Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 4790 – 4803

doi:10.3906/elk-1412-193

Speech recognition using ANN and predator-influenced civilized swarm optimization algorithm

Teena MITTAL[1,*], **Rajendra Kumar SHARMA**[2]

[1]Department of Electronics and Communication Engineering, Thapar University, Patiala, Punjab, India
[2]School of Mathematics and Computer Applications, Thapar University, Patiala, Punjab, India

Received: 29.12.2014 • **Accepted/Published Online:** 13.09.2015 • **Final Version:** 06.12.2016

Abstract: This paper proposes a hybrid optimization technique, predator-influenced civilized swarm optimization, by integrating civilized swarm optimization (CSO) and predator–prey optimization (PPO) techniques. CSO is the integration of the attributes of particle swarm optimization and a society civilization algorithm (SCA). In the SCA, the swarm is divided into a few societies, and each society has its own society leader (SL); other individuals of the society are termed society members. The combination of all such societies forms a civilization, and the best-performing SL becomes the civilization leader (CL). In CSO, SLs and members update their positions through the guidance of their own CL and SLs, respectively, along with their best positions. In the proposed technique, the PPO technique is integrated with CSO, in which a predator particle is included in the swarm. The predator always tries to chase the CL in a controlled manner, which maintains diversity in the population and avoids local optimum solutions. The proposed optimization technique is applied to optimize the weights and biases of an artificial neural network (ANN) trained for speech recognition. Two databases have been used; one is a TI-46 isolated word database in clean and noisy conditions, and the other is a self-recorded Hindi numeral database. To evaluate the performance of the proposed technique, 2 performance criteria, correlation coefficient and mean square error, are applied. The results obtained by an ANN with the proposed technique outperform the results obtained by an ANN trained with particle swarm optimization, PPO, CSO, and backpropagation techniques in terms of correlation coefficient and mean square error.

Key words: Neural network, speech recognition, civilized swarm optimization, predator–prey optimization, particle swarm optimization, TI-46 database

1. Introduction

Particle swarm optimization (PSO) is a well-recognized global search optimization method based on swarm intelligence such as birds flocking, fish schooling, etc. [1]. In the PSO algorithm, diversity of the swarm is maintained because the information of the most successful particle is shared among all the particles of the swarm. Despite several advantages of PSO, the exploitation capability of the algorithm is not very satisfactory. To improve the performance of PSO, the research community has proposed numerous modifications. One potential suggestion is the inclusion of a predator particle with a prey swarm proposed by Silva et al. [2] as the PPO model. In the PPO model, prey particles try to search for the global optimum solution, and the predator particle always chases the global best prey position. The predator particle improves the search capability of the algorithm; it has been successfully applied to solve various multimodal optimization problems [3].

*Correspondence: tnarang28@gmail.com

CSO is one potential global search technique and is applied to solve various complex optimization problems [4]. CSO is basically an integrated technique of SCA [5] and PSO [1]. In CSO, SMs update their positions with the information obtained from the SL and their personal best positions. The SLs are updated by following the CL and using information acquired from their best positions. In this paper, predator-influenced civilized swarm optimization (PCSO) is proposed, based on the integration of CSO [5] and PPO [2]. In PCSO, the predator particle is always trying to chase the CL particle, which improves the search capability of the proposed technique.

Artificial neural networks (ANNs) [6] are effective tools to solve classification and recognition problems. ANNs are inspired by biological neural network systems in which input information is processed through an interconnected group of neurons [7]. The objective or fitness function is defined to minimize the mean squared error (MSE) between the actual output and the desired output of ANN. Hence, proper training of the ANN is required to minimize the MSE. The MSE is minimized in weight space by determining the optimal weights for the ANN [8]. For training ANNs, conventional search techniques such as backpropagation (BP) techniques and the recursive least squares (RLS) learning algorithm are commonly used [9]. BP is a gradient-based method having slow convergence characteristics and may be trapped at a local optimum solution [10]. The RLS algorithm is very fast as compared to the BP technique, but it requires more complicated mathematical operations [11]. To overcome this problem, various optimization techniques have been proposed by several researchers to train the weights and biases of ANNs. Wang et al. [12] applied a teaching-learning–based optimization algorithm with neighborhood search to optimize weights and biases of ANNs. Jànos [13] applied a global optimization framework for parameterization of ANNs. Das et al. [14] applied an ANN trained with PSO to the problem of channel equalization. Although some of the global search techniques have been applied to optimize weights of ANNs, these algorithms may still converge to the local optimum solution due to lack of exploitation capability.

Speech is the most natural means of communication between human beings and the environment. For centuries, researchers have been trying to develop machines that can understand and produce speech as humans do [15]. Automatic speech recognition (ASR) has been the subject of intensive research for the last several decades. The challenges behind research in speech recognition (SR) lie in its interdisciplinary nature [15]. Feature extraction from the speech signal and recognition of speech based on these features are 2 key steps in ASR. A number of features have been reported in the literature for SR. The linear predictive cepstral coefficient (LPCC) [15], mel frequency cepstral coefficient (MFCC) [15], and wavelet packet-based coefficients [16] are widely used features.

