

A New Learning Algorithm for Single Hidden Layer Feedforward Neural Networks

Virendra P. Vishwakarma
Department of CSE

Jaypee Institute of Information Technology,
Sector 128, Noida-201304 (U.P.) India.

M. N. Gupta
Department of CSE

Amity School of Engineering Technology
(Guru Gobind Singh Indraprastha University),
New Delhi-110061, India

ABSTRACT

For high dimensional pattern recognition problems, the learning speed of gradient based training algorithms (back-propagation) is generally very slow. Local minimum, improper learning rate and over-fitting are some of the other issues. Extreme learning machine was proposed as a non-iterative learning algorithm for single-hidden layer feed forward neural network (SLFN) to overcome these issues. The input weight and biases are chosen randomly in ELM which makes the classification system of non-deterministic behavior. In this paper, a new learning algorithm is presented in which the input weights and the hidden layer biases of SLFN are assigned from basis vectors generated by training space. The output weights and biases are decided through simple generalized inverse operation on output matrix of hidden layer. This makes very fast learning speed and better generalization performance in comparison to conventional learning algorithm as well as ELM.

General Terms

Artificial Neural Network, Pattern Recognition

Keywords

Face recognition, non-iterative learning algorithms, SLFN.

1. INTRODUCTION

The robustness and success of machine based automated face recognition system severely degrade under uncontrolled environment constraints, like illumination, pose, expression, age, disguise, facial hair, glasses and background [1]-[3]. Artificial neural network (ANN) is a powerful tool to resolve the nonlinearity imposed by different constraints [4], [5]. ANN based classifiers can integrate both structural and statistical information and achieve better performance than that of minimum distance classifiers [2]. There are different variants of ANN based classifiers which have been used for face recognition [6]-[12]. For training of ANN, it mostly employs backpropagation (BP) algorithm. Using BP algorithm, the face recognition system can learn effectively on small training set. Since BP is gradient descent based learning, it leads to slow convergence during the training for large scale training set. In [6], the face patterns were divided into several small-scale sub networks based on fuzzy c-means clustering rule. This is done in order to reduce the size of large scale training set [6].

An ensemble of classifiers is gaining more attention in face recognition as these provide remarkable improvement over the single classifier system. By assembling of multiple classifiers,

the deficiencies of each classifier may be compensated for the efficiency of others. Haddadnia and Ahmadi used a hybrid N-feature neural network for this purpose, which extracts a set of different kind of features from face images with radial basis function networks. These are combined together for classification purpose through the majority rule. They have used three different feature domains for features extraction from input images [7]. Similarly in [8], in place of a single type of feature extractor and classifier, localized random facial features were constructed using internally randomized networks. The ensemble classifier was finally constructed by combining the multiple networks via a sum rule [8].

A hybrid neural network system for face recognition has been presented by Lawrence et al. which combine local image sampling, a self organizing map neural network and a convolutional neural network [9]. Similarly two variants of convolutional neural networks: neocognitron and NEO were used for face recognition in [10]. Intrator et al. implemented a face recognition system using hybrid neural network based on supervised and unsupervised learning method [11]. BP network was integrated with fuzzy based feature extraction to utilize the feature-wise degree of belonging of patterns to all classes by Ghogh et al. [12].

In all the above variants of neural network implementations, the network parameters are iteratively tuned to achieve the required performance index. The learning speed of this type of training algorithms is in general very slow. Most of these networks employ BP algorithm for training which is gradient descent based learning. It leads to slow convergence during the training of the networks. Moreover, it suffers from some other issues like presence of local minima, imprecise learning rate, over fitting, and selection of number of hidden layer neurons.

Huang et al. proposed a non-iterative learning algorithm named extreme learning machine (ELM) to train the SLFN [13]. The input weights and hidden layer neuron biases were arbitrarily (small random numbers) assigned. Though this makes fast learning speed, but the recognition rate varies with some standard deviation value. This makes the classification system nondeterministic. As the face recognition problem is highly non-linear and non-convex, ELM does not provide desirable performance, if the number of hidden layer neurons is small. To achieve less error rate, we require large hidden layer neurons for ELM. It also requires manual tuning for the number of hidden layer neurons to achieve faster training.

To overcome these problems, we propose a new learning algorithm for SLFN, in which the input weights and biases are assigned from approximate basis vectors of input training space. The output weights and biases are decided through inverse operation on output matrix of hidden layer. Our learning algorithm provides not only better generalization performance but also faster learning rate. The rest of the paper is organized as follows. The description of SLFN along with BP is presented in Section 2. Section 3 describes the proposed learning algorithm. In Section 4, the experimental results and discussions on the data set are presented. Finally, conclusions are drawn in Section 5.

