EAST: An Exponential Adaptive Skipping Training Algorithm for Multilayer Feedforward Neural Networks

R.MANJULA DEVI
Research scholar and Assistant Professor(Senior Grade)
Department of Computer Science and Engineering
Kongu Engineering College,
Perundurai – 638 052, Erode
INDIA
rmanjuladevi.gem@gmail.com
S.KUPPUSWAMI
Principal
Kongu Engineering College, Perundurai – 638 052, Erode
INDIA
skuppu@gmail.com

Abstract: - Multilayer Feedforward Neural Network (MFNN) has been administered widely for solving a wide range of supervised pattern recognition tasks. The major problem in the MFNN training phase is its long training time especially when it is trained on very huge training datasets. In this accordance, an enhanced training algorithm called Exponential Adaptive Skipping Training (EAST) Algorithm is proposed in this research paper which intensifies on reducing the training time of the MFNN through stochastic manifestation of training datasets. The stochastic manifestation is accomplished by partitioning the training dataset into two completely separate classes, classified and misclassified class, based on the comparison result of the calculated error measure with the threshold value. Only the input samples in the misclassified class are exhibited to the MFNN for training in the next epoch, whereas the correctly classified class is skipped exponentially which dynamically reducing the number of training input samples exhibited at every single epoch. Thus decreasing the size of the training dataset exponentially can reduce the total training time, thereby speeding up the training process. This EAST algorithm can be integrated with any supervised training algorithms and also it is very simple to implement. The evaluation of the proposed EAST algorithm is demonstrated effectively using the benchmark datasets - Iris, Waveform, Heart Disease and Breast Cancer for different learning rate. Simulation study proved that EAST training algorithm results in faster training than LAST and standard BPN algorithm.

Key-Words: - Adaptive Skipping, Neural Network, Training Algorithm, Training Speed, MFNN, Learning Rate

1 Introduction

Multilayer Feedforward Neural Network (MFNN) with a single hidden layer has been explored as the best neural network architecture for nonlinear classification problem due to its capability to approximate any nonlinear function mapping [1][2][3]. The Back Propagation (BPN) is the most popular supervised training algorithm that has been used to train MFNN extensively for the past two decades [4]. It is fragmented into two phases: Training Phase (also called as Learning Phase) and Testing Phase (also called as Evaluation Phase). Among these two phases, the training phase plays an important role in establishing nonlinear models. Still it requires many epochs to obtain better performance in training the MFNN for simple problem. So the BPN is unfortunately very slow. And also BPN training performance is literally associated with the type and size of network architecture, the number of epochs and patterns to be trained, training speed, and the dimensionality of the training datasets.

In order to enhance the training performance, the training speed is the factor that is considered to be very important. The training speed is highly depended on the dimensionality of training dataset. In general, training MFNN with a larger training datasets will generalize the network well. However, ample amount of training data normally requires very long training time [5] which affects the training speed. Much iteration is required to train small networks for even the simplest problems.

This research proposes a new training algorithm to improve the training speed by reducing the training time of MFNN through the stochastic manifestation of training datasets. Hence, the overall training time for actual training of the MFNN is often reduced by several hundred times than in the standard training algorithm. This algorithm can be incorporated into any kind of supervised algorithm.

The content of this research paper is materialized as follows. Section II gives the brief review of the previous works done relevant to the research problem. Section III shows the formulation of the given research problem. Section IV presents the proposed EAST algorithm. Performance evaluation of EAST is simulated in Section V using the benchmark datasets for the classification problems. In Section VI, the experimental results are summarized and analyzed. Finally, Section VII draws the conclusions of the research paper.

2 Related Works

In order to speed up the MFNN training process, many researchers have investigated the above detriments and devoted many of their research works through various formation ranges from different amendments of existing algorithms to evolution of new algorithms. The formation of such works includes initialization of optimal initial weight [6,7], adaptation of learning rate [8], adaptation of the momentum term [9], adaptation of the momentum term in parallel with learning rate adaptation [10], and using second order algorithm [11-13] in favor of speeding up the training process and maintaining generalization.

By estimating the proper initial value of the network's weight will reduce the number of epochs in the training process thereby speeding up the training process. Many weight initialization methods have been developed by the researchers. Nguyen and Widrow initialize the layer's intermediate weight within the specified range for faster learning [6]. Varnava and Meade used the polynomial mathematical models to obtain initial values of the network weights [7]. The learning rate is one of the training parameters that fine-tune the size of the network's respective old weights during learning. The constant learning rate secures the convergence but considerably slows down the training process. But, adaptation of learning rate using the Barzilai and Borwein is proposed by Plagianakos et al in order to improve the convergence speed [8]. Hence several methods based on heuristic factor have been proposed for changing the learning rate dynamically. Behera et al. applied convergence theorem based on Lyapunov stability theory for attaining the adaptive learning rate[10]. Last, Second order training algorithms employ the second order partial derivative information of the error function to perform network pruning. This algorithm is very apt for training the neural network that converges quickly. The most popular second order methods such as conjugate gradient (CG) methods, quasi-Newton (secant) methods or Levenberg–Marquardt (LM) method are considered popular choices for training neural network. Nevertheless, it is not certain that these methods are very computationally expensive and requires large memory particularly for large networks. Ampazis and Perantonis presented Levenberg–Marquardt with adaptive momentum (LMAM) and optimized Levenberg–Marquardt with adaptive momentum (OLMAM) second order algorithm that integrates the advantages of the LM and CG methods[11]. Wilamowski and Yu incorporated some modification in LM methods by rejecting Jacobian matrix storage and also replacing Jacobian matrix multiplication with the vector multiplication [12,13] which results in the reduction of memory cost for training very huge training dataset.

However, the disadvantages found in the traditional method are not surmounted by the above discussed techniques. All of the above mentioned efforts are focused directly or indirectly on tuning the network's training parameters. And also, each and every formation utilizes all the training input samples for classification at each and every single epoch. If a large amount of training data with high dimension is rendered for classification, then a problem is introduced by the above discussed technique which will slow down classification. So, the intention of this research is to impart a simple and new algorithm EAST for training the ANN in a fast manner by presenting the training input samples randomly based on the classification.

3 Problem Formulations

BPN algorithm is an iterative gradient training algorithm designed to estimate the coefficients of weight matrices that minimizes the total Root Mean Squared Error (RMSE). The RMSE is defined between the desired output and the actual output summed over all the training pattern input to the network.

$$RMSE = \frac{1}{P}\sum_{p=1}^{P} E^p \tag{1}$$

E^p is calculated using the following formula

$$E^p = \frac{1}{2}\sum_{k=1}^{m}(t_k^p - y_k^p)^2 \tag{2}$$

Where P is the total number of training sample patterns, m is the number of nodes in the output layer, t_k^p is the target output of the kth node for the pth sample pattern, and y_k^p is the actual output of the k^{th} node estimated by the network for the p^{th} sample pattern.

According to the Equation (2), there is a real fact that the correctly classified input samples does not involve in the updating of weight since the error value generated by that sample pattern is zero. Here the intention of this research is to partition the training input samples into two distinct classes, classified and misclassified class, based on the comparison result of the calculated error measure with the maximum threshold value. By doing so, the training input samples whose actual output is same as target output will belong to the classified class; the remaining training input samples will belong to the misclassified class. Only the input samples in the misclassified class are presented to the next epoch (Epoch is one complete cycle of populating the MFNN with the entire training samples once) for training, whereas the correctly classified class will not be presented again for the subsequent n epochs. In the LAST algorithm [14], the input samples are skipped linearly. Our adaptive skipping algorithm is used to determine the value of n i.e., the skipping factor. In the EAST algorithm, the correctly classified class input samples will be skipped exponentially from the training for the consecutive n epochs. Thereby, the EAST algorithm dynamically reducing the number of training input pattern samples exponentially exhibited at every single epoch. Thus decreasing the size of the training input samples exponentially can reduce the total training time, thereby speeding up the training process. The dominance of this EAST algorithm is that its implementation is extremely simple and easy, and can lead to significant advances in the training speed.

