Nonlinear enhancement of noisy speech, using continuous attractor dynamics formed in recurrent neural networks

Louiza Dehyadegary, Seyyed Ali Seyyedsalehi, Isar Nejadgholi

Department of Biomedical Engineering, Amirkabir University of Technology, 424, Hafez Ave, Tehran, Iran

1. Introduction

Recent advances in speech recognition technology have been impressive, but robust recognition of speech in noisy acoustic environments still remains a largely unsolved problem [8]. Automatic speech recognition systems performed reasonably well in controlled and matched training and recognition conditions. However, performance deteriorated when there is a mismatch between training and testing conditions, caused for instance by additive noise [15].

Human beings are capable of recognizing speech that has been heavily corrupted by both stationary or nonstationary noise [14,17]. Hence, speech enhancement has been an interesting area of research for the past two decades [20].

Various approaches to speech enhancement have been proposed to date [7,12]. The aim of all state-of-the-art speech enhancement techniques is to improve the perceptual aspects of quality and intelligibility of speech by utilizing the estimated noise [9]. For example spectral subtraction is a simple technique for speech enhancement [4,16]. In this method, noise is achieved by averaging the noisy frames of speech. Then this noise is subtracted from speech. However this technique is not suitable for nonstationary noise and cannot achieve an appropriate noise filtering [4].

Neural networks applied to speech enhancement, efficiently provide a smoother estimation of signal. The capability of artificial neural networks (ANNs) to approximate any nonlinear function also makes them suitable for nonstationary noise and nonlinear transformations commonly used in speech feature extraction such as LHCB parameters. More details of this issue will be discussed in Section 4.1.

Several researches have used ANNs for speech enhancement [12,18,19,26]. Despite the strong generalization capability of conventional ANNs, they cannot easily model the temporal behavior of speech signal [10]. Thus, the only way to address this issue is to use recurrent and time delay connections in the network [21].

For example, the Hopfield network is a simple auto-associative recurrent neural network [11]. The nodes in this network are a vast simplification of real neurons. They can only exist in one of the two possible states, firing or not firing. The Hopfield network is designed to store a number of patterns so that they can be retrieved from noisy or partial cues. The structure of a Hopfield Neural Network is shown in Fig. 1. This retrieval is achieved by creating an energy surface with some point attractors representing each of the patterns. The noisy and partial cues are states of the system which are close to the attractors. Through the Hopfield network cycles, it slides from the noisy pattern, down the energy surface into the closest point attractor, representing the closest stored pattern.

Discrete Hopfield networks have serious limitations in memory capacity. A Boltzmann Machine is the generalized form of Hopfield network with hidden units [1]. The hidden units, stochastic activations, and simulated annealing in its learning
procedure give Boltzman Machines exceptional power. The hidden units allow a Boltzman Machine to find higher order correlations in the data than a Hopfield network can find, so it can learn arbitrarily complex patterns (Fig. 2).

The Boltzman Machine learning algorithm is based on the fact that the output can be predicted from input units. In the clamping phase, the first phase of training, activation of input and output units is kept constant. In the second or the free-running phase, only the input units are clamped. The weights are modified to reduce the difference between the desired and observed outputs. In this learning algorithm \( \Delta w \), the weight modification matrix is computed according to Eq. (1):

\[
\Delta w = \gamma (p - p'), \quad w_{\text{new}} = w_{\text{old}} + \Delta w
\]

where \( w_{\text{new}} \) is the modified weight matrix, \( w_{\text{old}} \) is the weight matrix before modification, \( p \) and \( p' \) are the desired and observed outputs, and \( \gamma \) is the learning rate.

In general, neural networks with feed-forward structures can map a noise added signal to the clean one because of their capabilities in many to one mapping. To achieve this mapping, we should have enough data of different noisy conditions and the network should have the sufficient capacity to learn the patterns. These networks map the noisy speech signal space to that of the clean speech signal.

However, the efficiency of these networks would depend on the learning noise and usually not all kinds of noise are accessible. So, achieving a method of noise removing and clean signal retrieving with no need of noise estimation is essential. In this work, the efficiency of attractors in recurrent neural networks is applied to propose and implement such a method. In these networks the state trajectory slides down from the noisy patterns to the desired one as an attractor.

