International Journal of Emerging Technologies in Engineering Research (IJETER)
Volume 3, Issue 2, November (2015) www.ijeter.everscience.org

Particle Swarm Optimization Based Neural Network

Er. Jagdeep Kaur
Research Scholar, Computer Science Engineering, Desh Bhagat University, Mandigobindgarh, Punjab, India.

Prof. (Dr.) Jatinder Singh
Professor & Head, DAV University, Jalandhar, Punjab, India.

Abstract – Intelligence means capability of an individual to learn from the environment. There are many artificial techniques, neural networks, genetic algorithms, fuzzy logic which emphasize on detecting and describing relationships among vast amount of unrelated data. In this research, we have also used Evolutionary techniques which deal with the concept of global optima through coordination and cooperation between agents. The stress is on Particle Swarm Optimization based evolutionary technique and its hybrid with Neural networks.

Index Terms – Neural Networks, Evolutionary Techniques, Particle swarm optimization.

1. WHY NEURAL NETWORK

Neural networks serves as base to smart and complex vision systems for research and industrial applications. Much of the inspiration in the field of Neural Network came from Human Brain, the way it works in an intelligent manner to solve various complex problems.

All Neural Networks undergo training process, similar to the way a child learns from the surroundings by looking at the examples

Neural computing can never be a competitor to conventional computing. Instead, they work in conjunction with each other to bring about successful solutions.

In practice, NNs are applied where rules formed for expert system are not applicable. These are especially useful for classification and function approximation problems.

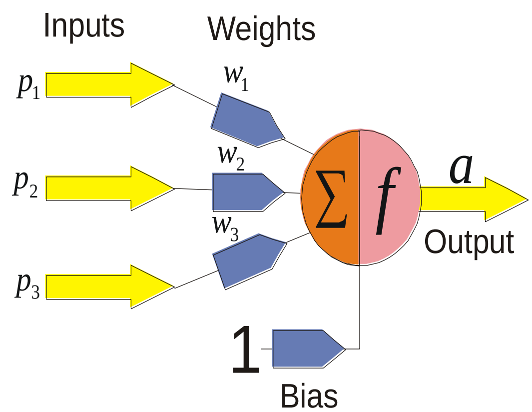
The major limitation with Neural Networks, is the difficulty in successfully applying Neural Networks to problems where the major concern is on manipulation of symbols and memory. It is not possible to implement a method for training Neural Networks that is not contained in the training data.

2. HOW NEURAL NETWORK WORKS

Neural computing basically consist of neurons which are connected together to form Neural Network. Neurons are arranged in layers.

In Neural Network each neuron takes one or more input, to which is already associated weight and produces an output. The neuron adds all inputs along with their weights and calculates output.

There is an activation function used in Neural Network. It is used to transform activation level of neurons into output signal.



$$a = f\left(p_1w_1 + p_2w_2 + p_3w_3 + b\right) = f\left(\sum p_iw_i + b\right)$$

Figure-1 Working Model of Neural Network

3. TRAINING METHODS

3.1. Supervised learning

In supervised training, a guide is always available. For all inputs the corresponding outputs are given. The network processes the inputs and compares its resulting outputs with the desired outputs. In case error occurs, they are moved back through the system, and the system adjusts the weights which control the network. This process keeps on repeating itself as the weights are continually changed. The set of data which is used in the training is called the training set. When the network is being trained, the same set of data is processed number of times as the connection weights are changed. Example of architectures- : Multilayer perceptrons.

3.2. Unsupervised learning

In unsupervised training, no guide is available. The network is provided with inputs but corresponding desired outputs are not available. It is responsibility of the system to decide the feature.

It is going to use for grouping of the input data. This is often referred to as self-organization or adaption. Example of architectures: Kohonen, ART.

4. PERCEPTRONS

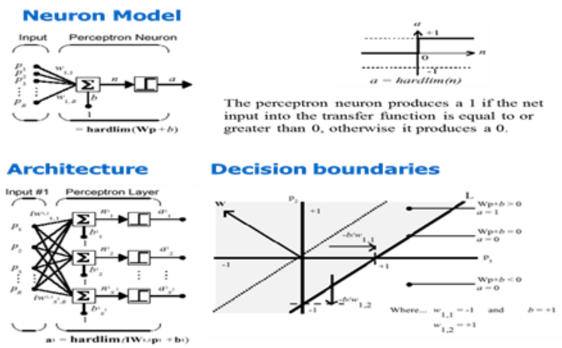


Figure-2: Architecture Of Neural Network

5. FEED FORWARD NEURAL NETWORK

The structure off a feed-forward Neural Network is shown, where in we three layers- Input, Hidden and Output layer have.

