World Applied Sciences Journal 4 (6): 772-780, 2008
ISSN 1818-4952

On the Application of Various Probabilistic Neural Networks in Solving Different Pattern Classification Problems

[1]Ibrahiem M.M. El Emary and [2]S. Ramakrishnan

[1]Department of Computer Engineering Faculty of Engineering,
Al Ahliyya Amman University, Amman, Jordan
[2]Department of Information Technology,
Dr. Mahalingam College of Engingeering and Technology, Udumalai Road, Pollachi-642003, India

Abstract: A Probabilistic Neural Network (PNN) is defined as an implementation of statistical algorithm called Kernel discriminate analysis in which the operations are organized into multilayered feed forward network with four layers: input layer, pattern layer, summation layer and output layer. A PNN is predominantly a classifier since it can map any input pattern to a number of classifications. Among the main advantages that discriminate PNN is: Fast training process, an inherently parallel structure, guaranteed to converge to an optimal classifier as the size of the representative training set increases and training samples can be added or removed without extensive retraining. Accordingly, a PNN learns more quickly than many neural networks model and have had success on a variety of applications. Based on these facts and advantages, PNN can be viewed as a supervised neural network that is capable of using it in system classification and pattern recognition. The main objective of this paper is to describe the possible use of various PNN in solving some problems arising in signal processing and pattern recognition. The main attention is devoted to application of PNN in various classification problems like: classification brain tissues in multiple sclerosis, classification image texture, classification of soil texture and EEG pattern classification. Experimental results have been carried out and it verify the ability of modified PNN in achieving good classification rate in compared with traditional PNN or back propagation neural network BPNN and KNN.

Key words: Probabilistic Neural Network (PNN) . Radial Basis Function (RBF) . Back Propagation Neural Network (BPNN) . Weighted Probabilistic Neural Network (WPPN)

INTRODUCTION

Probabilistic Neural Network PNN often learn more quickly than many neural network models such as back propagation networks and have had success on a variety of applications. PNNs are a special form of Radial Basis Function (RBF) network used for classification which is the major job of this paper. The network learns from a training set T which is a collection of examples called instances [1]. Each instance i has an input vector y_i and an output class denoted as $class_i$. During execution, the network receives additional input vectors denoted as x and outputs the class that x seems most likely to belong to.

With regards to the real time classification problem which is the main attention of this paper, PNN has proven to be more time efficient than conventional back propagation based networks. In order to classify a feature pattern [2] vector $x \in R^m$, that is to assign the pattern to one among k predefined classes, the conditional density $P(x|C_k)$ of each class C_k is estimated since it represents the uncertainty associated to class attribution; then these estimates are combined by the rule of Bayes to yield a posteriori class probabilities $P(C_k|x)$ that allow to make optimal decisions. In the PNN, conditional density estimation is accomplished by implementing the Parzen window technique [2]. One possible way of looking at this technique is to build a sphere of influence p(s, x) around each training (known) sample s and to add them up for each of the k classes

$$P(x|C_k) = \sum_{x \in C_k} p(s, x) \quad (1)$$

In original Specht's implementation, the basis function used as window is Gaussian Kernel is given by the following formula [2]:

Corresponding Author: Dr. S. Ramakrishnan, Department of Information Technology,
Dr. Mahalingam College of Engingeering and Technology, Udumalai Road, Pollachi-642003, India

$$p(s, x) = \exp(-\|x-s\|^2/2s^2) \quad (2)$$

Where the only free parameter is the width σ of the Gussians.

The neural implementation of this theory is quite direct and yields the PNNs. A PNN consists of a node in layer one for each of the N training samples. The weights leading from the input to a layer one node are the coordinates of the corresponding sample. The node computes the distance d(s, x) from the test vector x to the training sample s and outputs the value of the Gaussian according to Eq.2. The outcome of each of the layer one cell is added separately for the different classes according to Eq.1 by the connections to the output cells with weight one [2]. In this paper, we describe the possible use of PNN in solving some problems arising in signal processing. Here, the main attention is devoted to the application of PNN in various classification problems like: classification brain tissues in multiple sclerosis, classification image texture, classification of soil textures and classification of liver tumors, EEG pattern classification and other applications. The experimental results have been carried and it verify the ability of modified PNN in achieving good classification rate in compared to traditional PNN or back propagation Neural Networks BPNN.

