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

Compression of CTC-Trained Acoustic Models by Dynamic Frame-Wise Distillation Or Segment-Wise N-Best Hypotheses Imitation

Haisong Ding[1,2], *Kai Chen*[2], *Qiang Huo*[2]

[1]University of Science and Technology of China, Hefei 230026, China
[2]Microsoft Research Asia, Beijing 100080, China

dinghs11@mail.ustc.edu.cn, {kaic,qianghuo}@microsoft.com

Abstract

Knowledge distillation (KD) has been widely used for model compression by learning a simpler student model to imitate the outputs or intermediate representations of a more complex teacher model. The most commonly used KD technique is to minimize a Kullback-Leibler divergence between the output distributions of the teacher and student models. When it is applied to compressing CTC-trained acoustic models, an assumption is made that the teacher and student share the same frame-wise feature-transcription alignment, which is usually not true due to the topology difference of the teacher and student models. In this paper, by making more appropriate assumptions, we propose two KD methods, namely dynamic frame-wise distillation and segment-wise N-best hypotheses imitation. Experimental results on Switchboard-I speech recognition task show that the segment-wise N-best hypotheses imitation outperforms the frame-level and other sequence-level distillation methods, and achieves a relative word error rate reduction of 5%-8% compared with models trained from scratch.

Index Terms: CTC-trained acoustic model, knowledge distillation, dynamic frame-wise distillation, segment-wise N-best hypotheses imitation

1. Introduction

Acoustic models based on long short-term memory (LSTM) [1] and trained with a connectionist temporal classification (CTC) criterion [2] have been one of the state-of-the-art solutions for large vocabulary continuous speech recognition (LVCSR) (e.g., [3–18]). To achieve high recognition accuracy, multiple LSTM layers must be used. To deploy deep LSTM-CTC based acoustic models in products, model compression is a useful technique to reduce both footprint and runtime latency.

Knowledge distillation (KD), also known as teacher student learning, is a widely used model compression technique for deep neural networks (DNNs) (e.g., [19–21]). KD tries to transfer knowledge from a complex teacher model to a simple student model, which outperforms a model with the same topology trained from scratch, therefore offers a good product solution. The idea of KD is first proposed in [19] to compress DNN in a DNN-HMM hybrid system by minimizing a Kullback-Leibler divergence (KLD) between the output distributions of the teacher and student models at each frame. This method has also been verified in e.g., [22–25].

There are several variants of KD when applied to CTC-trained acoustic models (e.g., [8, 26–28]). In [8], two frame-wise distillation methods are tried. The first one tries to match student's output distributions with teacher's using the same cross entropy (CE) criterion as in [19], and is referred thereafter as "Output-CE". The second method trains the student model with frame-level targets obtained by doing Viterbi alignment on transcription with the teacher model. A CE criterion is also used, but the target sequence represents the best alignment path, therefore we call this method "BestAlign-CE". Yet in [26], another alignment-based method is proposed. It forces the student to learn all correct alignment paths by minimizing a frame-wise CE loss between the output of student and the state occupation distribution obtained from doing a forward-backward calculation on transcription with the teacher model. Since the state occupation distribution of CTC can be treated as a soft alignment, we call this method "SoftAlign-CE". All the above three methods make an implicit assumption that teacher and student share the same frame-wise alignments, which is usually not true due to their topology differences. To handle this inconsistent-alignment issue, [27] proposes an improved knowledge distillation (IKD) method. For each frame, given a student output, it will first select a most similar teacher output from nearby frames through comparison, and then train the student to minimize the corresponding KLD. However, this strategy can not guarantee that the teacher and student target on the same transcription.

Different from the above frame-level methods, [28] investigates a sequence-level distillation method, which is first proposed in [29]. Given a sentence, its N-best hypotheses and their posterior probabilities are first extracted by a teacher model. Then a student model will be trained using the hypotheses under a sequence-level CE criterion. So we call this method "Sequence-CE". Obviously, Sequence-CE doesn't leverage the alignment information of the teacher model.

