**EARLY TERMINATION OF ANN USING ENTROPY**



**MINOR PROJECT**

Information Technology

III Year B.TECH., VI semester

Computer Science Engineering Department

Delhi Technological University

MENTOR:

Dr. SEBA SUSAN

MEMBERS:

ROHIT RANJAN 2K13/IT/075

UDYANT TALUJA 2K13/IT/092

SHIVANG RAI 2K13/IT/082

PRANAV AGGARWAL 2K13/IT/061

**ACKNOWLEDGEMENT**

THE COMPLETION OF ANY INTER-DISCIPLINARY PROJECT DEPENDS UPON COOPERATION, CO-ORDINATION AND COMBINED EFFORTS OF SEVERAL SOURCES OF KNOWLEDGE. WE ARE GRATEFUL TO, **DR SEBA SUSAN,** FOR HER EVEN WILLINGNESS TO GIVE US VALUABLE ADVICE

AND DIRECTIONS, WHENEVER WE APPROACHED HER WITH A PROBLEM .WE ARE THANKFUL TO HER FOR PROVIDING IMMENSE GUIDANCE FOR THIS PROJECT.

AND ALL THE STAFF MEMBERS OF INFORMATION TECHNOLOGY DEPARTMENT FOR THEIR IMMENSE COOPERATION AND MOTIVATION OF COMPLETING OUT THE PROJECT.

**CERTIFICATE**

THIS IS TO BE CERTIFIED THAT

**ROHIT RANJAN, UDYANT TALUJA, SHIVANG RAI &**

**PRANAV AGARWAL**

STUDENTS OF III YEAR B.TECH, V SEMESTER OF INFORMATION TECHNOLOGY DEPARTMENT,DELHI TECHNOLOGICAL UNIVERSITY HAVE COMPLETED THEIR MINOR PROJECT ENTITLED:

IMPLEMENTATION OF MULTI LAYER PERCEPTRON USING EVOLUTIONARY OPTIMISATION ALGORITHMS

THEY HAVE SUBMITTED THEIR PROJECT REPORT FOR THE PARTIAL FULFILMENT OF THE CURRICULUM OF THE DEGREE OF BACHELOR OF TECHNOLOGY FROM

DELHI TECHNOLOGICAL UNIVERSITY.

**Dr Seba Susan**

Assistant Prof. Deptt. of  Computer Science Engineering*.*

**ABSTRACT:**

In this project we are trying to optimize the artificial neural network by introducing the concept of entropy of weights. Instead of using traditional back propagation algorithm we have used particle swarm optimization algorithm which is way better and faster than back propagation. We also introduced the concept of mean cross entropy instead mean cross error.

We tried to establish a relationship between accuracy and entropy so as to minimise the no of epoochs and make the learning faster.

**INTRODUCTION:**

Artificial Neural Networks (ANNs) are programs designed to simulate the way a simple biological nervous system is believed to operate. They are based on simulated nerve cells or neurons, which are joined together in a variety of ways to form networks. These networks have the capacity to learn, memorize and create relationships amongst data. ANN is an information-processing paradigm, implemented in hardware or software that is modeled after the biological processes of the brain. It’s probably pretty obvious to that there are problems that are incredibly simple for a computer to solve, but difficult for you. Take the square root of 964,324, for example. A quick line of code produces the value 982, a number Processing computed in less than a millisecond. There are, on the other hand, problems that are incredibly simple for you or me to solve, but not so easy for a computer. Show any toddler a picture of a kitten or puppy and they’ll be able to tell you very quickly which one is which. Say hello and shake my hand one morning and you should be able to pick me out of a crowd of people the next day. But need a machine to perform one of these tasks? Scientists have already spent entire careers researching and implementing complex solutions.

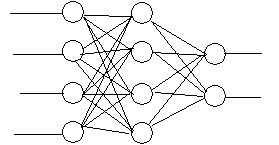
The most common application of neural networks in computing today is to perform one of these “easy-for-a-human, difficult-for-a-machine” tasks, often referred to as pattern recognition. Applications range from optical character recognition (turning printed or handwritten scans into digital text) to facial recognition.

***OBJECTIVES OF OUR PROJECT :***

* Understand the basic theory behind neural networks (backward propagation neural networks in particular)
* Understand how neural networks actually 'work'
* Implement a neural network that runs standard databases such as the IRIS
* Think about new possibilities of neural network programming

***ARCHITECTURE OF A NEURAL NETWORK :***

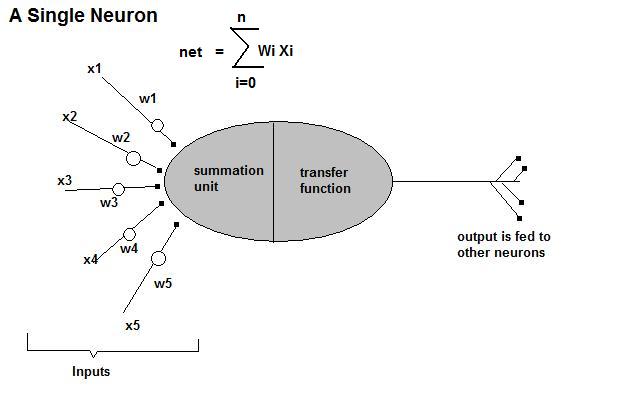
A neural network consists of several layers, and each layer has a number of neurons in it. Neurons is one layer is connected to multiple or all neurons in the next layer. Input is fed to the neurons in input layer, and output is obtained from the neurons in the last layer.



**Fig: A Fully Connected 4-4-2 neural network with 4 neurons in input layer, 4 neurons in hidden layer and 2 neurons in output layer.**

***ARTIFICIAL NEURONS :***

Now, let us have a look at the model of an artificial neuron.



An artificial neuron consists of various inputs, much like the biological neuron. Instead of Soma and Axon, we have a summation unit and a transfer function unit. The output of one neuron can be given as input to multiple neurons.

**TRAINING OF A NEURAL NETWORK *:***

**A artificial neural network can *learn* from a set of samples.**

One of the key elements of a neural network is its ability to *learn*. A neural network is not just a complex system, but a complex ***adaptive*** system, meaning it can change its internal structure based on the information flowing through it. Typically, this is achieved through the adjusting of *weights*. In the diagram above, each line represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a ***weight***, a number that controls the signal between the two neurons. If the network generates a “good” output (which we’ll define later), there is no need to adjust the weights. However, if the network generates a “poor” output—an error, so to speak—then the system adapts, altering the weights in order to improve subsequent results. This ability of a neural network to learn, to make adjustments to its structure over time, is what makes it so useful in the field of artificial intelligence.

**HOW A NEURAL NETWORK ACTUALLY WORKS :**

Working with a neural network includes

**Training the network - by providing inputs and corresponding outputs.**

* + In this phase, we train a neural network with samples to perform a particular task.

**Running the network - by providing the input to obtain the output.**

* + In this phase, we will provide an input to the network, and obtain the output. The output may not be accurate always. Generally speaking, the accuracy of the output during running phase depends a lot on the samples we provided during the training phase, and the number of times we trained the network.