4 Proposed EAST Method

4.1 Overview of EAST Architecture

The EAST algorithm that is contained in the prototypical MFNN architecture is outlined below

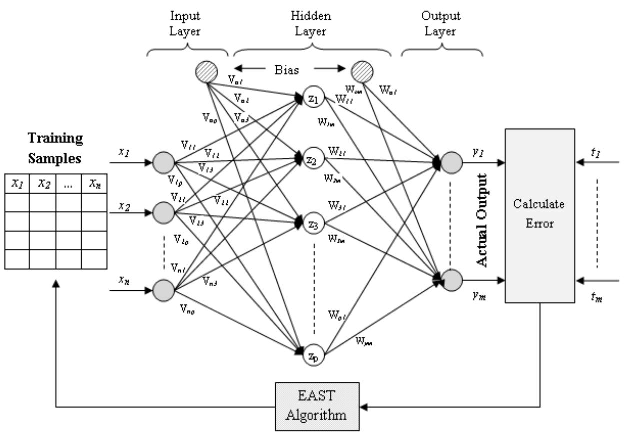

Figure 1: Architecture of MFNN with EAST algorithm

Assume that the network contains n input nodes in the input layer, p hidden nodes in the hidden layer and m output nodes in the output layer. Since the above network is highly interconnected, the nodes in each layer are connected with all the nodes in the next layer. Let P represent the number of input patterns in the training dataset. The input matrix, X, of size p × n is presented to the network. The number of nodes in the input layer is equivalent to the number of columns in the input matrix, X. Each row in X is considered to be a real-valued vector $x_i \in \Re^{n+1}$ where $1 \leq i \leq n$. The summed real-valued vector generated from the hidden layer is represented $z_i \in \Re^{p+1}$ where $1 \leq i \leq p$. The estimated output real-valued vector generated from the network is denoted as $y_i \in \Re^m$ where $1 \leq i \leq m$ and the corresponding target vector is represented as $t_i \in \Re^m$ where $1 \leq i \leq m$. Let it signifies the it^{th} iteration number.

Let $f_N(x)$ and $f_L(x)$ be the non-linear logistic activation function and linear activation function used for computation in the hidden and output layer respectively. Let v_{ij} be the $n \times p$ weight matrix contains input-to-hidden weight coefficient for the link from the input node i to the hidden node j and v_{oj} be the bias weight to the hidden node j. Let w_{jk} be the $p \times m$ weight matrix contains hidden-to-output weight coefficient for the link from the hidden node j to the output node k and w_{ok} be the bias weight to the output node k.

4.2 Proposed EAST Algorithm

The working principle of the EAST algorithm that is incorporated in the BPN algorithm is summarized below:

Step 1. Weight Initialization: Initialize weights to small random values;

Step 2. Furnish the input sample: Disseminate to the input layer an input sample vector $\mathbf{x}_k$ having desired output vector $\mathbf{y}_k$;

Step 3. Forward Phase: Starting from the first hidden layer and propagating towards the output layer:

a. **Calculate the activation values** for the Hidden layer as:

i. Estimate the net output value

$$z_{inj}(it) = v_{oj}(it) + \sum_{i=1}^{n} \mathrm{x_i(it).v_{ij}(it)} \qquad (3)$$

ii. Estimate the actual output

$$z_j(it) = \frac{1}{1 + e^{-z_{inj}}} \qquad (4)$$

b. **Calculate the activation values** for the Output layer as:

i. Estimate the net output value

$$y_{ink}(it) = w_{ok}(it) + \sum_{j=1}^{p} z_j(it).w_{jk}(it) \quad (5)$$

ii. Estimate the actual output

$$y_k(it) = \frac{1}{1 + e^{-y_{ink}}} \quad (6)$$

Step 4. Output errors: Calculate the error terms at the output layer as:

$$\delta_k(it) = [t_k - y_k(it)].f'(y_k(it)) \quad (7)$$

Differentiate the activation function in Equation 6,

$$f'(y_k(it)) = \frac{\partial(y_k(it))}{\partial x}$$
$$= y_k(it) \times (1 - y_k(it)) \quad (8)$$

Substitute the resultant value of Equation (8) in (7)

$$\delta_k(it) = y_k(it).[1 - y_k(it)].[t_k - y_k(it)] \ (9)$$

Step 5. Backward Phase: Propagate error backward to the input layer through the hidden layer using the error term

$$\ddot{a}_j(it) = \left[\sum_{k=1}^{m} \delta_j(it).w_{jk}(it)\right].f'\left(z_j(it)\right) \ (10)$$

Differentiate the activation function in Equation 4,

$$f'\left(z_j(it)\right) = \frac{\partial\left(z_j(it)\right)}{\partial x}$$
$$= z_j(it) \times \left(1 - z_j(it)\right) \quad (11)$$

Substitute the resultant value of Equation (11) in (10)

$$\delta_j(it) = \left[\sum_{k=1}^{m} \delta_j(it).w_{jk}(it)\right] z_j(it).[1 - z_j(it)] \quad (12)$$

Step 6. Weight Amendment: Update weights using the Delta-Learning Rule

a. **Weight amendment** for Output Unit

$$W_{jk}(it+1) = W_{jk}(it) + \alpha(it).\delta_k(it).z_j(it) \ (13)$$

b. **Weight amendment** for Hidden Unit

$$V_{ij}(it+1) = V_{ij}(it) + \alpha(it)\ \delta_j(it)\ x_i(it) \quad (14)$$

Step 7. EAST Algorithm: Incorporating the EAST algorithm

a. **Compare** the error value, $|t_k - y_k|$ with threshold value, d_{max}

$$|t_k - y_k(it)| < d_{max} \quad (15)$$

If equation 15 generates 0, then the x_i is correct

b. **Compute** the probability value for all input samples

$$prob(x^i) = \begin{cases} 0, \text{if } x_i \text{ is correct and epoch number} < n \\ 1, otherwise \end{cases} \quad (16)$$

c. **Calculate** the skipping factor, $sf_{i,}$ for all input samples

i. Initialize the value of sf_i to zero (for first epoch)

ii. Increment the value of sf_i exponentially for correctly classified samples alone.

d. **Skip** the training samples with prob (=0) for the next sf_i epoch

Step 8. Repeat steps 1-7 until the halting criterion is satisfied, which may be chosen as the Root Mean Square Error (RMSE), elapsed epochs and desired accuracy

4.3 Working Flow of EAST

The block diagram of the proposed strategy is illustrated in the Fig.2.

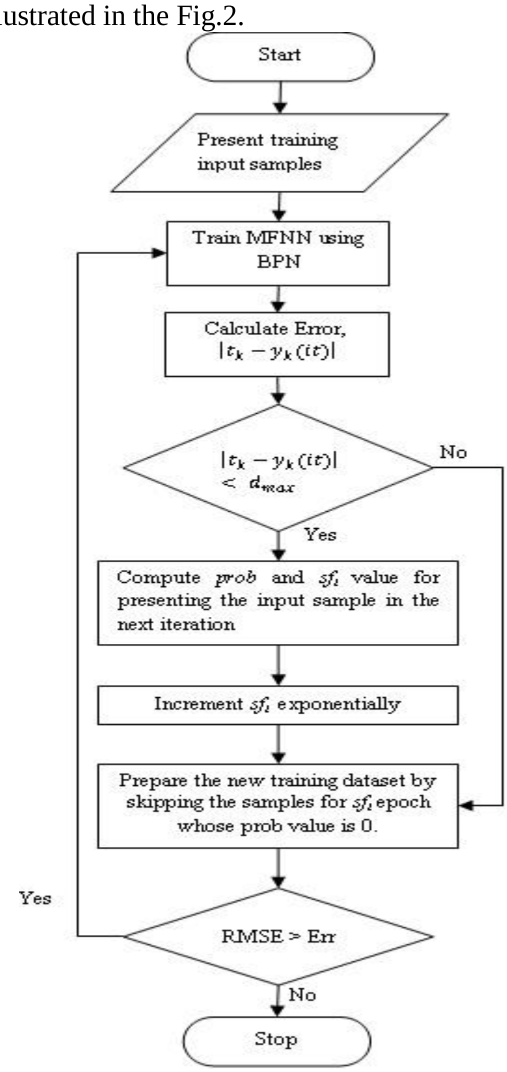

Figure 2: Flow Diagram of EAST Training Algorithm

5 Empirical Result And Analysis

This section holds about the description of the dataset used for the research, the experimental design and results.

5.1 Dataset Properties

In this section, the performance of our proposed (EAST) algorithm is evaluated on the benchmark two-class classification and multi-class classification problems. The benchmark datasets used for two-class classification problem are Iris and Waveform Data Set, and multiclass classification problem are Heart and Breast Cancer Data Set. The fore-mentioned datasets are fetched from the UCI (University of California at Irvine) Machine Learning Repository [15]. The extracted results are compared with the existing BPN and LAST algorithms for both two- and multiclass classification problems.

The characteristic of the training datasets used in the research is summarized in the Table 1.

