Network Oral English Teaching System Based on Speech Recognition Technology and Deep Neural Network

Na He[1*], Weihua Liu[2]
School of Foreign Languages, Pingxiang University, Pingxiang, 337000, China[1]
Pingxiang Branch of Jiangxi Telecom Company, Pingxiang, 337000, China[2]

***Abstract*—With the development of computer technology, computer-aided instruction is being used more and more widely in the field of education. Based on speech recognition technology and deep neural network, this paper proposes an online oral English teaching system. Firstly, the speech recognition technology is introduced and its feature extraction is elaborated in detail. Then, three basic problems and three basic algorithms that need to be solved in speech recognition system using Markov model are discussed. The application of HMM technology in speech recognition system is studied, and some algorithms are optimized. The logarithmic processing of Viterbi algorithm, compared with the traditional algorithm, greatly reduces the amount of computation and solves the overflow problem in the operation process. By combining deep network with HMM, continuous speech signal modeling is realized. According to the characteristics of the DNN-HMM model, it is proposed that the model cannot model the long-term dependence of speech signals and train complex problems. Based on Kaldi, the model training comparison experiments of monophonon model, triphonon model and adding feature transformation technology are carried out to continuously improve the model performance. Finally, through simulation experiments, it is found that the recognition rate of the optimized DNN-HMM mixed model proposed in this paper is the highest, reaching 97.5%, followed by the HMM model, which is 95.4%, and the lowest recognition rate is the PNN model, which is 90.1%.**

***Keywords*—Deep neural network; Markov model; voice design technology; Viterbi algorithm; oral English teaching**

I. INTRODUCTION

With the evolution of computer science and technology, computer-aided instruction is being used more and more widely in the field of education. Nowadays, with the help of computer-aided instruction, people can learn languages more conveniently [1]. The rich graphics and sound processing functions of the computer effectively promote the language learning effect of people. At present, research hotspots in this field focus on exploring effective language learning methods that combine speech recognition technology with multimedia technology [2]. The development of software for teaching spoken English with speech identification has emerged as a hot topic in this type of language teaching.

For the time being, given the cutting-edge lookup development of monosyllabic recognition, Zhang Jing et al. first added the algorithm primarily based on finite country vector quantization and the lookup consequences of its expanded algorithm in monosyllabic recognition, then added the algorithm based totally on the implicit Markov model, and special brought the lookup consequences of syllable attention combining hidden Markov mannequin with different administration strategies [3]. Yang et al. introduced the Fisher criterion and L2 regularization constraint to ensure the minimization of parameter errors in the stage of backpropagation adjustment of parameters, the dispersion of samples between classes and the concentration of intra-class distributions after classification, and the proper order of magnitude of network weights to effectively alleviate the overfitting problem [4]. Sun et al. utilized the end-to-end technological know-how primarily based on hyperlink timing classification to Japanese speech recognition. Considering the traits of Hiragana, katakana, and kanji in more than one writing type in Japanese, they explored the effect of special modeling gadgets on awareness overall performance via experiments on Japanese records units [5]. Huang et al. proposed a finite local weight shared convolutional neural network (CNN) speech recognition based on the Meir spectral coefficient (MFSC) feature to address the problem of unsatisfactory recognition effect in traditional speech recognition applications [6]. Hou Yimin et al. mainly analyzed and summarized several current representative deep learning models, introduced their applications in speech identification for speech feature extraction and acoustic modeling, and finally summarized the problems faced before and the development direction [7].

Compared with the traditional single model for speech classification, this paper innovatively proposes to optimize the deep neural network and fuse it with the Markov model, and the recognition rate of the DNN-HMM fusion model is greatly improved. The application of HMM technology in speech recognition system is studied, and some algorithms are optimized. The logarithmic processing of Viterbi algorithm, compared with the traditional algorithm, greatly reduces the amount of computation and solves the overflow problem in the operation process. By combining deep network with HMM, continuous speech signal modeling is realized. According to the characteristics of the DNN-HMM model, it is proposed that the model cannot model the long-term dependence of speech signals and train complex problems.

*Corresponding Author.

II. SPEECH RECOGNITION TECHNOLOGY

A. Speech Recognition Technology and Feature Extraction

Speech attention means the ability to transform voice symbols into corresponding textual content [8]. Fig. 1 shows the shape of conventional voice attention and it consists in most cases of five parts: feature extraction, acoustic modeling, pronunciation lexicon, language modeling, and decoding search.

The mathematical description of this process is shown in Eq. (1):

$$\hat{W} = \arg\max_{W} P(W|\ O) \tag{1}$$

W is the candidate word sequence.

From Bayes' formula, Eq. (1) could be further written as:

$$\hat{W} = \arg\max_{W} P(W|\ O) = \arg\max_{W} \frac{P(W)P(O|\ W)}{P(O)} \tag{2}$$

P(W) stands for language model and represents the probability of occurrence of word sequence W; P(O|W) stands for acoustic model, which represents the probability of generating feature sequence O given word sequence W; P(O) represents the likelihood of watching the acoustic characteristic O, whose price has no impact on the closing cognizance result and could be overlooked [9]. So, system in Eq. (2) can be written as below:

$$\hat{W} = \arg\max_{W} P(W)P(O|\ W) \tag{3}$$

The speech signal is a kind of non-stationary signal, so it cannot use the traditional signal processing method. It is found that the characteristics of speech signals remain relatively stable in a short period (10~30ms). Therefore, it is necessary to do some processing before feature extraction, that is, pre-emphasis, frame, and window.

The speech signal is attenuated by 12dB/ octave after it is emitted from the glottis and by 6dB/ octave after it is radiated through the mouth. Therefore, in the speech spectrum generated after short-time FFT, the components of the high-frequency part are smaller, and the entire spectrum becomes steeper [10]. To flatten the signal spectrum, a pre-weighting process is generally required to improve the high-frequency portion at a rate of 6 dB per octave. This is usually accomplished using a first-order high-pass numerical filter:

$$H(z) = 1 - \mu z^{-1} \tag{4}$$

Frame processing is primarily used to enable the extraction of acoustic features and is predicated on the short-term stability of speech signals. The technique of overlapping segmentation is used to carry out frame segmentation, allowing for a seamless transition between different voice frames to guarantee their temporal continuity [11]. In the realm of digital signal processing, the most often utilized window functions are the rectangular window and the Hamming window, among others. In particular, the most popular window for voice recognition is the Hamming window, which may significantly reduce the spectrum leakage brought on by the truncation effect [12]. It has the following window functions:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right) & 0 \le n \le N-1 \\ 0 & \text{others} \end{cases} \tag{5}$$

The human ear perceives different frequency components of speech signals to different degrees, in which it is more sensitive to the low-frequency part and less distinguishable from the high-frequency part. MFCC is designed primarily based on this auditory grasp attribute of the human ear. Fig. 1 suggests the ordinary method of extracting MFCC features:

1) The enter voice sign is preprocessed to attain the time-domain sign after including the window.