**LIMITATIONS OF BACK PROPAGATION TECHNIQUE:**

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.

**LITERATURE REVIEW:**

The most widely used method of training for feed-forward ANNs is back-propagation (BP) algorithm. Feed-forward ANNs are commonly used for function approximation and pattern classifications. Back-propagation algorithm and its variations such as QuickProp and RProp 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.

***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.

***PARTICLE SWARM OPTIMIZATION (PSO):***

Particle swarm optimization (PSO) 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) 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.

***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, 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.

***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

****

**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.

****

**EVOLUTIONARY OPTIMIZATION ALGORITHMS:**

**PARTICLE SWARM OPTIMIZATION:**

**1. Introduction**  
  
Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy  in 1995, inspired by social behaviour of bird flocking or fish schooling.  
  
PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The detailed information will be given in following sections.   
  
Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.   
  
  
  
**2. Background: Artificial life**  
  
The term "Artificial Life" (A Life) is used to describe research into human-made systems that possess some of the essential properties of life. A Life includes two-folded research topic:   
  
1. A Life studies how computational techniques can help when studying biological phenomena  
2. A Life studies how biological techniques can help out with computational problems  
  
The focus of this report is on the second topic. Actually, there are already lots of computational techniques inspired by biological systems. For example, artificial neural network is a simplified model of human brain; genetic algorithm is inspired by the human evolution.   
  
Here we discuss another type of biological system - social system, more specifically, the collective behaviours of simple individuals interacting with their environment and each other. Someone called it as swarm intelligence. All of the simulations utilized local processes, such as those modelled by cellular automata, and might underlie the unpredictable group dynamics of social behaviour.   
  
  
  
There are two popular swarm inspired methods in computational intelligence areas: Ant colony optimization (ACO) and particle swarm optimization (PSO). ACO was inspired by the behaviours of ants and has many successful applications in discrete optimization problems.   
  
The particle swarm concept originated as a simulation of simplified social system. The original intent was to graphically simulate the choreography of bird of a bird block or fish school. However, it was found that particle swarm model can be used as an optimizer.   
  
**3. THE CONCEPT:**

As stated before, PSO simulates the behaviours of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food.   
  
PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. 

**4. THE ALGORITHM:**

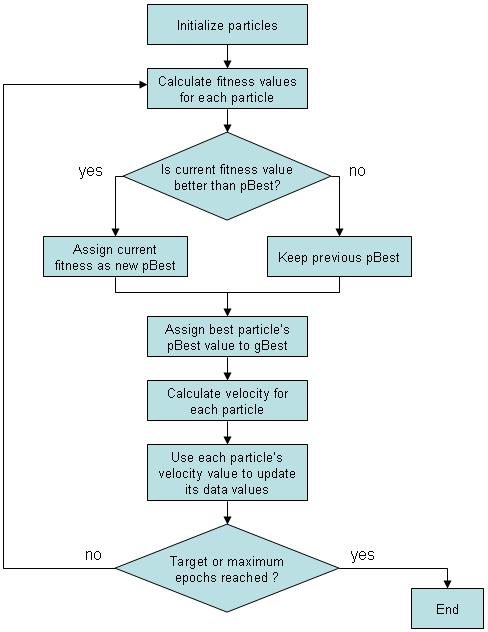
PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called lbest.  
  
After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[] + c1 \* rand() \* (pbest[] - present[]) + c2 \* rand() \* (gbest[] - present[]) (a)  
present[] =present[] + v[] (b)  
  
v[] is the particle velocity,

Present[] is the current particle (solution).

pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. (usually c1 = c2 = 2)

# The Algorithm

Flow diagram illustrating the particle swarm optimization algorithm.  
  


The pseudo code of the procedure is as follows  
  
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 the 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  
  
Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.  
   
**5. Artificial neural network and PSO**  
  
An artificial neural network (ANN) is an analysis paradigm that is a simple model of the brain and the back-propagation algorithm is the one of the most popular method to train the artificial neural network. Recently there have been significant research efforts to apply evolutionary computation (EC) techniques for the purposes of evolving one or more aspects of artificial neural networks.   
  
Evolutionary computation methodologies have been applied to three main attributes of neural networks: network connection weights, network architecture (network topology, transfer function), and network learning algorithms.   
  
Most of the work involving the evolution of ANN has focused on the network weights and topological structure. Usually the weights and/or topological structure are encoded as a chromosome in GA. The selection of fitness function depends on the research goals. For a classification problem, the rate of mis-classified patterns can be viewed as the fitness value.  
  
The advantage of the EC is that EC can be used in cases with non-differentiable PE transfer functions and no gradient information available. The disadvantages are

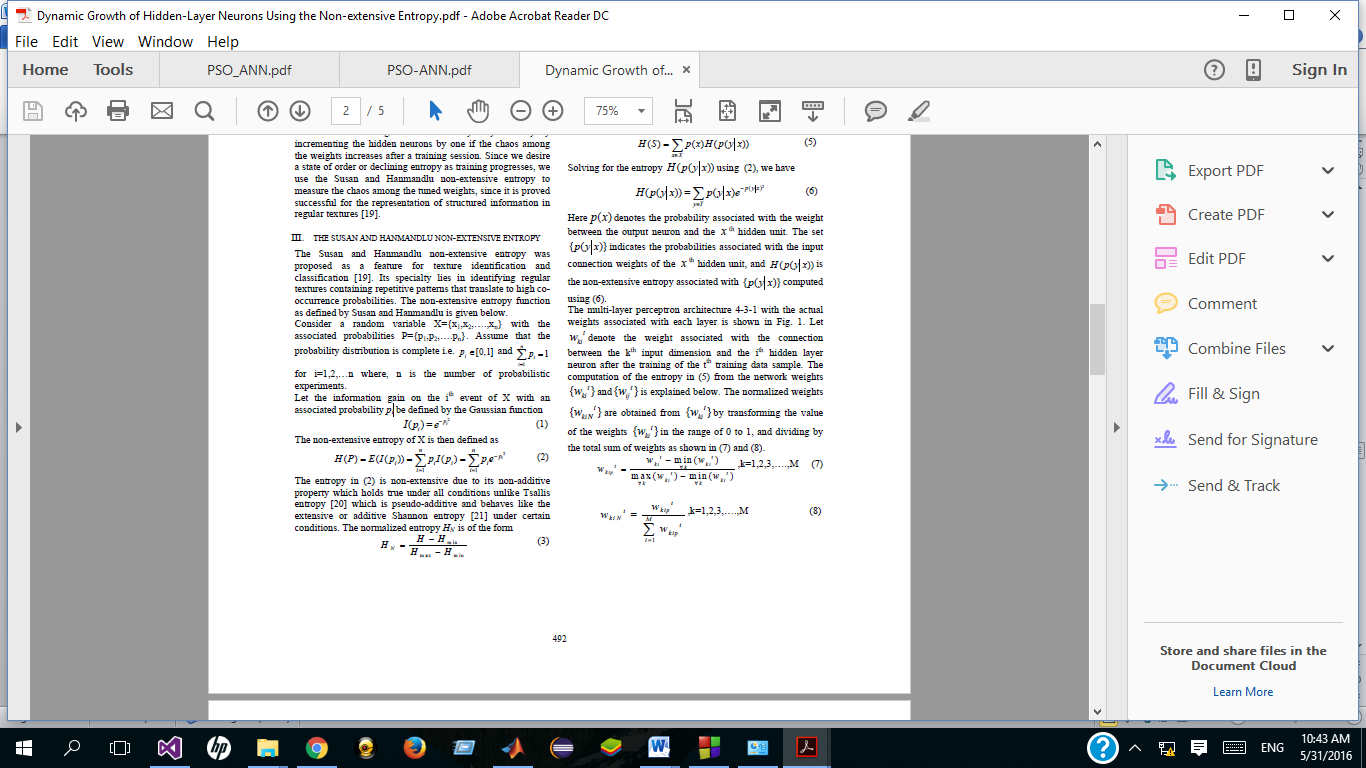
1. The performance is not competitive in some problems.

