Pattern Recognition Based On Intelligent System

Faez H. Ali
Al-Mustansiriya University
Dept. of Maths, College of Science

Hazem N. Abdulrazzak
Al-Rafidain University College
Computer Communication Eng. Dept.

Abstract: *Neural Network (NN) is an information processing system that has been developed as generalization models of human cognition of neural biology.*

Two important learning methods for NN are discussed to be our comparative study; these methods are Back Propagation (BP) and Particle Swarm Optimization (PSO). These two methods are found to learn the NN by modifying the weights of the NN, this is done by calculating the fitness value which is considered as a threshold value.

In this work the Pattern Recognition (PR) problem will be manipulated, which is considered one of the important applications of classification filed. The Greek Text Letter Recognition (A..Ω) is the objective of this paper, the upper and lower cases are discussed.

The proposed system needs the Microsoft windows, in all new versions, as an operating system with Delphi version 7.0 as a programming language.

Keywords: Artificial Neural Network, Particle Swarm Optimization, Pattern Recognition.

1. Introduction

A Neural Network (NN) is not programmed to solve a problem- instead, it learns to solve a problem [1]. A NN may benefit from the application of solving methods to its training. One of the important new learning methods is a Particle Swarm Optimization (PSO), which is simple in concept, has few parameters to adjust and easy to implement. PSO has found applications in a lot of areas. In general, all the application areas that the other evolutionary techniques are good at are good application areas for PSO [2].

In 1995, Kennedy J. and Eberhart R. [3], introduced a concept for the optimization of nonlinear functions using particle swarm methodology. The evolution of several paradigms outlined, and an implementation of one of the paradigms had been discussed.

In 1999, Eberhart R.C. and Hu X. [4], arranged a new method for the analysis of human tremor using PSO which is used to evolve a NN that distinguishes between normal subject and those with tremor.

In 2004, Shi Y. [2], surveyed the research and development of PSO in five categories: algorithms, topology, parameters, hybrid PSO algorithms, and applications. There are certainly other research works on PSO which are not included due to the space limitation.

In this paper, first we will put the matrix design of each of the Greek language Upper/Lower case letters then construct the suitable neural network which is learned by PSO and BP as learning methods and then compare the learning results obtained from the learning process of NN finally, evaluate which is better in such letter recognition problem.

2. Artificial Neural Network

Artificial Neural Network (ANN) is an information-processing system that has certain performance characteristics in common with biological neural networks. ANNs represent an important area of research, which opens a variety of new possibilities in different fields including classification or pattern recognition or predictions. It is known by NN that can approximate functions and mathematical operators arbitrarily as well as the number of neurons in the network tends to infinity. In this respect FeedForward Artificial Neural Networks (FFANNs) can be considered as "universal approximations" which is capable of describing the input-output relationships of mechanical systems [5].

With classification in application domain of ANNs, we have a long list of researches and real applications include speed processing, image processing and computer vision, pattern classification and recognition, system control, robotics, forecasting and modeling, optimization and management of information and medical diagnosis. Also there is a great role of NNs as a means for implementing expert systems, because of their ability to solve a specific problem, producing it as if it were a black-box solution where the mode of producing answers is not clearly understood. Due to its adaptive and parallel processing ability, it has many applications in the engineering field [6].

NNs can be grouped into six areas of applications: prediction, pattern recognition, associative memories, classification, optimization and general mapping.

3. Neural Network Learning Algorithms

The learning algorithm is a procedure for modifying the weights on the connection links in a neural net, and also known as training algorithms or learning rules [7].

NNs are trained by two main types of learning algorithms: supervised learning algorithms and unsupervised learning algorithms.

Unsupervised learning algorithms do not require the unknown desired outputs. A supervised learning algorithm adjusts the strengths or weights of the inter-neuron connections according to the difference between the desired and actual network outputs corresponding with a given input. The most common examples of supervised learning algorithms include Back propagation algorithm (BP) [6].

4. Particle Swarm Optimization (PSO)

PSO was originally developed by a social-psychologist J. Kennedy and an electrical engineer R. Eberhart in 1995 and emerged from earlier experiments with algorithms that modeled the "flocking behavior" seen in many species of birds. Where birds are attracted to a roosting area in simulations they would begin by flying around with no particular destination and in spontaneously formed flocks until one of the birds flew over the roosting area [8]. PSO has been an increasingly hot topic in the area of computational intelligence. It is yet another optimization algorithm that falls under the soft computing umbrella that covers genetic and evolutionary computing algorithms as well [9].