In this paper, we propose two novel approaches to transferring knowledge from a CTC-trained wide, deep bidirectional LSTM (DBLSTM) model to thin/shallow ones. Our first approach tries to solve the inconsistent-alignment problem at frame-level. Given a student output, we use dynamic time warping (DTW) to determine an aligned teacher output from nearby frames so that we can encourage the teacher and student to target on the same transcription. Since a well-trained CTC-based model can roughly segment modelling units from an input sentence, our second approach leverages it to improve KD result. We first use Viterbi alignment to split the teacher output into segments, then extract N-best hypotheses and their posterior probabilities for each segment. A student will be trained with these segment-level hypotheses similar to Sequence-CE. Experimental results on Switchboard-I show that this approach performs best among all KD methods we tried.

The remainder of this paper is organized as follows. We introduce the proposed approaches in Section 2 and compare them with other approaches in Section 3. We conclude the paper in Section 4.

This work was done when Haisong Ding was an intern in Speech Group, Microsoft Research Asia, Beijing, China.

http://dx.doi.org/10.21437/Interspeech.2019-2182

2. Our Approach

2.1. Dynamic Frame-wise Distillation

Given a CTC-trained teacher model, a student model and a K frame sentence $\mathbf{s}$, we denote $\mathcal{P}^{(T)} = (\mathbf{P}_1^{(T)}, \mathbf{P}_2^{(T)}, \cdots, \mathbf{P}_K^{(T)})$ and $\mathcal{P}^{(S)} = (\mathbf{P}_1^{(S)}, \mathbf{P}_2^{(S)}, \cdots, \mathbf{P}_K^{(S)})$ as their respective output posterior probabilities over a symbol set $\mathcal{A}' = \mathcal{A} \cup \{\_\}$, where $\mathcal{A}$ is a phone model set and "_" denotes the "blank" symbol in CTC criterion. The conventional KD is to minimize the KLD between $\mathcal{P}^{(T)}$ and $\mathcal{P}^{(S)}$ frame-wisely, which is equivalent to minimizing CE loss, so we call this method "Output-CE" and its loss function is as follows:

$$\mathcal{L}^{(\text{Output-CE})} = -\sum_{i=1}^{K}\sum_{v_i=1}^{|\mathcal{A}'|} P_{i,v_i}^{(T)} \log P_{i,v_i}^{(S)} \ . \tag{1}$$

Besides using teacher's output directly, student model could also be trained with targets obtained from Viterbi or forward-backward calculation on transcription with the teacher model as [8, 26] do. Targets obtained by Viterbi alignment represent the best alignment path, so its loss function is as follows:

$$\mathcal{L}^{(\text{BestAlign-CE})} = -\sum_{i=1}^{K}\sum_{v_i=1}^{|\mathcal{A}'|} t_{i,v_i}^{(T)} \log P_{i,v_i}^{(S)} \ , \tag{2}$$

where $t_{i,v_i}^{(T)} = 1$ if the i-th frame is aligned to the v_i-th symbol, otherwise it will be 0. For targets obtained by forward-backward calculation, they represent soft alignment results, so the loss function will be

$$\mathcal{L}^{(\text{SoftAlign-CE})} = -\sum_{i=1}^{K}\sum_{v_i=1}^{|\mathcal{A}'|} \sigma_{i,v_i}^{(T)} \log P_{i,v_i}^{(S)} \ , \tag{3}$$

where $\sigma_{i,v_i}^{(T)}$ represents accumulated occupation probabilities on the v_i-th symbol of all correct alignment paths at i-th frame.

All the above three methods assume that teacher and student models share the same frame-wise alignment between acoustic signal and transcription, which is often not held for CTC-trained models. Since CTC tries to maximize the accumulated probability of all correct alignment paths, different models may have different spiking behaviors as shown in Fig. 1. To handle this inconsistent-alignment issue, we propose a dynamic frame-wise distillation (DFD) method, which leverages DTW to find the best warping path between the output sequences of teacher and student.

