



A modified brain emotional learning model for earthquake magnitude and fear prediction

Seyyedeh Hoorah Fakhroosavy¹ · Saeed Setayeshi² · Arash Sharifi¹

Received: 12 September 2017 / Accepted: 3 November 2017 / Published online: 17 November 2017
© Springer-Verlag London Ltd., part of Springer Nature 2017

Abstract

Brain emotional learning (BEL) model has been used frequently for predicting a quantity or modeling complex and nonlinear systems in recent years. In this research, two methods proposed for improving the efficiency of original BEL model using fuzzy rules, learning automata concepts and optimization algorithms. In the first proposed method, different optimization algorithms and continuous action-set learning automata (CALA) were used for finding the weights of BEL model, while in the second proposed model, the weights obtained using original rules of BEL model. In fact, in the second model finite action-set learning automata, CALA and different optimization algorithms were used for calibrating the learning parameters of the model. Also in the both proposed methods after extracting frequency features in thalamus, deep belief network is used in the sensory cortex for reducing the size of features. In addition, ANFIS is used for making fuzzy rules in the amygdala. The proposed models were used for magnitude and consequently fear prediction of the earthquakes. The results show that although both proposed methods are more accurate than original BEL model and could be used successfully, the second proposed model is more precise and reliable than the first proposed model.

Keywords Brain emotional learning · Learning automata · Adaptive neuro-fuzzy inference system · Deep belief network · Fear Prediction

1 Introduction

Artificial neural networks are developed based on the analogy of human's brain for making decision or learning different activities. Recently, a new model is under consideration, which is inspired by the emotional process of brain's limbic system of mammalian. The brain emotional learning model has the advantages of lower computational complexity, high speed of convergence, and its stability. Moren and Balkenius proposed one of the famous

computational models of brain emotional learning (BEL) [1, 2]. This model or modified version of it was used in different engineering applications [3–10]. Fakhroosavy et al. proposed an intelligent method for generating artificial earthquake records based on hybrid PSO-parallel brain emotional learning inspired model [10]. In spite of high ability of ANNs to simulate complex nonlinear problems, appropriate selection of learning parameters and initial weights have an important role on the training convergence of the model. Suitable selection of these parameters could lead the model to more results that are accurate and making it more reliable model. Different modified version of BEL proposed by researchers and used in real applications. Lotfi and akbarzadeh proposed the brain emotional learning-based pattern recognizer by extending the computational model of human brain limbic system. The authors used the proposed model for solving classification and chaotic time series prediction problems. The proposed model has advantages of more accuracy, less complexity, and more speed of training in comparison with standard MLP [11]. Parsapoor combined emotionally inspired structure with neuro-fuzzy inference

✉ Saeed Setayeshi
setayesh@aut.ac.ir

Seyyedeh Hoorah Fakhroosavy
h.fakhroosavy@srbiau.ac.ir

Arash Sharifi
a.sharifi@srbiau.ac.ir

¹ Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

² Department of Medical Radiation Engineering, Amirkabir University of Technology, Tehran, Iran

system to propose a new model for solar activity prediction. The author found that the predictor model has a faster convergence than ANFIS and is reliable model for predicting the solar activity and other similar prediction problems [12]. Lotfi used the model of emotional process in the brain and proposed an image classifier model named brain emotional learning-based picture classifier. The author applied the activation function in the brain emotional model to improve the efficiency of proposed model. The results of simulation show the high speed of proposed model in comparison with multilayer perceptron neural network in training [13]. Parsapoor merged the model of brain emotional learning-based fuzzy inference system with radial basis function network and tested it by complex systems [14]. Lotfi and Keshavarz introduced the fuzzy mathematical model of brain system and used it for predicting the chaotic activity of earth's magnetosphere. The authors fuzzified the connections in the limbic system model and implemented the inhibitory task of orbitofrontal cortex as a fuzzy decision making layer. The simulation showed the higher correlation of obtained results of fuzzy model in comparison with non-fuzzy models [15]. Lucas and Moghimi used a modified version of brain emotional learning model for designing an intelligent controller for auto landing system of aircrafts. As the existing automotive landing systems are activated only in a well-specified wind speed lamination condition, the proposed controller was more robust and had a better performance in the condition of strong wind gusts [16]. Parsapoor and Bilstrup proposed a new model of brain emotional learning-based prediction model by assigning adaptive networks to the different parts of original brain emotional learning model. The author used proposed model for predicting geomagnetic storms using the disturbance storm time index [12].

In this study, we combined optimization algorithms and learning automata with original model of BEL to propose modified new models to increase the accuracy and performance. This article is organized as follows: the concepts of brain emotional learning inspired model are presented in Sect. 2. Learning automata is described in Sect. 3. The optimization methods, which are used in this research, are expressed in Sect. 4. Power spectral density function as the frequency feature of earthquake record is presented in Sect. 5. Adaptive neuro-fuzzy inference system is explained in Sect. 6. Deep belief network for size reduction of features is described in Sect. 7. Two methods for improving the efficiency of original BEL model are proposed in Sect. 8. Numerical examples are illustrated in Sect. 9. Finally, concluding and remarks are presented in Sect. 10.

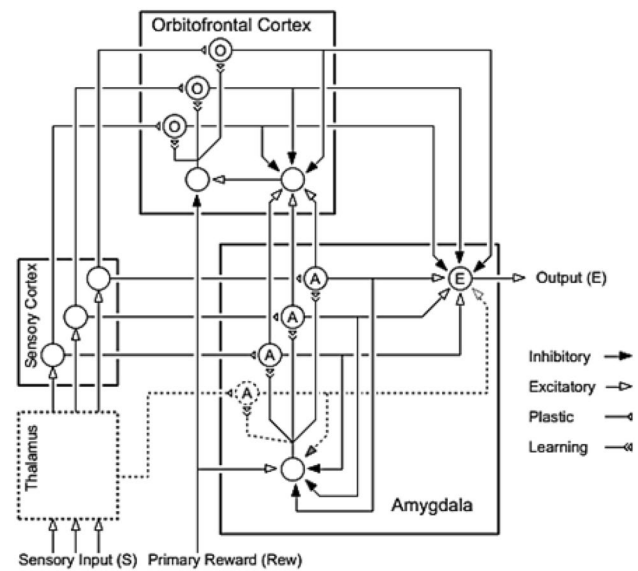


Fig. 1 Different parts of limbic system in the BEL model [2]

2 Brain emotional learning inspired model (BELIM)

The emotional part of mammalian's brain has an important role of making a rapid response to the environmental stimuli. This part, which is called limbic system, includes different sections such as Thalamus, Sensory Cortex, Amygdala, and Orbitofrontal Cortex. Each section has a role in analyzing the input stimuli and making emotional response. The thalamus receives input information from environment. Pre-processing of signals is done in this section. The amygdala receives information of initial signals from thalamus that has not yet entered to the sensory cortex. Information of the other parts of the limbic system is also entered to the amygdala. Sensory cortex acts as a transition area in which signals through it from the rest of the cerebral cortex transmitted to the limbic system and vice versa. Therefore, it works as a communication area of the brain for controlling the behaviour. The orbitofrontal cortex (OFC) receives information from the sensory cortex. It has an important role in decision-making by its cognitive processors. In this area, learning is done by processing the stimuli and applying positive and negative reinforcement in the sense of reward and penalty. The other duty of OFC is controlling amygdala's irrelevant responses.

Many researchers recently studied and used brain emotional learning inspired model to solve complex and non-linear problems of mapping and making decision in the engineering science and industrial applications [4–10, 14]. Moren and Balkenius proposed one of the most important computational models in the field of emotional learning. This model shows the relationship between different parts

of limbic system as shown in Fig. 1 with emphasize on interaction between amygdala and orbitofrontal cortex [2].