The classification process is the main decision-making process of an SR system. It uses the features extracted from the speech signal in the form of coefficients to identify speech according to the implemented rules. Various classification methods, such as ANNs, the hidden Markov model (HMM), and support vector machines (SVMs), are being employed by researchers for ASR. Wu and Chan [17] presented a speaker-independent isolated word speech recognition model using an ANN. A comparison of different spectral analysis models for SR using neural networks was performed by Zebulun et al. [18]. Krishnan et al. [16] used an ANN for recognition of Malayalam words. Dede and Sazli [19] worked on isolated SR with different ANN topologies. Jay et al. [20] worked on a connected word SR algorithm using the HMM. Yoshua et al. [21] presented integration of an ANN and the HMM for speaker-independent SR using the TIMIT continuous speech database. Maheshwari et al. [22] discussed an ANN- and HMM-based speaker-independent SR system for British English.

The intent of this paper is to propose an optimization algorithm, PCSO, to optimize ANN weights

and biases to minimize the MSE between predicted and actual outputs for speech recognition. In the proposed optimization algorithm, a predator particle tries to chase the CL. The interactions between the predator particle and the CL improve the searching capability of the algorithm by creating diversity in the population. The predator plays the role of searching around the CL in a concentrated manner, whereas the SMs of that society explore a solution space for escaping the predator. The ANN classifier optimized with the proposed technique is applied to recognize standard database TI-46 isolated words [23] and a self-recorded database of Hindi numerals.

2. Feature extraction

In the feature extraction module, the speech signal is converted into a set of features that are further explored in the classification process. The widely used features that can be extracted from the speech signal are the LPCC, MFCC, and wavelet packet-based MFCC (WPMFCC) [24]. To obtain the LPCC, a linear prediction mathematical operation is used where future values of a digital signal are estimated as a linear function of previous samples.

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p) \tag{1}$$

For each sample, a prediction error $e(n)$ is defined as follows:

$$e(n) = s(n) - \bar{s}(n), \tag{2}$$

where n is the index of the current sample, $\bar{s}(n)$ is the linearly predictive sample, s(n) is the actual sample, p is the degree of the LPC model, and a_i is the filter predictor coefficients where $i = 1, 2, ..., p$. By minimizing the mean-square prediction error $e(n)$, over a finite interval, a unique set of predictor coefficients can be determined.

To extract the MFCCs, the speech sample is taken as the input, and a hamming window is applied to minimize the discontinuities of a signal. A discrete Fourier transform (DFT) is used to generate the mel filter bank. According to mel frequency warping, the width of the triangular filters varies; thus, the log total energy in a critical band around the center frequency is included. After warping, a number of coefficients are obtained. Finally, the inverse DFT is used to compute cepstral coefficients [15].

Wavelet packet transform (WPT) has been explored as a powerful tool for speech feature extraction. It uses variably sized time-windows for different frequency bands, which results in high frequency resolution in low bands and low frequency resolution in high bands. WPT decomposes the signal in approximation and detail coefficients. Each subspace in the tree is indexed by its depth j and number of subspaces p below it. The 2 wavelet packet (WP) orthogonal bases at a parent node $(j,\ p)$ are defined by:

$$\psi_{j+1}^{2p}(k) = \sum_{n=-\infty}^{\infty} h[n]\psi_j^p(k - 2^j n), \tag{3}$$

$$\psi_{j+1}^{2p+1}(k) = \sum_{n=-\infty}^{\infty} g[n]\psi_j^p(k - 2^j n), \tag{4}$$

where h[n] is a low pass and g[n] is a high pass filter given by the following equations:

$$h[n] = \left\langle \psi_{j+1}^{2p}(u), \psi_j^p(u - 2^j n) \right\rangle, \tag{5}$$

$$g[n] = \left\langle \psi_{j+1}^{2p+1}(u), \psi_j^p(u - 2^j n) \right\rangle. \tag{6}$$

Thus, a balanced binary tree is formed by using full j level wavelet packet decomposition, having more than $2^{2^{j-1}}$ orthogonal bases. To extract the WPMFCC, the wavelet packet transform coefficients of the speech signal are first computed. The MFCC of these coefficients is then calculated.

3. Artificial neural network

ANN refers to the interconnections between the neurons in the different layers of a system. In a 3-layer feedforward ANN, an input layer, an intermediate or hidden layer, and an output layer are connected in a systematic manner [13]. The input layer sends data via synapses to the hidden layer, and then via more synapses to the output layer. The synapses store parameters are called weights; they update the value based on the training process.

A feedforward network computes, for each layer, the outputs of the corresponding nodes given the layer input vector $x_i = (x_1, x_2, ..., x_n)$ as:

$$\sum_j = b_j + \sum_{i=1}^{n} x_i W_{ji}, \tag{7}$$

$$out_j = f(sum_j), \tag{8}$$

where n is the number of neurons in the current layer, x_i represents the input at neuron i, W_{ji} is the weight connection between neuron j and neuron i, b_j is the bias of neuron j, $f()$ is the transfer function, and $\sum_j$ and out_j are the net input and output, respectively, at the jth neuron.

In this work, the hyperbolic tangent sigmoid transfer function is taken between the input and hidden layers, and the linear transfer function is chosen between the hidden and output layers. The expressions for both transfer functions are given by the following equations:

$$sigmoid(sum_j) = \frac{\exp(sum_j) - \exp(-sum_j)}{\exp(sum_j) + \exp(-sum_j)}, \tag{9}$$

$$Purelin(sum_j) = sum_j. \tag{10}$$

The training of the ANN is carried out by applying an iterative optimization process to minimize the MSE by updating the weights and biases appropriately [14]. The MSE is defined as the error between the expected output and actual outputs, given as:

$$MSE = \frac{\sum_{i=1}^{n} (A_i - E_i)^2}{n}, \tag{11}$$

where A_i is actual output and E_i is expected output; n is the number of training points.