Achieving the above objectives passes through six sections which organize this paper. In section two, architecture of the PNN and its operation concept is presented. Section three deals with the classification theory of PNN. Training of PNN is devoted in section four. Section five covers the major applications that use various types of PNN. The paper is ended in section six with conclusion and future works suggested to be done by others.

PNN ARCHITECTURE AND THEORY OF OPERATION

The probabilistic Neural Network used in this paper is shown in Fig. 1. The first (leftmost) layer contains one input node for each input attribute in an application. All connections in the network have a weight of 1, which means that the input vector is passed directly to each hidden node [1].

In PNN, there is one hidden node for each training instance i in the training set. Each hidden node h_i has a center point y_i associated with it, which is the input vector of instance i. A hidden node also has a spread factor, s_i, which determines the size of its respective field. There are a variety of ways to set this parameter. s_i is equal to the fraction f of the distance to the nearest neighbor of each instance i. The value of f begins at 0.5 and a binary search is performed to fine tune this value. At each of five steps, the value of f that results in the highest average confidence of classification is chosen [1]. A hidden node receives an input vector x and outputs an activation given by the Gaussian function g, which returns a value of 1 if x and y_i are equal and drops to an insignificant value as the distance grows [1]:

$$g(x, y_i, s_i) = \exp(-D^2(x, y_i)/2s_i^2) \quad (3)$$

The distance function D determines how far apart the two vectors are. By far the most common distance function used in PNNs is Euclidean distance. However, in order to appropriately handle applications that have both linear and nominal attributes, a heterogeneous distance function HVDM [3,4] is used to normalize Euclidean distance for linear attributes and the Value Difference Metric (VDM) [5] for nominal attributes. It is defined as:

$$HVDM(x, y) = \sqrt{\sum_{a=1}^{m} d^2_a(x_a, y_a)} \quad (4)$$

Where m is the number of attributes. The function $d_a(x, y)$ returns a distance between the two values x and y for attribute a and is defined as:

$$d_a(x, y) = \begin{cases} 1 & \text{if x or y is unknown} \\ vdm_a & \text{if a is nominal} \\ diff_a & \text{if a is linear} \end{cases} \quad (5)$$

The function $d_a(x, y)$ uses the following function, based on the Value Difference Metric (VDM) [5]for nominal (discrete, unordered) attributes:

$$Vdm_a(x, y) = \sqrt{\sum_{c=1}^{C} |(N_{a,x,c}/N_{a,x}) - (N_{a,y,c}/N_{a,y})|^2} \quad (6)$$

Where $N_{a,x}$ is the number of times attribute a had value x; $N_{a,x,c}$ is the number of times attribute a had value x and the output class was c and C is the number of output classes.

For linear attributes, the following function is used [5]:

$$Diff_a(x, y) = |x-y|/4s_a \quad (7)$$

Where s_a is the sample standard deviation of the value occurring for attribute a in the training set. Each hidden node h_i in the network is connected to a single class node. If the output class of instance i is j, then h_i is connected to class node C_j. Each class node C_j

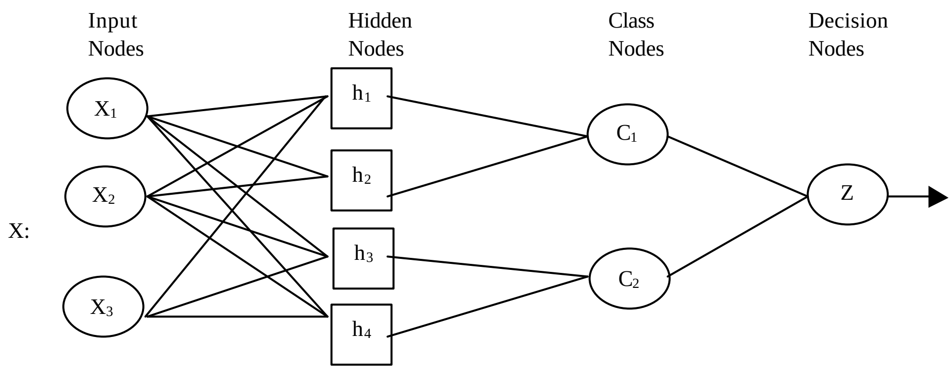

Fig. 1: Probabilistic neural network architecture

computes the sum of the activations of the hidden nodes that are connected to it (i.e., all the hidden nodes for a particular class) and passes this sum to a decision node. The decision node outputs the class with the highest summed activation. One of the greatest advantages of this network is that it doesn't require any iterative training and thus can learn quite quickly. However, one of the main disadvantages of this network is that it has one hidden node for each training instance and thus requires more computational resources (storage and time) during execution than many other models. When simulated on a serial machine, O(n) time is required to classify a single input vector. On parallel system, only O(log n) time is required, but n nodes and nm connections are still required (where n is the number of instances in the training set and m is the number of input attributes) [1].