The sensory inputs enter to the thalamus, which is a pathway for sending the input stimuli to sensory cortex and amygdala. In fact, the noise reduction and pre-processing will be done in the thalamus. In the case of having n inputs, they will be pre-processed in the thalamus. Then, it sends n normalized stimuli to the sensory cortex and one stimulus to the amygdala, which are as follows [2]:

$$S_{n+1} = \max(S_i); \quad i = 1, 2, \dots, n. \tag{1}$$

Sensory cortex is a transition area which transmits the sensory inputs to both amygdala and orbitofrontal cortex. Amygdala, which has the role of final decision-making, contains $n + 1$ A -nodes. For each A -node, there is a connection weight V_i . The output of each node is made by multiplying inputs S_i to weights V_i :

$$A_i = S_i V_i; \quad i = 1, 2, 3, \dots, n + 1. \tag{2.2}$$

The output of amygdala is obtained by the summation of weighted inputs [2]:

$$E_a = \sum_{i=1}^{n+1} A_i. \tag{2.3}$$

The connection weights V_i are modified monotonically based on the difference between the reinforcer R and summation of A -nodes:

$$\Delta V_i = \alpha S_i \left[R - \sum_{j=1}^{n+1} A_j \right]^+, \tag{2.4}$$

where α is the learning rate in the interval of [0,1]. If emotional reaction is learned by amygdala, it should be permanent; thus, the modification of weights V_i is monotonically based on Eq. (2.4). There is another important part in the limbic system, called orbitofrontal cortex, which controls amygdala’s irrelevant responses. There is no -node in the orbitofrontal cortex, which is obtained by multiplying sensory inputs S_i by connection weights W_i [2]:

$$O_i = S_i W_i. \tag{2.5}$$

Summation of O_i makes the output of orbitofrontal cortex:

$$E_o = \sum_{i=1}^n O_i. \tag{2.6}$$

Connection weights of W_i are updated as a function of the input and internal reinforcer for the orbitofrontal cortex:

$$\Delta W_i = \beta S_i R_0, \tag{2.7}$$

where β is a learning rate parameter and internal reinforcer R_0 could be calculated by the following equation:

$$R_0 = \begin{cases} \left[\sum_i A_i - R \right]^+ - \sum_i O_i & \text{if } R \neq 0 \\ \left[\sum_i A_i - \sum_i O_i \right]^+ & \text{otherwise} \end{cases} \tag{2.8}$$

The final output of limbic system will be calculated by the following equation [2]:

$$E = E_a - E_o. \tag{2.9}$$

3 Learning automata

A stochastic learning automaton is a useful tool for making decision under uncertain conditions. It has been used in many engineering problems with a nondeterministic nature. Learning automata is an iterative process of selecting an action randomly, based on a probability density function and applying the action to the environment. The environment responds to the action and this response change the probability vector for selecting the next action. This process will be repeated to find the best action for finding an optimum or goal response of environment. Learning automata is divided to two main categories, finite action-set learning automata and continuous action-set learning automata, which are described in the following [17–19].

3.1 Finite action-set learning automata (FALA)

The action-set in FALA is always considered to be finite and predefined. Let $A = \{ \alpha_1, \dots, \alpha_r \}$, $r < \infty$ be the set of actions available at each instant n , the automaton selects an action $\alpha(n)$ randomly based on its probability distribution, $p(n) = \{ p_1(n), \dots, p_r(n) \}$. The selected action is applied to the environment. Then, the environment responds to this action by stochastic reinforcement signal, $\beta(n)$ as shown in Fig. 2. Afterward, the LA updates the probability distribution $p(n)$ based on the selected action and reinforcement signal. The process of updating is done using different learning algorithms. Two kinds of fixed and variable structure FALA exist.

Examples of the fixed structure LA type are Tsetline, krinsky, and Krylov automata. Linear reward-inaction L_{R-I} ,

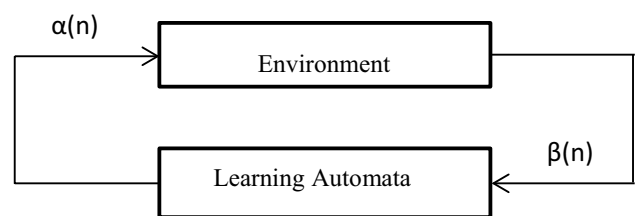


Fig. 2 Structure of learning automata

linear reward- ε -penalty $L_{R-\varepsilon p}$, linear reward-penalty L_{R-p} , and pursuit algorithm are examples of variable structure FALA. Variable structure FALA is used in this research. In the following, the algorithms, which are used in this study, will be described briefly [18, 19].

3.1.1 Linear reward-inaction algorithm, L_{R-I}

The L_{R-I} algorithm updates the action probabilities as described below. Let $\alpha(n) = \alpha_i$, then the action probability vector $p(n)$ is updated as follows:

$$\begin{aligned} p_i(n+1) &= p_i(n) + \lambda \beta(n)(1 - p_i(n)) \\ p_j(n+1) &= p_j(n) - \lambda \beta(n)p_j(n); \quad j \neq i, \end{aligned} \quad (3.1)$$

where λ is the learning (step-size) parameter satisfying $0 < \lambda < 1$ [18].

3.1.2 Linear reward-penalty algorithm, L_{R-p}

If $\alpha(n) = \alpha_i$, then the probability vector is updated as follows:

$$\begin{aligned} p_i(n+1) &= p_i(n) + \lambda_1 \beta(n)(1 - p_i(n)) \\ &\quad - \lambda_2 (1 - \beta(n))p_i(n) \\ p_j(n+1) &= p_j(n) - \lambda_1 \beta(n)p_j(n) \\ &\quad + \lambda_2 (1 - \beta(n))\left(\frac{1}{r-1} - p_j(n)\right); \quad j \neq i, \end{aligned} \quad (3.2)$$

where λ_1 and λ_2 are learning parameters which usually $\lambda_1 = \lambda_2$ [18].

3.1.3 Pursuit Algorithm

The reward probabilities of actions are estimated in this algorithm by considering the history of selected actions and obtained reinforcement signal.

Let $\alpha(n) = \alpha_i$. The number of times which action α_i is chosen till instant n and the total reinforcement obtained in response to action α_i are saved in vectors $(\eta_1(n), \dots, \eta_r(n))^T$ and $(Z_1(n), \dots, Z_r(n))^T$, respectively. These vectors updates as follows:

$$\begin{aligned} Z_i(n) &= Z_i(n-1) + \beta(n) \\ Z_j(n) &= Z_j(n-1); \quad \forall j \neq i \\ \eta_i(n) &= \eta_i(n-1) + 1 \\ \eta_j(n) &= \eta_j(n-1); \quad \forall j \neq i \\ \hat{d}_i(n) &= \frac{Z_i(n)}{\eta_i(n)}; \quad i = 1, \dots, r, \end{aligned} \quad (3.3)$$

where \hat{d} is the estimator vector, which is used for updating the probability of actions. Let $\hat{d}_{M(n)}$ be the highest estimated reward probability at instant. If the estimates are true, the value of $p_m(n)$ should be one and the rest of action probabilities should be zero. In other words, $p(n) = e_{M(n)}$, where $e_{M(n)}$ is a vector, which its m th element is one and the other elements are zero. This algorithm updates probability of actions by moving $p(n)$ towards $e_{M(n)}$ by a small amount determined by a learning parameter as follows:

$$p(n+1) = p(n) + \lambda(e_{M(n)} - p(n)), \quad (3.4)$$

where $0 < \lambda \leq 1$ is the learning parameter and the index $M(n)$ is determined by the following [18, 19]:

$$\hat{d}_{M(n)} = \max \hat{d}_i(n). \quad (3.5)$$

3.2 Continuous action-set learning automata (CALA)

In the FALA, the actions are a finite set with predefined values. These actions could not be more suitable for finding optimal parameter values to maximize a performance index. In fact, in this case, a continuous action set is needed. The automaton, which uses continuous action set, is called CALA. In this algorithm, the probability distribution of actions at instant n is $N(\mu(n), \sigma(n))$, which is the normal distribution with mean $\mu(n)$ and standard deviation $\sigma(n)$. By updating $\mu(n)$ and $\sigma(n)$ in each instant, the CALA updates the probability distribution of actions. Let $\alpha(n) \in \mathbb{R}$ be the action chosen and let $\beta(n)$ be the reinforcement signal at instant n . A reward function $f: \mathbb{R} \rightarrow \mathbb{R}$ instead of reward probabilities is defined by $f(x) = E[\beta(n) | \alpha(n) = x]$. The reinforcement in response to action x denoted by β_x :

$$f(x) = E\beta_x. \quad (3.6)$$

The role of CALA is finding the value of x to maximize the $f(x)$. In this case, the $N(\mu(n), \sigma(n))$ converges to $N(x_0, 0)$, where the reward function has its maximum value at x_0 . To avoid of being stuck at a non-optimal point the CALA, lets $\sigma(n)$ converge to σ_l instead of zero, which σ_l has a very small value. CALA interact with the environment by choosing of two actions $x(n)$ and $\mu(n)$ at each instant. The value of $x(n)$ is generated randomly using probability distribution of $N(\mu(n), \phi(\sigma(n)))$. Two actions $x(n)$ and $\mu(n)$ apply to the environment and CALA updates the probability distribution by updating mean and standard deviation of actions as follows [19]:

$$\begin{aligned} \mu(n+1) &= \mu(n) + \lambda \frac{(\beta_x - \beta_\mu)}{\phi(\sigma(n))} \frac{(x(n) - \mu(n))}{\phi(\sigma(n))} \\ \sigma(n+1) &= \sigma(n) + \lambda \frac{(\beta_x - \beta_\mu)}{\phi(\sigma(n))} \left[\left(\frac{(x(n) - \mu(n))}{\phi(\sigma(n))} \right)^2 - 1 \right] \\ &\quad + \lambda \{C[\sigma_l - \sigma(n)]\}, \end{aligned} \tag{3.7}$$

where

$$\phi(\sigma) = \begin{cases} \sigma_l; & \sigma \leq \sigma_l \\ \sigma; & \sigma > \sigma_l \end{cases}$$

and, λ : learning parameter ($0 < \lambda \leq 1$), c : large positive constant.