4. Predator-influenced CSO

The PCSO is an integrated technique of CSO and PPO. The CSO is further an integrated technique of SCA and PSO. In SCA, the swarm is divided into civilized societies, and every society has its own society leader. The SL is the best-performing particle of the society, and the other particles are SMs. The best-performing SL is treated as the CL. The SMs belongs to a particular society depending on their Euclidean distance in the

parametric space and are computed as:

$$D_s = (\sum_{i=1}^{N} ((SL_s - SM_r)^2)^{\frac{1}{2}} \quad (s = 1, 2, ..., N_s), \quad (r = 1, 2, ..., N_r). \tag{12}$$

The society member SM_r is assigned to society 's' if it is closer to SL_s, where N_s and N_r represent the number of societies and society members, respectively, and N represents the number of dimension.

In the proposed PCSO algorithm, the predator particle is incorporated with the civilized swarm. The predator always tries to chase the CL, and it becomes difficult for SMs and the CL to stay at the preferred location to explore the search area effectively. The effect of the predator is controlled by probability fear (pf).The predator searches around the CL in a concentrated manner, while swarm particles explore the search space in escaping from the predator.

The procedure steps for PCSO are as follows:

Step 1: The predator velocity $V_{Pi}(k)$ and position $X_{Pi}(k)$ at the kth iteration are updated as:

$$V_{Pi}(k+1) = C_p\,(CL_i(k) - X_{Pi}(k)) \quad (i = 1, 2, ..., N), \tag{13}$$

$$X_{Pi}(k+1) = X_{Pi}(k) + V_{Pi}(k+1) \quad (i = 1, 2, ..., N), \tag{14}$$

where C_p is a uniformly distributed random number determining how fast the predator catches the CL; $CL_i(k)$ is civilization leader position at the kth iteration.

Step 2: The societies that do not have a CL update the velocity of society leaders $V_{is}^{SL}(k)$ by following the CL, and information is acquired from their best positions as:

$$V_{is}^{SL}(k+1) = wV_{is}^{SL}(k) + C_{SL1}r_1(Pbest_{is}^{SL}(k) - SL_{is}(k)) + C_{SL2}r_2(CL_i(k) - SL_{is}(k))$$

$$(i = 1, 2, ..., N) \quad (s = 1, 2, ..., N_s), \tag{15}$$

where w is inertia weight and its value decreases from 0.9 to 0.4 with iteration; C_{SL1} and C_{SL2} are acceleration coefficients, which accelerate the SL towards its own position and CL, respectively; r_1 and r_2 are uniformly distributed random numbers; $Pbest_{is}^{SL}$ is the personal best position of s^{th} SL; and $SL_{is}(k)$ is the sth SL position at the kth iteration.

The velocity of society members $V_{ir}^{SM}(k)$ is updated by following the corresponding SL, and information is acquired from their best positions as:

$$V_{ir}^{SM}(k+1) = wV_{ir}^{SM}(k) + C_{SM1}r_3(Pbest_{ir}^{SM}(k) - SM_{ir}(k)) + C_{SM2}r_4(SL_{is}(k) - SM_{ir}(k))$$

$$(i = 1, 2, ..., N) \quad (r = 1, 2, ..., N_r), \tag{16}$$

where C_{SM1} and C_{SM2} are acceleration coefficients, which accelerate the SM towards its own position and SL, respectively; r_3 and r_4 are uniformly distributed random numbers; $Pbest_{ir}^{SM}$ is the personal best position of the rth SM; and $SM_{ir}(k)$ is the rth SM position at the kth iteration.

Step 3: The society that contains the CL particle updates the velocity of civilized leader $V_i^{CL}(k)$ and society member $V_{ir}^{SM}(k)$ as follows:
For civilized leader:

$$V_i^{CL}(k+1) = \left\{ \begin{array}{l} wV_i^{CL}(k)+C_{L1}r_5(Pbest_i^{CL}(k)-CL_i(k))\ pf<pf\max \\ wV_i^{CL}(k)+C_{L1}r_5(Pbest_i^{CL}(k)-CL_i(k))+C_{L2}a_i\exp(-b_id)\ \ pf\geq pf\max \end{array} \right\},$$

$$(i=1,2,...,N)\ \ (r=1,2,...,N_r) \tag{17}$$

where $Pbest_i^{CL}(k)$ is the personal best position of CL at the kth iteration; C_{L1} is the acceleration coefficient, which accelerates the CL towards its own best position; C_{L2} is a uniformly distributed scaled random number that influences the effect of the predator on the prey; a_i provides the maximum amplitude of the predator's effect on SMs and b_i controls the effect of the predator; d is the Euclidean distance between the predator and SMs; pf and $pf_{\max}$ are probability fear and the maximum probability fear, respectively; pf is a uniformly distributed random number; and r_5 is uniformly distributed random number.

For society members:

$$V_{ir}^{SM}(k+1) = \left\{ \begin{array}{l} wV_{ir}^{SM}(k)+C_{SM1}r_3(Pbest_{ir}^{SM}(k)-SM_{ir}(k))+C_{SM2}r_4(SL_{isr}(k)-SM_{ir}(k));\ pf<pf\max \\ wV_{ir}^{SM}(k)+C_{SM1}r_3(Pbest_{ir}^{SM}(k)-SM_{ir}(k))+C_{SM2}r_4(SL_{isr}(k)-SM_{ir}(k))+C_{SM3}a_i \\ \exp(-b_id); pf\geq pf\max \end{array} \right\}$$

$$(i=1,2,...,N)\ \ (r=1,2,...,N_r). \tag{18}$$

Step 4: The personal best positions $(Pbest)$ of the CL, SLs, and SMs are updated based on objective function evaluation and given as:

$$Pbest(k+1) = \left\{ \begin{array}{l} X(k)+V(k+1); \phi(Pbest(k+1))<\phi(X(k)) \\ Pbest(k);\ \ \text{otherwise} \end{array} \right\}, \tag{19}$$

where $\phi(X(k))$ is the objective function, evaluated at $X(k)$ position for the kth iteration; $X(k)$ and $V(k)$ are society particles' position and velocity for the kth iteration, respectively.