2. SLFN and BACKPROPAGATION

2.1 Single hidden Layer Feedforward Neural Networks (SLFN)

An ANN is a computational structure inspired by the study of biological neural processing. Different types of neural networks, from relatively simple to very complex, are found in literature [14], [15]. It was shown by Tamura and Tateishi [16] that the feedforward neural network with single-hidden layer containing S neurons with arbitrarily chosen input weights can learn S distinct observations with arbitrarily small error. Fig.1 shows the architecture of single-hidden layer feedforward neural network (SLFN) with S hidden neurons. In Fig.1(a), interconnection

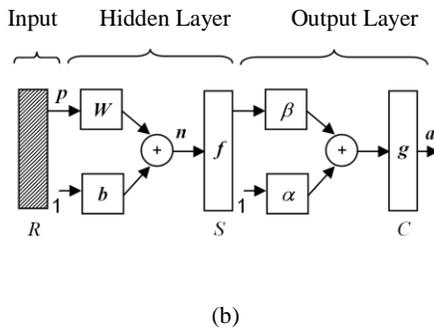
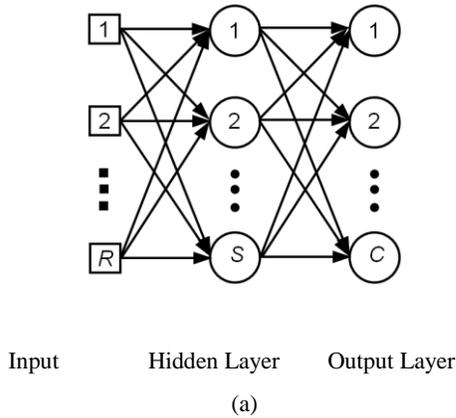


Fig. 1. Architecture of single hidden layer feedforward neural network (a) schematic of interconnection; (b) abbreviated notation.

architecture of SLFN has been shown. This network has been depicted using the abbreviated notations defined by Hagan et al. [15] in Fig.1(b). The elements of this architecture have been further explained in the following subsections, where these are used.

2.2 Gradient Descent based Learning Algorithm (BP)

In supervised learning, BP is mostly used to train SLFN. BP algorithm is a stochastic algorithm based on the steepest decent principle, wherein the weights of the neural network are updated along the negative gradient direction in the weight space. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly i.e. the negative of the gradient [15]. The algorithm trains with a set of examples which represent proper network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\} \quad (1)$$

Here $p_q = [p_{q1}, p_{q2}, \dots, p_{qR}]^T \in \mathfrak{R}^R$ is an input to the network, and $t_q = [t_{q1}, t_{q2}, \dots, t_{qM}]^T \in \mathfrak{R}^M$ is the corresponding target. In appearance-based approach of face recognition, the input patterns are generated from training face images. All pixel values are read line by line to form an input pattern which is a column vector in \mathfrak{R}^R dimensional real space. Here R is the total number of pixels or extracted features from the face image. p_{qr} is the intensity value of r^{th} pixel or r^{th} feature value in the q^{th} face image.

The algorithm adjusts the network weights and biases in order to minimize the approximate mean square error [15], where the approximate mean square error at k^{th} iteration is given by:

$$\mathcal{J}(k) = E[e^T(k)e(k)] = (t(k) - a(k))^T(t(k) - a(k)) \quad (2)$$

where $E[\]$ is used to denote the expected value. Then the network parameters are updated. The input weights and biases for the next iteration are given as

$$W(k+1) = W(k) - \eta \frac{\partial \mathcal{J}(k)}{\partial W}, \text{ and } b(k+1) = b(k) - \eta \frac{\partial \mathcal{J}(k)}{\partial b} \quad (3)$$

Similarly, the output weight and biases are updated as:

$$\beta(k+1) = \beta(k) - \eta \frac{\partial \mathcal{J}(k)}{\partial \beta}, \text{ and } \alpha(k+1) = \alpha(k) - \eta \frac{\partial \mathcal{J}(k)}{\partial \alpha} \quad (4)$$

Here η is the learning rate. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. There are different variants of BP algorithm. With standard steepest descent BP, the learning rate is held constant throughout training. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

The performance of the steepest descent BP algorithm can be improved if we allow the learning rate to change during the training process, which is used in adaptive learning [15]. In present investigation, to find the error rate of face recognition

system with SLFN trained with BP algorithm, we train the SLFN with adaptive learning rate along with momentum training.

3. PROPOSED LEARNING ALGORITHM FOR SLFN

We have proposed a new learning algorithm for SLFN, in which the network parameters are analytically determined and these are not iteratively tuned. The input weights and biases are assigned by approximate basis vectors of input space. The output weights and biases are decided through simple generalized inverse operation on output matrix of hidden layer. This not only provides faster training, but also reduces the error rate (thus better generalization performance). As the input weights and biases of SLFN are assigned from approximate *basis vectors* and *output weights* are derived by *inverse* operation, we name our learning algorithm as SLFN_BVOI. This is further explained in the following subsections.