2. Representation of the weights is difficult and the genetic operators have to be carefully selected or developed.   
  
There are several papers reported using PSO to replace the back-propagation learning algorithm in ANN in the past several years. It showed PSO is a promising method to train ANN. It is faster and gets better results in most cases. It also avoids some of the problems GA met.  
  
Here we show a simple example of evolving ANN with PSO. The problem is a benchmark function of classification problem: iris data set. Measurements of four attributes of iris flowers are provided in each data set record: sepal length, sepal width, petal length, and petal width. Fifty sets of measurements are present for each of three varieties of iris flowers, for a total of 150 records, or patterns.  
  
A 3-layer ANN is used to do the classification. There are 4 inputs and 3 outputs. So the input layer has 4 neurons and the output layer has 3 neurons. One can evolve the number of hidden neurons. However, for demonstration only, here we suppose the hidden layer has 6 neurons. We can evolve other parameters in the feed-forward network. Here we only evolve the network weights. So the particle will be a group of weights, there are 4\*6+6\*3 = 42 weights, so the particle consists of 42 real numbers. The range of weights can be set to [-100, 100] (this is just an example, in real cases, one might try different ranges). After encoding the particles, we need to determine the fitness function. For the classification problem, we feed all the patterns to the network whose weights is determined by the particle, get the outputs and compare it the standard outputs. Then we record the number of misclassified patterns as the fitness value of that particle. Now we can apply PSO to train the ANN to get lower number of misclassified patterns as possible. There are not many parameters in PSO need to be adjusted. We only need to adjust the number of hidden layers and the range of the weights to get better results in different trials.

**PROPOSED EARLY TERMINATION:**

**THE SUSAN AND HANMANDLU NON-EXTENSIVE ENTROPY:**

The Susan and Hanmandlu non-extensive entropy was proposed as a feature for texture identification and classification . Its specialty lies in identifying regular textures containing repetitive patterns that translate to high co-occurrence probabilities. The non-extensive entropy function as defined by Susan and Hanmandlu is given below. Consider a random variable X={x1,x2,….,xn} with the associated probabilities P={p1,p2,….pn}. Assume that the probability distribution is complete i.e. p € [0,1] and ∑p=1.



**PROPOSED EARLY MINIMISATION**

**FLOW CHART**

LEARNING

NO

EPOCH

< MAX

**NO**

YES

NO

ENTROPY STABILIZED

NO

YES

YES

ACCEPTABLEACCURACY

**EXPERIMENTAL RESULTS:**

**IRIS DATASET:**

**Description of dataset:**

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.   
  
Predicted attribute: class of iris plant.   
  
This is an exceedingly simple domain.   
.

**Attribute Information:**

1. sepal length in cm   
2. sepal width in cm   
3. petal length in cm   
4. petal width in cm   
5. class:   
-- Iris Setosa   
-- Iris Versicolour   
-- Iris Virginica

**DIVISION OF IRIS DATASET:**

Number of training data : 75

Number of test data : 75

**ANN-PSO NETWORK PARAMETERS USED:**

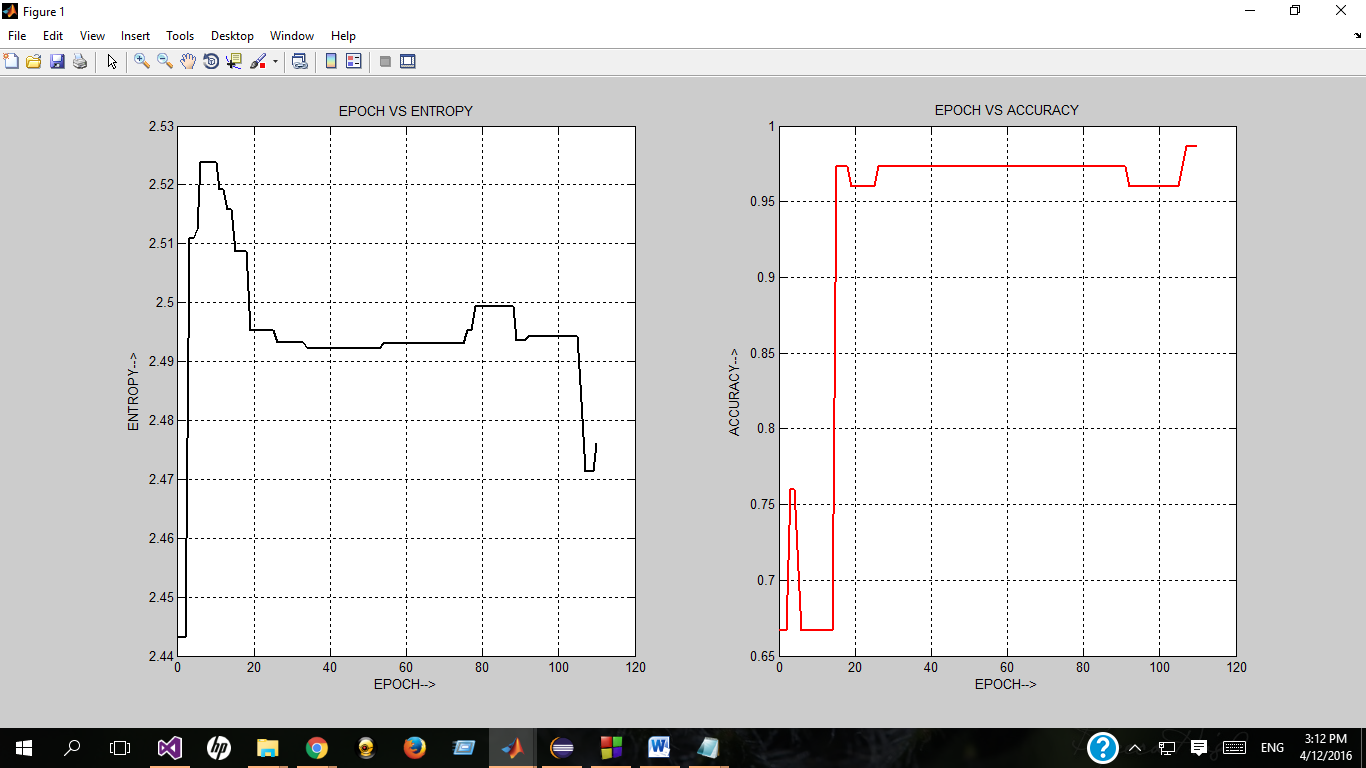
1. No of hidden layers used (numHidden} = 6

2. No. of particles used in PSO-ANN network (numParticles) = 10

3. Total no. of loops (epoch) = 110

**ACCURACY OBTAINED :**

The accuracy obtained on applying ANN-PSO algorithm on Seeds Dataset is 98.6% .