Celebrated uses of the recurrent neural networks and attractor dynamics which are biologically simple models of memory [25], include the storage of associative memories [2], the reconstruction of noisy images [13], and the search for shortest paths in the traveling salesman problem [11].

Attractor neural networks are connected with the idea of an energy surface. In these networks, the output is again given to the network as an input with recurrent connections and this cycle is repeated many times to update the state of the network progressively. We call this procedure ‘‘cycling in the network’’. By gradual changes, the state of the network, steps forward to decrease the energy and finally reach an equilibrium point corresponding to a local minimum or an attractor.

In this paper, we plan to improve the noisy speech recognition, using recurrent neural networks with nonlinear functions and the characteristics of the attractors. The recurrent neural network is trained with clean signal. Thus, in training phase, the clean speech patterns trained to the network will figure out the attractors in the state space of the network. In the testing phase, the noisy signal is given to the network and by cycling, the state trajectory is taken away from noisy signals to the attractors which are clean patterns. Then, these cleaned signals are used in speech recognition.

Therefore, we apply the characteristics of attractor dynamics, to nonlinear reduction of the noisy signal space to clean signal subspace. Formation of clean speech subspace as a continuous attractor by the recurrent neural network is necessary to achieve this nonlinear dimension reduction. Basin of attraction corresponding to this continuous attractor is the total input space consisting of all noisy and disturbed or changed signals.

In this work, a reference model is introduced to evaluate the designed models. The structure of this reference network is the same as the main recurrent neural network model, without recurrent connections. The designed models have the ability of noise reduction, so they are expected to reach better recognition accuracies as compared to the reference model.

In the following, we will talk about the attractors. Robust speech recognition using neural networks is explained in Section 3. Experiments and database are discussed in Section 4. Section 5 analyzes the results and conclusion is the last section of the paper.

2. Attractor dynamics

As mentioned in the previous section, the proposed model in this paper which is used to robust recognition of noisy speech is a neural network with recurrent connections and the idea is the application of attractors to reduce the noise. So, in this section, we discuss attractors and explain how they can be used to denoise speech in nonlinear recurrent neural networks.

Suppose an \( m \) dimensional discrete signal \( s(p) \), being trained to an auto-associative neural network, the structure of which is shown in Fig. 3. This is a feed forward network with a bottleneck layer with \( k \) neurons where \( k \ll m \). Usually the values extracted in this layer are considered to be related to principal components of the signal. Here, the order of samples is not concerned, that is, \( s(p) \)'s are considered as \( P \) samples in the input space which are trained to the network. At first the activation function of neurons is supposed to be hard limit step function. In such a network, each neuron forms a hyperplane in its input space (the output space of the previous layer) and the input space is quantized by these hyperplanes. Every area constructed by quantizing hyperplanes is indicated with a unique binary code, i.e. if a sample in an area is affected by noise but it is still in the same area, the changes will not come into sight in the next layer (Fig. 4). This quality is gained by generalization of one point to an area (interpolation).
Practically, we decompose the input signal to its components using a series of basis functions \( \varphi_j(x) \). Here \( \varphi_j(x) \) is a hyperplane corresponding to \( j \)th neuron which would be fired (1) or not fired (0).

In this case, some important issues must be considered: first, the position of each hyperplane should be determined so that the input sample would be placed in the center of the area to be able to tolerate the maximum disturbance. Second, the hyperplanes should be placed in a way that they can discriminate all samples in the input space and this discrimination in the last layer leads to reconstructing the input samples with the least error.

Now, the activation functions of neurons are supposed to be a soft nonlinear function such as sigmoid, for instance. Therefore, the back-propagation algorithm can be used to train the network and the boundaries will be soft and fuzzy. The output of each neuron is a continuous value dependent to the position of the \( \Sigma(p) \) in the input space and each input sample \( \Sigma(p) \) has an indication in each layer with these neurons.

At first we suppose that only one sample is trained to the network with soft activation function. Only one neuron in the hidden layer is enough to indicate this sample. It was observed in some simulations that for more neurons their output will become equal after training.