4 Used optimization methods

4.1 Genetic algorithm

Genetic algorithm is an evolutionary computing method for finding the optimum value of a simple to very complex and nonlinear function. This algorithm was inspired by the biological rules in the nature and has been used in many optimization problems especially in the engineering fields. The GA consists of three main operators, selection, crossover, and mutation [20–22]. Figure 3 shows the steps of GA and its application on this research. GA used two times in this study: first time, for finding the optimum value of BEL’s weights in one optimization problem and the second time for finding the best values of learning rate parameters.

4.2 Particle swarm optimization (PSO)

PSO algorithm was inspired by the social behaviour of animals, such as bird flocking or fish schooling. Each individual in this algorithm is called particle. Any particle moves to new position based on its past position and current velocity. Each particle updates its velocity based on the information which it gets by own and from other particles in the swarm [23, 24]. This algorithm was used in many areas of engineering fields [25–32]. PSO algorithm is used in this research for finding the optimum value of BEL weights or learning rate parameters as follows.

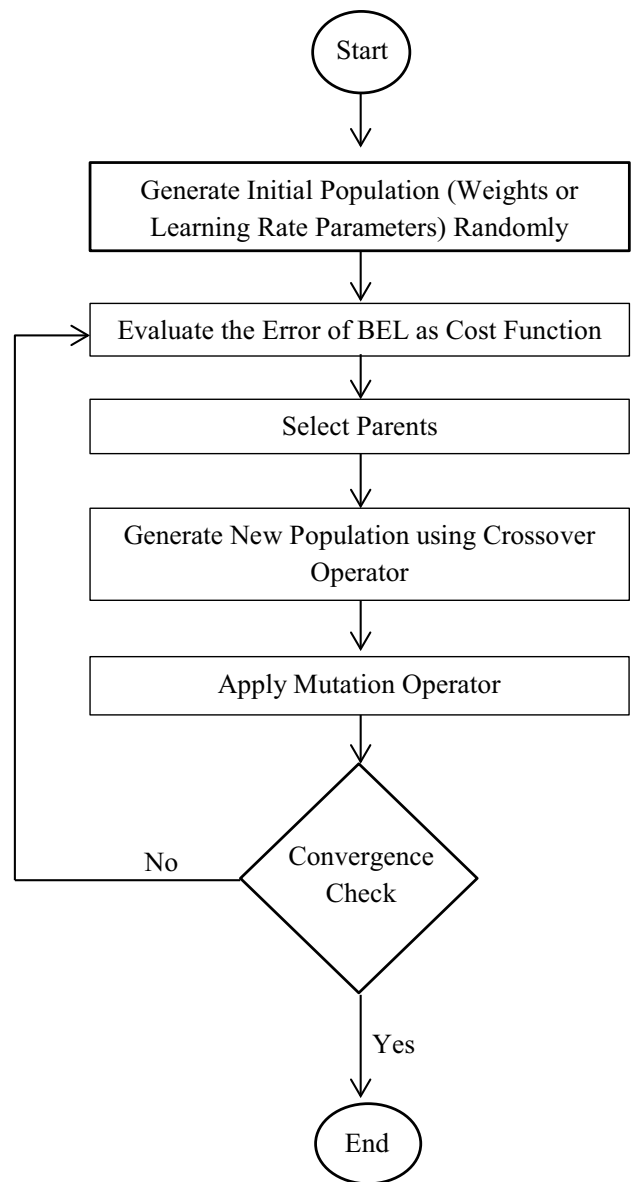


Fig. 3 Flowchart of GA for finding the weights or learning parameters of model

PSO-Based Learning Algorithm

- 1) Initiate particles randomly (weights or learning rate parameters)
- 2) Initiate particles velocity randomly
- 3) Train BEL for each particle
- 4) Evaluate BEL error as fitness function
- 5) Save the personal best position
- 6) Save the global best position
- 7) Update the particle's velocity

$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + c_1 r_1 (\vec{x}_{pbest_i} - \vec{x}_i(t)) + c_2 r_2 (\vec{x}_{gbest_i} - \vec{x}_i(t))$$

- 8) Update the particle's position

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$$

- 9) If the step criteria is not satisfied goto step 3 else goto 10
 - 10) END
-

where

\vec{x} : current position of particles

\vec{x}_{pbest} : best personal position of particles

\vec{x}_{gbest} : best global position of flock

ω : inertia weight which shows the contribution of current velocity in updating the particle's velocity

r_1, r_2 : random values in the range of [0,1]

c_1, c_2 : positive acceleration constants

4.3 Artificial bee colony (ABC)

ABC was inspired by the behaviour of honeybees in the nature for finding food sources. In this algorithm, there are three groups of honeybees, which are named employee, onlooker, and scout bees. Scout bees find the food sources, which are the solution for each parameter randomly, and

then, employee bees go to the food sources. They evaluate the amount of food source nectar (fitness function). Afterwards, the onlooker bees select the best food source after evaluating the information, which obtained from employee bees [33]. ABC algorithm is used in this research for finding the optimum value of BEL weights or learning rate parameters as follows.

ABC-based Learning Algorithm

- 1) Initialize the population of solution x_{ij} (weights or learning rate parameters)
- 2) Evaluate the population (calculate BEL error as fitness function)
- 3) Create new solution v_{ij} in the neighbourhood of x_{ij} for the employee bees and calculate the fitness function at new food sources.

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj})$$
- 4) Compare x_{ij} with v_{ij} and if v_{ij} has better fitness function then replace x_{ij} by v_{ij} otherwise keep the x_{ij}
- 5) Calculate the probability value p_i for the solution x_i by evaluating their fitness function and using the following equation

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}$$

- 6) Create the new position v_i for the onlooker for the solution x_i , selected depending on p_i and evaluate them
- 7) Apply the greedy selection process for the onlookers between x_i and v_i
- 8) Determine the abandoned solution, if exist, and replace it with a new randomly produced solution x_i for the scout

$$x_{ij} = min_j + rand(0,1) \times (max_j - min_j)$$

- 9) Keep the best food source position
 - 10) If step criteria is not satisfied go to 3, else go to 11
 - 11) END
-

where

Φ_{ij} : Random value in [0,1]

5 Power spectral density function (PSDF)

If $x(t)$ is a random variable, its time-history is not periodic, in general. Thus, it is not possible to show it by its discrete Fourier series. In addition, for a stationary process $x(t)$, the condition below may not be satisfied:

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty. \tag{5.1}$$

Therefore, it is not possible to calculate the Fourier series of $x(t)$. It may be overcome to this problem using autocorrelation function $R_x(\tau)$ instead of $x(t)$. Autocorrelation function contains the frequency content of process indirectly and defined as the mean value of the product $x(t)x(t + \tau)$, which τ is a time difference between sampling. If $x(t)$ is a stationary process, the value of $E[x(t)x(t + \tau)]$ is independent of time:

$$E[x(t)x(t + \tau)] = f(\tau) = R_x(\tau), \tag{5.2}$$

where $E[.]$ is the expected value.

Power spectral density function, $S_x(\omega)$, is the Fourier transform of autocorrelation function of x [34]:

$$S_x(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R_x(\tau)e^{-i\omega\tau} d\tau. \tag{5.3}$$

6 Adaptive neuro-fuzzy inference system (ANFIS)

- A fuzzy inference system with fuzzy if–then rules can model the qualitative aspects of human knowledge and reasoning processes without quantitative analysis. ANFIS acts in this way. Suppose that there is a system with two inputs x, y and one output z . The rule base contains two fuzzy if–then rules of Takagi and Sugeno type [35].

