Knowledge Distillation For CTC-based Speech Recognition Via Consistent
Acoustic Representation Learning

Sanli Tian\textsuperscript{1,2}, Keqi Deng\textsuperscript{1,2}, Zehan Li\textsuperscript{1,2}, Lingxuan Ye\textsuperscript{1,2}, Gaofeng Cheng\textsuperscript{1}, Ta Li\textsuperscript{1,2}, Yonghong Yan\textsuperscript{1,2}

\textsuperscript{1}Key Lab of Speech Acoustics and Content Understanding, Institute of Acoustics, CAS, China
\textsuperscript{2}University of Chinese Academy of Sciences, China
\{tiansanli,dengkeqi,lizehan,yelingxuan,chenggaofeng,lita\}@hccl.ioa.ac.cn

Abstract

Recently, end-to-end ASR models based on connectionist temporal classification (CTC) have achieved impressive results, but their performance is limited in lightweight models. Knowledge distillation (KD) is a popular model compression method for improving the performance of lightweight models. However, CTC-models emit spiky posterior distribution making KL-divergence hard to converge, thus hindering the application of KD. To address this issue, we propose a new frame-level KD method that significantly improves the performance of lightweight CTC-based ASR models. First, we design a blank-frame-elimination mechanism that addresses the difficulty of applying KL-divergence on CTC posterior distribution. Second, we propose a consistent-acoustic-representation-learning (CARL) method to improve the representation ability of student model. Unlike matching the student model’s feature to the teacher model’s feature directly, CARL passes the teacher and student encoder’s output features through the teacher’s pre-trained classifier to produce similar outputs by blank-frame-elimination, making teacher and student represent acoustic features in a consistent way. Third, we introduce a two-stage process to further improve the accuracy of ASR, which performs feature-level KD via cosine-similarity in stage1 and softmax-level KD by CARL in stage2. Compared to the vanilla CTC-baseline model, our method relatively reduces CER by 16.1\% and WER by 26.0\% on Aishell-1 and Ted-lium2.

Index Terms: Speech Recognition, Connectionist Temporal Classification (CTC), Knowledge Distillation

1. Introduction

Recently, end-to-end (E2E) automatic speech recognition (ASR) models transcribe input speech directly into the corresponding transcripts and outperform traditional ASR models such as hidden Markov models (HMM) on most public corpora [1, 2, 3, 4]. Among them, the E2E ASR system based on connectionist temporal classification (CTC) has also achieved impressive results [5, 6, 7] and gained attention due to its fast decoding speed and simple structure. However, the success of end-to-end ASR often comes at the cost of considerable computation and memory consumption because of a huge amount of model parameters, making it hard to deploy those algorithms on devices with limited computing and memory resources [8].

To alleviate this problem, some model compression techniques have been created for training compact neural networks, such as network pruning [9, 10], parameter quantization [11, 12], and knowledge distillation (KD) [13, 14, 15, 16].

This work is partially supported by the Goal-Oriented Project Independently Deployed by Institute of Acoustics, Chinese Academy of Sciences(MBJX202106)

Figure 1: (a) Previous knowledge distillation framework lets the student directly learn the knowledge from the teacher’s output. (b) Our proposed CARL mechanism trains the student encoder’s output feature by the teacher’s pre-trained classifier to make the student’s and teacher’s features represent knowledge in a consistent way.

KD technique is first proposed by Hinton et al in [13] and is used to improve the performance of a small network (student) via knowledge transferring from a big network (teacher). This process is achieved by training the student network under joint supervision of the teacher’s output and ground truth [13]. Recently, there are several works on KD that have achieved promising results on computer vision [17, 18] and natural language processing tasks [19, 20].

