Hybrid Heuristic Optimization for Benchmark Datasets

Pravin Kshirsagar
Research Scholar,
Department of ElectronicsEngineering,
Rajiv Gandhi College of
Engineering and Technology,
Chandrapur (M.S.) India

Sudhir Akojwar
Senior Member IEEE,
Professor, Department of
Electronics Engineering, Rajiv Gandhi College of
Engineering and Technology,
Chandrapur (M.S.)

ABSTRACT

This paper introduces hybridization of particle swarm optimization (PSO) with genetic algorithm (GA) denoted as PSO+GA provides an efficient approach which is used to solve non linear chaotic datasets. The proposed algorithm employed in probabilistic neural network(PNN) which is a variant of radial basic function artificial neural network (RBFANN) for finding precise value spread factor for accurate classification of chaotic time series. Hybridizing of particle swarm optimization (PSO) and genetic algorithm (GA) in social learning helps collective efficiency, robustness and global effectiveness. The hybrid approached which then is resulted in the integrated framework for complete determination of spread factor with evaluation parameters. The algorithm is tested on two benchmark problems and compared the performance with arbitrary spread factor of PNN. The results showed that the PSO+GA based heuristic optimization algorithm outperform in terms of higher classification and prediction accuracies with short computation time.

Keywords

Particle swarm optimization (PSO), probabilistic neural network (PNN), convergence, benchmark, genetic algorithm (GA), radial basic function artificial neural network (RBFANN).

1. INTRODUCTION

Classification and prediction has been the goal of many research activities in the last century is an important problem for human, arising due to the fear of unknown phenomena and calamities all around the infinitely large world with its many variables to show highly nonlinear and chaotic behavior. From the scrupulous review of the related research work, it is observed that no simple model is available for classification of chaotic time series. The work is focused on nonlinear time series classification which have been successfully used in many applications in wide area of research. The aim of the work is to determine precise spread of the RBFANN. Due to the rapid learning capability of RBFANN and the faster convergence of heuristic optimization algorithm helped to get promising results over the problems considered in the work as compared to standard RBFANN, the RBFANN performs better when center, variance and the weights of RBF are properly tuned. T he accuracy of RBFANN networks degrade if these parameters are not appropriately chosen. The RBFANN are employed in real-time for detecting all kinds of intrusions for the purpose of network security which is also employed as a pattern classification. The parameters here are optimized using the new hybrid method; it showed superior results to the Conventional RBF artificial neural network. The inertia weight was used to balance the local and global search Abilities of the PSO. They suggested and proved that including the factor with current velocity of the particle (w*vold) improved the local and global search. A large value of w is more appropriate for global search and a small value of the w can facilitate the local search. Further a decreasing value of w linearly yields better solution was suggested by Shi & Eberhart [3]. Weight was set to zero and used only when re-initialization was required. Constriction factor C1, C2 was used to improve convergence velocity which guaranteed convergence. The topology which the swarm uses is an active area of research, for complex problems a small neighborhood Works better to find the solution while for simple problems, a

large neighborhood performs better. The neighborhood was dynamically adjusted by selecting the closet particles in each generation. A unified PSO was implemented by combining local and global versions together. The fitness value of each particle in the neighborhood was considered when velocity was changed. Instead of all neighbors, a single neighbor having higher fitness value was selected to update a particle position. In the proposed algorithm the diversity was increased by incorporating crossover and mutation in PSO to maintain the best particle. This was done to overcome of getting trapped in local minimum. The evolutionary techniques were combined with PSO operators such as – mutation was used to control w, avoid particle collision and relocate the particle when they get close to each other. Concept of dividing a single population into several sub-populations has been effectively used to maintain and increase population diversity.

2. PARTICLE SWARM OPTIMIZATION

Eberhart and Kennedy in 1995 proposed the concept of particle swarm. The method first developed was based on individual particle to perform optimization but later on the algorithm was simplified and it was realized that the individuals here typically known as particles were actually performing optimization [9]. It was originally meant for simulating the social behavior of a bird flock. Particle Swarm Optimization is based on the social behavior of a colony of swarm of insects, such as ants, birds, fishes, termites, bees, and wasps. Particle basically meant a single bird in the flock or a bee in the colony of the bees. And PSO mimics their behavior. Each one in the swarm behaves in the distributed way using its own personal intelligence and the group intelligence collectively. Therefore, when one particle is able to find good path to the food, the other members of the team follow that particle irrespective of their distances from the good one [10] [11]. Optimizations based on these facts of swarm intelligence are called behaviorally inspired algorithms and algorithms such as genetic algorithms are called

Evolutionary algorithms, where the next generation is formed by mutating the older generations.

