Eng. & Tech. Journal, Vol.29, No. 13, 2011

Training Artificial Neural Network Using Back-Propagation & Particle Swarm Optimization for Image Skin Diseases

Dr. Hanan A. R. Akkar* & Samem Abass Salman*

Received on: 2/5/2011
Accepted on: 21/7/2011

Abstract

This work is devoted to compression Image Skin Diseases by using Discrete Wavelet Transform (DWT) and training Feed-Forward Neural Networks (FFNN) by using Particle Swarm Optimization(PSO) and compares it with Back-Propagation (BP) neural networks in terms of convergence rate and accuracy of results .The comparison between the two techniques will be mentioned. A MATLAB 6.5 program is used in simulation.

The structure Artificial Neural Network (ANN) of training image skin diseases is proposed as follows:

1- The proposed structure of NN that performs three compressions Images Skin training by BP algorithms with log sigmoid activation function, and three neurons in output layer.

2- The proposed structure of FFNN using PSO that performs three compressions Images Skin with hardlim activation function, and three neurons in output layer.

The results obtained using PSO are compared to those obtained using BP. Learning iterations (602-4700 epoch), convergence time (1sec.- 100 sec.), number of initial weights (1set - 75set), number of derivatives (0 - 38 derivatives) and accuracy (60% - 100%) are used as performance measurements. The obtained Mean Square Error (MSE) is 10^{-7} to check the performance of algorithms. The results of the proposed neural networks performed indicate that PSO can be a superior training algorithm for neural networks, which is consistent with other research in the area.

Keywords: Artificial Neural Network (ANN), Particle Swarm Optimization (PSO), Back-Propagation (BP), Wavelet Transforms (WT).

تدريب الشبكة العصبية الإصطناعية بأستخدام أمثلية الحشد الجزيئي و الأنتشار العكسي للصورأمراض الجلد

الخلاصة

هذا العمل مكرس لدراسة ضغط صور لامراض الجلد بواسطة استخدام محول الموجات المنفصلة (DWT)ثم يتم تدريبها بواسطة الشبكة العصبية الاصطناعية باستخدام امثلية الحشد الجزيئي (PSO) ومقارنتها بالانشار العكسي (BP) من حيث (الدقة –عدد hidden layer- الزمن –الاوزان –عدد التكرار للتدريب) .استخدام برنامج الماتلاب 6.5 للحصول على النتائج.

تم أقتراح مجموعه من تصاميم الشبكات العصبية الإصطناعية التي تدرب عدد من صور أمراض الجلد وهي كما يلي :-

1- التصميم المقترح للشبكة العصبية الإصطناعية التي تنفذ ثلاثة صور من أمراض الجلد بواسطة (BP) الانتشار العكسي.

2- التصميم المقترح للشبكة العصبية الإصطناعية التي تنفذ ثلاثة صور من أمراض الجلد بواسطة (PSO) أمثلية الحشد الجزيئي .

*Electrical Engineering Department, University of Technology/ Baghdad

https://doi.org/10.30684/etj.29.13.12
2412-0758/University of Technology-Iraq, Baghdad, Iraq

وبمقارنة النتائج التي ظهرت بإستخدام خوارزمية PSO المقترحة مع تلك النتائج التي ظهرت بإستخدام خوارزمية الإنتشار العكسي وهي (((التكرار (602-4700 دورة) ، الزمن (1ثانية الى100 ثانية) ، عدد الأوزان الإبتدائية (1 مجموعة الى 75 مجموعة) ، عدد المشتقات (0 إلى 38 مشتقة) والدقة (60% الى 100%))) أستخدمت كمقاييس للمقارنة. وبمعدل مربع الخطأز 10^{-7} وجدت أن نتائج الشبكات العصبية الاصطناعية المقترحة بواسطة أفضلية الحشد الجزيئي أفضل خوارزمية لتدريب الشبكات العصبية الإصطناعية والتي تطابق البحث في هذا المجال من حيث السرعة ، الدقة و الكفاءة

1. Introduction

A. Artificial Neural Network (ANN)

An Artificial Neuron (AN) is a model of biological neuron, where each AN receives signals from the environment or other ANs, gathers these signals applying some activation function to the signals sum, and when fired transmits signal to all connected neurons. Input signals are inhibited or excited through positive or negative numerical weights associated with each connection to AN, the firing of the AN and the strength of the exciting signal are controlled via a function referred to as activation function. The AN collects all incoming signals and computes a net input signal as a function of the respective weights. The net input serves to the activation function which calculated the output signal of the AN. An ANN is a layered network of artificial neurons. ANN may consist of input, hidden and output layers. ANs in one layer are connected fully or partially to the ANs in the next layer [1].