IRIS DATESET GRPPHS USING ANN-PSO

‘

**IONOSPHERE DATASET:**

**Data description:**

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.   
  
Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

**Attribute Information:**

-- All 34 are continuous   
-- The 35th attribute is either "good" or "bad" according to the definition summarized above. This is a binary classification task.

**ANN-PSO ALGORITHM ON IONOSPHERE DATASET:**

**DIVISION OF IONOSPHERE DATASET:**

Number of training data : 45

Number of test data : 39

**ANN-PSO NETWORK PARAMETERS USED:**

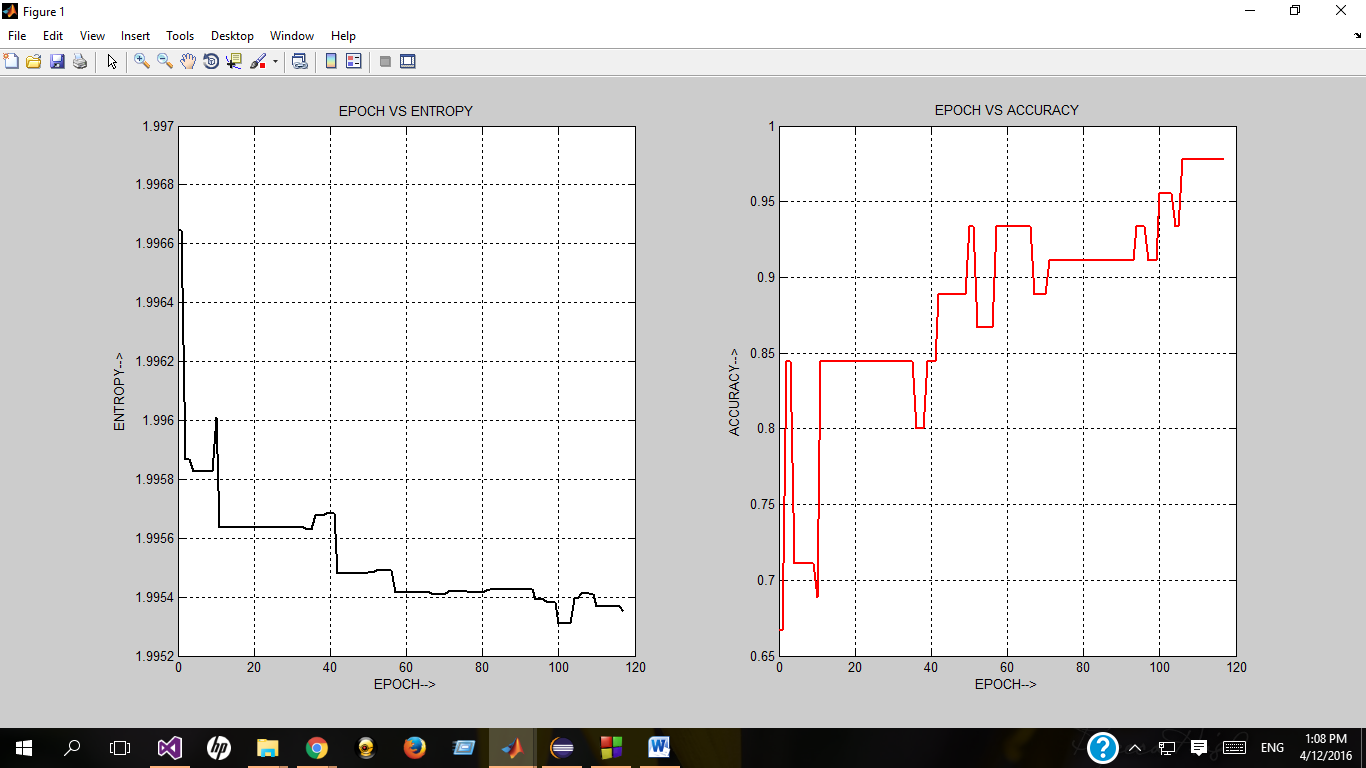
1. No of hidden layers used (numHidden} = 6