4.1 Fitness Criterion

One of these stopping criterions is the fitness value. Since the PSO algorithm is chosen to be a supervised learning algorithm, then there are observed values of (ti) and desired output values of (fi). These two values have to be compared, if they are closed to each other then the fitness is good, else the algorithm must continue its calculations until this condition is satisfied or the specified number of iterations is finished.

The corrections to the weights are selected to minimize the residual error between ti and fi output. The Mean Squared Error (MSE) is one solution for the comparison process:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (ti - fi)^2 \qquad \dots (1)$$

Where n is the number of the compared categories.

4.2 PSO Algorithm

The PSO algorithm depends in its implementation in the following two relations:

$$vid=w*vid+c1*r1*(pid-xid)+c2*r2*(pgd-id) \quad ...(2a)$$
$$xid = xid + vid \quad …(2b)$$

where c1 and c2 are positive constants, r1 and r2 are random function in the range [0,1], xi=(xi1,xi2,…,xid) represents the ith particle; pi=(pi1,pi2,…,pid) represents the best previous position (the position giving the best fitness value) of the ith particle; the symbol g represents the index of the best particle among all the particles in the population, v=(vi1,vi2,…,vid) represents the rate of the position change (velocity) for particle i [2].
The original procedure for implementing PSO is as follows:

1. Initialize a population of particles with random positions and velocities on d-dimensions in the problem space.
2. PSO operation includes:
   a. For each particle, evaluate the desired optimization fitness function in d variables.
   b. Compare particle's fitness evaluation with its pbest. If current value is better than pbest, then set pbest equal to the current value, and pi equals to the current location xi.
   c. Identify the particle in the neighborhood with the best success so far, and assign it index to the variable g.
   d. Change the velocity and position of the particle according to equation (2a) and (2b).
3. Loop to step (2) until a criterion is met.

Like the other evolutionary algorithms, a PSO algorithm is a population based on search algorithm with random initialization, and there is an interaction among population members. Unlike the other evolutionary algorithms, in PSO, each particle flies through the solution space, and has the ability to remember its previous best position, survives from generation to another. The flow chart of PSO algorithm is shown in figure (1) [10].

