J. Shanghai Jiao Tong Univ. (Sci.), 2020, 25(1): 70-75
https://doi.org/10.1007/s12204-019-2147-6

Joint CTC-Attention End-to-End Speech Recognition with a Triangle Recurrent Neural Network Encoder

ZHU Tao (朱　涛), *CHENG Chunling** (程春玲)
(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: Traditional speech recognition model based on deep neural network (DNN) and hidden Markov model (HMM) is a complex and multi-module system. In other words, optimization goals may differ between modules in traditional model. Besides, additional language resources are required, such as pronunciation dictionary and language model. To eliminate the drawbacks of traditional model, we hereby propose an end-to-end speech recognition method, where connectionist temporal classification (CTC) and attention are integrated for decoding. In our model, the complex modules are replaced by a single deep network. Our model mainly consists of encoder and decoder. The encoder is constructed by bidirectional long short-term memory (BLSTM) with a triangular structure for feature extraction. The decoder based on CTC-attention decoding utilizes advanced features extracted by shared encoder for training and decoding. The experimental results on the VoxForge dataset indicate that end-to-end method is superior to basic CTC and attention-based encoder-decoder decoding, and the character error rate (CER) is reduced to 12.9% without using any language model.

Key words: end-to-end, connectionist temporal classification (CTC), attention, speech recognition
CLC number: TP 183　　**Document code:** A

0 Introduction

Speech is the most natural way of communication. Automatic speech recognition (ASR) converts the speech signal into text to support the machine to understand the information contained in the speech[1]. In recent years, the researches on speech recognition have made remarkable achievements. In the 1980s, Rabiner[2] proposed the application of hidden Markov model (HMM) to speech recognition, which made speech recognition evolve from isolated word recognition to large vocabulary continuous speech recognition (LVCSR). HMM has become a milestone in the field of speech recognition. With the successful application of HMM in speech systems, it is considered as the most successful model in the field of speech recognition. It has been proved to be a fairly good statistical model for estimating speech acoustic sequence data, especially the temporal characteristics of acoustic data. Therefore, since the mid-80s, HMM has become a popular model in the field of speech recognition. However, HMM also owns some well-known defects, such as excessive reliance on conditional independence assumptions and the reduction of the discriminative ability of the acoustic model by standard maximum likelihood (ML) training algorithm. These limitations make it difficult to improve its performance continuously. With the development of machine learning and the emergence of neural networks, deep neural network (DNN) has greatly improved the performance of ASR[3]. People began to combine DNN with traditional statistical models such as HMM or Gaussian mixture model (GMM), and got mixed system such as DNN-HMM. However, different objective functions cause optimization results to be irrelevant and poor performance in these mixed systems.

Recent studies have shown that end-to-end speech recognition method can overcome the shortcomings of mixed systems and simplify the model into a single network which consists of encoder and decoder. There are two main end-to-end speech recognition methods: connectionist temporal classification (CTC) method[4-5], and encoder-decoder model based on attention mechanism[6]. Markov assumption is used in CTC to solve sequence problems based on dynamic programming, and has been successfully applied to phoneme recognition and character-based LVCSR. The attention-based method can directly learn the mapping from acoustic frames to character sequences. The problem of variable-length input and output sequences can be addressed by both methods[8]. However, it is similar to HMM that CTC relies on conditional independence assumptions, so training is not as stable as expected.

Received date: 2018-11-08
***E-mail:** chengcl@njupt.edu.cn

Attention has no such concerns, but attention performs poorly on sequence alignment problems due to lack of left-to-right constraints as used in CTC.

To address the problem of inconsistent target optimization of mixed systems, we propose a joint CTC-attention end-to-end speech recognition method with a triangular recurrent neural network (RNN) encoder. The key of this method is that we use a hierarchically decreasing triangular structure and utilize the CTC and attention mechanisms for joint decoding. The main contributions of this paper are as follows: RNN encoder with a triangular structure, and joint CTC-attention decoding. In the encoder network, the RNN with a triangular structure is used, and the RNN is composed of bidirectional long short term model (BLSTM) units which consist of four layers[9]. Starting from the second layer, the hidden layer state is read every two units, and the feature length can be reduced to 8 times of the original one. During decoding, we use multi-task learning framework in conjunction with CTC and attention for decoding.

