Training Artificial Neural Network using Particle Swarm Optimization Algorithm

Kaustav Choudhury

International Journal of Advanced Research in Computer Science and Software Engineering

Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Training Artificial Neural Network using Particle Swarm Optimization Algorithm

Argha Roy *
Dept. of CSE
Netaji Subhash Engineering College
West Bengal, India.

Diptam Dutta
Dept. of CSE
Heritage Institute of Technology
West Bengal, India.

Kaustav Choudhury
Dept. of CSE
Heritage Institute of Technology
West Bengal, India.

Abstract - In this paper, the adaptation of network weights using Particle Swarm Optimization (PSO) was proposed as a mechanism to improve the performance of Artificial Neural Network (ANN) in classification of IRIS dataset. Classification is a machine learning technique used to predict group membership for data instances. To simplify the problem of classification neural networks are being introduced. This paper focuses on IRIS plant classification using Neural Network. The problem concerns the identification of IRIS plant species on the basis of plant attribute measurements. Classification of IRIS data set would be discovering patterns from examining petal and sepal size of the IRIS plant and how the prediction was made from analyzing the pattern to form the class of IRIS plant. By using this pattern and classification, in future upcoming years the unknown data can be predicted more precisely. Artificial neural networks have been successfully applied to problems in pattern classification, function approximations, optimization, and associative memories. In this work, Multilayer feed- forward networks are trained using back propagation learning algorithm.

Keywords - Artificial neural network, particle swarm optimization, machine learning, back-propagation, IRIS.

I. Introduction

We view particle swarm optimization as a mid-level form of A-life or biologically derived algorithm, occupying the space in nature between evolutionary searches, which requires neural processing, which occurs on the order of milliseconds. Social optimization occurs in the time frame of ordinary experience - in fact, it is ordinary experience. In addition to its ties with A-life, particle swarm optimization has obvious ties with evolutionary computation. Conceptually, it seems to lie somewhere between genetic algorithms and evolutionary programming. Here we describe the use of back propagation neural networks (BPNN) towards the identification of iris plants on the basis of the following measurements: sepal length, sepal width, petal length, and petal width. There is a comparison of the fitness of neural networks with input data normalized by column, row, sigmoid, and column constrained sigmoid normalization. Also contained within the paper is an analysis of the performance results of back propagation neural networks with various numbers of hidden layer neurons, and differing number of cycles (epochs). The analysis of the performance of the neural networks is based on several criteria: incorrectly identified plants by training set (recall) and testing set (accuracy), specific error within incorrectly identified plants, overall data set error as tested, and class identification precision.

II. Literature Review

The most widely used method of training for feed-forward ANNs is back-propagation (BP) algorithm [10]. Feed-forward ANNs are commonly used for function approximation and pattern classifications. Back-propagation algorithm and its variations such as QuickProp [11] and RProp [12] are likely to reach local minima especially in case that the error surface is rugged. In addition, the efficiency of BP methods depends on the selection of appropriate learning parameters. The other training methods for feed-forward ANNs include those that are based on evolutionary computation and heuristic principles such as Genetic Algorithm (GA), and PSO.

A. Artificial Intelligence :

A precise definition of intelligence is unavailable. It is probably explained best by discussing some of the aspects. In general, intelligence has something to do with the process of knowledge and thinking, also called cognition. These mental processes are needed for, i.e., solving a mathematical problem or playing a game of chess. One needs to possess a certain intelligence to be able to do these tasks. Not only the deliberate thought processes are part of cognition, also the unconscious processes like perceiving and recognizing an object belong to it.

B. Particle swarm optimization (PSO):

Particle swarm optimization (PSO) [1] [2] is a stochastically global optimization method that belongs to the family of Swarm Intelligence and Artificial Life. Similar to artificial neural network (ANN) and Genetic Algorithms (GA) [7][8] which is the simplified models of the neural system & the natural selections of the evolutionary theory, PSO is based on the principles that flock of birds, school of fish, or swarm of bee's searches for food sources where at the beginning the perfect location is not known. However, they eventually they reach the best location of food source by means of communicating with each other.

C. **Artificial Neural Network (ANN):**

An Artificial Neural Network, often just called a neural network, is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.

