Online Hybrid CTC/Attention Architecture for End-to-End Speech Recognition

Conference Paper · September 2019

DOI: 10.21437/Interspeech.2019-2018

INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Online Hybrid CTC/Attention Architecture for End-to-end Speech Recognition

Haoran Miao[1,2], *Gaofeng Cheng*[1,2], *Pengyuan Zhang*[1,2], *Ta Li*[1,2], *Yonghong Yan*[1,2,3]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, China
[2]University of Chinese Academy of Sciences, China
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, China

{miaohaoran, chenggaofeng, zhangpengyuan, lita, yanyonghong}@hccl.ioa.ac.cn

Abstract

The hybrid CTC/attention end-to-end automatic speech recognition (ASR) combines CTC ASR system and attention ASR system into a single neural network. Although the hybrid CTC/attention ASR system takes the advantages of both CTC and attention architectures in training and decoding, it remains challenging to be used for streaming speech recognition for its attention mechanism, CTC prefix probability and bidirectional encoder. In this paper, we propose a stable monotonic chunkwise attention (sMoChA) to stream its attention branch and a truncated CTC prefix probability (T-CTC) to stream its CTC branch. On the acoustic model side, we utilize the latency-controlled bidirectional long short-term memory (LC-BLSTM) to stream its encoder. On the joint CTC/attention decoding side, we propose the dynamic waiting joint decoding (DWDJ) algorithm to collect the decoding hypotheses from the CTC and attention branches. Through the combination of the above methods, we stream the hybrid CTC/attention ASR system without much word error rate degradation.

Index Terms: speech recognition, CTC, attention, online decoding

1. Introduction

End-to-end speech recognition system has shown promising performance, making it competitive to conventional hybrid system on large scale automatic speech recognition (ASR) tasks. End-to-end system integrates acoustic model, lexicon and language model, which directly converts acoustic features into target labels. Two mainstream frameworks are applied in end-to-end speech recognition field. One is characterized by the frame synchronous prediction, that is, one target label per input frame. Connectionist temporal classification (CTC) [1] loss function has always been employed to train the frame synchronous models. The other one is featured by the label synchronous prediction, that is, the ASR model decides when to output a target label. The attention-based encoder-decoder architecture [2, 3] is widely used in such framework, where the attention determines which encoder features should be attended to.

So far, several modifications have been proposed to enhance the performance of CTC framework and attention-based encoder-decoder framework. To remove the conditional independence assumption of CTC, some works have already incorporated attention mechanism within the CTC framework [4, 5, 6]. The Listen, Attend and Spell (LAS) [7] applied pyramid BLSTM to make it easier for the attention to model broader input context from the sub-sampled features. Furthermore, multi-head attention [8, 9], self-attention networks [10, 11] and other sophisticated attentions were also introduced. Recently, hybrid CTC/attention architecture has been proposed [12] to combine CTC and attention frameworks into a single neural network. In this architecture, the CTC branch will guide the attention to perform monotonic alignments, and thus the joint of CTC and attention can produce high-quality hypotheses.

Currently, most competitive end-to-end ASR systems are not suitable for online ASR tasks due to bidirectional encoder networks and global attention mechanism. In terms of hybrid CTC/attention architecture, both the CTC and attention branches perform in an offline way in joint CTC/attention decoding [13]. Fortunately, there have been some works focusing on low latency bidirectional acoustic modeling [14, 15] and online attentions such as Hard Monotonic Attention [16] and Monotonic Chunkwise Attention (MoChA) [17]. In addition, some online end-to-end ASR models [17, 18, 19] have also been proposed recently.

This work is the first attempt to stream the hybrid CTC/attention architecture. First, we find that standard MoChA is unstable in our system, and thus we propose a stable MoChA (sMoChA), which changes the way of computing attention weights, so as to replace the global attention. Second, we leverage the CTC-based network to segment audio and compute truncated CTC (T-CTC) prefix probability on the segmented audio rather than on the complete audio. This study illustrates that T-CTC prefix probability is a close approximation to the original CTC prefix probability [20]. After streaming both the CTC and attention branches, we design Dynamic Waiting Joint Decoding (DWJD) algorithm to deal with the problem that these two branches predict labels asynchronously in beam search. Finally, this work implements an online hybrid CTC/attention architecture and conducts experiments on LibriSpeech. Compared with the offline hybrid CTC/attention architecture, the degradation in our online hybrid CTC/attention architecture is 1.8%/3.3% absolute word error rate on test-clean/test-other.