1 Related Work

The emergence of neural networks makes us no longer rely on traditional HMM for acoustic modeling. RNN is applied to speech recognition because sequence problems can be easily solved. Then it is found that unidirectional RNN is still insufficient. Unidirectional RNN can only be modeled by using past information. Future information is equally important for speech decoding in speech recognition. To overcome such shortcoming, Sak et al.[9] proposed a bidirectional RNN, in which past and future information was comprehensively applied. The bidirectional RNN improved the performance of speech recognition systems. However, RNN also suffered gradient explosion and gradient disappearance. To solve this problem, Sak et al.[9] also proposed a long short-term memory (LSTM) model and combined it with bidirectional RNN. The results showed that the BLSTM was obviously superior to traditional unidirectional RNN.

In recent years, the general modeling method of sequence-to-sequence based on RNN has been favored by academia and successfully applied to machine translation and image annotation due to artificial intelligence. This type of modeling has also brought tremendous changes in speech recognition technology, resulting in an end-to-end speech recognition system. End-to-end speech recognition has challenged HMM as a dominant technology for speech recognition core technology. For end-to-end based LVCSR, Hannun et al.[10] proposed a new decoding method to eliminate the reliance of recognition systems on HMM. Two main end-to-end methods were currently available: CTC method, and encoder-decoder model based on attention mechanism.

Graves et al.[4] in 2006 firstly proposed CTC and used it to solve the problem of phoneme classification and handwritten digit recognition sequence marking. CTC is remarkably featured with the introductions of a “blank” symbol, which makes it possible for the same symbols to occur in the same line[5]. In Ref. [4], a system based on integration between deep BLSTM and CTC was proposed, and a modification to the objective function was introduced to train the network and minimize the expectation of an arbitrary transcription loss function. This allowed a direct optimization of the word error rate (WER), even in the absence of a lexicon or language model. It is also proved by the results of Ref. [5] on the benchmark dataset, TIMIT. However, dynamic programming that is used to calculate the label probability in CTC method is excessively dependent on conditional independence assumption, which makes the results not accurate as expected. Miao et al.[11] proposed an end-to-end model called EESEN. Miao et al. used the weighted finite state transducers (WFSTs) to combine CTC and language model (LM) for decoding[12], and achieved excellent results, -7.34% WER, on WSJ task.

Subsequently, attention mechanism, which was originally widely used for machine translation, has emerged. Encoder-decoder model based on attention mechanism was firstly applied to neural machine translation[6]. The model could realize the conversion between word sequences of different lengths in two languages, and automatically generate the alignment relationship between word sequences during recognition. Continuous speech recognition can be seen as a “translation” of speech features to phonemes (characters), which can be realized by means of the model[13]. Vaswani et al.[7] introduced attention mechanism into ASR. Vaswani et al. firstly proposed definition and gave a comprehensive introduction about attention. Chorowski et al.[13] applied attention to ASR and achieved 17.6% phoneme error ratio (PER) on TIMIT dataset. However, the results also showed that attention was sensitive to noise and was not suitable for recognition with long sentences. In Ref. [14], an attention model with LM was proposed. WFST was used to integrate end-to-end model with LM. The experiment was performed on WSJ dataset, achieving 11.3% WER and 4.8% character error rate (CER).

Although CTC and attention mechanism greatly promote the development of ASR, these methods still have their own limitations. For example, CTC relies on conditional independence assumptions and attention is sensitive to noise. Although traditional RNN structure can be used to solve the problem of sequence learning, past information can only be used while future information is ignored. Thus, the bidirectional RNN with triangular structure is adopted to accelerate the speed of feature extraction and to construct the network. In summary, this paper proposes an end-to-end speech recognition method which consists of an encoder with a triangular RNN and a decoder that works by CTC and attention.