The AN Output (N) is

$$N_i\,(x_1, x_2, \ldots, xm_i) = a_i\left(\textstyle\sum_{1\le j\le mj}(wji * xj + bj)\right). \quad \ldots\ (1)$$

Where x_j is the input signal, w_{ji} is the weight, aj is the activation function and b_j bias weight.

There are several methods of training ANN Back-propagation is by far the most common. In this research PSO are Supposed as the best training algorithm to our application.

B. Back-Propagation Algorithm (BP)

The Back-Propagation (BP) algorithm was proposed in 1986 by Rumelhart, Hinton and Williams for setting weights and hence for the training of Multi-Layer Perceptrons (MLP) [2].

The BP algorithm propagates backward the error between the desired signal and the network output through the network. After providing an input pattern, the output of the network is then compared with a given target pattern and the error of each output unit calculated. This error signal is propagated backward, and a closed-loop control system is thus established. The weights can be adjusted by a gradient-descent-based algorithm. In order to implement the BP algorithm, a continuous, nonlinear, monotonically increasing, differentiable activation function is required. The two most-used activation functions are the logistic function equation (2) and the hyperbolic tangent function equation (3), and both are sigmoid functions [3].

$$F\left(net\right) = \frac{1}{1 + e^{-net}} \qquad (2)$$

$$F(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \qquad (3)$$

$F(net)$ Represent the actual output.

The Back-Propagation (BP) algorithm to compute weights of neurons may tend to instability under certain operation conditions. To reduce the tendency to instability, Rumelhart in (1986) suggested to add a momentum term (a), a is the momentum coefficient in the range of $0 < a < 1$, usually around 0.9). The employment of a will tend to avoid fast fluctuations, but it may not always work, or could even harm convergence [2]. Another parameter suggested improving learning process, it is a learning rate h. Weight changes in back propagation are proportional to the negative gradient of the error. This guideline determines the relative changes that must occur in different weights when a training pattern is presented. But, it does not fix the exact magnitudes of the desired weight changes. These magnitude changes are dependent on the learning rate. A large learning rate leads to rapid learning but the weights may oscillate, while lower learning lead to slower learning [4].

C. Particle Swarm Optimization (PSO)

Particle swarm optimization is a population based evaluation optimization technique developed by J. Kennedy and R. Eberhart in 1995 motivated by the social behavior of bird flocking or fish schooling [5].

PSO is a kind of random search algorithm that simulates nature evolutionary process and performs good characteristic in solving some difficulty optimization problems. The basic concept of PSO comes from a large number of birds flying randomly and looking for food together. Each bird is an individual and called a particle. As the birds looking for food, the particles fly in a multidimensional search space looking for the optimal solution. Here all the particles are composed of a family rather than the isolated individual for each other; they can remember their own flying experience and share their companion's flying experience [6].

The basic PSO model consists of swarm of particles moving in a D-dimensional search space. The direction and distance of each particle in the hyper dimensional space is resolute by its fitness and velocity. In general the fitness is primarily related with the optimization objective and velocity is updated according to a sophisticated rule [7].

In PSO, populations starts with random initialization of individuals in the search space and then repeat the social behavior of the particles in the swarm till achieves the best possible result by iterative searching. At each iterative step the velocity (position change) is updated and the particle is moved towards a new position. The best previously visited position at the n^{th} particle is denoted by the personal best position pbest, while the position of the best individuals of the whole swarms is denoted as the global best position gbest. In other words, the particle swarm optimization idea consists of, at each time step, changing the velocity and location of each particle towards its pbest and gbest locations according to (4) and (5):

$$V_{id} = W*V_{I+}C_1*rand1*(P_{id}-X_{id}) + C_2*rand2*(P_{gd}-X_{id}), \quad (4)$$

$$X_{id} = X_{id} + V_{id}. \quad (5)$$

Where W is the inertia weight which bring stability between global exploration and local exploration, C_1 and C_2 are two constants called learning factors [8], rand1 and rand2 are two independent random numbers uniformly spread in the range of [0, 1], For (4) the first part represents the inertia of previous velocity; the second part is the cognition part which represents the private thinking by itself; the third part is the social part, which represents the assistance among the particles [9]. P_{jd} represent personal best position recorded by particle i and P_{gd} is the global position and d is the index of dimension in the search space.

During the past few years PSO has been shown successful for many applications [10- 12] several papers discuss how to apply PSO in training NNs and their advantages [13- 15].

For the purpose of NNs learning the empirical error referred to as the objective function (Mean Square Error) (MSE) to be optimized by the optimization method (minimized to 0) is
given by:

$$MSE=\sqrt{\frac{\sum_{j=1}^{n}\sum_{k=1}^{m}(Tjk-Yjk)^2}{nm}} \qquad (6)$$

Where n is the number of training patterns, m is the number of outputs, T is the target and Y is the actual value [16].

