[\mu_k(\mathrm{x}_p)\ln(o_{pk}^o) + (1-\mu_k(\mathrm{x}_p))\ln(1-o_{pk}^o)\right] \quad \text{(E-5)}$$

$$= \text{Pal } \textit{et al.}\text{'s entropy}$$

References

[1] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[2] S. Haykin. *Neural Networks - A Comprehensive Foundation.* Macmillan College Publishing Company, New York, 1994.

[3] G. S. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty and Information.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[4] N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy C-means model. *IEEE Transactions on Fuzzy Systems*, 3(3):330–379, August 1995.

[5] S. K. Pal and D. Dutta Majumder. *Fuzzy Mathematical Approach to Pattern Recognition.* Wiley (Halsted Press), New York, 1986.

[6] S. K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets and classification. *IEEE Transactions on Neural Networks*, 3(5):683–697, September 1992.

[7] W. Pedrycz. Fuzzy neural networks with reference neurons as pattern classifiers. *IEEE Transactions on Neural Networks*, 3(5):770–775, September 1992.

[8] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliff, NJ, 1993.

[9] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, 1976.

Authorized licensed use limited to: INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on August 26, 2009 at 02:16 from IEEE Xplore. Restrictions apply.