Rule 1: if x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$,

Rule 2: if x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$

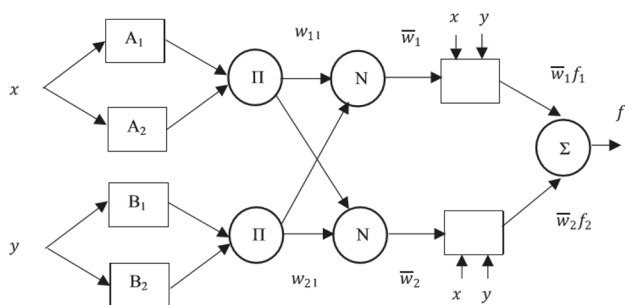


Fig. 4 Structure of the ANFIS with two inputs of x and y

- A_i, B_i and F_i are fuzzy sets and p_i, q_i and r_i are the outputs of the system which obtains from the learning process. Figure 4 shows the type-3 ANFIS.

- Layer 1 (fuzzification): in this layer, the membership grades are generated by Eqs. (6.1), (6.2), and (6.3) [35]:

$$O_i^1 = \mu A_i(x), \tag{6.1}$$

- where O_i^1 is the membership function of A_i . $\mu A_i(x)$ is bell shaped with maximum of 1 and minimum of 0 such as

$$\mu A_i(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^2 \right]^{b_i}} \tag{6.2}$$

- Or

$$\mu A_i(x) = \exp \left\{ - \left(\frac{x - c_i}{a_i} \right)^2 \right\}, \tag{6.3}$$

- which $\{a_i, b_i, c_i\}$ is the parameter set.
- Layer 2 (production) : Each node in this layer is a circle node and multiplies the incoming signals. Each node output represents the firing strength of a rule [35]:

$$w_i = \mu A_i(x) \times \mu B_i(y) ; i = 1, 2. \tag{6.4}$$

- Layer 3 (normalization): The i th node calculates the ratio of the i th rule’s firing strength to the sum of all rules’ firing strengths [35]:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2. \tag{6.5}$$

- Layer 4 (defuzzification): Every node i in this layer is a square node with a function [35]:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \tag{6.6}$$

- where \bar{w}_i : Output of layer 3
- $\{p_i, q_i, r_i\}$: Parameter set.
- Layer 5 (output): The summation of all incoming signals is computed by the following [35]:

$$O_1^5 = \text{overall output} = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{6.7}$$

7 Deep belief network (DBN)

- Artificial neural network is one of the most important tools in the field of artificial intelligence. It has many applications such as object recognition, mapping, signal analysis, and so on. According to the theoretical and biological reasons, it is recommended to use deep architecture including many nonlinear processing layers in the structure of ANNs. These deep models have many hidden layers and parameters, which should be trained. It increases the computational effort and decreases the speed of training significantly. In addition, it may cause to get into local minimum. Using DBN could overcome

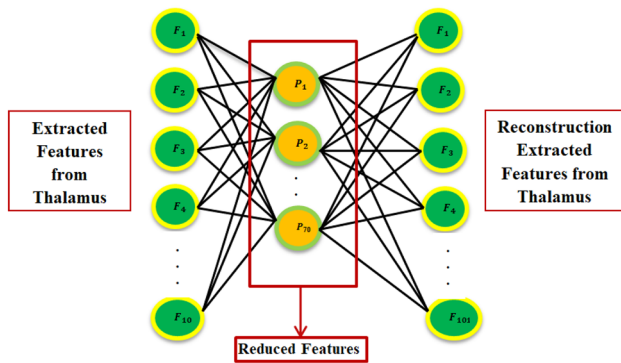


Fig. 5 Deep belief network for size reduction of features

these problems [36, 37]. The layers of DBN consist of Restricted Boltzmann Machines (RBM) which are probabilistic models with one hidden layer. The DBN tries to reconstruct the inputs at the output layer. For this purpose, the hidden layer should be described the data of input layer, as well. Not only DBN is useable for classification, but also it is capable of extracting feature. In fact, DBN is able to extract the most important features of training data [38]. In addition, DBN could be used to reduce the dimensionality of data [38]. After extracting the features of earthquake records (PSDF) in the thalamus, the DBN is used in the sensory cortex to reduce the size of features. Therefore, the most effective features will be found. For this purpose, the number of neurons in the hidden layer should be less than the number of neurons in the input layer. If the reduced-dimension data could explain the input data, it could be used as representative of whole features. In this research, 101 features obtain for each earthquake record and considered as the inputs and outputs of DBN, as shown in Fig. 5. The number of hidden neurons is chosen 70, which are less than input and output neurons. The network will be trained. Therefore, the features in the hidden layer could be used as inputs of amygdala and orbitofrontal cortex.

8 Proposed method

- Two methods are proposed in this research for predicting the level of fear encounter to an earthquake. In both methods, a modified BEL model is used. The input signal of the model is earthquake record, which shows the

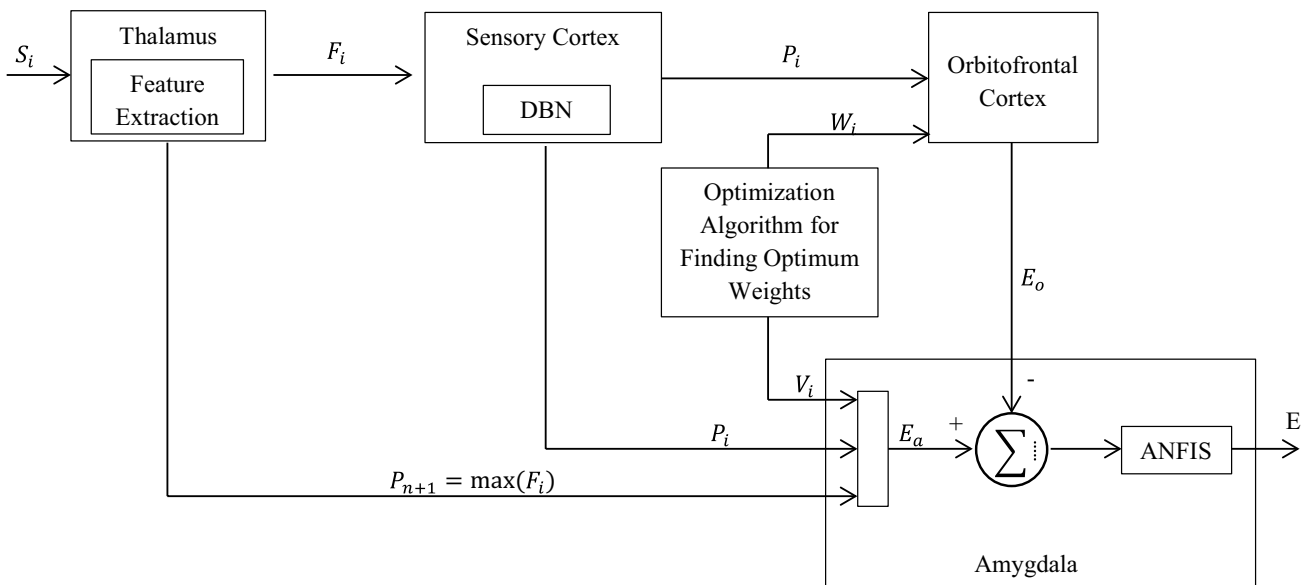


Fig. 6 Structure of First Proposed Model

ground motion accelerations. As the ground accelerations are recorded in a short time steps (for example 0.02 s), there are many data points in each earthquake record, which increase the calculations of model. In addition, two earthquakes, which are similar in magnitude and intensity, could be completely different in acceleration–time diagram. Therefore, the feature extraction is considered in the thalamus to use a more relevant stimulus for sensory cortex. Frequency content is the most important feature of earthquake signals. In this research, power spectral density function is used as the frequency content of earthquake records for sending to sensory cortex. In this way, the calculations will be reduced significantly. Furthermore, more sensible stimulus will be used for predicting the fear induced by earthquake in the brain emotional model. In sensory cortex, we used a deep belief network for decreasing the size of PSDF. Afterwards, it sends the reduced features to orbitofrontal cortex and amygdala. It should be mentioned that the output of the model is the level of fear. In fact, the level of fear is a qualitative value, which could not be predicted directly using brain emotional learning (BEL) model. To make

a relationship between the values, which calculate at amygdala and the level of fear, an adaptive neuro-fuzzy inference system, is used in the amygdala to make fuzzy rules. These fuzzy rules are used in the amygdala to find the level of fear caused by the earthquake.

8.1 First proposed method

- In the first proposed method, the structure of BEL model is used without considering the original learning algorithm of BEL model. In this method, the weights are found and modified using different optimization algorithms such as GA, PSO, ABC, and CALA. In fact, this is an optimization problem, which its design variables are weights of modified BEL model and the goal is minimizing the error of model to find the fear. Different optimization algorithms are considered in the first proposed methods to find the most efficient algorithm for the kind of input–output data at hand. Figure Fig. 6 shows the structure of the first proposed method.