D. Wavelet Transforms (WT)

Wavelets are functions which allow data analysis of signals or images, according to scales or resolutions. The processing of signals by wavelet algorithms in fact works much the same way the human eye does; or the way a digital camera processes visual scales of resolutions, and intermediate details. But the same principle also captures cell phone signals, and even digitized color images used in medicine. Wavelets are of real use in these areas, for example in approximating data with sharp discontinuities such as choppy signals, or pictures with lots of edges.

The applications of the wavelet idea include big parts of signal and image processing, data compression, fingerprint encoding, and many other fields of science and engineering [17].

In this work, will be discussed briefly an image compression, the nature of wavelets, and some of the salient features of image compression technologies using wavelets. Since this is a very rapidly evolving field, only the basic elements are presented show in figures (1), (2) and (3).

2. Theory Descreption

There has been a significant increase in research and development in the area of applying Evolutionary Computation (EC) techniques for the purposes of evolving one or more aspects of ANNs. These EC techniques have usually been used to evolve NN weights, but sometimes have been used to evolve network structure or the network learning algorithm [18].

The first applications of PSO involved accelerating the evaluation of the transfer functions in NNs. This phase is often called training when the network itself modifies according to data provided gradually to entries and to the expected results, it is called training because the evaluations are imposed by a third party, in fact, precisely, by PSO [19].

A new learning algorithm combined ANN was proposed to determine the optimal weights. The weights of the ANN are adjusted by PSO. Because PSO has the probabilistic mechanism and multi-starting points, hence the PSO can avoid getting into the local optimal solutions [20]. Numerous studies have further explored the power of PSO as a training algorithm for a number of different NN architectures. Studies have also shown for specific applications that NNs trained using PSO provide more accurate results [21]. In order to train FFNN by using PSO and compare the performance with BP, the following steps have been followed show in figure (4):

a) Determine training image.

b) Define neural network architecture.

c) Determine network parameters.

d) Run FNNPSO program or Run FNNBP program.

e) Comparison and analysis.

The first real implementation of the particle swarm algorithm was a model that bridges psychological theory and engineering applications. The FFNN is a statistical model of cognition that inputs vectors of independent variables and outputs estimates of vectors of dependent variables. The network is structured as a set of weights, usually arranged in layers, and the optimization problem is to find values for the weights that make the mapping with minimal error [18].

For the purpose of NN learning, the empirical error in equation (6) is referred to as the objective function to be optimized by the optimization method. Several optimization algorithms for training NNs have been developed. These algorithms are grouped into two classes [1]:

• *Local optimization*, where the algorithm may get stuck in a local optimum without finding a global optimum. Back-propagation algorithm is an example of local optimizers.

• *Global optimization*, where the algorithm searches for the global optimum by employing mechanisms to search larger parts of the search space. Global Optimizers include evolutionary algorithms and PSO.

In order to use PSO to train an NN, a suitable representation and fitness function needs to be found. Since the objective is to minimize the error function, the fitness function is simply the given error function (e.g. the MSE given in equation (6)). Each particle represents a candidate solution to the optimization problem, and since the weights of a trained NN are a solution, a single particle represents one complete network. Each component of a particle's position vector represents one NN weight or bias, using this representation [1].

The particle represents the weight vector of NN, including all biases. The dimension of the search space is therefore the total number of weights and biases. The fitness function is the Mean Squared Error (MSE) over the training set as in equation (6) [21]. Changing the position means updating the weight of the network in order to reduce the error. All the particles update their position by calculating the new velocity, which they use to move each particle to the new position. The new position is a set of new weights used to obtain the new error. For PSO, the new weights are adapted even though no improvement is observed. This process is repeated for all the particles. The particle with the lowest error is considered as the global best particle so far. The training process continues until satisfactory error is achieved by the best particle or computational limits are exceeded. When the training ends, the weights are used to calculate the classification error for the training patterns. The same set of weights is used then to test the network using the test patterns.

There is no back-propagation concept in FNNPSO, where the FFNN produced the learning error (particle fitness) based on set of weight and bias (PSO positions). The Pbest value (each particle's lowest learning error so far) and Gbest value (lowest learning error found in entire learning process so far) are applied to the velocity update equation (4) to produce a value for positions adjustment to the best solution or targeted learning error. The new sets of positions (NN weight and bias) are produced by adding the calculated velocity value, equation (4), to the current position value using movement equation (5). Then that new set of positions is used for producing new learning error (particle fitness) in FFNN.