3.1 Input weights and biases

The input weight matrix W can be expressed as:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1R} \\ w_{21} & w_{22} & \dots & w_{2R} \\ \vdots & \vdots & & \vdots \\ w_{S1} & w_{S2} & \dots & w_{SR} \end{bmatrix} \quad (5)$$

This matrix is of $S \times R$ dimension corresponding to S neurons in hidden layer and R input elements (i.e. R -dimensional input) (Fig.1(b)). A vector composed of the elements of i^{th} row of input weight matrix W is defined as:

$$w_i = [w_{i1}, w_{i2}, \dots, w_{iR}]^T \quad (6)$$

where w_{ir} is the weight of the connection between i^{th} neuron in hidden layer and r^{th} input. The input weight matrix W can be rewritten as $W = [w_1^T, w_2^T, \dots, w_i^T, \dots, w_S^T]^T$.

If n_q is defined as net input to hidden layer activation function, it may be treated as the target corresponding to the input pattern p_q under supervised Hebb learning rule for single layer networks. Based on this learning rule [15], the augmented input weights X can be obtained for Q input patterns as:

$$X = \sum_{q=1}^Q n_q z_q^T = NZ^T \quad (7)$$

where $N = [n_1, n_2, \dots, n_q, \dots, n_Q]$, and $Z = [z_1, z_2, \dots, z_q, \dots, z_Q]$. The dimension of N is $S \times Q$ and that of Z is $(R + 1) \times Q$. z_q is the augmented input which includes the input pattern p_q and unit input corresponding to the bias as: $z_q = [p_q, 1]^T$. Similarly, the augmented input weights include the input weights along with the hidden layer neurons biases, i.e. $x_i = [w_i, b_i]^T$, where w_i is the weight vector for i^{th} neuron of the hidden layer given by Equation (6) and b_i is its bias which is an element of \mathfrak{R} . The net input to the i^{th} neuron of the hidden layer can be expressed as:

$$n_i = w_i^T \cdot p_q + b_i = x_i^T \cdot z_q \quad (8)$$

$w_i^T \cdot p_q$ represents the inner product of w_i and p_q .

It can be shown that the elements of weight matrix will be bounded, if the input patterns are orthonormal [15]. But in

general, the input patterns are not orthonormal and it is required to restrict the large values of input weight elements. This is done by different normalization techniques in many learning algorithms [17]-[19]. It was shown by Bertlett [20] that the size of weights is more important than the size of network. He proved that the feedforward neural networks achieve smaller training error and better generalization performance, if the norm of weights is smaller [20]. In present learning algorithm, it is done by dividing the input matrix by a variable Γ , which is defined as:

$$\Gamma = (\sum (P^T \cdot P)^2 / Q)^{1/2} / c = \|P^T \cdot P\| / (cQ^{1/2}) \quad (9)$$

where $P = [p_1 \ p_2 \ \dots \ p_q \ \dots \ p_Q]$ is the input matrix and input vector $p_q \in \mathfrak{R}^R$ represents the q^{th} face image of training set. c is a positive constant and $\|\chi\|$ denotes the norm of χ .

Now the input weight matrix can be obtained by:

$$W = P^T / \Gamma \quad (10)$$

The size of W is $Q \times R$ as it is governed by input matrix P . The bias of i^{th} hidden layer neuron is obtained as:

$$b_i = \sum_{j=1}^R w_{ij} = E[w_i] \quad (11)$$

This variable Γ is inversely proportional to the norm of square of input matrix. The elements of resultant matrix (W) by Equation (10) are very small numbers and each row of this matrix may be treated as approximate basis vectors of input space. This results in very small norm of the weight matrices and thus the present learning algorithm provides better generalization performance which has been found in experimental results.

3.2 Output weights and biases

Let us consider f be a real valued function $f: \mathfrak{R} \rightarrow \mathfrak{R}$ as activation function of hidden layer; so that $f(w_i^T \cdot p_q + b_i)$ be the output of the i^{th} neuron in hidden layer. The bias b_i is the element of \mathfrak{R} . The output of the SLFN with Q number of hidden layer neurons, is given by

$$\sum_{i=1}^Q \beta_i f(w_i^T \cdot p_j + b_i) = a_j, \forall j = 1, 2, \dots, Q \quad (12)$$

where β_i is the weight vector connecting the i^{th} neuron of the hidden layer and the output neurons:

$$\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{it}, \dots, \beta_{ic}]^T \quad (13)$$

β_{it} is the weight of the connection between i^{th} neuron of the hidden layer and t^{th} neuron of the output layer.