2 Models

The end-to-end network can be divided into two sections: shared encoder, and joint decoder. In this model, the speech signal $\boldsymbol{x} = (x_1, x_2, \cdots, x_T)$ is received by the encoder and treated as an input feature. It outputs a corresponding text $\boldsymbol{y} = (y_1, y_2, \cdots, y_S)$ by the decoder. The encoder processes raw speech input that has been received and obtains an advanced feature representation $\boldsymbol{h} = (h_1, h_2, \cdots, h_u)$, which is used for training and decoding by joint decoder. The architecture of the model is shown in Fig. 1.

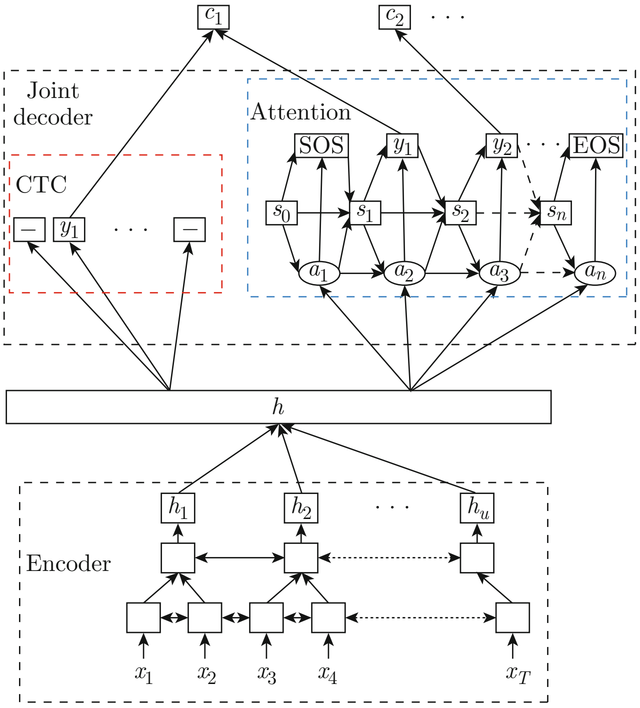

Fig. 1 Framework of the end-to-end model

As show in Fig. 1, the encoder consists of a BLSTM, where the BLSTM is featured in a triangular structure. The decoder includes CTC and attention mechanism that share the advanced features extracted by the encoder for decoding.

2.1 Encoder

The triangular BLSTM (tBLSTM) is used for encoder. The structure is used to reduce the length of an input. Since the speech signal input is usually hundreds of thousands of frames, direct application will directly cause low convergence. The triangle structure is used to reduce the length of an input. Due to too many input features, it is difficult for the decoder to extract the relevant information from numerous inputs, resulting in inaccurate results and further reducing the convergence speed. We solve this problem by using tBLSTM, and extract a feature between two units. The ith unit of the jth layer, h_i^j, can be expressed as

$$h_i^j = \text{tBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}]). \tag{1}$$

Four layers of tBLSTM are used in our model, because when the number of network layers is too large, the parameters of the model will increase rapidly, and the convergence speed will decrease. The convergence speed changes significantly with the increase of the model size and it is limited by graphical computational unit (GPU) memory[15]. Moreover, due to the lack of gradient attenuation of LSTM, such shortcoming will be greatly amplified when the number of layers continues to increase. Obviously, when the number of layers is too small, the number of features cannot be reduced. After being processed by the encoder, the extracted features are reduced by a factor of 8, allowing the joint decoder to extract relevant information more quickly and efficiently. In addition, deep network also makes the model even more complex and nonlinear. The structure of encoder with tBLSTM is shown in Fig. 1.

