International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 4 (2012), pp. 425-434

http://www.irphouse.com

Hand Written Digit Recognition using Kalman Filter

VVKDV Prasad and PPS Subhashini

Department of Electronics & Communication Engineering
RVR & JC College of Engineering, Chowdavaram,
Guntur, 522019-A.P., India
E-mail: varrevkdvp@rediffmail.com, k_shiva_111@yahoo.co.in

Abstract

Different pattern recognition techniques are used in various fields for different applications. Methods based on Radial basis function (RBF) neural networks are found to be very successful in pattern classification problems. Training neural network is in general a challenging nonlinear optimization problem. Several algorithms have been proposed for selection of RBF neural network prototypes and training the network. This paper proposes RBF neural network using kalman filter training method for recognition of hand written digits. The working of the proposed method is tested on hand written digits of different fonts. Experiments revealed that this method is successful in recognizing the digits.

Keywords: Neural network, RBF neural network, kalman filter Training, Zoning method

Introduction

There has been a drastic change in our perspective of concept of communication and connectivity with digital revolution. Biometrics play vital role in dealing with problem of authentication. It is the science of identifying or verifying the identity of a person based on physiological or behavioural characteristics. Physiological characteristics include fingerprints, iris, hand geometry and facial image. The behavioural characteristics are actions carried out by a person in a characteristic way and include recognition of signature, machine printed characters, handwriting, and voice. There have been attempts to explore the possibility of efficient man-machine communication through hand written characters. Pattern recognition techniques using RBF neural network are helpful in classification of hand written characters of different users [1].

The applications of the character recognition include postal address system, signature verification system, recognition of characters from form filled applications and so on. Character recognition is basically of two types, off-line character recognition and on-line character recognition [2]. Image is accepted as input from the scanner in off-line character recognition. It is more difficult than on-line character recognition because of unavailability of contextual information and lack of prior knowledge like text position, size of text, order of strokes, start point and stop point. Noise will also exist in the images acquired in off-line character recognition e.g Machine Printed character recognition. In on-line character recognition, system accepts the moment of pen from the hardware such as graphic tablet, light pen and lot of information is available during the input process such as current position, moment's direction, start points, stop points and stroke orders. Hand written character recognition comes under this category. In this paper RBF neural network using Kalman filter training method is proposed for recognition of hand written digits. The efficiency of the proposed method is tested on the handwritten digits of different fonts. From the results it is found that the proposed method has good success rate in hand written digit recognition.

Artificial Neural Networks

A neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest. To achieve good performance, they employ a massive interconnection of simple computing cells referred to as 'Neurons' or 'processing units' [3]. A neural network can be viewed as an adaptive machine and it is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects: One is Knowledge acquired by the network from its environment through a learning process. Second is Interneuron connection strengths, known as synaptic weights, used to store the acquired knowledge. Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems[4]. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Neural networks that are designed to work with patterns can be classified as pattern classifiers or pattern associators.

RBF Neural Networks

Radial basis function network (RBF) is a type of artificial network for applications to problems of supervised learning e.g. regression, classification and time series prediction. RBF networks can be used to solve classification problems [5]. The classification problem can be treated as a non-parametric regression problem if the outputs of the estimated function are interpreted as the probability that the input belongs to the corresponding classes.

The training output values are vectors of length equal to the number of classes. After training, the network responds to a new pattern with continuous values in each component of the output vector and these values are interpreted as being proportional to class probability. This network consists of three layers, input layer, hidden layer, output layer as shown in Fig. 1. The m-dimensional input X is passed directly to a hidden layer. Suppose there are c neurons in the hidden layer, each of the c neurons in the hidden layer applies an activation function which is a function of the Euclidean distance between the input and an m-dimensional prototype vector.