The most directly way to reduce storage requirement and speed up execution is to reduce the number of nodes in the network. One common solution to this problem is to keep only a randomly selected subset of the original training data in building the network. However, arbitrary removing instances can reduce generalization accuracy. In addition, it is difficult to know how many nodes can be safely removed without a reasonable stopping criterion. Other subset selection algorithm exist in linear regression theory, including forward selection, in which the network starts with no nodes and nodes are added one at a time to the network. Another method that has been used is k-means clustering.

CLASSIFICATION THEORY OF PNN

The PNN is inspired from Bayesian classification and classical estimators for probability density functions. The basic operation performed by the PNN is an estimation of the probability density function of features of each class from the provided training samples using Gussian Kernel. These estimated densities are then used in a Bayes decision rule to perform the classification.

If the probability density function (pdf) of each of the population is known, then an unknown x belong to class i if:

$$f_i\,(x)>f_j\,(x),\ \text{for all}\ j\neq i \tag{8}$$

Where f_k is the pdf for class k.

There are other parameters which may be included during the parameter calculations and these parameters are:-Prior probability (h) which represents the probability of an unknown sample being drawn from a particular population and Misclassification cost (c) which expresses the cost of incorrectly classifying an unknown. According to the above definition of the pdf and the other parameters that should be included, the classification decision becomes:-

$$h_i c_i f_i>h_j c_j f_j,\ \text{for all}\ j\neq i \tag{9}$$

This is defined as Bayes optimal decision rule.

Estimating the pdf is done using the samples of the populations (the training set), accordingly: PDF for a single sample (in a population) is calculated from the following formula:

$$1/sW((x\text{-}x_k)/s) \tag{10}$$

Where: x: unknown (input), x_k:k^{th} sample, W: weighting function and s : smoothing parameter.

So, the PDF for a single population is calculated from the following formula which is known as Parzen's pdf estimator:

$$1/n\sigma\sum_{k=1}^{n} w(x\text{-}x_k/\sigma) \tag{11}$$

Which is exactly expresses the average of the pdf's for the" n" samples in the population. The estimated pdf approaches the true pdf as training set size increases as long as the true pdf is smooth. With regards to the weighting function [6], we see that it provides a sphere of influence since there is:-large values of small distances between the unknown and the training samples and it rapidly decreases to zero as the distance increases. The weighting function commonly use Gaussian function since it behaves well and easily computed and also it isn't related to any assumption about a normal distribution. When the weighting function use Gaussian function, the estimated pdf is given by [6]:

$$g(x) = 1/n\sigma\sum_{k=1}^{n}\exp\text{-}(x\text{-}x_k)^2/\sigma^2 \qquad (12)$$

In case of inputting the network a vector, here the PDF for a single sample (in a population) will be given by the following formula:

$$1/(2\prod)^{p/2}s^p\exp(-\| x\text{-}x_k \|^2/2s^2) \qquad (13)$$

Where x: unknown (input), x_k : k^{th} sample, s : sampling parameter, p : length of vector

And in that case of inputting the network a vector, the PDF for a single population is expressed as:

$$g_i(x) = 1/(2\prod)^{p/2}\sigma^p n_i\sum_{k=1}^{n_i}\exp(-\|x\text{-}x_k\|^2/2\sigma^2) \qquad (14)$$

Which is the average of the pdf's for the n_i samples in the i^{th} population. The classification criteria in this case of multivariate input will be expressed as follows:

$$g_i(x) > g_j(x), \text{ for all } j \neq i \qquad (15)$$

$$g_i(x) = (1/n_i)\sum\exp(-\|x\text{-}x_k\|^2/2\sigma^2) \qquad (16)$$

Which eliminates common factors and absorb the '2' into s.