In DTW, a warping path is defined as a sequence $\mathbf{w} = (w_1, w_2, \cdots, w_L)$ with each element $w_l = (s_l, t_l)$ meaning that $\mathbf{P}_{s_l}^{(S)}$ is aligned with $\mathbf{P}_{t_l}^{(T)}$. The DTW warping path must satisfy the following restrictions:

1. $w_1 = (1, 1)$ and $w_L = (K, K)$,
2. $s_1 \le s_2 \le \cdots \le s_L$ and $t_1 \le t_2 \le \cdots \le t_L$,
3. $w_{l+1} - w_l \in \{(0, 1), (1, 0), (1, 1)\}$ for $1 \le l < L$.

The warping cost function we used is CE in this paper. After the best warping path $\mathbf{w}^*$ is obtained, the student model is optimized by minimizing the following criterion:

$$\mathcal{L}^{(\text{DFD-CE})} = -\sum_{l=1}^{|\mathbf{w}^*|}\sum_{v_i=1}^{|\mathcal{A}'|} P_{t_l^*,v_i}^{(T)} \log P_{s_l^*,v_i}^{(S)} \tag{4}$$

where $(s_l^*, t_l^*) = w_l^*$.

Since the student model is randomly initialized, there is a risk that at the early stage of training process, a short segment of $\mathcal{P}^{(S)}$ is aligned with a pretty long segment of $\mathcal{P}^{(T)}$ or vice versa. This undesirable behavior is called a *pathological alignment problem* [31]. To avoid this, we enforce one more restriction on admissible DTW warping paths called *Sakoe-Chiba band* [30]:

$$|s_l - t_l| \le \tau \tag{5}$$

where τ is an integer called *windows length*. With τ, $\mathbf{P}_i^{(S)}$ is only allowed to match $\mathbf{P}_j^{(T)}$ where $j \in \{i - \tau, \cdots, i + \tau\} \cap \{1, 2, \cdots, K\}$. With τ, the computation cost of finding the best warping path can also be greatly reduced. It is worth noting that $\mathcal{L}^{(\text{DFD-CE})}$ will be equivalent to $\mathcal{L}^{(\text{Output-CE})}$ if we set $\tau = 0$.

A similar idea of selecting teacher's outputs from nearby frames as student's distillation targets was proposed in [27], which is called "IKD". Its main difference with ours is that no restriction except *windows length* is used, so its student will easily target on wrong transcription. As a result, IKD has to be used along with CTC, or it will fail to converge. As a comparison, the warping path obtained by our approach encourages the student to target on the same transcription with the teacher even without the guidance of CTC.

2.2. Segment-wise N-Best Hypotheses Imitation

Besides frame-level distillation, we can also transfer knowledge of CTC-trained models at segment-level. As shown in Fig. 1, a well-trained CTC model can roughly segment modelling units according to outputs. Inspired by this phenomenon, we propose a segment-wise N-Best hypotheses imitation (SegNBI) strategy.

Given a K-frame sentence, its Viterbi path $\boldsymbol{\pi}_{1,K} = (\pi_1, \pi_2, \cdots, \pi_K)$ on transcription is identified first with teacher outputs $\mathcal{P}^{(T)}$. Then $\boldsymbol{\pi}_{1,K}$ is split according to following rules:

1. If there is no "_" between two different non-blank symbols, split them directly. For example $\boldsymbol{\pi}_{1,5} = (\_, x, x, y, \_)$ will be split into $\boldsymbol{\pi}_{1,3} = (\_, x, x)$ and $\boldsymbol{\pi}_{4,5} = (y, \_)$;
2. If two non-blank symbols are separated by consecutive blank symbols, they will be split at the middle of "_"s. For example, $\boldsymbol{\pi}_{1,14} = (\_, x, x, \_, \_, \_, y, \_, \_, \_, \_, z, z, \_)$ will be split into $\boldsymbol{\pi}_{1,4} = (\_, x, x, \_)$, $\boldsymbol{\pi}_{5,5} = (\_)$, $\boldsymbol{\pi}_{6,8} = (\_, y, \_)$, $\boldsymbol{\pi}_{9,9} = (\_)$ and $\boldsymbol{\pi}_{10,14} = (\_, \_, z, z, \_)$.