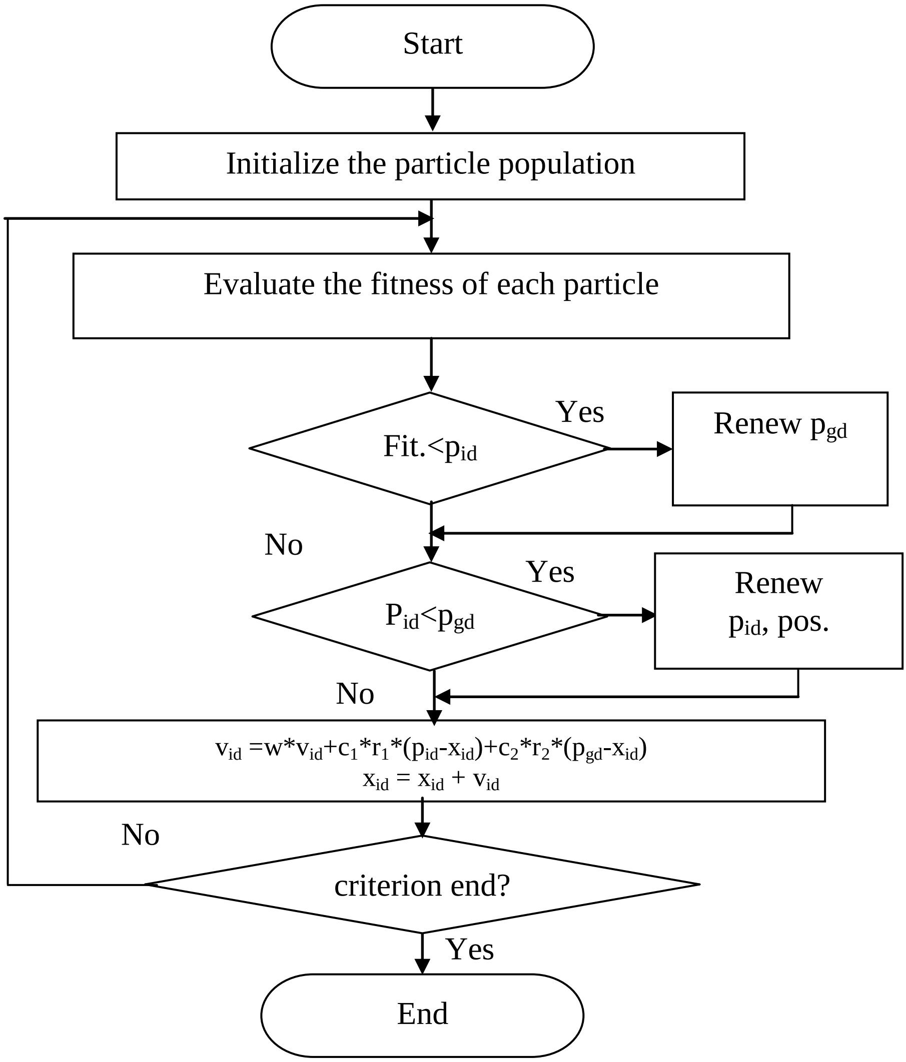

Figure (1) Flowchart of PSO Algorithm [6].

4.3 The Parameters of PSO [11],[12]

A number of factors will affect the performance of the PSO. These factors are called PSO parameters, these parameters are:

1. Number of particles in the swarm affects the run-time significantly, thus a balance between variety (more particles) and speed (less particles) must be sought.
2. Maximum velocity (vmax) parameter. This parameter limits the maximum jump that a particle can make in one step.
3. The role of the inertia weight w, in equation (2a), is considered critical for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current one.
4. The parameters c1 and c2, in equation (2a), are not critical for PSO's convergence. However, proper fine-tuning may result in faster convergence and alleviation of local minima, c1 than a social parameter c2 but with c1 + c2 = 4.
5. The parameters r1 and r2 are used to maintain the diversity of the population, and they are uniformly distributed in the range [0,1].

4.4 Training the ANN by using PSO

The PSO algorithm is vastly different than any of the traditional methods of training. PSO does not just train one network, but rather training networks. PSO builds a set number of ANN and initializes all network weights to random values and starts training each one. On each pass through a data set, PSO compares each network's fitness. The network with the highest fitness is considered the global best [13].

Each neuron contains a position and velocity. The position corresponds to the weight of a neuron. The velocity is used to update the weight. If a neuron is further away then it will adjust its weight more than a neuron that is closer to the global best [9].

As usual the number of interconnections (IC) for NN can be calculated by the following relations for one hidden layer:

$$IC = n * m + m*k \qquad \dots (3)$$

Where n is the number of nodes in the input layer, m is the number of nodes in the hidden layer, while k is the number of nodes in the output layer.

5. NN-Learning System

PSO is an extremely simple concept, and can be implemented without complex data structure. No complex or costly mathematical functions are used, and it doesn't require a great amount of memory [1]. The facts of PSO has fast convergence, only a small number of control parameters, very simple computations, good performance on neural networks, and the lack of derivative computations made it an attractive option for training the NN.

In this paper, the binary PSO was implemented in classification applications. From the classification application, the letter recognition problem is chosen.

A comparative study is made on the computational requirements of the PSO and BP as a training algorithm for NN's. A NN-Learning system has been proposed to solve the mentioned problems by Binary PSO (BPSO) and BP algorithms.

5.1 BPSO Algorithm

The original procedure of binary PSO algorithm is as follows:

1. Read NN information file.
2. Change (Integer or real) input data to binary data.
3. Initialize a population of particles with random positions and velocities on d-dimensions in the problem space.
4. PSO operations.
5. Loop to step (4) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

5.2 NN-Learning System Implementation

The proposed system called NN-learning system since it can be applied on BP algorithm as well as on PSO algorithm which applied on Greek letter recognition problem. The block diagram of the implementation of NN-learning system shown in figure (2).