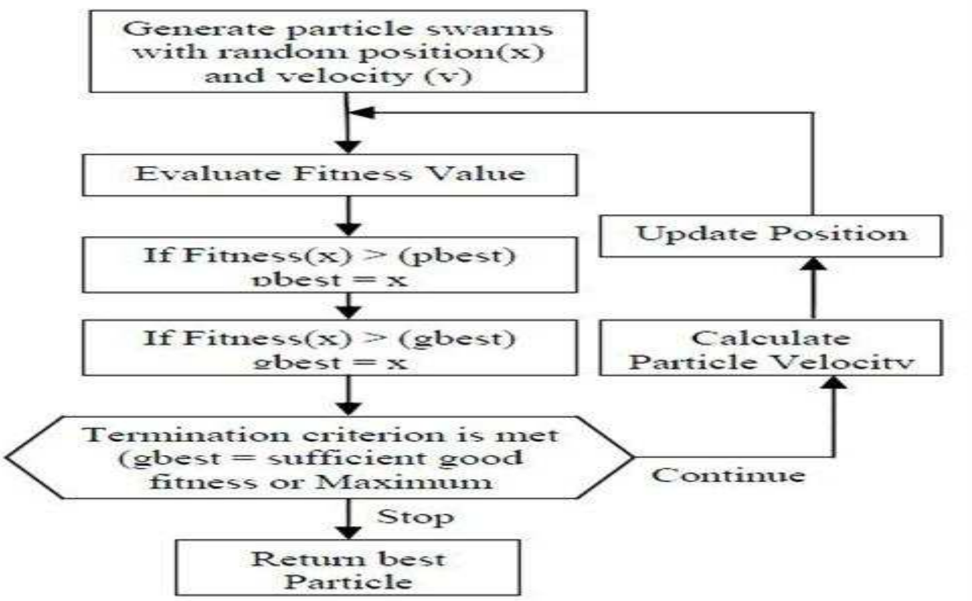
III. PSO-BACK PROPAGATION (BP) ALGORITHM

The PSO–BP is an optimization algorithm combining the PSO with the BP. Similar to the GA, the PSO algorithm is a global algorithm, which has a strong ability to find global optimistic result, this PSO algorithm, The BP algorithm, on the contrary, has a strong ability to find local optimistic result, but its ability to find the global optimistic result is weak. By combining the PSO with the BP, The fundamental idea for this hybrid algorithm is that at the beginning stage of searching for the optimum, the PSO is employed to accelerate the training speed. When the fitness function value has not changed for some generations, or value changed is smaller than a predefined number, the searching process is switched to gradient descending searching according to this heuristic knowledge. Similar to the APSO algorithm [7], the PSO–BP algorithm's searching process is also started from initializing a group of random particles. First, all the particles are updated according to the Equations. Until a new generation set of particles are generated, and then those new particles are used to search the global best position in the solution space. Finally the BP algorithm is used to search around the global optimum. In this way, this hybrid algorithm may find an optimum more quickly.

A. **Pseudo Code for the Algorithm:**

```
  For each particle
   Initialize particle
 END
 DO
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value (pbest) in history
        Set current value as the new pbest
End
    Choose the particle with the best fitness value of all the particles as gbest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    End
 While maximum iterations or minimum error criteria is not attained
```

B. **Flow Chart:**



IV. PROPOSED WORK

The proposed optimization algorithm combines the PSO with the back-propagation (BP). Similar to the GA, the PSO algorithm is a global algorithm, which has a strong ability to find global optimistic result, this PSO algorithm, The BP algorithm, on the contrary, has a strong ability to find local optimistic result, but its ability to find the global optimistic result is weak. By combining the PSO with the BP, The fundamental idea for this hybrid algorithm is that at the beginning stage of searching for the optimum, the PSO is employed to accelerate the training speed. When the fitness function value has not changed for some generations, or value changed is smaller than a predefined number, the searching process is switched to gradient descending searching according to this heuristic knowledge. The algorithm's searching process is also started from initializing a group of random particles. First, all the particles are updated according to the Equations. Until a new generation set of particles are generated, and then those new particles are used to search the global best position in the solution space. Finally the BP algorithm is used to search around the global optimum. In this way, this hybrid algorithm may find an optimum more quickly.