2.2 Decoder

2.2.1 *CTC*

CTC is an algorithm that can measure the similarity between input sequence and output sequence without a priori alignment information. The CTC-based speech recognition system consists of the layers of RNN and CTC. During the recognition process, RNN is used to calculate the posterior probability of each phoneme. The CTC-based model assumes that the frames output by the network are independent, and then the posterior probability of each sequence can be calculated from such hypothesis. On this basis, dynamic programming algorithm is further used by CTC to combine frame-based phoneme sequences into multiple phoneme sequences, and each phoneme sequence has a similarity score. Finally, the appropriate phoneme sequence is filtered by the decoding algorithm as an output result.

It is already known that CTC can train RNN without the alignment between input and output sequences. Each label (such as characters, and phonemes) plus a special symbol “blank” (used to indicate no output) is separately represented in the output layer. For an input sequence $\boldsymbol{x}$ of length T, the output vector $\boldsymbol{Y}_t$ is normalized using the softmax function and can be interpreted as the probability of a label (include the blank label) k appearing at the time t:

$$P(k, t|\boldsymbol{x}) = \exp(y_t^k) \Big/ \sum_{k'} \exp(y_t^{k'}), \tag{2}$$

where y_t^k represents the kth element of $\boldsymbol{Y}_t$. The alignment of the CTC, α, can be expressed as a tag index of length T, and then the probability $P(\alpha|\boldsymbol{x})$ is obtained by the product of the probabilities in each step:

$$P(\alpha|\boldsymbol{x}) = \prod_{t=1}^{T} P(\alpha_t, t|\boldsymbol{x}). \tag{3}$$

There is one such operation κ in a given sequence: firstly, removing repeated labels, and then deleting blank labels from alignment (e.g., κ(bb–a–b) = κ(b–aaab) = bab, where "–" denotes blank labels). Therefore, the total probability of the output sequence $\boldsymbol{y}$ can be expressed as the sum of the probabilities of all possible alignments corresponding to it:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\alpha\in\kappa^{-1}(\boldsymbol{y})} P(\alpha|\boldsymbol{x}). \tag{4}$$

Equation (4) can be calculated through dynamic programming method. For a target sequence $\boldsymbol{y}^*$, the network can be trained by minimizing the objective function of CTC:

$$\mathrm{CTC}(\boldsymbol{x}) = -\log P(\boldsymbol{y}^*|\boldsymbol{x}). \tag{5}$$

2.2.2 *Attention-Based Encoder-Decoder Model*

In Ref. [16], an attention-based recurrent sequence generator (ARSG) was proposed to be used as decoder. As an RNN, ASRG can randomly generate an output sequence $\boldsymbol{y}$ by using input $\boldsymbol{h}$. ARSG consists of an RNN and an attention mechanism. The attention mechanism selects a subsequence of the input sequence and uses it to update the hidden states of RNN and predict the next output.

In the step i, the ARSG generates output through $\boldsymbol{h}$:

$$\beta_i = \mathrm{Attend}(s_{i-1}, \beta_{i-1}, \boldsymbol{h}), \tag{6}$$

$$g_i = \sum_{j=1}^{U} \beta_{i,j} h_j, \tag{7}$$

$$y_i \sim \mathrm{Generate}(s_{i-1}, g_i), \tag{8}$$

where, s_{i-1} is the (i-1)th state of RNN, also called Generator; the Generate function indicates a feedforward network; $\beta_i \in \mathbf{R}^U$ means the weight of attention and it is also called alignment. The attention mechanism integrates all inputs $\boldsymbol{h}$ into g_i on the basis of the attention weight $\beta_i \in \mathbf{R}^U$. The steps are accomplished with computing new generator state as follows:

$$s_i = \mathrm{Recurrency}(s_{i-1}, g_i, y_i), \tag{9}$$

where the Recurrency function denotes a recurrent network. Attend function in Eq. (6) describes mixed attention. If the term β_{i-1} is dropped from Attend arguments, we call it content-based attention. The problem is that the elements identical or very similar to $\boldsymbol{h}$ are scored equally regardless of their positions in the sequence. In this paper, $\boldsymbol{h}$ has been dropped from Attend function, i.e., $\beta_i = \mathrm{Attend}(s_{i-1}, \beta_{i-1})$, called location-based attention. It can be found that it computes the alignment from the generator state and the previous alignment only.