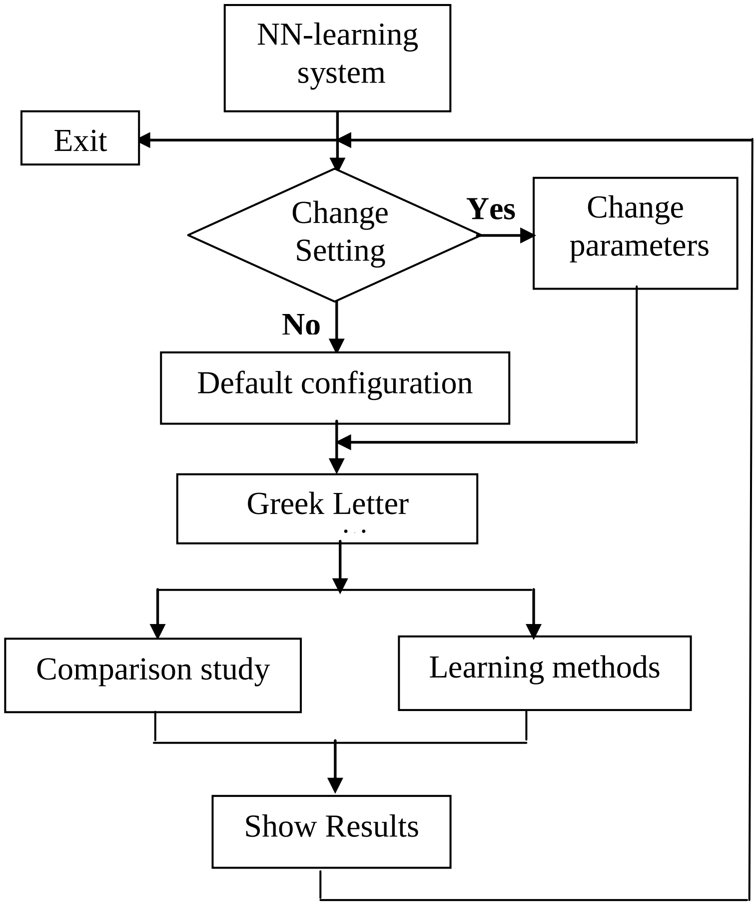

Figure (2)Block diagram of NN-Learning system

5.3 Pattern Recognition Applications

One of the most important applications of NN is the classification problems which are divided into many fields; our interest of classification problem is the Pattern Recognition (PR) problems.

Pattern Recognition is the research area that studies the operation and design of systems that recognize patterns in data.

PR aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. Important application areas are image analysis, speech analysis, man and machine diagnostics, person identification and industrial inspection [14].

Binary PSO algorithm has been used to recognize patterns of character, number or text letter. The network is initially learned upon symbols and tested on another set of symbols to illustrate its efficiency in recognizing new pattern.

5.4 LR-Neural Network Construction

In this paper, the patterns are represented by d*e matrix. The data of the matrix consists of 1's and 0's only. The objects of interest are the 1's and the background consists of 0's. The output symbols consist of k bits which must cover all the possible patterns that mean 2k possible, these 2k is the targets. The d*e binary values are stored in a file in a one dimensional array row by row arrangement. In this manner since the number of output nodes k>1, then the MSE value, must be rewritten as in relation (4).

$$MSE = \frac{1}{Ns * k} \sum_{i=1}^{Ns} \sum_{j=1}^{k} (tij\text{-}fij)^2 \qquad \ldots (4)$$

Where Ns is the number of training set.

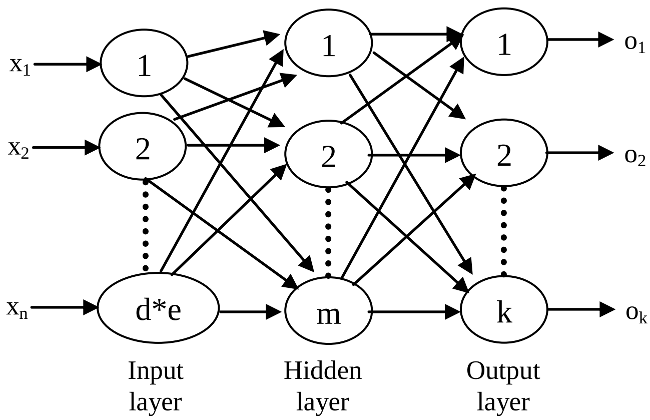

Figure (3) the proposed PR-NN (d*e-m-k).

The proposed LR neural network consists of 3 layers, with (d-m-k) neurons which represent the number of neuron in input, hidden and output layers respectively. The proposed LR-NN is shown in figure (3).