PNN TRAINING SET

The training set of PNN must be done through representative of the actual population of effective classification and it is characterized by the following:- more demanding than most NN's, sparse set sufficient and erroneous samples and outliers tolerable. Adding and removing training samples simply involves adding or removing "neurons" in the pattern layer. As the training set increases in size, the PNN asymptotically converges to the Bayes optimal classifier. The training process of a PNN is essentially the act of determining

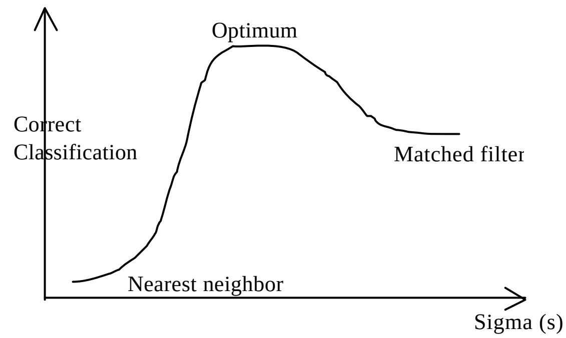

Fig. 2: Relationship between s and correct classification

the value of the smoothing parameter, sigma. The training of PNN is fast as orders of magnitude faster than backpropagation. Figure 2 shows the relationship between sigma (s) and its corresponding correct classification. To determine sigma, the educated guess is based on knowledge of the data and estimating a value using a heuristic technique [6].

VARIOUS APPLICATIONS OF PNN

A number of applications of the PNN in solving some problems in signal processing are presented in this section. These application covers: classification brain tissues in multiple sclerosis, classification image textures, classification of liver tumors, EEG pattern classification, cloud classification, power disturbance recognition and classification and allocation of the load profiles to consumers,….etc. Another type of PNN application regarding pattern recognition is also included in this section to cover: human face recognition, speech recognition and independent text recognition.

PNN application in pattern recognition problems: Neural networks plays an active role in pattern recognition either in human face or independent text or speech recognitions. Multilayer perceptron, time delay neural networks, backpropagation, radial basis function networks and fuzzy neural networks are some of them. PNN supply flexibility and straightforward design which make the system easily operable along with the successful classification results.

PNN in text independent speaker identification systems: Being widely used in pattern recognition tasks, neural networks have also been applied in speaker recognition. In [7], a text independent speaker identification system based on PNN was developed. The presented method proves that PNN based approach is suitable for real world speaker identification applications where a moderate number of speakers are available such as teleconference and speaker tracking. Using 100 feature vectors which are extracted from 0.8 second long speech, 27 of 28 speakers (96%) are correctly identified. Beside its high accuracy, another important factor to be considered is the ease at which the solution is obtained. The method is reasonably fast for working in real time. New speakers can directly be added to the system without the need to retrain the entire network. As the number of speakers increase, it is obvious that the success rate would decrease. The experiments showed that increasing the number of Mel Frequency Cepstral Coefficient (MCCs), codebook size or testing period could reduce the error rate. Success rate would further increase for shorter periods of testing samples if samples are processed to detect the voiced and unvoiced parts. In system [7], no such discrimination is applied. Silent parts of the samples are also considered as speech which adversely affects the success rate.

PNN application in human face recognition: The problem of face recognition in color images was dealt in [8]. Unlike in face recognition, where the classes to be discriminated are different faces, in face detection, the two respective classes are the 'face area' and the 'non face area'. A novel approach to face detection was presented in [8] based on PNN and fuzzy logic rules especially defined for skin are detection within the image frame. A PNN was trained for the identification of the facial areas which were extracted using the fuzzy logic rules.

The performance of the whole system was tested using 317 color images with different illumination conditions containing human faces. The images were taken using a digital camera having a resolution analysis of 1280x960 pixels. The proposed PNN joint with the fuzzy logic rules become the basis for developing a computer based face detection system whose overall identification performance was measured to be 83%. However, this performance level is achieved for frontal parallel faces, since the classification performance deteriorates when extended to different views of a human face.

PNN application in speaker verification: To improve speaker verification performance, [9] extends the well known PNN to Locally Recurrent Probabilistic Neural Networks LRPNN. In contrast to PNN that posses no feedback, LRPNN incorporates internal connections to the past outputs of all recurrent neurons, which render them sensitive to the context in which events occur. Thus, LRPNN were capable of identifying time and spatial corrections. A fast three step method was proposed for training the LRPNN. The first two steps are identical to the training of traditional PNN, while the third step is based on the differential evolution optimization method. A comparative experimental result for text independent speaker's verification was presented. They demonstrated the superior performance of the LRPNN architecture over the original PNN. A relative reduction of the error rate by more than 28% was achieved.

PNN applications in solving signal processing problems: This section covers a lot of applications related to classifications of: cloud, power disturbance recognition and classification, allocation of the load profiles to consumers and others.

