The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Epilepsy Diagnosis Using PSO based ANN

Nesibe Yalçın[1], Cihan Karakuzu[1], Gülay Tezel[2]

[1] Bilecik Şeyh Edebali University, Turkey
[2] Selçuk University, Turkey
(Tel: 90-228-214-1412)

[1]nesibe.yalcin@bilecik.edu.tr

Abstract: Electroencephalogram (EEG) is used routinely for diagnosis of diseases occurring in the brain. It is a very useful clinical tool in classification of epileptic attacks and epilepsy diagnosis. In this paper, epilepsy diagnosis by evaluation of EEG records is presented. Artificial Neural Networks (ANN) is used as a classification technique. Particle Swarm Optimization (PSO) method, which doesn't require gradient calculation, derivative information and any solution of differential equations is preferred for ANN training. This training method is compared with back propagation algorithm, which is one of the traditional methods, and the results are interpreted. In case of using the PSO algorithm, the training and test classification accuracies are %99.67 and %100, respectively. PSO based neural network model (PSONN) has a better classification accuracy than back-propagation neural network model (BPNN) for epilepsy diagnosis.

Keywords: Artificial neural networks, back propagation algorithm, EEG, epilepsy diagnosis, particle swarm optimization.

1 INTRODUCTION

Epilepsy disease is a major brain disorder. Particularly, waveforms contained in EEG recorded during the occurrence of epileptic discharges can show similarity with waveforms of some other diseases occurring in the brain. So epilepsy disease cannot be detected easily [1]. EEG signals as shown in Fig. 1. and Fig. 2. are not periodic; their frequency, phase and amplitude continuously change. The changing forms of EEG signals are very complex and difficult to define and interpret [2], [3].

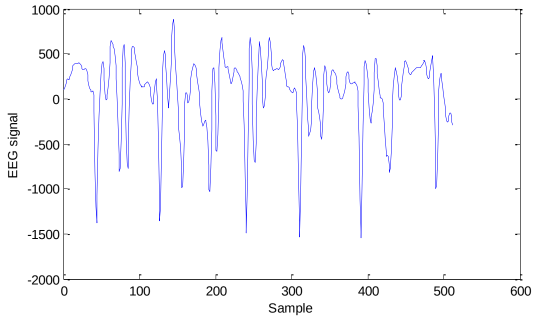

Fig. 1. Example of EEG signal for an epileptic person

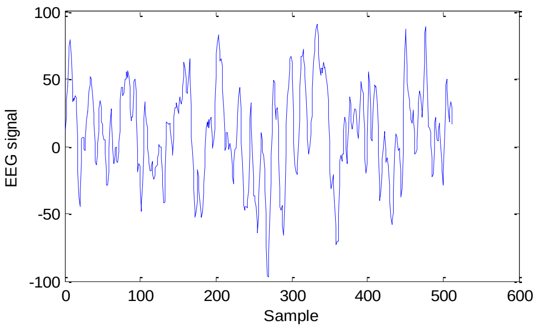

Fig. 2. Example of EEG signal for a healthy person

In recent years, there have been lots of studies using artificial intelligence techniques to recognize EEG signals. ANN as an artificial intelligence technique is used frequently in the classification of EEG signals. In addition to these techniques, heuristic optimization methods are used to increase the success, performance and / or the speed of these techniques. PSO which is one of the heuristic methods is successfully applied to the training of ANN. Because number of PSO parameters is small, PSO studies with real numbers, its realization and implementation are easy [4].

In this work, we aimed to able to diagnose epilepsy whether evaluate automatically EEG records with PSO based ANN. For this purpose, EEG signals taken from epileptic and healthy volunteers are normalized and then classified with ANN. PSO algorithm is used for training of ANN (PSONN) and performance of PSONN is compared with traditional ANN trained using back propagation algorithm (BPNN).

Following this introduction section, the rest of the paper is organized as follows: in the next section, materials and methods used in this study have been mentioned. In the third section, the ANN's trained with back propagation algorithm and the PSO method are emphasized. The fourth section provides information on the applications. The obtained results are presented with tables and the comparisons. The final section summarizes and concludes the results.

2 MATERIAL AND METHODS

2.1 EEG dataset

In this study, we used publicly available EEG data which is described in [5]. The complete data consists of five sets: Set A and Set B include data that has been taken healthy volunteers with eyes open and closed respectively. Set C and Set D taken from people, having epilepsy disease contain activities which has been measured in the interval has no attack. Set E includes only epileptic attack activities [5, 6].