In PSO, the particles are initially scattered at random positions in the search-space, moving in random directions with different velocities. The direction of a particle is then gradually changed depending on the best previous positions of itself and its best neighbor, searching in their vicinity and wishing that discovering even better positions. The inertia weight controls the amount of recurrence in the particle's velocity so that no two particles moving in the search space are at the same position at any instant [12][14]. The particle previous own best position and the swarm's previous best position are the factors through which the particles communicate implicitly with one another. These particles are weighted by the stochastic variables and the user-defined behavioral parameters, both being initialized to proper values in the beginning. When the velocity is added to the particle position, the position of the particle changes in the search-space, regardless of any improvement to its fitness. Also enforcing search-space boundaries after a particle's position is update, it is also required to impose limitations on the distance
update, it is also required to impose limitations on the distance
the particle may move in only one step [13]. This is done by limiting a particle's velocity to the full dynamic range of the search-space, so the particle may at most moves from 1 search-space boundary to the other in single step

3. GENETIC ALGORITHM

John Holland proposed genetic algorithm is generally based on evolutionary ideas of natural selection of genes. The adaptive heuristic search algorithm was designed for replicate process in expected system required for evolution. Crossover, Mutation and selection are the main operator of genetic algorithm to explore group of possible solutions. The crossover and mutation process for GA was carried in a different way as follows,

a. The first two digits after decimal point were extracted from both the particles. E.g. suppose the position values of particles are 0.9472 and 0.3326, thus the first two digits 94 and 33 were extracted by multiplying the numbers by 100 and then taking only the integer part.

b. Now the numbers extracted were converted in binary. Two crossover schemes were provided; single and double crossover as, Single point Crossover

Single point Crossover

0	1	0	1	1	1	1	0

Performer

0	0	1	0	0	0	0	1

Non performer

0	0	1	0	1	1	1	0

New Non Performer – 45

Figure 1. - Single point crossover for non performer particle with the performer

For single point crossover the last four digits of non performer were changed as per the performer particle.

Double point crossover

0	1	0	1	1	1	1	0

Performer

0	0	0	0	1	1	0	1

Non performer

0	0	0	1	1	1	0	1

New Non Performer - 29

Figure 2. – Double point crossover for non performer particle with the performer

For double crossover the middle 4 bits were changed for the non performer particle.

c. The mutation process involved by changing a random bit after crossover from 0 to 1, and 1 to 0. The bit position to flip the bit was randomized for individual iteration.

d. To update the complete particle value by placing the two most significant digits, the original value 0.3326 was again multiplied by 100 resulting in 33.26. Then the integer part was extracted and subtracted as 33.26-33=0.26. The value was then divided by 100 to have 0.0026. The new value 45 (single crossover) was divided by 100 to have 0.45. Finally 0.0026 was added to 0.45 to result 0.4526.

e. If number of digits obtained after crossover and mutation are greater than three, then the most significant digit (first one out of three) is discarded.

f. The process was repeated for only 15 non performer particles and remaining particles (performer) were kept unchanged.

g. Thus 15 worst performer particles were updated and again combined with the 15 performer particles to sort them as earlier. The generation thus obtained was used for the next iteration using PSO.

4. THE PROPOSED SYSTEM

A comprehensive algorithm was designed for probabilistic neural network with a facility to select the best spread factor, with mean squared error as a performance parameter. The spread was initialized at random to be the particles location in the search space. The no. of iterations for the probabilistic neural network with PSO+GA was fixed to 25.

