Signal Processing and Renewable Energy
June 2023, (pp. 47-68)
ISSN: 2588-7327
eISSN: 2588-7335

Persian Speech Recognition Through the Combination of ANN/HMM

Ladan Khosravani pour [1] and Ali Farrokhi *[1]

[1] Department of Electrical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.

Received: 09-Oct-2022, Revised: 31-Jan-2023, Accepted: 02-Feb-2023.

Abstract

The goal is to create a speech recognition system that is able to recognize Persian speech. Prosodic speech is attributed to the hierarchical structure from speech rhythm and tonal expression to the smallest syllable components and provides important information about trans segmental features such as F0 (fundamental frequency), intensity, and duration, which are crucial for natural sound. Prosodic features are highly language dependent, however, the relationship between linguistic features and prosodic data is not well understood in some languages. While relatively high-performance prosodic generators have been developed for many languages, very limited work has been done on prosodic generators in Farsi. In this article, we first use a simple four-layer RNN to extract prosodic information, then we investigate the hybrid ANN/HMM model for Persian speech recognition. 210 samples of the speech of a male person were collected and after removing the noise, 47 of the samples were manually labeled phonetically. Then, the remaining training samples were automatically labeled and new neural networks (ANN) were created for the final recognition of the three-layer MLP type. Four methods including MEL, MEL derivative, energy, and energy derivative were used to extract features, and the values of each of these four methods were combined and given to the neural network. Then we use the neural network to classify these feature vectors and get the most similar vowels. We give the order of vowels as "observations" to HMMs (which are created based on pronunciations) and then find the most probable HMM (or in other words, the most words) to the input sound and output it. By applying recognition on 99.4% of test data, we even reached 100% accuracy in one case, which is a very favorable result considering the small number of speech data.

Keywords: Artificial Neural Networks, Hidden Markov Models, Discrete Fourier Transform, Vector Digitizer, Linear Predictive Coding, Viterbi Algorithm, Fuzzy Expectation Maximization, Probabilistic Neural Networks, Recurrent Neural Networks.

***Corresponding Authors Email:** ali_farrokhi@azad.ac.ir

1. INTRODUCTION

The topic of speech recognition has been the most important topic of speech processing in the past few decades. It can be informally said that the problem of speech recognition is recognizing (not understanding) words from a dictionary uttered by a speaker. This information should be extracted only from the information in the speech signal and the prior probabilities of the problem. In 1950s, when speech recognition research was in its infancy, many researchers believed that future computer technologies would make speech recognition much easier. Unfortunately, we now see that this assumption was wrong. The problem of speech recognition is a very difficult problem and today there are many problems and unanswered questions have remained unsolved despite many efforts and the existence of many research groups. These problems are related to the increase in the number of words (more than 50,000 words), recognition of continuous speech versus isolated words, the number of speakers and phonetic characteristics of speakers in speaker-independent recognizers (SI) versus simpler speaker-dependent systems, the problem Involuntary pronunciation of words in speech (such as "e" or "ah") or words that are not in the dictionary, resistance to environmental conditions (noises and distortions created on the channel), and many other issues. When transferring from a laboratory to real conditions, many of these issues arise. In laboratory conditions, simulations usually show a high recognition percentage, but in real world conditions, a significant decrease in this percentage is almost certain. Nevertheless, in some applications, speech recognition has given favorable results. These applications include an automatic generation of written text from speech signals, access to databases, human-machine interface, access to remote automated services over telephone lines, and control of machines. The goal of speech recognition is to convert speech into understandable written language for users, here we have considered Persian as the target language. In order to address this issue, we will first distribute the following basic model:

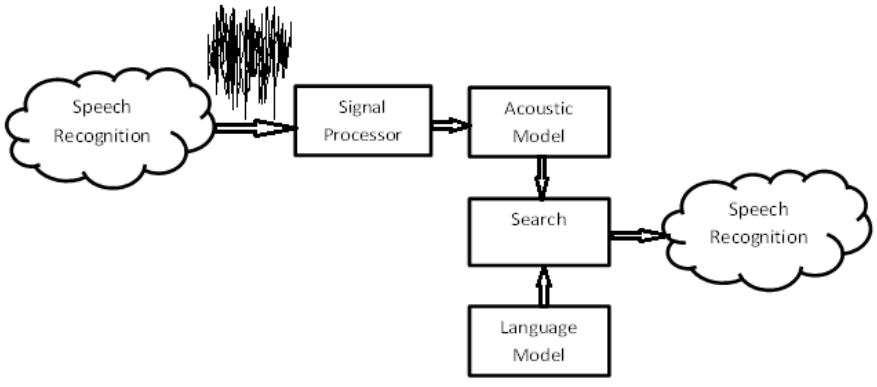

Fig. 1. Simplified block diagram of an automatic speech recognition system [1].

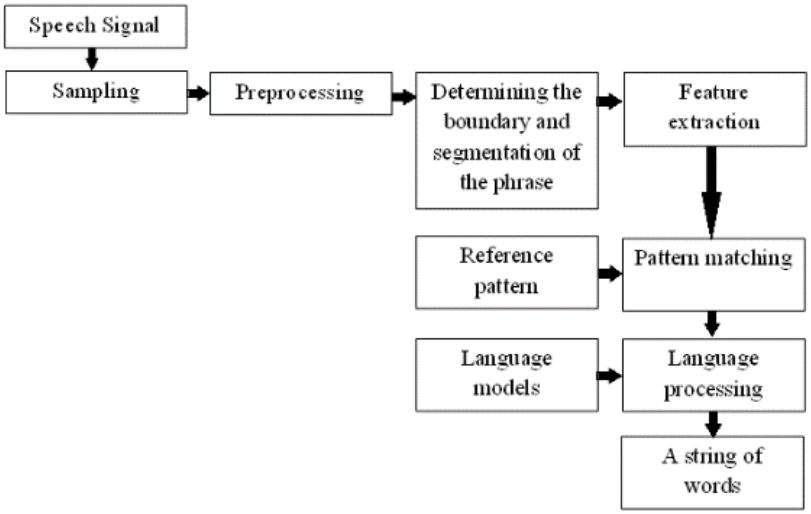

Fig. 2. Block diagram of speech recognition systems (modified from [7]).

Figure 1 shows a summarized speech recognition system. In this figure, there are two sources of knowledge: one is the phonetic model and the other is the language model. A phonetic model is usually stored as a set of templates. The language model or (grammar model) places restrictions on the words that make up a sentence. Each template may represent phrases, words, or smaller units. Finding a suitable unit for speech is one of the important steps in speech recognition. During the training phase, templates are represented as speech parameterizations for each speech unit. Speech parameterization is done by the signal processing part. During the recognition stage, the speech is pre-processed by the signal processor and then the result is submitted to the phonetic model. This is then compared to all existing formats and scored. Then, according to these points and the points of the language model, the best string of words is found (by means of these two sources of knowledge).

A speech recognition system consists of various components. Depending on the type of speech recognition (separate, connected, or continuous), some of these components may not be available in the system or more details may be considered in that system. The following figure shows the block diagram of this system.