6. Greek Letter Representation in PR-NN

The aim was to recognize 24 upper–case (lower–case) alphabets in Greek language i.e. From A to Ω (α to ω). Every text letter pattern is represented by 9*9 matrix (d=9,e=9), this means that 81 neurons are in the input layer. For example if the text letter is "Σ=sigma" which is the 17th upper case text letter of the Greek alphabet, then the output data is "17=10001" and if the text letter is "δ=delta" which is the 3rd lower text letter of the Greek alphabet, then the output data is "3=00011". Since the output layer consists of 5 neurons and the training set Ns=24 for Greek alphabet, each letter was represented by one matrix. The matrix data for the letters "Σ" and "δ" are shown in figure (4).

	Σ=sigma										δ=delta								
1	1	1	1	1	1	1	1	1	0		0	0	0	0	1	1	0	0	0
2	1	0	0	0	0	0	0	1	0		0	0	0	1	0	0	1	0	0
3	0	1	0	0	0	0	0	0	0		0	0	0	1	1	0	0	0	0
4	0	0	1	0	0	0	0	0	0		0	0	0	0	1	1	0	0	0
5	0	0	0	1	0	0	0	0	0		0	0	0	1	1	0	1	0	0
6	0	0	1	0	0	0	0	0	0		0	0	1	0	0	0	0	1	0
7	0	1	0	0	0	0	0	0	0		0	0	1	0	0	0	0	1	0
8	1	0	0	0	0	0	0	1	0		0	0	0	1	0	0	1	0	0
9	1	1	1	1	1	1	1	1	0		0	0	0	0	1	1	0	0	0
	1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9

Figure (4) Input (9*9) matrix data of the text letters "Σ" and "δ".

The binary coding of Greek text letters in upper and lower cases has been shown in table (1).

Table (1) output binary coding of Greek text letters.

No.	Text Letter UC	Text Letter LC	Integer value	Binary Coding	No.	Text Letter UC	Text Letter LC	Integer value	Out put
1	Α	α	0	00000	13	Ν	ν	12	01100
2	Β	β	1	00001	14	Ξ	ξ	13	01101
3	Γ	γ	2	00010	15	Ο	ο	14	01110
4	Δ	δ	3	00011	16	Π	π	15	01111
5	Ε	ε	4	00111	17	Ρ	ρ	16	10000
6	Ζ	ζ	5	01111	18	Σ	σ	17	10001
7	Η	η	6	00110	19	Τ	τ	18	10010
8	Θ	θ	7	00111	20	Υ	υ	19	10011
9	Ι	ι	8	01000	21	Φ	φ	20	10100
10	Κ	κ	9	01001	22	Χ	χ	21	10101
11	Λ	λ	10	01010	23	Ψ	ψ	22	10110
12	Μ	μ	11	01011	24	Ω	ω	23	10111

10 neurons can be suggested to be used in the hidden layer of the proposed Greek Letter Recognition-NN (GLR-NN).

7. Results of PR-NN Trained by BP and PSO

We need to use the following symbols in the tables of BP and PSO implementation:

- **Exp** : Experiment number.
- **NoI** : Number of Iterations.
- **MSE**: least fitness (Mean Square Error).
- **AL** : Accuracy Level of the learning.
- **T** : computation Time.

7.1 GLR-NN Trained by PB

In this subsection we will train Letter recognition NN (GLR-NN) with layers (81-10-5) by using BP. The results of GLR-NN trained by BP for 5 experiments for each sample of the NN mentioned in section (7) are shown tables (1) and (2).

Table (2) Results of Gr_LC-NN trained by BP.

Exp.	NoI	MSE	AL %	Time/sec.
1	10011	0.01463	95.83%	10
2	1897	0.01000	95.83%	1
3	10004	0.01232	95.83%	10
4	1835	0.01000	95.83%	2
5	1804	0.01000	95.83%	1
Best	1804	0.01000	95.83%	1
Average	5110.2	0.01139	95.83%	4.8

Table (3) Results of Gre_UC-NN trained by BP.

Exp.	NoI	MSE	AL %	Time/sec.
1	326	0.00873	100%	1
2	766	0.00867	100%	1
3	983	0.01000	95.83%	1
4	554	0.00952	95.83%	1
5	746	0.00998	95.83%	1
Best	326	0.00867	100%	1
Average	675	0.00938	97.5%	1

7.2 GLR-NN Trained by PSO

In this subsection we will train Letter recognition NN (LR-NN) with layers (81-10-5) by using PSO. The results of GLR-NN trained by PSO for 5 experiments for each sample of the NN mentioned in section (8) are shown in tables (4) and (5).