2) Due to the challenging nature of analyzing the features of speech symbols in the temporal region, they are usually transformed into the spectral domain for evaluation [13]. The linear spectrum of the time-domain symbols is obtained by a short-time FFT transform:

$$S(k) = \sum_{n=0}^{N-1} s_w(n) e^{-j\frac{2\pi nk}{N}} \ (0 < k < N) \tag{6}$$

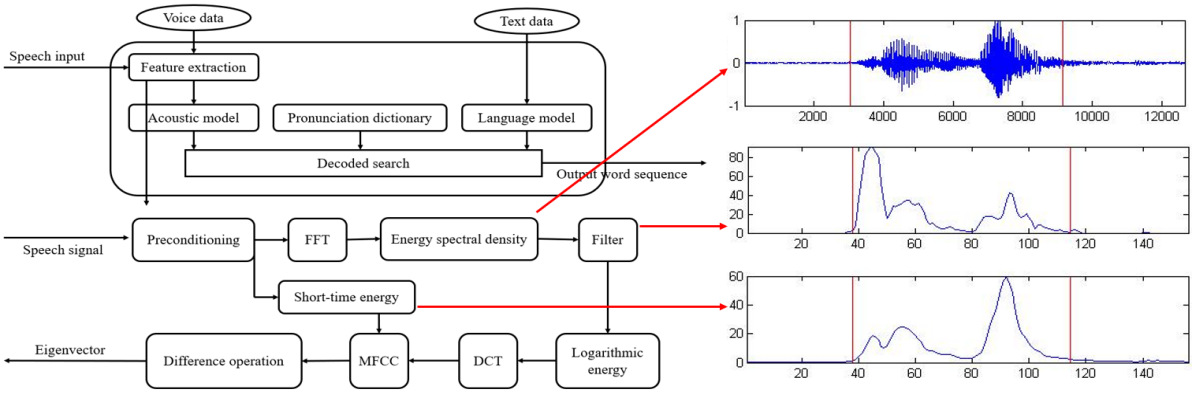

Fig. 1. Speech recognition architecture diagram.

After FFT transformation, the short-time emission profile can be obtained directly:

$$P(k) = S(k)S^*(k) = |S(k)|^2 \quad (7)$$

3) The MFCC features are constructed based on the auditory perception of the human ear, and the Mel frequency corresponds to the perceived frequency of the human ear. Therefore, to convert the linearity spectrum to Mel frequency, it is mainly realized by a set of delta strip bandpass filters uniformly distributed on the Mel frequency ruler [14].

The expression for this set of delta band pass factors is shown in Eq. (8):

$$H_m(k) = \begin{cases} \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) \le k \le f(m+1) \\ 0, \text{ others} \end{cases} \quad (8)$$

The conversion formula from linear frequency to Mel frequency is as follows:

$$Mel(f) = 1125 \times \ln\left(1 + \frac{f}{700}\right) \quad (9)$$

Or:

$$Mel(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \quad (10)$$

In this case, the triangle bandpass filter bank serves the following purposes: it may minimize the quantity of feature data and make calculations easier; it can smooth the spectrum and remove the effects of harmonics.

4) Determine the subband energies at the Mel scaling, or the logarithmic quantities of the energies of each factor in the channel bank, and perform a discharge cosine shift on them.

$$E_m = \ln\left(\sum_{k=0}^{N-1} P(k) H_m(k)\right) \quad (11)$$

$$C_d = \sum_{m=0}^{M-1} E_m \cos\left[m\left(k - \frac{1}{2}\right)\frac{\pi}{M}\right], d = 0,1,\ldots,D-1 < M \quad (12)$$

After a series of operation steps, a frame of speech signal can be represented by a multi-dimensional MFCC vector. For Fbank features, its extraction method is very similar to that of MFCC, except that DCT transformation is not required, so the correlation information between various dimensions is completely retained [15].

Short-time average zero crossing rate applications that can play the following roles in endpoint detection applications:

Distinguish between voiceless and voiceless sounds by the cost of the zero-crossing rate. The corresponding speech section with a greater zero crossing charge is voiceless, whilst the corresponding speech phase with a decreased zero crossing price is dulled [16]; Judging the beginning of speech by the price of the zero-crossing rate, the usage of the zero-crossing charge can be aware of that that is the noise, that is the actual beginning factor of speech; The zero-crossing rate combined with short-time energy can also be used as a basis for judging whether there is speech generation.

B. Linking Temporal Classification Algorithm

Both GMM-HMM and DNN-HMM belong to the mixed model in which a variety of models work together. There are many problems in this mixed modeling method. For example, in terms of model training, the process of fully training a hybrid modeling system is very complicated, and the relatively independent training of each module will make it difficult for the system to carry out overall optimization. Moreover, before training the DNN-HMM system, it is necessary to train a GMM-HMM to obtain the frame-level correspondence between speech data and labels, which leads to the final recognition accuracy of the system to a certain extent depending on whether the GMM-HMM model alignment is accurate [17]. In terms of the structural characteristics of the model itself, to obtain a high recognition rate in continuous speech recognition tasks with a large vocabulary, only the modeling units below the word can be selected, such as mono phonemes, tri phonemes, etc., which need to be escaped by pronunciation dictionaries. In addition, the HMM model used to model speech sequence information fails to take into account the contextual correlation between speech frames.

Proposed by Alex Graves in 2006, the linked time sequence classification method is a key technology for end-to-end speech recognition. Neural networks alone complete the entire process of continuous speech recognition, which belongs to an integrated modeling method [18]. The final text sequence can be generated directly, which solves the problems of traditional methods such as forced alignment, cumbersome recognition process, and non-consistent optimization.