2. Related works

2.1. Hybrid CTC/attention architecture

The hybrid CTC/attention architecture is composed of encoder, attention-based decoder and CTC-based network. Given the T-length acoustic features $X = \{x_1, \cdots, x_T\}$, the encoder produces the U-length encoder features $H = \{h_1, \cdots, h_U\}$ ($U \leq T$). The attention-based decoder receives H and predicts target labels step by step in the attention branch. The following

http://dx.doi.org/10.21437/Interspeech.2019-2018

formulas describe the specific process:

$$
\begin{aligned}
H &= \text{Encoder}(X), &(1)\\
c_i &= \text{Attention}(s_{i-1}, H), &(2)\\
y_i &\sim \text{Decoder}(s_{i-1}, y_{i-1}, c_i), &(3)
\end{aligned}
$$

where $s_i,\ c_i,\ y_i$ are the hidden state, "context" vector and output label in the attention-based decoder. Considering the L-length label sequence $Y = \{y_1, \cdots, y_L\}$, the CTC and attention branches will compute sequence posteriors $P_{\text{ctc}}(Y|X)$ and $P_{\text{att}}(Y|X)$ respectively. Lastly, the training objective of hybrid CTC/attention architecture is defined as follows:

$$L = \lambda \text{log} P_{\text{ctc}}(Y|X) + (1-\lambda)\text{log}P_{\text{att}}(Y|X), \qquad (4)$$

where λ is a hyperparameter satisfying $0 \leq \lambda \leq 1$.

2.2. Monotonic Chunkwise Attention (MoChA)

MoChA aims to learn a monotonic alignment between the encoder features H and the label sequence Y. Besides, MoChA performs local attention within a w-length chunk. Algorithm 1 [17] summarizes how MoChA computes the "context" vector c_i in decoding stage. t_i is the location where the chunk stops at when predicting label y_i. Lines 3-13 enforce the monotonic behavior through keeping t_i moving forward. Specifically, $p_{i,j}$ in line 4 is defined as the probability of selecting h_j for y_i. In lines 6-9, MoChA will perform local attention from h_{j-w+1} to h_j if $p_{i,j} \geq 0.5$.

Algorithm 1 MoChA Decoding Algorithm

```
Input: encoder features $H = \{h_1, \cdots, h_U\}$, output index $i$, decoder
    hidden state $s_i$, output label $y_i$, endpoint $t_i$, sigmoid function $\sigma(\cdot)$,
    attention chunk width $w$
 1: Initialize $s_0 = \mathbf{0},\ y_0 = \langle sos \rangle,\ t_0 = 1,\ i = 1$
 2: while $y_{i-1} \neq \langle eos \rangle$ do
 3:     for $j = t_{i-1}$ to $U$ do
 4:         $p_{i,j} = \sigma(\text{Energy}(s_{i-1}, h_j))$
 5:         if $p_{i,j} \geq 0.5$ then
 6:             for $k = j - w + 1$ to $j$ do
 7:                 $u_{i,k} = \text{ChunkEnergy}(s_{i-1}, h_k)$
 8:             end for
 9:             $c_i = \sum_{k=j-w+1}^{j} \frac{\exp(u_{i,k})}{\sum_{l=j-w+1}^{j} \exp(u_{i,l})} h_k$
10:             $t_i = j$
11:             break
12:         end if
13:     end for
14:     if $p_{i,j} < 0.5,\ \forall j \in \{t_{i-1}, \cdots, U\}$ then
15:         $c_i = \mathbf{0}, \quad t_i = t_{i-1}$
16:     end if
17:     $y_i \sim \text{Decoder}(s_{i-1}, y_{i-1}, c_i), \quad i = i + 1$
18: end while
```

However, Algorithm 1 cannot fit into backpropagation framework. We have to compute the expected value of c_i in training stage in accordance with Algorithm 2 [17]. To simulate the behavior of moving forward, the expected probability $\alpha_{i,j}$ of selecting h_j for y_i is provided in line 5 of Algorithm 2.

