Lalit P. Bhaiya, Virendra Kumar Verma / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.751-756

Classification of MRI Brain Images Using Neural Network

Lalit P. Bhaiya[1] , Virendra Kumar Verma[2]
[1] Associate Prof. & Head of ETC , RCET , Bhilai , C.G. India
[2]M.Tech Student, Department of Electronics & Tele-Communication, RCET, Bhilai, C.G. India

ABSTRACT

There are many difficult problems in the field of pattern recognition. These problems are the focus of much active research in order to find efficient approaches to address them. We have tried to address the problem of classification MRI brain images by creating a robust and more accurate classifier which can act as an expert assistant to medical practitioners. Magnetic Resonance Imaging (MRI) is the state-of the-art medical imaging technology which allows cross sectional view of the body with unprecedented tissue contrast. MRI plays an important role in assessing pathological conditions of the ankle, foot and brain.

In proposed methodology three supervised neural networks has been used: Back Propagation Algorithm (BPA), Learning Vector Quantization (LVQ) and Radial Basis Function (RBF). The features of magnetic resonance images have been reduced, using principal component analysis (PCA), to the more essential features. The proposed technique has been carried out over a larger database as compare to any previous work and is more robust and effective.

Keywords- Magnetic Resonance Image(MRI) , Principal Component Analysis (PCA), Radial Basis Function (RBF), Back Propagation (BP), Learning Vector Quantization (LVQ), Multi Layer Neural Network .

INTRODUCTION

Magnetic resonance imaging (MRI) is often the medical imaging method of choice when soft tissue delineation is necessary. This is especially true for any attempt to classify brain tissues [1]. The most important advantage of MR imaging is that it is non-invasive technique [2]. The use of computer technology in medical decision support is now widespread and pervasive across a wide range of medical area, such as cancer research, gastroenterology, hart diseases, brain tumors etc. [3, 4]. Fully automatic normal and diseased human brain classification from magnetic resonance images (MRI) is of great importance for research and clinical studies. Recent work [2, 5] has shown that classification of human brain in magnetic resonance (MR) images is possible via supervised techniques such as artificial neural networks and support vector machine (SVM) [2], and unsupervised classification techniques unsupervised such as self organization map (SOM) [2] and fuzzy c-means combined with feature extraction techniques [5]. Other supervised classification techniques, such as k-nearest neighbors (k-NN) also group pixels based on their similarities in each feature image [1, 6, 7, 8] can be used to classify the normal/pathological T2-wieghted MRI images. We used supervised machine learning algorithms (ANN and k-NN) to obtain the classification of images under two categories, either normal or abnormal.

Usually an image of size $p \times q$ pixels is represented by a vector in $p.q$ dimensional space. In practice, however, these ($p.q$) -dimensional spaces are too large to allow robust and fast object recognition. A common way to attempt to resolve this problem is to use dimension reduction techniques. In order to reduce the feature vector dimension and increase the discriminative power, the principal component analysis (PCA) has been used.

In these approaches, the 2-dimensional image is considered as a vector, by concatenating each row or column of the image. Each classifier has its own representation of basis vectors of a high dimensional face vector space. The dimension is reduced by projecting the face vector to the basis vectors, and is used as the feature representation of each images. [8],[15]

The Back Propagation (BP) algorithm looks for the minimum of the error function in weight space using the method of gradient descent. Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs. [3]

The RBF network performs similar function mapping with the BP, however its structure and function are much different. An RBF is a local network that is trained in a supervised manner contrasts with the BP network that is a global network. A BP performs a global mapping, meaning all inputs cause an output, while an RBF performs a local mapping, meaning only inputs near a receptive field produce activation.

The LVQ network has two layers: a layer of input neurons, and a layer of output neurons. The

network is given by prototypes W=(w(i),...,w(n)). It changes the weights of the network in order to classify the data correctly. For each data point, the prototype (neuron) that is closest to it is determined (called the winner neuron). The weights of the connections to this neuron are then adapted, i.e. made closer if it correctly classifies the data point or made less similar if it incorrectly classifies it. [16]

We performed classification of MRI brain images on a database of 192 images which contains 107 normal images and 85 pathological images. We experimented with three different sets of training and testing taken from clump of images. In first case s 98 (55 normal and 43 pathological) images have been used for training purpose and remaining 94 images for testing. In second case we swapped the testing and training database and in third case we used 90(50 normal and 40 pathological) images for training and remaining 102 images for testing.