The activation function g (Fig.1(b)) of the output layer neurons is chosen linear. No bias has been taken for the output layer neurons, i.e. $\alpha = 0$, has been chosen for present learning algorithm.

The SLFN can approximate the Q input samples represented by Equation (1) with zero error, if there exist β_i, w_i and b_i such that

$$\sum_{i=1}^Q \beta_i f(\mathbf{w}_i^T \cdot \mathbf{p}_j + b_i) = t_j, \forall j = 1, 2, \dots, Q \quad (14)$$

The above set of Q equations can be rewritten compactly as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \quad (15)$$

where

$$\mathbf{H} = \begin{bmatrix} f(\mathbf{w}_1^T \cdot \mathbf{p}_1 + b_1) & \dots & f(\mathbf{w}_Q^T \cdot \mathbf{p}_1 + b_Q) \\ \vdots & & \vdots \\ f(\mathbf{w}_1^T \cdot \mathbf{p}_Q + b_1) & \dots & f(\mathbf{w}_Q^T \cdot \mathbf{p}_Q + b_Q) \end{bmatrix}, \quad (16)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_Q^T \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_Q^T \end{bmatrix} \quad (17)$$

If the activation function of hidden layer of SLFN is infinitely differentiable, for fixed input weight and bias of hidden layer neurons, the least-squares solution $\hat{\boldsymbol{\beta}}$ of the linear system given by Equation (15) can be obtained with minimum norm of output weight $\boldsymbol{\beta}$. With this, the smallest training error may be reached by the solution $\hat{\boldsymbol{\beta}}$ which is given by:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^* \cdot \mathbf{T} \quad (18)$$

where \mathbf{H}^* is generalized inverse [21] of matrix \mathbf{H} . Although \mathbf{H} is a square matrix of dimension $Q \times Q$, to avoid the problem of 'close to singular' (very small determinant), generalized inverse has been taken. The generalized inverse is also called as Moore-Penrose inverse or pseudo-inverse [21].

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

For evaluating the proposed learning algorithm, the experiments have been performed on AT&T [24], [25] and Yale [26] face databases. To establish the improvement in generalization capability and learning speed of the present algorithm (SLFN_BVOI), these parameters of SLFN_BVOI have been compared with those of BP and ELM [21]-[23], [27] learning algorithms. The error rate variations for SLFN_BVOI learning algorithm along with BP and ELM have been evaluated and compared on these databases to show the generalization capability of SLFN_BVOI algorithm. The training time has been measured for these algorithms to establish the improvement in the learning speed of the present algorithm. The experiments have been performed on a laptop pc with 1.73 GHz core 2, duo processor using MATLAB 7.0.1.

4.1 Databases Used

4.1.1 AT&T face database

This database comprises ten different images of each of forty distinct subjects. The images were taken with varying illumination, pose, expression and facial details (glasses / no glasses) and at different times. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The images are stored as a 112×92 pixel array with 256 grey levels [24], [25].

4.1.2 Yale face database

There are 165 gray scale images of 15 subjects in GIF format in this database [26]. Original image size is 320×243 pixels. All images are manually cropped to include all internal structures of a face (forehead with hairs, eye brow, eyes, nose, and mouth). After cropping the image size becomes 220×175 pixels. These images are further sub-sampled by 1.6 to get a resolution of 137×109 pixels. The main challenge on this database is facial expressions (normal, happy, sad, sleepy, surprised, and wink), occlusion (with/without glasses) and misalignment along with illumination variations.

The experiments have been performed with different size of training set. The size of training set and test set is varying based on the number of images per subject, used for training. For example, if we take one image per subject for training, the training set size is 40 for AT & T face database. Similarly for two images per subject; the training set size is 80 and so on. The remaining images of the database are used for testing. The images are taken sequentially from database to build training set and test set, i.e. if number of images per subject for training is four, then the first four images per subject are used in training set and remaining six images are used for testing. The images of the database which have been used for training are not used for testing. For Yale face database, when one image per subject is used for training, the training set and test set will contain 15 and 150 images respectively, as there are 165 images of 15 subjects in this database. Similarly for two images per subject; the training set and test set size is 30 and 135 respectively, and so on.

4.2 Experimentation on AT&T face database

Fig. 2 shows the percentage error rate variation with respect to c used in Equation (12) for four different size of training set on AT&T face database. The size of training set is varying based on the number of images per subject used for training, as specified earlier. The percentage error rate decreases marginally for initial values of c (1 to 11). For larger values of c (greater than 12), it increases rapidly for the training set containing one or two images per subject. As seen from this figure (Fig. 2), the optimal value of c is found to be 10, which has been used for further analyses on this database.