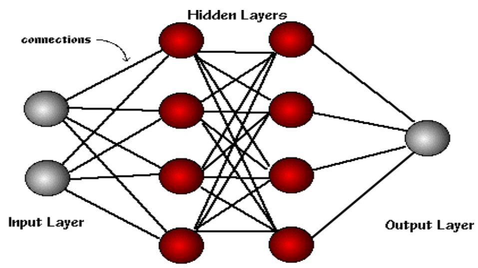


Figure-3. Feed Forward Neural Network

The learning rule keeps on changing the weights according to the input patterns that it is given with. This means, Artificial Neural Networks [1] follows learning rule that is learn by example.

When the information of the desired output is available we have supervised learning or learning with a teacher.

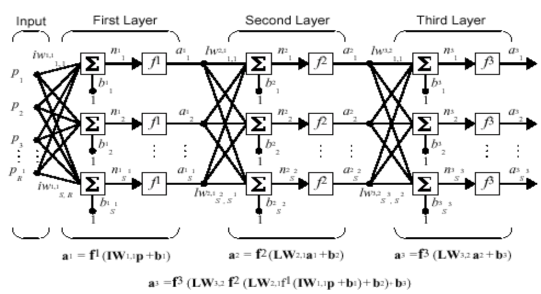


Figure-4. Artificial Neural Network

6. AN OVERVIEW OF BACK PROPOGATION

1. Training set is formed by collecting examples for training the network. We have number of cases, where in each case consists of a problem statement, which includes the input into the network and the corresponding solution, which represents the desired output obtained from the network.
2. The input data is feeded into the network through the input layer.
3. Each neuron in the network evaluates the input data with the resultant values slowly across the network, layer followed by the layer, until a result is generated by the output layer.
4. For a particular input, actual output of the network is compared to expected output. This results in an *error value.* The connection weights in the network are slowly adjusted, moving backwards from the output layer, through the hidden layer, and to the input layer. This keeps on happening until the correct output is produced. Fine tuning the weights across the network has the effect of teaching the network, given a particular input, how to produce the correct output, which simply means, the network *learns*.

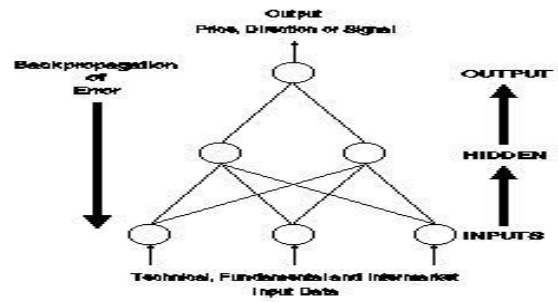


Figure-5. Overview of Back Propagation

7. THE LEARNING RULE

The delta rule [4] is often utilized by the back propagational neural networks.

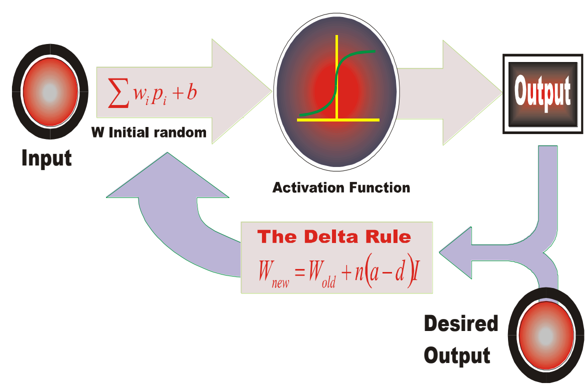


Figure-6. Learning rule

At the initialization, wwhen a neural network is presented with a pattern it makes a random guess, unaware what it might be. It then checks the length by which its answer was away from the actual one and makes an appropriate adjustment to its connection weights.