For feature vectors generation, images are preprocessed by PCA which has been described shortly below.

PCA Preprocessing

PCA can be used to approximate the original data with lower dimensional feature vectors. The basic approach is to compute the eigenvectors of the covariance matrix of the original data, and approximate it by a linear combination of the leading eigenvectors. By using PCA procedure, the test image can be identified by first, projecting the image onto the eigen face space to obtain the corresponding set of weights, and then comparing with the set of weights of the faces in the training set. [2],[5]

The problem of low-dimensional feature representation can be stated as follows: Let X= (x_1 ,x_2,…, x_i,…, x_n) represents the $n \times N$ data matrix, where each x_i is a face vector of dimension n, concatenated from a $p \times q$ face image. Here n represents the total number of pixels (p.q) in the face image and N is the number of face images in the training set. The PCA can be considered as a linear transformation (1) from the original image vector to a projection feature vector, i.e.

$$Y = W^T X \quad (1)$$

where Y is the $m \times N$ feature vector matrix, m is the dimension of the feature vector, and transformation matrix W is an $n \times m$ transformation matrix whose columns are the eigenvectors corresponding to the m largest eigen values computed according to the formula (2):

$$\lambda e_i = S e_i \quad (2)$$

where e_i ,λ are eigenvectors & eigen values matrix respectively.

Here the total scatter matrix S and the mean image of all samples are defined as

$$S = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T, \quad \mu = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (3)$$

After applying the linear transformation W^T, the scatter of the transformed feature vectors $\{y_1, y_2, \ldots, y_N\}$ is $W^T S W$. In PCA, the projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projected samples, i.e.,

$$W_{opt} = \arg \max_W |W^T S W| = [w_1 w_2 \ldots w_m] \quad (4)$$

Where $\{w_i \mid i = 1, 2, \ldots, m\}$ is the set of n – dimensional eigenvectors of S corresponding to the m largest eigen values. In other words, the input vector (face) in an n -dimensional space is reduced to a feature vector in an m -dimensional subspace. We can see that the dimension of the reduced feature vector m is much less than the dimension of the input faces vector n.

PREPROCESSING OUTPUT

After preprocessing images by PCA, feature vectors of reduced dimension are produced. PCA produces feature vector of dimension 20. We experimented with three different sets of training and testing taken from clump of images. In all the cases considering the training sample n so input to neural network has become the feature vector matrix of size 20 by n for PCA.

Classification:-

Input matrix to the neural network is of size 20 by n while target matrix size is determined on the basis of number of classes. Target matrix is of size 2 by n where if input feature vector (column wise) belong to class 2 then corresponding output vector will have 1 at 2nd row and 0 at other rows. Here value 1 in any target vector denotes the belongingness of an image to the class denoted by respective row value of target vector.

To classify input feature vectors into target vectors, we used Back Propagation (BP), Radial Basis Function (RBF) & Learning Vector Quantization (LVQ). We configured and tested each neural network with various configurations. Variations are made in the following components: Number of input to neural network, Number of hidden layers, Number of nodes in hidden layers, learning rate. In case of RBF SPREAD is also varied considering the condition that SPREAD is large enough so that the active input regions of the radial neurons overlap enough so that several radial neurons always have fairly large outputs at any given moment. However, SPREAD should not be so large that each neuron is effectively responding in the same, large, area of the input space. [11],[13] The optimum configurations which have generated good testing results are shown in tables.

Back Propagation as Classifier

The weighting factor of the input-to-hidden neurons can be computed by (5)

$$w_{ij}^{(k+1)} = w_{ij}^{k} - \eta \frac{\partial E^{(k)}}{\partial w_{ij}} \quad (5)$$

Where k is iteration number; i, j are index of input and hidden neuron, respectively; and η is step size $\frac{\partial E}{\partial w_{ij}}$ can be calculated from the following series of equations (6)-(8). The error function is given by

$$E = \frac{1}{2}\sum_{l=1}^{p}(t_l - o_l)^2 \quad (6)$$

Where p is the number of output neurons, l is the index of neuron, t_l and o_l are the target and output values, respectively. The activation function, net function and output function are given by equation (7)

$$s_i = \frac{1}{1+e^{(-\lambda net_i)}} \quad (7)$$

$$net_i = \sum_{l=1}^{n} w_{il} x_l + w_{in+1} \quad (8)$$

$$o_i = \sum_{l=1}^{m} v_{il} s_l + v_{im+1} \quad (9)$$

Where n is the number of input neurons, and m is the number of output neurons. Let us define

$$\frac{\partial E}{\partial net_i} = \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial net_i} \quad (10)$$

And,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \quad (11)$$

then we obtain the weight update equation (5) for the input-to-hidden layer by computing Eq. (10) and Eq. (11) with the Eqs. from (6) to (9). Next, v_{ij}, hidden–to–output neurons' weight update can also be derived in the same way.