Table 1. Specification of Benchmark Data Sets

Datasets	No. of Attributes	No. of Classes	No. of Instances
Iris	4	3	150
Waveform	21	3	5000
Heart	13	2	270
Breast Cancer	31	2	569

5.2 Experimental Design

A 3-layer feedforward neural network is adopted for the simulations of all the training algorithms with the selected training architecture and training parameters mentioned in the Table 2. The simulations of all the training algorithms are repeated for two different learning rates such as 1e-4 (0.0001) and 1e-3(0.001).

Table 2. Selected Training Architectures and Parameters

Datasets	Learning Rate	MLP Architecure	Momentum
Iris	1e - 4	$4 \times 5 \times 1$	0.8
	1e – 3		
Waveform	1e – 4	21×10×1	0.7
	1e – 3		
Heart	1e – 4	13×5×1	0.9
	1e – 3		
Breast Cancer	1e – 4	31×15×1	0.9
	1e – 3		

The simulations of all the above training algorithms are done using MATLAB R2010b on a machine with the configuration of Intel® Core I5-3210M processor, 4 GB of RAM and CPU speed of 2.50GHz.

The most popular Nguyen–Widrow (NW) initialization method [6] was used for initializing the MFNN initial weights coefficients. The Fivefold cross validation method is applied to train and test the above training algorithms. Each dataset is split into five disjoint subsets. Among these subsets, a single subset is retained for testing, and the remaining four subsets are used for training. The validation process is repeated five times with each of the five subset used exactly once for testing.

5.3 Experimental Result

5.3.1 Multiclass Problems

5.3.1.1 Iris Data Set

The IRIS dataset is furnished with 150 iris flower samples collected equally from three different varieties of iris flowers. The varieties are listed as Iris Setosa, Iris Versicolour and Iris Virginica. These varieties are identified based on the four characteristics of iris flower such as width and length of Iris sepal, and width and length of Iris petal. Among these varieties, Iris Setosa is easier to be separated from the other two varieties, while the other two varieties, Iris Virgincia and Iris Versicolour, are partially obscured and harder to be distinguished.

The visual representation of the total number of IRIS input samples consumed by BPN, LAST and EAST algorithms for training at every single epoch is laid out in the Fig 3 and Fig 4 with the learning rate of 1e-4 and 1e-3 respectively.

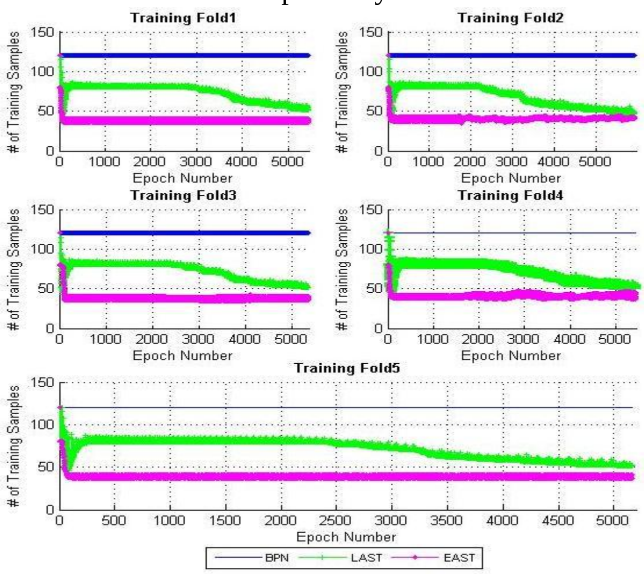

Figure 3: IRIS Epoch wise Input Samples with 1e-4 learning rate

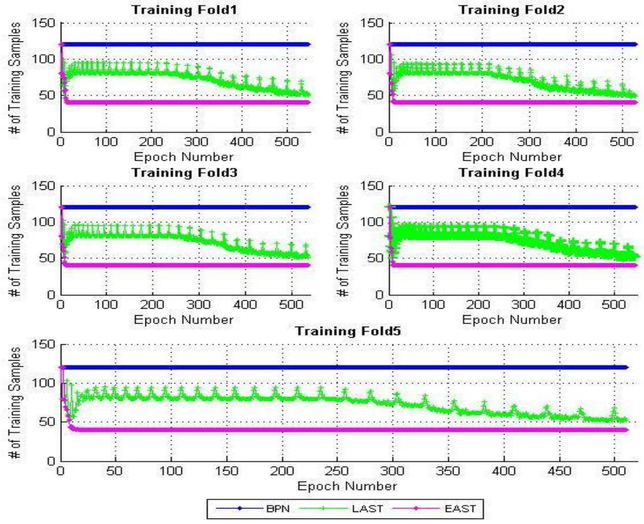

Figure 4: IRIS Epoch wise Input Samples with 1e-3 learning rate

Fig 5 and Fig 6 illustrates the epoch wise training time comparison between BPN, LAST and EAST training algorithm for the learning rates 1e-4 and 1e-3 respectively.

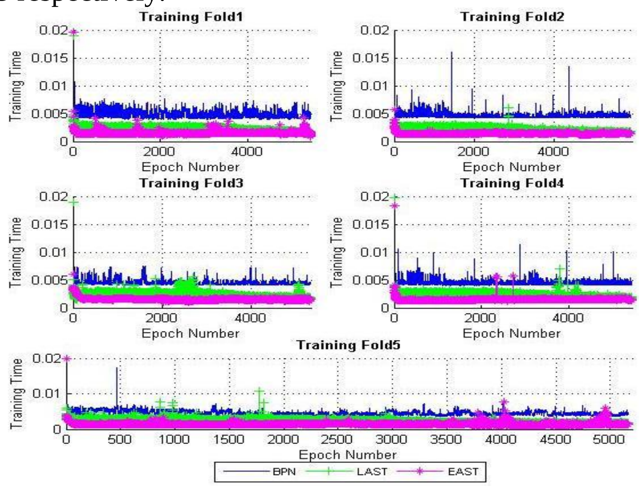

Figure 5: IRIS Epoch wise Training Time with 1e-4 learning rate

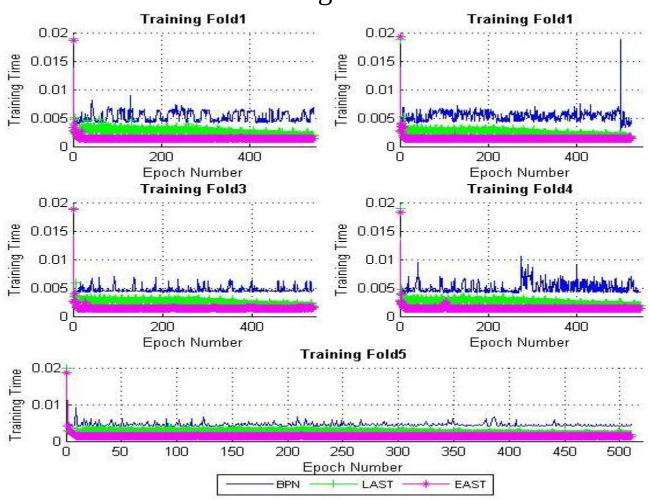

Figure 6: IRIS Epoch wise Training Time with 1e-3 learning rate

5.3.1.2 Waveform Data Set

The Waveform database generator data set consists of measurements of 5000 wave’s samples. The 5000 wave’s samples are equally scattered (about 33%) among the three classes of waves [15]. These samples are collected from the generation of 2 of 3 "base" waves. It contains 21 attributes of numeric values which are involved in the categorization of each class of waves.

The visual representation of the total number of Waveform input samples consumed by BPN, LAST and EAST algorithms for training at every single epoch is laid out in the Fig 7 and Fig 8 with the learning rate of 1e-4 and 1e-3 respectively.