In the ASR field, the spike posterior property of CTC makes it challenging to apply KD to the CTC-based ASR models. The posterior distribution of CTC-based models is spiky, where most frames emit the blank tokens with high probability, and only a few frames emit the non-blank target tokens [21, 22]. In addition, CTC is an alignment-free algorithm and can have different spike timings despite being trained with the same data, which makes the frame-level KD learning process hard to converge when optimized over KL-divergence objective [23]. To address this problem, Takashima et al [24] proposed a KD method that uses sequence-level knowledge constructed by teacher network’s N-best hypotheses. Kurata et al [25] proposed a KD method named Guiding-CTC for CTC-based models, which only keeps the highest posterior per frame using a mask to guide the student training.

Previous works have greatly promoted the application of KD to the CTC-based models, but these methods still have some limitations. Guiding-CTC [25] can align the spike timings between teacher and student, but it uses a hard one-hot target to perform KD thus failing to make student’s posterior distributions close to that of the teacher. Although KL-divergence can help to make two distributions similar, it is not feasible for the CTC posterior distribution during CTC-based KD [23]. Besides, previous KD methods, as shown in Figure 1 (a), directly employ the teacher’s outputs or features as the learning target of the student, while ignoring that the teacher’s classifier is also available and can be used to improve the representation learning of the student model.
In this paper, we propose a new frame-level KD method that improves the performance of the lightweight CTC-based ASR model. Firstly, we design a blank-frame-elimination (BE) mechanism that solves the difficulty of applying KL-divergence on CTC posterior distribution. Secondly, we propose a consistent-acoustic-representation-learning (CARL) method to improve the representation ability of the student model, which passes the teacher and the student encoder’s output features through the teacher’s pre-trained classifier to produce similar outputs by BE, making teacher and student represent acoustic features in a consistent way. Thirdly, we introduce a two-stage KD process to further improve the ASR accuracy, which performs feature-level KD via cosine-similarity in stage1 and softmax-level KD by CARL in stage2. According to our prior work [26], CTC/attention multi-task learning [4] can also be used to improve the performance of CTC-models. Thus, to gain a strong CTC-baseline, all CTC-based models in our experiments are the CTC part of the models trained by the CTC/Attention mechanism [4]. We conducted experiments on Aishell-1 and Ted-lium2 corpus. Compared to the vanilla CTC student baseline model, our KD method reduces CER by 16.1% and WER by 26.0% relatively on Aishell-1 [27] and Ted-lium2 [28], respectively.

2. Proposed Method

First, we introduce a blank-frame-elimination mechanism (BE) that addresses the difficulty of applying KL-divergence on CTC posterior distribution. Then we introduce a two-stage KD process. The stage1 performs KD on feature-level by cosine-similarity. The stage2 conducts KD on the softmax-level, in which the proposed BE and CARL will be used.

2.1. blank-frame-elimination Mechanism

Previous studies [29, 24, 30] have shown that directly applying the conventional frame-level KD to the CTC-based ASR system by KL-divergence will have negative effect. The main reason why the KL-divergence is not feasible for the CTC posterior distribution during KD is that there are spikes in the teacher’s and student’s posterior distribution and the spike timings of teacher and student are different. To solve this problem, we design a blank-frame-elimination (BE) method, which is illustrated in Figure 2. BE aims to eliminate the frames that the blank token has the highest probability from the teacher’s posterior distribution to form a soft-target-matrix. Then the KL-divergence can be calculated between the soft-target-matrix and the corresponding frames in the student’s posterior distribution.

For an input sample X with N time steps, we feed it into the teacher model and can get the teacher’s posterior distribution $P_T = [P_{T1}, P_{T2}, ..., P_{TN}]$ for each time step, where most frames emit the blank token. Then, we extract the frame at the time step $t_T$, from $P_T$ where the non-blank token has the highest probability, we use $t_T = \{t_{T1}, t_{T2}, ..., t_{TN}\}$ to denote the set of time steps of all the extracted frames. Then all the frames corresponding to time steps $t_T (i = 1 \ldots n)$ in $P_T$ are concatenated together to form a soft-target-matrix $Y$. Different from spiky $P_T$, all the frames of $Y$ emit the non-blank tokens with the highest probability, as shown in Figure 2. Next, we feed the same sample $X$ to the student model and can get the student’s posterior probabilities $P_S$, which has the same shape as $P_T$. And then, we extract all the frames corresponding to the same time steps $t_T (i = 1 \ldots n)$ from $P_S$ and concatenate them into a matrix $P_S'$, which has the same shape as $Y$. Finally, KL-divergence can be calculated between $P_S'$ and $Y$.