Assuming that the conditions between the output symbols of each speech frame are independent, for a single sample (O, W), the final optimization goal of CTC can be described as minimizing the negative logarithm of the posterior probability of the output symbol sequence:

$$L_{CTC} = -\ln P(\mathbf{W}|\ \mathrm{O}) = -\ln \sum_{\pi \in B^{-1}(\mathbf{W})} \prod_{t=1}^{T} p\left(\pi_t|\ \mathbf{o}_t\right) \quad (13)$$

From Eq. (13), it can be seen that if the CTC loss function is computed directly, it is necessary to obtain a summation of the conditional probabilities of all possible symbol orders capable of yielding the output character sequence W. Therefore, similar to the forward-backward algorithm in HMM calculation, dynamic programming is usually used to calculate the loss of CTC in practice. By combining the symbol sequence that can get the same output character sequence in the same time step, the calculation amount is reduced and the double calculation is avoided.

$$\alpha(s,t)=\sum_{\pi_{1:t}\in B^{-1}(\mathbf{W}_{1:s/2}),\pi_t=w_s'}\prod_{i}^{t}P\left(\pi_i|\ x_i\right) \tag{14}$$

$$\alpha(1,1)=P\left(\varphi|\ o_1\right) \tag{15}$$

$$\alpha(1,2)=P\left(w_1|\ o_1\right) \tag{16}$$

$$\alpha(1,s)=0,\quad \forall s>2 \tag{17}$$

According to the introduction of the neural network, after establishing the neural network, it is necessary to reduce the mistake between the forecasting result of the neural network and the actual sample labeling as much as possible, i.e., to reduce the loss function as much as possible. By analyzing the damage factor of CTC and the calculation method of the loss function, it can be found that the output probability of each time step is microscopic. Therefore, the backpropagation algorithm can be used to update each parameter value of the neural network using the CTC method and minimize the loss function to realize the classification of time series. The following describes the reverse derivation of the CTC loss function on the output of the neural network.

Assume that the number of neurons in the output level of the neural network is |U '|, SoftMax is used as the activation function, and each neuron outputs a posterior probability of a specific symbol on the time step. Then, for a single sample, the bias of the posterior probability of the output character label sequence as a whole to the output of a single neuron of the neural network is:

$$\frac{\partial P(\mathbf{W}|\ \mathbf{O})}{\partial P\left(q|\ o_t\right)}=-\frac{1}{P\left(q|\ o_t\right)^2}\sum_{s\in lab(\mathbf{W},q)}\alpha(s,t)\beta(s,t) \tag{18}$$

Then, the derivative of the CTC loss function to the output of the neural network is:

$$\frac{\partial L_{CTC}}{\partial P\left(q|\ o_t\right)}=\frac{1}{P(\mathbf{W}|\ \mathbf{O})P\left(q|\ o_t\right)^2}\sum_{s\in lab(\mathbf{W},q)}\alpha(s,t)\beta(s,t) \tag{19}$$

III. MARKOV MODEL

A. Basic Problem and its Structure

Given the remark sequence and HMM model, if the kingdom transition sequence is known, the chance that the HMM mannequin produces the output remark sequence is:

$$P(O|\ q)=b_{q_1}\left(O_1\right)b_{q_2}\left(O_2\right)\ldots b_{q_T}\left(O_T\right) \tag{20}$$

The probable gas output sequence q of the HMM model is:

$$P(q|\ \lambda)=\pi_{q_1}a_{q_1q_2}a_{q_2q_3}\ldots a_{q_{T-1}q_T} \tag{21}$$

For all possible state transition sequences q, the model outputs the probability of observing sequence O:

$$\begin{aligned}P(O|\ \lambda)&=\sum_{\forall q}P(O|\ q,\lambda)P(q|\ \lambda)\\&=\sum_{q_1,q_2,\ldots,q_T}\pi_{q_1}b_{q_1}\left(O_1\right)a_{q_1q_2}b_{q_2}\left(O_2\right)\ldots a_{q_{T-1}q_T}b_{q_T}\left(O_T\right)\end{aligned} \tag{22}$$

In practice, this amount of computation cannot be borne, so forward algorithm and backward algorithm are adopted to reduce the amount of computation [19].

Define the forward probability as:

$$\alpha_t(i)=P\left(O_1O_2\ldots O_T,q_t=S_i|\ \lambda\right) \tag{23}$$

The forward probability may be computed using the recursion formula that follows:

Starting Point:

$$\alpha_1(i)=\pi_ib_i\left(O_1\right)\quad 1\le i\le N \tag{24}$$

Iteration:

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j\left(O_{t+1}\right),\quad 1\le t\le T-1,1\le j\le N \tag{25}$$

Terminate:

$$P(O|\ \lambda)=\sum_{i=1}^{N}\alpha_T(i) \tag{26}$$

The first step is to set the forward probability to the combined probe of the state and the first observable. The main component of the algorithm is the second step. If the phantom is in any of the N potential ones at moment t, it will transfer to the state at moment t+1 with a specific probability. Step 3 The sum of all aT(i) is P(O|λ) according to the definition of forward probability.

Corresponding to the forward probability, the backward probability is defined as:

$$\beta_t(i)=P\left(O_{t+1}O_{t+2}\ldots O_T|\ q_t=S_i,\lambda\right) \tag{27}$$

Initialization:

$$\beta_T(i)=1,\quad 1\le i\le N \tag{28}$$

Iteration:

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j\left(O_{t+1}\right)\beta_{t+1}(j),\quad 1\le t\le T-1,1\le j\le N \tag{29}$$

Terminate:

$$P(O|\ \lambda)=\sum_{i=1}^{N}\beta_1(i) \tag{30}$$

By using forward and backward probabilities, the likely output of the entire observational series to the HMM model can be divided into the product of the probabilities of the two

observational series. By using the corresponding recurrence formula, the following output probability calculation formula can be obtained:

$$P(O|\ \lambda)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j\left(O_{t+1}\right)\beta_{t+1}(j),\quad 1\le t\le T-1 \tag{31}$$

The Viterbi algorithm is usually used, which is a dynamical engineering-based approach for searching for a singularly best sequence of states [20]. The problem to be solved by the Viterbi algorithm is to determine a sequence of states that maximizes the probability of the outcome given a given sequence of observations and a module, i.e., the problem of identification.

$$\delta_t(i)=\max_{q_1,q_2,\ldots,q_{t-1}}P\left(q_1,q_2,\ldots,q_t,q_t=S_i,O_1,O_2,\ldots O_t|\ \lambda\right) \tag{32}$$

If the state of the system at point t is Si, the optimal path is traced back to time T-1 along the optimal route, and a state labeling of the system is introduced, then the process of searching for the optimal succession of states is as follows:

Initialization:

$$\begin{aligned}&\delta_1(i)=\pi_i b_i\left(O_1\right),1\le i\le N\\&\psi_1(i)=0\end{aligned} \tag{33}$$

Iterative calculation:

$$\begin{aligned}&\delta_t(j)=\max_{1\le i\le N}\left|\delta_{t-1}(i)a_{ij}\right|b_j\left(O_t\right),\quad 2\le t\le T,1\le j\le N\\&\psi_t(j)=\arg\max_{1\le i\le N}\left|\delta_{t-1}(i)a_{ij}\right|,\quad 2\le t\le T,1\le j\le N\end{aligned} \tag{34}$$

Termination calculation:

$$\begin{aligned}&p^*=\max_{1\le i\le N}\delta_T(i)\\&q_T^*=\arg\max_{1\le i\le N}\delta_T(i)\end{aligned} \tag{35}$$

Path backtracking:

$$q_t^*=\psi_{t+1}(i)\left(q_{t+1}^*\right),\quad t=T-1,T-2,\ldots,1 \tag{36}$$

A statistical model of voice signals' time series structure, known in mathematics as a double stochastic process, is called the hidden Markov model.

