International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

A Deep Comparative Study between Different Neural Networks Classifiers

Alka Kumari[1], Ankita Sharma[2]

[1]M.Tech Student, Punjab Technical University Regional Centre, SSIET, Derabassi, Punjab, India

[2]Assistant Professor, Punjab Technical University Regional Centre, SSIET, Derabassi, Punjab, India

Abstract: ***In Machine learning and artificial intelligence have seemingly never been as typical and relevant to real-time applications as they are in these days autonomous, big data era. The fortune of machine learning and artificial intelligence depends on the coexistence of three important conditions: powerful computing environments, rich and/or large data, and efficient learning techniques (algorithms). The Extreme Learning Machine (ELM) as an emerging learning method provides efficient unified solutions to generalized feed-forward networks including but not limited to (both single- and multi-hidden-layer) neural networks, radial basis function (RBF) networks, and kernel learning. The widely used supervised neural network is the Support Vector Machine (SVM). It is known as being very accurate but at the expenditure of high computational complexity, particularly in the learning phase, making it less appropriate for hardware-oriented applications. A deep belief network (DBN) is an originative graphical model, or alternatively a type of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. In this paper a new comparative study is proposed on different neural networks classifiers. The technique is implemented for accuracy of different algorithms.***

Keywords: Neural network, support vector machine, extreme learning machine, deep belief network, data

1. Introduction

Artificial neural networks (ANN) have become very useful and successful in domains as pattern classification and regression, because using rule-based programming does not offer the right solution or any solution at all. The ANN-s where created to imitate the biological neural system and are mostly classified into three major categories based on their learning technique: i) ANN-s with unsupervised learning, ii) ANN-s with supervised learning, iii) ANN-s with reinforcement learning.

Unsupervised learning ANN-s uses their neural network topology as a self-organizing cluster where the outputs are adapted to minimize a function of the spacing between the elements [1]. The most commonly used unsupervised learning algorithms are the self-organizing map (SOM) and the adaptive resonance theory (ART). SOM use a neighborhood function to preserve the topological properties of the input space [2]. ART consists of a comparison field, a recognition field, a vigilance parameter and a reset module [3].

FSVC (Fast Support Vector Classifier) A low complexity neural network is the FSVC previously named as RBF-M (a modified RBF – Radial Basis Function) first introduced in [5]. It has a simple constructive training algorithm with the complexity of O(N), where N is the number of the samples in the dataset. The training algorithm is a simple Adaline trained in an m dimensional space formed by the outputs of the dataset, where the center vectors for each RBF-neuron created are determined [6].

Basically ELM is a single-hidden layer feed forward neural network (SLFN), where the first weight matrix is arbitrarily chosen as described in [8] and [9]. This allows the output matrix to be estimated via least squares in a very fast way. The Bartlett's theory described in [14] which stands for this type of neural network is based on the idea that the smaller the norm of weights is the better generalization performance the network tends to have.

Deep belief nets are probabilistic generative models that are composed of multiple layers of stochastic, latent variables. The latent variables typically have binary values and are often called hidden units or feature detectors. The top two layers have undirected, symmetric connections between them and form an associative memory. The lower layers receive top-down, directed connections from the layer above. The states of the units in the lowest layer represent a data vector

2. Literature Survey

M. Bucurica et al. [14] performed a comparative study between two types of single-hidden layer feed forward neural network (SLFN). Here interest was to see if FSVC performs similarly to ELM which one of the most featured neural paradigms. FSVC presents, in comparison to ELM several advantages, particularly when targeting embedded systems (FPGA, microcontrollers, etc.): i) there is only one radius so all units have a similar architecture, easy to clone; ii) there is no need for special resources (PRNG) to generate the centers since they are simply selected among the available data from the training set; iii) the pseudo-inverse learning algorithm (easy to implement in Matlab but less convenient for simpler hardware platforms) is replaced with the much simpler to implement iterative LMS algorithm.

A wide range of problems was used leading to the following conclusions: i) Accuracy is similar for both architectures in most cases; ii) The number of neurons appears to be smaller in most cases for FSVC. A speed comparison is considered

although is not entirely relevant since for FSVC there was no special effort to optimize the implementation. Still, these results indicate that in general ELM is a bit faster (some cases 10 times faster as for FSVC) using the Matlab implementation. However, in particular embedded systems a further, more careful study must be conducted to compare the speed of them. Further research involves optimizing the FSVC for speed and also adding some improvements in order to increase accuracy, up to the best obtained with ELM.