In this case, the soft (fuzzy) hyperplanes are set in a way that the network achieves better representation for the sample and the sample is reconstructed in the output space with nearly zero error. Besides, a cluster will be made around the attractor which is constructed by soft hyperplanes. The input space is interpolated with hyperplanes and every other point \( \Sigma'(p) = \Sigma(p) + \Pi(p) \) (a noisy pattern) in this space will be projected on this unitary component. However, the value of hidden layer neuron for this noisy pattern is less than that for the original one. Due to the interpolation realized by the unitary kernel function \( \phi(x) \), this value is dependent on the distance and similarity between the noisy sample and the original one:

\[
\exists(p) \Rightarrow \phi_{\text{max}}(x)
\]

\[
\exists(p) + \Pi(p) \Rightarrow \phi(x) < \phi_{\text{max}}(x) \Rightarrow \hat{x}(p) + \hat{\Pi}(p)
\]

Corresponding to the original pattern here named \( \exists(p) \), the kernel function \( \phi(x) \) will be maximum: \( \phi_{\text{max}}(x) \). Now, if the sample is added to noise, \( \exists(p) + \Pi(p) \), the value of \( \phi(x) \) will be less than \( \phi_{\text{max}}(x) \). This value of hidden layer results in an output which can be considered as \( \hat{x}(p) + \hat{\Pi}(p) \) where \( \|\hat{\Pi}(p)\| < \|\Pi(p)\| \), that is, the noise is reduced. The noise reduction happens because there is no unit in the network to represent the nonlinear principal components of the noise, but such a unit exists for the pattern. So, the noise will be filtered nonlinearly. This output is again given to the input, and by cycling the noise will be reduced more and more. In every cycle, the noisy pattern moves towards the original pattern for one step. The cycling will be continued until the noise value is less than a threshold. The experimental results confirm the applicability of these facts.

Now, suppose that we have \( p \) training samples. In this case, \( p \) different kernel functions, \( \phi_j(x) \), \( j=1,2,\ldots,p \) are needed to discriminate the samples with soft boundaries. If the network is trained properly, so that the soft hyperplanes can be positioned correctly, then we have \( p \) kernel functions, \( p \) attractors corresponding to \( p \) patterns, and \( p \) basins of attraction. This structure can nonlinearly filter the added noise to the samples. The most important concern is a training method by which the kernels and soft hyperplanes be formed correctly and in best locations in the space.

If this structure is used to learn the speech signal (the trajectory of representation vectors), actually the adjacent sampled points in the trajectory are the training patterns of the network. By training these points to the auto-associative neural network, point attractors will be formed and the network will learn the trajectory as discrete points.

The approach of this paper is training the trajectory of a desired signal to a recurrent neural network as a continuous attractor. Then the knowledge stored in this network is applied to nonlinear denoising of the signal. The attractors of this network have the ability of attraction of every noisy or changed pattern, which is still in the corresponding basin of attraction. So, the noise is filtered and undesired variety is removed.

Here, continuous attractors are a set of point attractors which are located in succession and construct one or some continuous trajectories or nonlinear higher dimensional manifolds which are trainable to proper structures of neural networks with appropriate databases. A beneficial characteristic of these structures is the ability of learning nonlinear and complex trajectories or manifolds in many dimensional input spaces and nonlinear filtering of signals due to this learning. Since in many practical tasks, the noise effects and specially the variety effects are nonlinear, the ability of nonlinear manifold learning, is extremely essential for filtering systems (here neural networks).

3. Robust speech recognition using neural networks

3.1. Robust speech recognition

The issue of robust speech recognition is treated in many works, all of which aim to compute the output of a recognition system when the input is disturbed because of some changes like noise. In some studies robust speech recognition is considered as a missing data problem and different structures of neural networks are proposed for robust speech recognition dealing with the missing data [6,14]. Also in recent years Parveen applied a recurrent neural network to estimate the value of missing data in input vectors [20–24]. In this...
structure which is a hybrid of Benjio and Gingras [3], some parts of input pattern are randomly missed. The network is planned to reconstruct and complete the input patterns considering previous patterns, according to the following equation:

$$\tilde{x}(n) = (1-\gamma)x(n-1) + \gamma y(n-1)vf$$  \hspace{1cm} (4)

where $\tilde{x}(n)$ is the reconstructed data at time $n$, $x(n-1)$ is the input data at time $n-1$, $\gamma$ is the reconstruction coefficient, $vf$ is the weights of recurrent connections from hidden layer to the input for reconstruction of missed parts and $y(n-1)$ is the output of hidden layer at time $n-1$.