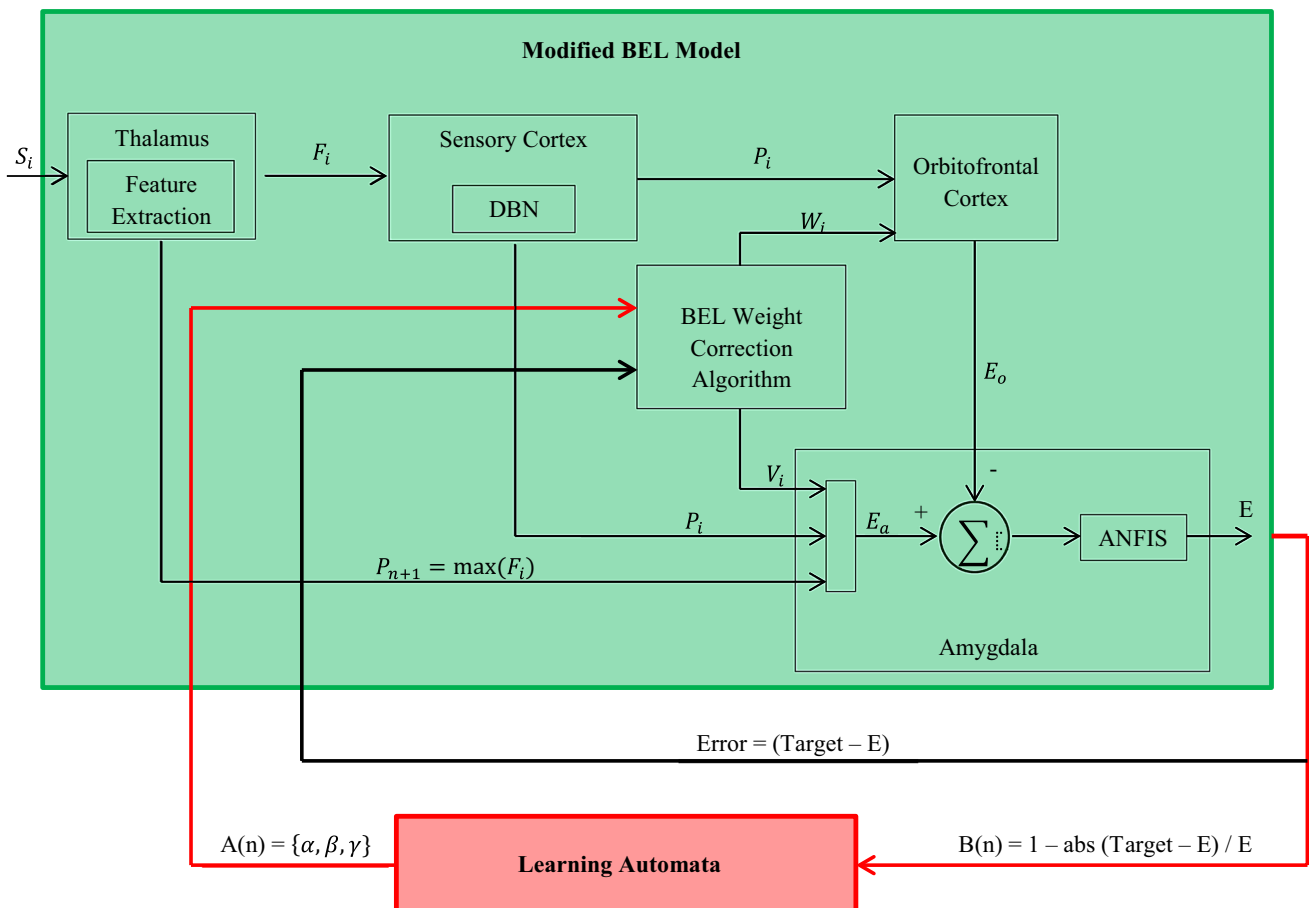
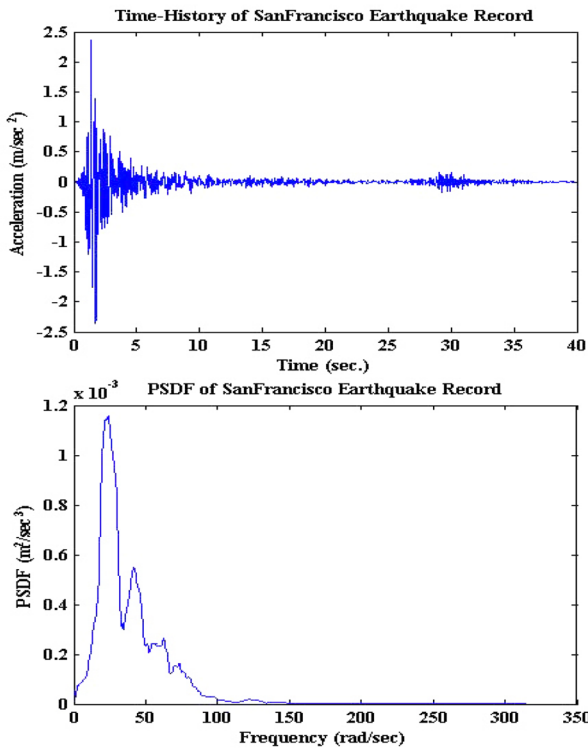
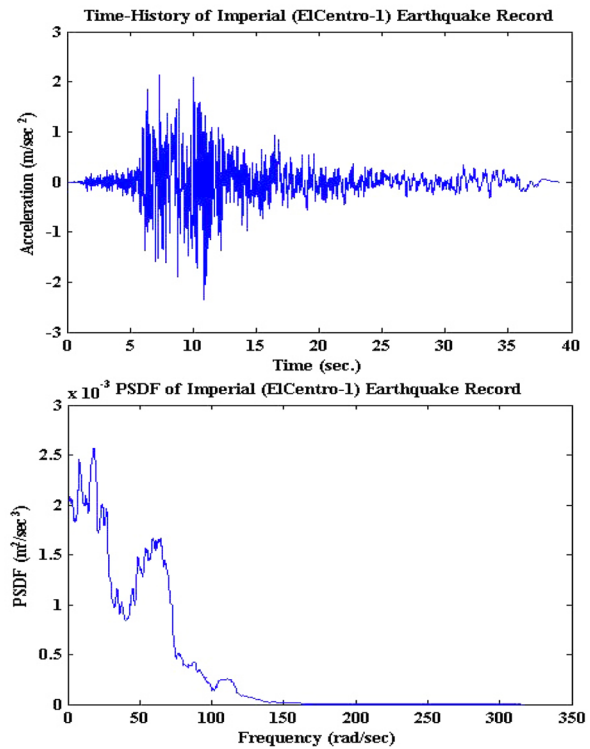


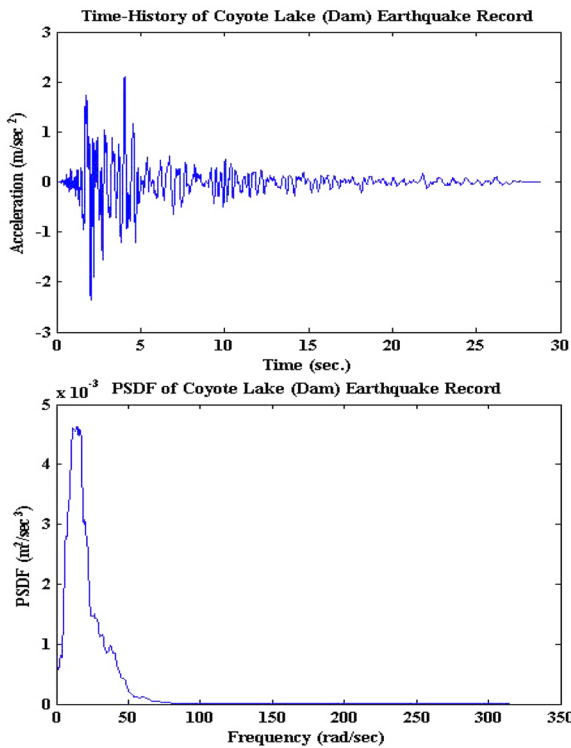
Fig. 7 Structure of Second Proposed Model



