RESEARCH ARTICLE OPEN ACCESS

PSO optimized Feed Forward Neural Network for offline Signature Classification

Pratik R. Hajare* Dr. Narendra G. Bawane**
*Research Scholar, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, India
**Principal, S. B. Jain Institute of Technology, Management & Research, Nagpur, Maharashtra, India

ABSTRACT
The paper is based on feed forward neural network (FFNN) optimization by particle swarm intelligence (PSI) used to provide initial weights and biases to train neural network. Once the weights and biases are found using Particle swarm optimization (PSO) with neural network used as training algorithm for specified epoch, the same are used to train the neural network for training and classification of benchmark problems. Further the approach is tested for offline signature classifications. A comparison is made between normal FFNN with random weights and biases and FFNN with particle swarm optimized weights and biases. Firstly, the performance is tested on two benchmark databases for neural network, The Breast Cancer Database and the Diabetic Database. Result shows that neural network performs better with initial weights and biases obtained by Particle Swarm optimization. The network converges faster with PSO obtained initial weights and biases for FFNN and classification accuracy is increased.

Keywords- Particle swarm intelligence, feed forward Neural Network, Backpropagation, convergence, benchmark, realistic problems, prediction error, local minima, local maxima, offline, signature.

I. INTRODUCTION

In many classification applications Neural networks are widely used where data is non linear and diverse. For better result, the networks are tuned by adjusting various network parameters. Research shows that different researchers are still at work related to find optimum network architecture that is topology, layer transfer functions, initial guess to weights and biases, training algorithm, minimum gradient, epochs etc. Training neural networks for better accuracy is a cumbersome and tedious job. One cannot expect good results at the first stroke when a neural network is trained. And the network has to be trained multiple times unless expected output with minimum mean square error is obtained. Also a compromise has to be made most of the time between expected output and training samples.

When the solution represents the network topological information but not the weight values, a network with a full set of weights must be used to calculate the training error for the cost function. This is often done by performing a random initialization of weights and by training the network using one of the most commonly used learning algorithms, such as Backpropagation. This strategy may lead to noisy fitness evaluation, since different weights and biases as starting point initializations and training parameters can produce different results for the same topology.

Swarm intelligence [10] algorithms draw inspiration from the collective behavior and emergent intelligence that arise in socially organized populations. They have been designed primarily to address problems that cannot be tackled through traditional optimization algorithms. Such problems are characterized by discontinuities, lack of derivative information, noisy function values and disjoint search spaces [12, 14].

The general purpose optimization method known as Particle Swarm Optimization (PSO) [2] is due to Kennedy, Eberhart and Shi and works by maintaining a swarm of particles that move around in the search-space influenced by the improvements discovered by the other particles. The advantage of using an optimization method such as PSO is that it does not use the gradient of the problem to be optimized, so the method can be readily employed for a host of optimization problems and with other optimization techniques. This is especially useful when the gradient is too laborious or even impossible to derive.

Particle swarm optimization (PSO) is a computational method that optimal solution [3, 4]. PSO optimizes a problem by having a population of candidate solutions, and moving these particles around in the search - space according to simple mathematical formulae. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space [9].

Simultaneous optimization of neural network weights and biases is an interesting approach for the generation of efficient networks. In this case, each point in the search space is a fully specified neural network with complete weight and bias information, and the cost evaluation becomes more accurate. Wrong initial guess to weights and biases may lead to long learning and thus takes large amount of CPU time, the tendency of Backpropagation to get stuck and produce wrong results, chance of getting overstep, and difficult to consider the best performance since every time the output changes with the initial weights and biases initializes [15]. Swarm intelligence is a relatively new category of stochastic, population-based optimization algorithms. These algorithms are closely related to evolutionary algorithms that are based on procedures that imitate natural evolution [13].

Many other evolutionary approaches can be seen in research papers to optimize Neural Networks. Already research shows artificial intelligence individually and in conjunction with other optimization techniques have solved many challenging task. This paper uses PSO as a learning algorithm to find initial starting weights and biases for FFNN and improves the classification rate for training and testing samples of benchmarking databases. Finding exact or optimum layers and number of neurons in the layers is again an issue, choice of proper neural network with optimum parameters is again based on trial and error. And most often it becomes a tedious job. For such case PSO with FFNN at least finds the weights and biases values that can help FFNN to push itself nearer the convergence [11] even with the wrong guess. In most cases, it is found that the Backpropagation tends to under or over a problem.

Each individual has its own signature different from others but a person cannot do exactly the same signature every time. Signature recognition finds its application in the field of passport verification system, provides authentication to a candidates in public examination from their signatures, credit cards, bank cheques. Therefore basic need of this type of application is accuracy and time. A person can do the signatures in different forms depending upon the type of pen available, space available to do the signature, angle of signature etc. Human signature is a biometric measure of person's identification. Many sectors like banks, official documents, receipts etc. use handwritten signature to secure and identify concerned person. Each individual has his own signature different with others but a person cannot do exactly the same signature every time so it is very important to recognize the signature. The signature verification problem aims to minimize the intrapersonal differences. Signature recognition can be categorized into following two parts: online and offline. Online handwritten signature recognition deals with automatic conversion of characters which are written on a special digitizer, tablet PC or PDA where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. In offline technique only scanned images of signatures are available. There are many methods developed for the signature recognition but the neural networks (NN) gives good performance in handwritten character recognition.

II. Signature Database

In this paper, signatures from 7 different individuals were acquired [25]. A separate algorithm is developed in MATLAB which generates different size, different angle of signatures from one signature. These signatures samples are then passed through some morphological operations like dilation, erosion and some global operation in different ranges.

1. After converting the signature into binary form, the height of signature was reduced by 15% and 30% and then the signature was cropped as per new height.

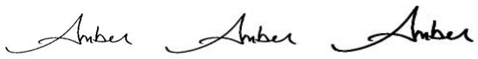
Signature samples (A) Original signature
(B) Reduced by 15% (C) Reduced by 30%

2. The area of the signature is defined as the number of black pixels. To reduce the area first we have to calculate black pixels of the signature which is done by subtracting the white pixel area from the total pixel area of the signature the total signature. The signature images are resized by scaling down the calculated area. For database creation the signature image area are reduced by 10%, 20%, 30%.

(A) Original signature (B) Reduced by 10% (C) Reduced by 20%
(D) Reduced by 30%

3. The morphological features like dilation and erosion is applied on the signature.

(A) Original signature (B) Dilated signature



(A) Original signature (B) 1^{st} Erode signature (C) 2^{nd} Erode signature

4. Last operation is performed on the signature is rotation. The signatures are rotated within 15^0,30^0,45^0.

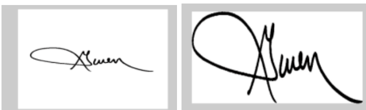
(A) Original signature (B) 15^0 rotated (C) 30^0 rotated (D) 45^0 rotated

By applying various global and morphological features in combination on a signature i.e. H/W ratio, area, dilation/erosion, rotation, a sample database of 192 images of a single signature is created. Overall for 7 such individual signatures a database consisting of 1344 sample images of the signatures are created. Out of this database 1050 samples images are used for training the neural network and 294 sample images are used for testing the ANN techniques.

5. Sample` Signatures 7 Specimens

III. Proposed system

(A) For higher accuracy the unwanted region was removed using cropping by finding extreme column and row and then the remaining part was resized them to some specified size [60 100]. And the 2 dimensional data was reshaped in a single column vector.



(A) Original Image (B) Cropped Image

(B) The column vector was decomposed (dimension reduced) by level 8 debauchees 3 mother wavelet.
(C) The wavelet coefficients were the normalized by dividing it by maximum wavelet coefficient value from all samples.
(D) The training and test data was then separated for training and classification.
(E) The neural network is designed using feed forward back propagation algorithm. Following parameters are used to design the neural network.

a) Numbers of Layers = 3
b) Total number of neurons = [8 12 1]
c) Transfer function = logsig, tansig, purelin
d) Training phase: No. of epochs = 1000, Goal = 1e-8

(F) The same parameters were used when the neural network was trained and used to classify test specimens using PSO optimized weights and biases.

Parameters used for Particle Swarm
% Particle Swarm constants
c1=2; % Constant
c2=2; % Constant
w=2; %Inertia weight
wMax=0.9; %Max inertia weight
wMin=0.5; %Min inertia weight [18]
% Velocity retardation factor
dt=0.8;

Number of particles was 30. Initial values for local and global best were assumed to be zeros. The training algorithm was levenberg-marquardt [16]. At each iteration the inertial weight was updated as,

%Update the w of PSO [8]
w=wMin-iteration*(wMax-wMin)/Max_iteration;

where iteration is the current iteration and Max_iteration is 100.

Velocities were updated as,

Vnew = w*Vold + c1 * rand () * (Pbest-P) + c2 * rand () * (Pgbest-P); where $0 < \text{rand}() < 1$

And the particles (weights and biases) were updated as,

Pnew = dt * Vnew + Pold;

A complete toolbox was designed for neural network with Backpropagation with a facility to select the network topology, layer transfer functions, and epochs with MSE (mean squared error) as parameter. The weights and biases arrays were initialized at random to be the particles position in the search space. The number of iterations for the neural network with PSO was fixed to 100.

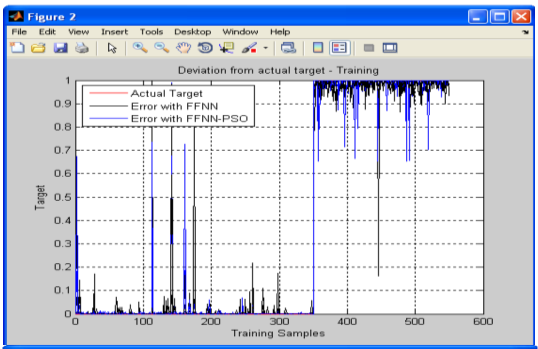
IV. Results

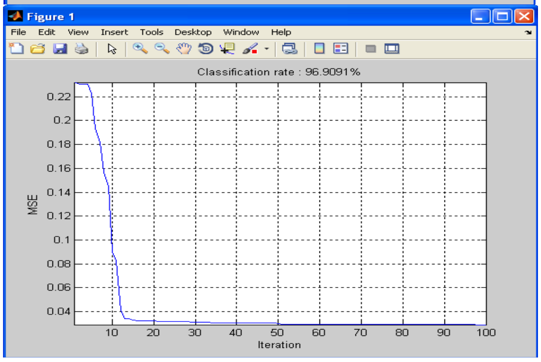
A) Breast Cancer Databases [19]

Diagnosis of breast cancer is to classify a tumor as either benign or malignant based on cell descriptions gathered by microscopic examination. There are 9 inputs, 2 outputs, 699 examples. All inputs are continuous; 65.5% of the examples are benign. This dataset was created based on the breast cancer Wisconsin problem dataset from the UCI repository of machine learning databases. The data was originally obtained from the University of Wisconsin Hospitals, Madison, from Dr. William H. Wolberg [1].

The data for 2 (2 and 4) classes have 458 and 241 samples corresponding to benign or malignant. Approximately 75% of the total sample was used for training and remaining 25% of the samples were used for testing. Further the classes target was set to 0 & 1 instead of 2 & 4 for log sigmoid function at the output layer of FFNN.

For neural network, the neurons in the hidden layer were selected to be 6 & 8 and in the output layer to be 1 with transfer functions log sigmoid for all three layers. The network was trained for 550 samples and then tested for remaining 149 samples inclusive of both classes.



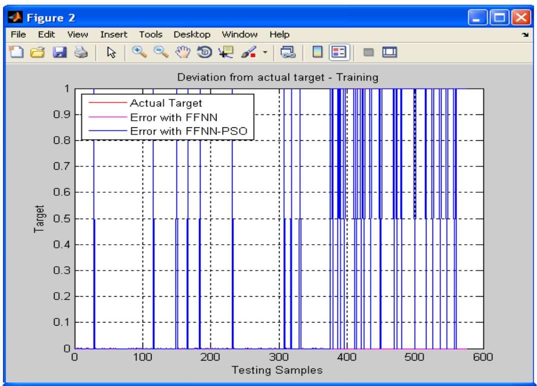


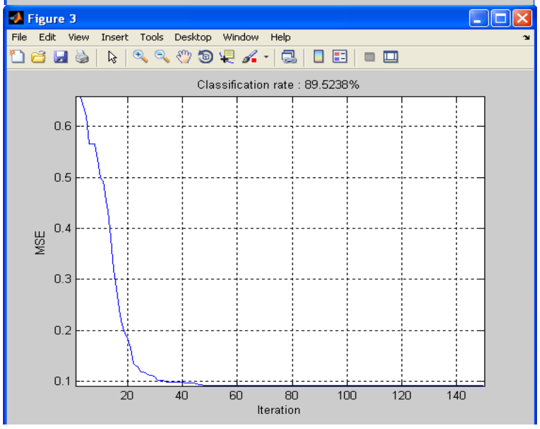
(B) Diabetic data Databases [19]

Diagnose diabetes of Pima Indians. Based on personal data (age, number of times pregnant) and the results of medical examinations (e.g. blood pressure, body mass index, result of glucose tolerance test, etc.), try to decide whether a Pima Indian individual is diabetes positive or not. There are 8 inputs, 2 outputs, 768 examples. All inputs are continuous. 65.1% of the examples are diabetes negative. Although there are no missing values in this dataset according to its documentation, there are several senseless 0 values. These most probably indicate missing data. Based on other samples the missing values were approximated and filled by values as per other samples. This dataset was created based on the Pima Indians diabetes problem dataset from the UCI repository of machine learning databases. Hence the classification rate was lowered but could have been increased, if data was complete.

The data for 2 (0 and 1) classes have 500 and 268 samples corresponding to positive or not. Approximately 75% of the total sample was used for training and remaining 25% of the samples were used for testing.

For neural network, the neurons in the hidden layer were selected to be 6 & 8 and in the output layer to be 1 with transfer functions log sigmoid, tansig and log sigmoid for the layers. The network was trained for 576 samples and then tested for remaining 192 samples inclusive of both classes.





The result shows that the classification by Neural Network in Breast and Diabetic database with PSO gained weights and biases have higher classification rate for training samples and have improved accuracy with test samples than normal FFNN with random weights and biases.

Parameters	**Benchmark Databases**	
	Breast Cancer data	Diabetic data
PSO iterations	100	100
MSE with PSO-NN	0.029146	0.17615
MSE with FFNN	0.00093935	0.34896
MSE with FFNN with PSO weights and biases	7.5965e-007	0.052083
Classification Rate with FFNN-training data	19.4545	20.4861
Classification Rate with PSO-NN-training data	80.5455	82.9861
Classification Rate with FFNN-testing data	94.302	80.7292
Classification Rate with PSO-NN-testing data	96.9091	89.5238
Network structure	[6 8 1]	[6 8 1]
Transfer functions	[logsig logsig logsig]	[logsig tansig logsig]

Sr. No.	Name of Neural Network	Number of samples for testing	Neurons required for training	Number of matched sample	Accuracy
1.	FFNN with BP	294	[8 12 1]	286	98%
2	FFNN-BP with PSO weights and Biases	294	[8 12 1]	294	100%

V. CONCLUSIONS & SCOPE

For classification (Breast and Diabetic Databases) problems, it is clear that the classification is more accurate when weights and biases are priory obtained by PSO and then given to FNN than what is achieved by random initialization of weights and biases with normal FFNN. When applied for offline signature verification, the training and testing data was classified with better accuracy than that of normal FFNN with Backpropagation. Also, it was found that the PSO based weights and biases converges the system faster. Also, Backpropagation requires mean square value smaller comparatively for classification, whereas when initial weights and biases were given using PSO based FFNN to FFNN; the mean square value was higher.

Thus PSO is an optimization tool for Neural Networks. Also the Network parameters and PSO parameters can be adjusted to find more accurate results. The same approach can be utilized for other networks for faster convergence by identifying constant or variable parameters, such as spread factor in Radial Basis networks. More specimen signatures can be acquired and samples can be generated based on blur, incompleteness, low contrast, high illumination, different human mood etc.

Acknowledgement

We thank to Mr. Ketan Taksande, Mtech., IIT Kharagpur, and Working at Cemetec, Pune, for his valuable guidance and for generating specimen database for offline signature. 192 specimens for each individual signature were created using program in MATLAB software.

REFERENCES

[1] L. Prechelt, Proben1—*A set of neural network benchmark problems and benchmarking rule*s, Karlsruhe, Germany, Tech. Rep. 21/94, Sep. 1994.
[2] J. Kennedy and R. Eberhart. *Particle swarm optimization*. In Proceedings of IEEE International Conference on Neural Networks, volume IV, pages 1942{1948, Perth, Australia, 1995.
[3] H.-P. Schwefel, *Evolution and Optimum Seeking*, Wiley, New York, 1995.
[4] D. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing and Oxford University Press, New York, 1997.
[5] Y. Shi and R.C. Eberhart. *A modified particle swarm optimizer*. In Proceedings of 1998 IEEE International Conference on Evolutionary Computation, pages 69-73, Anchorage, AK, USA, 1998.
[6] Y. Shi and R.C. Eberhart. *Parameter selection in particle swarm optimization*. In Proceedings of Evolutionary Programming VII (EP98), pages 591-600, 1998.
[7] R. S. Sexton, R. E. Dorsey, and J. D. Johnson, "*Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing*", Eur. J. Oper. Res., vol. 114, pp. 589–601, 1999.
[8] R.C. Eberhart and Y. Shi. *Comparing inertia weights and constriction factors in particle swarm optimization.* Proceedings of the 2000 Congress on Evolutionary Computation, 1:84-88, 2000.
[9] E. Bonabeau, M. Dorigo, *From Natural to Artificial Swarm Intelligence*, Oxford University Press, New York, 1999.
[10] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.
[11] M. Clerc and J. Kennedy. *The particle swarm - explosion, stability, and convergence in a multidimensional complex space*, IEEE Transactions on Evolutionary Computation, 6:58-73, 2002.
[12] K.E. Parsopoulos, M.N. Vrahatis, *Recent approaches to global optimization problems through particle swarm optimization, Natural Computing* 1 (2–3) (2002) 235–306.
[13] I.C. Trelea, *The particle swarm optimization algorithm: Convergence analysis and parameter selection*, Information Processing Letters 85 (2003) 317–325.
[14] K.E. Parsopoulos, M.N. Vrahatis, *On the computation of all global minimizers through particle swarm optimization*, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 211–224.
[15] M. Ventresca and H. R. Tizhoosh (2006). *Improving the convergence of back-propagation by opposite transfer functions*, in Proc. EEE WorldCongr. Comput. Intell. Vancouver, BC, Canada, 9527–9534.
[16] B. Subudhi, D. Jena (2008). *Differential Evolution and Levenberg Marquardt Trained Neural Network Scheme for Nonlinear System Identification*, Neural Processing Letters, 27(3), 285-296.

[17] Cai, X.J., Cui Z.H., Zeng, J.C. & Tan, Y. (2008). *Particle Swarm Optimization with Self adjusting Cognitive Selection Strategy, International Journal of Innovative Computing, Information and Control* (IJICIC), Vol.4, No.4, 943-952.

[18] Ziyu, T., & Dingxue, Z. (2009). *A modified particle swarm optimization with adaptive acceleration coefficients*. In Proceedings of the IEEE international conference on information processing (pp. 330–332). Washington DC: IEEE Computer Society.

[19] Lutz Prechelt, "*A set of Neural Network Benchmark Problems and Benchmarking rules*", technical report, September 30, 1994.

[20] Yann Le Cun, John S. Denker & Sara A. Solla, Optimal Brain Damage. In [22] pages 598-605, 1990.

[21] G.E.P. Box and G.M. Jenkins (1970), *Time Series Analysis, Forecasting and Control*, Holden Day, San Francisco.

[22] Suhail M. Odeh and Manal Khalil, "*Off-line signature verification and recognition: Neural Network Approach*," 2011 IEEE.

[23] Maya V. Karki, K. Indira, Dr. S. Sethu Selvi, "*Off-Line Signature Recognition and Verification using Neural Network*",2007 IEEE.

[24] Milton Roberto Heinen and Fernando Santos Osorio "*Handwritten Signature Authentication using Artificial Neural Networks*", Member, 2006IEEE.

[25] Priyanka Narkhede, Prof. V.R. Ingale "*Performance Comparison of ANN Techniques involved in Offline Signature Recognition on the basis of Morphological and Global Features*", Quality Up-Gradation in Engineering Science & Technology, BDCE 2014, ISSN: 978-81-923623-1-1, International Journal of Research in Engineering & Technology (IJRET).