The choices for Energy and ChunkEnergy functions are

$$\text{Energy}(s_{i-1}, h_j) = g\frac{v^\top}{||v||}\tanh(W_s s_{i-1} + W_h h_j + b) + r, \qquad (5)$$

$$\text{ChunkEnergy}(s_{i-1}, h_j) = v^\top \tanh(W_s s_{i-1} + W_h h_j + b), \qquad (6)$$

where g, r are scalars, v, b are vectors and W_s, W_h are matrices. All of them are learnable parameters.

Algorithm 2 MoChA Training Algorithm

```
Input: encoder features $H = \{h_1, \cdots, h_U\}$, output index $i$, decoder
    hidden state $s_i$, output label $y_i$, sigmoid function $\sigma(\cdot)$, attention
    chunk width $w$, Gaussian noise $\epsilon$
 1: $s_0 = \mathbf{0},\ y_0 = \langle sos \rangle,\ \alpha_{0,0} = 1,\ \alpha_{0,k} = 0 (k \neq 0),\ i = 1$
 2: while $y_{i-1} \neq \langle eos \rangle$ do
 3:     for $j = 1$ to $U$ do
 4:         $p_{i,j} = \sigma(\text{Energy}(s_{i-1}, h_j) + \epsilon)$
 5:         $\alpha_{i,j} = p_{i,j} \sum_{k=1}^{j} \left( \alpha_{i-1,k} \prod_{l=k}^{j-1} (1 - p_{i,l}) \right)$
 6:     end for
 7:     for $j = 1$ to $U$ do
 8:         $u_{i,j} = \text{ChunkEnergy}(s_{i-1}, h_j)$
 9:         $\beta_{i,j} = \sum_{k=j}^{j+w-1} \frac{\alpha_{i,k} \exp(u_{i,j})}{\sum_{l=k-w+1}^{k} \exp(u_{i,l})}$
10:     end for
11:     $c_i = \sum_{j=1}^{U} \beta_{i,j} h_j$
12:     $y_i \sim \text{Decoder}(s_{i-1}, y_{i-1}, c_i), \quad i = i + 1$
13: end while
```

2.3. Latency-controlled BLSTM (LC-BLSTM)

LC-BLSTM is designed to reduce the redundant computation in Context-sensitive-chunk BPTT [14]. To be specific, the input sequence is firstly split into chunks of fixed length N_c. Then, N_r future frames are concatenated after each chunk as the right context. For each chunk, the hidden states of forward LSTM are copied from the previous chunk, and the hidden states of reversed LSTM are provided by the right context instead of the complete future context. Therefore, the latency of LC-BLSTM is limited to $N_c + N_r$ frames [21].

3. Online hybrid CTC/attention architecture

3.1. Stable Monotonic Chunkwise Attention (sMoChA)

In Section 2.2, it can be found that c_i in decoding stage is not equivalent to that in training stage unless $p_{i,j}$ is discrete in Algorithm 2, i.e. $p_{i,j} \in \{0, 1\}$. It is called the mismatch between the training and decoding scenario. To alleviate this problem, MoChA enforces $p \approx 0$ or $p \approx 1$ through initializing r to a negative or positive value [16] in Eq 5. However, we find that this initialization strategy will cause attention weights to attenuate to zeros quickly. We can rewrite $\alpha_{i,j}$ in detail as the following form:

$$\alpha_{i,j} = p_{i,j} \left(\frac{1 - p_{i,j-1}}{p_{i,j-1}} \alpha_{i,j-1} + \alpha_{i-1,j} \right). \qquad (7)$$

Obviously, the series $\{\alpha_{i,\cdot}\}$ exponentially decays by $(1 - p)$, and thus the attention weights will attenuate to zero along index j when $p \approx 1$. Similarly, the series $\{\alpha_{\cdot,j}\}$ exponentially decays by p, so the attention weights will attenuate to zero along index i when $p \approx 0$. Consequently, there exists a dilemma between discreteness and stability in standard MoChA.