Classification of brain tissues in multiple sclerosis: Ramakrishnan *et al*. [10] have proposed a new approach for the classification of brain tissues into White Matter, Gray Matter, Cerebral Spinal Fluid, Glial Matter, Connective and MS lesion in multiple sclerosis. The approach employs singular value decomposition on multiwavelet transformed images. Single level multiwavelet transformation decomposes images into 16 sub bands and each subband represents the image in a specific time frequency plane. Singular value decomposition is then employed on the subband coefficient matrices. Lower singular values are affected more by noise than higher singular values and hence only higher singular values are used to classify textures in the presence of noise.

The probability density function of the selected singular values is then modeled as an exponential distribution and the model parameter for the distribution is estimated using the maximum likelihood estimation technique. The model parameters, one for each subband are used as features for the classification. The classification is carried out using Weighted Probabilistic Neural Network (WPNN). Experiments have been carried out using data sets composed of three modalities of brain MR images, namely T1 and T2 relaxation times and proton density (PD) weighted MR images. The performance of the algorithm is analyzed in terms of classification rate at various noise levels and intensity non-uniformity levels.

Figure 3 shows the average classification rates of the proposed approach on MS brain dataset at different noise levels with various intensity non-uniformity levels. It may be observed from the figure that the proposed approach gives an average classification rate of 99% at 0% NL and INU and at 9% NL and 40% INU the average classification rate is about 84%.

Confusion matrix for the proposed approach at 0% NL and 0% INU is shown in Table 1. White matter,

Table 1: Confusion matrix for the proposed PDF based approach at 0% noise level and 0% intensity non-uniformity level

	White matter	Gray matter	CSF	Glial matter	Connective	MS lesion
White matter	75	3	3	0	0	0
Gray matter	3	76	2	0	0	0
CSF	3	2	76	0	0	0
Glial matter	0	0	0	78	0	3
Connective	0	1	1	0	79	0
MS lesion	0	0	0	5	0	76

Table 2: Average classification rate of the proposed approach with WPNN, PNN, back propagation neural networks and kNN

		Number of training texture images						
Classifier		4	5	6	7	8	9	10
BPNN	Avg. CR	88.2600	89.6800	90.7100	91.6600	92.1300	92.8400	93.2100
	Time required for training (in sec)	3.6212	4.3863	5.8937	7.0749	8.4913	9.5322	10.0978
kNN	Avg. CR	91.4400	92.3100	93.2700	93.8600	94.5200	94.7600	95.1700
	Time required for training (in sec)	0.5812	0.6236	0.7690	0.8025	0.9452	1.2040	1.7439
PNN	Avg. CR	95.2600	97.1600	97.5900	98.1800	98.2300	98.6400	98.9300
	Time required for training (in sec)	Training is instantaneous. Hence, zero time is required for training						
WPNN	Avg. CR	97.8700	98.2300	99.3400	99.4100	99.5600	99.5900	99.6200
	Time required for training (in sec)	0.1634	0.1863	0.1954	0.2082	0.2359	0.2589	0.2812

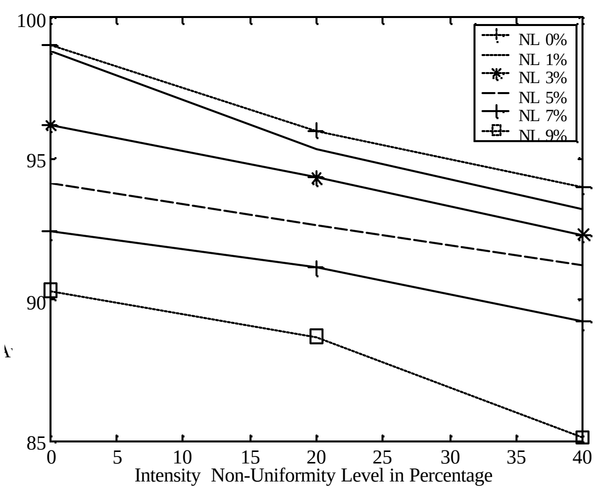

Fig. 3: Average classification rates of the proposed PDF based approach versus various intensity non-uniformity levels at various noise levels

gray matter and CSF, all three shows, 3-4% of misclassifications among them. 4% of the Glial matter is misclassified as MS lesion tissue. About 1% of Connective is misclassified as gray matter and CSF. About 6% of the MS lesion tissue is misclassified as Glial matter. Kappa value of the above confusion matrix is 0.9037.