4.2.1 Error rate on AT&T face database

The percentage error rate variation on AT&T face database for present learning algorithm (SLFN_BVOI) as well as for ELM and BP with respect to training set size has been shown in Fig. 3. Table 1 lists the percentage error rate for these training algorithms. The percentage reduction in error rate by SLFN_BVOI in comparison to ELM and BP are also listed here. The percentage error rate decreases as the number of images per subject for training, increases. There is significant reduction in percentage error rates, when the network is trained using SLFN_BVOI for all size of training set over percentage error rates for other two algorithms.

The experimental results for BP (adaptive learning with momentum) algorithm are obtained using `traingdx` function of MATLAB. This variant of BP algorithm has been used in place of some other faster BP variants like `trainlm` (Levenberg-Marquardt BP) due to high dimensional training set (Out of memory error is generated when using `trainlm`). The number of

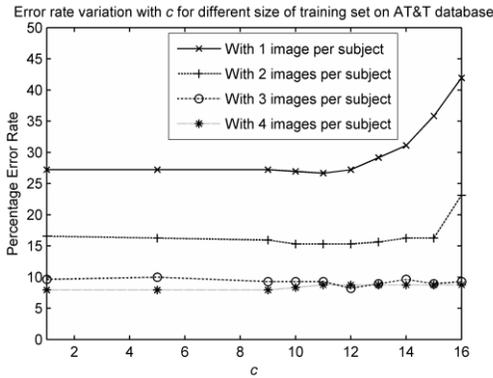


Fig. 2. Percentage error rate variation with parameter c (Equation12) of SLFN_BVOI learning algorithm, for four different size of training set on AT&T face database.

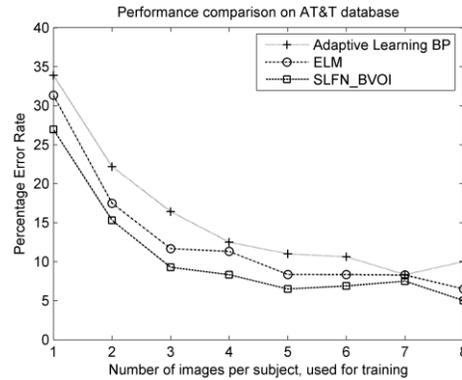


Fig. 3. Comparison of percentage error rate for BP, ELM and SLFN_BVOI (proposed algorithm) learning algorithms on AT&T face database.

Table 1. Comparison of percentage error rate on AT&T face database (percentage error rates using BP, ELM and SLFN_BVOI training algorithms and percentage reduction in error rate by SLFN_BVOI in comparison to BP and ELM)

% Error Rate & % Reduction SLFN Training Algorithm	Number of images per subject used for training							
	1	2	3	4	5	6	7	8
% error rate using adaptive learning BP	33.89	22.19	16.43	12.50	11.00	10.63	8.33	10.00
% error rate using ELM	31.03	17.50	11.32	10.67	8.65	8.38	8.17	6.00
% error rate using SLFN_BVOI	26.94	15.31	9.29	8.33	6.50	6.88	7.50	5.00
% reduction in error rate by SLFN_BVOI in comparison to BP	20.49	30.99	43.48	33.33	40.91	35.29	10.00	50.00
% reduction in error rate by SLFN_BVOI in comparison to ELM	13.16	12.50	17.98	21.88	24.86	17.91	8.16	16.67

Table 2. Comparison of training time (in seconds) on AT&T face database (training time using BP, ELM and SLFN_BVOI training algorithms and ratio of training times)

Training Time (Sec.) & Ratio SLFN Training Algorithm	Number of images per subject used for training							
	1	2	3	4	5	6	7	8
Training time using adaptive learning BP	185.6	284.4	326.6	445.1	723.7	580.6	545.3	602.3
Training time using ELM	2.17	3.28	4.74	6.46	6.91	7.62	9.90	11.16
Training time using SLFN_BVOI	0.094	0.219	0.516	0.875	1.109	1.656	2.203	2.984
Ratio of training time of BP with that of SLFN_BVOI	1979.2	1300.3	633.4	508.7	652.4	350.5	247.5	201.8
Ratio of training time of ELM with that of SLFN_BVOI	23.10	14.99	9.19	7.39	6.23	4.60	4.49	3.74

neurons in hidden layer has been taken 70 with the experimentation using BP.

In ELM implementation, the activation function for the hidden layer neurons has been chosen *sigmoid*, for which minimum error rate has been obtained compared to other activation functions. The deviation in percentage error rate is larger for same training set size, when the number of hidden layer neurons for ELM is lesser. For experimentation with ELM, the percentage error rates have been obtained with 1500 hidden layer neurons. It has been found that the standard deviation in error rate with this number of hidden layer neurons varies from 1.09% to 1.94%. Also, similar deviation exists in learning speed of ELM. The results for ELM shown in Fig. 3 and listed in Table 1, are averaged over 20 repeated experiments.