For implementation of an algorithm the various chaotic datasets are stored in a .mat file and are arranged as training vector for input ,target vector for input training vector, the test vector for output and test target vector for output and all these vectors are loaded at the beginning of an execution of program .Work is implemented as

1. Set the size of population denoted as N to be 30.
2. Set particles initial value randomly (spread value).
3. Calculate initial fitness value of each particle.
4. Create generalized regression neural artificial network with training input and expected output.
5. Simulate the network with test samples
6. Calculate deviation from expected output.
7. Calculate fitness value of each particle and store.
8. Sort total N particles in ascending order according to its fitness value. Top N/2 particles are called superior particles and remaining N/2 particles are called inferior particles
9. The fitness values of superior and inferior particles are real decimal four point floating number. The first two digits after decimal point are encoded into eight bit binary string, from which last four digits of superior particle is crossover into last four digits of inferior particle.
10. Next, Mutation is performed between resultant crossover fitness value of superior and inferior particle.
11. Step 8-10 is repeated upto the desired iteration to get best fitness value corresponding to best spread is saved.
12. PNN is trained with best spread; the input vector and its target vector and performance parameter are evaluated on training samples.
13. PNN is tested with the test samples and performance parameters are evaluated by comparing the actual values of tested PNN and the values of test target vector.

The spread that is position of particles at the last iteration were then taken as spread for probabilistic neural network available with Matlab R2010a version with the following parameters.

```
% Maximum Iterations

Net.trainParam.epochs =maxiter=25;

% Inertial weight factors

wmax=0.9;

wmin=0.5;

% Maximunm and Minimum Value of velocity

vmax=0.5;

vmin=0.001;

% Maximum and Minimum value for particle position -
spread

popmax=1;

popmin=0.001;

% Parameters for calculation of C1 and C2 Constants

c1=(c1f-c1i)*j/maxiter+c1i;

c2=(c2f-c2i)*j/maxiter+c2i;
```

Where c1f=1, c1i=2, c2f=2, c2i=1 and j is a current iteration value.

Number of particles was 30. Initial values for local and global best were assumed to be zeros.. At each iteration the inertial weight was updated as,

```
%Update the w of PSO

 w=wmax-(wmax-wmin)*j/maxiter;
```

where, j is the current iteration and Max_iteration is 25.

```
%Velocities were updated as,

v(i,:)= w*v(i,:)+(c1*rand*(pbest (i,:) -
pop(i,:))+c2*rand*(gbest-pop(i,:)))/2;

% The particles position were updated as,

pop(i,:)=pop(i,:) + v(i,:);

%Update the fitness value of the particle

spread (i)=pop(i,:);
```

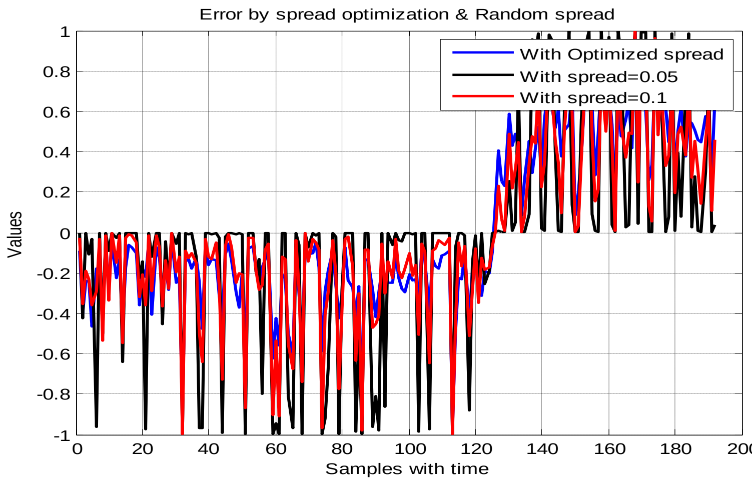
5. BENCHMARKING DATASETS

Two classification problems are taken into account for testing of designed algorithm. Basically this work shows the performance of PSO+GA when it is blended into PNN to find accurate spread factor for PNN. The mean square error obtained for classifications shows the effectiveness of the research work also the designed algorithm is implemented for the network in such a way that when dataset varies, most of the PSO does not require change in values that is they are generalized from research studies and experiments.

5.1 Classification Dataset and Result Diabetes dataset [17]

- Two class Problem – Benign and Malignant
- There are 768 examples with 8 inputs and 2 outputs.
- Benign = 65.1% (negative) approximately.
- The data for 2 (0 and 1) classes have 500 (normal) and 268 (affected) samples corresponding to positive or not.

Approximately training samples = 75% and testing = 25% of the total sample. The figure shows the error in classification by normal PNN with random spread and PSO+GA gained spread to PNN. The result shows the classified actual output for all train and test samples.