In this study, a network is designed inspired by Parveen’s model, in which the test data has become noisy with stationary and nonstationary noise. We used the white noise as the stationary one and the sound of crossing cars as the nonstationary one and both of them are applied as additive noise. The recurrent neural network is trained with clean data and for every input pattern the training procedure is repeated until the least classification error and the best estimation of input in the output is achieved. The structure of the recurrent neural network and its training method is discussed later.

3.2. The structure of the recurrent neural network

Since this network is trained with clean data and tested with the noisy one, it should learn to reconstruct the input in the output besides high-quality classification.

At the beginning of this study which is inspired by Parveen’s model, we designed a neural network, with two recurrent connections and utilized it in nonlinear filtering of noisy speech signal and acceptable results were acquired. Recurrent connections are full connections in the structure of the network, one of which connects the previous hidden layer to the next one and the other connects the hidden layer to input layer with a delay unit. The first one notices long term durations and predicts the signal considering its previous values, in the training phase. There are 252 neurons in the input layer and every 14 frames of speech signal are considered to 64, to have enough capacity to discriminate the input patterns according to the quantity of Persian phones.

This network is a nonlinear recurrent network and its recurrent connection in the hidden layer results in noise filtering with ratios $\gamma$ and $1-\gamma$.

The recurrent connection can be transferred to hidden layer as shown in Fig. 6. Since, any nonlinear functions are not used in the output of recurrent layer, Eq. (7) can be written for the recurrent connection to the hidden layer:

$$\tilde{y}(n) = f(\tilde{x}(n)vf)$$  \hspace{1cm} (7)

where $f$ is the nonlinear bipolar sigmoid function and $vf$ is the forward weight matrix connecting input layer to the hidden one. Substituting Eq. (6) in Eq. (7), we have

$$\tilde{y}(n) = f[(1-\gamma)x(n)+\gamma y(n)vf]$$

$$\tilde{y}(n) = f[1-\gamma)x(n)+\gamma y(n)vf]$$  \hspace{1cm} (8)

The product of $vf$ can be transferred to $vf$, and the recurrent connection will be from the hidden layer to itself. This structure, used in the experiments, is shown in Fig. 6. This network consists of input, hidden, and output layers. The recurrent connection is in the hidden layer and relates the value of hidden layer in each epoch to its previous values, in the training phase. There are 252 neurons in the input layer and every 14 frames of speech signal are considered as input patterns [described in Section 4–1]. After implementing many experiments, the number of neurons in the hidden layer is set to 64, to have enough capacity to discriminate the input patterns along with most possible simplicity. 35 neurons are put in the output layer according to the quantity of Persian phones.

This network is a nonlinear recurrent network and its recurrent connection in the hidden layer results in noise filtering with the help of attractor dynamics. So, in network training phase and during the implementation of error back propagation, two errors should be considered: first, the phone classification error, second, the hidden layer input reconstruction error. Fig. 7 shows the network connections and the method of error back propagation. The solid lines are representing the full connections which are
updated in each training epoch. However, dashed lines are showing the paths of error back propagation. Two dashed lines are representing $\delta_z$ and $\delta_y$ in Eq. (1). $\delta_z$ is related to phone classification error and $\delta_y$ is regarding to error between the present and previous values generated in the hidden layer (Fig. 8). The network is trained to classify the phones accurately and recognizes the seventh frame of input in its output. In Eq. (9), $d(n,i)$ is the desired output at time $n$, $z(n,i)$ is the acquired output at time $n$, both of $i$th output node, and $E_{1n}$ is sum of squared errors related to phone classification at time $n$:

$$E_{1n} = \sum_{i=1}^{m} (d(n,i) - z(n,i))^2$$  \hspace{1cm} (9)

The recurrent connection is used to reconstruct the input when the data is noisy. Since, training is done with clean data, the weights of recurrent connection are converged in a way that the network is able to reconstruct the noisy signal because of the recurrent connections. In each cycle, the values of recurrent connections will become closer to the original attractor and the value of the hidden layer being reconstructed by the recurrent connections. In each cycle, the values of hidden neurons are computed according to Eq. (12). So, a very accurate classification is obtained in the output.

The recurrent connection is used to reconstruct the noisy signal because of the attractor dynamics formed by these recurrent connections, and then the recognition task is carried out on the reconstructed signal.