In the proposed learning algorithm (SLFN_BVOI), the number of hidden layer neurons is assigned from the training set size. For example, if only one image per subject is used for training, then the number of training face images will be 40 for AT&T face database and that is the number of hidden layer neurons in SLFN_BVOI. The maximum number of hidden layer neurons is 320 for present investigation of SLFN_BVOI corresponding to 8 images per subject used for training. The activation function for the neurons in hidden layer in SLFN_BVOI learning algorithm is chosen as *radial basis (radbas)* function.

When percentage error rate of SLFN_BVOI is compared with that of BP on AT&T face database, the percentage reduction in error rate varies from minimum 10% to maximum 50% as seen from Table 1. Similarly, this reduction varies from 8.16% to 24.86% when SLFN_BVOI is compared with ELM. The average reduction in error rate is 33.06% and 16.64% when the performance of SLFN_BVOI learning algorithm is compared with that of BP and ELM respectively.

4.2.2 Learning speed on AT&T face database

SLFN_BVOI provides very fast learning in comparison to BP as well as ELM because the number of hidden layer neurons is less in SLFN_BVOI learning algorithm in comparison to ELM and there is no iterative tuning of network parameters. The comparison of training time on AT&T database is listed in Table 2. The ratio of training time of BP to that of SLFN_BVOI varies from 201.8 to 1979.2 with mean value as 734.2. Similarly, the ratio of training time of ELM to that of SLFN_BVOI varies from 3.7 to 23.1 with mean value 9.2 on this database.

4.3 Experimentation on Yale face database

Illumination normalization has been applied on Yale face database images before performing experiments. For illumination normalization, the database images have been processed using the technique developed by the authors [28], [29]. After taking discrete cosine transform (DCT) on the images of this database, only first six low-frequency coefficients are divided by a constant 50, as there are small illumination variations. Then inverse DCT has been taken to obtain the illumination normalized face images. The value of constant *c* in Equation (12) has been taken as 12.8 for Yale face database. This value has been obtained in the similar way as done for AT&T database.

4.3.1 Error rate on Yale face database

Fig.4 and Table 3 show the percentage error rate variation on Yale face database for SLFN_BVOI, ELM and BP algorithms

with respect to varying training set size. Same experimental setup has been used for these algorithms as done with the experimentation on AT&T face database. The number of neurons in hidden layer in SLFN_BVOI algorithm is different as this value is governed by the training set size of the database. In the analysis on Yale face database, the number of hidden layer neurons is 15 in SLFN_BVOI, when only one image per subject has been used for training. The maximum number of hidden layer neurons is 135 in this algorithm corresponding to 9 images per subject used for training. With 1500 number of hidden layer neurons, the standard deviation in error rate varies from 0.81% up to 2.13%, corresponding to training set for one image up to nine images respectively.

Significant reduction in error rate has been achieved by SLFN_BVOI learning algorithm on Yale face database. When percentage error rate of SLFN_BVOI is compared with that of BP, the percentage reduction in error rate varies from minimum 11.48% to maximum 100% as seen from Table 3. This reduction varies from 3.23% to 100% for comparison of SLFN_BVOI with ELM. For 7,8 and 9 images per subject used for training, SLFN_BVOI learning algorithm provides zero % error rate on Yale face database (all test images are correctly recognized); and these cases correspond to 100% reduction in error rate with respect to other two learning algorithms. The mean value of reduction in error rate is 58.5% and 48.1% when the performance of SLFN_BVOI learning algorithm is compared with that of BP and ELM respectively.

4.3.2 Learning speed on Yale face database

Just like AT&T face database, on Yale face database also, fast learning speed has been provided by SLFN_BVOI algorithm. As Yale face database is smaller in comparison to AT&T database, the less number of hidden layer neurons is required in present learning algorithm. SLFN_BVOI algorithm provides faster learning speed on Yale face database, in comparison to AT&T database. Table 4 shows the comparison of training time of SLFN_BVOI, BP and ELM training algorithms on Yale face database. The ratio of training time of BP to that of

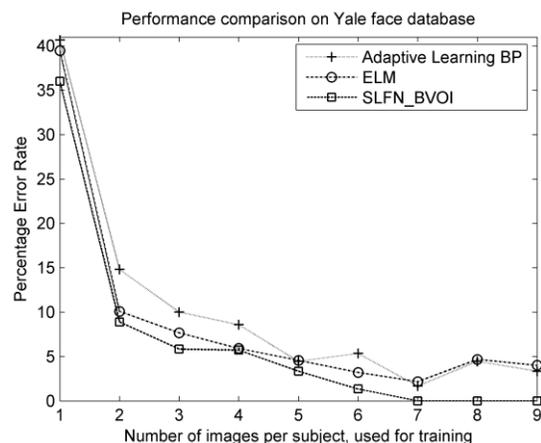


Fig. 5. Comparison of percentage error rate for BP, ELM and SLFN_BVOI (proposed algorithm) learning algorithms on Yale face database.