Step 5: Formation of new swarm: initially, the swarm is taken as an empty set after the positions of CL, SLs, and SMs are updated and included in the new swarm. The positions are updated as:

$$CL_i(k) = CL_i(k) + V_i^{CL}(k+1)\ \ (i=1,2,...,N), \tag{20}$$

$$SL_{is}(k) = SL_{is}(k) + V_{is}^{SL}(k+1)\ \ (i=1,2,...,N),\ (s=1,2,...,N_s), \tag{21}$$

$$SM_{ir}(k) = SM_{ir}(k) + V_{ir}^{SM}(k+1)\ \ (i=1,2,...,N),\ (r=1,2,...,N_r). \tag{22}$$

5. Implementation of PCSO for speech recognition

The main aim of this research is to minimize the MSE between expected and actual outputs by optimizing weights and biases of the ANN through the proposed algorithm for speech recognition. Initially, acoustic

features of speech signals are acquired by the feature extraction technique. These acoustic features are given as input to the ANN classification module. In this work, a 3-layer ANN architecture having M input neurons, H hidden neurons, and K output neurons has been undertaken. For this ANN model, the decision variables are computed as:

$$L = (M+1)H + (H+1)K. \tag{23}$$

Each decision variable is set in the range of –1 to +1, where $M \times H$ weights are used to connect the input layer and the hidden layer, and H biases are used for hidden layer neurons. In a similar way, $H \times K$ weights are used to connect the hidden layer and output layer and K biases are used for output layer neurons.

The swarm having society particles and a single predator is represented as:

$$Swarm = [[S^1]\ [S^2]\ \ldots\ [S^m]\ \ldots\ [S^{NP}]\ [PS]],$$

where the matrix $[S^m]$ represents the mth particle of the swarm and it is defined as $S^m = [[W^m]\,[b^m]]$, where W^m and b^m represent weights and bias, respectively; [PS] represents the predator particle having the same dimensions; and NP represents the number of society particles in a swarm.

The step-wise procedure of PCSO-based SR is described in Algorithm 1.

Algorithm 1: Proposed algorithm for SR.

1. Read training data, expected outputs, parameters of algorithm, and set maximum number of iterations $k^{\max}$.
2. Randomly initialize ANN weights and biases as society and predator positions.
3. Initialize society and predator velocity randomly.
4. Initialize iteration index $k = 1$.
5. Compute net input as given by Eq. (refeq7a).
6. Compute the output by passing net input through the transfer function.
7. Compare the obtained and desired result; compute MSE by Eq. (8).
8. Arrange society particles on the basis of MSE; best-performing particle is selected as SL and remaining particles are treated as SMs.
9. The Euclidean distances between SMs and SL are computed by Eq. (12) and SMs are selected for a particular society.
10. Select CL among SLs on the basis of MSE.
11. Randomly generate probability fear.
12. Update predator velocity and position as given by Eqs. (refeq13a) and (14), respectively.
13. For the societies that do not have a CL, update SL and SM velocity by applying Eqs. (15) and (16), respectively; their positions are updated by applying Eqs. (21) and (22), respectively.

14. For the society that has a CL, update CL and SM velocity by applying Eqs. (17) and (18), respectively; their positions are updated by applying Eqs. (20) and (22), respectively.

15. Update personal best positions as given by Eq. (19).

16. Repeat steps 4 to 15 until all training points are finished.

17. $K = k + 1$

    IF $(k \leq k^{\max})$ THEN

        Select first training point and GOTO step 5.

    ENDIF

18. STOP

---

6. Experimental details

In this experiment, a benchmark database, TI-46, and a self-created Hindi numeral database have been tested under both clean and noisy conditions. The database TI-46 is a speaker-dependent isolated word corpus having 2 subsets, TI-20 and TI-ALPHA [23]. All data samples have been digitized with sampling frequency of 12.5 kHz. The third database consists of Hindi speech samples recorded in a quiet room with sampling frequency of 44.1 kHz. For each clean speech database, the vocabulary, vocabulary size, number of instances used for training as well as testing, and number of speakers are given in Table 1.

Table 1. Databases.

Database	Vocabulary	Size	No. of instances		No. of speakers	
			Training	Testing	Male	Female
TI-20	10 English digits and 10 control words	20	10	16	8	8
TI-ALPHA	English alphabets	26	10	16	8	8
Hindi-digit Database	'shoonya' through 'nine'	10	10	10	5	5

Three different features, LPCC, MFCC, and WPMFCC, have been extracted for each sample of every database. To extract LPCC and MFCC features, the speech signal is divided into a fixed number of 40 frames each of 25 ms with 50% superposition. After framing, the hamming window is used for windowing, as it introduces the least amount of distortion. For each of the 40 frames, 13th order LPCC features are extracted for all speech utterances. For MFCC feature extraction, 20 triangular mel filters are used. The MFCC feature vector consists of 13 coefficients for each frame. To extract WPMFCC features, the signal is initially decomposed into subbands using the WPT. Each speech sample has been decomposed into a 2-level WPT. After decomposition of the speech signal into subbands, it is given to the MFCC analysis block, and 13 MFCC coefficients from each of the bands are extracted. The Daubechies 4 wavelet is used for the purpose of decomposition of the speech signal.