In each training epoch, the hidden layer is updated according to Eq. (8) and then error back propagation is implemented to minimize the difference between the product of hidden layer value in recurrent weight matrix and the product of input values in the input layer weight matrix:

$$E_2 = \sum_{i=1}^{m} \left[ \sum_{j=1}^{m} x(n,j)w_{ji} - \sum_{j=1}^{m} y_b(n,j)v_{ji} \right]^2$$  \hspace{1cm} (10)

Here, $n$ is the number of training input patterns, $y_b$ is the output of hidden layer in the last epoch, and $m$ is the number of hidden layer nodes.

The error signal, back propagated to layer $y$ is computed as follows:

$$\delta_y = (\delta_zW^T + (xvi - \tilde{y}v)vf^T)\gamma(1 - y)$$  \hspace{1cm} (11)

$$\tilde{y} = f((1 - \gamma)xvi + \gamma\tilde{y}_bvf)$$  \hspace{1cm} (12)

In Eqs. (11) and (12), $\delta_z$ is the error signal vector in network output, which is related to the phone classification error, $W$ is the network output layer weight matrix, $\delta_y$ is the vector of back propagated error signal to the hidden layer, $\tilde{y}$ is the output of hidden layer in the present epoch, $f$ is the nonlinear bipolar sigmoid function, $y' = f$ is the derivative of $f$ function, $\tilde{y}_b$ is the output of hidden layer in the previous epoch, and $\gamma$ is the reconstruction coefficient.

It is worth noting that in this structure of neural network and with this method of learning, actually a kind of multi-task learning is used; i.e. the weights of hidden layer neurons are trained so that they can form desired clusters of phones and also generate the values of $xvi$ as desired values in the output of recurrent connections.

In the start of the training phase, value of hidden layer is considered zero. So at first, the hidden layer value is totally constructed with input values. The estimation of input pattern by the recurrent connections will become closer to the original pattern gradually and so it will be more effective in hidden layer values. In Eq. (8), inspired by Parveen and Green’s studies [21], the connections coming from input and output are added with the ratio. The presence of the input pattern in this relation serves to restore the denoised pattern and to some extent guaranties the movement of the input pattern towards the accurate attractor. In other words, with cycling in the hidden layer the filtering of noise is achieved step by step.

In the test phase, the output is computed in the same way as the training phase. However, spending enough time is essential so that the trajectory is able to move in the state space towards the related attractor and the value of the hidden layer being reconstructed by the recurrent connections. In each cycle, the values of hidden neurons are computed according to Eq. (12). So, a very accurate classification is obtained in the output.

According to this functionality of the network, in the training phase, error is back propagated through two paths. The phone classification error is back propagated from the output layer to the hidden one (Eq. (9)) and the weights of hidden layer are updated to minimize this error. In the training phase, the desired output of the hidden layer is, in practice, the output coming from the input layer (Eq. (10)). When a clean signal is trained to the network, the recurrent connection should learn to estimate the accurate values of hidden layer corresponding to the clean signal in the input. The weight matrix of the input layer is updated so that it could provide the recurrent layer with a proper input and an accurate classification could be achieved simultaneously. Actually, these two errors collaborate in weights updating. The experiments carried out using this network are described in the following.

4. Experiments and database

4.1. Database

20 sentences of FARSADAT database [5], uttered by one speaker were used in the primary experiments of this study. 10 sentences are utilized as training data and the other 10 sentences as test data. These sentences contain all the Persian phones and can result in meaningful values. In later experiments, the database is extended to 800 sentences uttered by 10 speakers.

In order to classify the speech units, at first the feature extraction should be implemented. In this study the Logarithm of Hanning Critical Band filter bank (LHCB) is used as speech representation. This method of feature extraction resulted in well recognition accuracies for Persian phones in the previous studies. Then the extracted parameters should be normalized. Here, the
longitude normalization (normalization with respect to mean and variance) is utilized, in which the parameters are normalized to the variance of the entire database. The labels of all speech frames are determined from FarsDat database [5].

4.2. The reference network

In order to evaluate the efficiency of recurrent connections in improving noisy speech recognition and achieve a comparison, introducing a network without recurrent connections is essential. This network is called “reference network” which is a feedforward neural network and is trained to classify the frames of speech data. The structure of this model is shown in Fig. 9. It is a TDNN and in the input layer receives 14 subsequent frames. It recognizes the seventh frame as one of the 35 Persian phones. It has a hidden layer with 64 neurons and the activation functions of all its neurons are nonlinear bipolar sigmoid functions. Error back propagation algorithm is used to train this network and after training, its recognition accuracy is computed for all Persian phone groups on test data. These results are shown in Table 1. The structure of the reference network is the same as that of the main recurrent network without the recurrent connections. The designed networks are expected to show better results compared to the reference model.