To solve this problem, we propose sMoChA, dropping the term of $\alpha_{i-1,j}$ and computing expected selection probabilities as following:

$$\alpha_{i,j} = p_{i,j} \prod_{k=1}^{j-1} (1 - p_{i,k}). \qquad (8)$$

We initialize r to a negative value to ensure the discreteness of $p_{i,j}$ and prevent the series $\{\alpha_{i,\cdot}\}$ from decaying dramatically. In addition, we also evaluate the stability and discreteness of standard MoChA and our sMoChA in Section 4.2.

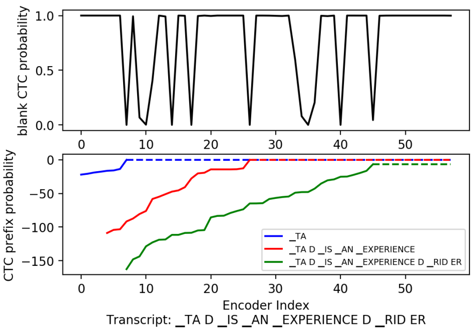

Figure 1: *Comparison between full and truncated CTC prefix probability: Upper figure is an example of CTC blank symbol probability distribution. Lower figure shows the dynamic summation of CTC prefix probability in the logarithmic domain. Solid line represents the computed part and dotted line represents the omitted part. The intersection of the solid and dotted line is the truncation point.*

3.2. Truncated CTC (T-CTC) prefix probability

Joint CTC/attention decoding [13] is applied to boost the performance in hybrid CTC/attention architecture. The CTC probabilities are considered to effectively exclude the incorrect hypotheses generated by the attention branch. In beam search, we refer to the prefix of the final hypothesis as partial hypothesis and denote it as l. The score for l in joint CTC/attention decoding is defined by:

$$\mathrm{score}(l) = \lambda \mathrm{log} P_{\mathrm{ctc}}(l|H) + (1-\lambda)\mathrm{log} P_{\mathrm{att}}(l|H). \quad (9)$$

In beam search, the cumulative CTC probability of all label sequences that take l as their prefix is used to score l, which is called CTC prefix probability [20] and can be calculated by:

$$P_{\mathrm{ctc}}(l, ...|H) = \sum_{\nu} P_{\mathrm{ctc}}(l \cdot \nu|H) = \sum_{t=1}^{U} P_{\mathrm{ctc}}(l|H_{[1:t]}), \quad (10)$$

where ν is an arbitrary suffix string. $P_{\mathrm{ctc}}(l, ...|H)$ is a substitute for $P_{\mathrm{ctc}}(l|H)$ in Eq 10, yet it depends on the whole frames, hampering the online decoding. To remove the global dependency, we can leverage peaky CTC probability distribution to segment encoder features for l. We propose T-CTC prefix probability to truncate the summation in Eq 10 as follows:

$$P_{\mathrm{ctc}}(l|H) \approx \sum_{t=1}^{k} P_{\mathrm{ctc}}(l|H_{[1:t]}), \quad (11)$$

where the upper bound of summation k satisfies that the CTC branch finishes generating l at h_k. Therefore, $p(\langle b\rangle|h_{k-1}) \approx 0$ and $p(\langle b\rangle|h_k) \approx 1$ in the CTC probability distribution, where $\langle b\rangle$ represents the blank symbol. We need to trace k as each hypothesis expands. Additionally, Figure 1 demonstrates that T-CTC prefix probability is a reasonable approximation to CTC prefix probability.