After applying the above rules to $\boldsymbol{\pi}_{1,K}$, we will get M segments $\{\boldsymbol{\pi}_{a_1,b_1}, \boldsymbol{\pi}_{a_2,b_2}, \cdots, \boldsymbol{\pi}_{a_M,b_M}\}$, where $a_1 = 1, b_M = K, b_i \ge a_i, a_{i+1} = b_i + 1$.

For the i-th segment, first calculate its N-best hypotheses $\{H_i^{(1)}, \cdots, H_i^{(N)}\}$ with teacher's output in $[a_i, b_i]$. Then calculate their CTC posterior probabilities $\{P^{(*)}(H_i^{(1)}|\mathbf{s}, i), \cdots, P^{(*)}(H_i^{(N)}|\mathbf{s}, i)\}, * \in \{T, S\}$ given teacher's and student's output in $[a_i, b_i]$, respectively. Finally, train a student model by minimizing the following accumulated segment-wise CE:

$$\mathcal{L}^{(\text{SegNBI-CE})} = -\sum_{i=1}^{M}\sum_{n=1}^{N} \hat{P}^{(T)}(H_i^{(n)}|\mathbf{s}, i) \log P^{(S)}(H_i^{(n)}|\mathbf{s}, i) \ , \tag{6}$$

where

$$\hat{P}^{(T)}(H_i^{(j)}|\mathbf{s}, i) = \frac{P^{(T)}(H_i^{(j)}|\mathbf{s}, i)}{\sum_{n=1}^{N} P^{(T)}(H_i^{(n)}|\mathbf{s}, i)} \ . \tag{7}$$

Obviously if $M = K, N = |\mathcal{A}'|$, $\mathcal{L}^{(\text{SegNBI-CE})}$ will become $\mathcal{L}^{(\text{Output-CE})}$, while if $M = 1$, the student will try to distill N-best

Table 1: *WER (in %) of teacher and student models trained from scratch with CTC criterion.*

Model	Eval2000			RT03S
	swbd	callhm	full set	
5×800	12.6	24.5	18.6	22.0
5×400	13.2	26.2	19.8	22.6
3×800	13.3	26.7	20.1	23.1
3×400	14.6	28.5	21.6	24.3

hypotheses at sequence-level, which is the same as Sequence-CE in [28]. Compared with Sequence-CE, our approach leverages alignment information to distill at segment-level, which makes training easier. Moreover, generating segment-wise N-best hypotheses can take more hypotheses into account, therefore more knowledge can be transferred from teacher to student.

3. Experiments

3.1. Experimental Setup

We evaluate our methods on 309-hour Switchboard-I conversational telephone speech recognition task [32]. Five hours of speech are chosen as validation set. Frequently duplicated sentences are filtered out from the remaining 304-hour speech as EESEN does [9], resulting to a 286-hour training set. We use 2000 NIST Hub5 evaluation set (Eval2000, about 2 hours of speech, with "Switchhoard" (swbd) and "Callhome" (callhm) subsets) and Spring 2003 NIST rich transcription set (RT03S, about 6.3 hours of speech) as testing sets, and word error rate (WER) as performance metric. 40-dimensional filter bank features along with 3 pitch features are extracted every 10ms as raw features using Kaldi toolkit [33].

We choose a DBLSTM model with 5 BLSTM layers, each containing 800 (400 forward and 400 backward) memory cells with peephole connections [34], denoted as 5×800, as teacher model, and 3 smaller DBLSTM models as students: a shallow model 3×800, a thin model 5×400 and a shallow thin model 3×400. All BLSTM layers of these models are followed by a non-recurrent projection layer with 200 neurons with batch normalization (BN) [35]. Before fed into the first BLSTM layer, a raw feature sequence is normalized by BN and stacked every 3 consecutive frames as [7] does. Then the 129-dimension feature sequence is processed by a 200-neuron feed-forward layer. The output of the last BLSTM layer is processed first by a 200-neuron feed-forward layer, then a ReLU layer before fed into the last feed-forward layer, which has 43 output units corresponding to 42 mono-phones and 1 "blank".