2. SPEECH RECOGNITION METHODS

In this article, we describe some of the methods that have already been used for speech recognition. The methods mentioned in this article are batch methods. This means that they are responsible for pattern matching in the speech recognition system. The leader of the methods in speech recognition is the HMM method [7] and most of the works have been done by applying changes to this method. For example, works such as using fuzzy HMMs have been done [2], and also in some research, HMM and ANN methods have been combined for speech recognition [3], [4], [5], and [6].

Of course, other methods have also been used that do not use HMM, such as the pure fuzzy method [8], the pure artificial neural network method [9], [10], [11], [12], methods

based on waveforms [13], or random correlation methods [14], which we take a look at in this article.

In this article, Markov models for speech recognition are mentioned first, and then the method of combining them with neural networks is also stated. Then we will go to pure methods such as fuzzy methods and pure neural network methods and finally, we will describe the combined method of hidden Markov model and artificial neural networks.

2.1. Review Stage

The act of recognition is done by comparing the sound pattern of the word to be recognized with the stored patterns and choosing the word that matches the word the most. The biggest common part of all recognition systems is the signal processing part. The main task of hidden Markov models is to model the speech patterns as a sequence from the vector of observations obtained from the probability function of a first-order Markov chain. The main problem of speech recognition is to get the correct sentence with sound. For the successful application of HMM methods, the following steps should usually be performed [17]:

1- Definition of a set L of the class of sounds for modeling, for example, sounds or words, this set is called set V- $\{V1, ..., VL\}$.

2- For each class, we collect a changeable set (training data set) of pronunciations of words with the corresponding label that we know are present in that class.

3- Based on each training set, we solve the estimation problem to reach the best model λi for each class V1.

4- In order to identify, one should calculate Pr(O|λi) (i=0,1,2,...,L) for each unknown O and identify the speech class that caused O to be produced as class Vi if :

$$Pr(O|\lambda i) = maxPr(O|\lambda i) \qquad (1)$$

According to the aforementioned paragraph, in an HMM model for speech sound patterns, the transition probabilities represent the ordinal structure and duration, and the output distributions in mode describe the changes between different pronunciations. For recognition purposes, it is necessary to be able to obtain the probability of a sequence of observations $O = o1, \dots, oz$ by the M:P(O|M) model [4]. Below is a diagram to show how HMM speech recognition systems work.

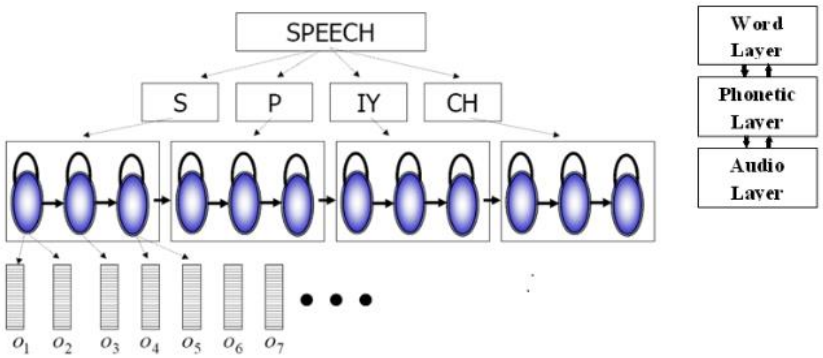

Fig. 3. How to use HMM in speech recognition.

Table 1 to 3. ***Results for Fuzzy HMMs Compared to Normal HMMs [2].***

Error	*Algorithm*
10%	*Discrete Hidden Markov model*
4.5%	*Fuzzy Discrete Hidden Markov model*

Table 1: Recognition Error Rate(%) of discrete Markov

Error	*Algorithm*
8%	*continuous Hidden Markov model*
3.5%	*Fuzzy continuous Hidden Markov model*

Table 2: Recognition Error Rate(%) of continuous Markov

Codebook size	*Recognition Error Rate(%)*	
	HMMs	*Fuzzy HMMs*
32	*4.3*	*3.8*
64	*2.4*	*2.22*
128	*1.64*	*1.6*

Table 3: Speech Recognition Results for 10-command set

2.2. Fuzzy HMMs in Speech Recognition

In [2], the fuzzy hidden Markov model was used. This method is called fuzzy speech and speaker recognition. It is an application of the fuzzy expectation maximization algorithm in HMM. This method is applied in Baum-Welch's Markov model algorithm. In this research, it has been shown that fuzzy HMMs obtain better results than conventional HMMs.

The premise of HMM is that the speech signal can be described as a parametric random process and the parameters of the random process can be accurately estimated. In HMM theory, the famous algorithm is Baum-Welch. It can be considered a kind of expectation maximization (EM) algorithm. The EM algorithm can consider observations as incomplete data. Each iteration of this algorithm consists of an estimation step (E), which is followed by a maximization step (M). EM type algorithms are very simple in terms of implementation and converge uniformly under normal conditions according to the log-likelihood of the observed data model [2]. Fuzzy c-means (FCM) classification is one of the most famous methods of cluster analysis. Many previous fuzzy methods for speech and speaker recognition focused on applying the FCM algorithm (also known as FVQ fuzzy vector quantization) and did not apply the normal vector gradient (hard cmeans). In speaker recognition, FVQ was used to generate speaker models. In HMM-based speech recognition, FVQ makes a soft decision about which of the codewords (average vector) is closest to the input vector and produces an output vector whose components represent the proximity of each codeword to the input. Although these methods are only suitable for applying to discrete HMMs (in this type of HMM, the observations are discrete) and the fuzzy method for continuous HMMs has not yet been described (observations are continuous and modeled by probability density functions). Consequently, finding a

fuzzy method that can be applied to discrete and continuous HMMs should be considered [2] and [15]. A simplifying assumption that is commonly made in modeling language and speech is the Markov property. In this way, it is assumed that the conditional probability distribution of the current event, having all past and present events, depends only on the previous event. A phenomenon called a Markov property. An observable Markov model is a process whose output is a set of states at any instant of time, and each state corresponds to an observable event. A hidden Markov model is a two-way stochastic process with an underlying process that is not directly observable (hidden) but can be observed by another set of stochastic processes that generate a sequence of observations. In [2], normal HMM and fuzzy HMM methods were applied to a database of 8 words and each word was pronounced by 8 people, and a 16-bit sound card was used and sampling was done with 8 KHz. The table below shows the recognition error rate. We can see that fuzzy methods work better than conventional methods.

2.3. Speech Recognition by Artificial Neural Networks

As stated in chapter 1, the speech recognition process can be divided into two stages. The first step is feature extraction where digital speech sampling is performed and digital audio signals are analyzed. Spectrum analysis is used to analyze and extract the content of signals. The next step is to identify sounds, a group of sounds, and words.

This step can be done in many ways (such as DTW), hidden Markov model, artificial neural networks, expert systems, and a combination of the above methods. Hidden Markov models in speech recognition are one of the famous statistical methods, but artificial neural networks have also gained a foothold in signal processing. So far, most of the used networks have been of the forward type such as MLP and RBF radial basis function networks [10] and [16]. These networks are useful for many other tasks other than classification. In speech recognition, the goal is to classify audio signals. In [12] it is focused on probabilistic neural networks (PNN). These networks are another type of feedforward networks that provide a general technique for solving pattern classification problems.