3. The Proposed Design of Training ANN using BP& PSO for Image Skin Diseases

The objective of this paper is to develop the BP neural network (ANN) and PSO neural network model that could training three images skin diseases. Although only the colour indices associated with image pixels the used as inputs, it was assumed that the ANN model could develop the ability to use other information, such as shapes, implicit in these data .The type of images skin diseases the RGB (128 × 128pixel). Images the taken to compression using Wavelet Transform (WT) .The first part convert image from RGB (128 × 128pixel) to Gray scale (128 × 128pixel).

After convert image to Gray scale (128 × 128pixel) get in to DWT that convert image Gray scale (128 × 128pixel) to Gray scale (8 × 8pixel) as an input data to input layer networks of PSONN or BPNN.

A NN is composed of a series of interconnected nodes and the corresponding weights between them. It aims at simulating the complex mapping between the input and output. A 3-layer feed forward ANN is used consists of input units, hidden units and output units. Let Wih denotes the weight between the input node and the hidden one. Likewise, Who denotes the weight between the hidden node and output one. ANN is characterized by the ability of self learning and error toleration. With the appropriate activation functions and trained weights, ANN can approximate any smooth, nonlinear function or relationship between the input and output. The training process is carried out on a set of data including input and output parameters. Usually, the data are split into two parts namely training samples and testing samples.

The learning procedure is based on the training samples and the testing samples are used to verify the performance of the trained network. During the training, the weights in the network are adjusted iteratively till a desired error depicted as equation (3.1) is obtained.

$$E=\frac{\sum_{i=1}^{m}\sum_{k=1}^{n}(t_i^k-y_i^k)^2}{2} \qquad (7)$$

Where, tik and yik represents the actual and the predicted function values respectively, m is the number of training samples, and n is the number of output nodes. The neural network is trained by minimizing the above error function in a search space based on weights. PSO generates possible solutions and measure their quality by using a forward propagation through the neural network to obtain the value of the error function. This error value is used as the particle's fitness function to direct it toward the more promising solution. The global best particle is corresponded to the desired trained network after adequate iterations.

For the purpose of NN implementation of training image ,the MATLAB PSO tools are modified to be suitable with this application.

The Modified PSO will give the exact integer weight needed for the training of the network, where the modified PSO algorithm will search the optimum weights in an integer search space only, thus it takes less effort and time finding the optimum weights, minimizing the error to zero which means 100% percent accuracy obtained. These integer weights will be restricted between minimum integers values needed for the training process based on trial and error show in figures (5), (6) and(7).

4. The Experimental Results

In this work, used two programs have been developed which are Feed-Forward Neural Network Particle Swarm Optimization (FNNPSO) and Feed-Forward Neural Network Back-Propagation algorithm (FNNBP) for training (One, Two and Three) Images Skin Diseases.

The results for each neural network are compared and analyzed based on the convergence rate and the accuracy of results. A MATLAB program is used to implement the simulation.

4.1 Simulation Results of the 1st Image Skin Diseases using FNN- BP and PSO

Table (1) shows the Simulation Results of the 1st Image Skin Diseases using FNN- BP and PSO show that FNNPSO convergence time depends on the swarm size and search space, so that, sometimes is fast and another times is slow depending on the initial weights, which are randomly generated in search space, These initial weights are changed over entire the search space when the run of program is repeated. Thus different convergence time is carried out for each run. Now for image diseases test. The convergence time for FNN learning based on PSO algorithm is 10 seconds at 20 iterations, compared with FNNBP, where it takes 4.4380 seconds at iteration 602 for overall learning process as show in table (1).

Both algorithms converged using the minimum error criteria. For the correct accuracy percentage, it shows that FNNPSO result is better than FNNBP with 100% compared with 97.5% show in figure (8) and (9).

4.2 Simulation Results of the 2nd Images Skin Diseases using FNN- BP and PSO

Table(2) shows the Simulation Results of the 2nd Images Skin Diseases using FNN- BP and PSO show that FNNPSO convergence time is slow, where it takes 30562sec at 1800 iterations compared with FNNBP, where it takes 12.130 sec at iteration 1000 for overall learning process.

Because the number of fitness evaluation (calculate minimum error for each bit and comparison with Pbest). Both algorithms are converged using the minimum error criteria. For the correct accuracy percentage, it shows that FNNPSO result is better than FNNBP with 100% compared to 85.5% but FNNBP convergence time is faster at 1000 iterations compared with 1800 iterations in FNNPSO as show in figure (10) and (11).

4.3 Simulation Results of the 3rd Images Skin Diseases using FNN- BP and PSO

Table (3) shows the Simulation Results of the 3rd Images Skin Diseases using FNN- BP and PSO show that FNNPSO is the best in terms of accuracy. But the convergence time is slow because the learning process by PSO tries to check all particle positions in search space, which is the best position. Thus the learning process by PSO depends on the swarm size and search space compared to FNNBP, where it takes 38.021sec at iteration 1000 for overall learning process. Both algorithms are converged using the minimum error criteria. For the correct accuracy percentage, it shows that FNNPSO result is better than FNNBP with 100% compared with 60% show in figure (12) and (13).