2. No. of particles used in PSO-ANN network (numParticles) = 12

3. Total no. of loops (epoch) = 117

**ACCURACY OBTAINED :**

The accuracy obtained on applying ANN-PSO algorithm on Seeds Dataset is 98.6%



IONOSPHERE DATASET GRAPHS USING ANN-PSO

**ANN-BP ALGORITHM ON IONOSPHERE DATASET:**

**DIVISION OF IONOSPHERE DATASET:**

Number of training data : 45

Number of test data : 39

**PARAMETERS USED:**

Number of Input neurons ( NUMIN ) = 34

Number of Hidden neurons ( NUMHID) = 6

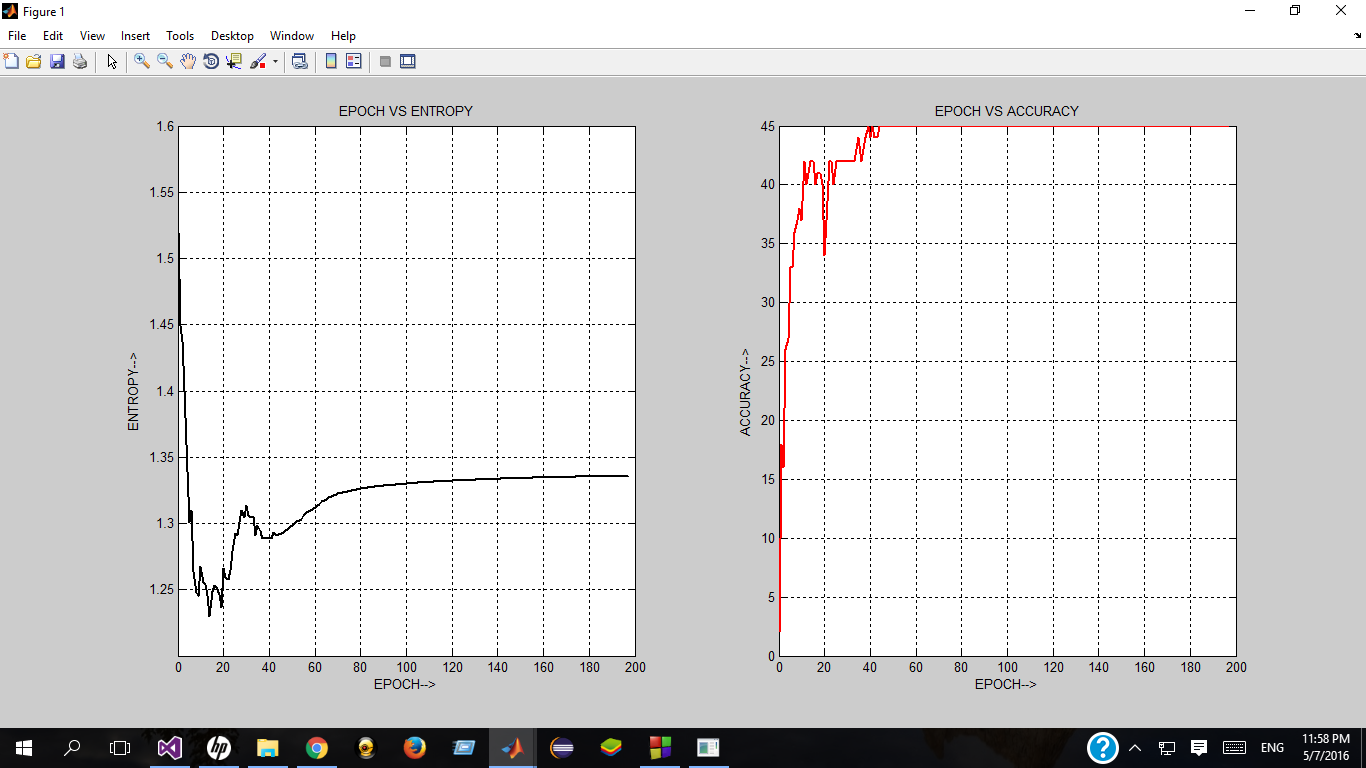
Number of Output neurons (NUMOUT ) = 2

Learning rate (eta) = 0.01

Momentum term (alpha) = 0.05

**ACCURACY OBTAINED :**

Accuracy on Test Data = 34 out of 45 or 75.6 %



IONOSPHERE DATASET GRAPHS USING ANN-BP

**OBSERVATION :**

As evident from the entropy and accuracy graph, the accuracy and entropy are inversely proportional to each other in an ANN-PSO network.

**SEEDS DATASET**

**DESCRIPTION:**

The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. High quality visualization of the internal kernel structure was detected using a soft X-ray technique. It is non-destructive and considerably cheaper than other more sophisticated imaging techniques like scanning microscopy or laser technology. The images were recorded on 13x18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agro physics of the Polish Academy of Sciences in Lublin.

The data set can be used for the tasks of classification and cluster analysis.

Attribute Information:

To construct the data, seven geometric parameters of wheat kernels were measured:

1. area A,

2. perimeter P,

3. compactness C = 4\*pi\*A/P^2,

4. length of kernel,

5. width of kernel,

6. asymmetry coefficient

7. length of kernel groove.

All of these parameters were real-valued continuous.

Total number of instances = 210

Total number of classes = 3

**ANN-PSO ALGORITHM ON SEEDS DATASET**

**DIVISION OF SEEDS DATASET:**

Number of training data : 120

Number of test data : 90

**ANN-PSO NETWORK PARAMETERS USED:**

1. No of hidden layers used (numHidden} = 6

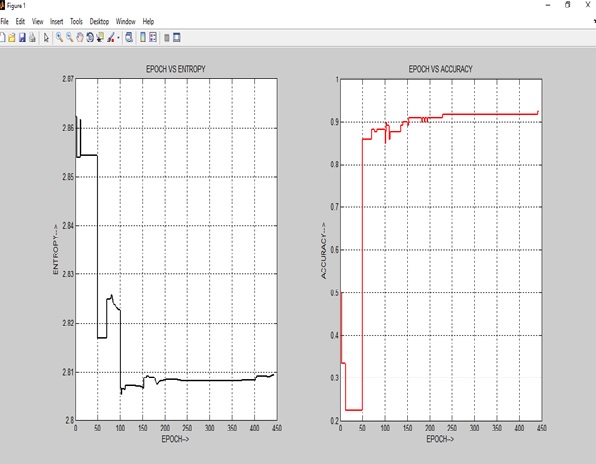
2. No. of particles used in PSO-ANN network (numParticles) = 10

3. Total no. of loops (epoch) = 444

**ACCURACY OBTAINED :**

The accuracy obtained on applying ANN-PSO algorithm on Seeds Dataset is 92.5 % .