2.3.1. Probabilistic Neural Networks

PNN is a type of neural network that is directly derived from the Bayes method for pattern classification in training and is able to decide complex decision boundaries that are the estimation of Bayes optimal ranges. Additionally, decision boundaries can change online when new data arrives. Although the key computational feature of PNN is the ability to estimate the probability distribution function by the bursting window method based on the data sample, which is a non-parametric density estimation method. The network is trained by providing data that is known to belong to which class. It then uses the training data to develop a distribution function, which it uses to estimate the likelihood of an input pattern among several given classes. This process can be combined with the prior probability to obtain the best category for the input pattern [18]. The following figure shows a diagram of a PNN for binary classification:

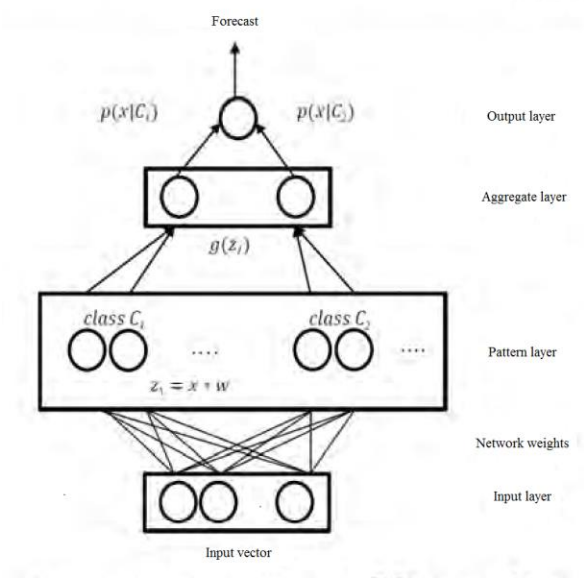

Fig. 4. Structure of a binary category (changed from [12]).

This network consists of four layers. These layers include input layer, pattern layer, addition layer, and output layer. The input layer receives an M-dimensional vector called X and sends it to all nodes in the template layer. The nodes in the model layer are divided into different categories according to their output class. These nodes perform a non-linear (exponential) transformation in the linear multiplication operation:

$$Zi = x.wi \quad (2)$$

$$G(Zi) = exp\left(\frac{Zi-1}{\sigma^2}\right) \quad (3)$$

where Wi is the weight vector of pattern node i and G(Zi) represents the Gaussian kernel function. The output sum nodes sum the pattern nodes together and obtain the probability density function (pdf) of each class P(x|ck). The output layer is used to implement a specific decision rule to predict the output.

PNN training is done in such a way that a pattern group is generated and connected to the target class sum node, and the input vector is assigned as a weight vector. In the N-class problem, N sum nodes are created (for every other class). In addition to the Gaussian kernel estimator, other kernel function models can be used, such as Euclidean distance, city-block distance, and point multiplication function that can work on inputs of different lengths. In general, the PNN algorithm is easy to implement and its training time is much less than the backpropagation and error method. Training in PNN is instantaneous and requires only one pass over the data. By providing enough data samples, the PNN algorithm is able to converge to complex and non-linear decision planes represented by Bayesian classification. Most importantly, it is able to change decision boundaries online (as new data arrives); As a result, PNN can be used for incremental training and real-time classification applications [12] and [18]. In [12], PNN was applied to classify vowels {a,e,i,o,u}. 200 men and women were asked to pronounce these vowels. All sounds were saved in Wave files. Instead of Fourier transform, Dalbechhis waved was used to filter audio signals.

Neural network is implemented by toolbox in MATLAB software. The results of the simulations are reported as follows:

2.3.2. Recurrent Neural Network

Prosodic speech is attributed to the hierarchical structure from speech rhythm and tonal expression to the smallest syllable components and provides important

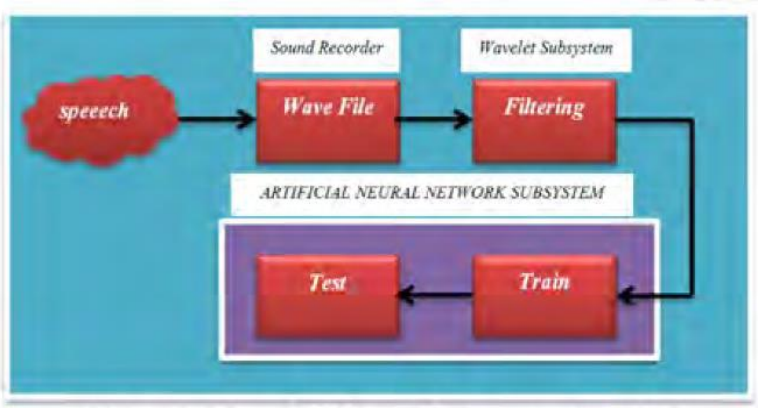

Fig. 5. How to use PNN in speech recognition.

Table 4. ***Results of simulations using PNN in speech recognition.***

Spread	Overall Accuracy (%)	Individual Accuracy(%)				
		a	*e*	*i*	*o*	*u*
0.001	98.4	99	100	100	99	94
0.002	98.0	98	99	100	98	95
0.003	96.2	97	98	96	97	93
0.004	92.6	95	95	90	96	87
0.005	88.8	92	91	86	89	86
0.006	84.2	91	85	82	80	83
0.007	79.2	86	78	77	76	79
0.008	76.0	82	75	74	73	76
0.009	72.6	81	70	71	70	71
0.010	70.4	78	68	68	69	69

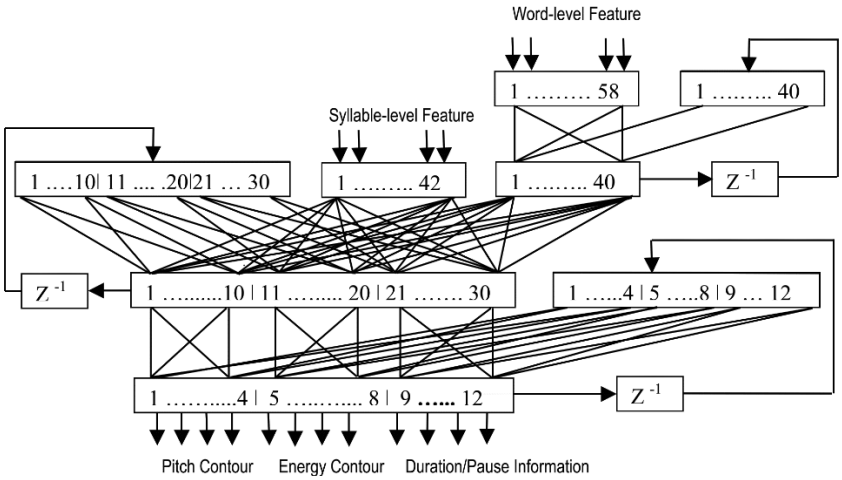

Fig. 6. Structure of the neural network generating prosodic data.

information about trans segmental features such as F0 (fundamental frequency), intensity, and duration, which are crucial for natural sound. The dynamic of prosodic features is highly language dependent, however, the relationship between linguistic features and prosodic data is not well understood in some languages. Accordingly, recent findings suggest the use of data-driven methods to generate prosodic information using neural networks or statistical models to achieve naturalness and fluency in automatic speech synthesizers [19]. While relatively high-performance prosodic generators have been developed for many languages, very limited work has been done on prosodic generators in Farsi. In this method, unlike some other approaches [24-02], all the main prosodic parameters are simultaneously generated by RNN. Furthermore, in contrast to most full prosodic generators, which are mostly developed for tonal languages [19], we do not use any complex syntactic or grammatical structures, such as main and subordinate phrases [25], in this method. In fact, very simple inputs at the word and syllable levels are used as linguistic features, and the prosody rules are all embedded in the network weights, which are learned automatically.