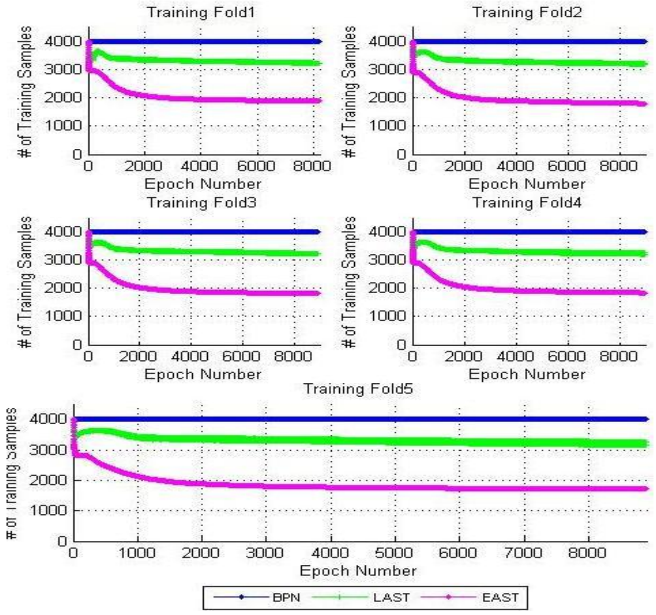

Figure 7: Waveform Epoch wise Input Samples with 1e-4 learning rate

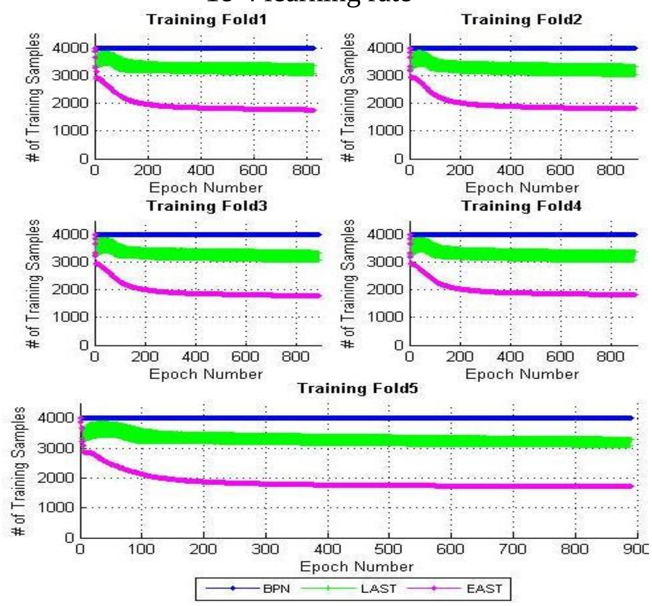

Figure 8: Waveform Epoch wise Input Samples with 1e-3 learning rate

Fig 9 and Fig 10 illustrates the epoch wise training time comparison between BPN, LAST and EAST training algorithm for the learning rates 1e-4 and 1e-3 respectively.

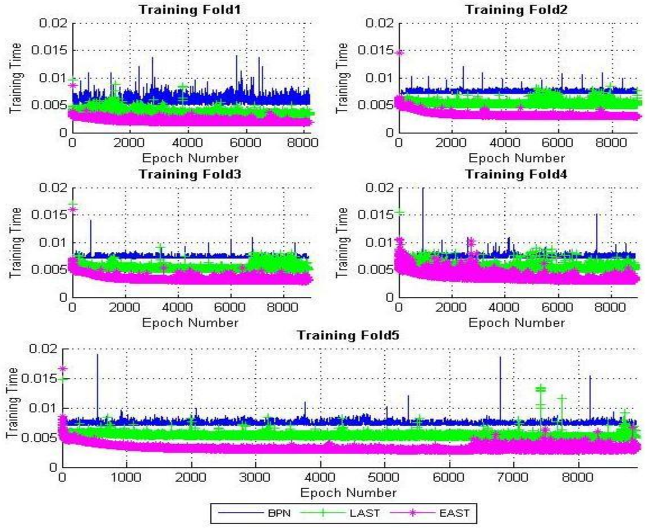

Figure 9: Waveform Epoch wise Training Time with 1e-4 learning rate

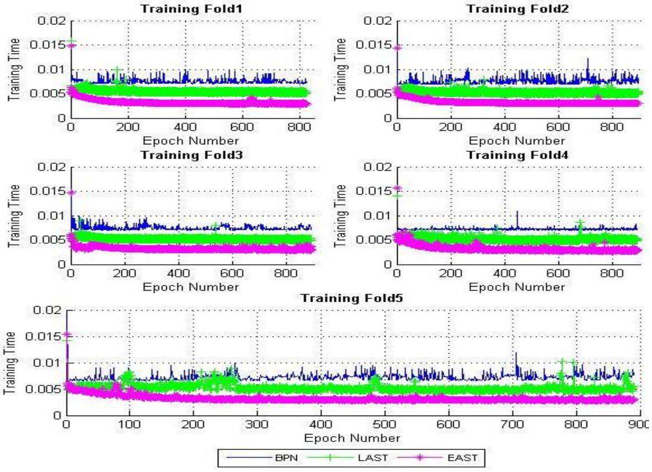

Figure 10: Waveform Epoch wise Training Time with 1e-3 learning rate

5.3.2 Two-Class Problems

5.3.2.1 Heart Data Set

The Statlog Heart disease database consists of 270 patient's samples. The presence or absence of each patient's heart disease is predicted using 13 attributes. Among these 270 patient's samples, 150 samples are the samples of heart disease which is 'absent' and 120 samples of heart disease which is 'present'.

The visual representation of the total number of Heart input samples consumed by BPN, LAST and EAST algorithms for training at every single epoch is laid out in the Fig 11 and Fig 12 with the learning rate of 1e-4 and 1e-3 respectively.

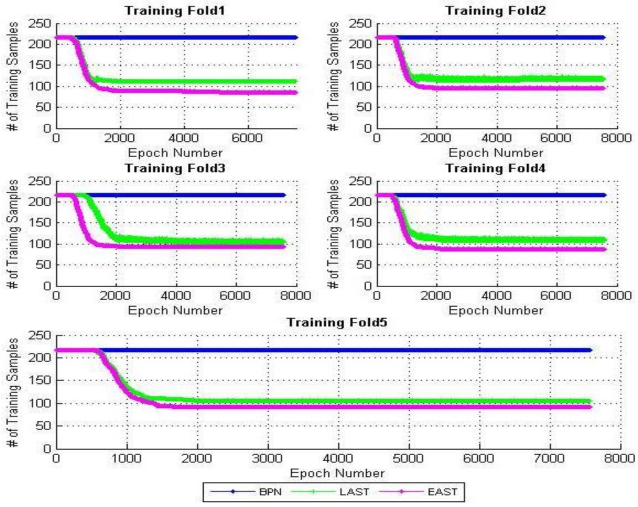

Figure 11: Heart Epoch wise Input Samples with 1e-4 learning rate

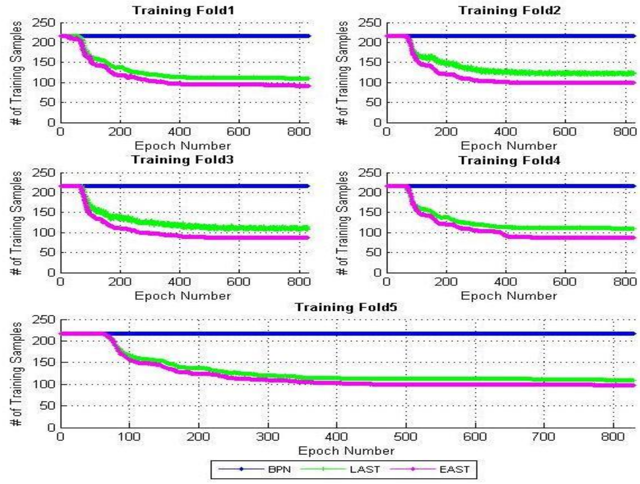

Figure 12: Heart Epoch wise Input Samples with 1e-3 learning rate

Fig 13 and Fig 14 illustrates the epoch wise training time comparison between BPN, LAST and EAST training algorithm for the learning rates 1e-4 and 1e-3 respectively.

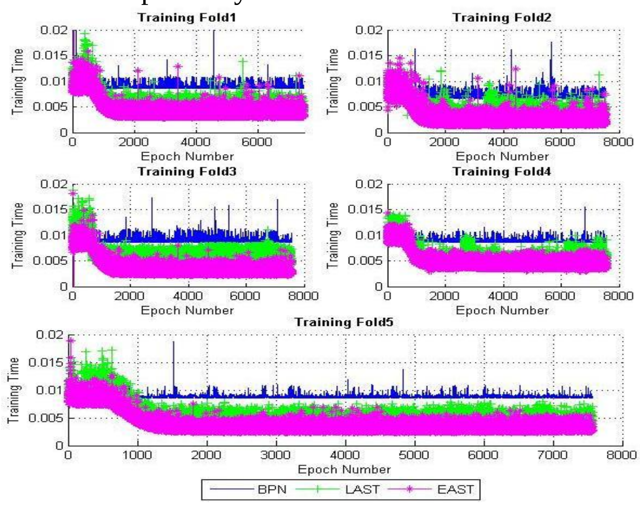

Figure 13: Heart Epoch wise Training Time with 1e-4 learning rate

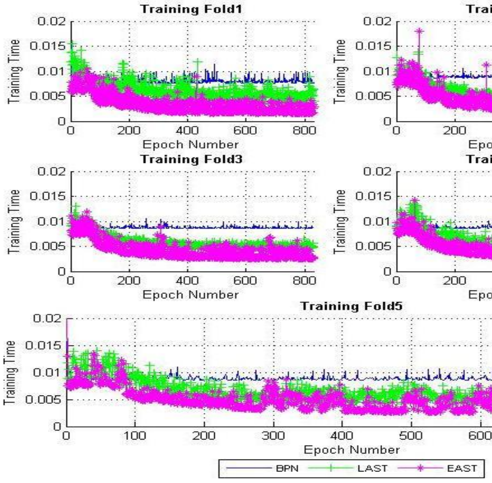

Figure 14: Heart Epoch wise Training Time with 1e-3 learning rate

5.3.2.2 Breast Cancer Data Set

The Wisconsin Breast Cancer Diagnosis Dataset contains 569 patient's breasts samples among which 357 diagnosed as benign and 212 diagnosed as malignant class. Each patient's characteristics are recorded using 32 numerical features.