3.3. Dynamic Waiting Joint Decoding (DWJD)

Since there is no theoretical guarantee for the synchronous prediction of the online attention and CTC branches, we propose Dynamic Waiting Joint Decoding algorithm, in which the two online branches wait for each other when generating the partial hypothesis. This algorithm is presented in Algorithm 3.The inner loop in lines 4-22 computes the necessary encoder features when predicting the next label y_i. In lines 6-11, t_{att} records the location where the attention chunk starts off and doAtt decides whether to perform attention or not. In lines 12-17, t_{ctc} records the truncation point as discussed in Section 3.2, and doCTC determines whether to compute CTC probability or not. To wait for the slower branch, the encoder index j is always set to the minimum of t_{att} and t_{ctc}.

Algorithm 3 Dynamic Waiting Joint Decoding (DWJD)

Input: acoustic features $X = \{x_1, x_2, \cdots\}$, output index i, encoder index j, encoder feature h_j, encoder hidden state s_i, output label y_i, start points t_{att} and t_{ctc}, Boolean variables doAtt and doCTC, sigmoid function $\sigma(\cdot)$, threshold θ

```
1:  $s_0 = \mathbf{0},\ y_0 = l = \langle sos\rangle,\ t_{\mathrm{att}} = t_{\mathrm{ctc}} = 1,\ j = i = 1,\ \theta = 0.5$
2:  while $y_{i-1} \neq \langle eos\rangle$ do
3:      doAtt $= true$, doCTC $= true$, $j = \min\{t_{\mathrm{att}},\ t_{\mathrm{ctc}}\}$
4:      while $h_{j-1}$ is not the last encoder feature do
5:          $h_j = \mathrm{Encoder}(X)$
6:          if $j \geq t_{\mathrm{att}}$ and doAtt then
7:              $p_{i,j} = \sigma(\mathrm{Energy}(s_{i-1}, h_j))$
8:              if $p_{i,j} \geq 0.5$ then
9:                  $t_{\mathrm{att}} = j$,   doAtt $= false$
10:             end if
11:         end if
12:         if $j > t_{\mathrm{ctc}}$ and doCTC then
13:             Compute $p(\langle b\rangle|h_j)$
14:             if $p(\langle b\rangle|h_{j-1}) < \theta$ and $p(\langle b\rangle|h_j) \geq \theta$ then
15:                 $t_{\mathrm{ctc}} = j$,   doCTC $= false$
16:             end if
17:         end if
18:         if doAtt $= false$ and doCTC $= false$ then
19:             break
20:         end if
21:         $j = j + 1$
22:     end while
23:     if doCTC $= true$ then
24:         $t_{\mathrm{ctc}} = j - 1$
25:     end if
26:     $c_i = \mathrm{sMoChA}(s_{i-1}, H_{[1:t_{\mathrm{att}}]})$
27:     $y_i \sim \mathrm{Decoder}(s_{i-1}, y_{i-1}, c_i)$
28:     $l = l \cdot y_i$,   $i = i + 1$
29:     Compute $P_{\mathrm{ctc}}(l|H_{[1:t_{\mathrm{ctc}}]}),\ P_{\mathrm{att}}(l|H_{[1:t_{\mathrm{att}}]})$
30: end while
```

In our experiments, we also utilize a separately trained LSTM language model [22], together with $P_{\mathrm{ctc}}(l|H_{[1:t_{\mathrm{ctc}}]})$ and $P_{\mathrm{att}}(l|H_{[1:t_{\mathrm{att}}]})$ computed in Algorithm 3, to score the partial hypothesis l in beam search.

4. Experiments

Our experiments are conducted on LibriSpeech, one 1000-hour reading English speech corpus. All models are trained on the 960-hour train set. The valid set consists of dev-clean and dev-other. Word error rates (WER) are reported on test-clean and test-other respectively. Our baseline and online model are based on ESPNet [23]. All experiments employ 83-dimensional features, including 80 filter banks, pitch, delta-pitch and Normalized Cross-Correlation Function (NCCF), computed with a 25ms window and shifted every 10ms. Besides, we choose subwords [24] as output labels to solve the out-of-vocabulary problem. A 5000-sized subwords set is achieved by subword segmentation algorithm based on a unigram language model [25]. We utilize this subwords set both in hybrid CTC/attention architecture and external language model.