Classification of image textures: Ramakrishnan *et al*. [11] have proposed a new approach for image texture classification based on curve fitting of wavelet domain singular values and probabilistic neural networks. Image textures are wavelet packet transformed and singular value decomposition is then employed on subband coefficient matrices after introducing non-linearity. The selected singular values are fitted to the exponential curve. The model parameters are estimated using population-sample analogues method and the parameters are used for performing the classification. A modified form of probabilistic neural network (PNN) called weighted PNN (WPNN) is employed for performing the classification. Experiments have been carried out to verify the ability of the proposed WPNN in achieving good classification rate, compared to PNN, back propagation neural networks (BPNN) and kNN. In this experiment, we have considered 111 images from Brodatz database. Out of 16 sub images, available for each texture, 4 to 10 sub images are used for training and remaining sub images are held for testing. Results are presented in Table 2.

It can be seen from the table that conventional BPNN requires more time for training compared to other classifiers. The kNN requires lees time for training than BPNN, however the PNN requires zero time, because the training is instantaneous. Unlike PNN, the WPNN requires approximately 0.1 to 0.3 seconds training. This is due to fact that the WPNN includes weighting factors between pattern layer and summation layer of the PNN, to achieve better discrimination among various patterns and computation

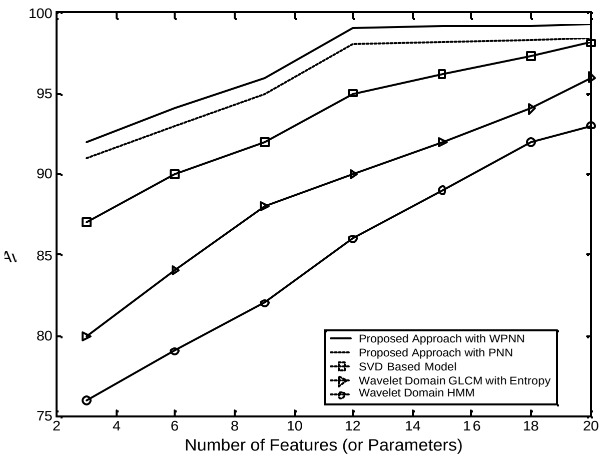

Fig. 4: Number of features (or Parameters) versus average classification rate various approaches

of the weighting factors require few seconds. It can be observed from the table that when small number of training texture images are used classification rate provided by BPNN and kNN is poor, whereas both PNN and WPNN provide good classification rate. However, WPNN provides better classification rate over other classifiers. This is due to the fact that WPNN basically enjoys the benefits of PNN such as single pass training and ability to approximate any PDF by sum of multivariate Gaussian functions. In addition to that, WPNN includes weighting factors based on class separabilities.

Classification of soil textures: Singular Value Decomposition (SVD) based novel approach using wavelet packet transformation is proposed by Ramakrishnan *et al*. [12] for classification of soil textures. The proposed approach extracts features such as energy, entropy, local homogeneity and min-max ratio from the singular values of wavelet packet transformation coefficients. A modified form of Probabilistic Neural Network (PNN) called Weighted PNN (WPNN) is employed for performing the classification.

Figure 4 shows the trade-off between number of features and classification rate for the proposed approach with PNN and with WPNN, wavelet domain SVD based model, wavelet domain HMM based approach and wavelet domain GLCM based approach [12]. It may be observed from the figure that the improvement in classification rate of the proposed approach over wavelet domain SVD based model, wavelet domain GLCM with entropy and wavelet domain HMM based approaches are approximately 4, 9 and 13% respectively. This improvement in classification rate is due to discrimination ability of the proposed feature and improved classification procedure using WPNN.

Classification of liver tumors: E-Liang Chen *et al*. [13] have presented a CT liver image diagnostic classification system which will automatically find, extract the CT liver boundary and further classify liver diseases. The system comprises a detect-before-extract (DBE) system which automatically finds the liver boundary and a modified probabilistic neural network classifier which uses specially designed feature descriptors to distinguish normal liver, two types of liver tumors, hepatoma and hemageoma. The DBE system applies the concept of the normalized fractional Brownian motion model to find an initial liver boundary and then uses a deformable contour model to precisely delineate the liver boundary. The neural network is included to classify liver tumors into hepatoma and hemageoma with feature descriptors which are generated by fractal feature information and the gray-level co-occurrence matrix. The system was evaluated by 30 liver cases and shown to be efficient and very effective. The confusion matrix for DBE system shows that about 1.5% misclassification between two class namely hepatoma and hemageoma.