The visual representation of the total number of Breast Cancer input samples consumed by BPN, LAST and EAST algorithms for training at every single epoch is laid out in the Fig 15 and Fig 16 with the learning rate of 1e-4 and 1e-3 respectively.

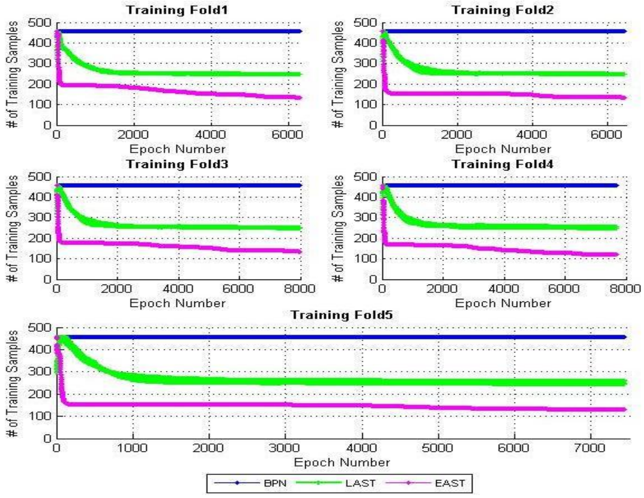

Figure 15: Breast Cancer Epoch wise Input Samples with 1e-4 learning rate

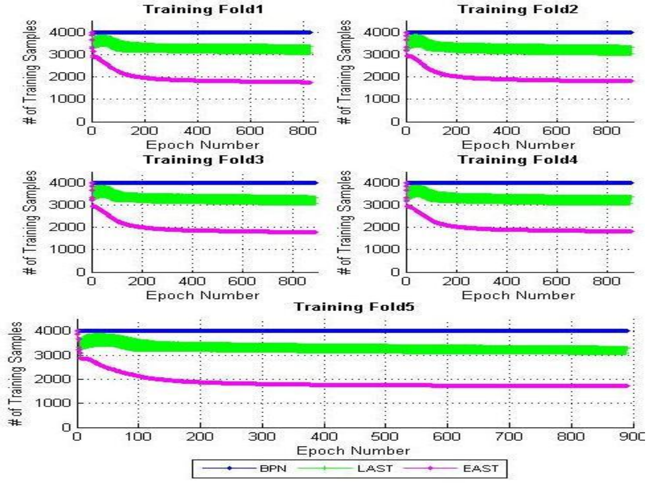

Figure 16: Breast Cancer Epoch wise Input Samples with 1e-3 learning rate

Fig 17 and Fig 18 illustrates the epoch wise training time comparison between BPN, LAST and EAST training algorithm for the learning rates 1e-4 and 1e-3 respectively.

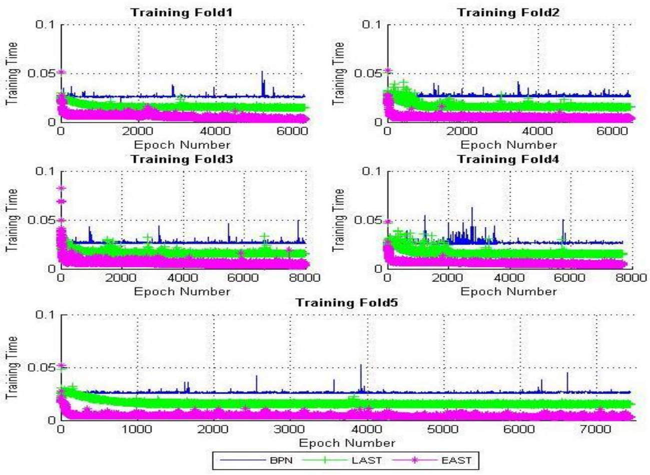

Figure 17: Breast Cancer Epoch wise Training Time with 1e-4 learning rate

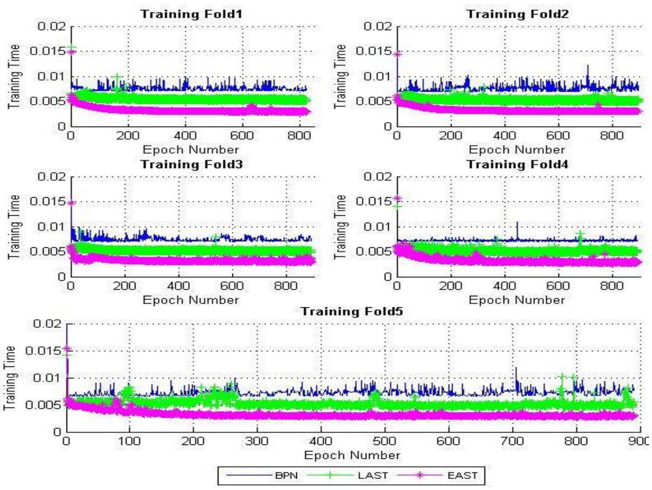

Figure 18: Breast Cancer Epoch wise Training Time with 1e-3 learning rate

5.4 Result Analysis and Comparison

Table 3 to 10 shows the experimental results of BPN, LAST and EAST algorithm observed at each step across five repeats of fivefold cross validation using two different learning rates such as 1e-4 and 1e-3.

From these table 3 to 10, the EAST algorithm yields improved computational training speed in terms of the total number of trained input samples as well as total training time over BPN and less than LAST. But, when the skipping factor goes higher, the accuracy of the system is affected highly.

5.4.1 Training Samples Comparison

The comparison results of the total number of input samples consumed for training by BPN, LAST and EAST with the learning rate of 1e-4 and 1e-3 are shown in Fig.19-26.

From the Fig.19, it is portrayed that the total number of IRIS data samples consumed by EAST algorithm for training under the learning rate of 1e-4 is reduced by an average of nearly 67% and 44% of BPN and LAST algorithm respectively.

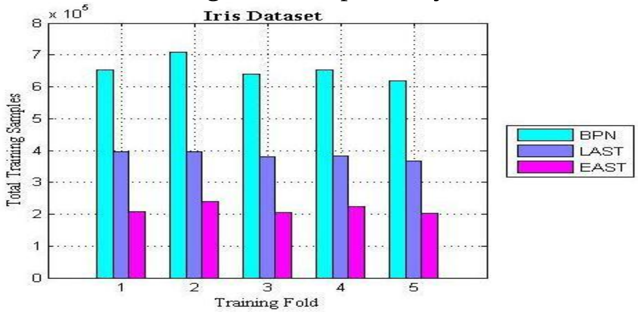

Figure 19: Comparison Result of IRIS Input Samples with 1e-4 learning rate

From the Fig.20, it is portrayed that the total number of IRIS data samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 66% and 44% of BPN and LAST algorithm respectively.

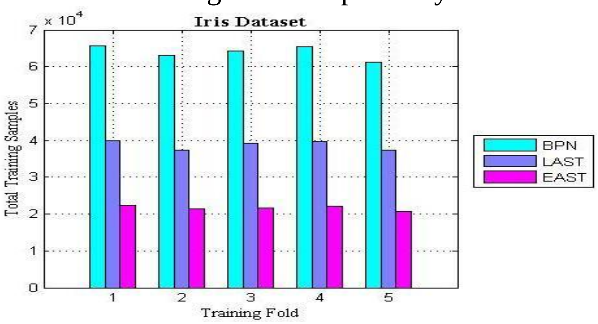

Figure 20: Comparison Result of IRIS Input Samples with 1e-3 learning rate

From the Fig.21, it is portrayed that the total number of Waveform data samples consumed by EAST algorithm for training under the learning rate of 1e-4 is reduced by an average of nearly 50% and 40% of BPN and LAST algorithm respectively.