4.1. Baseline

In our baseline, the encoder contains 2 blocks of VGG layer [26] followed by a 5-layer BLSTM. The downsampling rate of 2 VGG blocks is one quarter. The decoder is a 2-layer LSTM, which receives ground-truth label as the previous prediction in training stage. The location-aware attention mechanism [3] is used for better performance. The dimension of hidden states in LSTM and attention is 1024. The external language model contains a single layer LSTM, trained on the normalized LM training text of LibriSpeech. All experiments use the same external language model in decoding stage. The WERs of the baseline are 4.2% and 13.4% on test-clean and test-other respectively, which are listed in the first line of Table 1.

4.2. Streaming CTC/attention with sMoChA and T-CTC

Our first work is to evaluate the performance of the MoChA and sMoChA. We replace the location-aware attention in our baseline with the MoChA and sMoChA respectively, finding that the attention weights computed according to standard MoChA easily attenuate to zeros in various configurations, while the attention weights computed by sMoChA are much more stable. In Figure 2(b), the attention weights in standard MoChA decay quickly along decoder index when initializing r to a negative value as discussed in Section 3.1. The attention chunk width of sMoChA is 3 and the initial bias r in Eq 5 is -4. The WERs of sMoChA model are 4.7% and 13.6% on test-clean and test-other respectively, as listed in line 3 of Table 1.

We depict the attention weights learned by different attention mechanisms in Figure 2 for comparison. By comparing Figure 2(c) and Figure 2(d), we find that there is little difference in the attention weights between decoding and training, proving the discreteness of our sMoChA. Comparing Figure 2(a) and Figure 2(c), the attention weights in Figure 2(c) is constrained in a small chunk regardless how long the subwords are pronounced, which is the main cause for the accuracy loss.

Second, we examine the performance of T-CTC prefix probability in joint CTC/attention decoding. T-CTC prefix probability results in 0.1%/0.2% absolute degradation, comparing lines 1 and 2 in Table 1. By the combination of sMoChA and T-CTC prefix probability, our system achieves 4.8%/13.9% WERs without much degradation, as listed in line 4 of Table 1.

4.3. Streaming CTC/attention with VGG-LC-BLSTM

We replace the VGG-BLSTM in our baseline with VGG-LC-BLSTM, keeping the rest unchanged. Specifically, we train the VGG-LC-BLSTM model initialized by our baseline, rather than from scratch. The chunk length N_c is 32 and the right context length N_r is 16 in LC-BLSTM. The WERs of VGG-LC-BLSTM model are 5.4%/16.4% on test-clean/other, which can be found in line 5 of Table 1.

Finally, our online model consists of VGG-LC-BLSTM encoder, sMoChA-based decoder and the CTC-based network. We initialize the VGG-LC-BLSTM networks by our baseline. DWJD algorithm is applied to realize an online decoding process. The WERs of our online model are 6.0%/16.7% on test-clean/other, as shown in the last line of Table 1. Lines 1, 5 and 7 indicate that the VGG-LC-BLSTM brings about 1.2%/3.0% absolute degradation while sMoChA and T-CTC prefix probability lead to 0.6%/0.3% absolute degradation in total. The comparison with other published online models [27] on LibriSpeech is also provided in Table 1.

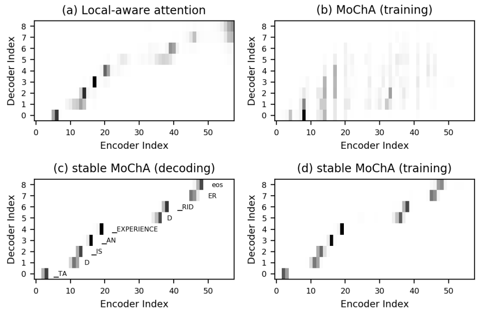

Figure 2: *Attention weights visualization in different attention. It should be noted that MoChA and sMoChA perform attention on a chunk in decoding stage, yet perform attention on the whole frames in training stage.*

Table 1: *Word error rate (WER) of different models: We conduct experiments on different encoders (Enc) including VGG-BLSTM, VGG-LSTM and VGG-LC-BLSTM, with different attention mechanisms (Att) including location-aware (Loc) and sMoChA. In joint CTC/attention decoding, we use CTC or T-CTC prefix probability. The various combinations are detailed in the second column.*