The extracted acoustic features are given as input to the ANN classification module. A feedforward ANN model consisting of an input layer, a hidden layer, and an output layer has been considered in this work. The output of the ANN model depends on the choice of transfer function between its layers, as it affects the

learning rate and performance of the model. Various combinations of transfer functions were tested and it was found that the combination of hyperbolic tangent sigmoid for the hidden layer and linear transfer function for the output layer produce better performance. Another critical issue that affects the performance of the ANN is the number of neurons in the hidden layer. To select the optimum number of neurons in the hidden layer, several experiments have been conducted by applying the considered algorithms. Five to 30 neurons were tried to construct the ANN model. The MSE was found to require a minimum of 15 neurons in the hidden layer for the Hindi digit database with MFCC features when the ANN is trained by the BP algorithm. Hence, a network with 15 neurons in the hidden layer has been chosen as the best architecture for the MFCC features with BP algorithm. Five neurons for LPCC features and 5 neurons for WPMFCC features in the hidden layer have been found suitable with the BP algorithm. The same procedure is applied for the other databases to select the optimum number of neurons in the hidden layer of the ANN model trained with all considered algorithms with MFCC, LPCC, and WPMFCC features.

7. Results and discussion

To obtain satisfactory results, parameters of any global search technique need to be adjusted; to get the parameter values of PCSO, 30 trials have been performed. In each trial run, the population size is set to 20 and 1 predator particle is undertaken. Parameters have been varied between minimum set values and maximum set values with a certain step size. Maximum and minimum value, step size, and parameters values are given in Table 2. For each trial, maximum iterations are set to 100. Figure 1 shows the block diagram of the speech recognition system. The first block shows the raw speech waveform of word "paanch"; the second block is of a silence-removed waveform from which MFCC features have been extracted, as presented in the third block. These features have been given as input to the ANN model, whose weights and biases have been optimized using PCSO with the optimal parameter setting to minimize the MSE. The variation in training MSE with iterations is shown in Figure 2. The results achieved by the ANN with the proposed technique are compared with results achieved by an ANN trained with PSO, PPO, and CSO techniques. To set the optimum parameters of PSO, PPO, and CSO, the same procedure is repeated as applied for the proposed technique. The optimum parameters of these algorithms are as follows:

Table 2. Parameter values of PCSO algorithm.

Algorithm	Parameter	Minimum value	Maximum value	Step size	Value
PCSO	N_s	2	10	1	4
	$(C_{SL1}, C_{SL2}, C_{SM1}, C_{SM2}, C_{L1})$	0.25	2.0	0.25	(0.5, 0.5, 0.25, 0.75, 2.0)
	C_{L2}	0.0	2.0	–	–
	a_i	0.25	1.0	0.25	0.5
	b_i	0.25	1.0	0.25	1.0
	$pf_{\max}$	0.50	1.0	0.05	0.95

- PSO: acceleration constants $C_1 = 1.5$, and $C_2 = 2$.
- PPO: $C_1 = 1.5$, $C_2 = 2$, maximum value of $C_3 = 2.0$, $a_i = 0.25$, $b_i = 0.75$, and $pf_{max} = 0.95$.
- CSO: $N_s = 5$, $C_{SL1} = 0.25$, $C_{SL2} = 0.5$, $C_{SM1} = 0.25$, $C_{SM2} = 0.75$, and $C_{L1} = 1.75$.

For validation purposes, the clean speech dataset is divided into training and testing parts as given in Table 1; to recognize the TI-20 database with noise, the 10-fold cross-validation approach is used. In this