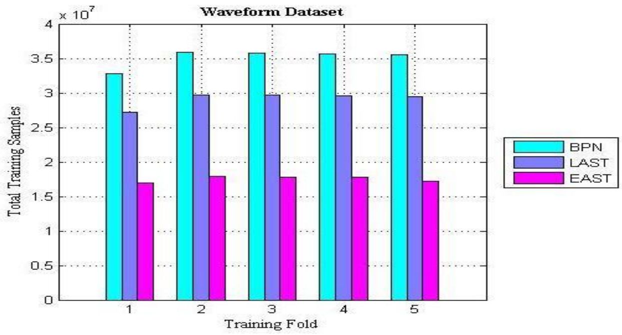

Figure 21: Comparison Result of Waveform Input Samples with 1e-4 learning rate

From the Fig.22, it is portrayed that the total number of Waveform data samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 51% and 41% of BPN and LAST algorithm respectively.

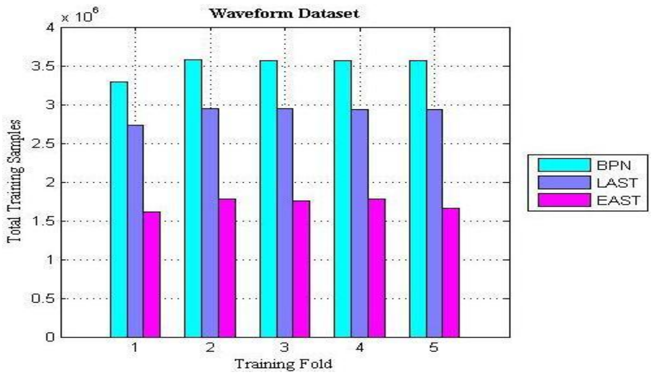

Figure 22: Comparison Result of Waveform Input Samples with 1e-3 learning rate

From the Fig.23, it is portrayed that the total number of Heart data samples consumed by EAST algorithm for training under the learning rate of 1e-4 is reduced by an average of nearly 51% and 17% of BPN and LAST algorithm respectively.

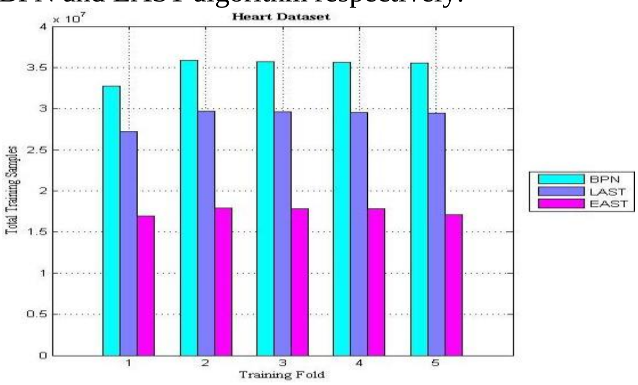

Figure 23: Comparison Result of Heart Input Samples with 1e-4 learning rate

Table 3. Comparison Results Trained by the Iris Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	BPN			LAST			EAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	5442	653040	26.7909	83.33	395718	13.1303	80	208755	8.2995	73.33
2	5902	708240	27.2332	83.33	396670	13.5337	83.33	240293	8.5218	76.67
3	5332	639840	23.6228	80	379759	12.9799	83.33	206029	8.2960	80
4	5439	652680	24.1885	83.33	383028	13.2143	80	223245	8.2565	80
5	5161	619320	23.2492	83.33	365940	12.7051	76.67	203116	7.8261	76.67
Average:		654624	25.0169	82.664	82.664	13.1127	80.666	80.666	8.23998	77.334

Table 4. Comparison Results Trained by the IRIS Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	BPN			LAST			EAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	547	65640	2.8833	83.33	39896	1.4390	83.33	22339	0.7867	76.67
2	526	63120	2.4651	80	37281	1.2867	80	21369	0.7537	80
3	535	64200	2.4906	80	39165	1.3472	80	21735	0.7667	76.67
4	545	65400	2.7546	83.33	39697	1.3740	83.33	22120	0.7756	80
5	510	61200	2.3283	83.33	37425	1.2840	83.33	20735	0.7306	76.67
Average:		63912	2.58438	81.998	38693	1.34618	81.998	21660	0.76266	78.002

Table 5. Comparison Results Trained by the Waveform Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	BPN			LAST			EAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	8187	3274800	47.6683	84.9	2722932	28.9716	85.1	1697498	17.2826	79.8
2	8973	3589200	66.7460	83.7	2966991	52.8073	84.6	1789743	30.3537	80.2
3	8929	3571600	65.7213	84.6	2965645	47.9644	84.5	1781229	30.2254	81.1
4	8903	3561200	64.8988	83.2	2957188	47.3533	83.1	1780697	29.094	80.9
5	8887	3554800	64.3973	82.1	2947611	47.3203	82.5	1714433	28.692	79.9
Average:		3510320	61.8863	83.7	2908211	44.8834	83.96	82.664	27.129	80.38

Table 6. Comparison Results Trained by the Waveform Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	BPN			LAST			EAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	823	3292000	6.1784	84.4	2729243	4.5310	85.6	1611594	2.6747	81.1
2	894	3576000	6.7595	83.8	2944663	4.7575	84.5	1785336	2.9381	80.6
3	891	3564000	6.6254	82.9	2944567	4.6765	83.9	1761213	2.8975	79.9
4	890	3560000	6.4547	83.5	2938903	4.6199	83.6	1784880	2.8904	80.5
5	890	3560000	6.4537	84.1	2937498	4.6656	84.6	1659327	2.8696	80.1
Average:		3510400	6.49434	83.74	2898974.8	4.6501	84.44	1720470	2.85406	80.44

Table 7. Comparison Results Trained by the Heart Dataset with 1e-4 Learning Rate

Testing Fold	**Number of Epochs**	**BPN**			**LAST**			**EAST**		
		Total Number of Input Samples	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**
1	7485	1616760	58.0715	81.48	81.48	43.3506	83.33	713559	23.2651	75.93
2	7529	1626264	60.2075	83.33	83.33	46.7666	81.48	809372	25.3458	74.07
3	7569	1634904	67.8729	83.33	83.33	48.6806	83.33	820114	27.8431	75.93
4	7567	1634472	66.8935	81.48	81.48	47.8751	79.63	813699	26.6308	79.63
5	7567	1634472	66.5249	81.48	81.48	47.3221	81.48	811180	25.9578	77.78
Average:		1629374	63.91406	82.22	82.22	82.22	81.85	793584.8	25.808518	76.668

Table 8. Comparison Results Trained by the Heart Dataset with 1e-3 Learning Rate

Testing Fold	**Number of Epochs**	**BPN**			**LAST**			**EAST**		
		Total Number of Input Samples	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**
1	830	179280	7.3662	81.48	107845	4.9837	83.33	95137	3.3133	74.07
2	828	178848	7.361153	83.33	116169	5.238218	81.48	98116	3.382314	75.93
3	829	179064	7.265956	83.33	108534	4.492601	83.33	90205	3.533761	75.93
4	829	179064	7.326156	79.63	107736	4.772563	81.48	93136	3.554815	74.07
5	829	179064	7.341574	81.48	107736	5.274545	81.48	99092	3.993784	77.78
Average:		179064	7.332208	81.85	81.85	4.95233	82.22	95137.2	3.5555948	75.556

Table 9. Comparison Results Trained by the Breast Cancer Dataset with 1e-4 Learning Rate

Testing Fold	**Number of Epochs**	**BPN**			**LAST**			**EAST**		
		Total Number of Input Samples	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**
1	6279	2856945	162.5596	87.72	165949	100.109	87.72	105584	34.0808	83.33
2	6460	2939300	172.0937	86.64	171832	105.638	86.64	966328	30.7942	79.82
3	7976	3629080	210.8542	88.6	214090	131.4230	87.72	128626	46.8745	84.21
4	7691	3499405	203.5600	86.84	207454	125.0857	85.97	113897	43.9744	80.07
5	7439	3392184	193.7257	87.61	199608	119.5164	87.61	109727	31.3622	84.07
Average:		3263383	188.5586	87.482	87.482	116.354	87.132	87.132	37.417	82.3

Table 10. Comparison Results Trained by the Breast Cancer Dataset with 1e-3 Learning Rate

Testing Fold	**Number of Epochs**	**BPN**			**LAST**			**EAST**		
		Total Number of Input Samples	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**	**Total Number of Input Samples**	**Training Time (in Sec)**	**Accuracy (%)**
1	609	277095	16.5255	87.72	161260	10.3436	85.97	101916	5.4285	83.33
2	647	294385	17.2322	86.64	172059	10.5972	86.64	107089	5.8950	84.21
3	785	357175	21.3841	88.6	210885	13.4171	87.72	132372	6.4982	84.21
4	750	341250	19.7409	86.84	202580	12.1622	85.97	128676	5.8950	83.33
5	743	338808	19.7142	87.61	199366	11.9810	87.61	120608	5.7421	84.07
Average:		321742.6	18.91938	87.482	87.482	11.7002	86.782	86.782	5.89176	83.83

From the Fig.24, it is portrayed that the total number of Heart data samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 47% and 13% of BPN and LAST algorithm respectively.