As shown in Figure 7, we use a four-layer RNN to train the network. About 1000 Persian sentences are used and segmented using automatic segmentation.

The method we previously developed at phoneme, syllable, and word levels is presented in [26]. The prosodic data of each syllable, which include pitch contour, energy line, duration and place of vowel onset, syllable length, and pause duration, are extracted from the signal. The inputs and outputs of the neural network are summarized in Table 5.

Table 5. ***Inputs and outputs of the neural network and the number of associated nodes in the RNN.***

	Input size	Definition	Symbol
In puts (word layer)	5	Length of current word in terms of syllable counts	L-W-0
	5	Length of next word in terms of syllable counts	L-W-1
	10	Number of words in sentence	NUM-WORD
	10	Position of current word in sentence	POSITION-WORD
	4	Sentence type	TYPE_SEN
	12	POS of current word	POS (Wi)
	12	POS of following word	POS (W_{i+1})
Inputs (syllable layer)	2	Punctuation marks after current syllable(: / ,)	INTERNAL PM
	6	Type of 1st consonant in current syllable	I-0
	6	Type of 1st consonant in next syllable	I-1
	6	Type of vowel in current syllable	V-0
	6	Type of vowel in next syllable	V-1
	6	Type of 2nd consonant in current syllable	SC
	6	Type of 3rd consonant in current syllable	TC
	4	Syllable's position (first, middle, last, monosyllable)	POS-SYLAB
Out puts(Syllable)	4	Log-Pitch freq. curve-Legendre coefficients	PITCH
	4	Log-energy curve-Legendre coefficients	ENERGY
	1	Pause duration before current syllable	PAUSE
	1	Length of syllable	LEN-SYL
	1	Length of vowel	LEN-VOWEL
	1	Place of vowel onset in current syllable	START-VOWEL

Table 6. ***Classification of consonants in RNN.***

C1	p, t, k	C4	s, sh, ch, f
C2	b, d, g	C5	z, j, zh, ch, v
C3	m, n, l, r, y	C6	h, x, e'

Different categories of sentence type (such as accusative, interrogative, imperative, and exclamatory) with TYPE_SEN are shown in Table 5. The input word layer and the first hidden layer work with the word, synchronized with a clock, to represent the phonological states of the current word in the prosodic structure of the text to be synthesized. The second hidden layer uses syllable level inputs along with the outputs of the previous layer to generate the desired prosodic parameters. Denoting vowels with V and consonants with C, Persian syllables can be found in one of the forms V, VC, CV, CVC, and CVCC [27]. To reduce the number of input nodes of the second layer that contain syllable data, Persian consonants are placed in six groups according to Table 6. Each of the six Persian vowels, including "A" (as in "Ran"), "E" (as in "Takht"), "W" (as in "Bishtar"), "E" (as in "Sheep"), "A" " (as in "car"), and "u" (as in "dub"), are checked separately.

The syllables are grouped according to their place in the input words: first, middle, last, and monosyllable (POS-SYLAB in table 5). In addition, four Legendre coefficients represent the log F0 curve (for voicing) and the log energy curve of each syllable to reduce the number of network output nodes [19].

In this system, all output values are evaluated linearly between zero and one, and the error function is RMSE (root mean square error). The data set for training the network was obtained from the segmentation of about 1000 Persian sentences uttered by a native Persian man, containing about 10000 syllables with an average speed of 4 syllables per second. Sentences were selected from a collection of everyday conversational speech in different types: declarative, interrogative, exclamatory, and imperative, with positive and negative modes, sampling at 10 kHz with 16-bit resolution.

To evaluate the performance of the system, 50 sentences (780 syllables) out of the training set have been used. The resulting RMSE of the trained network for the test set is as follows (see Table 5): LEN-SYL 40.8 ms, LEN-VOWEL 36 ms, PAUSE 31 ms, F0 18.7 Hz, and ENERGY 2.1 db, which showed slightly higher performance, compared to the prosodic generators reported in [19] and [24]. As seen in Figure 6, one of the main reasons for such superiority is the use of syllable energy lines in this network, instead of ignoring the syllable energy in [24] or taking a simple scalar as the syllable energy used in [19]. Moreover, compared to the network introduced in [19], the target network has fewer feedback connections, which reduces the system complexity without compromising performance.

Here we used a simple four-layer RNN that simultaneously generates prosodic information including pitch and energy contours, syllable length, and vowel, onset location.

The results show that this system performs better than some more complex prosodic production systems in terms of error in predicting prosodic parameters.

3. HYBRID SPEECH RECOGNITION SYSTEMS

Although HMMs have a standard and solid theoretical framework, ANN/HMM hybrid systems are a new field and do not have a

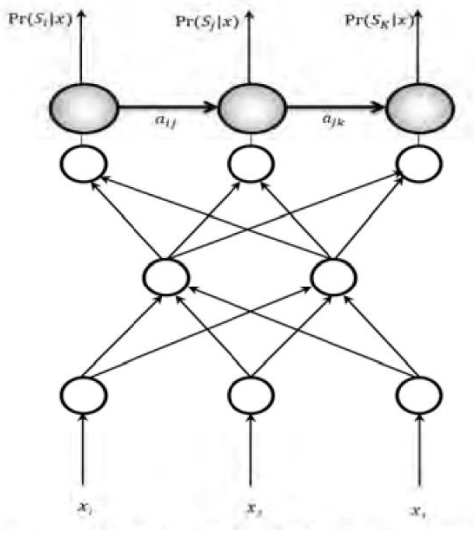

Fig. 7. A hybrid ANN/HMM basic structure where a three-layer ANN prior probability estimates the states of an HMM with observations X=(X1,X2,X3).

unified mathematical expression. Until now, different hybrid structures and algorithms have been presented. In this section, we discuss four main categories:

- Early attempts: Early methods (late 1980s and early 1990s) tried to imitate ANNs by means of HMM structures [36], [37], some of them were helped by dynamic programming algorithms in ANN itself [38 and 39]. These methods strengthened the idea that ANNs can be used in a very optimal way for speech recognition, but they could not overcome the limitations of ANNs by directly simulating HMMs.
- Using ANNs to estimate the prior probability of HMM states:

Some hybrid ANN/HMM systems assume that the output of an ANN is fed to a continuous HMM [32], [33], [34], [37]. The structures presented by [41], [42] are based on the description of the probabilities of ANN outputs. Each ANN output unit is trained to make a non-parametric estimate of the posterior probability of a continuous HMM state given the phonetic observations (and perhaps the prior state). This category is one of the basic categories of hybrid models which had a great impact on later methods. The strengths of this method are the ease of implementation (because its training is a combination of back propagation and standard Viterbi) as well as the training differentiation condition which improves the recognition efficiency. The weakness of this method is due to the lack of a global optimization method and also due to the need for very large ANN structures (perhaps millions of weight connections) for training in complex problems (for example, speaker-independent continuous speech recognition with a high number of words).