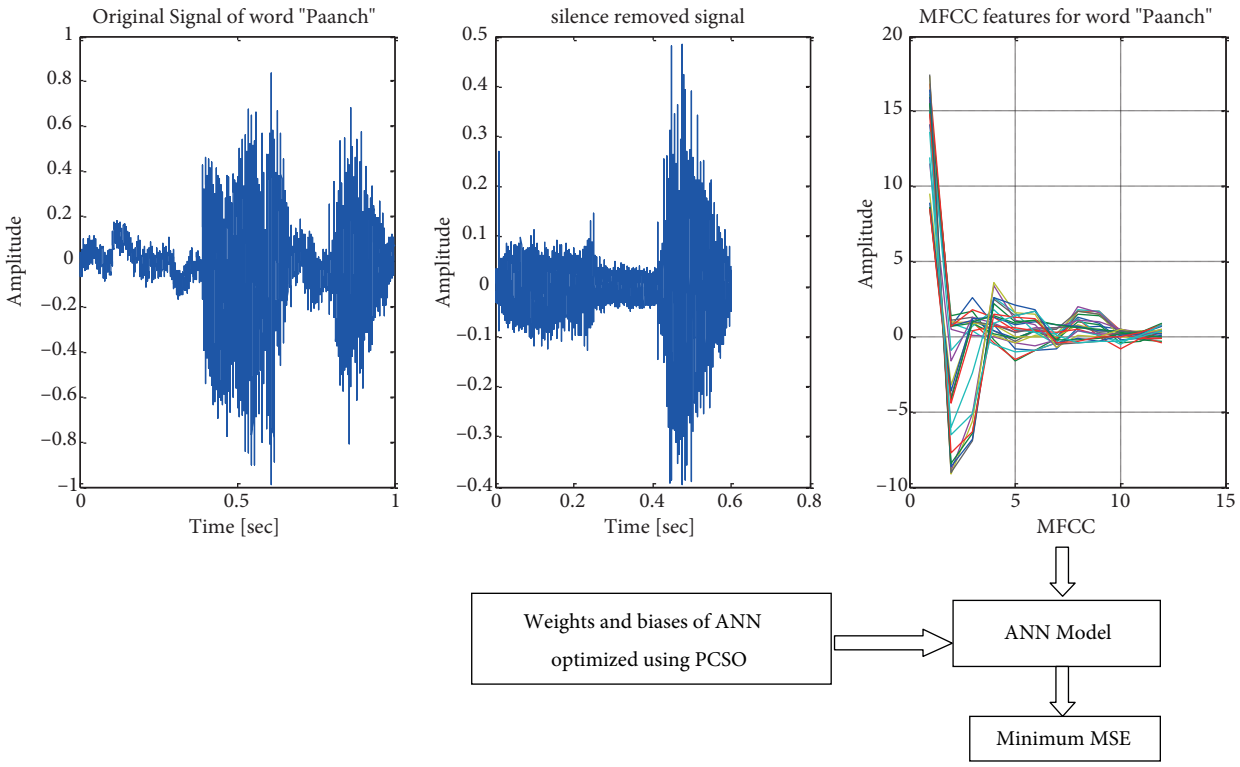

Figure 1. Basic building blocks for speech recognition system using optimized ANN with PCSO.

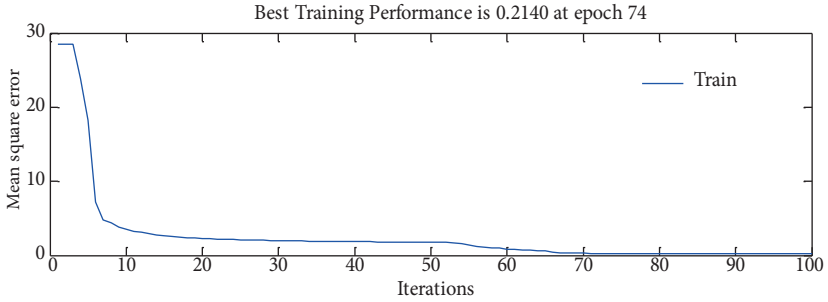

Figure 2. Variation in MSE with iterations obtained by optimized ANN with PCSO.

approach, the database is divided into 10 equal parts. Nine parts are used for training and the remaining part is used for testing the model. To compare the performance of the ANN trained with PCSO and the ANN trained with other algorithms, the correlation coefficient (R) and MSE are evaluated. R is computed to measure the linearity between expected values E_i and actual values A_i, and MSE measures the average squared error between E_i and A_i. The MSE is computed by Eq. (6), and the correlation coefficient is defined as:

$$R = \frac{\sum_{i=1}^{N} (E_i - \overline{E})(A_i - \overline{A})}{\sqrt{\sum_{i=1}^{N} (E_i - \overline{E})^2 \sum_{i=1}^{N} (A_i - \overline{A})^2}}, \tag{24}$$

where $\overline{E}$ and $\overline{A}$ are the average value of the expected and actual values, respectively.

It is evident from Table 3 that the MSE obtained by the ANN trained by PCSO is less than the MSE achieved by the ANNs trained by PSO, PPO, CSO, and BP algorithms for all considered databases with LPCC, MFCC, and WPMFCC features. The comparison of correlation coefficient is presented in Table 4, and it is observed that R obtained by the ANN with PCSO is better than Robtained by the ANNs trained by PSO, PPO, CSO, and BP algorithms for all considered databases with LPC, MFCC, and WPMFCC features. The regression plot for the Hindi database with WPMFCC features obtained from the optimized ANN with PCSO is presented in Figure 3. It is evident from Figure 3 that there are very small deviations between training, testing, and validation performances, and R is 0.9565. From this discussion, it is concluded that PCSO has more effectively optimized the weights and biases of the ANN as compared to the PSO, PPO, CSO, and BP algorithms.

Table 3. Comparison of MSEs obtained by different training algorithms for ANNs.

Database	Features	MSE				
		BP	PSO	PPO	CSO	PCSO
TI-20	LPCC	0.512	0.337	0.323	0.274	0.234
	MFCC	0.456	0.312	0.294	0.268	0.167
	WPMFCC	0.438	0.227	0.192	0.147	0.122
TI-ALPHA	LPCC	0.523	0.348	0.305	0.318	0.225
	MFCC	0.467	0.316	0.287	0.296	0.186
	WPMFCC	0.439	0.314	0.267	0.283	0.238
Hindi digits	LPCC	0.632	0.524	0.516	0.487	0.427
	MFCC	0.578	0.487	0.424	0.434	0.369
	WPMFCC	0.557	0.446	0.411	0.389	0.321

Table 4. Comparison of correlation coefficients of ANN trained with different algorithms.