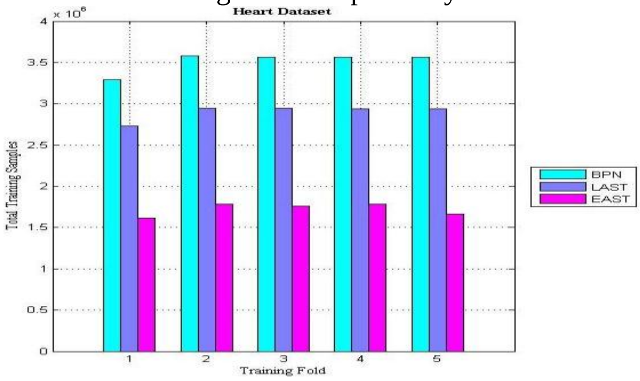

Figure 24: Comparison Result of Heart Input Samples with 1e-3 learning rate

From the Fig.25, it is portrayed that the total number of Breast Cancer data samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 66% and 42% of BPN and LAST algorithm respectively.

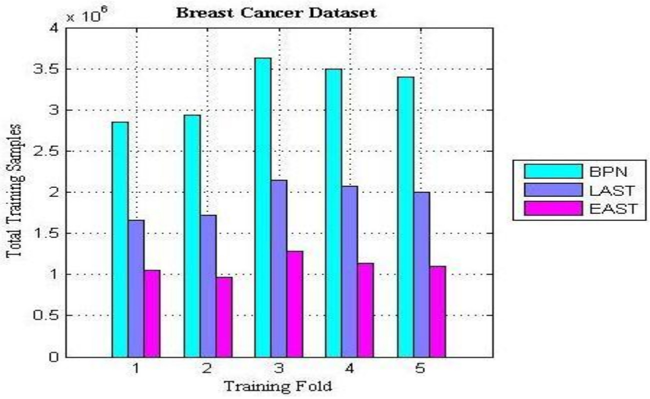

Figure 25: Comparison Result of Breast Cancer Input Samples with 1e-4 learning rate

From the Fig.26, it is portrayed that the total number of Breast Cancer data samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 63% and 38% of BPN and LAST algorithm respectively.

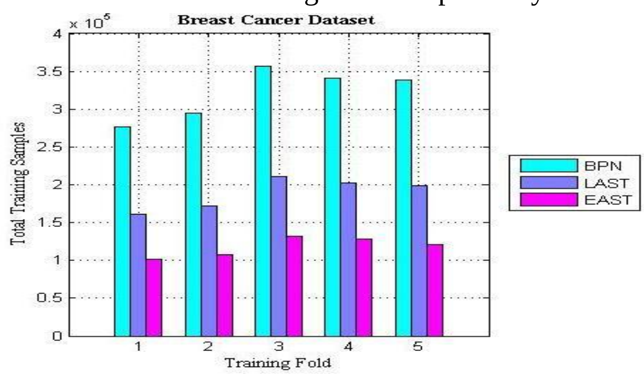

Figure 26: Comparison Result of Breast Cancer Input Samples with 1e-3 learning rate

5.4.2 Training Time Comparison

Thus decreasing the size of the trained input samples can reduce the training time which is shown in this section, thereby increasing the speed of the training process. Fig.27-34 illustrates the training time comparison between BPN, LAST and EAST training methods for different learning rate of 1e-4 and 1e-3.

From the Fig 27, the total training time for training IRIS dataset by EAST algorithm is reduced to an average of 67% of BPN algorithm and 37% of LAST algorithm for the learning rate of 1e-4.

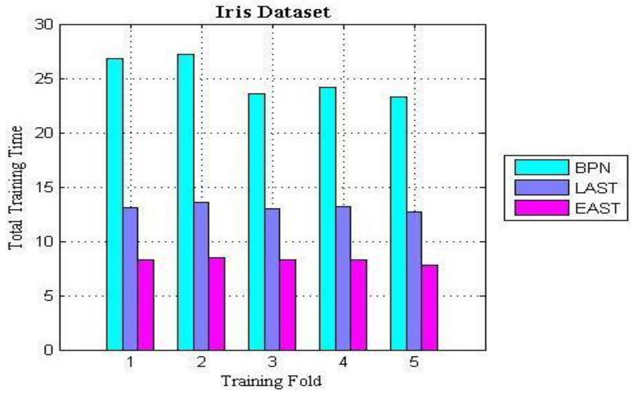

Figure 27: Comparison Result of IRIS Training Time with 1e-4 learning rate

From the Fig 28, the total training time for training IRIS dataset by EAST algorithm is reduced to an average of 70% of BPN algorithm and 43% of LAST algorithm for the learning rate of 1e-3.

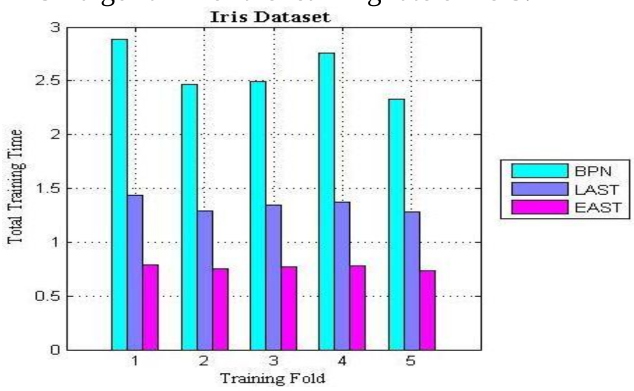

Figure 28: Comparison Result of IRIS Training Time with 1e-3 learning rate

From the Fig 29, the total training time for training waveform dataset by EAST algorithm is reduced to an average of 56% of BPN algorithm and 40% of LAST algorithm for the learning rate of 1e-4.

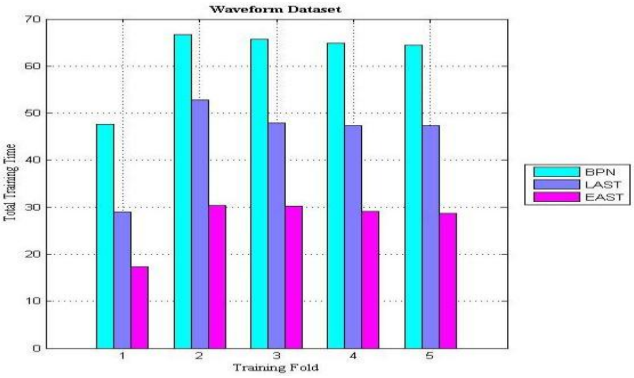

Figure 29: Comparison Result of Waveform Training Time with 1e-4 learning rate

From the Fig 30, the total training time for training waveform dataset by EAST algorithm is reduced to an average of 56% of BPN algorithm and 39% of LAST algorithm for the learning rate of 1e-3.

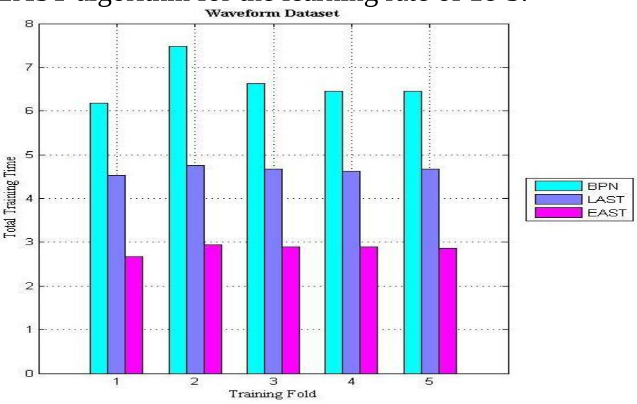

Figure 30: Comparison Result of Waveform Training Time with 1e-3 learning rate

From the Fig 31, the total training time for training Heart dataset by EAST algorithm is reduced to an average of 60% of BPN algorithm and 45% of LAST algorithm for the learning rate of 1e-4.