- Global optimization: ANN and HMM are usually trained separately but methods for training them together have been presented. In [31], [40], the application of ANN is for

transforming phonetic features into more effective observations of HMM. This is done based on the global optimization of all parameters of the combined system. Apart from the feature extraction, what is more important among them is the introduction of a decomposable training method whose purpose is to optimize a global conditional function. In recent years, other researchers have proposed other global optimal combinations. In [43], you can find a possible survey on hybrid structures with global differentiation training algorithms.

- Application of ANN as vector digitizer of discrete HMMs:

Discrete HMMs accept a limited alphabet. Since this limitation is not practical in speech recognition, a digitization operation must be performed on a phonetic feature space. Instead of using standard clustering algorithms, unsupervised ANN can be used as a vector digitizer. In this category of combinations, ANN and HMMs are used separately for training. Here, the lack of a global optimization method is compensated by reducing the complexity of the whole machine (in particular, it is easier to use discrete HMMs than continuous HMMs) and also by high efficiency (almost equal to continuous HMMs. [44], [45] and [46].

- Other methods: In addition to the structures introduced above, researchers have presented other hybrid systems that are based on certain combinations between HMMs and ANNs. These methods do not belong to any of the above categories and usually focus on the two concepts of scoring and word location detection. In [28], [29], and [35] ANN outputs are interpreted as "scores" that are used in a dynamic programming algorithm to perform tuning and segmentation. ANN can also be used to score the best N hypotheses generated by HMM [47]. However, these methods cannot be assumed to be the same.

3.1. Using Anns to Estimate the Prior Probability of HMM States

The method that is being investigated now is based on the second category. As a result, we will discuss this category in more detail in this section. In [41], Borlard used ANN/HMM combinations to estimate the prior probabilities of the states of HMMs. In these methods, the ultimate goal is to maximize the prior probability of a Markov model Mi (left to right) with a sequence of phonetic observations. The previous probabilities can be written as follows:

$$Pr(M_i|X) = \sum_{q_i{}^l} Pr(q_i{}^l . M_i|X) \quad (4)$$
$$= \sum_{q_i{}^l} Pr(q_i{}^l|M)\ Pr(M_i|q_i{}^l . X)$$
$$= \sum_{q_i{}^l} Pr(q_i{}^l|X)\ Pr(M_i|q_i{}^l)$$

Here it is assumed that Mi must have Q state S1,...,Si and the sequence of phonetic observations X=(X1,...,XL) has length L. The value of Pr does not depend on the vowels (the sequence of observations X), but only on the higher-level decisions that are included in the definitions of the models. And as a result, it can be calculated separately. By successively applying the conditional

probabilities, equation 5 can be written as follows:

$$Pr(M_i|X) = \sum_{q_i{}^l} Pr(q_l|X)\, Pr(q_l|X.q_1)\, Pr(q_l|X.q_2,\dots,q_{l-i}) Pr(M_l|q_i{}^l) = \sum_{q_i{}^l}\{\prod_{l=1}^{L} Pr\left(q_l|X, q_l{}^{l-i}\right) Pr(M_l|q^l,)\} \quad (5)$$

In the mentioned research, feedforward neural networks are used to estimate the prior probabilities of states (given the observations and sequence of previous states). For this purpose, the advantages of ANNs such as their discriminability and their ability to estimate prior probabilities (when trained by "back-propagation" with MSE condition) are used. The basic structure is shown in Figure 7.

Of course, an approximate version of equation 6 is used:

$$Pr(M_i|X) = \sum_{q_i{}^l}\{\coprod_{l=1}^{L} Pr(q_l|X_{l-x},\dots,X_{l+k},q_{l-1})\} \; Pr(M_l|q_i{}^l) \quad (6)$$

In fact, the network is trained to estimate the transition probability. This is the probability of having a sequence of phonetic observations. The previous probabilities can be expressed as Pr(ql|Xl-x,…,Xl+k,ql-1) having 2K+1 elements of the phonetic vector Xi-x,...,Xi+k (a window of length K around the current observation of vowel X1 is located) and the prior state is obtained. This is achieved by using a "back propagation" method to train the MLP (which has an output node to estimate the prior probability of each state). Singer and Lipman [48] used radial basis function networks [49] instead of MLP as the Bayesian probability estimator. The resulting combination is used in the recognition of isolated words.

Robinson and others [30] and [50] developed the Borlard method. In this way, a recurrent network is used instead of a static MLP to estimate the prior probabilities of the states. Their system is called Abbott and it works in the form of continuous speech, independent of the speaker and with a high number of words (more than 10 thousand words). Then this system was developed as follows:

1- A combination of neural models: Different recurrent networks are trained on different phonetic features (MEL and PLP coefficients). In addition, "return forward" networks are provided. As a result, the probability estimates are used in parallel. These models are then combined either linearly or logarithmically.

2- Introducing a "search space pruning": Pruning of the search space is done according to each input frame based on the estimation of the probability of monotones. Paths that contain vowels and whose prior probability is less than a certain value are quickly pruned.

Another method that is similar to Borlard's method is presented in [32]. Here, an HMM is not used as a detector, but a Viterbi algorithm is used on the score of the states (which are the outputs of the three-layer MLP output layer). Another method is described in [47] in which ANNs are used to estimate the prior probability of states. In this method, the output of networks is interpreted as a discriminant function (for example, by applying Bayes' rule and assuming that all states of the model share a prior probability). In this method, MLP is used for "mode recognition". These systems rely on the discriminability of feed-forward neural networks. In this method, networks are

placed sequentially with the aim of modeling and recognizing isolated words, and then the most similar word is selected by dynamic programming. In this system, words are placed in order of grids from left to right (like the corresponding HMM). Each network has an output unit for categories (words). The input phonetic feature is given to the networks and these networks calculate a score for each class. A decoding engine based on the Viterbi algorithm is run on the points obtained in this way. The training is carried out in the following two stages:

1- Each MLP is executed by the back propagation algorithm on labeled sequences (for example, in terms of a phonetic observation, it represents the equivalent vowels).

2- The improvement of MLP parameters is by means of an optimization method based on the overall slope of the probability and back propagation of the derivatives within the network layers.

4. IMPLEMENTATION OF ANN/HMM HYBRID MODEL FOR SPEECH RECOGNITION

To implement the combined ANN/HMM model for Persian speech recognition, CSLU toolbox has been used. This toolkit performs low-level operations using C code and high-level operations using a hard-coded language called TCL. In this way, the balance between speed and flexibility is achieved which cannot be achieved if each of these languages were used separately.

The steps of frame-based speech recognition by ANN/HMM structure are shown in Figure 8.