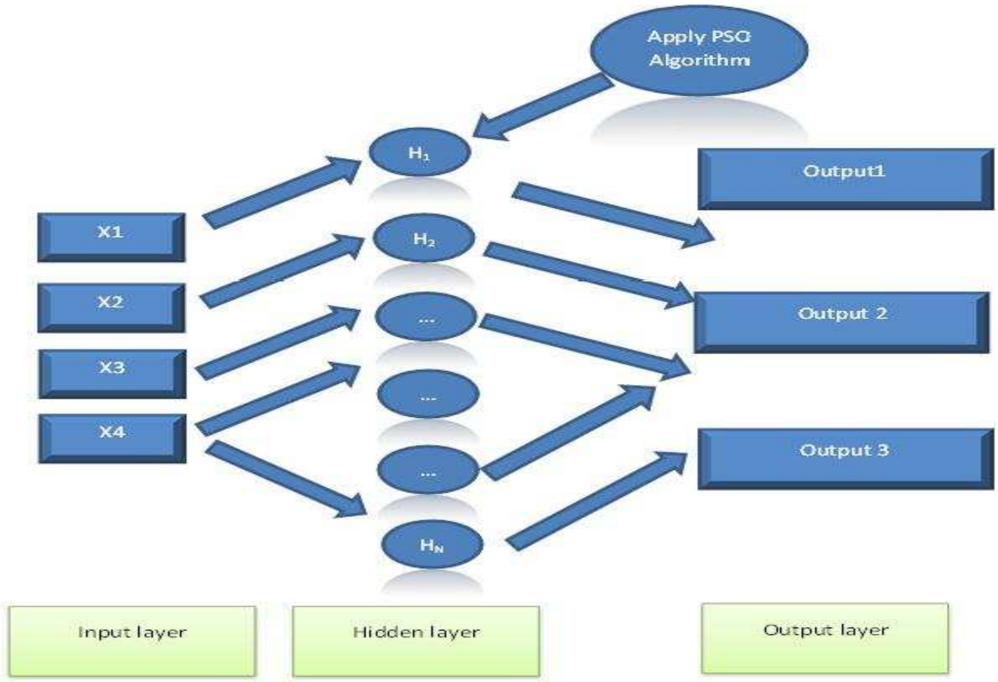


Fig 1: Proposed two layered feed forward neural network structure

V. RESULTS AND DISCUSSIONS

Different ranges of values are taken for x and y. for some specific ranges of x and y, we are analyzing different runs over each iterations. And by using MATLAB, we can easily find the difference between the particles. It is the language for technical computing. MATLAB is the easiest language for solving mathematical equations or these type of functions as compared to C programming, by which we can easily implement different functions. MATLAB is very time consuming. The fittest network architecture identified used column normalization, 54 cycles, 1 hidden layer with 6 hidden layer neurons, a step width of 0.15, a maximum non-propagated error of 0.1, and a value of 1 for the number of update steps.
We analyze the data using specific value given in IRIS dataset (sample provided 5 dataset shown)

X1	5.1	4.9	4.7	4.6	5
X2	3.5	3	3.2	3.1	3.6
X3	1.4	1.4	1.3	1.5	1.4
X4	0.2	0.2	0.2	0.2	0.2

Table 1: Sample IRIS Dataset

To get the output in a binary pattern we need to normalize the output value.

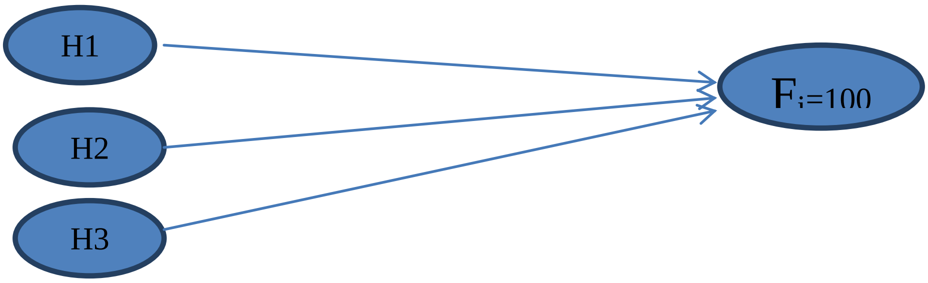


Fig 2: Process Normalization

A. **Output: Fi-Xai**

Where,
Fi= final weighted average
Xai (threshold function) = 0.5 (defined)