4.3. Experiments

At first, the experiments are performed with 20 sentences of Farsdat database in the test and training phase. The network is trained with clean data and tested with noisy data. 5 kinds of noisy data are utilized with signal to noise ratios of 0, 5, 10, 15, and 20 db. Both stationary and nonstationary noises are used. The stationary noise is white noise and the nonstationary one is the sound of passing cars in the street. In the testing phase, the robustness of phone recognition is observed for stationary and nonstationary noise and the performance of the network, to which the trajectory of clean signal is learned as attractors, is evaluated in comparison with the reference network. The results on train and test data are given in the following.

5. Results

5.1. Training the continuous attractors with clean speech of one speaker

At first, both of the reference and the main recurrent network are trained with 10 sentences of FARSDAT database. Then 10 other sentences uttered by the same speaker are added by noise with both stationary and nonstationary noise, with different signal-to-noise ratios and then given to the network. Obtained results are shown in Figs. 10 and 11 and also shown in Tables 1 and 2. The results indicate the performance of recurrent network in both stationary and nonstationary denoising and also in speech recognition that is superior to that of the reference network. For example, for SNR=0%, 20.42% improvement is gained in recognition accuracy.

In later experiments, the reference and proposed networks trained with one speaker sentences are tested with 400 sentences

![Fig. 9. Structure of reference network.](image)

![Fig. 10. Recurrent network in comparison with reference network for stationary noise.](image)

![Fig. 11. Recurrent network in comparison with reference network for nonstationary noise.](image)

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Recognition results for stationary noise.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>Recurrent net.</td>
</tr>
<tr>
<td>0 db</td>
<td>62.007</td>
</tr>
<tr>
<td>5 db</td>
<td>64.5592</td>
</tr>
<tr>
<td>10 db</td>
<td>66.9374</td>
</tr>
<tr>
<td>15 db</td>
<td>69.0835</td>
</tr>
<tr>
<td>20 db</td>
<td>70.0696</td>
</tr>
<tr>
<td>Clean signal</td>
<td>71.0557</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Recognition results for nonstationary noise.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>Recurrent net.</td>
</tr>
<tr>
<td>0 db</td>
<td>59.9768</td>
</tr>
<tr>
<td>5 db</td>
<td>63.1671</td>
</tr>
<tr>
<td>10 db</td>
<td>65.8353</td>
</tr>
<tr>
<td>15 db</td>
<td>68.2135</td>
</tr>
<tr>
<td>20 db</td>
<td>69.2575</td>
</tr>
<tr>
<td>Clean signal</td>
<td>71.0557</td>
</tr>
</tbody>
</table>
which consist of utterances of 40 speakers with different genders, dialects and accents. In this experiment, the performance of the network extremely deteriorates because of the incensement of varieties especially those related to speakers. But the performance of recurrent neural network deteriorates less than that of the reference model. As shown in Tables 3 and 4 and Figs. 12 and 13 the proposed model has an acceptable efficiency dealing with stationary and nonstationary noise, and improves the recognition of stationary noisy signal by 30%, nonstationary noisy signal by 21% and clean signal by 27% in contrast to the reference network.

Table 3
Recognition results for stationary noise (trained with one and tested with 40 speakers).

<table>
<thead>
<tr>
<th>SNR</th>
<th>Recurrent net.</th>
<th>Reference net.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 db</td>
<td>46.9402</td>
<td>17.41</td>
</tr>
<tr>
<td>5 db</td>
<td>50.8273</td>
<td>18.5545</td>
</tr>
<tr>
<td>10 db</td>
<td>53.7041</td>
<td>20.2289</td>
</tr>
<tr>
<td>15 db</td>
<td>55.362</td>
<td>22.3621</td>
</tr>
<tr>
<td>20 db</td>
<td>56.3789</td>
<td>25.2969</td>
</tr>
<tr>
<td>Clean signal</td>
<td>57.0645</td>
<td>30.63</td>
</tr>
</tbody>
</table>

Table 4
Recognition results for nonstationary noise (trained with one and tested with 40 speakers).