S. F. Mahmood et al. [15] proposed a new approach for selecting the optimal number of hidden nodes for ELM. In practical applications, the selection of hidden neurons of ELM has been considered crucial, most especially, for realtime problems. This new algorithm (SVM–ELM) select the candidate neurons of ELM by optimizing the network architecture with 1-norm SVM. Initially, an ELM regression model is trained to obtain the model error. Then, a criteria value, which is based on either the mean or median of the model's error is used to separate the approximated data matrix for training 1-norm SVM. The small number of support vectors that lie on the optimal separating hyperplane are considered as the candidate hidden neurons in the ELM.
The SVM–ELM has been evaluated using 10 regression datasets from the UCI ML repository and Statlib. The results indicated the effectiveness of the proposed node selection method, both in terms of training time and prediction performance.

L. Zhang et al. [16] presented a systematic comparison between SVMs and ELMs for object recognition with multiple domains based on the deep convolutional activation features trained by CNN on a subset of 1000-category images from ImageNet. I was aimed at exploring the most appropriate classifiers for high-level deep features in classification. In experiments, the deep features of 10-category object images of 4 domains from the 6-th layer and 7-th layer of CNN are used as the inputs of general classifiers including NN, SVM, LSSVM, ELM and KELM, respectively. The recognition accuracies for each method under three different experimental settings are reported. A number of experimental results clearly demonstrate that ELMs outperform SVM based classifiers in different settings. In particular, KELM shows state-of-the-art recognition performance among the presented 5 popular classifiers.

J. Chorowski et al. [17] presented that machine learning classifiers can be analyzed using a common framework of convex optimization. Four classifier models, the Support Vector Machine (SVM), the Least-Squares SVM (LSSVM), the Extreme Learning Machine (ELM), and the Margin Loss ELM (MLELM) are discussed to demonstrate how specific parametrizations of a general problem statement affect the classifier design and performance, and how ideas from the four different classifiers can be mixed. and used together. Furthermore, 21 public domain benchmark datasets are used to experimentally evaluate five performance metrics of each model and corroborate the theoretical analysis. Comparison of classification accuracies under a nested cross-validation evaluation shows that with an exception all four models perform similarly on the evaluated datasets.

The classical ELM algorithm yields usually good testing accuracy while being fast to train and fast to apply to new data. The MLELM fusion of SVM's loss function and ELM's feature transformation performs poorly, because it is often unacceptably slow to train and yields accuracy similar to the classical ELM algorithm. The SVM offers state-of-the art accuracy, it is however more expensive than the ELM to both train and apply to new data. The LSSVM often matches or surpasses SVM's accuracy, and its training algorithm is simpler that the one used for SVM.

3. Methodologies

3.1 Support Vector Machine

We are given a training dataset of n points of the form

$$(\vec{x}_1, y_1), \ldots\ldots\ldots, (\vec{x}_n, y_n)$$

where the y_i are either 1 or −1, each indicating the class to which the point $\vec{x}_1$ belongs. Each $\vec{x}_1$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\vec{x}_1$ for which $y_i = 1$ from the group of points for which $y_i = 1$, which is defined so that the distance between the hyperplane and the nearest point $\vec{x}_1$ from either group is maximized.
Any hyperplane can be written as the set of points $\vec{x}$ satisfying

$$\vec{w}.\vec{x} - b = 0$$

Where $\vec{w}$ is the normal vector to the hyperplane. The parameter $b/\|\vec{w}\|$ determines the offset of the hyperplane from the origin along the normal vector $\vec{w}$.

If the training data are linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. These hyperplanes can be described by the equations

$$\vec{w}.\vec{x} - b = 1 \text{ and } \vec{w}.\vec{x} - b = -1$$

3.2 Deep Belief Networks

The training algorithm for DBNs proceeds as follows. Let X be a matrix of inputs, regarded as a set of feature vectors.

1) Train a restricted Boltzmann machine on X to obtain its weight matrix, W. Use this as the weight matrix between the lower two layers of the network.
2) Transform X by the RBM to produce new data X', either by sampling or by computing the mean activation of the hidden units.
3) Repeat this procedure with $X \leftarrow X'$ for the next pair of layers, until the top two layers of the network are reached.
4) Fine-tune all the parameters of this deep architecture with respect to a proxy for the DBN log- likelihood, or with respect to a supervised training criterion (after adding extra learning machinery to convert the learned representation into supervised predictions, e.g. a linear classifier).

3.3 Extreme Learning Machine

The simplest ELM training algorithm learns a model of the form

$$\hat{Y} = W_2\sigma(W_{1x})$$

where $\mathbf{W}_1$ is the matrix of input-to-hidden-layer weights, σ is some activation function, and $\mathbf{W}_2$ is the matrix of hidden-to-output-layer weights. The algorithm proceeds as follows:

1) Fill $\mathbf{W}_1$ with Gaussian random noise;
2) Estimate $\mathbf{W}_2$ by least-squares fit to a matrix of response variables $\mathbf{Y}$, computed using the pseudoinverse $\cdot^+$, given a design matrix $\mathbf{X}$:

3.4 Proposed Methodology

The following strategy will be followed to get the desired results.

Step 1: Implement all three existing techniques i.e. SVM, ELM and DBN
Step 2: Select the database based on machine learning on which these techniques will be tested.
Step 3: Compare the results for different weight parameters.
Step 4: Compare the results on the basis of different tools.

4. Results

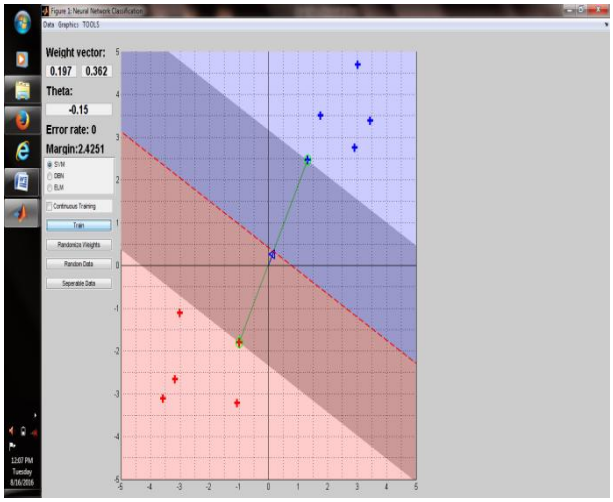

Figure 4.1: Result after training for SVM system

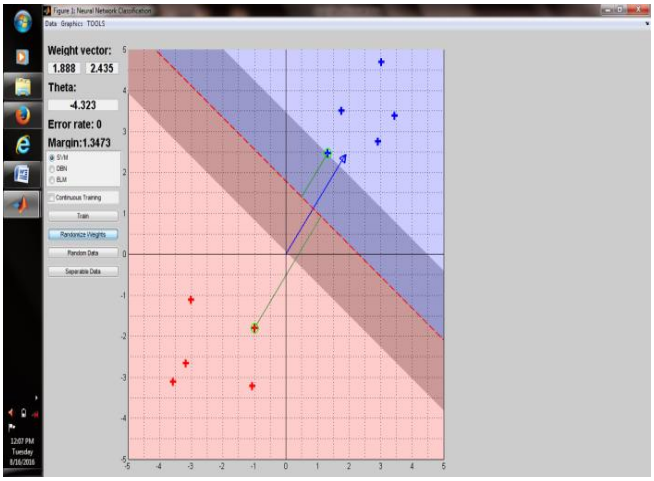

Figure 4.2: Result after initializing random weights for SVM system

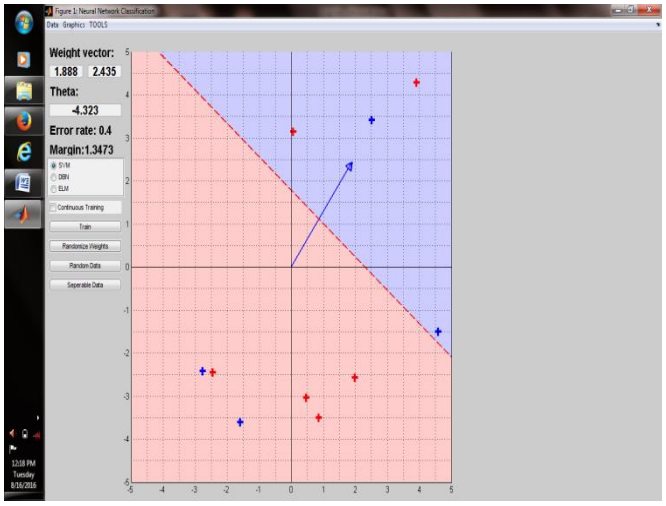

Figure 4.3: Result after initializing random data for SVM system

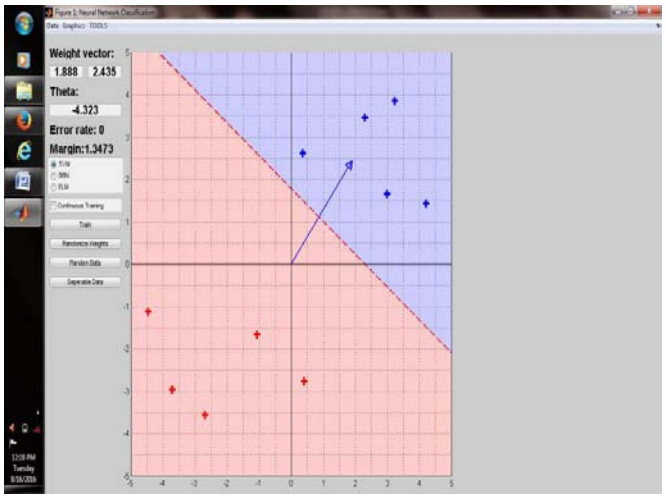

Figure 4.4: Result after applying separable data on SVM ystem

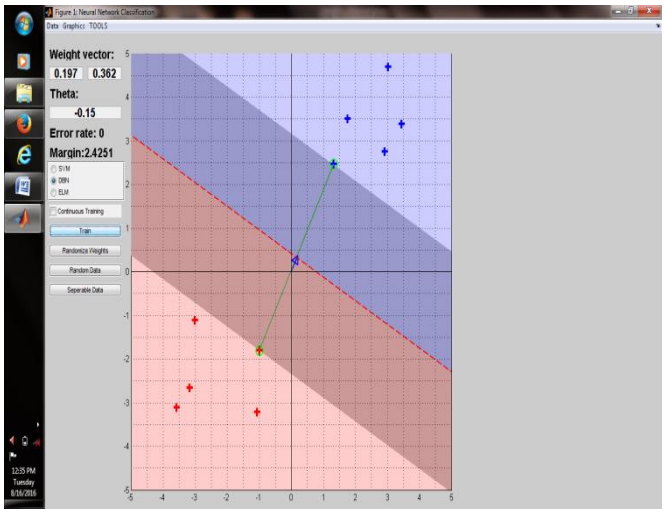

Figure 4.5: Result after training the data for DBN system

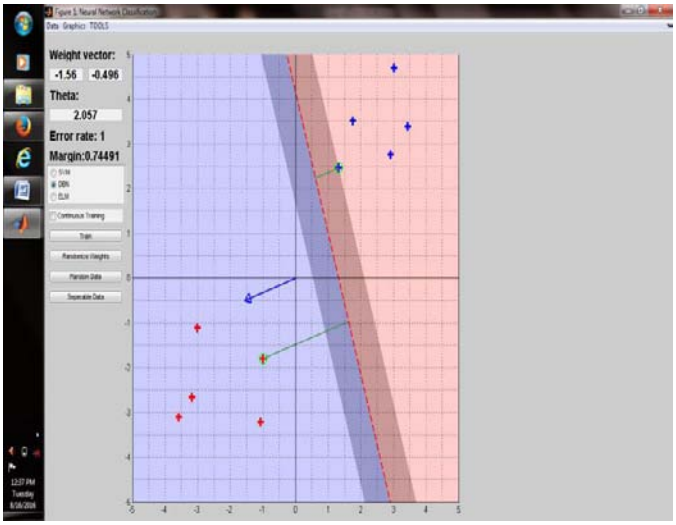

Figure 4.6: Result after training the data for DBN system

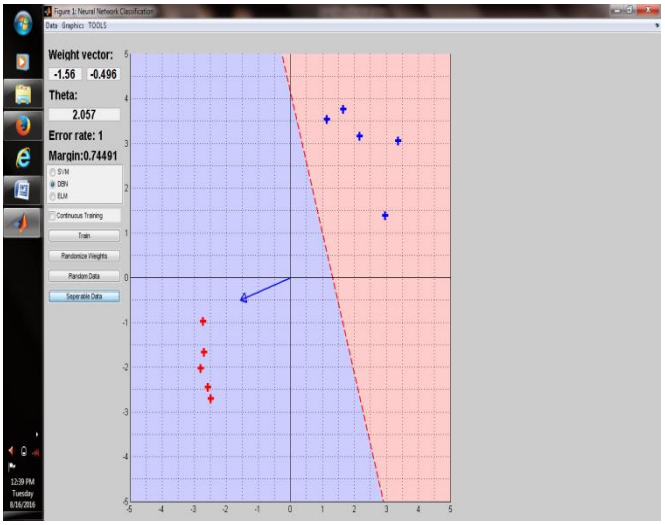

Figure 4.9: Result after separating the random data for DBN system

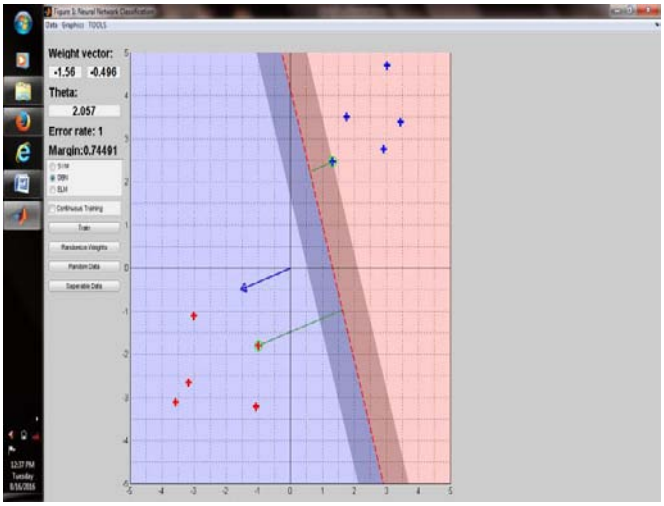

Figure 4.7: Result after assigning random weights for DBN system

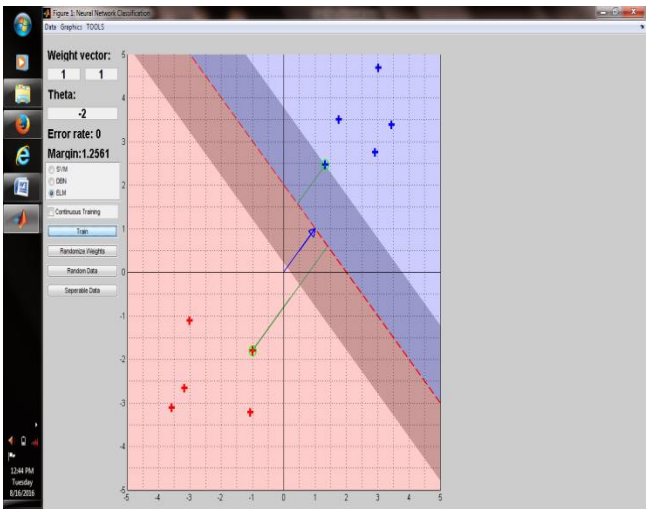

Figure 4.10: Result of training data for ELM system

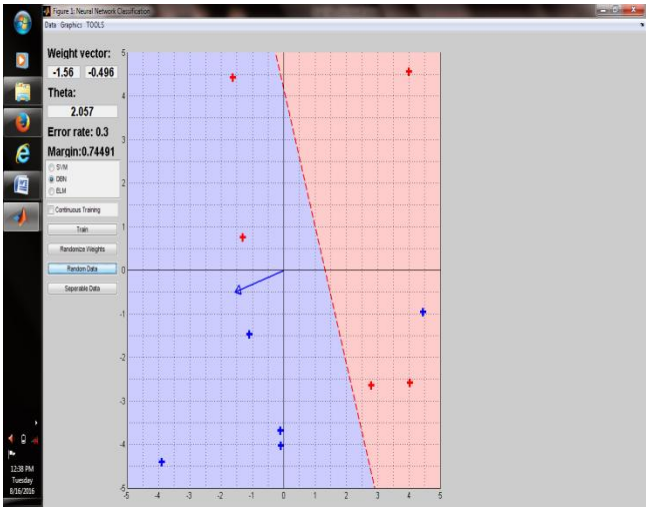

Figure 4.8: Result after initializing the random data for DBN system

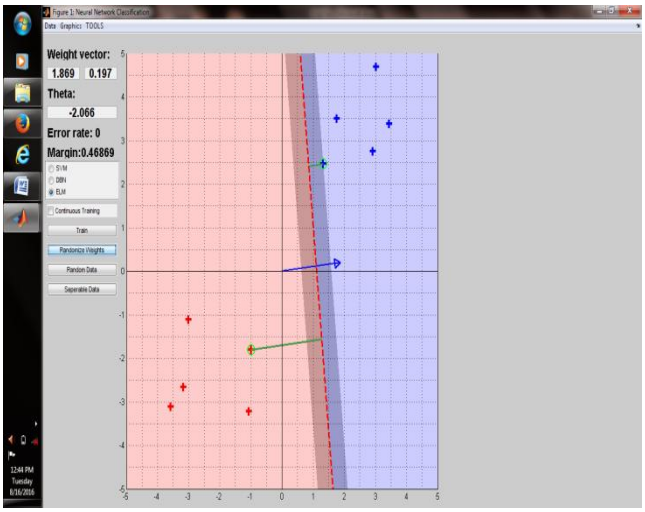

Figure 4.11: Result after initializing random weights for ELM system

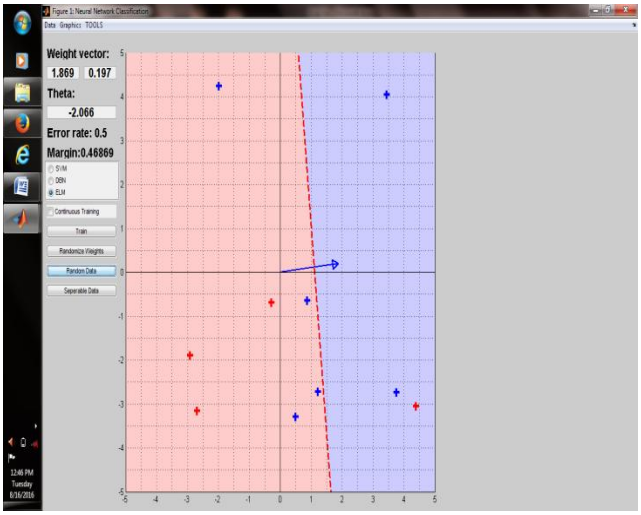

Figure 4.12: Result after assigning random data for ELM system

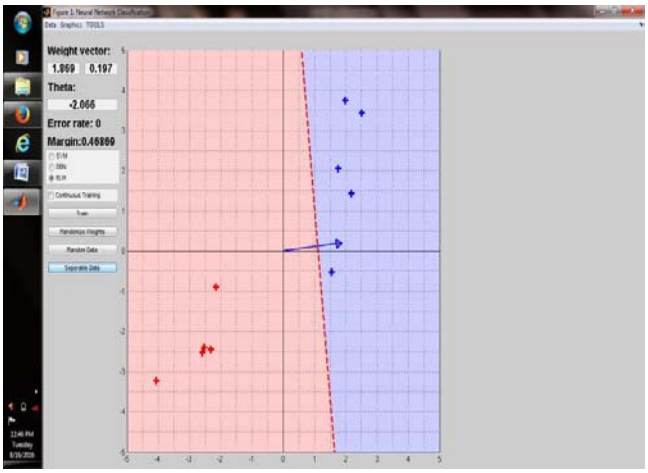

Figure 4.13: Result after separating random data for ELM system

5. Conclusion

The results show that SVM is more efficient than ELM and DBN. The classification accuracy is much better for the support vector machine while extreme learning machine and deep belief networks provides good results but are not upto the same level of support vector machine. In the future work others algorithm can be compared for the different types of other databases. The present research is based on two dimensional data and in the future three and more dimensional data can be taken for comparison.

References

[1] Vincent W. Porto, David B. Fogel, Lawrence J. Fogel, "Alternative neural network training methods", published by the IEEE Computer Society, June 1995 (vol. 10, pp. 16-22)

[2] Kohonen T., "The self-organizing map", in Proceedings of the IEEE Sep. 1990 (vol. 78, pp. 1464-1480)

[3] Gail A. Carpenter, Stephen Grossberg, David Rosen, "An adaptive resonance algorithm for rapid category learning and recognition", in Neural Networks, IJCNN-91-Seattle International Joint Conference on. IEEE, 1991, pp. 151-156.

[4] Corinna Cortes, Vladimir Vapnik, "Support-vector networks, "Machine learning", 1995 (vol. 20, pp. 273-297).

[5] R. Dogaru, A.T. Murgan, S. Ortmann, M. Glesner, "A modified RBF neural network for efficient current-mode VLSI implementation", in Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (Micro-Neuro 96), IEEE Computer-Press, Lausanne 12-14 Feb. 1996, pp. 265-270.

[6] R. Dogaru, I. Dogaru, "An efficient finite precision RBF-Mneural network architecture using support vectors", in Neural Network Applications in Electrical Engineering, Sept. 2010, pp. 127-130

[7] R. Dogaru, "A hardware oriented classifier with simple constructive training based on support vectors", in Proceedings of CSCS-16, the 16th Int'l Conference on Control Systems and Computer Science, May 22- 26, 2007, Bucharest, vol. 1, pp. 415-418.

[8] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, "Extreme Learning Machine: A new learning scheme of feedforward neural networks", in Neural Networks, 2004. Proceedings 2004 IEEE International Joint Conference on (vol. 2, pp. 985-990).

[9] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, Rui Zhang, "Extreme learning machine for regression and multiclass classification", in Systems, Man, and Cybernetics, IEEE Transactions, 2011 on (vol. 42, pp. 513-529)

[10] Guang-Bin Huang, Chee-Kheong Siew, "Extreme learning machine: RBF network case", in Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th (vol. 2, pp. 1029-1036)

[11] Nan-Ying Liang, Guang-Bin Huang, P.Saratchandran, N.Sundarajan, "A fast and accurate online sequential learning algorithm for feedforward networks", published in Neural Networks, IEEE Transactions, Nov. 2006 (vol. 17, pp. 1411-1423).

[12] Guang-Bin Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks", in Neural Networks, IEEE Transactions, Mar. 2003 (vol. 14, pp. 278- 281).

[13] Guang-Bin Huang, L. Chen, C.-K. Siew, "Universal approximation using incremental feedforward networks with arbitrary input weights", Revised and resubmitted to IEEE Transactions on Neural Networks (2003).

[14] M. Bucurica and R. Dogaru, "A comparison between Extreme Learning Machine and Fast Support Vector Classifier", IEEE ECAI, pp. 1-4, 2015.

[15] S. F. Mahmood, M. H. Marhaban, F. Z. Rokhani, K. Samsudin and O. A. Arigbabu, "SVM–ELM: Pruning of Extreme Learning Machine with Support Vector Machines for Regression", Journal of Intelligent Systems, pp. 1-12, 2015.

[16] L. Zhang and D. Zhang, "SVM and ELM: Who Wins? Object Recognition with Deep Convolutional Features from ImageNet", Computer Vision and Pattern Recognition, pp. 1-6, 2015.

[17] J. Chorowski, J. Wang and J. M. Zurada, "Review and performance comparison of SVM and ELM based

classifiers", Journal of Neurocomputing, vol. 128, pp. 507–516, 2014.
[18] G. Huang, S. Song, J. N. D. Gupta and C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. Vol. 44, pp.1–13, 2014.
[19] An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition.
[20] Neural Networks at Pacific Northwest National Laboratory.

Author Profile

Alka Kumari is pursuing Master of Technology in Computer Science & Engineering from PTU regional centre at SSIET Derabassi. Her areas of research are machine learning, data mining.