The steps are as follows:

- We divide the speech wave form into frames. A frame is the smallest part of speech that has the same number of wave form samples. It is usually assumed that the frame size is 10 milliseconds.
- For each frame, we calculate features. These features represent the spectral envelope of speech in each frame. It may also represent part of the features of neighboring frames.
- By the neural network, the feature vector is assigned to one of the vowel categories. When the neural network is used to classify all the frames, we get a probability matrix that has F columns and C rows where F is the number of frames and C is the number of categories.
- Using the matrix of possibilities, a set of pronunciation models (the same as HMMs), previous information about the duration of a sound category, it is possible to determine the most similar word in speech form by means of the Viterbi algorithm.

The steps of neural network training to classify vowels are as follows:

- Specifying the categories of vowels in which the neural network should classify them.
- Finding examples of these categories in training data.
- Repeated execution of neural network training algorithms. The output of each neural network is estimated from the probabilities of the specified categories.
- Selection of the best trained neural networks by applying it to data other

than training and testing data. In a hidden Markov model, this neural network is used to see which one gives the least error. It is used for test data in this network.

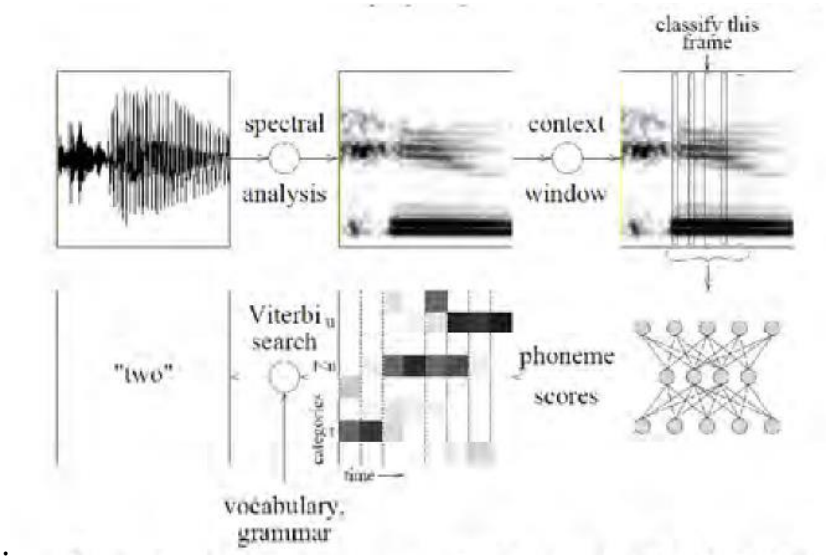

Fig. 8. Steps of frame-based speech recognition by ANN/HMM hybrid structure.

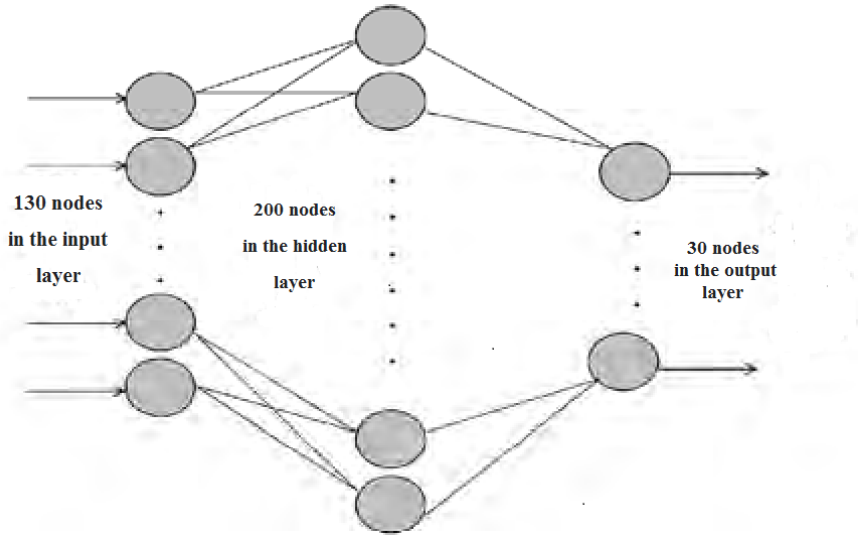

Fig. 9. The structure of the used neural network.

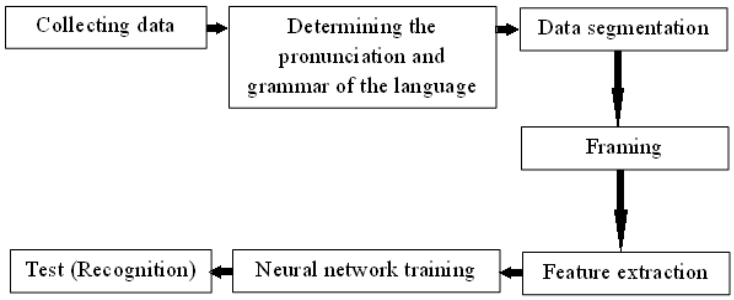

Fig. 10. Block diagram of ANN/HMM recognition training.

As you can see in Figure 8, the spectral features are calculated every 10 milliseconds and fed to a neural network for classification. Neighboring frames can also be given to the neural network. In this implementation, 11 MEL frequency spectral coefficients and 12 MEL delta coefficients and energy parameters and its derivative are also used for feature extraction. As a result, 26 coefficients are extracted for each frame. In calculating the MEL coefficients the subtraction of the Campestral mean (CMS) was used in which all frames were used to calculate the average. Due to the fact that the features of four frames around one frame are also extracted, in total the feature vector has 130 elements. All these elements are stored and then given to the neural network one by one.

The number of nodes in the hidden layer is considered variable, and all the experiments were performed with the number of 60, 100, 160, and 200 nodes in the hidden layer.

The stages of ANN/HMM recognition training are as follows:

The digital data is the speech of a male person so 210 sentences containing 501 words were recorded by the microphone. The sampling rate is 8 kHz and the audio files are saved by PCM. After this step, the noise reduction operation (by 30db) was performed by Wave Pad V3.12 software. This software uses the spectral subtraction method ([1] and [2]) to reduce noise. So, the text (in txt format) which includes the sentence that was played in a wav file was recorded with the same audio file but with txt extension. In the next step, which is one of the time-consuming steps of training, the vowels of all audio files were saved individually in files by the marking software available in all CSLW tools. The act of labeling is done in such a way that it should be determined what vowels are being played in a time period (1.10 to 1.12 seconds). This operation was performed for 47 training data. Using these 47 labeled data, a neural network was created, which of course, was not very accurate (due to the small number of training data). Then this training network was used for automatic labeling of all training data (168). As a result, the time-consuming and tedious work of manual labeling was converted to automatic labeling by means of a pre-trained neural network. This is very important in high data amounts. Because there is a high amount of data, especially in the case of continuous speech, it is not possible to manually label the training data.

Data selection: In this step, we divided the data into three parts. The first part was related to education data where we allocated 84% of the data or 160 data to education. We considered 32 samples (15%) for the test data. The test data is used to calculate the recognition error percentage. The next category is development data. Development data is used to select the best neural network. We allocated 10 samples (5%) to this section.

Framing and feature extraction: In this step, we first framed the training data. Audio data was divided into 10 ms frames. The feature vectors of these frames were calculated and stored individually.