Each hidden neuron contains its own prototype vector as a parameter. The output of each hidden neuron is then weighted and passed to the output layer. The outputs of the network consist of sums of the weighted hidden layer neurons. The design of an RBF requires several decisions that include the number of hidden units in the hidden layer, values of the prototypes, the functions used at the hidden units and the weights applied between the hidden layer and the output layer. The performance of an RBF network depends on the number and location (in the input space) of the centres, the shape of the RBF functions at the hidden units and the method used for determining the network weights. Some researchers have trained RBF networks by selecting the centers randomly from the training data. Others have used unsupervised procedures (such as the k-means algorithm) for selecting the RBF centers. Still others have used supervised procedures for selecting the RBF centers.

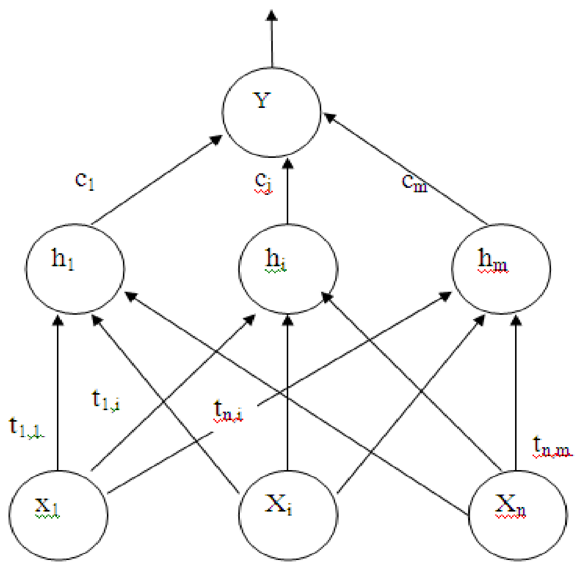

Figure 1: RBF Neural network architecture

A Working Principle: The principle of radial basis functions derives from the theory of functional approximation. Given N pairs (x_i, y_i), one looks for a function f of the form: $f(x) = \sum_{i=1}^{k} c_i h(|x_i - t_i|)$.In this, 'h' is the radial basis function (normally a

Gaussian function) and t_i are the k centers which have to be selected. The coefficients c_i are also unknown at the moment and have to be computed.Here x_i and t_i are elements of an n – dimensional vector space.

The hidden units compute the Euclidean distance between the input pattern and the vector, which is represented by the links leading to this unit. The activation of the hidden units is computed by applying the Euclidean distance to the function h as $h(x)=e(x-c_i)^2/r^2$. The parameters for the function h are its center c and its radius r. There have been a number of activation functions that include Gaussian function, Thin plate spline function, Multiquadric function and so on for the hidden layer of RBF neural network. In this paper the following function is considered.

$$g(v)=(g_0(v))^{1/(1-p)} \quad 1$$

where p is a parameter and $g_0(v)$ is a linear function of the form

$$g_0(v)=av+b \quad 2$$

where $a > 0$ and $b \geq 0$. The single output neuron gets its input from all hidden neurons. The links leading to the output neuron hold the coefficients c_i. The activation of the output neuron is determined by the weighted sum of its inputs. An RBF network is considered non-linear if the basis functions can move or change size or if there is more than one hidden layer otherwise the RBF network is considered linear. The above architecture can easily be extended to include more than one output node depending on the problem that the RBF network is to solve e.g classification into different classes would need as many output nodes as the number of classes.

When the RBF network is used in classification, the hidden layer performs clustering while the output layer performs classification. The hidden units would have the strongest impulse when the input patterns are closed to the centers of the hidden units and gradually weaker impulse as the input patterns moved away from the centers. The output layer would linearly combine all the outputs of the hidden layer. Each output node would then give an output value, which represents the probability that the input pattern falls under that class.