Table 3. Comparison of percentage error rate on Yale face database (percentage error rates using BP, ELM and SLFN_BVOI training algorithms and percentage reduction in error rate by SLFN_BVOI in comparison to BP and ELM)

% Error Rate & % Reduction SLFN Training Algorithm	Number of images per subject used for training								
	1	2	3	4	5	6	7	8	9
% error rate using adaptive learning BP	40.67	14.81	10.00	8.57	4.44	5.33	1.67	4.4 4	3.3 3
% error rate using ELM	39.47	10.07	7.67	5.90	4.56	3.20	2.17	4.6 7	4.0 0
% error rate using SLFN_BVOI	36.00	8.89	5.83	5.71	3.33	1.33	0	0	0
% reduction in error rate by SLFN_BVOI in comparison to BP	11.48	40.00	41.67	33.33	25.00	75.00	100.0	10 0.0	10 0.0
% reduction in error rate by SLFN_BVOI in comparison to ELM	8.78	11.76	23.91	3.23	26.83	58.33	100.0	100.0	100.0

Table 4. Comparison of training time (in seconds) on Yale face database (training time using BP, ELM and SLFN_BVOI training algorithms and ratio of training times)

Training Time (Sec.) & Ratio SLFN Training Algorithm	Number of images per subject used for training								
	1	2	3	4	5	6	7	8	9
Training time using adaptive learning BP	80.5	89.7	105.9	117.3	123.5	131.9	143.3	15 8.5	16 5.7
Training time using ELM	1.65	2.08	2.53	3.07	3.55	4.09	4.71	5.1 1	5.6 2
Training time using SLFN_BVOI	0.047	0.094	0.141	0.203	0.266	0.344	0.469	0.5 63	0.7 34
Ratio of training time of BP with that of SLFN_BVOI	1717.7	957.0	752.8	577.2	464.8	383.8	305.7	28 1.7	22 5.7
Ratio of training time of ELM with that of SLFN_BVOI	35.2	22.2	18.0	15.1	13.4	11.9	10.1	9.1	7.7

SLFN_BVOI varies from 225.7 to 1717.7 with mean value as 629.6 on this database. Similarly, the ratio of training time of ELM to that of SLFN_BVOI varies from 7.7 to 35.2 with mean value as 15.8.

5. CONCLUSION

A new learning algorithm named SLFN_BVOI for training of single-hidden layer feedforward neural network (SLFN) has been developed in this paper. SLFN_BVOI performs effectively on high dimension and high variations problems of face recognition. It has been evaluated on two databases, AT&T and Yale face databases. Significant performance improvement has been achieved by SLFN_BVOI learning algorithm over existing state of art learning algorithms. Unlike the traditional learning algorithms (BP), in SLFN_BVOI learning algorithm, the network parameters are not iteratively tuned but these are analytically determined. The percentage error rate for one size of training set is same and is not varying under different execution of training as occurs in ELM. Against non-deterministic performance of ELM, SLFN_BVOI provides deterministic performance.

No manual tuning of number of neurons in hidden layer is required in the developed learning algorithm (SLFN_BVOI).

The norms of weights are small hence the generalization performance of present learning algorithm is very good. The average reduction in error rate as 33.06% and 16.64% has been achieved by present learning algorithm in comparison to BP and ELM respectively on AT&T face database. This reduction in error rate is 58.5% and 48.1% when the performance of present learning algorithm is compared with that of BP and ELM respectively on Yale face database.

SLFN_BVOI algorithm learns on an average 734.2 times faster than BP and 9.2 times faster than ELM for different sizes of training set on AT&T face database. Similarly on Yale face database, it learns on an average 629.6 times faster than BP and 15.8 times faster than ELM.

6. REFERENCES

- [1] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," in *Proc. IEEE*, vol. 83, no. 5, pp. 705–740, May 1995.
- [2] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey", *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, Dec 2003.
- [3] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, Springer, 2005.