Two methods are available for simulating the change in the statistical properties of the speech signal: an implicit random process that employs a Markov chain with a finite number of states, and a random process that utilizes the observation sequence linked to each stage of the Markov chain. Although the latter expresses the previous, the former's actual boundaries are incalculable.

For voice recognition, there are four pattern-matching steps: feature extraction, template training, template classification, and judgment. Fig. 2 illustrates its fundamental framework:

Voice identification using HMM is essentially a probabilistic operation. After calculating the model parameters based on the training set dataset, it is also required to calculate the conditional probability (Viterbi's algorithm) of each model separately based on the test set data, and the model with the largest likelihood is the recognition result [21].

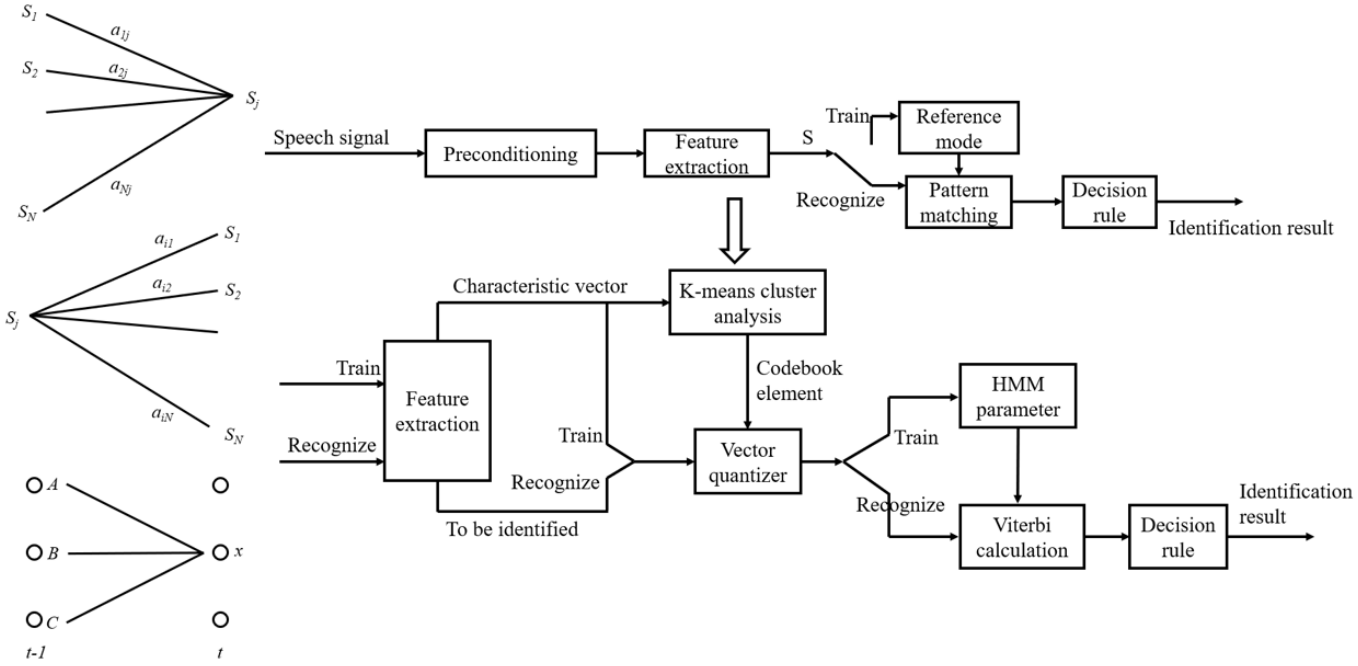

Fig. 2. Hidden markov model architecture.

Model parameter estimation is the process of training the pattern parameters, i.e., the series of known observations, and the output probability can be maximized by adjusting the model parameters. The specific steps are as follows: first, initialize the model parameters, then use some algorithm to input the same sequence of observations as the sequence of training samples, and repeatedly correct the sequence of observations, so that the pattern parameter with the maximum output potential becomes the final trained model parameter [22]. Since the Baum-Welch method is based on the maximum likelihood criterion, it has the advantages of fast convergence speed and monotonous growth of likelihood value, so the Baum-Welch method is usually used for parameter re-estimation.

The Baum-Welch algorithm uses the maximum likelihood criterion to obtain a fresh pair of variables by replacing the observed values and the initial model parameters. In other words, the results computed from multiple substitutions according to the re-estimation formula are more representative of the observed series than the original inputs. This process is repeated until convergence, which is the desired model parameter.

$$\xi_t(i,j) = p\left(q_t = i, q_{t+1} = j \mid O, \lambda\right) \tag{37}$$

According to the forward-backward algorithm, we can get:

$$\xi_t(i,j) = \frac{p\left(q_t = i, q_{t+1} = j, O \mid \lambda\right)}{P(O/\lambda)} = \frac{\alpha_t(i) a_{ij} b_j\left(O_{t+1}\right) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{38}$$

Then, the odds that the observed sequence is in a certain condition at time t:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{39}$$

The flowchart of the Baum-Welch algorithm is below in Fig. 3.

B. Application of the HHM Model in Speech Recognition

The choice of state type is the first problem to be considered when HMM is applied to speech recognition. There are two kinds of HMM model structure: each state traversal and left to right. The HMM model of state traversal can be applied to speaker recognition, language recognition, and so on. According to the characteristics of the human pronunciation process, the HMM of the right and left models is generally selected in speech recognition. In this paper, no span left-right model is adopted. There is no clear regulation on the number of states, which needs to be clarified through experiments and experience [23]. In English isolated word recognition technology, generally, the number of states between 4-8 is sufficient to achieve better results. Through the analysis of the test results, the amount of states N=4 is selected.

In addition to the number of states, the Gaussian mixture number of observed probability density functions also determines the final recognition rate. Through the analysis of subsequent experimental simulation results, the Gaussian mixture model number selected in this paper is M=3.