Mean Squared Error with optimized Spread factor - 30.2505

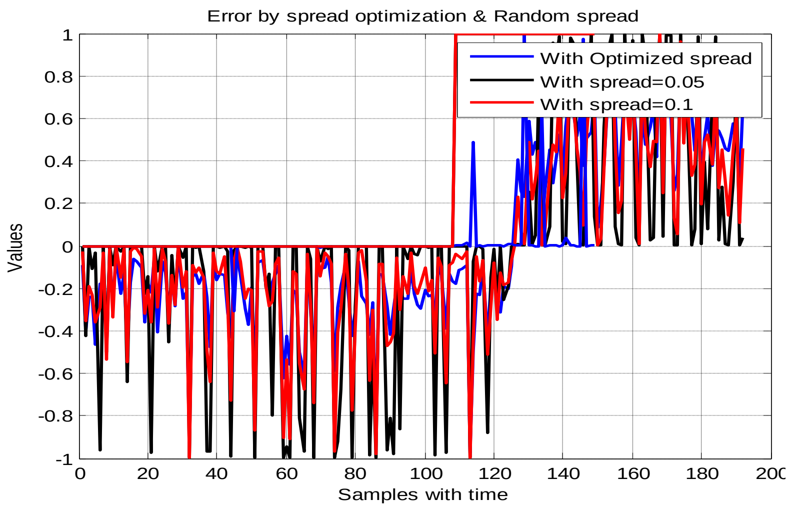
Mean Squared Error with random spread (0.05) - 48.5802

Mean Squared Error with random spread (0.1) - 32.7194

5.2 Breast Cancer Dataset [17]

- Two class Problem – benign and malignant
- The data for 2 (2 and 4) classes have 458 (normal) and 241 (affected) samples corresponding to benign or malignant.
- Approximately 75% for training and remaining 25% for testing of the total sample was used.

Some of the values in the dataset are missing and hence guess was to be made from the samples. The figure shows the error in classification by normal PNN with random spread and PSO+GA gained spread to PNN. The result shows the classified actual output for all train and test samples.

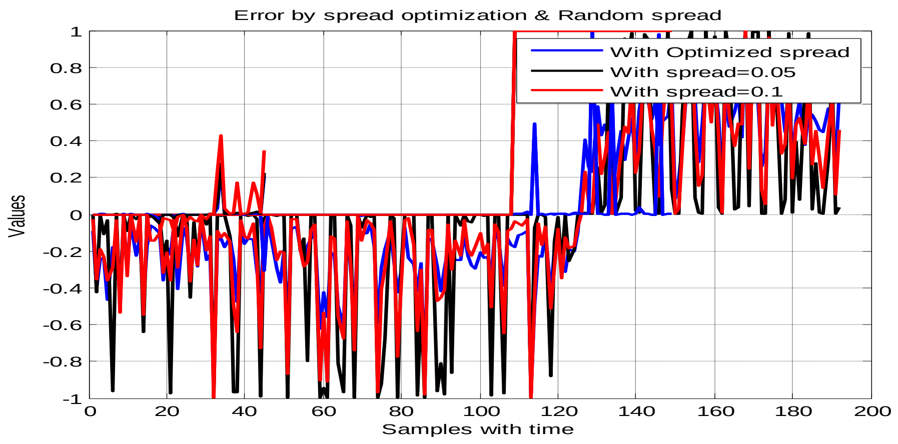


Mean Squared Error with PSO optimized Spread factor - 2.7454

Mean Squared Error with random spread (0.05) - 41
Mean Squared Error with random spread (0.1) – 41.12

5.3 Dataset for Iris [17]

This data set consists of Setosa, Versicolour, and Virginica belonging to types of irises that are with 4 features stored in a 150x4 array. The rows are the samples and the columns being the features of the irises. They are Sepal Length, Sepal Width, Petal Length and Petal Width. The lengths of the features are measured in millimeters. Type 1 is Setosa; type 2 is Virginica; and type 3 is Versicolour. Training samples are 105 and testing samples are 45. The figure shows the error in classification by normal PNN with random spread and PSO+GA gained spread to PNN. The result shows the classified actual output for all train and test samples.