Finally, based on the previous character $y_{1:u-1}$ and $\boldsymbol{h}$, the probability distribution of $\boldsymbol{Y}_u$ can be obtained, i.e., the loss function of attention:

$$\mathrm{Att}(\boldsymbol{x}) = -\log P(\boldsymbol{y}^*|\boldsymbol{x}) = \\ -\sum_u \log P(\boldsymbol{Y}_u^*|\boldsymbol{x}, y_{1:u-1}^*). \tag{10}$$

2.2.3 *Joint CTC-Attention Decoding*

Multitasking learning (MTL) is a machine learning method that is used to learn multiple related tasks together based on shared representation. A joint decoder based on the MTL framework is built in this paper. The network structure is shown in Fig. 1. The encoder consists of tBLSTM which receives speech signal input $\boldsymbol{x}$ and further converts the input into an advanced feature $\boldsymbol{h}$. The CTC and attention are used in the decoder for training. The forward-backward algorithm in the CTC can force monotonic alignment between speech and character sequence during training. In other words, it accelerates expected alignment estimation, rather than relying entirely on data-driven attention mechanism in long sequences. Our final objective function can be obtained by combining CTC objective function and attention objective function:

$$L_{\mathrm{hybrid}} = \lambda\mathrm{CTC}(\boldsymbol{x}) + (1-\lambda)\mathrm{Att}(\boldsymbol{x}) = \\ -\lambda\log P(\boldsymbol{y}^*|\boldsymbol{x}) - \\ (1-\lambda)\sum_u \log P(\boldsymbol{Y}_u^*|\boldsymbol{x}, y_{1:u-1}^*), \tag{11}$$

where λ is CTC weight, a tunable parameter ($0 < \lambda < 1$). Beam search is used for output-label synchronous decoding to perform the inference step of attention-based speech recognition. The joint decoding method combines the CTC and the attention-based sequence probability during inference and training, so that a better alignment hypothesis can be found for input speech, i.e., given the speech input $\boldsymbol{x}$, the decoder can find the most probable character sequence $\hat{C}$ by

$$\hat{C} = \underset{C\in U^*}{\arg\max}\,(L_{\mathrm{hybrid}}), \tag{12}$$

where C denotes the character, and U^* is the output set of possible character. During beam search, the decoder calculates the score for each partial hypothesis. This score $\alpha_{\mathrm{att}}(\cdot)$ can be calculated recursively by using the attention model:

$$\alpha_{\mathrm{att}}(g_l) = \alpha_{\mathrm{att}}(g_{l-1}) + \log P(c|g_{l-1}, \boldsymbol{x}), \tag{13}$$

where g_l is the partial hypothesis of length l, and c is the last character of g_l appended to g_{l-1}, i.e., $g_l = g_{l-1}c$.

During beam search, there is an important parameter: beam width which is generally predefined. It excludes assumptions with relatively low scores and significantly improves search efficiency.

However, integrating CTC and attention-based scores in beam search is also featured in some issues because the attention decoder is executed in character synchronization while CTC is executed in frame synchronization. In order to include the CTC probability to the score, rescoring method is adopted. The execution process of rescoring is divided into two steps. In the first step, the beam search is used to obtain a complete set of hypotheses, in which only the attention-based sequence probability is considered. In the second step, CTC and attention probabilities are used to redetermine complete hypothesis, where the CTC probability is obtained by the CTC forward algorithm.