EEG pattern classification: Toshio Tsuji *et al*. [14] have proposed a probabilistic neural network (PNN) based approach that can estimate a posteriori probability for a pattern classification problem. The structure of the network is based on a statistical model composed by a mixture of log-linearized Gaussian components. However, the forward calculation and the backward learning rule can be defined in the same manner as the error backpropagation NN. The PNN is applied to the electroencephalogram (EEG) pattern classification problem. In the experiments, two types of a photic stimulation, which are caused by eye opening/closing and artificial light, are used to collect the data to be classified. To examine the classification ability of the network, experiments are performed for five subjects (A, B, C, D: males, E: female). The network is trained using 112 data (56 for each class). Then, the ratio of the correct classification to 422 data that are not used in learning is computed. Classification rate of 84.3, 86.7, 88.7, 78.4 and 89.7% was achieved for subjects A, B, C, D and E, respectively.

Cloud classification: In cloud classification from satellite imagery, temporal change in the images is one of the main factors that cause degradation in the classifier performance. Bin Tian *et al*. [15] have developed a novel temporal updating approach for probabilistic neural network (PNN) classifiers that can be used to track temporal changes in a sequence of images. This is done by utilizing the temporal contextual information and adjusting the PNN to adapt to such changes. Whenever a new set of images arrives, an initial classification is first performed using the PNN updated up to the last frame while at the same time, a prediction using Markov chain models is also made based on the classification results of the previous frame. The results of both the old PNN and the predictor are then compared. Depending on the outcome, either a supervised or an unsupervised updating scheme is used to update the PNN classifier. Maximum likelihood (ML) criterion is adopted in both the training and updating schemes. The proposed scheme is examined on the Geostationary Operational Environmental Satellite (GOES) 8 satellite cloud imagery data. Each image was analyzed and classified into ten cloud/background classes. The overall classification rate increased from 65.8 to 75.4% after the updating process.

Power disturbance recognition and classification: Zwe-Lee Gaing [16] has implemented a prototype wavelet-based neural network classifier for recognizing power-quality disturbances and tested under various transient events. The discrete wavelet transform (DWT) technique is integrated with the probabilistic neural-network (PNN) model to construct the classifier. First, the multiresolution-analysis technique of DWT and the Parseval's theorem are employed to extract the energy distribution features of the distorted signal at different resolution levels. Then, the PNN classifies these extracted features to identify the disturbance type according to the transient duration and the energy features. Since the proposed methodology can reduce a great quantity of the distorted signal features without losing its original property, less memory space and computing time are required. Various transient events tested, such as momentary interruption, capacitor switching, voltage sag/swell, harmonic distortion and flicker show that the classifier can detect and classify different power disturbance types efficiently. Gaing has employed 35 (half of all training examples) and 70 training examples to train the PNN model, respectively. He also randomly created 20 distorted voltage waves to test the proposed approach. The experimental results shows that with 35 training examples were 35, the classified accuracy rate of the distorted signals of the approach was 80% (four distorted signal failures). When the training examples were 70, the classified accuracy rate was 90% (two distorted signal failures). The results show the more the training examples, the better the accuracy rate.

Allocation of the load profiles to consumers: The new emerged operating conditions in the power sector are forcing the power-market participants to develop new tools. Among them, load profiles are a key issue in retail power markets. For various types of small consumers without quarter-hourly load measurements, determination of Typical Load Profiles (TLPs) could serve as a tool for determining of their load diagrams. Their main function is in billing of consumers who have deviated from their contracted schedules. Moreover, a simple and straightforward method for assigning a TLP to a particular eligible consumer also needs to be established. David Gerbec *et al*. [17] devised a methodology for allocating consumers' load profiles using probabilistic neural network (PNN) is presented. It is based on the preprocessed measured load profiles (MLPs), using wavelet multiresolution analysis, clustered with a FCM clustering algorithm with an appropriate cluster-validity measure.

CONCLUSION AND FUTURE WORKS

Probabilistic Neural Network (PNN) is some kind of supervised neural network that is widely used for pattern recognition. The PNN is applicable to the same class of problems for which the back propagation neural network BPNN is typically used. Experimental results indicate that PNN has a number of major advantages over other traditional neural networks. The PNN learns from the training data instantaneously. This speed of learning gives the PNN the capability of adapting its learning in real time, deleting or adding training data as new conditions arises. A second advantage to PNN is that it can be shown to always converge to the Bayes optimal solution as the number of training samples increase. PNN belongs to the family of radial basis function neural networks, which due to their robustness are now widely used in various pattern classification tasks. As a case study with respect to application of PNN in text independent speaker identification system, PNN provides best matching speaker for each input vector. However, in order to increase the success rate, multiple input vectors from each sample are needed. At the decision level, a predefined number of outputs from the PNN are buffered and summed altogether. This way, each speaker in the system gets a score on the period of sample applied to the system. Then the speaker with the highest score is chosen to be the owner of the sample.