During training, models are optimized using mini-batch stochastic gradient descent (SGD) with a momentum of 0.9, weight decay of 0.0004 and clipping gradient as 2 [36]. We also augment training data by duplicating the first frame of each sentence 0∼2 times, which leads to a 3× larger training set. CAFFE tool [37] is used to train all acoustic models. For decoding, we use "CMU dictionary" as lexicon and a 3-gram language model trained from Switchboard and Fisher [38] transcriptions. Our decoding recipes are the same as EESEN [9].

3.2. Experimental Results

3.2.1. Performance comparison of different acoustic models

Table 1 shows performances of teacher and student models trained from scratch with CTC criterion. All 3 student models perform worse than the teacher model. From Fig. 1, we can find that even for a sentence recognized correctly by all models, their peaking behaviors are different, while the phone units can be roughly segmented with the peaks. Next we try our proposed KD methods to leverage these phenomena to improve the performances of student models.

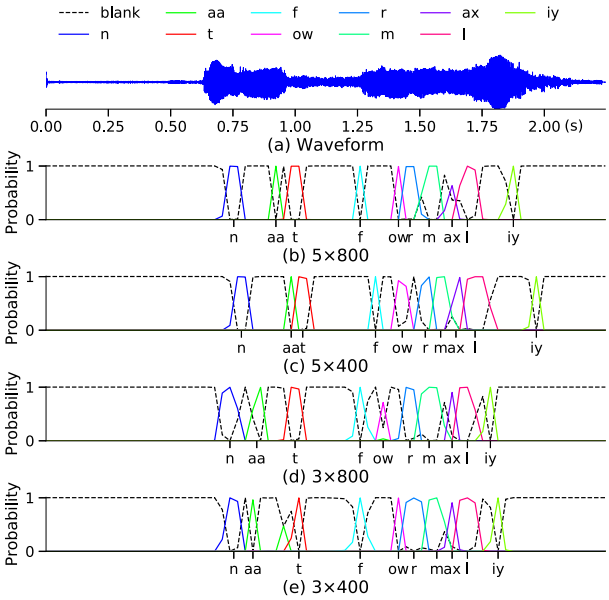

Figure 1: *Posterior distribution of phones in sentence "not formally" extracted from different DBLSTM models trained from scratch (best viewed in color).*

3.2.2. Effects of dynamic frame-wise distillation

We compare our dynamic frame-wise distillation method with other frame-wise methods in this part. The results are shown in Table 2. Two window lengths $\tau = 1, 2$ are tried for our "DFD-CE" method. Although IKD was proposed to transfer knowledge from BLSTMs to unidirectional LSTMs, we evaluated its performance of transferring from big DBLSTM to small one. Different from [27], for the i-th frame of a given sentence, we set $\tau = 1$ to make student model select teacher's outputs from the $\{i-1, i, i+1\}$-th frames instead of $\{i-5, \cdots, i\}$-th ones.

We also combine different KD criteria with CTC to improve students' performances:

$$\mathcal{L}^{(\text{combine})} = \alpha \mathcal{L}^{(\text{CTC})} + (1-\alpha)\mathcal{L}^{(\text{KD})} \ . \tag{8}$$

$\alpha = 0.1, 0.2, 0.3$ are tried and the results are listed in Table 3.

From Table 2, we find that Output-CE consistently outperforms BestAlign-CE and SoftAlign-CE over all student models. When combined with CTC, their performance margins become even larger. Since BestAlign-CE and SoftAlign-CE only consider correct path(s), combining with CTC is equivalent to re-weighting correct paths. As a comparison, Output-CE also takes competing paths into account, which helps improve performances of student models. Compared with other frame-wise based methods, DFD-CE with $\tau = 1$ performs best when no CTC is combined on 3×800 and 3×400 models, which shows that its alignment strategy helps learn the student model better. As Fig. 1 shows, the alignment of 5×400 is quite similar to that of 5×800, so DFD-CE with $\tau = 1$ performs similarly