8. DELTA RULE

Back propagation performs a gradient descent approach towards a global minimum. The error surface itself is a hyperparaboloid but it is not smooth as is shown in the figure below. Indeed, in most problems, the solution space is not regular, it has many pits and hills which may cause the network to collapse in a local minimum, and will not lead to overall best solution.

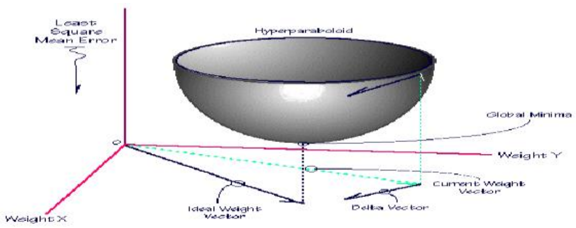


Figure-7. Delta rule

9. EARLY STOPPING

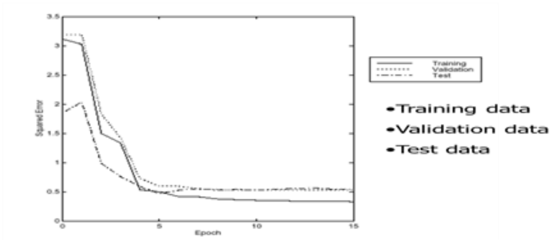


Figure-8. Stopping Criteria

10. OTHER ARCHITECTURES

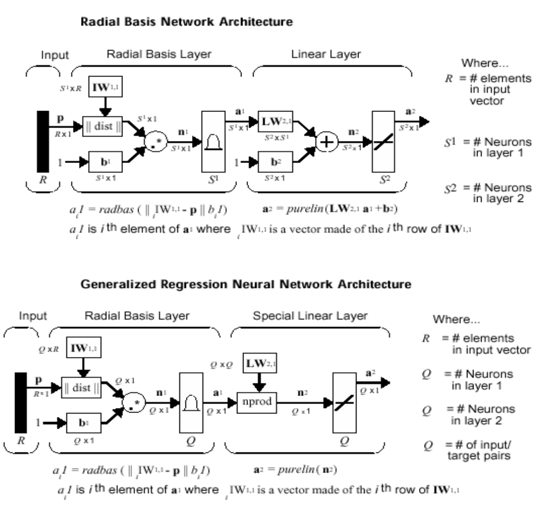


Figure- 9. Types of Neural Network Architecture

11. TIME DELAY NEURAL NETWORKS

A recurrent [10] neural network is one in which given a set of input units, the outputs received from the output layer is fed back to a set of input units. This is opposite to feed-forward networks, in which the outputs and inputs of units are connected in layers below or above it.

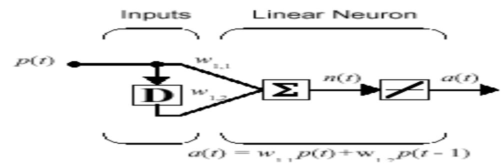


Figure- 10. Time Delay Neural Networks

Time Delay Neural networks are able to store information about time, and therefore they are suitable for forecasting and control applications: they have been used for predicting several types of time series, with negligible failure rate.

12. AUTOASSOCIATIVE NEURAL NETWORK

The autoassociative neural network, helps to retrieve data from small sample of itself. It is a special kind of MLP – which normally consists of two MLP networks, where back of one is connected to the other. The other important feature of auto-associative networks is that they are trained with a target data set that is similar to the input data set.

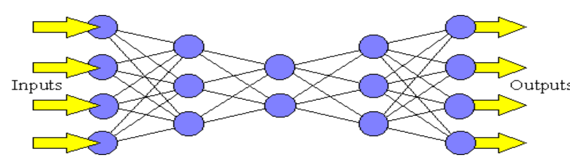


Figure-11. Auto associative Neural Network

In training, the network weights are adjusted every time until the outputs match the inputs, and the values assigned to the weights show the relationships between the various input data elements. This property of auto associative neural network is useful, for example, data validation: when invalid data is sent to the trained neural network, the learned relationships no longer exists and it is unable to generate the correct output. The match between the actual and correct outputs would reflect the closeness of the invalid data to valid values. Auto-associative neural networks are also used in data compression applications.