Kalman Filter Training Method

In this paper kalman filter training of RBF neural network is proposed.It is a supervised training method.The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states [6]. Kalman filter is a optimal Recursive data processing algorithm.The Kalman filter addresses the general problem of trying to estimate the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k =Ax_{k-1}+Bu_{k-1}+w_{k-1} \quad 3$$

with a measurement z i.e,

$$z_k = Hx_k + v_k \qquad 4$$

In this process, A in the difference equation relates the state at the previous time step k-1 to the state at the current step k in the absence of either a driving function or process noise. The B in the difference equation relates the optional control input u to the state x. The H in the measurement equation relates the state to the measurement z_k. The random variables w and v represent the process and measurement noise (respectively) and their normal probability distributions are

$$p(w) \sim N(0, Q) \qquad 5$$

$$p(v) \sim N(0, R) \qquad 6$$

Where Q, R are the process noise covariance and measurement noise covariance. Here it is assumed that they are constant.

If $\hat{x}_k^-$ is the priori state estimate at step k having the knowledge of the process prior to step k and $\hat{x}_k$ is the posteriori state estimate at step k knowing the measurement z_k, then priori estimate and posteriori estimate errors can be defined as $e_k^- \equiv x_k - \hat{x}_k^-$ and $e_k \equiv x_k - \hat{x}_k$ respectively. The kalman filter estimates a process by using a form of feedback control. The filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. As such, the equations for the kalman filter fall into two groups: Time update equations and Measurement update equations[7].

a. Time update equations: They are responsible for projecting (in time) the current state and error covariance estimates to obtain the priori estimates for the next time step. The time update equations can also be thought of as predictor equations. The time update equations project the state and covariance estimates forward from time step k-1 to step k. Let p_k^- is the priori estimate error covariance and p_k posteriori estimate error covariance, then

$$\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_{k-1} \qquad 7$$

$$p_k^- = Ap_{k-1}A^T + Q \qquad 8$$

b. Measurement update equations: They can be thought of as corrector equations and they are responsible for the feedback i.e for incorporating a new measurement into a priori estimate to obtain an improved posteriori estimate. If K is Kalman gain, R is measurement noise covariance

$$K_k = p_k^- H^T (H p_k^- H^T + R)^{-1} \qquad 9$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \qquad 10$$

$$p_k = (1 - K_k H) p_k^- \qquad 11$$

In the process first task is to compute kalman gain, K_k and next step is to actually measure the process to obtain z_k and then to generate an posteriori error covariance estimate. After each time and measurement update pair, the process is repeated with the previous posteriori estimates used to predict the new priori estimates. This recursive nature is one of the very appealing features of kalman filter. Also it requires less memory because each updated estimate of the state is computed from the previous estimate and new input data. So only previous estimate requires storage and no need of storing entire past observed data. One of the fruitful application of kalman filter is for training the RBF neural network. By assuming the state of the system as cascaded weight vector and prototype vector, kalman equations can be applied for weight updation.

Experimental Results

The results obtained on using the proposed RBF neural network based on Kalman filter training method for recognition of hand written digits of different fonts are presented in this section. Hand written digits from 0 to 9 of 25 different fonts are taken and they are recognized by using this method. For each hand written digit, after obtaining the thinned image, 16 features are extracted using Zoning method. These features are used to train RBF neural network [8], [9][10].

The different steps involved in recognition of the digits using this method are shown in the block diagram in the Fig.2