Table 2: *WER (in %) of student models trained with different KD criteria.*

Model	Criterion	Eval2000			RT03S
		swbd	callhm	full set	
5×800	CTC	12.6	24.5	18.6	22.0
5×400	CTC	13.2	26.2	19.8	22.6
	Output-CE	12.8	**24.6**	18.8	21.6
	BestAlign-CE	13.1	24.9	19.1	21.9
	SoftAlign-CE	12.9	25.3	19.1	21.7
	IKD, $\tau = 1$	/	/	/	/
	DFD-CE, $\tau = 1$	12.9	**24.6**	18.8	21.6
	DFD-CE, $\tau = 2$	12.9	**24.6**	18.8	22.0
	Sequence-CE	**12.6**	24.9	18.8	22.0
	SegNBI-CE	12.7	**24.6**	**18.6**	**21.3**
3×800	CTC	13.3	26.7	20.1	23.1
	Output-CE	12.9	25.1	19.0	21.8
	BestAlign-CE	13.2	25.6	19.4	22.1
	SoftAlign-CE	12.9	25.4	19.2	22.1
	IKD, $\tau = 1$	/	/	/	/
	DFD-CE, $\tau = 1$	12.7	24.8	18.8	21.8
	DFD-CE, $\tau = 2$	13.1	25.3	19.2	21.8
	Sequence-CE	13.1	25.3	19.3	22.1
	SegNBI-CE	**12.6**	**24.6**	**18.7**	**21.6**
3×400	CTC	14.6	28.5	21.6	24.3
	Output-CE	14.4	27.5	21.0	23.9
	BestAlign-CE	14.6	27.6	21.1	24.0
	SoftAlign-CE	14.6	27.3	21.0	24.0
	IKD, $\tau = 1$	/	/	/	/
	DFD-CE, $\tau = 1$	14.4	27.0	20.7	23.9
	DFD-CE, $\tau = 2$	14.5	27.1	20.9	24.0
	Sequence-CE	14.3	27.1	20.7	23.6
	SegNBI-CE	**14.1**	**26.6**	**20.4**	**23.4**

Table 3: *WER (in %) of student models trained with different KD criteria along with CTC.*

Model	Criterion	Eval2000			RT03S
		swbd	callhm	full set	
5×800	CTC	12.6	24.5	18.6	22.0
5×400	CTC	13.2	26.2	19.8	22.6
	Output-CE	**12.4**	24.5	18.5	**21.5**
	BestAlign-CE	13.1	24.6	18.8	21.9
	SoftAlign-CE	12.7	24.9	18.8	21.8
	IKD, $\tau = 1$	13.7	25.8	19.5	22.8
	DFD-CE, $\tau = 1$	12.5	24.6	18.6	21.7
	DFD-CE, $\tau = 2$	12.9	**24.4**	18.7	21.8
	Sequence-CE	12.9	24.6	18.8	21.8
	SegNBI-CE	**12.4**	**24.4**	**18.4**	**21.5**
3×800	CTC	13.3	26.7	20.1	23.1
	Output-CE	**12.5**	24.8	18.7	21.6
	BestAlign-CE	13.0	25.2	19.2	22.2
	SoftAlign-CE	12.8	25.5	19.2	22.0
	IKD, $\tau = 1$	13.2	26.5	20.0	22.9
	DFD-CE, $\tau = 1$	**12.5**	24.9	18.7	21.6
	DFD-CE, $\tau = 2$	**12.5**	25.5	19.0	21.7
	Sequence-CE	12.8	25.2	19.1	22.0
	SegNBI-CE	12.6	**24.3**	**18.5**	**21.3**
3×400	CTC	14.6	28.5	21.6	24.3
	Output-CE	14.0	26.8	20.4	23.5
	BestAlign-CE	14.4	27.4	21.0	23.8
	SoftAlign-CE	14.5	27.0	20.8	23.8
	IKD, $\tau = 1$	15.4	29.0	22.2	25.3
	DFD-CE, $\tau = 1$	14.2	26.9	20.6	23.4
	DFD-CE, $\tau = 2$	14.4	27.1	20.8	23.5
	Sequence-CE	14.0	27.1	20.6	23.6
	SegNBI-CE	**13.8**	**26.7**	**20.3**	**23.1**

to Output-CE on this model. When $\tau = 2$, DFD-CE's performance degrades slightly. During training, we find that models trained with $\tau = 2$ achieve lower warping costs than the ones with $\tau = 1$. Since student models are trained from random initialization, and with $\tau = 2$, each warped frame is chosen from 5 frames, student models tend to target on wrong transcription with a higher probability, which hurts performances.