Table (4) Results of Gr_LC-NN trained by PSO.

Exp.	NoI	MSE	AL %	Time/sec.
1	1571	0.009970	91.67%	65
2	1699	0.009990	91.67%	70
3	1940	0.009968	95.83%	81
4	1208	0.009917	95.83%	50
5	1650	0.009990	91.67%	69
Best	1208	0.009917	95.83%	50
Average	1613.6	0.009967	93.334	67

Table (5) Results of Gre_UC-NN trained by PSO.

Exp.	NoI	MSE	AL %	Time/sec.
1	2305	0.009955	95.83%	96
2	1622	0.009840	100%	69
3	1805	0.009956	100%	76
4	1961	0.009900	100%	83
5	1323	0.009923	100%	55
Best	1323	0.009840	100%	55
Average	1803.2	0.0099148	99.166	75.8

8. Comparison Study of NN Trained by BP and PSO

In the PR-problems we chose the same two NN's discussed before, the neural network is GLR-NN with Ns=24 for Greek language, for our comparative study between PSO and BP. The two NN's start from the same initial random weights for GLR-NN, in order to guarantee that the comparative is being fairer. The training is done for 5 experiments for GLR-NN. The results of training are shown in table (6):

Table (6) Comparison results between PSO and BP for Gr_LC-NN.

Exp	Algorithm	NoI	MSE	AL%	Time/sec
1	BP	10006	0.01362	95.83%	10
	PSO	1844	0.00991885	95.83%	73
2	BP	10001	0.01123	95.83%	10
	PSO	1508	0.00994792	95.83%	55
3	BP	10004	0.01353	95.83%	11
	PSO	1611	0.00999919	91.67%	64
4	BP	10008	0.01461	95.83%	12
	PSO	2278	0.00993319	91.67%	96
5	BP	10000	0.01344	95.83%	11
	PSO	2039	0.00999303	95.83%	82
Best	BP	10000	0.01123	95.83%	10
Average		10003.8	0.013286	95.83%	10.8
Best	PSO	1508	0.00991885	95.83%	55
Average		1856	0.00995843	91.67%	74

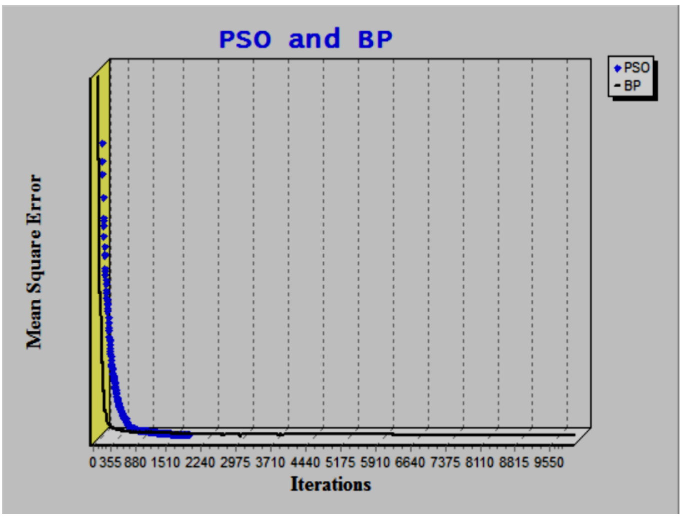

Figure (5) chart of curves of training of Gr_LC-NN using BP and PSO.

Figure (5) represents a chart which describes the difference in curves between the training of Gr_LC-NN using BP and PSO, these results taken from experiments (1).

Table (7) shows the comparison results between PSO and BP for Gr_UC-NN.

Table (7) Comparison results between PSO and BP for Gr_UC-NN.

Exp	Algorithm	NoI	MSE	AL%	Time/sec
1	BP	1135	0.00997	95.83%	1
	PSO	1178	0.00990224	100%	48
2	BP	288	0.00974	100%	1
	PSO	1647	0.00996093	100%	68
3	BP	848	0.00998	95.83%	1
	PSO	1171	0.00990244	100%	48
4	BP	286	0.00922	100%	1
	PSO	1194	0.00988632	100%	49
5	BP	320	0.00985	100%	1
	PSO	1201	0.00986088	100%	47
Best	BP	320	0.00922	100%	1
Average		575.4	0.009752	97.915	1
Best	PSO	1171	0.00986088	100%	47
Average		1278.2	0.00990256	100%	52

Figure (6) represents a chart which describes the difference in curves between the training of Gr_UC-NN using BP and PSO, these results taken from experiments (1).