Neural network training: In this step, the feature vectors were given to the input of the neural network and according to the labeling of the frames, the network was trained (back propagation method). We

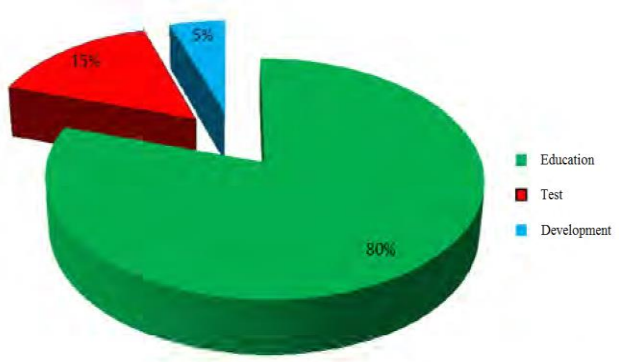

Diagram 1. Data segmentation.

Table 7. ***Word recognition accuracy.***

	Hand Labeled (Number of Sentences:47) (Number of Words:135)		Automatically Tagged (Number of Sentences:168) (Number of Words:501)	
Educational Data	88.15%	NN60	97.41%	NN60
	92.59%	NN100	99.60%	NN100
	93.33%	NN160	99.40%	NN160
	94.07%	NN200	99.40%	NN200
Test Data (Number of Sentences:32) (Number of Words:77)	71.79%	NN60	94.74%	NN60
	78.21%	NN100	98.70%	NN100
	85.90%	NN160	98.67%	NN160
	82.05%	NN200	100%	NN200

Table 8. ***Sentence recognition accuracy.***

	Hand Labeled (Number of Sentences:47) (Number of Words:135)		Automatically Tagged (Number of Sentences:168) (Number of Words:501)	
Educational Data	70.21%	NN60	92.26%	NN60
	78.72%	NN100	98.21%	NN100
	82.98%	NN160	98.21%	NN160
	82.98%	NN200	97.62%	NN200
Test Data (Number of Sentences:32) (Number of Words:77)	37.50%	NN60	85.25%	NN60
	53.13%	NN100	93.75%	NN100
	65.63%	NN160	93.80%	NN160
	59.38%	NN200	100%	NN200

trained the neural network for 20 rounds and saved the weights of each round. So we chose the best round of the neural network according to its error on the development data.

Test: Now having the neural network, we apply test data to it. We used 4 neural networks with different number of nodes in the hidden layer. As you can see in Table 7, the number of nodes is 60, 100, 160, 200. Below the column labeled "t", the neural network was trained on 47 of the manually labeled samples. As you can see, the percentages are not very acceptable. It was

decided to automatically label the rest of the training data by this trained neural network. The result was that we had 168 training samples on one side that were automatically labeled. In the next step, like the above procedure, another recognition based on these 168 data samples was created. As you can see in Table 7, the percentages have improved a lot.

Once we applied the recognition to the training data (first row) and once again we applied the recognition to the test data that the system had not seen before (not available in the training data).

The test is that we first frame the input sound for recognition and then extract the feature vectors (for each frame). Then we use the neural network to classify these feature vectors and get the most similar vowels. We give the order of vowels as "observations" to HMMs (which are created based on pronunciations) and then find the most probable HMM (or in other words, the most words) to the input sound and output it.

5. CONCLUTION

As stated in the first chapter; The three issues that are discussed in speech recognition are: continuous speech, dependence on the speaker, and a large number of words which contrasts the tasks as follows:

- Recognition of continuous speech versus isolated words.
- Speaker-independent vs. speaker-dependent speech recognition.
- Speech recognition of a large number of words versus a fixed number of words.

In this article, the simplest mode is reviewed; That is, the recognition of isolated spoken words, depending on the speaker with a limited number of words. In order to reach the highest and most difficult level, i.e. recognizing continuous speech independent of the speaker with a large number of words, some things must be considered. In order to deal with the problem of continuous speech, things like segmentation and pauses should be considered. In this implementation, these things have been observed, so this implementation can be used; with the difference that to reach a high percentage, the method of sounds dependent on the text should be used [1]. To achieve a speaker-independent system, one should focus on the feature extraction stage; In this way, he extracted features that are common among different voices (male and female, etc.). The article's implementation method can be used with modifications to implement speaker-independent systems. To support the high number of words, the same implementation can be developed, this system is able to support several thousand words.

REFERENCES

[1] DJ Kershaw, 1997, Phonetic Context-Dependency In a Hybrid ANNHMM Speech Recognition System the degree of Doctor of Philosophy, University of Cambridge.

[2] MR Tarihi, A Taheri, H. Bababeyk, 2011, "A New Method For Fuzzy Hidden Markov Models in Speech Recognition, E International Conference On Emerging Technologies.

[3] L Cang S Asghar, B Cong 2012, "Robust Speech Recognition Using

Neural Networks and Hidden Markov Models, Proceedings of the international conferences on Information Technology: Coding and Computing, 350.

[4] QAA Alim N Elboghdadly, NMB Shaar, 2009, "HMMANN Hybrids For Continuous Speech Recognition", Eighteenth National Radio Science Conference, 509-516 vol.2

[5] D. Albesano, RGamello and F.Manu, "Hybrid HMMANN for Speech Recognition and Prior dass Probalities", Proceedings of the 9 International Conference on Neural Information Processing (ICONPOZ, Vol.5

[6] ETrentin and M Gori, November 2012, "Robust Combination Of Neural Networks and Hidden Markov Models for Speech Recognition', Vol.14, NO.6

[7] B H Juang and L R Rabiner, August 1991, "Hidden Markov Models for Speech Recognition Technometrics Vol. 33, NO.3

[8] T. Z Peng, Y. Woo, 1999, "Fuzzy Speech Recognition", Neural Networks.

[9] S Yang, MJ Er, Y. Gao, 2001, "A High Performance Neural-Networks-Based Speech Recognition System".

[10] A Ahad A Fayyaz T. Mahmood, 2011, "Speech Recognition Using Multilayer Perceptron", Students Conference, 102, Proceedings. IEEE, 103-109 Vol. 1.

[11] M.M.El Choubassi, H. E. El Khoury, C. E. Jabra Algabra, J. A Skaf and MA Al Alaoui, 2003, "Arabic Speech Recognition Using Recurrent Neural Networks", Signal Processing and Information Technology, 2003.

[12] C P. Lim SC Woo, AS Loh, R Osman, 2009, "Speech Recognition Using Artifical Neural Networks, Preceedings of the first International Conference on Web Information System Engineering (WSE00)- Volume 1- Volume 1,419.

[13] B Gamulkiewicz, M Weeks, 2003, "Wavelet Based Speech Recognition, Craitsandsystems 2010.

[14] J Ming and FJ Smith, May 1996, "Stochastic correlation model for speech Recognition".

[15] J. Zeng ZQ Liu, June 2006, "Type 2 Fuzzy Hidden Markov Models and Their Application to Speech Recognition, IEE Transactions on Fuzzy Systems. 14, NO.3

[16] SU Khan, G Sharma, P.RK Rao, 2002, "Speech Recognition Using Neural Networks, Industrial Technology 2000.