Now if,

Output =1, (Fi-Xai) >= 0
=0, (Fi-Xai) < 0

Thus the final output result takes the shape of

Setosa	010
Versicolor	100
Virginnica	001

Table2: Output Pattern

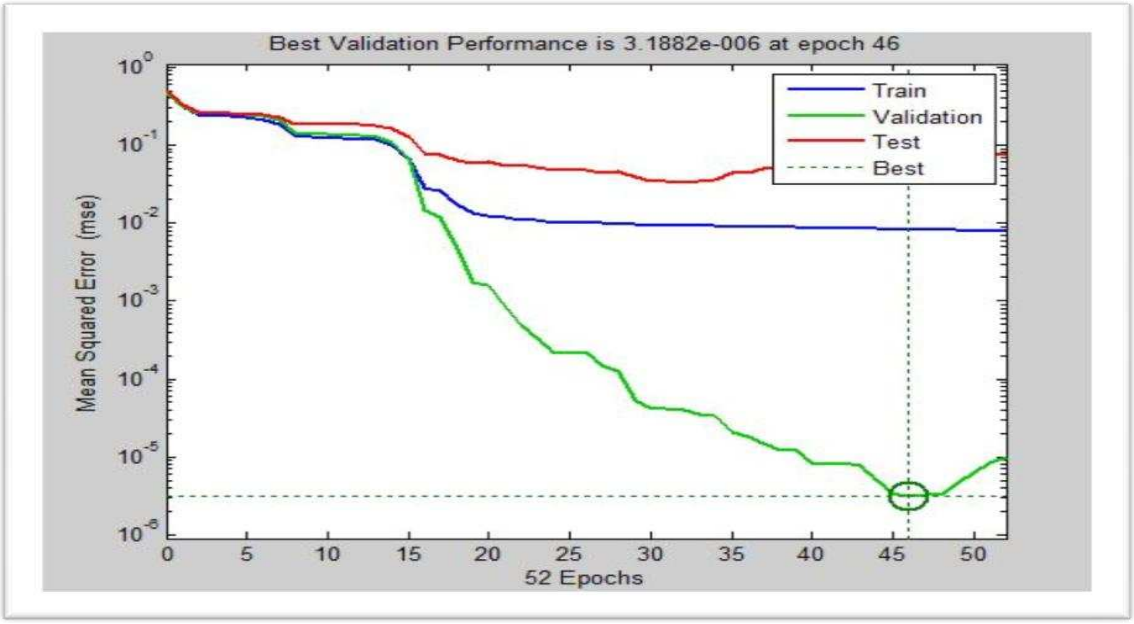
B. **Weight calculation:**

The Constant factor here taken as C1 =1, to calculate [6] [10] the weighted average value: H[i] = Hij *X[i]
Where,
0<=i<=150
0<=j<=5
F[i] = Wij * H[j]
Where,
0<=i<=150
0<=j<=5

C. **Classification performance:**

As shown in this plot at the Epoch 46 the validation performance returns less Mean square Error. Mean square error is the average square between output & target. The projected result for 54 Epoch we get the test data matrix with the accuracy rate of classified pattern of 97.3%

Fig 3: Plot of error per iteration



VI. CONCLUSION

Particle swarm optimization is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions. The adjustment p_i toward and p_g by the particle swarm optimizer is conceptually similar to the crossover operation utilized by genetic algorithms. It uses the concept of fitness, as do all evolutionary computation paradigms. Unique to the concept of particle swarm optimization is flying potential solutions through hyperspace, accelerating toward "better" solutions. In this simulation, we demonstrated the efficiency that this method possesses. Lastly, this method can be employed in training in various ANNs with different topologies.

REFERENCES

[1] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks.

[2] Kennedy, J.; Eberhart, R.C. (2001). Swarm Intelligence. Morgan Kaufmann.

[3] Poli, R. (2008). "Analysis of the publications on the applications of particle swarm optimisation". Journal of Artificial Evolution and Applications 2008.

[4] Shi, Y.; Eberhart, R.C. (1998). "Parameter selection in particle swarm optimization". Proceedings of Evolutionary Programming.

[5] Pedersen, M.E.H. (2010). "Good parameters for particle swarm optimization".

[6] Trelea, I.C. (2003). "The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection". Information Processing Letters.

[7] Zhang, G.P., 2000. Neural networks for classification: a survey. IEEE Transactions on Systems Man and Cybernetics.

[8] Rudolph, G., 1997. Local convergence rates of simple evolutionary algorithms with cauchy mutations.

[9] Liu, H.B., Tang, Y.Y., Meng, J., Ji, Y., 2004. Neural networks learning using vbest model particle swarm optimization.

[10] HechtNelso R., 1989, back propagation neural network

[11] Scott E. Fahlman: An Empirical Study of Learning Speed in Back-Propagation Networks, September 1988

[12] Rprop – Description and Implementation Details Martin Riedmiller, 1994. Technical report