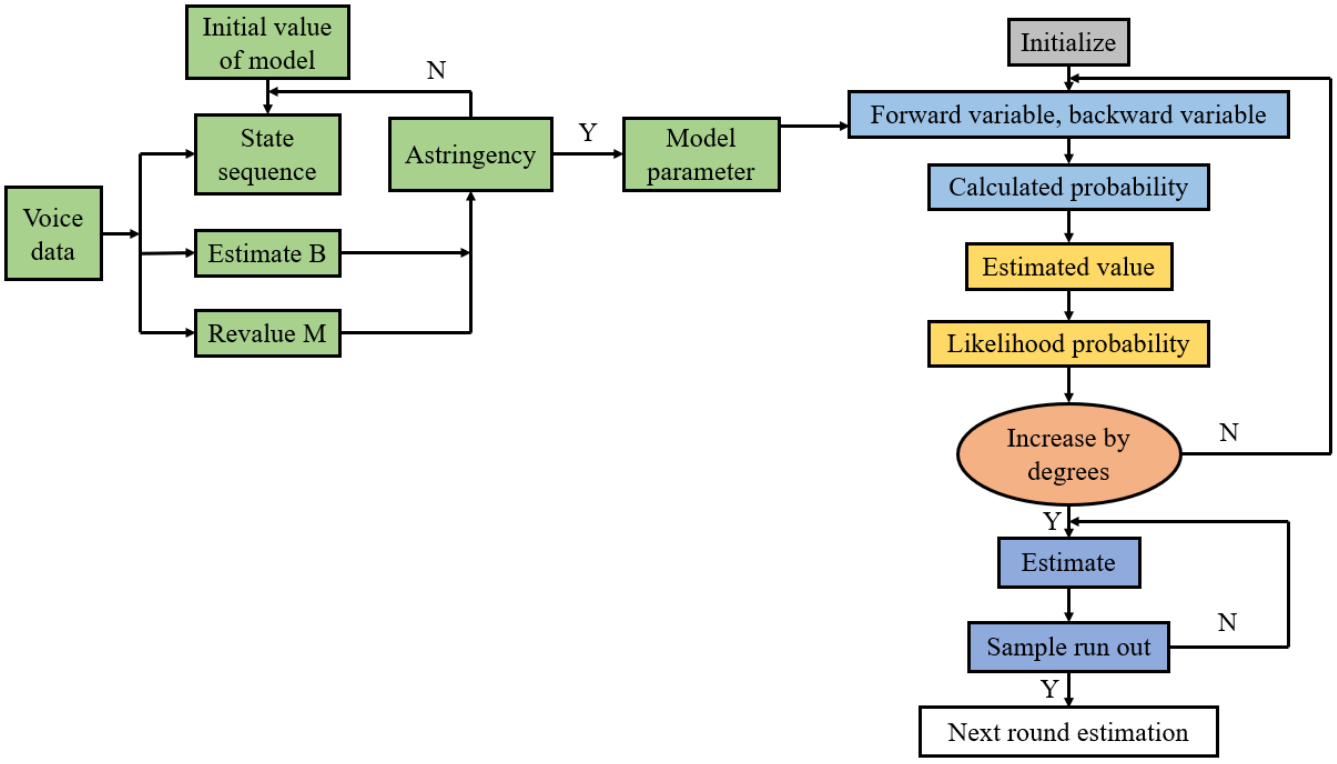

Fig. 3. Flowchart of Baum-Welch algorithm.

When applying the Baum-Welch algorithm to the parameter training of HMM models, how to correctly determine the initial parameters of the HMM to keep the partial maximization value as close as feasible to the global optimum is the focus of research [24]. In addition, a good choice of initial values can also reduce the number of iterations required for convergence, i.e., improve the computational efficiency. According to experience, the selection of the internal probability and initial value of the state transformation matrix does not have much influence on the recognition rate, and non-zero random numbers or uniform values can be used. Therefore, the initial probability of the state transition matrix and the choice of the initial value of the state transition matrix do not have much influence on the recognition rate, and non-zero random numbers or uniform values can be used. Based on this consideration, this paper adopts a composite clustering algorithm that combines the piecewise K-means algorithm, Viterbi algebras, and Baum-Welch algebras.

The basic idea of the method is as below:

1) Establish the initial parameters of the HMM model.

2) According to λ, use the Viterbi method to classify the input training speech database into the most probable state sequences.

3) Re-estimation of the input B using the "segmented K-means" method for the continuous HMM assumption using a hybrid Gaussian density function with M number of mixtures. The speech parameters corresponding to a particular state are pooled together and a K-means clustering operation is performed to classify the trained state speech clusters into M classes [25]. Then the meaning and covariance of the speech parameters of the same class are computed as the meaning vector and variance covariance mosaic of the class, thus obtaining the M normally distributed parameters of the M classes. Finally, the mixture weights of the class density functions are obtained by taking the number of speech frames included in each class and dividing it by the number of all speech in that state. This gives a new B, and thus a new set of initial values.

4) The λ obtained in step (3) is used as the initial value for the BW parameter re-estimation method to perform parameter re-estimation on the HMM module, thus obtaining the new module variables.

5) If the difference is less than the preset min value, it indicates that the model parameters have converged, there is no need to re-estimate, and λ is the final parameter output. Otherwise, λ continues the iteration as the new initial argument.

Since the segmented K-means algorithm is the idea of state optimization to carry out the maximum likelihood criterion, it can greatly accelerate the training speed of the model by realizing the initial parameter re-estimation.

IV. Optimize the DNN-HMM Model and its Analysis

A. Deep Neural Network

A neural network is also composed of neurons as shown in Fig. 4, which is a simple nervous web. The net is composed of an input level, a hidden level, and an output level. The nodes directly connected between layers are fully connected. This network model can fit simple nonlinear transformations.

When the amount of input data increases, it is necessary to fit complex nonlinear transformations, which can be achieved by adding the number of neurons in the hidden layer or the number of layers in the hidden layer. As shown in Fig. 4, the number of neurons and the number of hidden layers in a deep Neural Web increases accordingly.

The network can be trained using the error backpropagation algorithm, which can accommodate more complex functions. To meet the large increase in the amount of data, the number of layers of the deep neural network needs to be increased. At this point, the amount of network layers increases, the parameters surge, and training becomes more difficult, and the results are often difficult to converge.

The activation function in neurons is nonlinear, and the input is digital information, through which certain mathematical operations can be performed [26]. According to different excitation forms, there are different activation functions:

Sigmoid activation function:

$$f(x)=\frac{1}{1+\exp(-x)} \tag{40}$$

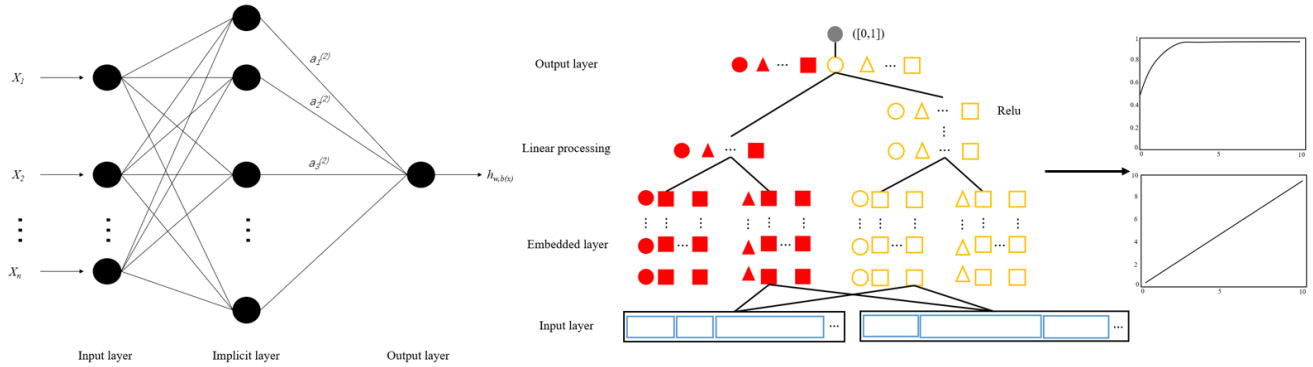

Fig. 4. Deep neural network model.