<table>
<thead>
<tr>
<th>SNR</th>
<th>Recurrent net.</th>
<th>Reference net.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 db</td>
<td>42.4022</td>
<td>21.7344</td>
</tr>
<tr>
<td>5 db</td>
<td>45.5307</td>
<td>22.854</td>
</tr>
<tr>
<td>10 db</td>
<td>48.5102</td>
<td>24.0663</td>
</tr>
<tr>
<td>15 db</td>
<td>50.9879</td>
<td>25.2886</td>
</tr>
<tr>
<td>20 db</td>
<td>52.8561</td>
<td>26.4049</td>
</tr>
<tr>
<td>Clean signal</td>
<td>57.0645</td>
<td>30.63</td>
</tr>
</tbody>
</table>

These results indicate that the formed attractors in the state space of the recurrent neural network are accountable for both speaker variety elimination and noise reduction. This network shows a high-performance dealing with speaker variety even in the existence of nonstationary noise. This performance is explainable with the superior functionality of continuous attractors learned by the recurrent network. These attractors, which correspond to the training speech patterns, attract the trajectory of speech patterns in the state space initiated by speech patterns of different speakers. In other words, test patterns are initial conditions of the trajectories and training patterns are their equilibrium points. As a result, all variety given patterns are transformed to the clean training ones. This capability of the proposed model in nonlinear filtering of noise and different kinds of varieties is an outstanding feature and makes it an excellent tool for nonlinear signal processing.

5.2. Training the continuous attractors with clean speech of 40 speakers

In this experiment, the reference and recurrent neural networks were trained and tested with a larger database which was gotten noisy with stationary and nonstationary noise, with different signal-to-noise ratios. The learning coefficient and the maximum number of epochs as the termination condition, is the same as the previous experiment. The number of neurons of the hidden layer was increased to 256. The results indicate that the recurrent neural network is able of recognizing the noisy signal, with 0 SNR ratio, by 6% improvement in contrast to the reference network. So the purpose of our work, which was improving the recognition of noisy signals, is achieved. However, the recognition accuracy of the recurrent network on the clean data is about 12% less than that of the reference network. The results can be seen in Figs. 14 and 15 and Tables 5 and 6.
Recognizable quality performance of the reference network is because of the increment of attractors. Since the network is not able to form such a huge number of attractors in its state space, it cannot discriminate the different input patterns. This conclusion confirms the hypothesis, which points out that; human beings do not form as many attractors as the speakers they know. But they transform the utterances of each speaker to their own speech signal and an efficient recognition is achieved after this many to one transformation.

6. Conclusion

In this work, we utilized nonlinear recurrent neural networks and benefited their capability in information retrieval and the characteristics of continuous attractors in denoising speech signal and phone recognition with denoised signals. Our proposed method could improve the recognition accuracy, by 20% in contrast with a reference model. Training the clean patterns to the model causes to form point attractors and the network is able to take each new pattern to one of these attractors. Cycling gives the opportunity of moving towards attractors and robust recognition of input patterns.

In our experiments, the network was trained with clean data and learns to slide down in the state space towards the attractors. For every training input, the network continues to cycle until the minimum error is reached. By this method of training, actually the network modifies the input of hidden layer using the value of this input in the previous step and this modification is repeated until an optimum (the attractor of that input) is reached. So, when a noisy input in given to the network, which is out of its corresponding attractor, the network takes it towards the attractor.

Finally, we increased the test data in order to evaluate the performance of the network in variety reduction, practically. The results show a very low quality performance of the reference network. It recognizes the clean data by 30.6% and noisy data with 0 SNR, by only 17% accuracy. However, the recurrent network improves this recognition accuracies by 30% on noisy and by 27% on clean data. Therefore, it can be concluded that the attractors formed in recurrent neural network, are able to remove the noise and variety simultaneously and the network has a high performance in speaker variety elimination.

Then, in addition to test data increment, we increased the number of training speakers, in order to obtain more phone variety enrichment in the training data. Because all the Persian phone combinations are not present in the speech signal of one speaker in the database. The results show that training huge databases to the reference model, improves its recognition accuracy. 35% improvement is observed in the recognition accuracy of this network in contrast to the case of training the network with the utterances of one speaker. It is also seen that the recurrent network is able to improve the recognition accuracy of noisy signal with 0 SNR, by 6% in contrast to the reference network. So, the proposed network could reach the aim of the work which was an improvement in noisy signal recognition. However, on the clean data the performance of the recurrent network was about 12% less than that of the reference network. This lower quality can be corresponded to the incomplete training and convergence of the network, which may be compensated using enhanced learning algorithms in the future works.