$$KL[\text{Softmax}(P_S'),\text{Softmax}(Y)] = KL_{BE}(P_S, P_T)$$

Where $T$ refers to the temperature parameter and $KL_{BE}$ is our further definition of the blank-frame-elimination process.

2.2. Two-Stage KD

We employ a two-stage KD process. The first stage performs KD on feature-level. The second stage conducts KD on the softmax-level, in which the proposed BE and CARL mechanism will be used.

2.2.1. stage1: Feature-Level KD

We wish the student to imitate the teacher’s features before learning from the labels. Thus, we conduct feature-level KD in the first stage. For an input speech $X$, we feed it to the teacher and student model and get the teacher and student encoder’s output features $f_T$ and $f_S$, respectively. Then $f_S$ is passed through a 1D convolutional layer to make it have the same dimension as $f_T$. The KD loss $L_{FKD}$ in the stage1 is computed as follows:

$$L_{FKD} = 1 - \text{cosine}[\text{Convl}(f_S), f_T]$$

2.2.2. stage2: Softmax-Level KD based on CARL and BE

The stage2 is illustrated in Figure 3. We think that the teacher’s classifier contains classification-related knowledge, which is about how the teacher model maps extracted features to target tokens. If similar outputs can be obtained after feeding the encoder’s output feature of teacher and student into the pre-trained teacher’s classifier, the student and teacher can be regarded as representing acoustic features in a consistent way. Therefore, in this stage, we aim to let the student output similar posterior distribution as that of the teacher, and use the teacher’s classifier to train the encoder output feature of the student to make the student and the teacher to represent acoustic features in a consistent way, which is both achieved with BE.

For an input speech $X$, we feed it to the teacher model and can get the teacher encoder’s output feature $f_T$. Then $f_T$ is passed through the teacher’s classifier $T_C$ to get the teacher’s posterior distribution $P_T$. Then the same input $X$ is fed to the
student model, and we can get the student encoder’s output feature \( f_S \). Next, there are two different operations on \( f_S \). In the first operation, the \( f_S \) passes through the student’s classifier \( S_C \) and obtain a posterior distribution \( P_S \):

\[
P_S = S_C \cdot f_S
\]

(3)

Then \( P_S \) calculates CTC-loss with ground truth and calculates \( KL_{BE} \) loss with \( P_T \), respectively:

\[
L_{CTC} = CTCLoss(P_S, GroundTruth) \quad (4)
\]

\[
L_{BE} = KL_{BE}(\frac{P_S}{T}, \frac{P_T}{T}) \quad (5)
\]

Where \( KL_{BE} \) is defined in equation (1) and \( T \) refers to the temperature parameter. \( L_{BE} \) is short for using \( KL_{BE} \) to calculate KD loss between the \( P_S \) and \( P_T \).

The second operation is symmetric to the first operation, denoted by the blue lines as shown in Figure 3. In the second operation, \( f_S \) passes through the Conv1d layer that was used in stage1 to get \( f_{ST} \), which has the same shape as \( f_T \):

\[
f_{ST} = Conv1d(f_S)
\]

(6)

Then we pass \( f_{ST} \) through the teacher model’s frozen and pre-trained classifier \( T_C \) to get a posterior distribution \( P_{ST} \):

\[
P_{ST} = T_C \cdot f_{ST}
\]

(7)

Then \( P_{ST} \) calculates CTC-loss with ground truth and calculates \( KL_{BE} \) loss with \( P_T \), respectively:

\[
L_{CARL-CTC} = CTCLoss(P_{ST}, GroundTruth) \quad (8)
\]

\[
L_{CARL-BE} = KL_{BE}(\frac{P_{ST}}{T}, \frac{P_T}{T}) \quad (9)
\]