Tanh activity feature:

$$f(x) = \tanh(x) \tag{41}$$

ReLU activity feature:

$$f(x) = \max(x, 0) \tag{42}$$

The featured graph of the Tanh function compensates the output value between [-1, 1] and suffers from the same saturation problem as the sigmoid feature, but its output is zero-centered. Therefore, in practice, the Tanh nonlinear function is widely used.

Compared with the sigmoid and tanh functions, the ReLU function has a great acceleration effect on the convergence of stochastic gradient descent.

B. Optimization of DNN-HMM Model

The DNN-HMM model in speech recognition combines HMM and DNN, each performing its duties and sharing different tasks. HMM models speech timing signals. DNN models the posterior probability distribution of the input sample.

The input of DNN is the feature vector extracted from each frame of a speech signal after it is divided into frames. However, the output vector is the probability of the corresponding HMM state, so its dimension is equal to the number of states in the HMM. But since this is a supervised learning task, the input-to-output mapping must be found. Although the speech information and its corresponding text information of each sample can be known from the training sample, it seems that this is the mapping relationship between the input speech and the output text, but the DNN-HMM does not directly model the whole speech, but the model of each frame signal. Therefore, it is necessary to obtain the HMM state label corresponding to each input vector in the DNN, and then use this label to train a DNN model. In order to obtain the HMM state labels corresponding to each utterance, a traditional GMM-HMM model is usually trained in advance. GMM-HMM is used to force align the training samples. Each signal in the observed sequence is aligned with the state of its corresponding HMM. After alignment, DNN can be used instead of GMM to calculate the observation probability in HMM for training.

The input to the model is the feature of successive frames, that is, in addition to the current frame, the information of previous and subsequent frames is included. In general, if the current frame time is t, the input feature also includes all information from time T-4 to time t+4. The combination of 9 consecutive features to represent the current situation effectively utilizes contextual information, which is one of the reasons that DNN is superior to GMM. However, for continuous speech recognition tasks, it is not enough to only use the context information of the frame to model the speech. It is necessary to take into account the long-term dependence of the speech signal in time. The model works better if it can model the long-term dependence of the speech signal using its historical information. However, DNN-HMM based on a feedforward neural network cannot do this.

TDNN also belongs to the feed forward neural network, and its network structure is shown in Fig. 5. Different from DNN, it is not fully connected between layers, but the connection is controlled by the delay parameter. For example, [-2,2] in hidden layer 1 represents that the current frame is taken as the basis, and the two frames before and after each frame are a total of five as the input of the network, and each neuron in the time domain connects with the previous layer according to this rule. The architecture of the hidden levels is the same, so the income of each hidden level in TDNN is not just the output of the preceding level at the present moment, but also the outcome of the preceding level at t moments around the previous level. This allows each hidden layer to extend the time domain, especially deeper into the network, which contains more information from the input layer.

From a local point of view, the same layer of TDNN can be divided into many repeated network structures. For example, the first hidden layer can be split into substructures as shown in Fig. 5.

C. Neural Network Model Training under Kaldi

DNN-HMM model training is supervised training, which means that alignment information is obtained by the GMM-HMM model before training. The regularity of input features is conducive to the training of model parameters, so the input features are transformed in different stages of the GMM-HMM model training. The total transformation process is shown in Fig. 6.

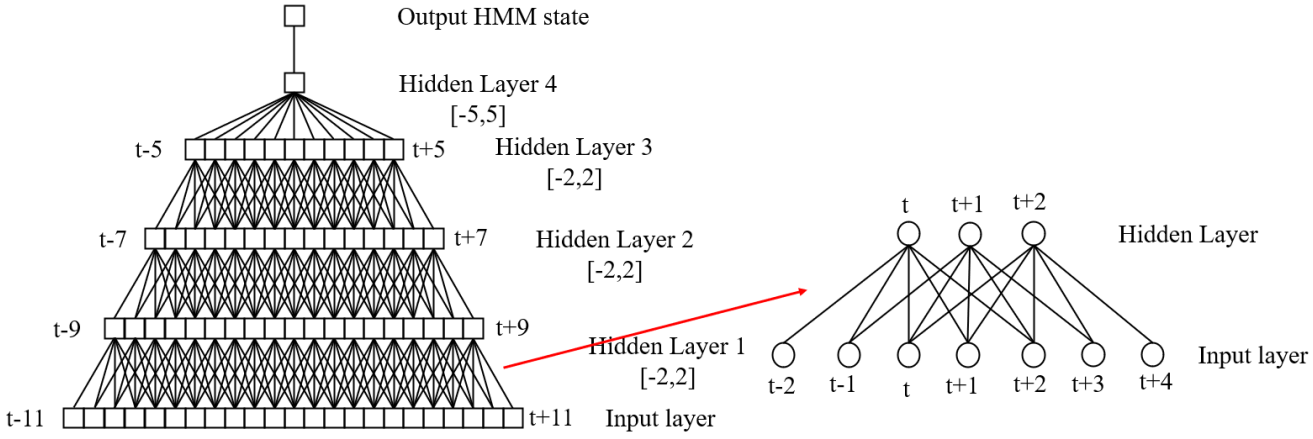

Fig. 5. TDNN structure diagram.