The experiments are carried out to analyze the optimization algorithm called PSO that is applied on FNN to explore the classification accuracy and convergence time compared to BPNN. Based on the results, it is clear that FNNPSO is better than FNNBP in terms of multi-starting points (initial weights), size of search space and accuracy of results.

In FNNPSO, network architecture and selection of network parameters for the pattern influence the convergence and the performance of network learning. By reducing the hidden nodes, it minimizes the number of weights (position) and also reduces the problem dimension.

In this work, there are several times that FNNPSO runs with reduced number of hidden nodes, and the results have proven that the learning becomes faster. However, to have fair comparison, Kolmogorov theorem is chosen in this work and both algorithms need to use the same network architecture. Choosing FNNPSO parameters also depend on the problem and pattern to be optimized. This parameter can be adjusted to achieve better optimization. But to have better comparison in this study, the same parameters for all patterns have been used. For FNNBP, learning rate and momentum rate are the same for all pattern.

The results in Figures (5, 7, and 9) indicate that the Convergence Time and Learning Iterations of Back-Propagation method is suboptimal compared to PSO. The suboptimal Convergence is not as noticeable, by comparison, in the smaller data case.

Conversely, the results in Figures (6, 8, and 10) indicate that PSO algorithm is more robust with more consistent convergence characteristics. Thus, PSO algorithm can be applied to wide-range nonlinear optimization problems with reliable performance. In addition, PSO and similar algorithms do not add a considerable computational burden.

5 Conclusions

This paper has presented the implmentation of ANNs trained by PSO learning algorithm to image skin diseases. NNs are trained by minimizing the MSE function in search space based on weights. PSO generates possible solutions and measure their quality by using a forward propagation through the NN to obtain the value of error function (minimized error to zero). This error value is used as the particle's fitness function to direct it towards more promising solution. The global best particle corresponded to the desired trained after adequate iterations. The result of training FFNN using PSO has shown that PSO is an efficient alternative to FFNN ordinary training algorithms like BP. Results have shown also that PSO training algorithm has more accurate results than other training algorithms.

MATLAB modified PSO toolbox is used in the training of the ANNs Minimum weights required to get the higher accuracy (100%) is calculated using trial and error method.

6 Summary

In this paper, the latest optimization algorithm called PSO is applied in FFNN to improve the neural network learning. PSO is an optimization algorithm based on SI. It directs the search through the intelligence generated from cooperation and competition among the individuals. The merits of this proposed algorithm can be summarized as follows:

1. Used DWT to compression image (8×8) .
2. Neural Network is very fast detections because the

Topology of this NN and the input values from features extraction stage.

3. Training ANN Using PSO and BP for Image Skin Diseases.
4. Min and Max particle couldn't be used to discriminate between Skins Images.
5. With parallel search strategy, it can locate the global optimization consistently.
6. Its velocity – displacement search model is simple and easy to implement.
7. Few parameters should be considered and setup.
8. This work is to reduce the number of neurons needed for the training process, and also reduces the single neuron complexity by the abstraction of the multiplication process needed to multiply each input by the corresponding weight.
9. PSO optimization algorithm is the most suitable training algorithm required to our application, implementation of training Image Skin Diseases using ANN, because of its advantages, specially its free derivative activation function and multi starting points.

Refrences

[1] A. P. Engelbrecht, " Computational Intelligence: An Introduction", *John Wiley & Sons Ltd, ISBN: 978-0-470-03561-0, South Aferica*, 2007.

[2] D. Graupe, "Principle of Artificial Neural Networks", *World Scientific Publishing Co. Pte. Lte., 2007.*

[3] K. L. Du, M. N. Swamy, *"Neural Network in Soft Computing", Springer – Verlag, 2006.*

[4] J. Chen, *"Neural Network Applications in Agriculture Economics", Jianhua Chen, 2005.*

[5] J. Kennedy and R. Eberhart, " Particle Swarm Optimization", *IEEE Int . Conf. on Neural Networks, Australia*, PP. 1942-1948, 1995.

[6] L. Wang, X. Wang, J. Fu and L. Zhen, "A Novel Probability Binary Partical Swarm Optimization Algorithm and its Application", *Academy publisher, Journal of software, China*, Vol. 3, No. 9, December, 2008.

[7] X. Xie, W. Zhang and Z. Yang, "Adaptive Particle Swarm Optimization on Individual Level", *IEEE International Conference on Signal Processing (ICSP)*, China, PP. 1215-1218, 2002.

[8] R. Eberhart and Y. Shi, " Particle Swarm Optimization: Developments, Applications and Resources", *IEEE Int. Conf. on Evolutionary Computation*, PP. 81-86, 2001.