3 Experiments

3.1 Setting

The VoxForge dataset is used in the experiment. CER is used as a criterion to evaluate system performance. The experimental result is trained by a single GPU. We use the 80-dimensional Mel-frequency cepstral coefficient (MFCC) feature with a pitch feature (a total of 83 dimensions) as an input feature of encoder. The encoder consists of a four-layer BLSTM with a triangular structure and there are 320 cells in each layer and direction. The top three layers of the encoder read hidden states every two layers in the network, reducing the length of feature to one eighth of the original one, which speeds up feature extraction and mitigates the effects of unrelated features on the experiment. The decoder consists of a single layer LSTM with 300 cells where a location-based attention mechanism is used and 10 central convolution filters with a width of 100 are used to extract the convolutional features. It is noted that no language models are used in the experiment. The AdaDelta algorithm with gradient clipping is used for optimization[16]. Model training is conducted by using ESPnet[17] toolkit and Kaldi[18] toolkit.

3.2 Results and Analysis

The parameter λ is critical for the results in this model. In order to find out the impact of λ in the model, the experiment of selecting λ is conducted. The value of λ in a step of 0.1 is adjusted in a range of 0 to 1 while other parameters are kept constant. For the value of each parameter, 10 experiments are conducted and their CERs are averaged. The results are shown in Fig. 2. From the figure we can see that the CER obtained at $\lambda = 0.3$ is the lowest. It is also found that λ is not too sensitive to the performance if we set λ at a value of 0.3. In addition, as λ increases, the performance of the model begins to decrease significantly, while as λ decreases, the CER does not significantly change, indicating that CTC plays an auxiliary role during training and attention has a dominant influence on the model.

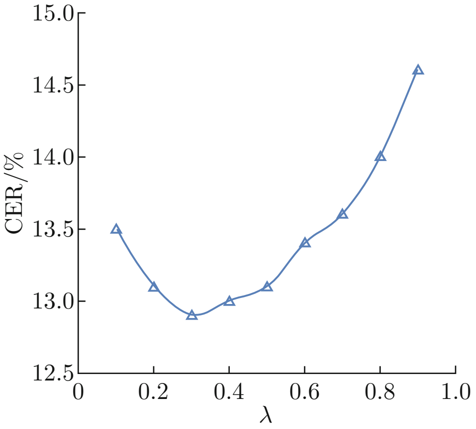

Fig. 2 Effect of CTC weight on CER

Subsequently, we experiment on the influence of the number of network layers in the encoder. The experimental results are shown in Table 1. It is found that that the performance can be significantly improved by increasing encoder layers; for example, CER can be reduced. As the network layers continue to increase, the performance of the model begins to decline. It may be because encoder is featured in a triangular structure. As the number of layers increases, the features associated with the results are eliminated and too few features are retained. Moreover, as the encoder layers continue to increase, the training time of the model increases dramatically. Therefore, it can be concluded that too deep network structure is unnecessary for training, because it consumes a lot of memory and time while failing to improve performance.

Table 1 Results of using multiple RNN layers in the encoder without LM

Encoder layers	CER
3-BLSTM	14.3
4-BLSTM	12.9
5-BLSTM	13.8
6-BLSTM	15.1

Finally, our approach is compared with traditional methods. The results are indicated in Table 2. The CERs obtained by using two different attention mechanisms are shown in Table 2. It is found that the location-based attention mechanism yields better results than content-based attention mechanism. It can also be intuitively seen from the table that the combined CTC-attention achieves 12.9% CER which is obviously better than the CER of independent CTC or independent attention models, and it is increased by

31% and 34% as compared with CTC and attention baseline, respectively. As compared with traditional mixed models, some gaps still need to be further improved. However, our method is superior due to simple structure and fast training.

Table 2 CER for the Voxforge dataset without LM

Model	CER/%
Attention, location-based	20.6
Attention, content-based	17.3
CTC	16.9
Attention and CTC	12.9
HMM/DNN	11.64

4 Conclusion

In this paper, an end-to-end CTC-attention speech recognition method is proposed against the inconsistent optimization goals in mixed systems. This method can be adopted to convert speech into text directly without predefined alignment. In addition, this model greatly simplifies network structure and can be seen as a single network. Therefore, by adopting this method, it is simpler to construct the model, and the amount of code and complexity is reduced. The entire model consists of an encoder-decoder network, and the encoder is constructed by a triangular structure of BLSTM, which reduces the impact of uncorrelated inputs in the model and speeds up feature extraction. The decoder combines CTC with attention. We experiment on the VoxForge dataset and achieve a CER of 12.9%. This result indicates that the joint decoding method does improve system performance.