13. RECURRENT NETWORKS

13.1. Elman Networks

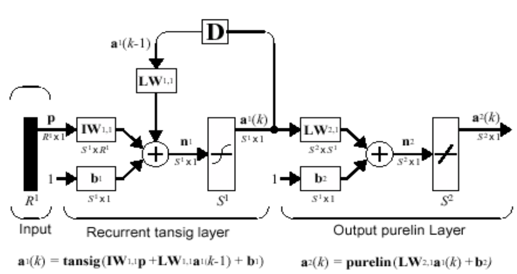


Figure-12. Elman Networks

13.2. Hopfield Networks

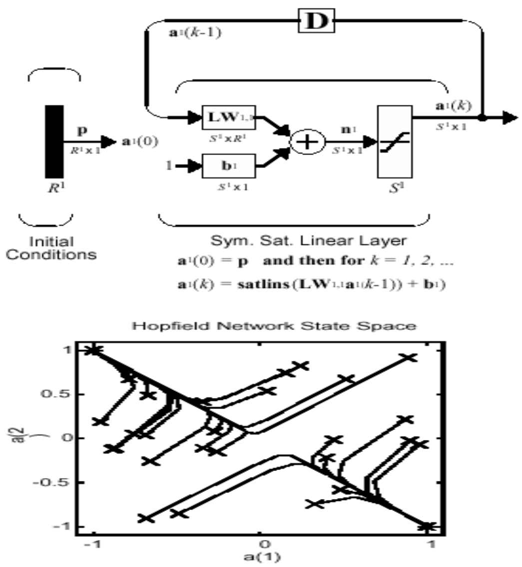


Figure-13. Hopfield Networks

14. SELF ORGANIZING MAPS (KOHONEN)

The Self-Organising Map or Kohonen network uses unsupervised learning.

Kohonen networks have a single layer of units and clusters of unit's link with different classes, having similar properties, during training. These cluster of units are present in the training data. The Kohonen network is useful in clustering [13] applications.

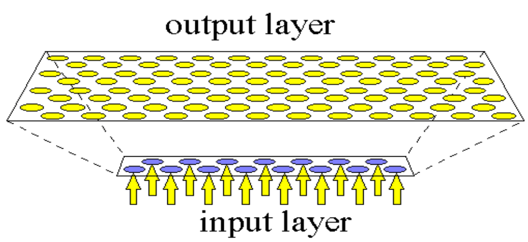


Figure- 14. Self-Organizing Maps (Kohonen)

15. NORMALIZATION

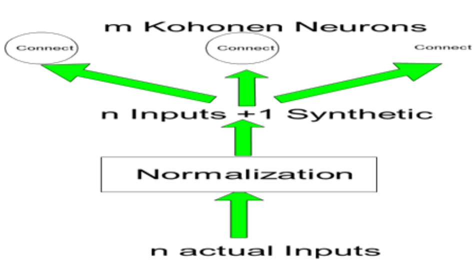


Figure- 15. Normalization

16. CHARACTERISTICS OF NEURAL NETWORK

Tendency to learn from experience: Network is Complex and it is difficult to solve problems, but lot of data is available to describe the problem.

Ability to generalize from examples: Network can interpolate from previous learning and give the correct response to unseen data.

Ability for Rapid applications development: Neural Networks are generic machines and are very much independent from domain knowledge.

Adaptability: Network has the capability of adapting to a changing environment, if it is properly designed.

Computational efficiency: In order to provide the training off a neural network a lot of computer power is required, a trained network demands almost nothing in recall mode.

Non-linearity: Network is not based on linear assumptions about the real word.

17. HOW NEURAL NETWORK APPLICATIONS ARE DIFFERENT

Data-Oriented: During Design process, there is a need to collect and analyse data and to train the neural network. This task is requires lot of effort and time, resources and time required is are frequently neglected. Impossible to specify full solution at the design stage: Therefore, In order to resolve design issues, it is necessary to build prototypes and experiment with them. This iterative development process can be difficult to control. Performance, and not the speed of processing, is the key issue: During the requirements analysis, design and test phases, performance issues need consideration. But, demonstrating that the performance meets the requirements can be relatively difficult.