Database	Features	Correlation coefficient (R)				
		BP	PSO	PPO	CSO	PCSO
TI-20	LPCC	0.65	0.84	0.89	0.92	0.93
	MFCC	0.79	0.89	0.90	0.90	0.96
	WPMFCC	0.84	0.92	0.95	0.96	0.98
TI-ALPHA	LPCC	0.72	0.85	0.85	0.88	0.90
	MFCC	0.77	0.87	0.86	0.89	0.93
	WPMFCC	0.79	0.89	0.94	0.95	0.96
Hindi digits	LPCC	0.59	0.75	0.71	0.80	0.83
	MFCC	0.78	0.84	0.85	0.91	0.95
	WPMFCC	0.80	0.88	0.90	0.93	0.95

To further evaluate the robustness of PCSO, a series of experiments have been conducted by considering noisy speech test samples. Noisy test samples are obtained by artificially adding white noise under a wide range of signal to noise ratio (SNR) from 0 to 40 dB in steps of 5 dB into the test samples of the TI-20 database. The MSE is presented in Figure 4 for different values of SNR obtained by ANNs trained with various techniques for WPMFCC features. It is observed from Figure 4 that the MSE obtained by the ANN trained by the PCSO algorithm is less than the MSE of the PSO, PPO, CSO, and BP algorithms.

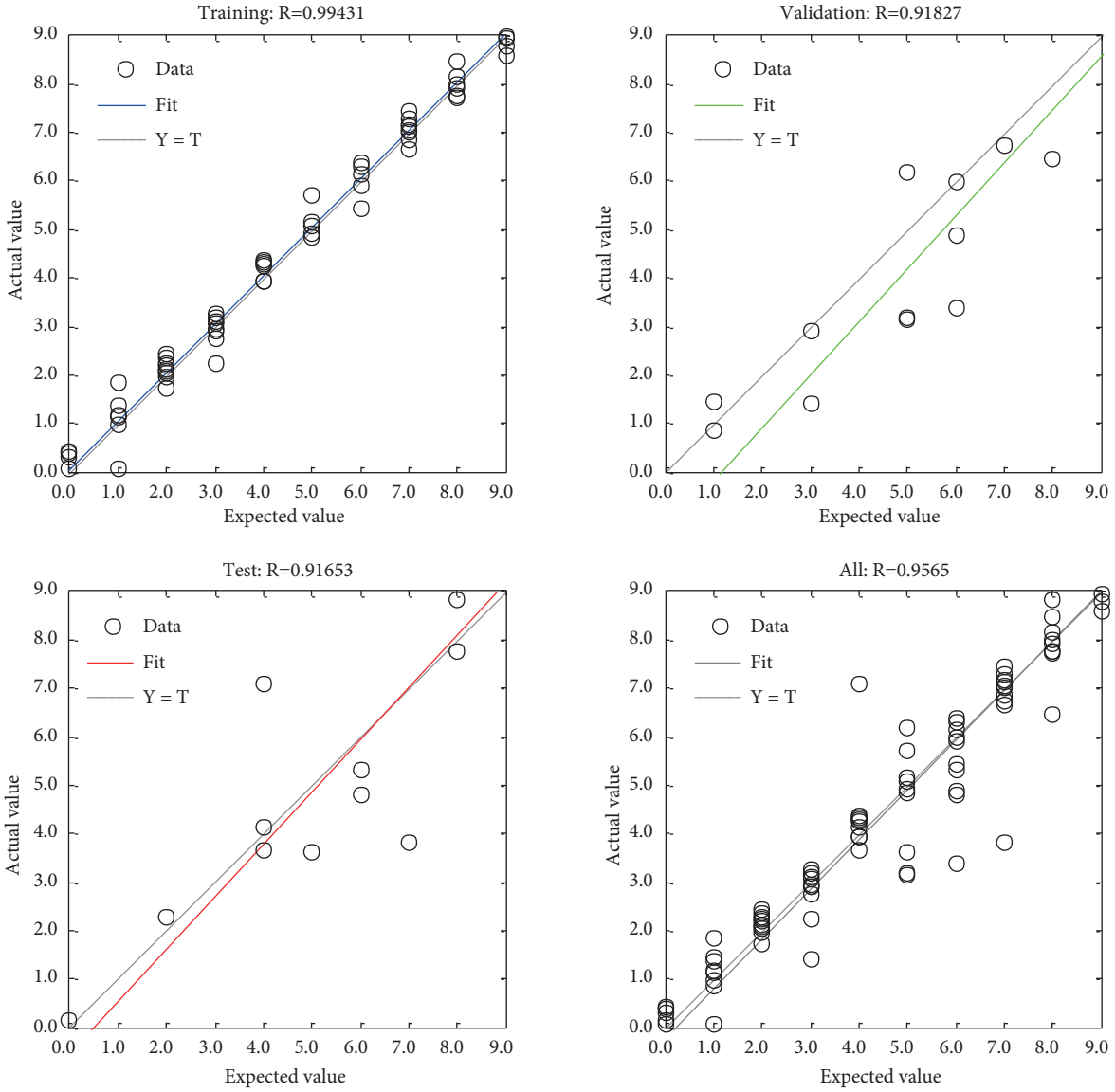

Figure 3. Regression plot for Hindi digit database obtained by optimized ANN with PCSO.

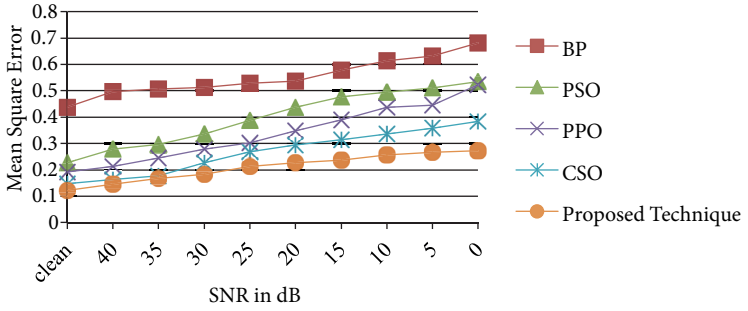

Figure 4. MSE versus SNR for TI-20 database with different training algorithms for ANNs.