[9] J. Kennedy, "The Particle Swarm: Social Adaptation of knowledg", IEEE Int. Conf. on Evolutionary Computation, PP. 303-308, 1997.

[10] V. G. Gudise and G. K. Venayagamoorthy, "FPGA Placement and Routing Using Particle Swarm Optimization", *IEEE ,Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design*, 2004.

[11] G. Kendall and Y. Su, " A Particle Swarm Optimization Approach in the Construction of Optimal Risky Portfolios", *Proceedings of a 23rd IASTED International Multi-Conference Artificial Inteligence and Applications, Australia*, PP. 140-145, February, 2005.

[12] K. Chandramouli and E.Izquierdo, “Image Classification using Chaotic Particle Swarm Optimization”, *IEEE. ICIP*, PP.3001-3004, 2006.

[13] V.G. Gudise and G.K. Venayagamoorthy, “Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks”, *IEEE Symposium on Swarm Intelligence, USA*, PP. 110-117, 2003.

[14] Z. Cui, J. Zeng, G. Sun, "A Fast Particle Swarm Optimization", ISSN, International Journal of Innovative Computing, Information and Control, Vol. 2, No. 6, December, 2006.

[15] W. Zha, G. K. Venayagamoorthy, “Neural Networks Based Non Uniform Scalar Quantizer Design with Particle Swarm Optimization”, *IEEE Transactions*, June, 2005.

[16] J. Zhang, J. Zhang, T. Lok and M. Lyu, “ A hybrid Particle Swarm Optimization Back-Propagation Algorithm for Feedforward Neural Network Training”, *Applied Mathematics and Computation185*, PP. 1026-1037, 2007.

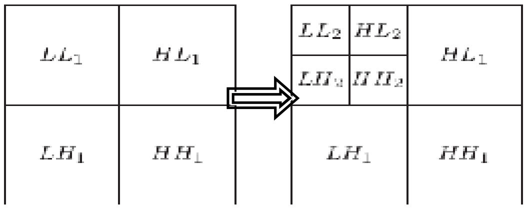
[17] Hung-Quoc Lai, Steven Tjoa\ *"ENEE631 Digital Image Processing: Wavelet-Based Image Compression".Dept. of Electrical and Computer Engineering, University of Maryland April, 2004.*

[18] G. K. Venayagamoorthy, R. G. Harley, *" Swarm Intelligence for Transmission System Control", IEEE, June, 2007.*

[19] M. Clerc, *"Particle Swarm Optimization", ISTE Ltd., 2006.*

[20] *M. Meissner, M. Schmuker and G. Schneider, “Optimized Particle Swarm Optimization (OPSO) and Its Application to Artificial Neural Network”BMC Bioinformatics, 1471-2105/7/125, 2006.*

[21] H. N. BIN Abdull Hamed, *" Particle Swarm Optimization for Neural Network Learning Enhancement", Master Thesis, University of Technology, Malaysia, June, 2006.*

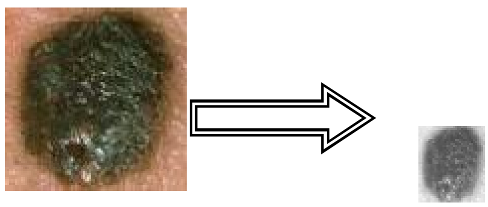


Original image

Decomposition image

Figure (1) Two-Level DWT Decomposition .

RGB TO Grayscale



(B) Grayscale- 8×8 pixels.

(A) Original Squamous Cell Carcinoma

Figure (2). (A) Original Squamous Cell Carcinoma (SCC) (RGB JPGE).
(B) Compression Image Squamous Cell