Back Propagation networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1. [5],[6]

The optimum configuration of BP neural network for PCA, used for training & testing is shown in table-I.

Table-I : BP neural network Configuration

Input Vector nodes	20
Number of hidden layers	2
Number of neurons (hidden layer 1 ,hidden layer 2 & output layer)	20 ,35,2
Transfer functions (hidden layer 1 , hidden layer 2 & output layer)	tansig, tansig, purelin
Network Learning rate	0.0001

Radial Basis Function as Classifier

The RBF network performs similar function mapping with the multi-layer neural network, however its structure and function are much different. A RBF is a local network that is trained in a supervised manner. RBF performs a local mapping, meaning only inputs near a receptive field produce activation. [9],[10]

The input layer of this network is a set of n units, which accept the elements of an n -dimensional input feature vector. n elements of the input vector x are input to the l hidden functions, the output of the hidden function, which is multiplied by the weighting factor $w(i, j)$, is input to the output layer of the network y (x). For each RBF unit k , k = 1, 2,3,..., l the center is selected as the mean value of the sample patterns belong to class k , i.e.

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N_k} x_k^i \quad , k=1,2,3, \dots ,m \quad (12)$$

Where x_k^i is the eigenvector of the i th image in the class k, *and* N_k is the total number of trained images in class k.

Since the RBF neural network is a class of neural networks, the activation function of the hidden units is determined by the distance between the input vector and a prototype vector. Typically the activation function of the RBF units (hidden layer unit) is chosen as a Gaussian function with mean vector μ_i and variance vector σ_i as follows

$$h_i(x) = \exp\left[-\frac{\|x-\mu_i\|^2}{\sigma_i^2}\right] \quad , i=1,2,\dots,l \quad (13)$$

Note that x is an n -dimensional input feature vector, μ_i is an n -dimensional vector called the center of the RBF unit, σ_i is the width of the i^{th} RBF unit and l is the number of the RBF units. The response of the j^{th} output unit for input x is given as:

$$y_j(x) = \sum_{i=1}^{l} h_i(x) w(i, j) \quad (14)$$

Where $w(i, j)$ is the connection weight of the i -th RBF unit to the j -th output node. The optimum configuration of RBF with PCA used for training and testing is shown in Table II.

Table-II: RBF neural network Configuration

Number of Radial Basis Layers	1
Number of neurons (input ,radial basis & output layer)	20,135,2
Spread	0.8

Learning Vector Quantization as Classifier

LVQ neural network combines the competitive learning with supervised learning and it can realize nonlinear classification effectively. There are several variations of the basic LVQ algorithm. The most common are LVQ1, LVQ2 and LVQ3. The basic LVQ neural network classifier (LVQ1), which is adopted in our work, divides the input space into disjoint regions. A prototype vector represents each region. In order to classify an input vector, it must be compared with all prototypes. The Euclidean distance metric is used to select the closest vector to the input vector. The input vector is classified to the same class as the nearest prototype. The LVQ classifier consists of an input layer, a hidden unsupervised competitive layer, which classifies input vectors into subclasses, and a supervised linear output layer, which combines the subclasses into the target classes. In the hidden layer, only the winning neuron has an input of one and other neurons have outputs of zero. The weight vectors of the hidden layer neurons are the prototypes. The number of the hidden neurons is defined before training and it depends on the complexity of the input-output relationship. Moreover it significantly affects the results of differentiation. We carefully select the number of hidden neurons based on extensive simulation experiments. [14]

The learning phase starts by initiating the weight vectors of neurons in hidden layer. The input vectors are presented to the network in turn. For each input vector X_j , the weight vector W_c of a winning neuron i is adjusted. The winning neuron is chosen according to:

$$\|X_j - W_c\| \leq \|X_j - W_k\|, \text{ for } k \neq c \qquad (15)$$

The weight vector W_c of the winning neuron is updated as follows:

If X_j and W_c belong to same class, then

$$W_c(n+1) = W_c(n) + \alpha(n)(X_j - W_c(n)) \qquad (16)$$

If X_j and W_i do not belong to the same class, then

$$W_c(n+1) = W_c(n) - \alpha(n)(X_j - W_c(n)) \qquad (17)$$

The weight vectors of other neurons keep constant.

$$W_k(n+1) = W_k(n) \qquad (18)$$

where $0 \leq \alpha(n) \leq 1$ is the learning rate. The training algorithm is stopped after reaching a pre-specified error limit. Because the neural network combines the competitive learning with supervised learning, its learning speed is faster than BP network. The optimum configuration of LVQ with PCA & R-LDA , used for training and testing is shown in Table III.

Table-III : LVQ neural network Configuration

Number of competitive Layers	1
Number of neurons (input ,competitive & output layer)	30,40,2
Transfer function	Lvq 1.0
Network Learning rate	0.001

Training Graphs and Results

Each neural network took different time for training input feature vectors. RBF neural network was the fastest while LVQ took much time than others. Training graphs of BP applied to PCA preprocessed training set are shown in figure 1 .

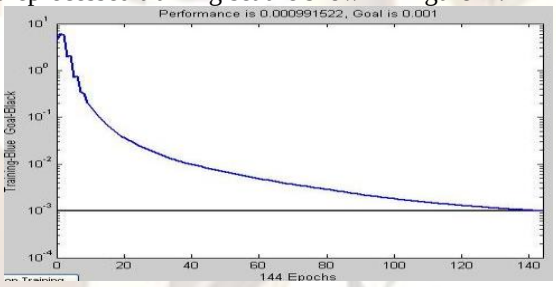

Fig1.Learning of BP after preprocessing by PCA.

RBF creates radial basis layer neurons one at a time when training starts. In each iteration network error is lowered by appropriate input vector. This procedure is repeated until the error goal is met, or the maximum number of neurons is reached. In our case RBF creates 135 neurons for PCA input vectors .Training graphs of RBF applied to PCA preprocessed training set are shown in figure 2:

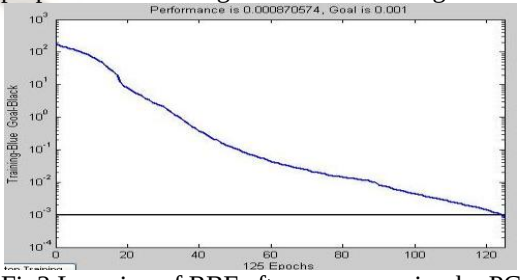

Fig2 Learning of RBF after preprocessing by PCA

Accordingly Training graphs of LVQ applied to PCA preprocessed training set are shown in figure 3.

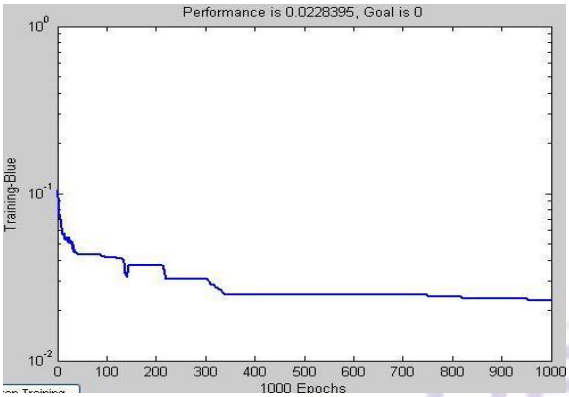

Fig3. LVQ Learning after preprocessing by PCA

PCA preprocessed input vectors' training results for first case shown in table IV.