With regards to applying PNN in EEG pattern classification, the PNN is applied to the electronic phonograms (EEG) pattern classification problem. In this experiments, two types of a photic simulation was done, which are caused by eye opening/closing and artificial light, are used to collect the data to be classified. To examine the classification ability of the network, experiments are performed for five subjects (A, B, C, D: males, E: females). The network is trained using 112 data (56 for each class). Then, the ratio of the correct classification to 422 data that are not used in learning is computed. Classification rate of 84.5, 86.7, 78.4 and 89.7% was achieved for subjects A, B, C, D and E respectively.

As a future work to be done by others, we recommend to use PNN combined with fuzzy logic in analyzing human chromosomes and handwritten recognition to both Arabic and English letters for both on line and off line writing.

REFERENCES

1. Randall Wilson, D. *et al*., 1997. Implemented Center Point Selection for Probabilistic Neural Networks. Proceeding of the International Conference on Artificial Neural Networks and Genetic Algorithms, ICANNGA'97, pp: 514-517.
2. www.unipg.it/burrasc/ArticoliPdf/090.eann01p051.pdf
3. Wilson, D. Rondall and Tony R. Martinez, 1996. Hetrogenous Radial Basis Functions. Proceedings of International Conference on Neural Networks (ICNN'96), 2: 1263-1267.
4. Wilson, D. Rondall and Tony R. Martinez, 1997. Improved Heterogenous Distance Function. Journal of artificial Intelligence Research (JAIR), 6 (1): 1-34.
5. Stanfill, C. and D. Waltz, 1986. Towards Memory Based Reasoning. Communication of the ACM, 29.
6. www.psi.toronto.edu/~vincent/research/presentations/PNN.pdf
7. www.bayrak.ualr.edu/symsel/mustafa/docs/CCCT_2004.pdf
8. www.med.ee.nd.edu/MED11/pdf/papers/t3-031.pdf
9. www.imaa-dtasouli.imaa.ic.ac.uk/~dtas/papers/GanchevTVF2004.pdf
10. Ramakrishnan, S. and S. Selvan, 2006. Classification of Brain Tissues Using Multiwavelet Transformation and Probabilistic Neural Network. International Journal of Simulation: Systems, Science and Technology, A publication of the United Kingdom Simulation Society, 7 (9): 9-25.
11. Ramakrishnan, S. and S. Selvan, 2007. Image Texture Classification Using Wavelet Based Curve Fitting and Probabilistic Neural Network. International Journal of Imaging Systems and Technology, A Publication of Wiley InterScience, 17 (4): 266-275.
12. Ramakrishnan, S. and S. Selvan, 2008. Classification of Soil Texture Based on Wavelet Domain Singular Values. International Journal of Tomography & Statistics, Indian Society for Development and Environment Research, To be Published, 9 (S08).
13. E-Liang Chen, Pau-Choo Chung, Ching-Liang Chen, Hong-Ming Tsai and Chein-I Chang, 1998. An Automatic Diagnostic System for CT Liver Image Classification. IEEE Transactions On Biomedical Engineering, 45 (6): 783-794.
14. Toshio Tsuji, Osamu Fukuda, Hiroyuki Ichinobe and Makoto Kaneko, 1999. A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification. IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 29 (1): 60-72.
15. Bin Tian, Mahmood R. Azimi-Sadjadi, Thomas H. Vonder Haar and Donald Reinke, 2000. Temporal Updating Scheme for Probabilistic Neural Network with Application to Satellite Cloud Classification. IEEE Transactions on Neural Networks, 11 (4): 903-920.
16. Zwe-Lee Gaing, 2004. Wavelet-Based Neural Network for Power Disturbance Recognition and Classification. IEEE Transactions on Power Delivery, 19 (4): 1560-1568.
17. David Gerbec, Samo Gasperic, Ivan Smon and Ferdinand Gubina, 2005. Allocation of the Load Profiles to Consumers Using Probabilistic Neural Networks. IEEE Transactions on Power Systems, 20 (2): 548-555.