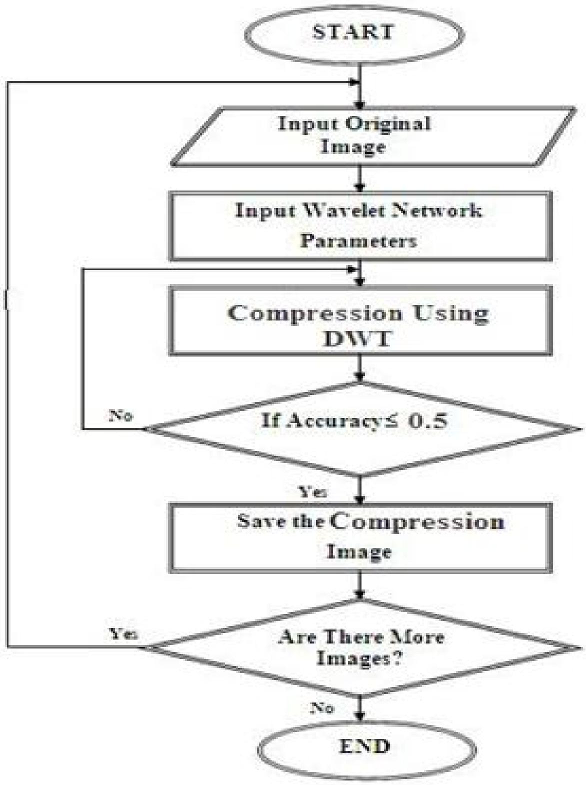


Figure (3) Flowchart of Compression Images using DWT

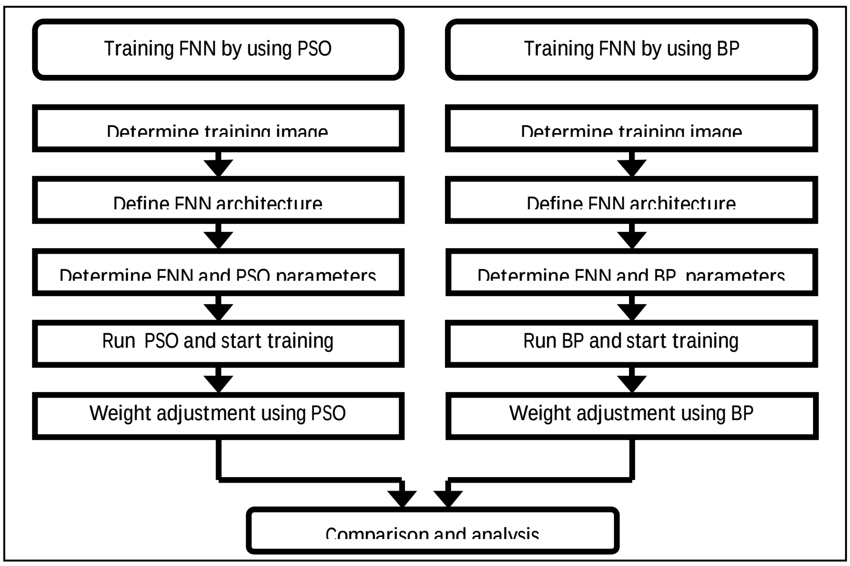
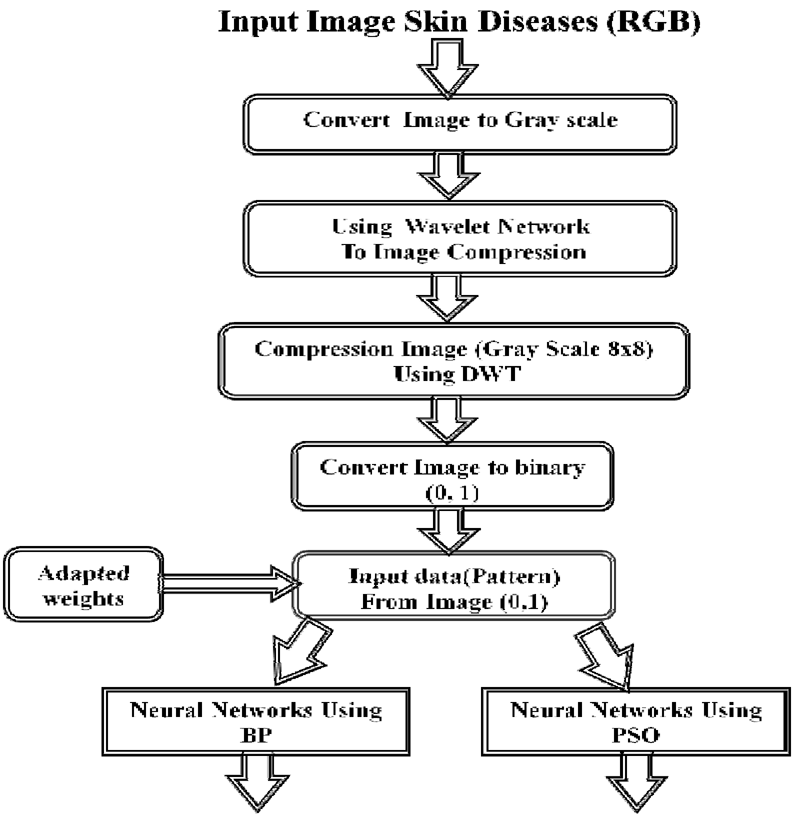