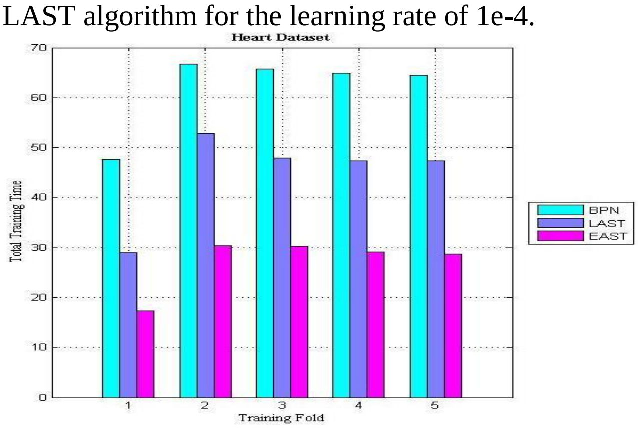

Figure 31: Comparison Result of Heart Training Time with 1e-4 learning rate

From the Fig 32, the total training time for training Heart dataset by EAST algorithm is reduced to an average of 52% of BPN algorithm and 28% of LAST algorithm for the learning rate of 1e-3.

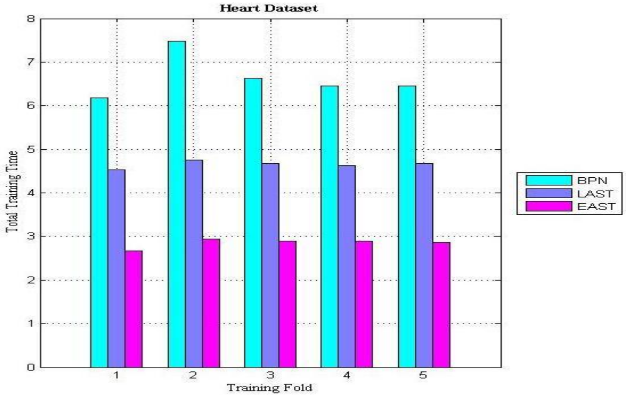

Figure 32: Comparison Result of Heart Training Time with 1e-3 learning rate

From the Fig 33, the total training time for training Breast Cancer dataset by EAST algorithm is reduced to an average of 80% of BPN algorithm and 68% of LAST algorithm for the learning rate of 1e-4.

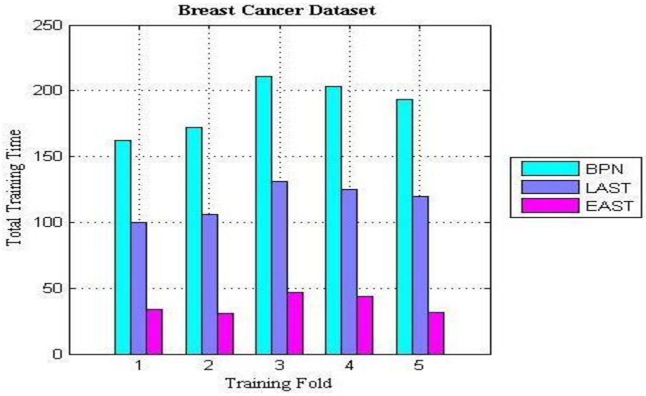

Figure 33: Comparison Result of Breast Cancer Training Time with 1e-4 learning rate

From the Fig 34, the total training time for training Breast Cancer dataset by EAST algorithm is reduced to an average of 69% of BPN algorithm and 50% of LAST algorithm for learning rate of 1e-3.

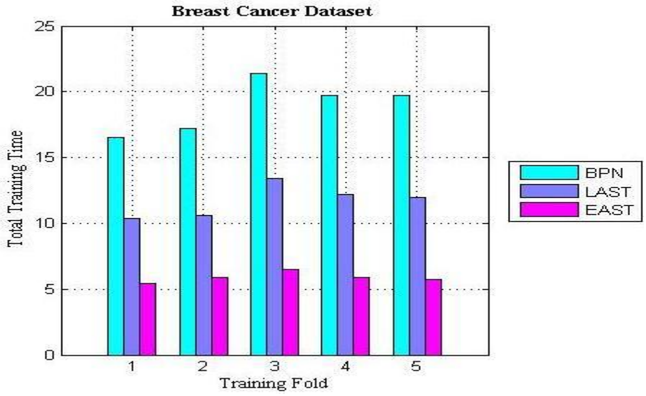

Figure 34: Comparison Result of Breast Cancer Training Time with 1e-3 learning rate

Although the training performance of EAST achieves faster, it still lacks in the accuracy rate due to high skipping factor. So, further work should be concentrated on how to improve the accuracy rate of the training algorithm also.

6 Conclusion

In this brief, a simple and fast training algorithm called Exponential Adaptive Skipping Training (EAST) Algorithm is presented. The simulation results showed that, compared to other training methods, the new algorithm could significantly reduces the total number of training input samples presented to the MFNN at every single cycle. Thus decreasing the size of the training input samples can reduce the training time thereby increases the training speed. Finally, the proposed EAST algorithm seems to be faster than the standard BPN and LAST algorithm in training MFNN and also the EAST Algorithm can be used in addition with any supervised training algorithm for any real-world supervised task classification. Although the training performance of EAST achieves faster, it still lacks in the accuracy rate due to high skipping factor. So, further work should be concentrated on how to improve the accuracy rate of the training algorithm also.

References

[1] Mehra, P. and Wah, B. W. "Artificial Neural Networks: Concepts and Theory", IEEE Computer Society Press,1992.

[2] Hornik, M., Stinchcombe, M., and White, H. "Multilayer feedforward networks are universal approximators", *Neural Networks, 2*, 359-366, 1989.

[3] G.-B. Huang, Y.-Q. Chen, and Babri, H. A. "Classification ability of single hidden layer feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 799–801, May 2000.

[4] Shao, H., and Zheng,H. "A New BP Algorithm with Adaptive Momentum for FNNs Training," In: GCIS 2009, Xiamen, China, pp. 16—20, 2009.

[5] Owens, Aaron, J. "Empirical Modeling of Very Large Data Sets Using Neural Networks", International Joint Conference on Neural Networks, vol. 6, pp. 6302-10, 2000.

[6] D. Nguyen, B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights", International Joint Conference on Neural Networks, vol. 3, San Diego, CA, pp.21-26,1990.

[7] Varnava, T.and Meade, A.J. "An Initialization Method for Feedforward Artificial Neural Networks Using Polynomial Bases," Advances in Adaptive Data Analysis, vol.3, No.3, pp.385-400,2011.

[8] Plagianakos, V.P., Sotiropoulos, D.G. and Vrahatis, M.N. "A Nonmonotone Backpropagation Training Method for Neural Networks", Dept. of Mathematics, Univ. of Patras, Technical Report No.98-04, 1998.

[9] Shao, H., and Zheng,H. "A New BP Algorithm with Adaptive Momentum for FNNs Training," In: GCIS 2009, Xiamen, China, pp. 16—20, 2009.

[10] Behera, L., Kumar, S. and Patnaik, A.,"On adaptive learning rate that guarantees convergence in feedforward networks," IEEE Transactions on Neural Networks, vol. 17, No. 5, pp. 1116-1125, 2006.

[11] Ampazis, N., Perantonis, S.J. "Two Highly Efficient Second Order Algorithms for Training Feedforward Networks," IEEE Transactions on Neural Networks, vol.13 No.5, pp.1064–1074, 2002.

[12] Yu, H., Wilamowski,B .M. "Improved Computation for Levenberg Marquardt Training," IEEE Transaction on Neural Networks, vol.21 No.6, pp.930-937, 2010.

[13] Yu, H., Wilamowski,B .M. "Neural Network Training with Second Order Algorithms," Human- computer Systems Interaction, AISC 99, Part II, pp.463-476,2012.

[14] Manjula Devi, R., Kuppuswami,S., and Suganthe, R. C. "Fast Linear Adaptive Skipping Training Algorithm for Training Artificial Neural Network," Mathematical Problems in Engineering, vol. 2013, Article ID 346949, 9 pages, 2013.

[15] Asuncion, A. and Newman, D.J. UCI Machine Learning Repository [http://www. ics. uci. edu / ~mlearn/MLRepository.html], School of Information and Computer Science, University of California, Irvine, CA,2007.