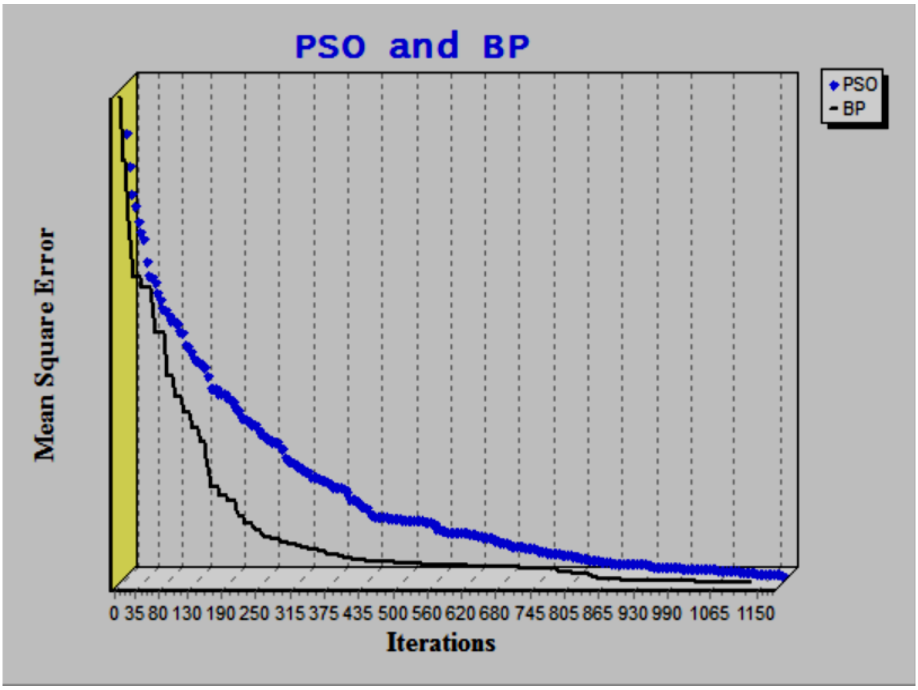

Figure (6) chart of curves of training of Gr_UC-NN using BP and PSO.

9. Analytical Evaluation of Training Results

In this section we try to evaluate the two training algorithms, BP and PSO, of NN's of our study cases. The evaluation process results based on the results of tables (6) and (7) of previous section.

We will give a pass degree for each training algorithm as follows:

1. If PB is better than PSO then the evaluation degree is -1.
2. If PSO is better than BP then the evaluation degree is 1.
3. If they are even then the evaluation degree is 0.

Table (8) shows the evaluation process results of the BP and PSO in training the neural network GRL-NN.

Table (8) Evaluation of the BP and PSO in training GLR-NN.

NN	Alg.	Fun.	NoI	MSE	AL%	Time	Sum	Better Alg.
Gr _LC-NN	BP	Best Av.	- -	- -	0 -1	-1 -1	-1 -2	PSO
	PSO	Best Av.	1 1	1 1	0 1	- -	2 3	
Gr_UC-NN	BP	Best Av.	-1 -1	-1 -1	0 -	-1 -1	-3 -3	BP
	PSO	Best Av.	- -	- -	0 1	- -	0 1	
Sum	BP	Best Av.	-1 -1	-1 -1	0 -1	-2 -2	-4 -5	-9
	PSO	Best Av.	1 1	1 1	0 2	- -	2 4	6
Final Sum		Best Av.	0 0	0 0	0 1	-2 -2	-2 -1	BP
Better Algorithm			0	0	1	-4	-3	

10. Conclusions

This paper concludes the following aspects:

1. The PSO algorithm has fewer parameters to tune, thus it is more universal tool, and can be used to locate or track stationary as well as nonstationary extremes.
2. The BP algorithm, however, is a local search method. It is easily falling into local minima and fails to find the global optimum when used to train a perceptron, while PSO algorithm, in spite of its population stochastic search method, it is less in falling in local minima and it can find global weights with both large probability and fast convergence rate during the training of NN.
3. The binary BP algorithm performs better than the PSO algorithm in Greek Letter Recognition. For this reason the BP is better than PSO in NN's learning.