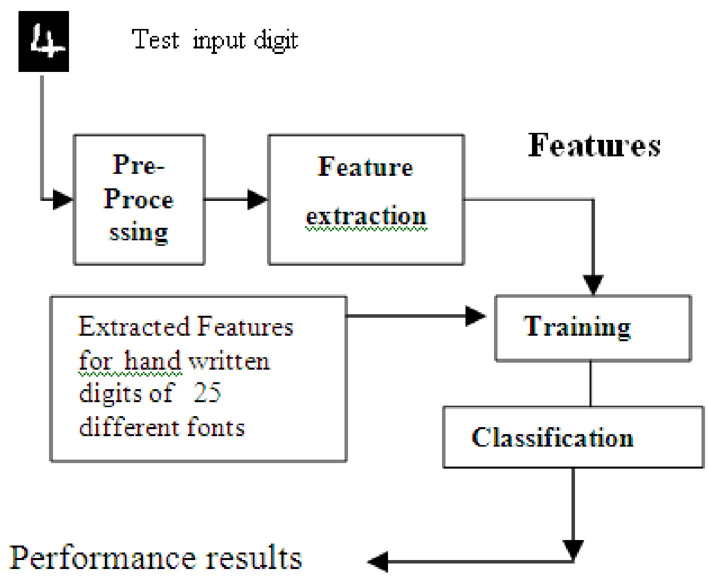

Figure 2: Block Diagram of Hand Written Digit recognition

The RBF networks are trained using the hidden layer function of Eq. (1) The exponential parameter p in Eq. (1) is taken as 2 in the results presented in this section. The training algorithms are initialized with prototype vectors randomly selected from the input data and with the weight matrix W set to zero[10]. The performance of each of the training methods is explored by averaging its performance over five trials, where each trial consists of a random selection of training and test data[11]. The number of hidden units in the RBF network is varied between 1 and 15.

Fig. 3 shows the number of iterations required for convergence. Percentage of correct classification for number of hidden units is given in Fig.4 for Kalman Filter training. The RBF network is trained with 13 different fonts of each digit and is also tested with 12 other fonts which are different from the trained fonts. It is observed that this method is successful in recognizing the digits of all these 25 fonts.

For space consideration, the results of six hand written digits viz 0 to 5 are only shown here. The features of these handwritten digits are extracted and tabulated in Table 1 and 2. Each digit has 16 features. The combination of all these features is called a feature vector. The feature vector is applied to neural network. If the digit is recognized, then the output node corresponding to the digit is one and the remaining are zeros [12]. The training method here applied is supervised training method. This requires the target vector that is the desired output, which is tabulated in Table 3.

Table 1: Features of Handwritten Digits 0 to 5

Hand Written Digit	Features of Handwritten Digits (8 features)							
0	0.526316	0.526316	0.526316	0.526316	0	1.105263	1	0
1	0	0	0	0	0	1.842105	0	0
2	0	0	0	0	0	1.157895	0.631579	0
3	0	0	0	0	0	1.842105	1.315789	0
4	0	0.052632	0	0	0	1	1.526316	0
5	0	0	0	0	0	0	1	0.052632

Table 2: Features of Handwritten Digits 0 to 5

Hand Written Digit	Features of Handwritten Digits (8 features)							
0	0	1.421053	1.105263	0	0	0	0	0
1	0	0.842105	0	0	0	0	0	0
2	0	1.684211	1.157895	0.105263	0	0	0	0
3	0	0.894737	0.842105	0	0	0	0	0
4	0	0.631579	2	0.105263	0	0	0	0
5	0	0	1.210526	0	0	0.105263	0.421053	0

Table 3: Target vector for Handwritten digits 0 to 5

Node / Digit	Output node1	Output node2	Output node3	Output node4	Output node5	Output node6
0	1	0	0	0	0	0
1	0	1	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1