Figure (4) Framework of the Study Training ANN



Output Training ANN using BP for Image Skin Diseases

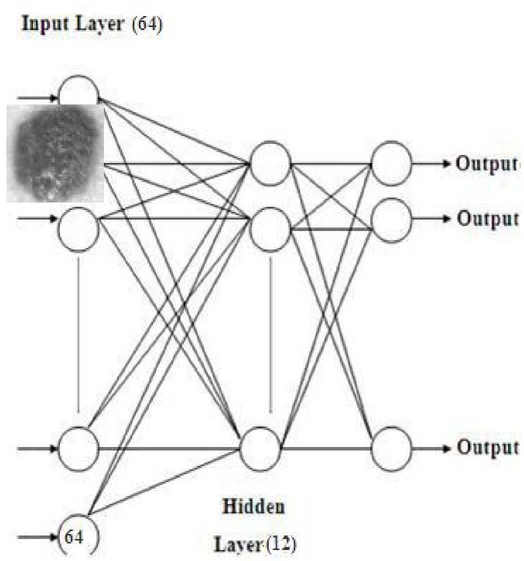


Figure (5) A block diagram training ANN using I

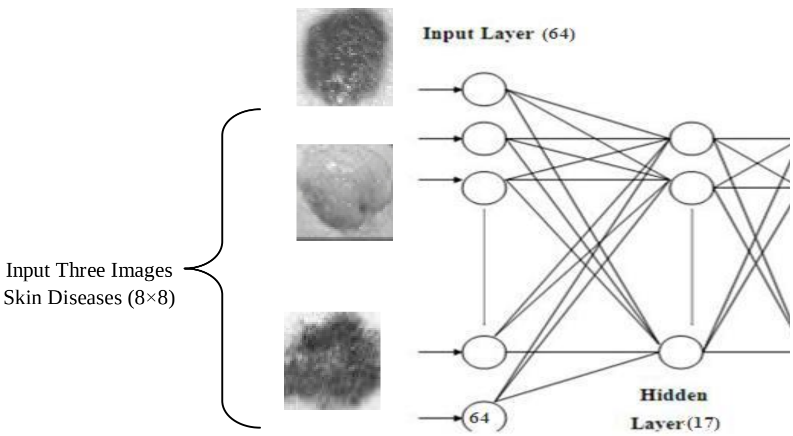
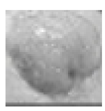


Figure (6) Training ANN Using BP Three Images Skin iseases.

Skin Diseases (8×8)



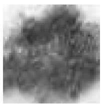
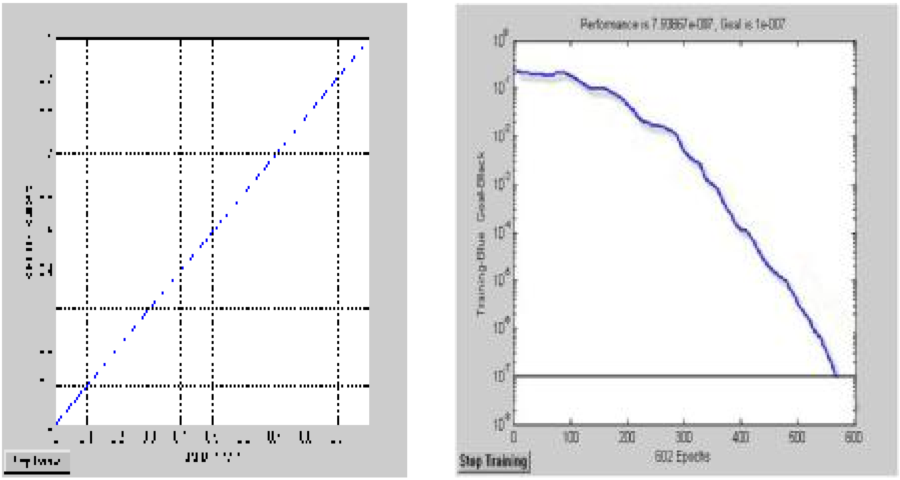


Figure (7) Training ANN Using PSO Three Images Skin Diseases.



(B)

(B) The relationship between desired and actual output.

Figure (8) (A) The Training FNNBP for the 1st Image Skin Diseases.
(B) The relationship between desired and actual output.

Table (1) Results of FNNPSO and FNNBP on the 1st Image

Results	FNNPSO	FNNBP
Learning Iterations	20	602
Error Convergence	4.33e-08	7.93819e-07
Convergence Time	10 sec	4.4380 sec
Accuracy (%)	100	97.5

Table (2) Results of FNNPSO and FNNBP on the 2nd Image Skin Diseases.

Results	FNNPSO	FNNBP
Learning Iterations	1800	1000
Error Convergence	1.3201-08	8.43268e-05
Convergence Time	30.562 sec	12.130 sec
Accuracy (%)	100	85.5

Table (3) Results of FNNPSO and FNNBP on the 3rd Image Skin Diseases.

Error Convergence	2.301e-07	0.0197712
Convergence Time	60sec	38.012 sec
Accuracy (%)	100	60%