These above three issues affect the following areas:

Project planning
Project management
Project documentation

18. PROJECT LIFE CYCLE

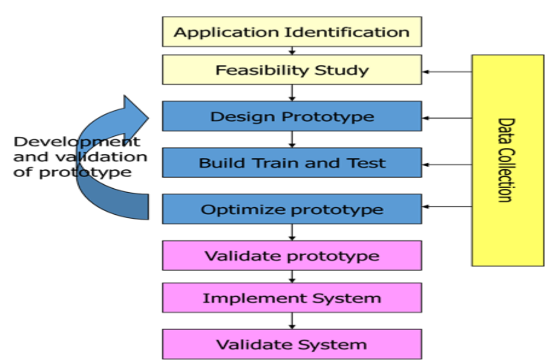


Figure- 16. Steps in Project Life Cycle

19. NEURAL NETWORKS IN REAL WORLD PROBLEMS

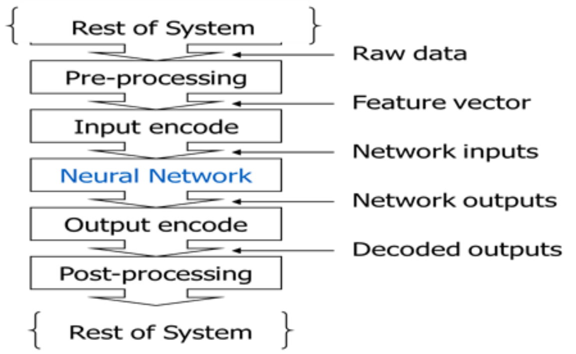


Figure-17. Neural Network in Real World Problems

20. PRE-PROCESSING

- Transform data to Neural Network Inputs
- By applying a mathematical or statistical function
- By encoding textual data present in the database
- Selection of the most relevant data and removing outlier
- Minimizing network inputs
- Feature extraction[8]
- Principal components analysis [1]
- Waveform / Image analysis
- Coding pre-processing data to network inputs

21. SWARM INTELLIGENCE

- An artificial intelligence (AI) technique based on the collective behavior in decentralized, self-organized systems
- Generally made up of agents who interact with each other and the environment
- No centralized control structures
- Work in groups, behavior found in nature

21.1. What is Swarm

A loosely structured collection of interacting agents
AGENTS-

- Individuals that belong to a group (but are not necessarily identical)
- They contribute to and benefit from the group
- They can recognize, communicate, and/or interact with each other
- The instinctive perception of swarms is a group of agents in motion – but that does not always have to be the case.
- A swarm is better understood if thought of as agents exhibiting a collective behavior

21.2. Particle Swarm Optimization

- Particles flying within the boundary are population based, means they reach the target through cooperation and coordination among themselves
- Search for an optimal solution using simple computations involving change in position and velocity.
- Developed in 1995 by Dr. Eberhart and Dr. Kennedy[5]
- Major Inspiration in development of Particle Swarm Optimization is Swarms of Bees, Flocks of Birds, Schools of Fish

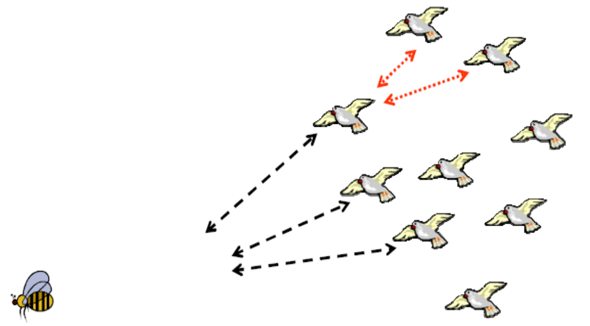


Figure- 18. Bird Flocking

Bird flocking is one of the best example of PSO in nature. One motive of the development of PSO was to model human social behavior.