[17] M D. Wächter, M M K Demuynck, May 2013, P. Wambacq, R Cools, D.V. Compernolle, 2007, "Template-Based Continuous Speech Recognition", IEEE Transactions on Audio Speech, and Language Processing, Vol.15, NO.4

[18] H. Bourlard "Continuous Speech Recognition: Hidden Markov Models vs the connectionist hops Continuous speech recognition", Signals Systemand Computers1989. Twenty-Third Asilomar Conference on, 331-335.

[19] S. Chen, 1998 S. Hwang, and Y. Wang, “An RNN-based prosodic information synthesizer for Mandarin text-to-speech" IEE Transactions on

speech and audio processing, vol.6, No.3, 226-238, May.

[20] C. Tbler, 1992,“F0 generation with a database of natural F0 patterns and with a neural network," in Talking Machines: Theories Models and Applications. Amsterdam, The Netherlands: Elsevier.

[21] M. S. Scordilis and J. N. Gowdy, 1989, “Neural network based generation of fundamental frequency contours," in Proc. ICASSP, 219-222.

[22] M. Riedi, 1995, “ A neural-network-based model of segmental duration for speech synthesis," in Proc. EUROSPEECH, 599-602.

[23] Y. Hifny and M. ashwan, 2002 “Duration modeling for Arabic text to speech synthesis," 7th International conference on spoken language processing ICSLP, 1773– 1776.

[24] O. Jokisch, H. Ding, H. Kruschke, and G. Streacha, 2002, “Learning syllable duration and intonation of Mandarin Chinese,"7th International conference on spoken language processing ICSLP, 1777-1780.

[25] Y. Sagisaka, 1990, “On the prediction of global F0 shape for Japanese text-to-speech." in Proc. ICASSP, 325-328.

[26] Farrokhi, S. Ghaemmaghami, M, Tebyani, and M. Sheikhan, 2002, “Automatic segmentation of speech signal to extract prosodic information,” Proc. 10th Iranian Conf. Electr. Eng. (Computer Sessions), 424-431, Tabriz, Ira

[27] Y. Samareh, 1995, “Phonetics of Persian Language,"University Press Center, University of Tehran, Iran.

[28] M Sato, 1990, "A Real Time learning algorithm for recurrent analog neural networks", Biol. Cybernet.237-241.

[29] A J. Robinson, F.fallside, 1988, "Static and dynamic error propagation networks with application to speech coding, in: DZ Anderson (Ed), Neural Information Processing Systems American Institute of Physics, New York, Denver, CO, 632-641.

[30] Y Bengio, P.Simard, P. Frasconi, March 1994, "Learning long-term dependencies with gradient descent is difficult", IEE Trans. Neural Networks 5 157-166.

[31] Y Bengio, R De Mori, G Flammia, R Kompe, 1992, "Global optimization of a neural network-hidden Markov model hybrid", IEEE Trans. Neural Networks, 252-259.

[32] MA Franzini, K F. Lee, A Waibel, 1990, "Connectionist Viterbi training a new hybrid method for continuous speech recognition, International Conference on Acoustics Speech and signal processing, Albuquerque, NM,425-428.

[33] P. Haffner, M Franzini, A Waibel, 1991, "Integrating time alignment and neural networks for high performance continuous speech recognition", International Conference on Acoustics Speech and Signal Processing, Toronto, 105-108.

[34] E Levin, 1990, "word recognition using hiden control neural architecture, international conference on acoustics speech and signal processing, Albuquerque, nm, 433-436.

[35] N Morgan, H Bourland, 1990, "Continuous Speech Recognition Using Multilayer Perceptrons With Hidden Markov Models, International Conference On Acoustics Speech And Signal Processing, Albuquerque, Nm, 413-416.

[36] L T. Niles, HF. Silverman, 1990, "Combining Hidden Markov Models And Neural Network Classifiers, Onference On Acoustics, Speech And Signal Processing, Albuquerque, Nm, 417- 720

[37] J Tebelskis, A Waibel, B Petek, a Schmidbauer, 1991, "Continuous Speech Recognition Using Linked Predictive Networks, In RP. Lippman, R Moody, D. S Touretzky (Eds), Advances In Neural Information Processing Systems 3 Morgan Kaufmann, San Mateo, Denver, 199-205.

[38] J S Bridle, 1990, "Alphanets A Recurrent & Neural Network Architecture With A Hidden Markov Model Interpretation, Speech Commun. 83-92

[39] E Levin, R Pieraccini, E Bocchieri, 1992, "Time Warping Network A Hybrid Framework For Speech Recognition", In: J. E Moody, SJ Hanson, R. P. Lippmann (Eds), Advances In Neural Information Processing Systems 4, Denver, Co, 151-158.

[40] Y Bengio, 1996, "Neural Networks For Speech And Sequence Recognition, International Thomson Computer Press London, Uk.

[41] H Bourlard, N Morgan, 1994, "Connectionist Speech Recognition. A Hybrid Approach", The Kluwer International Series In Engineering And Computer Science Vol. 247, Kluwer Academic Publishers, Boston.

[42] N Morgan, Y. Konig S L Wu, H. Bourlard, 1995, Transition-Based Statistical Training For Asr", Proceedings Of Ieee Automatic Speech Recognition Workshop (Snowbird), 133- 134.

[43] F. T. Johansen, 1996, "A Comparison Of Hybrid Hmm Architectures Using Global Discriminative Training", Proceedings Of Icslp, Philadelphia, 498-501.

[44] H Iwamida, S Katagiri, E Modermott, Speaker- Independent Large Vocabulary Word Recognition Using An Lvq/Hmm Hybrid Aigorithm, 1991, International Conference On Acoustics, Speech And Signal Processing, Toronto, 553-556.

[45] D. Kimber, MA Bush, G N Tajchman, 1990, "Speaker- Independent Vowel Classi "Cation Using Hidden Markov Models And Lvq2 International Conference On Acoustics Speech And Signal Processing", Albuquerque, Nm, 497-500.

[46] G Rigoll, 1994, "Maximum Mutual Information Neural Networks For Hybrid Connectionist-Hmm Speech Recognition Systems, leee Trans. Speech Audio Process. 175-184.

[47] G Zavaliagkos, S Austin, J. Makhoul, R Schwartz, 1993, "A Hybrid Continuous Speech Recognition System Using Segmental Neural Nets

With Hidden Markov Models, Int. J. Pattern Recognition Artif. Intell, 305-319. (Special Issue On Applications Of Neural Networks To Pattern Recognition (1. Guyon Ed.)).

[48] E Singer, R P. Lippmann, March 1992, "A Speech Recognizer Using Radial Basis Function Neural Networks In An Hmm Framework", International Conference On Acoustics, Speech And Signal Processing, Vol. 1, Sanfranscisco, 629-932.

[49] MJD Powell, 1987, "Radial Basis Functions For Multivariable Interpolation: A Review, In: J. G Mason, M. G Cox (Eds), Algorithms For Approximation Ima-1985 Conference, Clarendon Press, Oxford.

[50] M M Hochberg S J. Renals A J. Robinson, D. J. Kershaw, 1994, "Large Vocabulary Continuous Speech Recognition Using A Hybrid Connectionist-Hmm System', Proceedings Of Csip, Yokohama, 1499-1502.