Set A and Set E have been used in this work. Dataset consists of 1600 segments (800 segments for each class (epileptic and healthy)) and 512 samples for each segment. Dataset has been pre-processed using statistical features which are minimum, maximum, mean and standard deviation of each sample and thus number of samples in each segment has been reduced to 4. Then the new dataset has been normalized into the interval of [0, 1].

2.2 Neural network learned by back propagation (BPNN)

Back propagation [7], is one of error correction algorithms, is used in multi layer and feed forward ANN's training. A multi-layer back propagation network includes an input layer, at least one hidden layer and an output layer. Back propagation algorithm is a supervised learning method and aims to optimize weights and biases between input layer and the output layer depending on output error of the network. Input vector is given to input layer, and then reaches to the final output layer after passing through hidden layers. Each neuron in the network transmits the result to the all neurons of the next layer after receiving the arithmetical addition of the weighted signal from the previous layer's neurons, depending on the activation function.

ANN's training by back propagation is implemented consistently forward computing seen in Fig. 3. and backward computing given in Fig. 4.

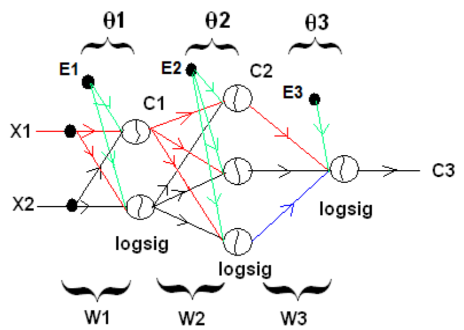

Fig. 3. Forward computing schematic structure

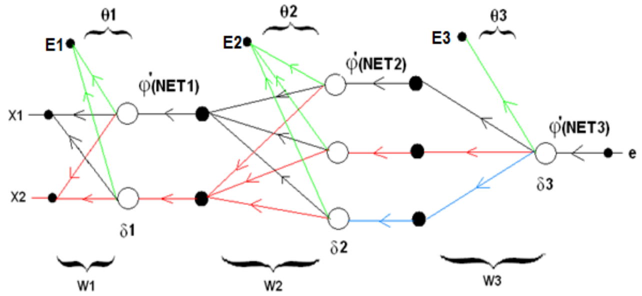

Fig. 4. Backward computing schematic structure (transpose network)

In Fig. 3., X1 and X2 are inputs; C1, C2 and C3 are output vectors of the layers. In Fig. 3. and Fig. 4., W1 and W2 are weight matrixes; W3, θ1, θ2 and θ3 are bias vectors; values of E1, E2 and E3 biases are chosen as 1. NET1, NET2 and NET3 are net input vectors for the related layer. sigmoid activation function φ is preferred for all neurons. φ' is derivative of the activation function. δ1, δ2 and δ3 are local gradient vectors.

2.3 Neural network learned by PSO (PSONN)

Particle swarm optimization (PSO) which is a heuristic method based on social behavior of bird flocking was developed by Eberhart and Kennedy in 1995 [8].

PSO algorithm starts with a set of random particles (candidate solutions for the problem) and then searches for an optimal solution updating its individuals by generations. In each generation, each particle is updated based on two special particles. The first one called "pbest" is the personal best particle for each particle found so far. The other one called "gbest" is a global best particle found so far by any particle in the swarm (population). Fig. 5. illustrates position and velocity updating of a particle.

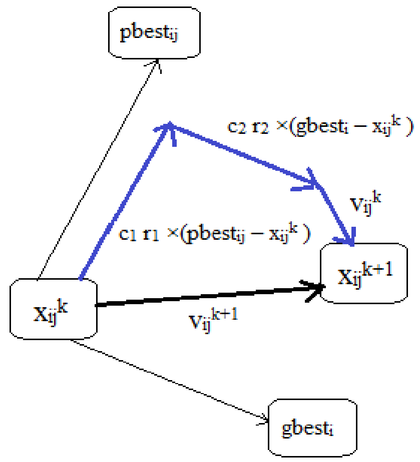

Fig. 5. The velocity and position updating of a particle

In Fig. 5., v_{ij} and x_{ij} variables are respectively the *j*th (j=1, 2, ..., D) velocity component and the *j*th position component of *i*th (i=1, 2, 3, …, N) particle at generation k. N is the number of particles in the swarm. D is dimension size of the search space. In an improved PSO version, the velocity updating and the position updating are determined by (1) and (2), respectively. In these equations, r_1 and r_2 are two random numbers uniformly distributed in the range of (0, 1). c_1 and c_2 are acceleration constants. w is inertia weight updated using (3) by generations. α variable in (3) is decrease factor and used to decrease linearly inertia weight.