References

[1] ANUSUYA M A, KATTI S K. Speech recognition by machine: A review [J]. *International Journal of Computer Science and Information Security*, 2009, **6**(3): 181-205.

[2] RABINER L R. A tutorial on hidden Markov models and selected applications in speech recognition [J]. *Proceedings of the IEEE*, 1989, **77**(2): 257-286.

[3] HINTON G, DENG L, YU D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups [J]. *IEEE Signal Processing Magazine*, 2012, **29**(6): 82-97.

[4] GRAVES A, FERNÁNDEZ S, GOMEZ F, et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks [C]//*23rd International Conference on Machine Learning*. Pittsburgh, Pennsylvania, USA: ACM, 2006: 369-376.

[5] GRAVES A, JAITLY N. Towards end-to-end speech recognition with recurrent neural networks [C]//*31st International Conference on Machine Learning*. Beijing, China: W&CP, 2014: 1764-1772.

[6] BAHDANAU D, CHO K H, BENGIO Y. Neural machine translation by jointly learning to align and translate [C]//*International Conference on Learning Representations*. San Diego, CA, USA: Computational and Biological Learning Society, 2015: 0473.

[7] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//*31st Conference on Neural Information Processing Systems*. Long Beach, CA, USA: NIPS, 2017: 5998-6008.

[8] MARKOVNIKOV N, KIPYATKOVA I, LYAKSO E. End-to-end speech recognition in Russian [C]//*International Conference on Speech and Computer*. Leizig, Germany: Springer, 2018: 377-386.

[9] SAK H, SENIOR A, BEAUFAYS F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition [C]//*15th Annual Conference of the International Speech Communication Association*. Singapore: ISCA, 2014: 1128.

[10] HANNUN A Y, MAAS A L, JURAFSKY D, et al. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs [EB/OL]. (2014-08-12) [2018-11-08]. https://arxiv.org/pdf/1408.2873.pdf.

[11] MIAO Y, GOWAYYED M, METZE F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding [C]//*IEEE Workshop on Automatic Speech Recognition and Understanding*. Scottsdale, AZ, USA: IEEE, 2015: 167-174.

[12] MOHRI M, PEREIRA F, RILEY M. Weighted finite-state transducers in speech recognition [J]. *Computer Speech & Language*, 2002, **16**(1): 69-88.

[13] CHOROWSKI J K, BAHDANAU D, SERDYUK D, et al. Attention-based models for speech recognition [C]//*29th Conference on Advances in Neural Information Processing Systems*. Montreal, Canada: NIPS, 2015: 577-585.

[14] BAHDANAU D, CHOROWSKI J, SERDYUK D, et al. End-to-end attention-based large vocabulary speech recognition [C]//*41st IEEE International Conference on Acoustics, Speech and Signal Processing*. Shanghai, China: IEEE, 2016: 4945-4949.

[15] LU L, ZHANG X, CHO K, et al. A study of the recurrent nerual network encoder-decoder for large vocabulary speech recognition [C]//*Proceedings of the Interspeech*. Dresden, Germany: ISCA, 2015: 3249-3253.

[16] ZEILER M D. Adadelta: An adaptive learning rate method [EB/OL]. (2012-12-22) [2018-11-08]. https://arxiv.org/pdf/1212.5701.pdf.

[17] WATANABE S, HORI T, KARITA S, et al. ESPnet: End-to-end speech processing toolkit [C]//*Proceedings of the Interspeech*. Hyderabad, India: ISCA, 2018: 2207-2211.

[18] POVEY D, GHOSHAL A, BOULIANNE G, et al. The Kaldi speech recognition toolkit [C]//*IEEE Workshop on Automatic Speech Recognition and Understanding*. Hawaii, USA: IEEE, 2011: 1-4.