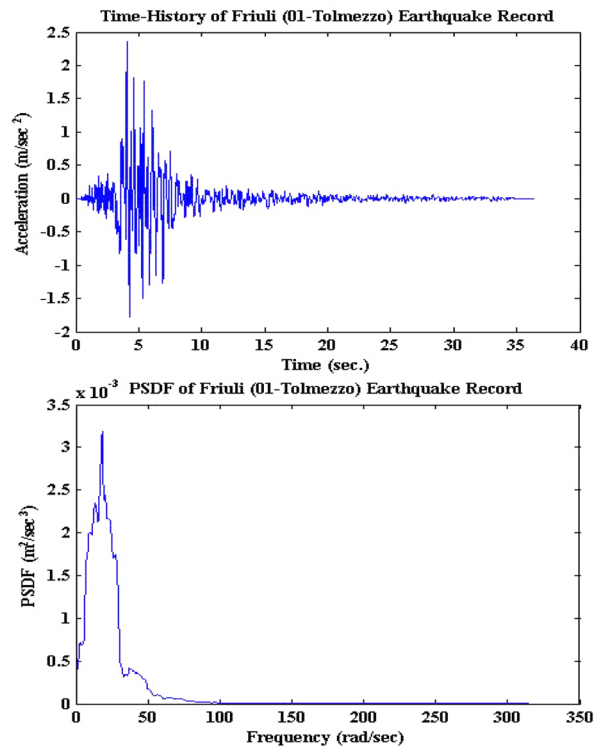
(a) San Francisco 1957 Earthquake Record



(b) Imperial (El Centro) 1979 Earthquake Record



(c) Coyote Lake (Dam) 1979 Earthquake Record



(d) Friuli (Tolmezzo) 1976 Earthquake Record

Fig. 8 PSDF of four used samples as the input of models

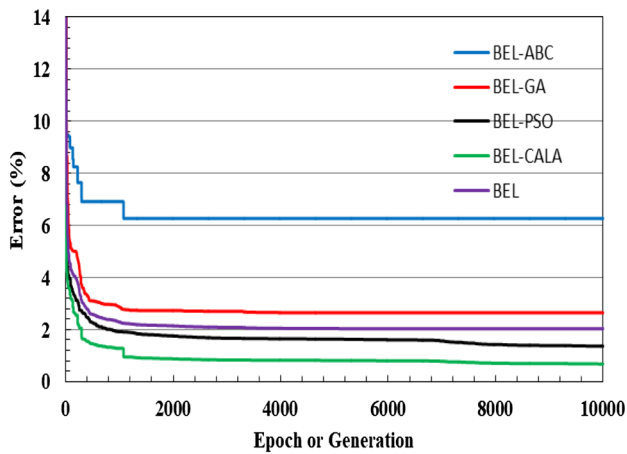
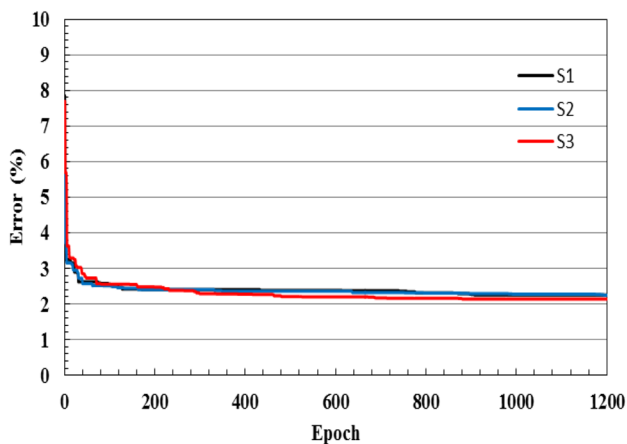


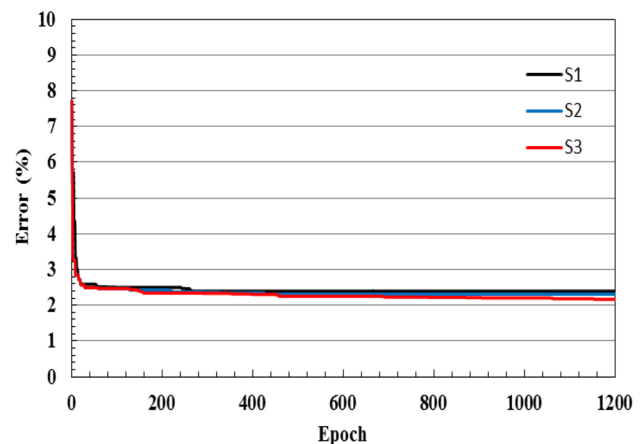
Fig. 9 Results of training first proposed model with different optimization algorithms

8.2 Second proposed method

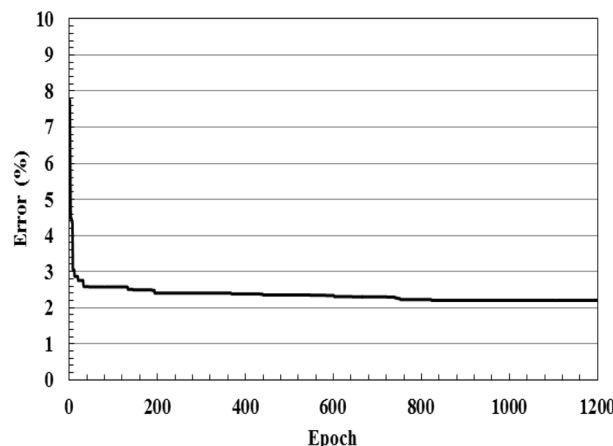
- In the second proposed method, learning automata is used for finding the learning parameters of modified BEL model (MBEL). In fact, MBEL model is used as the environment. The difference between the output of amygdala and target value shows the absolute error of the model. The relative error could be calculated and the response of environment will be defined as unity minus relative error. In the BEL model, the output and the target are scaled to be in the interval of [0, 1]. Therefore, the relative error could be in the interval of [0, 1], as well. If action selects properly in an ideal case, the relative error will be zero and the response of environment will be one, which shows a reward. In the other case: if an improper action selects, the relative error in the worst case will have its maximum value of one and the response of environment will be zero



(a) Training Model by using L_{R-I} Algorithm



(b) Training Model by using L_{R-P} Algorithm



(c) Training Model by using L_{P-A} Algorithm

Fig. 10 Training second proposed model with different learning algorithm of LA

Table 1 Results of the first proposed model for ten samples of test data

Event	Real Mag.	BEL		BEL-GA		BEL-ABC		BEL-PSO		BEL-CALA	
		Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)
Figueroa Mountain	2.99	3.15	5.35	3.27	9.22	3.27	9.45	3.23	7.93	3.13	4.68
Angel Island	3.70	3.46	6.49	3.35	9.46	3.64	1.73	3.70	0.00	3.72	0.54
Ancona	4.00	4.09	2.25	4.02	0.55	3.92	2.12	4.00	0.00	3.98	0.50
Central Calif	5.00	5.48	9.60	5.66	13.20	4.78	4.43	5.07	1.31	4.95	1.00
Managua Nicaragua	5.20	5.24	0.77	5.29	1.78	5.21	0.13	5.02	3.53	5.31	2.12
Coyote Lake	5.74	5.70	0.70	5.67	1.10	5.02	12.63	5.82	1.31	5.67	1.22
Santa Barbara	5.91	5.82	1.52	5.74	2.84	6.36	7.61	6.03	2.04	5.96	0.85
Northern Calif	6.50	6.49	0.15	6.56	0.97	7.24	11.32	6.48	0.35	6.51	0.15
Irpinia Italy	6.90	6.97	1.01	6.90	0.00	6.73	2.40	7.04	2.03	6.80	1.45
Gulf of Aqaba	7.20	7.17	0.42	7.20	0.00	5.74	20.31	7.20	0.00	7.20	0.00
NRMSE	–	0.1378		0.1950		0.4474		0.0872		0.0542	
NMSE	–	0.0190		0.0380		0.2001		0.0076		0.0029	
Average Error (%)	–	2.83		3.91		7.21		1.85		1.25	

Table 2 Relationship between earthquake magnitude and assigned fear level (USGS)

Magnitude range	Fear level
$0 < M \leq 2.2$	Very low
$2.20 < M \leq 3.45$	Low
$3.45 < M \leq 4.70$	Moderate
$4.70 < M \leq 5.90$	High
$5.90 < M \leq 7.20$	Very high
$7.20 < M \leq 8.00$	Extreme

- Selection of actions is done in three different methods:
- selecting actions randomly (S_1)
- selecting actions randomly with equal number of all actions (S_2)
- selecting actions based on their probability (S_3)

Figure Fig. 7 shows the second proposed method in detail.

which shows a penalty. The response applies to LA for making decision of finding the best action, which are the learning parameters of network. These parameters are used in the weight correction algorithm of MBEL model. Three learning algorithm, L_{R-I} , L_{R-P} , L_{P-A} , are used in this research to find suitable algorithm for the data which are used in this study. These algorithms described in Sect. 3.