$$v_{ij}^{k+1} = w^k v_{ij}^k + c_1 r_1 (pbest_{ij} - x_{ij}^k) + c_2 r_2 (gbest_i - x_{ij}^k) \quad (1)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2)$$

$$w^{k+1} = \alpha w^k \quad (3)$$

Limitations (V_{min} and V_{max}) defined in (4) determine the minimum and maximum velocity change of a particle during one generation. They are used to supply detailed searching, to allow the particles to escape local minima and to prevent the particles to leave the space research.

$$v_{ij}^{k+1} = \begin{cases} v_{ij}^{k+1}, & V_{min} < v_{ij}^{k+1} < V_{max} \\ V_{min}, & v_{ij}^{k+1} \leq V_{min} \\ V_{max}, & V_{max} \leq v_{ij}^{k+1} \end{cases} \quad (4)$$

In this study, we used PSO for training of ANN to obtain the best classifier model. During the training phase, we computed fitness value of each particle using Mean Squared Error (MSE) by (5) where, S is the number of segments in the training dataset, *e* is the difference between expected and obtained output after presenting *i*th segment (data) to the network. Structure of a particle (P_i) of PSO to train ANN is used as shown by (6).

$$MSE = \frac{1}{2S} \sum_{i=0}^{S} e_i^2 \quad (5)$$

$$P_i = [W1^i_{11} W1^i_{12} \ldots \theta1^i_{11} \ldots W2^i_{11} W2^i_{12} \ldots \theta2^i_{11} \ldots W3^i_{1} \ldots \theta3^i_{11} \ldots] \quad (6)$$

Algorithm's pseudo code is as following.

Initialize values for PSO and ANN
Initialize all weights with random values
Do
  for** each particle **do
    Adjust weights according to the particle
    Present training dataset to ANN
    Calculate MSE of the particle
  end for
  fitness value of gbest=min(MSE of all particles)
  if** stop criteria is not provided **then
    Update pbest and gbest particles
  end if
***while** stop criteria (maximum generation number or target fitness value of gbest) is provided*
Adjust weights according to gbest particle
Present testing dataset to ANN
Calculate output

3 EXPERIMENTAL STUDIES

In this work, we used EEG dataset belonging to epileptic and healthy volunteers. Then dataset divided to two subsets: training dataset which was formed by 1200 segments (600 epileptic) and test dataset which was prepared as 400 segments (200 epileptic).

PSONN and BPNN models have been developed via Matlab® Software Package without Neural Network Toolbox. Training dataset has been used to train these network models. Each network has been prepared as three layers: an input layer, a hidden layer and an output layer. The different number of particles, generations and neurons in the hidden layer have been used to determine optimum network model. Optimum numbers of these parameters has been experimentally obtained as 30, 200 and 3, respectively.

In order to determine training and test accuracies, the classification threshold value has been selected as 0.4. If output value is lower than this value, the classification result is 0 or healthy else 1 or epileptic.

Initialization values of α and w in (3) are chosen 0.975 and 0.9, respectively [9]. Acceleration constants are $c_1= c_2=2.1$. Limitations are $V_{min}= -0.1$ and $V_{max}= 0.1$. These values provided fast convergence to the target and increased the performance of network models.

Sensitivity, specificity and accuracy statistics are used frequently to determine the success of a classifier. Sensitivity is the proportion of data belonging to epileptic patients; specificity is the proportion of data belonging to healthy people; accuracy is true classification proportion [10]. These statistical measures are calculated by (7), (8) and (9).

$$sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$specificity = \frac{TN}{TN+FP} \quad (8)$$

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (9)$$

In the above equations, TP (True Positive) is the total number of epileptic patients which are correctly diagnosed, TN (True Negative) is the total number of healthy people which are correctly diagnosed, FP (False Positive) is the total number of epileptic patients which are incorrectly diagnosed, FN (False Negative) is the total number of healthy people which are incorrectly diagnosed.