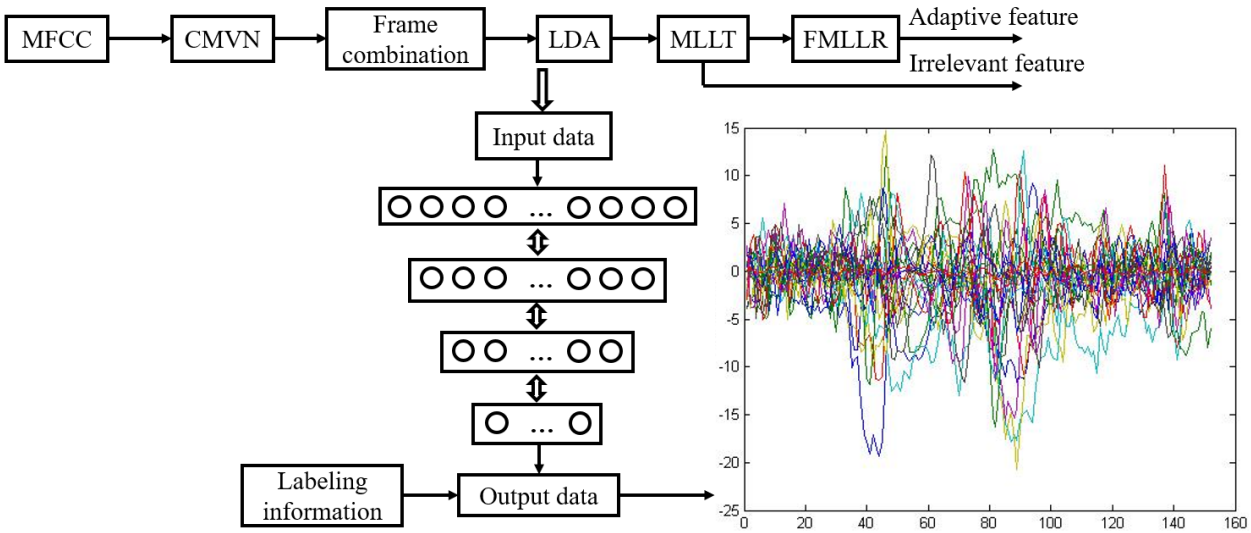

Fig. 6. DNN-HMM training process.

The GMM input feature has not been normalized after a series of processing, so another normalization process is needed. Based on this, several frames of data are spliced together forward and backward in the time domain to augment the modeling capability of the neural net, and the feature data need to be normalized and de-correlated again. Because the feature dimensions of time domain splicing are strongly correlated, it is not conducive to the training of neural networks. In this paper, CMVN is used to normalize the mean and variance of each dimension to achieve the purpose of de-correlation.

where, each element of the weight matrix specifies the weight of the edge between the concealed level cell and the visual level cell, with a bias for each visible level cell and a bias for each concealed level cell. Depending on the distribution of the random variables, the "energy" of the RBM is defined in two ways:

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} v_i \tag{43}$$

$$E(v,h) = \sum_i (a_i - v_i)^2 - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} v_i \tag{44}$$

Eq. (43) represents the equivalent energy function for the Bernoulli distribution and Eq. (44) represents the equivalent energy function for the Gaussian distribution. According to the Gibbs distribution, the probability of RBM in the current state is given by Eq. (45).

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{45}$$

$$Z = \sum_v \sum_h e^{-E(v,h)} \tag{46}$$

This probability can be considered as a joint probability distribution of the deep-level state and the hidden-level state, which is obtained by the RBM energy of the current state being normalized by the RBM energy of all possible states according to the exponential rule, where, Z is the partition function, which is a regular term that takes into account the energy of RBM in all states. Therefore, the edge distribution of the state vector of the display layer could be derived from the above joint distribution.

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \tag{47}$$

After Kaldi calls steps/net/pre-train dbn. sh for pre-training, it is time to train the neural network. The Truncated BPTT algorithm is usually used, that is, BPTT is carried out in a data block, which can contain a sentence or a fragment of fixed length. In Kaldi, steps/nnet/train.sh is called for training, the number of input layer units is set to 40, the number of hidden layers is fixed to 4, and the number of units in each level is set to 1024. the initial studying ratio is defined to be 0.008. the results obtained from the model's alignment will be used for the subsequent discriminative training.

D. Result and Discussion

According to the implementation method of the HMM model, PNN model, and DNN-HMM mixed model, MATLAB programming is carried out for the three models, and the three recognition systems are tested and analyzed with the help of the MATLAB simulation experiment platform.

All experiments were programmed using MATLAB for the three models, and the recognition system model program was split into two parts: the speech trainer and the speech recognition program. The speech test samples were 10 place names, namely Beijing, Shanghai, Guangzhou, Tianjin, Chongqing, Wuhan, Shenyang, Dalian, Changchun, and Harbin. The speakers were five men and five women, each pronouncing each word five times. The top 250 samples were taken as test samples and the bottom 250 samples were taken as training samples. The speech training program first builds a reference model library for these 10 words, and then the speech

recognition program recognizes the results by comparing the input parameters with the reference model library. To validate the anti-jamming performance of the DNN-HMM mixture module, the HMM model and the PNN module are compared in a pure speech environment and a signal-to-noise environment, respectively. Table I displays the word recognition rate compared between the models in the pure speech environment. The first digit in the table is the number of correctly recognized words, and the last digit is the overall number of recognized letters. 24/25 means that 25 samples are input to be recognized, and the number of correctly recognized words is 24.

According to the speech recognition rate data in the above table, the test of this model in Beijing, Tianjin, Wuhan, Changchun and Harbin has reached 100%, while the test of this model in other regions is also close to 98%. The recognition rate of the optimized DNN-HMM mixed model is the highest, reaching 97.5%, followed by the HMM model with 95.4%, and the PNN model with 90.1% is the lowest. The data comparison shows that the hybrid model has better recognition performance than the single model, and the recognition rate is improved, which achieves the expected effect. There is little difference between the HMM and the optimized DNN-HMM modules in terms of recognition rate. To further illustrate the benefits and performance of hybrid models in speech recognition, we analyzed the anti-interference performance of each model. Table II displays a comparative identification ratio after adding different signal-to-noise ratios to each model.

From Tables I and II, it can be observed that as the SNR gradually decreases, the identification rate of the speech identification system continues to decrease, and the identification rates of the HMM model and the PNN model decrease obviously, however, the identification rate of the hybrid module decreases less obviously than that of the single model. The optimized DNN-HMM hybrid module incorporates the strong sequential handling capability of the HMM model and the excellent classifying capability of the probabilistic neural net. This combined ability can characterize the linguistic content of voice in a more comprehensive and detailed way, and improve the recognition rate, anti-interference performance, and ruggedness of the speech recognition system.