9 Numerical examples

9.1 Data set

A data set consists of 100 real earthquake records, which is provided by the Pacific Earthquake Engineering Research Center (PEER) Website [39]. 70% of the data set are used for training the proposed model and the rest of data are used

Table 3 Comparison between assigned and predicted levels of fear

Event	Fear	BEL Pred. fear	BEL-GA Pred. fear	BEL-ABC Pred. fear	BEL-PSO Pred. fear	BEL-CALA Pred. fear
Figueroa Mountain	Low	Low	Low	Low	Low	Low
Angel Island	Moderate	Moderate	Low	Moderate	Moderate	Moderate
Ancona	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
Central Calif	High	High	High	High	High	High
Managua Nicaragua	High	High	High	High	High	High
Coyote Lake	High	High	High	High	High	High
Santa Barbara	Very High	High	High	Very High	Very High	Very High
Northern Calif	Very High	Very High	Very High	Extreme	Very High	Very High
Irpinia Italy	Very High	Very High	Very High	Very High	Very High	Very High
Gulf of Aqaba	Very High	Very High	Very High	High	Very High	Very High

The bolditalic values are wrong values and these values are different with desired values

for testing the performance of trained model. Earthquakes are different in time duration, frequency content, magnitude, and peak ground acceleration. These records, which show the acceleration time-history of earthquakes, are used as the input of proposed models.

9.2 Example for the first proposed model

The earthquakes select randomly from the data set for applying to the model as the input. Feature extraction is done in the thalamus and PSDF is obtained for each earthquake record. In fact, PSDF shows the frequency content of earthquake which is one of the most important features of each earthquake record. Figure 8 shows four samples of PSDF which are used in this research.

The features entered into sensory cortex. DBN is used in this section for reducing the size of features. The training of the model is done using all specified training data. Figure 9 shows the comparison between original BEL and using different optimization algorithms for finding the appropriate weights of the first proposed method. According to Fig. 9, BEL-CALA is the best optimization algorithm for this example even better than original BEL.

To illustrate the ability of trained model, 30% of provided data were used as test data. The average relative error for BEL, BEL-GA, BEL-ABC, BEL-PSO, and BEL-CALA is 2.97, 3.82, 7.04, 1.92, and 1.21%, respectively. Table 1 demonstrates the results of ten samples of test data for different optimization algorithm and original BEL, which are used in the first proposed method.

As expected from the training results, BEL-CALA is the best model, which uses CALA algorithm for training the model. The level of fear is related to the amount of magnitude of the earthquakes. Table 2 illustrates the relationship between the earthquake magnitude and the level of fear, which is considered in this research.

Although there are small differences between the real and predicted magnitude using the PSO and CALA optimization algorithms in the first proposed model, the level of fear predicted successfully by these algorithms, as shown in Table 3.

Table 3 shows that the BEL-PSO and BEL-CALA models could exactly predict the level of fear and are reliable models for this purpose.

9.3 Example for the second proposed model

In this example, the second proposed method is used for training the model. In fact, the weights are corrected using BEL Formula and learning automata is used for finding the best values of learning parameters. Different learning algorithms are used for this purpose, which their results are shown in Fig. 10.

Table 4 Results of the second proposed model on predicting the magnitude of ten samples from test data

Event	Real Mag.	BEL		BEL-GA		BEL-ABC		BEL-PSO		BEL-L_RI		BEL-CALA	
		Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)	Pred. Mag.	Error (%)
Figueroa Mountain	2.99	3.15	5.35	3.18	6.35	3.19	6.69	3.11	4.01	3.17	6.02	2.88	3.68
Angel Island	3.70	3.46	6.49	3.45	6.76	3.61	2.43	3.69	0.27	3.42	7.57	3.69	0.27
Ancona	4.00	4.09	2.25	4.01	0.25	4.05	1.25	3.98	0.50	3.97	0.75	4.01	0.25
Central Calif	5.00	5.48	9.60	5.41	8.20	4.80	4.00	4.97	0.60	5.38	7.60	4.98	0.40
Managua Nicaragua	5.20	5.24	0.77	5.26	1.15	5.18	0.38	5.30	1.92	5.28	1.54	5.12	1.54
Coyote Lake	5.74	5.70	0.70	5.79	0.87	5.17	9.93	5.77	0.52	5.81	1.22	5.78	0.70
Santa Barbara	5.91	5.82	1.52	6.02	1.86	6.23	5.41	5.98	1.18	6.07	2.71	5.93	0.34
Northern Calif	6.50	6.49	0.15	6.45	0.77	7.22	11.08	6.54	0.61	6.41	1.38	6.47	0.46
Irpinia Italy	6.90	6.97	1.01	6.88	0.29	6.77	1.88	6.97	1.01	6.93	0.43	6.84	0.87
Gulf of Aqaba	7.20	7.17	0.42	7.17	0.42	6.14	14.72	7.18	0.28	7.15	0.69	7.18	0.28
NRMSE	-	0.1378	-	0.1272	-	0.3491	-	0.0464	-	0.1306	-	0.0382	-
NMSE	-	0.0190	-	0.0162	-	0.1219	-	0.0022	-	0.0170	-	0.0015	-
Average Error (%)	-	2.83	-	2.69	-	5.78	-	1.09	-	2.99	-	0.88	-

Table 5 Results of the second model on predicting the level of fear for ten samples from test data

Event	Fear	BEL Pred. Fear	BEL-GA Pred. Fear	BEL-ABC Pred. Fear	BEL-PSO Pred. Fear	BEL-L_RI Pred. Fear	BEL-CALA Pred. Fear
Figuroa Mountain	Low	Low	Low	Low	Low	Low	Low
Angel Island	Moderate	Moderate	Low	Moderate	Moderate	Low	Moderate
Ancona	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
Central Calif	High	High	High	High	High	High	High
Managua Nicaragua	High	High	High	High	High	High	High
Coyote Lake	High	High	High	High	High	High	High
Santa Barbara	Very High	High	Very High	Very High	Very High	Very High	Very High
Northern Calif	Very High	Very High	Very High	Extreme	Very High	Very High	Very High
Irpinia Italy	Very High	Very High	Very High	Very High	Very High	Very High	Very High
Gulf of Aqaba	Very High	Very High	Very High	Very High	Very High	Very High	Very High

The bolditalic values are wrong values and these values are different with desired values

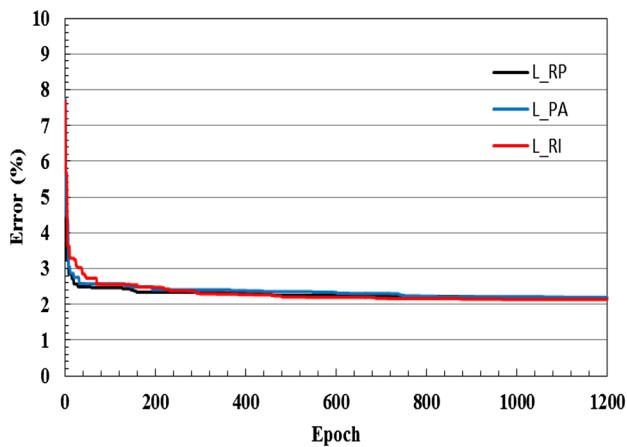


Fig. 11 Comparison between different learning algorithms of LA on training second proposed model

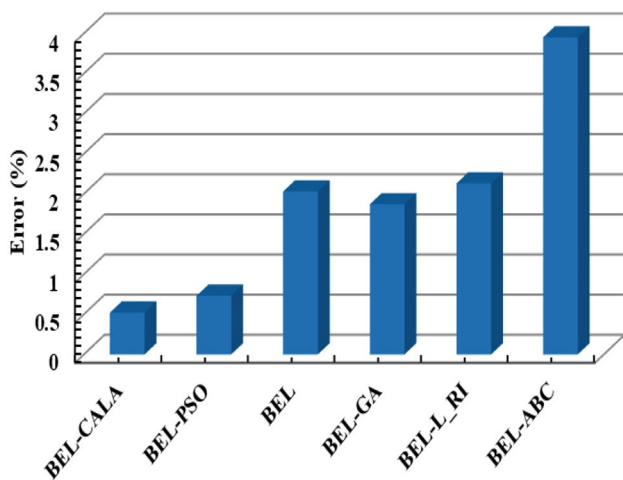


Fig. 12 Performance of different algorithms on training second proposed model

Three methods of selection are used in L_{R-I} and L_{R-P} learning algorithms, while L_{P-A} works with S_3 method only. According to Fig. 10, the third method of selection is the best choice for all learning algorithms.

Figure 11 shows the comparison between different learning algorithms. Although Fig. 11 illustrates that L_{R-P} has better training in the early epochs, but the results show that L_{R-I} has the best training in overall.