PSONN is run 30 times for the training set. The classification accuracies and computed results of sensitivity and specificity analysis in the training and test dataset for BPNN and PSONN are given in Table1. The results about PSONNs are based on the best and worst run among 30 runs. The best, worst and average fitness values changing by generations are shown in Fig. 6.

Table 1. Performance of Comparison Networks

Performance Measures	Network Type		
	Best PSONN	Worst PSONN	BPNN
Fitness value of gbest / MSE	0.0041	0.2289	0.0009
Training classification acc.(%)	99.6667	50	99.8333
Test classification acc. (%)	100	50	90.7500
Training sensitivity	1	0.5	0.9967
Training specificity	0.9934	0	1
Testing sensitivity	1	0.5	0.8439
Testing specificity	1	0	1

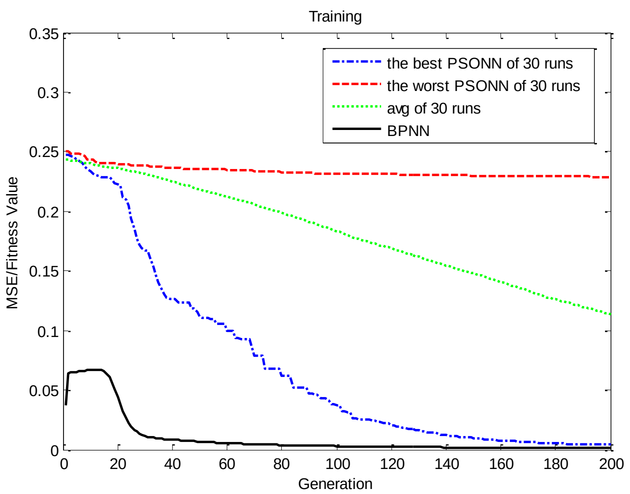

Fig. 6. Training graphics of PSONN and BPNN

4 CONCLUSION

In this work, PSO and back propagation (traditional) algorithms are used to train ANN for epilepsy diagnosis. Comparative results of PSONN and BPNN (traditional ANN) are presented in Table 1. According to table, training performance of BPNN is 99.83% and test performance of BPNN is 90.75%. Training and test classification results of best PSONN are too high (approximately 99.67% and 100%, respectively). The computed results of sensitivity analysis of PSONN are 1 in both the training and test dataset. Thus, it can be said PSONN is more successful than BPNN. PSONN can be adapted for different medical diagnosis problems and other PSO versions can be used to improve the success of the PSO as training algorithm.

This study shows any heuristic optimization method can be used as a learning algorithm to increase ANN's classification success.

REFERENCES

[1] Barışçı N, Müldür S (2003), Classification of epileptic EEG signals using neuro - fuzzy system. Journal of Polytechnic, vol. 6, No:2, pp. 445–449

[2] Toprak IB (2007), Analysis of EEG signals using the wavelet transform and artificial neural network. Süleyman Demirel University Graduate School of Applied and Natural Sciences, M.Sc. Thesis, Turkey

[3] Yazgan E, Korurek M (1996), Medikal Elektronik. ITU, Istanbul, No:15741, pp. 386

[4] Cavuslu MA, Karakuzu C, Karakaya F (2012), Neural identification of dynamic systems on FPGA with improved PSO learning. Applied Soft Computing, vol. 12, pp. 2707–2718

[5] Andrzejak RG, Lehnertz K, Mormann F, Rieke C, David P, Elger CE (2001), Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E, 64

[6] Tezel G, Özbay Y (2009), A new approach for epileptic seizure detection using adaptive neural network. Expert Systems with Applications, 36 (2009), pp. 172–180.

[7] Rumelhart D, McClelland J (1986), Parallel Distributed Processing. MIT Press, Cambridge, Mass

[8] Kennedy J, Eberhart RC (1995), Particle swarm optimization. Proc. IEEE int'l conf. on neural networks, IEEE service center, Piscataway, NJ, vol. 4, pp. 1942–1948

[9] Şevkli M, Yenisey MM (2006), A particle swarm optimization for the job shop scheduling problems. itüdergisi/d mühendislik, vol. 5, No: 2(1), pp. 58–68

[10] Özbeyaz A (2010), LS-SVM hyper parameter selection Based on PSO in EMG and EEG signals classification. University of Kahramanmaraş Sütçü Imam, Institute of Natural and Applied Sciences, Msc Thesis, Turkey.