**ENTROPY AND ACCURACY GRAPH :**

****

**ANN-BP ALGORITHM ON SEEDS DATASET**

**DIVISION OF SEEDS DATASET:**

Number of training data : 120

Number of test data : 90

**PARAMETERS USED:**

Number of Input neurons ( NUMIN ) = 7

Number of Hidden neurons ( NUMHID) = 6

Number of Output neurons (NUMOUT ) = 3

Learning rate (eta) = 0.01

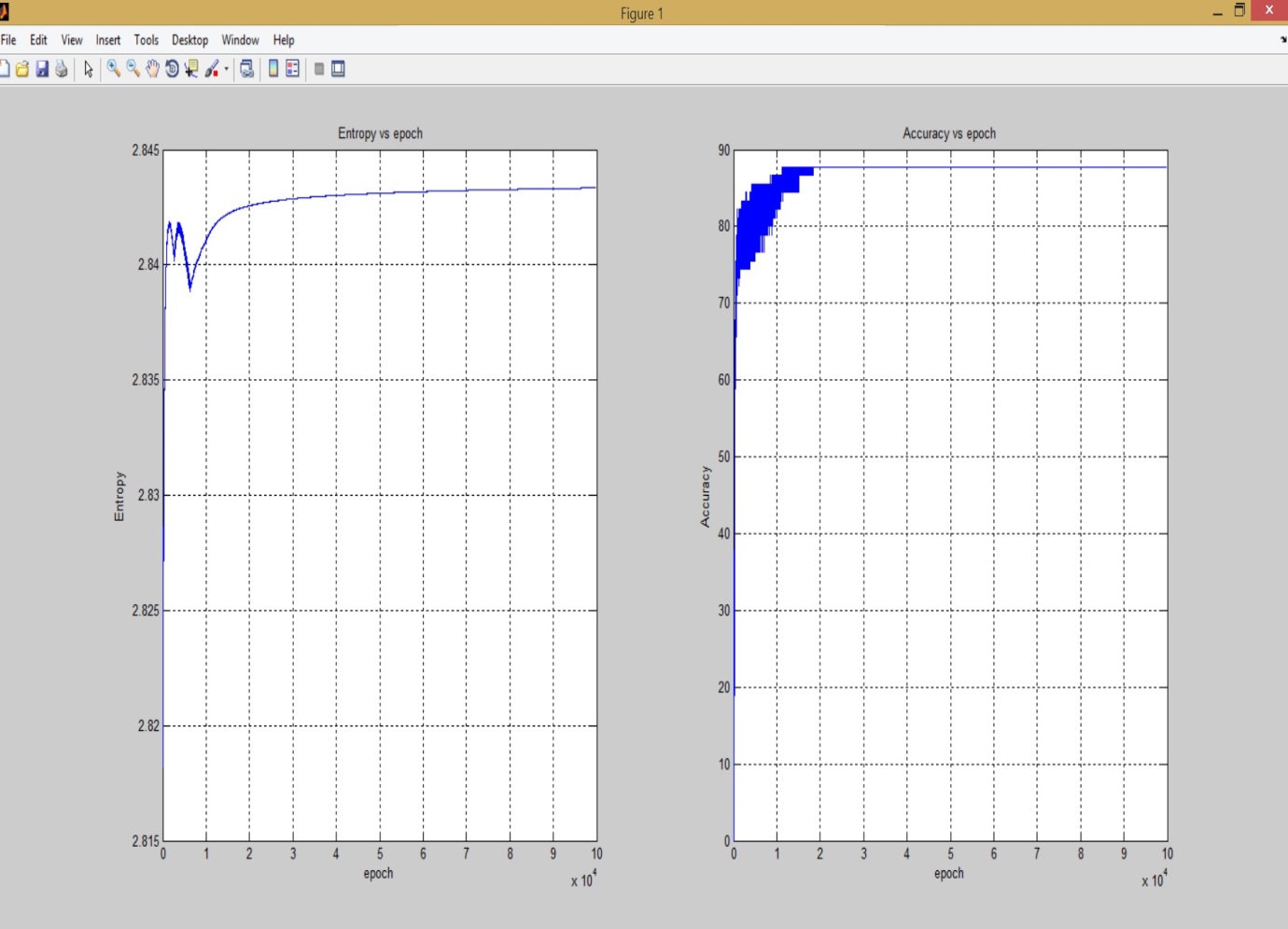
Momentum term (alpha) = 0.05

**ACCURACY OBTAINED :**

Accuracy on Training Data = 120 out of 120 or 100 %

Accuracy on Test Data = 79 out of 90 or 87.77 %

**ENTROPY AND ACCURACY GRAPH :**



**WINE DATASET**

**DESCRIPTION:**

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivators.

The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Class Distribution: number of instances per class

class 1 59

class 2 71

class 3 48

**Attribute Information:**

The attributes are

1) Alcohol

2) Malic acid

3) Ash

4) Alcalinity of ash

5) Magnesium

6) Total phenols

7) Flavanoids

8) Nonflavanoid phenols

9) Proanthocyanins

10)Color intensity

11)Hue

12)OD280/OD315 of diluted wines

13)Proline

**ANN-PSO ALGORITHM ON WINE DATASET**

**DIVISION OF WINE DATASET:**

Number of training data : 142

Number of test data : 36

**ANN-PSO NETWORK PARAMETERS USED:**

1.Number of Input neurons ( NUMIN ) = 13

2.Number of Hidden neurons ( NUMHID) = 30

3.Number of Output neurons (NUMOUT ) = 3

4. No. of particles used in ANNPSO network (numParticles) = 36

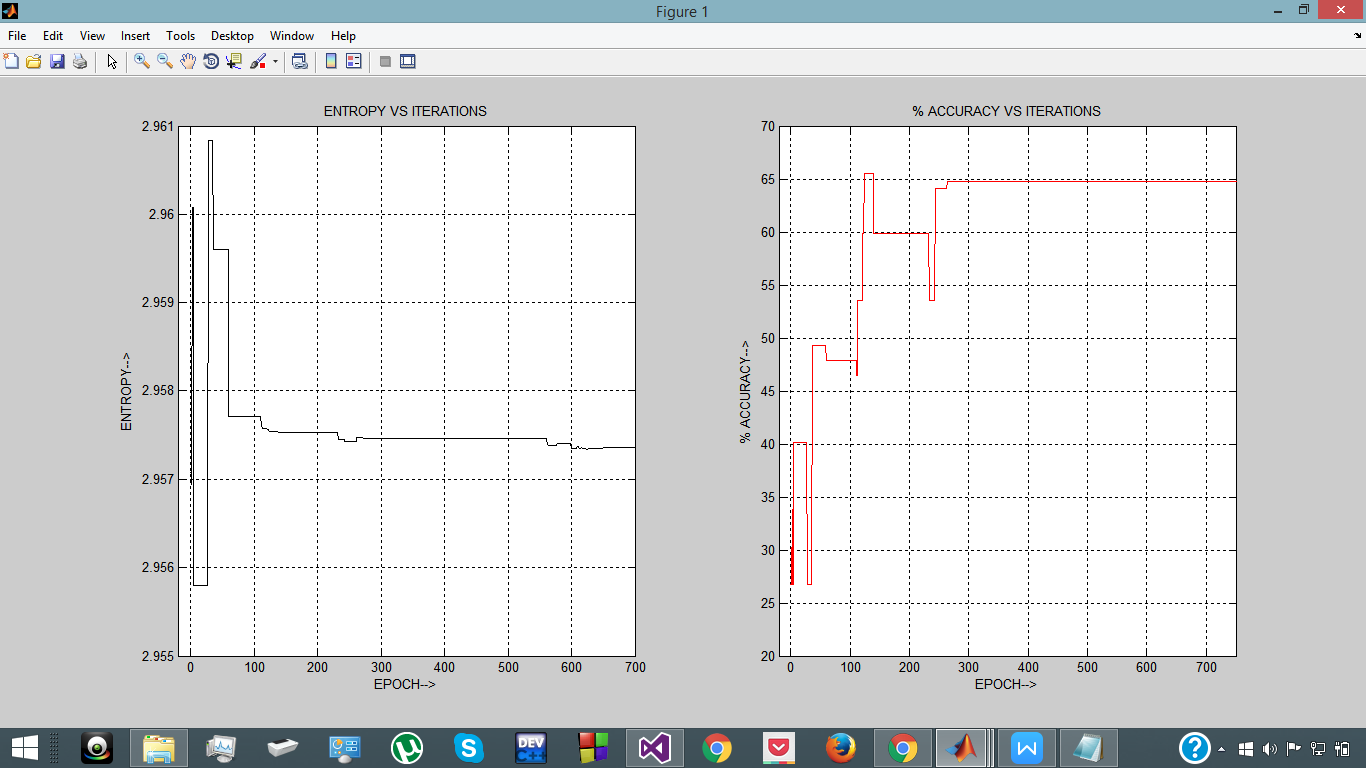
5. Total no. of loops (epoch) = 700

**ACCURACY OBTAINED :**

Accuracy on training data = 64.79%

Accuracy on test data = 66.67%

**ENTROPY AND ACCURACY GRAPH :**



**OBSERVATION :**

As evident from the entropy and accuracy graph, the accuracy and entropy are inversely proportional to each other in an ANN-PSO network.