TABLE I. COMPARISON OF RECOGNITION RATES OF DIFFERENT RECOGNITION MODELS

Model / *Recognizer*	***HMM model***	***PNN model***	***Optimized DNN-HMM***
Peking	24/25	22/25	25/25
Shanghai	24/25	21/25	24/25
Guangzhou	21/25	22/25	24/25
Tianjin	25/25	23/25	25/25
Chongqing	24/25	22/25	24/25
Wuhan	23/25	24/25	25/25
Shenyang	21/25	22/25	24/25
Dalian	23/25	22/25	23/25
Changchun	25/25	21/25	25/25
Harbin	24/25	22/25	25/25
Product recognition rate	95.4%	90.1%	97.5%

TABLE II. COMPARISON OF RECOGNITION RATES OF DIFFERENT SNRS OF DIFFERENT MODELS

model / *Signal-to-noise ratio*	***HMM model (%)***	***PNN model (%)***	***Optimized DNN-HMM (%)***
5db	31.3	45.3	64.1
10db	63.4	70.3	79.9
15db	75.1	77.3	89.1
20db	88.1	86.5	93.6
25db	89.8	89.3	94.7
30db	93.6	90.4	96.7

V. CONCLUSION

In this paper, a web-based oral English teaching system is proposed by combining speech recognition technology with a deep neural network-Markov model. The continuous speech signal is modeled by combining DNN and HMM. To solve the long-term dependence problem of DNN, TDNN is used to reconstruct the acoustic model, its structure and implementation principle are introduced, and the neural network model is built and trained by Kaldi. Specific conclusions are as follows:

First, this paper takes the technical principles of speech identification as the theoretical basis for a thorough study and research. Speech identification system according to the Hidden Markov Model. In speech processing, MFCC is used as the feature principle of speech, and the Hidden Markov Model is utilized for training and recognition, to produce a speech recognition system.

Second, because GMM-HMM does not utilize the context information of frames, DNN is introduced to construct a new acoustic model DNN-HMM, and its structure and training algorithm are introduced. However, DNN does not take into account the long-term dependence on the time of speech signals, so TDNN is used to reconstruct the acoustic model to introduce its structure and advantages, build and train the neural network based on Kaldi, and apply the DT method to train the neural network model to continuously improve the performance of the acoustic model.

Third, the simulation results show that in terms of speech identification rate, the optimized DNM-HMM hybrid model has the highest recognition rate of 97.5%, followed by the HMM model with 95.4%, and the PNN model has the lowest identification rate of 90.1%. The optimized DNN-HMM hybrid model achieves a high recognition rate of 96.7% when the signal-to-noise ratio is 30db. It incorporates the strong sequential processing capability of the HMM model and the excellent categorization capability of the probabilistic neural network, which can characterize the semantic meaning of speech in a more comprehensive and detailed way.

REFERENCES

[1] Abdel-Hamid O, Mohamed A, Jiang H, et al. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, 2014, 22(10): 1533-1545. Doi: 10.1109/TASLP2014.2339736.

[2] Kamble B C. Speech recognition using an artificial neural network–a review. Int. J. Comput. Commun. Instrum. Eng, 2016, 3(1): 61-64.

[3] Zhang Jing, Yang Jian, Su Peng. A review of monosyllabic recognition in Speech recognition. Computer Science, 2019, 47(S2):172-174+203. (in Chinese).

[4] Yang Yang, Wang Yuduo. Speech recognition based on improved Convolutional neural networks. Applied Acoustics, 2018, 37(06):940-946. (in Chinese).

[5] Sun Jian, Guo Wu. Japanese Speech recognition based on Link sequential classification. Minicomputer Systems, 2018, 39(10):2129-2133. (in Chinese).

[6] Huang Yulei, Luo Xiaoxia, Liu Duren. MFSC coefficient characteristics of locally finite weighted sharing CNN speech recognition. Journal of control engineering, 2017, 24 (7): 1507-1513.

[7] Hou Yimin, Zhou Huiqiong, Wang Zhengyi. Application Research of Computers, 2017, 34(08):2241-2246.

[8] Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. International conference on machine learning. PMLR, 2014: 1764-1772.

[9] Amberkar A, Awasarmol P, Deshmukh G, et al. Speech recognition using recurrent neural networks. 2018 international conference on current trends towards converging technologies (ICCTCT). IEEE, 2018: 1-4. Doi: 10.1109/ICCTCT.2018.8551185.

[10] Nassif A B, Shahin I, Attili I, et al. Speech recognition using deep neural networks: A systematic review. IEEE access, 2019, 7: 19143-19165. Doi:10.1109/ACCESS.2019.2896880.

[11] Dua S, Kumar S S, Albagory Y, et al. Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network[J]. Applied Sciences, 2022, 12(12): 6223. https://doi.org/10.3390/app12126223.

[12] Swietojanski P, Ghoshal A, Renals S. Convolutional neural networks for distant speech recognition. IEEE Signal Processing Letters, 2014, 21(9): Doi:1120-1124. 10.1109/LSP.2014.2325781.

[13] Lokesh S, Malarvizhi Kumar P, Ramya Devi M, et al. An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map[J]. Neural Computing and Applications, 2019, 31: 1521-1531. https://doi.org/10.1007/s00521-022-08144-x.

[14] Islam J, Mubassira M, Islam M R, et al. A speech recognition system for Bengali language using recurrent neural network. 2019 IEEE 4th international conference on computer and communication systems (ICCCS). IEEE, 2019: 73-76. Doi: 10.1109/CCOMS.2019.8821629.

[15] Fohr D, Mella O, Illina I. New paradigm in speech recognition: deep neural networks. IEEE international conference on information systems and economic intelligence. 2017.

[16] Vydana H K, Vuppala A K. Residual neural networks for speech recognition. 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017: 543-547. Doi: 10.23919/EUSIPCO.2017.8081266.

[17] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. 2013 IEEE international conference on acoustics, speech and signal processing. Ieee, 2013: 6645-6649. Doi:10.1109/ICASSP.2013.6638947.

[18] Siniscalchi S M, Yu D, Deng L, et al. Exploiting deep neural networks for detection-based speech recognition. Neurocomputing, 2013, 106: 148-157. https://doi.org/10.1016/j.neucom.2012.11.008.

[19] Rani P, Kakkar S, Rani S. Speech recognition using neural network. International journal of computer applications, 2015, 4: 11-14.

[20] Waibel A. Modular construction of time-delay neural networks for speech recognition. Neural computation, 1989, 1(1): 39-46. https://doi.org/10.1162/neco.1989.1.1.39.

[21] Song W, Cai J. End-to-end deep neural network for automatic speech recognition. Standford CS224D Reports, 2015: 1-8.

[22] Sainath T N, Weiss R J, Wilson K W, et al. Multichannel signal processing with deep neural networks for automatic speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(5): 965-979. Doi:10.1109/TASLP.2017.2672401.

[23] Mustafa M K, Allen T, Appiah K. A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition[J]. Neural Computing and Applications, 2019, 31: 891-899. https://doi.org/10.1007/s00521-017-3028-2.

[24] Chan W, Jaitly N, Le Q, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016: 4960-4964. Doi: 10.1109/ICASSP.2016.7472621.

[25] Al Smadi T, Al Issa H A, Trad E, et al. Artificial intelligence for speech recognition based on neural networks. Journal of Signal and Information Processing, 2015, 6(02): Doi:66. 10.4236/jsip.2015.62006.

[26] Saksamudre S K, Shrishrimal P P, Deshmukh R R. A review on different approaches for speech recognition system. International Journal of Computer Applications, 2015, 115(22).