21.2.1. What Happens in PSO

- Individuals in a population learn from previous experiences and the experiences of those around them
- The direction of movement is a function of:
- Current position
- Velocity (or in some models, probability)
- Location of individuals "best" success
- Location of neighbors "best" successes
- Therefore, each individual in a population will gradually move towards the "better" areas of the problem space
- Hence, the overall population moves towards "better" areas of the problem space

21.2.2. Algorithm of PSO

- Each particle (or agent) evaluates the function to maximize at each point it visits in spaces.
- Each agent remembers the best value of the function found so far by it (pbest) and its co-ordinates.
- Secondly, each agent know the globally best position that one member of the flock had found, and its value (gbest).
- Using the co-ordinates of pbest and gbest, each agent calculates its new velocity as:

- $vi = vi + ci * xrand(\ \ ) * x(pbestxi - presentxi) + c2 * xrand(\ \ ) * x(gbestx - presentxi)\ where\ 0 < rand(\ \ ) < 1$ … Equation-1
- $presentxi = presentxi + (vix\Delta t)$
- …Equation-II

```
Randomly generate an initial population
repeat
    for i = 1 to population_size do
        if f(present_i) < f(pbest)
            then pbest = present_i;
        gbest = best(pbest);
        for d =1 to dimensions do
            velocity_update();
            position_update();
        end
    end
until termination criterion is met.
```

Figure- 19- Algorithm of PSO

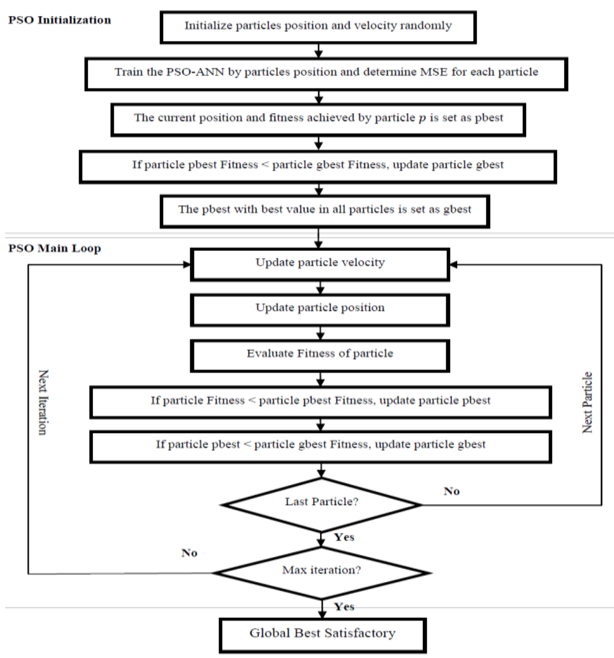


Figure- 20- PSO Flow Chart

22. PSO BASED NEURAL NETWORK

1) Initialize position and velocity of all the particles randomly in the d-dimension space.

2) Training the PSO-ANN by using the particles position and determine MSE (particle fitness) for each particle.

3) The current position and fitness achieved by particle p is set as its best history amount, also called the personal best (pbest).The pbest with best value in all particles are set as global best (gbest).

4) Change the velocity of the particle according to the given Equation.

$$vi[t+1] = wvid[t] + c1r1[pid(t) - xid(t)] + c2r2[gd(t) - xid(t)]\quad where\ 0 < r1 < 1$$ …Equation-III

5) Update particle position by adding the calculated velocity value to the current position value according to the Equation given below-

$$xid(t+1) = xid(t) + vid(t+1)$$

…Equation-IV

Where, a d-dimensional vector in problem space xid=($xi1$,$xi2$,…,xid) represents the position of each particle, vid= ($vi1$,$vi2$,…,vid) is the velocity of the ith particle,pid=($pi1$,…,pid) is the best position encountered by ith particle (pbest), gd shows the best position found by any member in the entire swarm population (gbest), t is iteration counter; $c1$, $c2$ are acceleration coefficients and $r1$, $r2$ are two similar random numbers in [0, 1].

6) Using the new sets of positions to generate new learning error.

7) Comparing the MSE of each particle with its pbest MSE then updating pbest, if the current MSE is lower than the pbest MSE.

8) Finding the minimum calculated MSE in the swarm then comparing it by the global best MSE then updating gbest, if the minimal MSE is lower than gbest MSE.

9) The optimization output is based on gbest position value. The iteration loop continues until reaching the MSE of the gbest lower than the desirable threshold or a maximum iteration number. The gbest weights are used as the training results when the iteration is finished.