- [4] V. P. Vishwakarma, S. Pandey, and M. N. Gupta, "Illumination Normalization under Varying Illumination Conditions using Artificial Neural Network", in *Proc. 3rd International Conference on Advance Computing & Communication Technologies*, Nov. 2008, pp. 455-460.
- [5] V. P. Vishwakarma, S. Pandey, and M. N. Gupta, "Pose Invariant Face Recognition using Virtual Frontal View Generation," in *Proc. International Conference on Computing, Communication and Networking (ICCCN 2008)*, Dec. 2008, pp. 1-5.
- [6] L. Lu, X. Yuan, and T. Yahagi, "A method of face recognition based on fuzzy clustering and parallel neural networks," *Signal Processing*, vol. 86, no. 8, pp. 2026-2039, Aug. 2006.
- [7] J. Haddadnia and M. Ahmadi, "N-feature neural network human face recognition," *Image and Vision Computing*, vol. 22, no. 12, pp. 1071-1082, Oct. 2004.
- [8] K. Choi, K. A. Toh, and H. Byun, "A Random Network Ensemble for Face Recognition," *Lecture Notes in Computer Sciences*, vol. 5558, pp. 92-101, June 2009.
- [9] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 98-113, Jan. 1997.
- [10] C. Neubauer "Evaluation of Convolutional Neural Networks for Visual Recognition," *IEEE Trans. Neural Networks*, vol. 9, no. 4, pp. 685-696, July 1998.
- [11] N. Intrator, D. Reisfeld, and Y. Yeshurun, "Face Recognition using a Hybrid Supervised/Unsupervised Neural Network," *Pattern Recognition Letters*, vol. 17, no. 1, pp. 67-76, Jan. 1996.
- [12] A. Ghosh, B. U. Shankar and S.K. Meher, "A novel approach to neuro-fuzzy classification," *Neural Networks*, vol. 22, pp. 100-109, 2009.
- [13] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks," in *Proc. Int. Joint Conf. on Neural Networks (Budapest, Hungary)*, July 2004, pp. 985-990.
- [14] S. Haykin, *Neural Networks—A Comprehensive Foundation*, 2nd Ed. Prentice Hall, 1999.
- [15] M. T. Hagan, H.B. Demuth, and M. Beale, *Neural Network Design*, Thomson Learning, 2002.
- [16] S. Tamura and M. Tateishi, "Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three," *IEEE Trans. Neural Networks*, vol. 8, no. 2, pp. 251-255, Mar. 1997.
- [17] H. M. El-Bakry, "A Simple Design for High Speed Normalized Neural Networks Implemented in the Frequency Domain for Pattern Detection," in *Proc. International Joint Conference on Neural Networks, Vancouver, Canada*, July 2006, pp. 1317-1324.
- [18] T. Kwok and K. A. Smith, "A Noisy Self-Organizing Neural Network With Bifurcation Dynamics for Combinatorial Optimization," *IEEE Trans. Neural Networks*, vol. 15, no. 1, pp. 84-98, Jan. 2004.
- [19] L. Wang, "On Competitive Learning," *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 1214-1217, Sep. 1997.
- [20] P. L. Bartlett, "The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network," *IEEE Trans. Information Theory*, vol. 44, no. 2, pp. 525-536, Mar. 1998.
- [21] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing*, vol. 70, pp. 489-501, 2006.
- [22] M. B. Li and M. J. Er, "Nonlinear System Identification Using Extreme Learning Machine," in *Proc. IEEE Int. Conf. on Control, Automation, Robotics and Vision*, Dec. 2006, pp. 1-4.
- [23] Q. Y. Zhu, G. B. Huang, and C. K. Siew, "A Fast Constructive Learning Algorithm For Single Hidden Layer Neural Networks," in *Proc. IEEE Int. Conf. on Control, Automation, Robotics and Vision*, Dec. 2004, pp. 1907-1911.
- [24] F. Samaria and A. Harter, "Parameterization of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota, Dec. 1994, pp. 138-142.
- [25] AT&T face database, [Online]. Available: http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.tar.Z.
- [26] Yale face database, [Online]. Available: <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.
- [27] V. P. Vishwakarma, S. Pandey and M. N. Gupta, "Face Recognition using Extreme Learning Machine," in *Proc. International Conference on Quality, Reliability and Infocom Technology, Delhi University, India*, Dec. 2009.
- [28] V. P. Vishwakarma, S. Pandey and M. N. Gupta, "A Novel Approach for Face Recognition Using DCT Coefficients Re-scaling for Illumination Normalization," in *Proc. 15th IEEE Conference on Advanced Computing & Communication (ADCOM 2007)*, pp.535-539, Dec. 2007.
- [29] V. P. Vishwakarma, S. Pandey and M. N. Gupta, "An Illumination Invariant Accurate Face Recognition with Down Scaling of DCT Coefficients," *Journal of Computing and Information Technology*, vol. 18, no. 1, pp. 53-67, Mar. 2010.