As shown in experiments, it is not necessary to teach noisy utterances to the network. But it is sufficient to form the clean speech subspace by training the network with an adequate amount of speech data of that speaker. So, the continuous attractor should represent the subspace related to the speech data of that speaker. In other words, this continuous attractor is formed by putting together the basic speech components of training speaker each of which play a role in robust recognition of input patterns as a point attractor. It is thought that the high capability of human brain in robust pattern recognition is based on a method similar to this nonlinear method of input patterns dimension reduction.

References


Table 5  
<table>
<thead>
<tr>
<th>SNR</th>
<th>Recurrent net.</th>
<th>Reference net.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 db</td>
<td>51.07</td>
<td>42.052</td>
</tr>
<tr>
<td>5 db</td>
<td>56.2133</td>
<td>49.9528</td>
</tr>
<tr>
<td>10 db</td>
<td>59.5704</td>
<td>58.1146</td>
</tr>
<tr>
<td>15 db</td>
<td>61.3508</td>
<td>65.0971</td>
</tr>
<tr>
<td>20 db</td>
<td>62.1922</td>
<td>69.568</td>
</tr>
<tr>
<td>Clean signal</td>
<td>62.24</td>
<td>74.07</td>
</tr>
</tbody>
</table>

Table 6  
<table>
<thead>
<tr>
<th>SNR</th>
<th>Recurrent net.</th>
<th>Reference net.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 db</td>
<td>41.5592</td>
<td>36.3554</td>
</tr>
<tr>
<td>5 db</td>
<td>44.7457</td>
<td>39.9526</td>
</tr>
<tr>
<td>10 db</td>
<td>47.8776</td>
<td>43.3744</td>
</tr>
<tr>
<td>15 db</td>
<td>50.5623</td>
<td>46.5791</td>
</tr>
<tr>
<td>20 db</td>
<td>52.982</td>
<td>49.6944</td>
</tr>
<tr>
<td>Clean signal</td>
<td>62.24</td>
<td>74.07</td>
</tr>
</tbody>
</table>

This inferior recognition may be related to the deficient training of the network. Enhanced modified training algorithms should be searched for the better convergence of the network. High increment in the speaker variety can be mentioned as the other reason for the issue. To overcome this problem, another network could be designed, as a preprocessor, to normalize the data with respect to the speaker variety.

Low quality performance of the recurrent network is because of the increment of attractors. Since the network is not able to form such a huge number of attractors in its state space, it cannot discriminate the different input patterns. This conclusion confirms the hypothesis, which points out that; human beings do not form as many attractors as the speakers they know. But they transform the utterances of each speaker to their own speech signal and an efficient recognition is achieved after this many to one transformation.


Louiza Dehyadegary received her B.Sc. and M.Sc. degrees in biomedical engineering from University of Isfahan, Isfahan, Iran in 2003 and Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2005, respectively. She currently teaches at the Azad University of Sirjan. Her research interests focus on signal processing, speech processing, biological artificial neural networks and quantum neural networks.

Seyyed Ali Seyyedsalehi received his B.Sc. and M.Sc. degrees both in Electrical Engineering, from Sharif University of Technology, Tehran, Iran in 1982, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 1989 and Ph.D. degree in Biomedical Engineering from Tarbiat Modarres University, Tehran, Iran in 1996. Dr. Seyyedsalehi’s research interests are in the areas of speech processing and recognition, biological and artificial neural networks, neural modeling and linear and nonlinear signal processing. He is now an Associate Professor in the Faculty of Biomedical Engineering, Amirkabir University of Technology.

Isar Nejadgholi received her B.Sc. degree in Electrical Engineering, from Sharif University of Technology, Tehran, Iran in 2001 and M.Sc. degree in Biomedical Engineering from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2004. She is currently studying her Ph.D. program in Biomedical Engineering. Her main research interests are Design and Application of Neural Networks, Computational Cognitive Science, Intelligent Systems, Intelligent Signal processing and Chaos Theory.