IKD is also proposed to handle inconsistent-alignment issue. When $\tau = 1$, IKD chooses warped frame from the same candidates as DFD-CE ($\tau = 1$) does, but fails to converge when no CTC is used. When combined with CTC, it still performs worse than DFD-CE. These results verify the effectiveness of DTW restrictions in our DFD-CE method.

3.2.3. Effects of segment-wise N-best hypotheses imitation

In our SegNBI experiments, we set $N = 10$ and the segment-wise 10-best hypotheses are computed by prefix beam search algorithm [39] without using lexicon and language model. On training set, the average segment length for symbols in transcription is 3.3 frames. As a comparison, we also tried Sequence-CE [28] with 10-best hypotheses. Table 2 and Table 3 list the comparison results without and with CTC, respectively.

Experimental results show that SegNBI-CE performs best over all KD methods we tried, and students trained with this method even outperform teacher model on 5×400 and 3×800. The performance gaps between 3×400 and 5×800 on Eval2000 and RT03S are reduced by 40% and 39% respectively when no CTC is used, and when combined with CTC, the numbers become 43% and 52% respectively. Compared with models trained from scratch, SegNBI achieves 6%, 7% and 5.6% WER reduction (WERR) on Eval2000 for 5×400, 3×800 and 3×400 respectively without using CTC. If CTC is used, the WERRs become 7.1%, 8.0% and 6% on Eval2000, respectively.

Since SegNBI leverages the alignment information extracted by teacher model and generates N-best hypotheses segment-wisely, it takes into account much more sequence-level hypotheses than Sequence-CE does. Compared with the frame-wise methods, SegNBI does not suffer from inconsistent-alignment issue because it only relies on a rough segmentation of modelling units. Overall, SegNBI leverages teacher's alignment information in a novel way and makes distillation more effective.

4. Conclusion

From the above results, we conclude that our proposed SegNBI method can make use of the rough segmentation capacity of teacher's alignment and bypass its inconsistency with student's alignment to transfer knowledge from a large DBLSTM model to small one. Experimental results on thin, shallow and shallow thin students all verify the effectiveness of SegNBI, which outperforms other KD methods. As future works, we will investigate better alignment strategy based on DFD, explore other segment generation rules in SegNBI, and apply our methods to learn unidirectional LSTMs from BLSTMs.

5. References

[1] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp.1735-1780, 1997.

[2] A. Graves, S. Fernndez, F. J. Gomez, J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," *Proc. ICML-2006*, pp.369-376.

[3] A. Graves, A. Mohamed, G. E. Hinton, "Speech recognition with deep recurrent neural networks," *Proc. ICASSP-2013*, pp.6645-6649.

[4] A. Graves, N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *Proc. ICML-2014*, pp.1764-1772.

[5] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *CoRR*, vol.abs/1412.5567, 2014.

[6] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," *Proc. ICASSP-2015*, pp.4280-4284.

[7] H. Sak, A. Senior, K. Rao, F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *Proc. Interspeech-2015*, pp.1468-1472.

[8] A. Senior, H. Sak, F. de Chaumont Quitry, T. N. Sainath, K. Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNs," *Proc. ASRU-2015*, pp.604-609.

[9] Y. Miao, M. Gowayyed, F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," *Proc. ASRU-2015*, pp.167-174.

[10] G. Pundak, T. N. Sainath, "Lower frame rate neural network acoustic models," *Proc. Interspeech-2016*, pp.22-26.

[11] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., "Deep speech 2: end-to-end speech recognition in English and Mandarin," *Proc. ICML-2016*, pp.173-182.

[12] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," *Proc. Interspeech-2017*, pp.959-963.