4. In the four criteria of NN training for all cases study, we notice the following aspects from table (8):
   i. NoI and MSE: the two algorithms are performing even.
   ii. AL: in Best function the two algorithms are performing even, while in Average function PSO is performs better.
   iii. T: in Best and Average BP is performs better than PSO.
5. We can add some input binary strings treated as a wrong or noise in the shape of the character or the letter. This mean increasing in the number of training set (Ns).

References

[1]. Clow B. "A Comparison of Neural Network Training Methods for Character Recognition", Department of Computer Science Carleton University, 95.495, 2003.

[2]. Shi Y., "Particle Swarm Optimization", Electronic Data Systems, Inc. Kokomo, IN 46902, USA Feature Article, IEEE Neural Networks Society, February 2004.

[3]. Kennedy J. and Eberhart R. C. "Particle Swarm Optimization", Proceedings of IEEE International Conference on NN, Piscataway, pp. 1942-1948, 1995.

[4]. Eberhart R. C. and Hu X. "Human Tremor Analysis Using Particle Swarm Optimization" Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1999), Washington D.C. pp. 1927-1930, 1999.

[5]. Eskandarian A., and et al, "Vehicle Crash Modeling Using Recurrent Neural Networks", Math. & Computer Modeling vol. 28, No. 9, pp. 31 - 42, 1998.

[6]. Haykin S., "Neural Network–a Comprehensive Foundation", prentice Hall, 1999.

[7]. Zurada J. M., "Introduction to Artificial Neural Network", Jaico Publishing House, 1996.

[8]. Amin, S.M. and Rodin E. Y., "Neurocontrol of Nonlinear System via Local Memory Neurons", Math. & computer Modeling vol. 27, No. 3, pp. 65-92, 1998.

[9]. Pomeroy P. "An Introduction to Particle Swarm Optimization", Article, Mar, www.adaptiveview.com, pages. 1-7, 2003.

[10]. Ribeiro P. F. and Kyle W. S., "A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control", Member, 2003.
http://engr.calvin.edu/WEBPAGE/.../engir302/Reactivepower-PSO-wks.pdf

[11]. Settles M. and Rylander B., "Neural Network Learning using Particle Swarm Optimizers", Advances in Information Science and Soft Computing, pp. 224-226, 2002.

[12]. Parsopoulos K. E. and Vrahatis M.N., "Recent Approaches to Global Optimization Problems through Particle Swarm Optimization", Kluwer Academic Publishers, Netherlands, Natural Computing 1, pp 235–306, 2002.

[13]. Hu X., Eberhart R.C. and Shi Y.,"Swarm Intelligence for Permutation Optimization: A Case Study of n-Queens Problem", Indiana, USA, 2003.

[14]. Denoeux T., "Pattern Recognition for System Monitoring - An Overview", University of Compiègne CNRS (National Center for Scientific Research), France, Oct., 2002.

تمييز الانماط باستخدام الأنظمة الذكية

م. فائز حسن علي
الجامعة المستنصرية
كلية العلوم/ قسم الرياضيات

م.م. حازم نوري عبد الرزاق
كلية الرافدين الجامعة
قسم هندسة اتصالات الحاسوب

المستخلص

الشبكات العصبية (NN) هي نظام معالجة البيانات الذي طور كنموذج لتعميم الادراك البشري لعلم الاحياء البيولوجية.

تم في هذا البحث اجراء دراسة مقارنة فيها مناقشة طريقتين مهمة لتدريب NN لتكون محور دراسة هذا البحث، وهما طريقة امثليه السرب الجزيئي (PSO) وطريقة الانتشار التراجعي (BP). هاتين الطريقتين وجدت لتدريب الشبكة من خلال تعديل أوزانها وهذا يعتمد على حساب قيمة دالة الصلاحية والتي تمثل هنا قيمة دالة العتبة .

تم في هذا العمل معالجة مسالة تمييز الانماط، والتي تعتبر احد اهم تطبيقات حقل التصنيف، حيث تم تمييز حروف اللغة الاغريقية (A..Ω). وقد تم الاخذ بنظر الاعتبار حالات الحروف العالية والواطئة.

النظام المقترح يحتاج عند التنفيذ الى Microsoft Windows بكافة أجياله الحديثة، وقد تمت برمجة النظام باستخدام الجيل السابع من لغة Delphi.