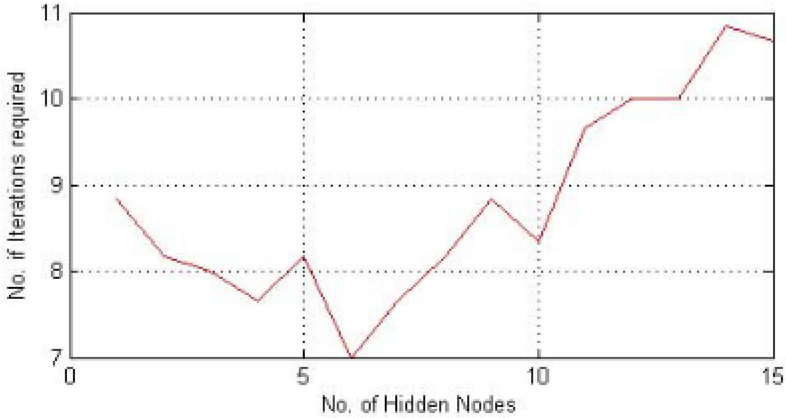

Figure 3: No.of iterations required for No. of hidden nodes

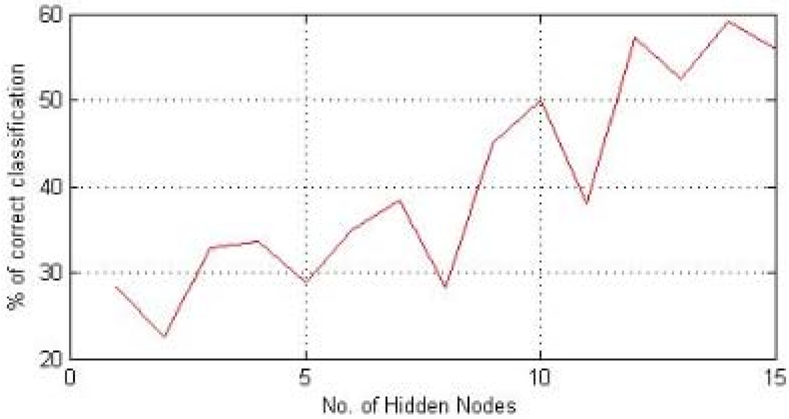

Figure 4: Percentage of Correct Classification for No. Of Hidden Nodes

Conclusion

The RBF Neural Network using Kalman filter training method is proposed for recognition of handwritten digits of different fonts. The proposed method is tested on handwritten digits of 0 to 9 of 25 different fonts. It is found that this method requires less time for traing the RBF network and has good success rate in recognizing the handwritten digits. This method can be extended to recognize handwritten characters also.

Acknowledgment

The authors place on record their thanks to the authorities of RVR&JC College of Engineering, Guntur, A.P., India for the facilities provided.

References

[1] Sittisak Rodtook and Yuttapong Rangsanseri, 2001, “Adaptive Thresholding of Document Images on Laplacian Sign”, IEEE Transactions on Pattern analysis and Machine Intelligence, 17(24).
[2] Jaekyu Ha & Robert M. Haralick Ihsin, T. Phillips, 1995, “Document Page Decomposition by the Bounding-Box Projection Technique”, IEEE Transactions on Systems, Man, and Cybernetics, 18(1), pp.1118-1122.
[3] S. Haykin, 1994, Neural Networks A Comprehensive Foundation, McMillan.
[4] O.Batsaikhan and Y.P. Singh, 2002, “Mongolian Character Recognition using Multilayer Perceptron”, Proceedings of the 9^{th} International Conference on Neural Information Processing, Vol.2.
[5] S. Chen, Y. Wu, B. Luk, 1999, “Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks”, IEEE Trans. Neural Networks pp. 1239–1243.
[6] M. Birgmeier, 1995, “A fully Kalman-trained radial basis function network for nonlinear modelling”, IEEE International Conference on Neural Networks, Perth, Western Australia, , Vol.11, pp. 259–264.
[7] G. Puskorius, L. Feldkamp, 1994“Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks”, IEEE Trans. Neural Networks, pp.5.
[8] Brijesh K. Verma, 2004, “Handwritten Hindi Character Recognition Using Multilayer Perceptron and RBF Neural Networks”, IEEE Trans. On Neural Networks, Vol.21, pp. 2111-2115.
[9] O.Batsaikhan and Y.P. Singh, 2002, “Mongolian Character Recognition using Multilayer Perceptron”, Proceedings of the 9^{th} International Conference on Neural Information Processing, Vol.2.
[10] Oivind Due Trier, Anil K.Jain and Torfinn Taxt, 1996, “Feature Extraction methods for Character recognition–A Survey”, Pattern Recognition, 29 (4), pp.641–662.
[11] P. S. P. Wang and Y. Y. Zhang, 1989, “A fast and flexible thinning algorithm”, IEEE Transactions on Computers, 38 (5).

[12] D.Y. Lee, 3, “Handwritten digit recognition using K nearest neighbour, radial basis function and backpropogation neural networks”, *Neural Computation*, pp. 440- 449.