**ANN-BP ALGORITHM ON WINE DATASET**

**DIVISION OF WINE DATASET:**

Number of training data : 142

Number of test data : 36

**PARAMETERS USED:**

Number of Input neurons ( NUMIN ) = 13

Number of Hidden neurons ( NUMHID) = 10

Number of Output neurons (NUMOUT ) = 3

Learning rate (eta) = 0.0002

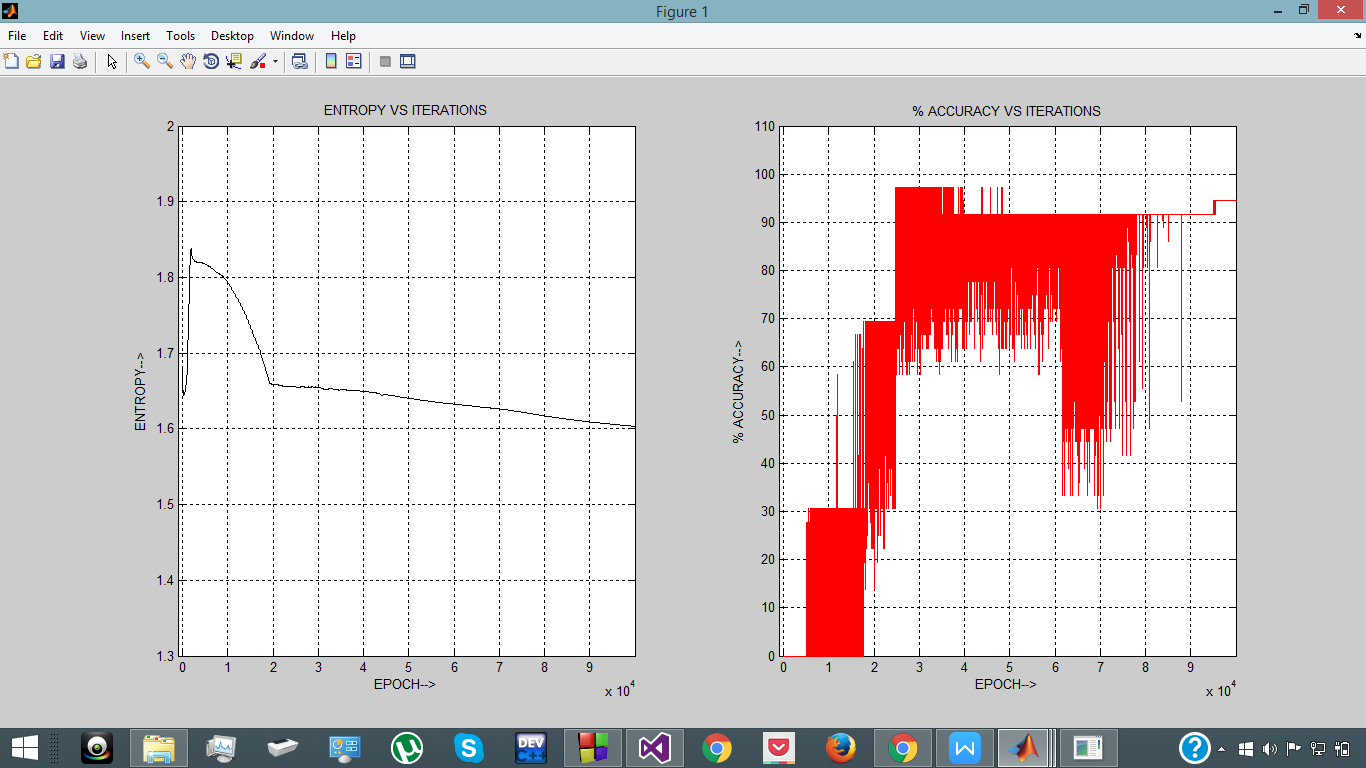
Momentum term (alpha) = 0.005

**ACCURACY OBTAINED :**

Accuracy on Training Data = 142 out of 142 or 100 %

Accuracy on Test Data = 34 out of 36 or 94.44 %

**ENTROPY AND ACCURACY GRAPH :**



**OBSERVATION:**

In case of an ANN-BP network for the wine dataset , entropy decreases as the number of iterations increase . While , on the other hand , the accuracy increases as the number of iterations increase . There is ambiguity in the results for iterations 60k to 80k , but overall we can say that there is an inverse relation between the two.

Early termination point can be chosen to be the one, where the error has been reduced significantly (so as to achieve acceptable accuracy) and entropy starts stabilizing too (for iterations 3000-4000).

**BREAST CANCER DATASET**

**DESCRIPTION:**

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.   
  
Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) , a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

Patients outcome prediction for breast cancer. The training data contains 78 patient samples, 34 of which are from patients who had developed distance metastases within 5 years (labelled as "relapse"), the rest 44 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labelled as "non-relapse"). Correspondingly, there are 12 relapse and 7 non-relapse samples in the testing data set. The number of genes is 24481. We replaced "NaN" symbol in original ratio data with 100.0.

Attribute Information:

The output is classified into two classes, viz:

1) ID number   
2) Diagnosis (M = malignant, B = benign)   
3-32)   
  
Ten real-valued features are computed for each cell nucleus:   
  
a) radius (mean of distances from center to points on the perimeter)   
b) texture (standard deviation of gray-scale values)   
c) perimeter   
d) area   
e) smoothness (local variation in radius lengths)   
f) compactness (perimeter^2 / area - 1.0)   
g) concavity (severity of concave portions of the contour)   
h) concave points (number of concave portions of the contour)   
i) symmetry   
j) fractal dimension ("coastline approximation" - 1)

All of these parameters were real-valued continuous.

Total number of instances = 569

Total number of classes = 2

**ANN-PSO ALGORITHM ON SEEDS DATASET**

**DIVISION OF SEEDS DATASET:**

Number of training data : 68

Number of test data :68

**ANN-PSO NETWORK PARAMETERS USED:**

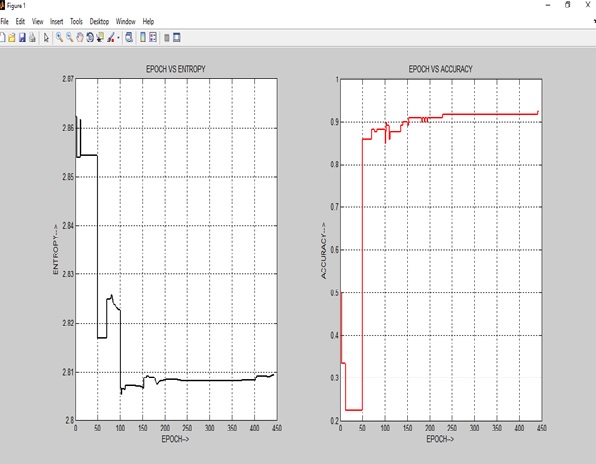
1. No of hidden layers used (numHidden} = 27

2. No. of particles used in PSO-ANN network (numParticles) = 27

**ACCURACY OBTAINED :**

The accuracy obtained on applying ANN-PSO algorithm on Seeds Dataset is 76.81%.

**ENTROPY AND ACCURACY GRAPH :**

****

**OBSERVATION:**

As evident from the entropy and accuracy graph, the accuracy and entropy are inversely proportional to each other in an ANN-PSO network.