Mean Squared Error with PSO optimised Spread factor - 0.10935

Mean Squared Error with spread=0.05 - 0.13688

Mean Squared Error with spread=0.1 - 0.49964

Table 1. Performance of algorithm on the basis of Precision (P), Recall (R)and F-Measure(F) for benchmark datasets

Breast Cancer Dataset		Breast Cancer Dataset		Iris Dataset		
Class 0	Class 1	Class 0	Class 1	Class 1	Class 2	Class 3
P=76.316	P =77.5	P=97.297	P =100	P=100	P=100	P=100
R=92.8	R=46.269	R=100	R=92.683	R=100	R=100	R=100
F =83.755	F =57.944	F=98.63	F =96.203	F- =100	F- =100	F- =100

6. CONCLUSIONS

Most often when a normal PNN with radial basis function neural network is trained for an input data, the optimal output cannot be expected due to number of radial basis functions or hidden units, center of hidden units and the spread factor at the first execution. The implemented work is used to find accurate spread factor for radial basis variant probabilistic neural network (PNN) so that the PNN will able to classify various benchmark problems with minimum error. For classification problems, it is clear that the classification is more accurate when spread is priory obtained by hybrid heuristic optimization and then given to PNN than what is achieved by random spread with normal PNN. Thus combination of PSO+GA with PNN is an optimization tool for neural networks. The same can be with Backpropagation for weights and biases.

7. REFERENCES

[1] Russell Eberhart and James Kennedy, "A New Optimizer Using Particle Swarm Theory", sixth international symposium on Micro Machine and Human Science, IEEE, volume 8, issue 95, pp. 39-43, 1995.

[2] Kennedy, I., Eberhart, R., Particle Swarm Optimization, Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, volume 3, issue 95,, pp. 1942- 1948, I995.

[3] Shi, Y., Eberhart, R., "Parameter Selection in Particle Swarm Optimization", Proceedings of the Seventh Annual Conference on Evolutionary Programming, pp. 591-601, Springer-Verlag, New York, 1998.

[4] R.C. Eberhart and Y. Shi.," Comparing inertia weights and constriction factors in particle swarm optimization", Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 84-88, 2000.

[5] M. Clerc and J. Kennedy," The particle swarm - explosion, stability, and convergence in a multidimensional complex space", IEEE Transactions on Evolutionary Computation, vol. 6, pp. 58-73, 2002.

[6] S. Ujjin and P. J. Bentley, "Particle Swarm Optimization Recommender System", IEEE Swarm Intelligence Symposium 2003, volume 24, issue 26, pp. 124-131, April 2003.

[7] F. Van den Bergh, "Particle Swarm Weight Initialization in Multi-Layer Perceptron Artificial Neural Networks", Development and Practice of Artificial Intelligence Techniques, Durban, South Africa, pp. 41-45, 1999.

[8] Zhiyong li, Wei Zhou, Bo Xu, Kenlili, "An ant colony genetic algorithm based on pheromone diffusion", fourth IEEE international conference on natural computation, volume 7, pp. 471-474, 2008.

[9] Thomas Stutzle and Holger H. Hoos, "Min-Max Ant System", Future generation computer systems, Elsevier, pp. 889-914, 2000.

[10] Marco Dorigo and Thomas Stutzle, "Ant colony optimization", A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 2003.

[11] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man and Cybernetics – Part B, volume 26, issue 1, pp. 29-41, 1996.

[12] B. S. Jung1, B. W. Karney and M. F. Lambert," Benchmark Tests of Evolutionary Algorithms: Mathematic Evaluation and Application to Water Distribution Systems", Journal of Environmental Informatics vol. 7, issue 1, pp. 24-35, 2006.

[13] Rahib H. Abiyev and Mustafa Tunay," Optimization of High-Dimensional Functions through Hypercube Evaluation", Computational Intelligence and Neuroscience, Hindavi, volume 2015, pp. 1-11.

[14] Lutz Prechelt, Fakultat Fur. 1994. A set of Neural Network Benchmark Problems and Benchmarking Rules. Technical Report 21/94, University of Germany.

[15] Neural Networks reference manual, pdf, www.mathworks.com.

[16] Box, G. E. P., & Jenkins, G. M. "Time Series Analysis, forecasting and control", San Fransisco, CA: Holden Day (1970).

[17] UC Irvine Machine Learning Repository (Center for Machine Learning and Intelligent Systems), http://archive.ics.uci.edu/ml/

[18] Pravin Kshirsagar and Sudhir Akojwar." Prediction of Neurological Disorders using PSO with GRNN" In the Proceeding of IEEE International Conference on Communication Networks, ICCN.2015.

[19] Pravin Kshirsagar and Sudhir Akojwar." Novel Approach for Classification and Prediction of Non Linear Chaotic Databases "" In the Proceeding of IEEE-ICEEOT-2016