For comparing the results, other algorithms such as CALA, GA, PSO, and ABC are used for finding the learning parameters. Figure 12 shows the performance of different algorithms on training second proposed model. The result shows that BEL-CALA and BEL-PSO have the minimum error, BEL, BEL-GA, and BEL-L_RI trained with a moderate amount of error, and BEL-ABC has the maximum error.

Test data are used to show the ability of trained model for predicting the magnitude of earthquakes. The mean relative error of BEL, BEL-GA, BEL-ABC, BEL-PSO, BEL- L_{R-I} , and BEL-CALA obtained 2.97, 2.43, 5.13, 1.18, 2.63, and 0.83%, respectively. Table 4 illustrates the results of ten samples of test data. According to Table (4), the BEL-CALA model has the best performance among different algorithms. In addition, BEL-PSO could predict the magnitude with acceptable amount of error.

Table 5 shows the result of different algorithms used in the second proposed model for predicting the level of fear for ten samples of test data. In spite of small errors in predicting the magnitude, Table (5) shows high accuracy of second model to predict the level of fear.

10 Conclusions

In this research, two modified BEL models proposed to improve the efficiency and accuracy of original BEL model.

In both proposed models, ANFIS was used in amygdala to make fuzzy rules. Moreover, deep belief network was used in the sensory cortex for size reduction of features of earthquake record. GA, PSO, ABC, and learning automata were used in the first proposed model for finding the weights, and they were applied in the second proposed model to obtain learning parameters. The proposed models were used for magnitude and fear prediction of earthquakes to illustrate their performance. The following conclusions are drawn according to the results of numerical examples:

- Both proposed methods are more accurate than original BEL model.
- Except ABC, the other algorithms which were used in the first proposed model lead to improve the efficiency of original BEL model.
- All algorithms which were used in the second proposed model could improve the accuracy of model in comparison with BEL model.
- CALA algorithm in both proposed models illustrates the best accuracy among other used algorithms.

References

- Moren J (2002) Emotion and Learning—a computational model of the amygdala, PhD dissertation. Cognitive studies, Lund University, Lund, Sweden
- Moren J, Balkenius C (2000) A computational model of emotional learning in the amygdala. *Cybern Syst* 32(6):611–636
- Parsapoor M, Bilstrup U (2013) Chaotic time series prediction using brain emotional learning based recurrent fuzzy system (BELRFS). *Int J Reasoning-based Intell Syst* 5(2):113–126
- Lotfi E, Setayeshi S, Taimory S (2014) A neural basis computational model of emotional brain for online visual object recognition. *Appl Artif Intell* 28(8):814–834. <https://doi.org/10.1080/08839514.2014.952924>
- Maleki M, Nourafza N, Setayeshi S (2016) A novel approach for designing a cognitive sugarscape cellular society using an extended moren network. *Intell Autom Soft Comput* 22(2):193–201. <https://doi.org/10.1080/10798587.2015.1090720>
- Asadi Ghanbari A, Heidari E, Setayeshi S (2012) Brain emotional learning based brain computer interface. *Int J Comput Sci Issues* 9(5):146–154
- Lotfi E, Akbarzadeh-T MR (2014) Adaptive brain emotional decayed learning for online prediction of geomagnetic activity indices. *Neuro Comput* 126:188–196. doi:<https://doi.org/10.1016/j.neucom.2013.02.040>
- Lotfi E (2013) Mathematical modeling of emotional brain for classification problems. *Proceedings of IAM* 2(1):60–71
- Lucas C (2010) Introducing BELBIC: Brain emotional learning based intelligent controller. *Integrated Syst Design Technol* 3:203–214
- Fakhroosavy SH, Setayeshi S, Sharifi A (2017) An intelligent method for generating artificial earthquake records based on hybrid PSO-parallel brain emotional learning inspired model. *Eng Comput* (In Press)
- Lotfi E, Akbarzadeh -T, M. R (2013) Brain emotional learning-based pattern recognizer. *Cybernet Syst* 44(5):402–421. <https://doi.org/10.1080/01969722.2013.789652>
- Parsapoor M, Bilstrup U (2012) Brain emotional learning based fuzzy inference system (BELFIS) for solar activity forecasting. Paper presented at the Proceedings of the 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, vol 01
- Lotfi E (2013) Brain-inspired emotional learning for image classification. *Majlesi J Multimed Process* 2(3):21–26
- Parsapoor M, Bilstrup U (2013, 10–12 Sept. 2013) Brain emotional learning based fuzzy inference system (modified using radial basis function). Paper presented at the Eighth international conference on digital information management (ICDIM 2013)
- Lotfi E, Keshavarz A (2014) A simple mathematical fuzzy model of brain emotional learning to predict kp geomagnetic index. *Int J Intell Syst Appl Eng* 2(2):22–25
- Lucas C, Moghimi S (2004) Applying BELBIC (brain emotional learning based intelligent controller) to an autoland system. *WSEAS Transactions on Systems* 3(1):284–290
- Narendra KS, Thathachar MAL. (1974) Learning automata—a survey. *IEEE Trans Syst Man Cybernet SMC* -4:323–334
- Narendra KS, Thathachar MAL (1989) Learning automata: an introduction: Prentice-Hall, Inc, Upper Saddle River
- Thathachar MAL, Sastry PS (2002) Varieties of learning automata: an overview. *IEEE Trans Syst Man Cybern B* 32(6):711–722. <https://doi.org/10.1109/TSMCB.2002.1049606>
- Lotfi E, Khosravi A, Akbarzadeh-T MR, Nahavandi S (2014) Wind power forecasting using emotional neural networks. Paper presented at the 2014 IEEE International Conference on Systems, Man, and Cybernet (SMC)
- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
- Ojaghzadeh Mohammadi SD, Bahar A, Fakhroosavy SH, Setayeshi S (2010) Optimal column base plate design using a modified genetic algorithm based on Newton-Raphson method. In: 3rd International conference on advanced computer theory and engineering. Chengdu, China
- Kennedy J, RC E (1995) Particle swarm optimization. Paper presented at the Proc. IEEE International Conference on Neural Networks (Perth, Australia), Piscataway
- Kennedy J (2010) Particle Swarm Optimization. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning. Springer, Boston, pp 760–766
- Kumar N, Vidyarthi DP (2016) A novel hybrid PSO–GA metaheuristic for scheduling of DAG with communication on multiprocessor systems. *Eng Comput* 32(1):35–47. <https://doi.org/10.1007/s00366-015-0396-z>
- Hasanipanah M, Noorian-Bidgoli M, Jahed Armaghani D, Khamesi H (2016) Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling. *Eng Comput* 32(4):705–715. <https://doi.org/10.1007/s00366-016-0447-0>
- Hasanipanah M, Armaghani Jahed, Amnieh D., Bakhshandeh, Majid H, M. Z. A., & Tahir M. M. D. (2016) Application of PSO to develop a powerful equation for prediction of flyrock due to blasting. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-016-2434-1>
- Hasanipanah M, Naderi R, Kashir J, Noorani SA, Qaleh A. Z. A. (2017) Prediction of blast-produced ground vibration using particle swarm optimization. *Eng Comput* 33(2):173–179
- Hasanipanah M, Amnieh HB, Arab H, Zamzam MS (2016) Feasibility of PSO-ANFIS model to estimate rock fragmentation produced by mine blasting. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-016-2746-1>
- Ghasemi E, Kalhori H, Bagherpour R (2016) A new hybrid ANFIS–PSO model for prediction of peak particle velocity

- due to bench blasting. *Eng Comput* 32(4):607–614. <https://doi.org/10.1007/s00366-016-0438-1>
31. Zhang J, Xia P (2017) An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models. *J Sound Vib* 389:153–167. <https://doi.org/10.1016/j.jsv.2016.11.006>
 32. Gholizad A, Ojaghzadeh Mohammadi SD (2017) Reliability-based design of tuned mass damper using Monte Carlo simulation under artificial earthquake records. *Int J Struct Stab Dyn* 17(10):1750121
 33. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
 34. Newland DE (1993) An introduction to random vibrations, spectral and wavelet analysis, 3rd edn. Wiley, New York
 35. Jang J. S. R. (1993) ANFIS: adaptive-network-based fuzzy inference system. *IEEE Trans Syst Man Cybernet* 23(3):665–685. <https://doi.org/10.1109/21.256541>
 36. Liu Y, Zhou S, Chen Q (2011) Discriminative deep belief networks for visual data classification. *Pattern Recognit* 44(10):2287–2296. <https://doi.org/10.1016/j.patcog.2010.12.012>
 37. Lee H, Ekanadham C, Ng AY (2007) Sparse deep belief net model for visual area V2. In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Vancouver, British Columbia, Canada: Curran Associates Inc., 873–880
 38. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507
 39. https://earthquake.usgs.gov/learn/topics/mag_vs_int.php