Parameters used for running PSO are shown in Table 1:

Table 3. Parameters used for running PSO	
Parameter	Value
No. of population	5
No. of dimensions	2
C1,C2	1.49618
Inertia weight	1
Inertia Weight Damping Ratio	0.9
Search space range	(-1,1)
Initial velocity	0.9
final velocity	0.8

Table-1 Parameters used in PSO

23. RESULTS

USER INPUT VALIDATION
Local Best (Lbest) PSO
35 iterations maximum

Position clamping inactive.
Velocity reset inactive.
Velocities clamped to 50% of the range on each dimension.
History, “ghist,” of global bests active.
History, “lhist,” of local bests active.
History, “phist,” of personal bests active.
History, “fhist,” of all function values active.
History, “xhist,” of all positions active.
History, “vhist,” of all velocities active.
1 trial(s)
5 particles
Social acceleration coefficient, c2: 1.49618
Inertia weight linearly varied from 0.9 to 0.4 per grouping.
Cognitive acceleration coefficient, c1: 1.49618
Griewangk: 2 dimensions
Symmetric Initialization: [-600,600]
Threshold required for success: 1e-005
“OnOff_Terminate_Upon_Success” inactive.
Are the displayed settings as you intended (Y or N)? Y

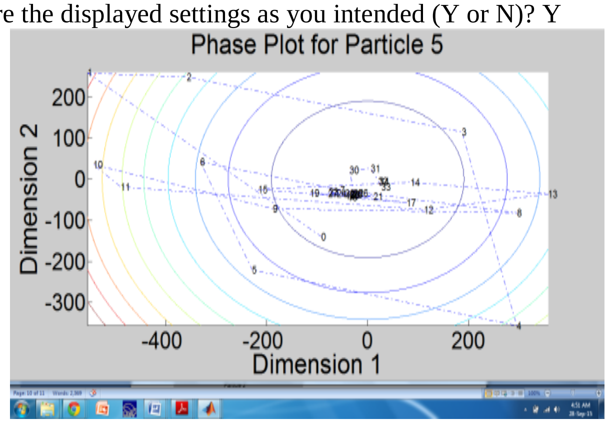
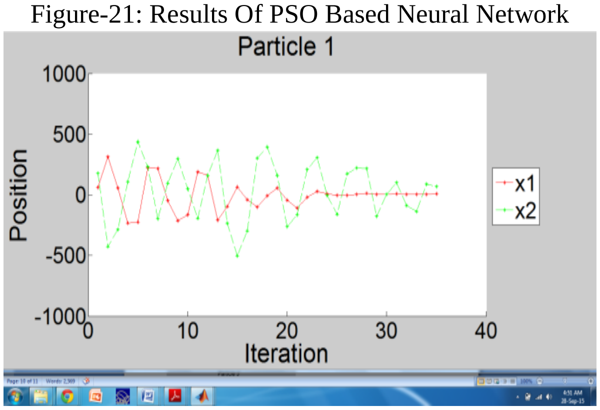
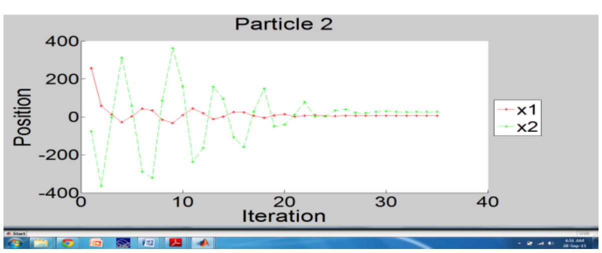