8. Conclusions

In this paper, a PCSO optimization algorithm is proposed based on integration of CSO and PPO. In the proposed algorithm, a predator particle chases the CL, which includes an additional capability to escape from the local optimum solution. The predator exploits the search around the CL, whereas the society particles explore the solution space in escaping from the predator, so that the society particles play the role of diversification and the predator particle exploits the search space. The PCSO is applied to train ANN weights and biases to minimize the MSE between predicted and actual outputs. The experiment is tested on a TI-46 clean speech word database, a Hindi numerals database, and a TI-20 database with different ranges of SNR. Finally, from the experimental results, it is concluded that the MSE and R obtained by an ANN optimized with PCSO outperforms the results achieved by the PSO, PPO, CSO, and BP algorithms.

References

[1] Kennedy J, Eberhart R. Particle swarm optimization. In: IEEE 1995 International Conference on Neural Networks; 27 November–1 December 1995; Perth, Australia. Piscataway, NJ, USA: IEEE. pp. 1942-1948.

[2] Silva A, Neves A, Costa E. An empirical comparison of particle swarm and predator prey optimization. In: O'Neill M, Sutcliffe RFE, Ryan C, Eaton M, Griffith NJL, editors. Artificial Intelligence and Cognitive Science. Berlin, Germany: Springer, 2002. pp. 103-110.

[3] Costa E, Silva A, Coelho LDS, Lebensztajn L. Multiobjective biogeography based optimization based on predator–prey approach. IEEE T Magn 2012; 48: 951-954.

[4] Selvakumar AI, Thanushkodi K. Optimization using civilized swarm: solution to economic dispatch with multiple minima. Electr Power Syst Res 2009; 79: 8-16.

[5] Ray T, Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behaviour. IEEE T Evol Comput 2003; 7: 386-396.

[6] Mehrotra K, Mohan CK, Ranka S. Elements of Artificial Neural Networks. Cambridge, MA, USA: MIT Press, 1997.

[7] Haykin S. Neural Networks: A Comprehensive Foundation. 1st ed. New York, NY, USA: Macmillan, 1991.

[8] Man ZH, Lee K, Wang DH, Cao ZW, Khoo SY. Robust single–hidden layer feedforward network-based pattern classifier. IEEE T Neural Networks Learn Syst 2012; 23: 1974-1986.

[9] Bilski J, Rutkowski L. A fast training algorithm for neural networks. IEEE T Circuit Syst Express Briefs 1998; 45: 1580-1591.

[10] Zhang R, Xu ZB, Huang GB, Wang DH. Global convergence of online BP training with dynamic learning rate. IEEE T Neural Networks Learn Syst 2012; 23: 330-341.

[11] Al-Batah MS, Isa NAM, Zamli KZ, Azizli KA. Modified recursive least squares algorithm to train the hybrid multilayered perceptron network. Appl Soft Comput 2010; 10: 236-244.

[12] Wang L, Zou F, Yang D, Chen D, Jiang Q. An improved teaching–learning–based optimization with neighborhood search for application of ANN. Neurocomputing 2014; 143: 231-247.

[13] Jànos DP. Calibrating artificial neural networks by global optimization. Expert Syst Appl 2012; 39: 25-32.

[14] Das G, Pattnaik PK, Padhy SK. Artificial neural network trained by particle swarm optimization for non–linear channel equalization. Expert Syst Appl 2014; 41: 3491-3496.

[15] Rabiner L, Juang BH. Fundamentals of Speech Recognition. 1st ed. New York, NY, USA: Pearson Education, 1993.

[16] Krishnan VRV, Jayakumar A, Anto PB. Speech recognition of isolated Malayalam words using wavelet features and artificial neural network. In: IEEE 2008 International Symposium on Electronic Design, Test and Applications; 23–25 January, 2008; Hong Kong. Piscataway, NJ, USA: IEEE. pp. 240-243.

[17] Wu J, Chan C. Isolated word recognition by neural network models with cross-correlation coefficients for speech dynamics. IEEE T Pattern Anal Mach Intell 1993; 15: 1174-1185.

[18] Zebulun RS, Perelmuter G, Velasco M, Pacheco MA. A comparison of different spectral analysis models for speech recognition using neural networks. In: IEEE 1997 39th Midwest Symposium on Circuits and Systems; 18–21 August 1996; Ames, IA, USA. Piscataway, NJ, USA: IEEE. pp. 1428-1431.

[19] Dede G, Sazli MH. Speech recognition with artificial neural networks. Digital Signal Process 2009; 20: 763-768.

[20] Jay GW, Lawrence RR, Chin-Hui L, Goldman ER. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE T Acoust Speech Signal Process 1990; 38: 1870-1878.

[21] Yoshua B, Renato DM, Giovanni F, Ralf K. Global optimization of a neural network–hidden Markov model hybrid. IEEE T Neural Networks 1992; 3: 252-259.

[22] Maheshwari NU, Kabilan AP, Venkatesh R. Speech recognition system based on phonemes using neural networks. Int J Comput Sci Network Secur 2009; 9: 148-153.

[23] TI 46–Word Speaker Dependent Isolated Word Corpus. NSIT Speech Disc 1991; 7–1.1.

[24] Farooq O, Datta S. Mel filter-like admissible wavelet structures for speech recognition. IEEE Signal Proc Let 2001; 8: 196-198.