**ANN-BP ALGORITHM ON SEEDS DATASET**

**DIVISION OF SEEDS DATASET:**

Number of training data : 68

Number of test data : 68

**PARAMETERS USED:**

Number of Input neurons ( NUMIN ) = 10

Number of Hidden neurons ( NUMHID) = 27

Number of Output neurons (NUMOUT ) = 2

Learning rate (eta) = 0.01

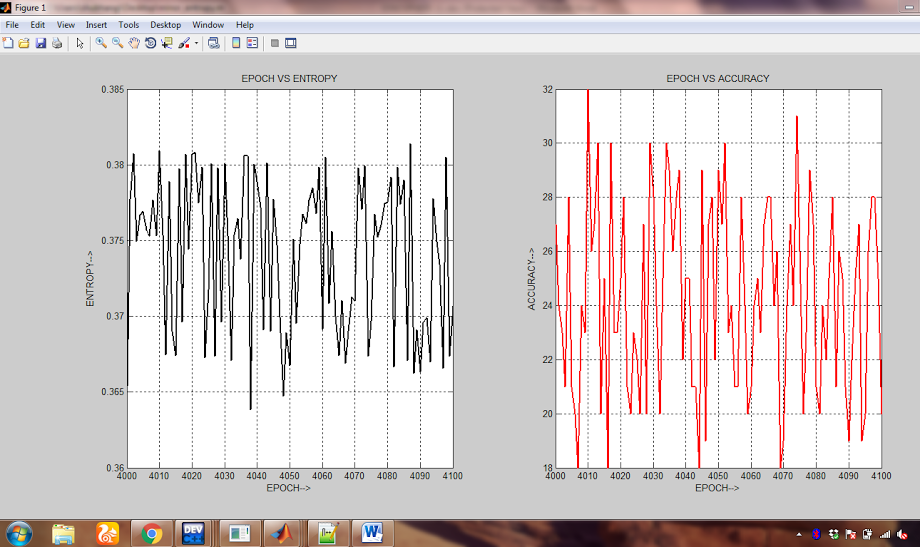
Momentum term (alpha) = 0.05

**ACCURACY OBTAINED :**

Accuracy on Training Data = 53.43 %

Accuracy on Test Data = 40 out of 68

**ENTROPY AND ACCURACY GRAPH :**



**OBSERVATION:**

In case of an ANN-BP network, entropy avs accuracy relation is not established. as accuracy is very i.e ANN is not able to perfectly classify this data.

**RESULTS FOR ANN-PSO USING MEAN CROSS ENTROPY:**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DATASET** | **ACCURACY**  **(%)** | **MAX EPOCH** | **NUM HIDDEN** | **NUM INPUT** | **NUM OUTPUT** |
| **IRIS** | 98.6 | 110 | 6 | 4 | 3 |
| **IONOSPHERE** | 97.8 | 117 | 6 | 34 | 2 |
| **SEEDS** | 92.5 | 444 | 6 | 7 | 3 |
| **BREAST CANCER** | 98 | 4500 | 10 | 10 | 2 |
| **WINE** | 64.7 | 1000 | 30 | 13 | 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DATASET** | **ACCURACY**  **(%)** | **MAX EPOCH** | **NUM HIDDEN** | **NUM INPUT** | **NUM OUTPUT** |
| **IRIS** | 78 | 100000 | 6 | 4 | 3 |
| **IONOSPHERE** | 75.6 | 100000 | 6 | 34 | 2 |
| **SEEDS** | 87.7 | 100000 | 6 | 7 | 3 |
| **BREAST CANCER** | 53.6 | 100000 | 10 | 10 | 2 |
| **WINE** | 66.7 | 100000 | 30 | 13 | 3 |

**RESULTS FOR ANN-BP USING MEAN CROSS ENTROPY:**

.

**SYSTEM CONFIGURATION:**

PROCESSOR: Intel core i7-4700MQ

CLOCK SPEED: 2.40 GHz

RAM: 8GB DDR3

OPERATING SYSTEM: WINDOWS 10 x64 BASED PROCESSOR

**CONCLUSION**

One fascinating thing about artificial neural networks is that, they are mainly inspired by the human brain. This doesn't mean that Artificial Neural Networks are *exact* simulations of the biological neural networks inside our brain - because the actual working of human brain is still a mystery. The concept of artificial neural networks emerged in its present form our very limited understanding about our own brain .The most common application of neural networks in computing today is to perform one of these “easy-for-a-human, difficult-for-a-machine” tasks, often referred to as pattern recognition. Applications range from optical character recognition (turning printed or handwritten scans into digital text) to facial recognition.

It is clear that Artificial Neural Networks are a very powerful and accurate tool for reactive power dispatch. Due to the very nature of Artificial Neural Networks, and softcomputing

in general, there is no one solution equation. This requires networks to be customized for each system. Standard methods require the user to choose a network topology,

inputs, and transfer functions for a network before training. Particle Swarm Optimization overcomes these limitations because it is blind to what it is optimizing. Any network

parameter may be thrown into the mix along with the network= weights to be optimized.

On the basis of these observations we can establish an inverse relationship between ENTROPY and ACCURACY i.e.

ENTROPY=k\*(1/ACCURACY);

Where k is some proportionality constant

But these results hold for only those datasets in which the ACCURACY is above 80% on testing data.

We can stop the neural network whenever the entropy becomes consistent and check for the accuracy if it satisfies our constraints.

**REFERENCES :**

# [1] **Dr Seba Susan , Mayank Dwivedi , Dynamic Growth of Hidden-Layer Neurons Using the Non-extensive Entropy**

|  |
| --- |
| Published in: |
|  |
| CSNT '14 Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies |

[2] Ajith Abraham, An Evolving Fuzzy Neural Network Model Based

Reactive Power Control, In Proceedings of The Second International

Conference on Computers In Industry, ICCI 2000, Majeed A Karim et al

(Eds.), Bahrain, pp. 247-253, 2000.

[3] K.H. Abdul-Rahman, S.M. Shahidehpour, M. Daneshdoost, “AI

approach to optimal VAr control with fuzzy reactive loads” in *IEEE*

*Transactions on Power Systems*, volume 10, pp 88-97, Feb. 1995.

[4] V. Ajjarapu, Z. Albanna, “Application of genetic based algorithms to

optimal capacitor placement” in *Neural Networks to Power Systems*,

pp 251-255, Jul. 1991.

[5] G. Cartina, C. Bonciu, M. Musat, Z. Zisman, “AI approach to optimal

VAr control with fuzzy reactive loads” in *Electrotechnical*

*Conference*, volume 2, pp 1103-1106, May. 1998.

[6] T.W.S. Chow, Y.F. Yam, “Measurement and evaluation of instantaneous

reactive power using neural networks” in *IEEE Transactions on*

*Power Delivery*, volume 9, pp 1253-1256, Jul. 1994.

[7] Yuan-Yih Hsu, Feng-Chang Lu, “A combined artificial neural networkfuzzy

dynamic programming approach to reactive power/voltage control

in a distribution substation” in *IEEE Transactions on Power Systems*,

volume 13, pp 1265-1271, Nov. 1998.