Figure-21: Results Of PSO Based Neural Network





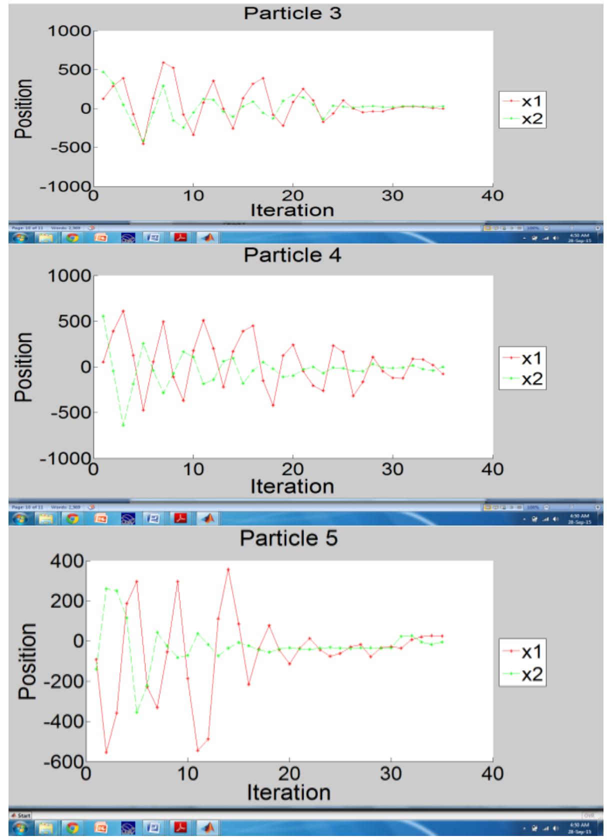


Figure-22: Iteration of Particles

24. FUTURE WORK

Advanced Evolutionary techniques besides Particle Swarm Optimization must be implemented along with other clustering techniques and their performance be compared in terms of accuracy and computational time.

REFERENCES

[1] Ashok, J. and Rajan, E.G., “Principal Component Analysis Based Image Recognition”, In the Proceedings of International Journal of Computer Science and Information Technologies, Vol. 1 (2) , pp- 44-50, 2010.

[2] Badaoui, H., Abdallaoui, A. and Chbaa, S. “Multilayer Perceptron and radial basis Function Network to predict the moisture”, Proceedings of International Journal of Innovation and Scientific Research, ISSN 2351-8014 Vol. 5 No. 1, pp. 55-67, Jul. 2014.

[3] ChandraShekhar, B. and Shoba, G.,”Classification Of documents using Kohonen’s Self-Organizing Maps”, Proceedings of International Journal of Computer Theory and Engineering, Vol. 1, No. 5, 1793-8201,pp- 610-613, December, 2009.

[4] Huk, M., “Backpropogation Generalized Delta rule for the selective attention Sigma-If Artificial Neural network”, International Journal of Applied Maths and Computer Science, Vol. 22, No. 2, pp-449–459, DOI: 10.2478/v10006-012-0034-5, 2012.

[5] Kennedy, J., Eberhart, R. and Shi, Y.,. Swarm Intelligence. Morgan Kaufmann, 1 edition, 2001.

[6] Kennedy, J. and Eberhart, R., “Particle Swarm Optimization”, Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942-1948, IEEE Press, 1995.

[7] Maind, B. Sonali and Wankar, P., “Research Paper on Basic of Artificial Neural Network”, In the Proceedings of International Journal on Recent

and Innovation Trends in Computing and Communication, Vol-2, Issue-1, pp-96-100, ISSN: 2321-8169, January, 2014.

[8] Rodriquez, R., Mexicano, A., Bila, J., Cervantes, S. and Ponce, R., "Feature extraction of Electrocardiogram signals by applying Adaptive Threshold and Principal Component analysis", In the Proceedings of Journal of Applied Research and Technology, Vol.13, Issue-2, pp: 261-269, April 2015.

[9] Singh, Y. and Chauhan, A., "Neural Networks in Data Mining", Proceedings of Journal of Theoretical and applied Informaton technology, pp. 37-42, 2005-2009 JATIT.

[10] https://en.wikipedia.org/wiki/Recurrent_neural_network

[11] Prince, N., "Real – World Applications of Neural Network", Proceedings of International Conference on Advancements in Information Technology With workshop of ICBMG 2011, IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore, 2011.

[12] https://en.wikipedia.org/wiki/Project_management#Project_development_stages

[13] V.K.Panchal, Harish Kundra, Jagdeep Kaur, "Comparative Study of Particle Swarm Optimization Based Unsupervised Clustering Techniques", Proceedings In International Journal of Computer Science and Network Security, Korea, October 2009.

[14] Jagdeep Kaur, Kirandeep Kaur, "Remote Image Classification Using Particle Swarm Optimization", Proceedings in International Journal of Emerging Technology and Advanced Engineering , Volume 2, Issue 7, pp-470,ISSN 2250-2459, July 2012.