NO.	Model (Enc+Att+CTC)	Test-clean	Test-other
1	VGG-BLSTM + Loc + CTC	4.2	13.4
2	VGG-BLSTM + Loc + T-CTC	4.3	13.6
3	VGG-BLSTM + sMoChA + CTC	4.7	13.6
4	VGG-BLSTM + sMoChA + T-CTC	4.8	13.9
5	VGG-LC-BLSTM + Loc + CTC	5.4	16.4
6	VGG-LC-BLSTM+sMoChA+CTC	5.8	16.4
7	VGG-LC-BLSTM+sMoChA+T-CTC	6.0	16.7
8	ConvNet + CTC [27]	5.1	16.0
9	ConvNet + ASG [27]	4.8	14.5

5. Conclusions

In this paper, we propose the sMoChA and T-CTC prefix probability for computing attention weights and decoding utterances in an online manner. Moreover, we put forward DWJD algorithm in joint CTC/attention decoding and design an online hybrid CTC/attention architecture for end-to-end speech recognition, consisting of VGG-LC-BLSTM encoder, sMoChA-based decoder and the CTC-based network. Our hybrid CTC/attention architecture can be online with a 1.8%/3.3% absolute degradation of WERs on test-clean/other of LibriSpeech. The experiments also show that the major performance loss is due to VGG-LC-BLSTM encoder while the degradation caused by sMoChA and T-CTC prefix probability is as little as 0.6%/0.3/% absolute WERs. Therefore, our sMoChA and T-CTC prefix probability are reliable online methods. Research on better low-latency encoder networks will be included in our future work.

6. Acknowledgements

This work is partially supported by the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFB0801200), the National Natural Science Foundation of China (Nos. 11590774, 11590770), the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No.2016A03007-1), the Pre-research Project for Equipment of General Information System (No.JZX2017-0994/Y306), and cooperation project with the Sony (China) Limited.

7. References

[1] A. Graves, S. Fernndez, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine Learning*, 2006.

[2] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," *Eprint Arxiv*, 2014.

[3] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Computer Science*, vol. 10, no. 4, pp. 429–439, 2015.

[4] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping." in *Interspeech*, 2017, pp. 1298–1302.

[5] L. Dong, S. Zhou, C. Wei, and X. Bo, "Extending recurrent neural aligner for streaming end-to-end speech recognition in mandarin," 2018.

[6] A. Das, J. Li, Z. Rui, and Y. Gong, "Advancing connectionist temporal classification with attention modeling," 2018.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International Conference on Acoustics*, 2016.

[8] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, and K. Gonina, "State-of-the-art speech recognition with sequence-to-sequence models," 2018.

[9] T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Multi-head decoder for end-to-end speech recognition," *arXiv preprint arXiv:1804.08050*, 2018.

[10] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[12] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[13] T. Hori, S. Watanabe, and J. Hershey, "Joint ctc/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 518–529.

[14] C. Kai and H. Qiang, "Training deep bidirectional lstm acoustic model for lvcsr by a context-sensitive-chunk bptt approach," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 24, no. 7, pp. 1185–1193, 2016.

[15] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5755–5759.

[16] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2837–2846.

[17] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[18] R. Fan, P. Zhou, W. Chen, J. Jia, and G. Liu, "An online attention-based model for speech recognition," *arXiv preprint arXiv:1811.05247*, 2018.

[19] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for asr using self-attention network and chunk-hopping," *arXiv preprint arXiv:1902.06450*, 2019.

[20] K. Kawakami, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Ph. D. thesis, Technical University of Munich, 2008.

[21] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[22] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," *arXiv preprint arXiv:1706.02737*, 2017.

[23] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[24] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[25] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," *arXiv preprint arXiv:1804.10959*, 2018.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] V. Liptchinsky, G. Synnaeve, and R. Collobert, "Based speech recognition with gated convnets," *arXiv preprint arXiv:1712.09444*, 2017.