Methods =>	PCA with BP	PCA with RBF	PCA with LVQ
No. of error images =>	4	7	9
Recognition Rate =>	95.7 % (90/94)	92.5% (87/94)	90.4 %(85/94)

Table-IV : Recognition Rate using PCA with BP, PCA with RBF and PCA with LVQ

PCA preprocessed input vectors' training results for second case shown in table V

Methods =>	PCA with BP	PCA with RBF	PCA with LVQ
No. of error images =>	3	6	8
Recognition Rate =>	96.9 % (95/98)	93.8% (92/98)	91.8% (90/98)

Table-V: Recognition Rate using PCA with BP, PCA with RBF and PCA with LVQ

PCA preprocessed input vectors' training result for third case shown in table VI

Methods =>	PCA with BP	PCA with RBF	PCA with LVQ
No. of error images =>	5	7	9
Recognition Rate =>	95.0 % (97/102)	93.1% (95/102)	91.1% (93/102)

Table-VI: Recognition Rate using PCA with BP, PCA with RBF and PCA with LVQ

CONCLUSION

In this study, we have developed a medical decision support system with normal and abnormal classes. The medical decision making system designed by the wavelet transform, principal component analysis (PCA), and supervised learning methods (BPA ,RBFN and LVQ) that we have built gave very promising results in classifying the healthy and pathological brain. The benefit of the system is to assist the physician to make the final decision without hesitation.

REFERENCES

[1] L. M. Fletcher-Heath, L. O. Hall,D. B. Goldgof, F.Murtagh; Automatic segmentation of non-enhancing brain tumors in magnetic resonance images; Arti_cial Intelligence in Medicine 21 (2001), pp. 43-63.

[2] Sandeep Chaplot, L.M. Patnaik, N.R. Jagannathan; "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network"; Biomedical Signal Processing and Control 1 (2006), pp. 86-92.

[3] http://www.abta.org/siteFiles/SitePages/5E8399DBEEA8F53CBBBBF21C63AE113.pdf

[4] A.Sengur, "An expert system based on principal component analysis, artificial immune system and fuzzy k-NN for diagnosis of valvular heart diseases" Comp. Biol. Med. (2007), doi: 10.1016/j.compbiomed.2007.11.004.

[5] M . Maitra, A. Chatterjee; "Hybrid multiresolution Slantlet transform and fuzzy c-means clustering approach for normal-pathological brain MR image segregation", MedEngPhys(2007),doi:10.1016/j.medengphy.2007.06.009.

[6] P. Abdolmaleki, Futoshi Mihara, Kouji Masuda, Lawrence Danso Buadu; Neural network analysis of astrocytic gliomas from

MRI appearances' Cancer Letters 118 (1997), pp. 69-78.

[7] T. Rosenbaum, Volkher Engelbrecht, Wilfried Kro?lls, Ferdinand A. van Dorstenc, Mathias Hoehn-Berlagec, Hans-Gerd Lenard; MRI abnormalities in neuro_bromatosis type 1 (NF1): a study of men and mice; Brain & Development 21 (1999), pp. 268-273. C. Cocosco , Alex P. Zijdenbos, Alan C. Evans; A fully automatic and robust brain

[8] MRI tissue classi_cation method; Medical Image Analysis 7 (2003), pp. 513-527..

[9] Lisboa, P.J.G., Taktak, A.F.G.: "The use of artificial neural networks in decision support in cancer: a systematic review." Neural Networks 19, 408–415 (2006)

[10] Alfredo Vellido , Paulo J.G. Lisboa "Neural Networks and Other Machine Learning Methods in Cancer Research" F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, pp. 964–971, 2007.

[11] Lisboa, P.J.G., Wong, H., Harris, P., Swindell, R.: "A bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer". Artif. Intell. Med. 28, 1–25 (2003)

[12] Lisboa, P.J.G., Vellido, A., Wong, H.: "A Review of Evidence of Health Benefit from Artificial Neural Networks in Medical Intervention." In: Artificial Neural Networks in Medicine and Biology, pp. 63–71. Springer, London (2000).

[13] X. Lu, Y. Wang and A. K. Jain, "Combining Classifier for Face Recognition," Proc. of IEEE 2003 Intern. Conf. on Multimedia and Expo. Vol. 3. pp. 13-16, 2003.

[14] Xin Ma; Wei Liu; Yibin Li; Rui Song, "LVQ Neural Network Based Target Differentiation Method for Mobile Robot" Advanced Robotics, 2005. ICAR '05. Proceedings. 12th International Conference on 18-20 July 2005.

[15] Xudong Jiang; Mandal, B.; Kot, A. "Eigenfeature Regularization and Extraction in Face Recognition" Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 30, Issue 3, March 2008.

[16] Yan Jun Wang Dian-hong "Sequential face recognition based on LVQ networks" VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop.