ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

Neural Network Optimization by Swarm Intelligence for Prediction & Classification of Benchmark Datasets

Anuradha R. Kondelwar, Pratik R. Hajare

Assistant Professor, Dept. of Electronics & Telecommunication Engineering, Priyadarshini College of Engineering, Nagpur, India

Assistant Professor, Dept. of Electronics & Telecommunication Engineering, S. B. Jain Institute of Technology, Management & Research, Nagpur, India

ABSTRACT: Particle swarm optimization (PSO) is a computational method which optimizes a problem by having a population of candidate solutions, and moving these particles around in the search - space according to simple mathematical formulae. The search is opted for finding initial weights and biases for the feed forward neural network (FFNN). The combination of particle swarm optimization (PSO) and FFNN greatly help in fast convergence of FFNN in classification and prediction to various benchmark problems. The benchmarking databases for neural network contain various data sets from various different domains. All datasets represent realistic problems which could be called diagnosis tasks and all but one consist of real world data. Two such benchmarking problems, one for prediction and other for classification are selected in this paper to evaluate the performance of PSO with FFNN. Further a comparison is made between normal FFNN and PSO-FFNN in terms of mean square error. The result shows that using PSO minimizes the prediction and the classification error.

KEYWORDS: Particle swarm intelligence, feed forward Neural Network, Backpropagation, convergence, benchmark, realistic problems.

I. INTRODUCTION

Classification and prediction applications need a good predictor and classifier so that the error in prediction and classification can be minimized. Several tools such as Linear Discriminant analysis, Support vector machine, Fuzzy, Neural Network etc are used in many research. Different Neural networks are been widely used by researchers as a tool or the same is used with other artificial intelligence system as a hybrid in such applications. Still, researchers find difficulty in using neural networks for many aspects. They range from selecting proper type of network to finding network architecture that is topology, layer transfer functions, initial guess to weights and biases, training algorithm, minimum gradient etc. 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.

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

that are based on procedures that imitate natural evolution. Swarm intelligence 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. This is especially useful when the gradient is too laborious or even impossible to derive. The particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space [3, 4, and 9]. The PSO particles are assumed to be the weights and biases of the Neural Network and are updated after every iteration for minimum mean square error between the expected and the actual output. The iterations are fixed and the particles are displaced in space for the fixed number of iterations. The weights and biases at the last iteration corresponding to the global best particle are then taken as an initial guess to the neural network. Thus the PSO finds weights and biases which are enough nearer to the convergence point and the rest is attained by FFNN.
The benchmarking datasets [1] are all presented in the same simple format, using an attribute representation that can directly be used for neural network training. Along with the datasets, a set of rules for how to conduct and how to document neural network benchmarking is also provided. The purpose of the problem and rule collection is to give researchers easy access to data for the evaluation of their algorithms and networks and to make direct comparison of the published results feasible. The benchmark problems dataset considered in this paper are 1) Chaotic Mackey Glass Time Series for prediction and the 2) Diabetic Data for classification. The Mackey Series is available with MATLAB R2010a version as mgdata.dat file, while the Diabetic data was created based on the Pima Indians diabetes problem dataset from the UCI repository of machine learning databases. The data values in the set were complete and therefore a guess was made on the basis of other samples available with the database.

II. RELATED WORK

Various researches are oriented towards optimization of neural networks from the early stage when neural network gained widespread familiarity. Various research papers shows effective use of various neural networks for various applications such as perceptrons, feed forward neural networks, RBN, PNN, GRN, SOM's etc. Even some research shows hierarchical modelling of neural networks. The thirst increased when the effectiveness of neural networks decreased for large amount of data for real world applications in various engineering, communication, biomedical, stock market etc fields. Therefore researches were oriented in having better neural network structures with good learning algorithms that could effectively handle large amount of data with early convergence. As a result of which, neural networks was accompanied by customize networks wizards. But the requirement could not fulfil the thirst of the end users. Therefore, researchers tried to make modifications in the existing networks by using various techniques for various parameters available with existing neural networks.
The various parameters taken into account to improve the performance of neural networks were topology, learning algorithm, weights, transfer functions, network layers, layer neurons, epochs, gradient, momentum etc and other random parameters which were either randomly initialized or experimentally chosen. But many of the techniques failed to devise a system for optimizing networks which could handle all sorts of applications. Various papers at conferences and journals shows dedicated applications for which their method of optimizing the network worked well but failed for other applications. The study seems to be diverged and all could not come on to a common base. Therefore, since the research were scattered, some benchmark problems for neural networks were devised so that any optimization technique thus implemented would be tested on a common platform. The benchmark problems were made publicly available so that researchers may be able to test their systems and prove their work.

III. THE PROPOSED SYSTEM

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

network with PSO was fixed to 100. For Mackey Series data, the neurons in the hidden layer were selected to be 4 and in the output layer to be 1 with transfer functions log sigmoid for both. The network was trained for 500 samples and then test for another 500 samples. For Diabetic data, 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. The weights and biases that is position of particles at the last iteration were then taken as initial weight and biases for feed forward neural network available with Matlab R2010a version with the following parameters with same network topology and transfer functions,

```
Net.trainParam.epochs=iteration=100;
Net.trainParam.goal=1e-6;
Net.trainParam.lr=0.1;
% 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
w=wMin-iteration*(wMaxwMin)/
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 < rand() < 1
And the particles (weights and biases) were updated as,
Pnew = dt * Vnew + Pold;

IV. RESULTS

Prediction – Mackey Series
500 samples were trained for Mackey series and another 500 samples were predicted. Samples were taken as x(t8), x(t-12), x(t-6) and x(t) and the predicted sample was x(t+6). Samples started from 101 to 600 and test samples started from 601 to 1100. 500 samples started from 101 and 601 respectively for train and test data in four columns. The result shows the predicted and the actual output for all train and test samples. Another figure plot shows the error in prediction by normal FFNN with random weights and biases and PSO gained weights and biases to FFNN.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

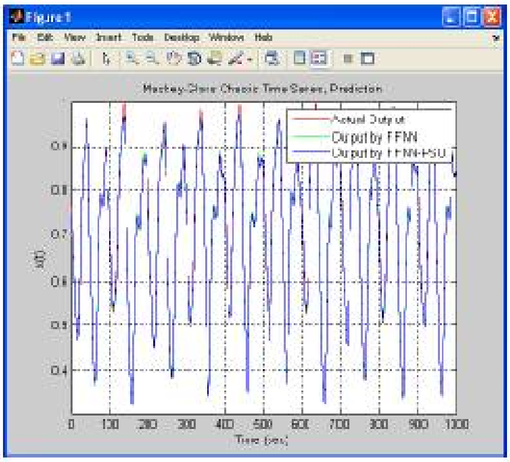

Fig. 1. Mackey series generation using FFNN and FFNN-PSO

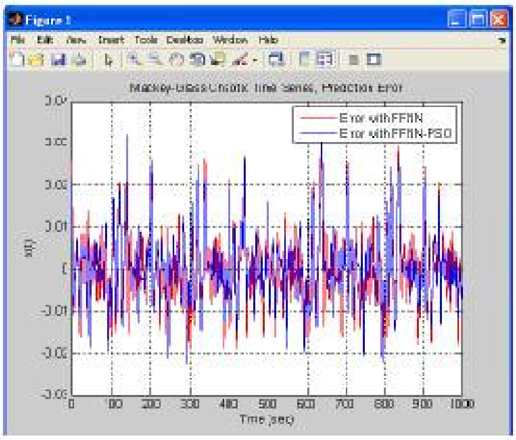

Fig. 2. Error with FFNN and FFNN-PSO for Mackey Series

Classification - Diabetic data
PSO is training FNN (Iteration = 100, MSE = 0.17615)
Training with Backpropagation
Mean Squared Error = 0.34896
Training with BP with initial weights by PSO
Mean Squared Error = 0.052083
Classification rate with BP = 20.4861
Classification rate with PSO obtained weights and Biases to FFNN = 82.9861
Classification rate with BP (Test) = 80.7292
Classification rate with PSO weights and Biases (Test) =89.5238

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

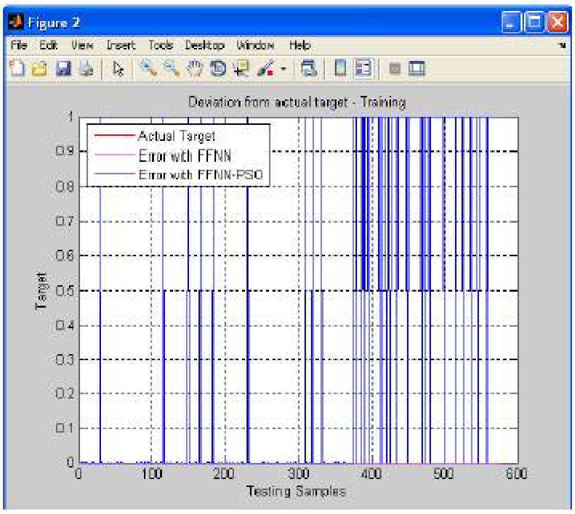

Fig. 3. Diabetic dataset classification error using FFNN and FFNN-PSO

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

V. CONCLUSIONS & SCOPE

Basically for prediction, Radial basis neural network are used and for classification feed forward neural network with back propagation is used. The paper concentrate on the effectiveness of PSO and only the FFNN is chosen for both the problems. Most often when a normal FFNN with Backpropagation is trained for an input data, the optimal output cannot be expected due to initial weights, epochs or network architecture at the first execution. And thus any of the parameters except weights and biases need to be altered and again training is required and the same is repeated until expected outcome is not obtained. But PSO has tendency to at least find a solution which may not be an optimum but the same can be given to the network for further training, where a normal FFNN fails to do so. Most often it is found that the PSO applied for weight and biases along with FFNN itself is sufficient to find a optimum or nearer optimum solution. For both benchmark problems, it is clear that the prediction and 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. Thus PSO is an optimization tool for Neural Networks. The same can be with Radial basis Neural Network for spread factor and other neural networks where either parameter are constant or randomly initialized. The classification rate would have been increased if the diabetic data values were not missing.

REFERENCES

1. L. Prechelt, Proben1—A set of neural network benchmark problems and benchmarking rules, 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. E. Bonabeau, M. Dorigo, From Natural to Artificial Swarm Intelligence, Oxford University Press, New York, 1999.
9. J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

10. 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.
11. K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, Natural Computing 1 (2–3) (2002) 235–306.
12. I.C. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317–325.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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
18. Computer Society.
19. G.E.P. Box and G.M. Jenkins (1970), Time Series Analysis, Forecasting and Control, Holden Day, San Francisco.