[13] H. Soltau, H. Liao, H. Sak, "Neural speech recognizer: acoustic-to-word LSTM model for large vocabulary speech recognition," *Proc. Interspeech-2017*, pp.3707-3711.

[14] G. Zweig, C. Yu, J. Droppo, A. Stolcke, "Advances in all-neural speech recognition," *Proc. ICASSP-2017*, pp.4805-4809.

[15] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, M. Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," *Proc. ICASSP-2018*, pp.4759-4763.

[16] S. Kim, M. Seltzer, J. Li, R. Zhao, "Improved training for online end-to-end speech recognition systems," *Proc. Interspeech-2018*, pp.2913-2917.

[17] C. Zhang, C. Yu, C. Weng, J. Cui, D. Yu, "An exploration of directly using word as acoustic modeling unit for speech recognition," *Proc. SLT-2018*, pp.64-69.

[18] K. Li, J. Li, Y. Zhao, K. Kumar, Y. Gong, "Speaker adaptation for end-to-end CTC models," *Proc. SLT-2018*, pp.542-549.

[19] J. Li, R. Zhao, J.-T. Huang, Y. Gong, "Learning small-size DNN with output-distribution-based criteria," *Proc. Interspeech-2014*, pp.1910-1914.

[20] G. Hinton, O. Vinyals, J. Dean, "Distilling knowledge in a neural network," *Proc. Deep Learning and Representation Learning Workshop, NIPS-2015*.

[21] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio, "Fitnets: Hints for thin deep nets," *Proc. ICLR-2015*.

[22] Y. Chebotar, A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," *Proc. Interspeech-2016*, pp.3439-3443.

[23] J. H. M. Wong, M. J. F. Gales, "Sequence student-teacher training of deep neural networks," *Proc. Interspeech-2016*, pp.2761-2765.

[24] L. Lu, M. Guo, S. Renals, "Knowledge distillation for small-footprint highway networks," *Proc. ICASSP-2017*, pp.4820-4824.

[25] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," *Proc. Interspeech-2017*, pp.3697-3701.

[26] M. Huang, Y. You, Z. Chen, Y. Qian, K. Yu, "Knowledge distillation for sequence model," *Proc. Interspeech-2018*, pp.3703-3707.

[27] G. Kurata, K. Audhkhasi, "Improved knowledge distillation from bi-directional to uni-directional LSTM CTC for end-to-end speech recognition," *Proc. SLT-2018*, pp.411-417.

[28] R. Takashima, S. Li, H. Kawai, "An investigation of a knowledge distillation method for CTC acoustic models," *Proc. ICASSP-2018*, pp.5809-5813.

[29] Y. Kim, A. M. Rush, "Sequence-level knowledge distillation," *Proc. EMNLP-2016*, pp.1317-1327.

[30] H. Sakoe, S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp.43-49, 1978.

[31] E. J. Keogh, M. J. Pazzani, "Derivative dynamic time warping," *Proc. SDM-2001*, pp.1-11.

[32] J. J. Godfrey, Edward. C. Holliman, J. McDaniel, "Switchboard: telephone speech corpus for research and development," *Proc. ICASSP-1992*, vol. 1, pp.517-520.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, "The Kaldi Speech Recognition Toolkit," *Proc. ASRU-2011*.

[34] F. A. Gers, J. Schmidhuber, "Recurrent nets that time and count," *Proc. IJCNN-2000*, vol. 3, pp.189-194.

[35] S. Ioffe, C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal convariate shift," *Proc. ICML-2015*, pp.448-456.

[36] R. Pascanu, T. Mikolov, Y. Bengio, "On the difficulty of training recurrent neural networks," *Proc. ICML-2013*, pp.1310-1318.

[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, T. Darrell, "Caffe: convolutional architecture for fast feature embedding," *CoRR*, vol.abs/1408.5093, 2014.

[38] C. Cieri, D. Miller, K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text," *Proc. LREC-2004*.

[39] A. L. Maas, A. Y. Hannun, D. Jurafsky, A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs," *CoRR*, vol.abs/1408.2873, 2014.