Where \( T \) refers to the temperature parameter. And \( L_{CARL-BE} \) is an abbreviation for using \( KL_{BE} \) to calculate KD loss between \( P_{ST} \) and \( P_T \). For convenience, we define two loss functions (10, 11) as below:

\[
L_{CE-CTC} = \beta L_{CE} + (1 - \beta) L_{CTC}
\]

(10)

\[
L_{CARL} = \gamma L_{CARL-CTC} + \lambda L_{CARL-BE}
\]

(11)

Where \( L_{CE} \) is the cross-entropy loss calculated between the output by the student’s decoder and the ground truth\(^1\). And the final loss function for stage2 can be formulated as:

\[
L_{total} = L_{CE-CTC} + L_{CARL} + \alpha L_{BE}
\]

(12)

3. Experiments

3.1. Corpus

We evaluated our proposed methods on two Mandarin and English corpora: Aishell-1 [27] and Ted-lium2 [28]. Aishell-1 corpus consists of about 120000 utters for training set, 7000 utterances for testing set and 14000 utterances for development set. Ted-lium2 corpus is an English speech recognition training corpus collected from TED talks.

\(^1\)The decoder of student is trained by the ground truth only. KD is only carried out to the model’s CTC part. And only the CTC part is used when decoding. All the CTC-based models in our experiments are the CTC part of the models that trained by the CTC/Attention mechanism [4], including the teacher and all the student model.

3.2. Model descriptions

All the CTC-models in our experiments are trained under a framework of hybrid CTC/attention [4]. During inference, we only keep the CTC part and the decoder is discarded for decoding. We use ESPnet2 toolkit [31] to build the models and followed the training configuration provided by ESPnet2.

The teacher and student models both contain a 12-block Transformer encoder and a 6-block Transformer decoder. The hyperparameter settings of teacher’s and student’s encoder and decoder are different, which are shown in the Table 1. We only use the CTC part (encoder) of the model for decoding during inference and the parameters number of the teacher and student CTC part model are shown in the Table 1 as well.

We employ the student model trained without KD as the student-baseline. Comparison methods are Guiding-CTC [25] and using KL-divergence directly in KD [13]. The number of training epochs for different models are shown in Table 2. We use character as the token units for Aishell-1 and use BPE as the token units for Ted-lium2, and the number of BPE is set to 2000. We average the parameters of the model for the last 10 epochs. We also trained an RNN-based LM following the ESPnet2 recipe. In addition, we also give the results without LM in the decoding process. During inference, we only use the CTC part of the models for decoding, and the beam search size is 20. When the external LM is used, its weight is set to 0.3.

### Table 1: The hyperparameter settings of teacher and student.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>dim of ATT</th>
<th>units of FFN</th>
<th>Params.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aishell-1</td>
<td>Teacher</td>
<td>256</td>
<td>2048</td>
<td>18.71M</td>
</tr>
<tr>
<td></td>
<td>Student</td>
<td>128</td>
<td>1024</td>
<td>4.97M</td>
</tr>
<tr>
<td>Ted-lium2</td>
<td>Teacher</td>
<td>256</td>
<td>2048</td>
<td>18.13M</td>
</tr>
<tr>
<td></td>
<td>Student</td>
<td>128</td>
<td>1024</td>
<td>4.68M</td>
</tr>
</tbody>
</table>

### Table 2: The training epochs of different models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>stage1</th>
<th>stage2</th>
<th>Total epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aishell-1</td>
<td>Teacher</td>
<td>-</td>
<td>-</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Student</td>
<td>-</td>
<td>-</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Guiding-CTC [25]</td>
<td>-</td>
<td>-</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>5</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>Ted-lium2</td>
<td>Teacher</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Student</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>KL-div [13]</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Guiding-CTC [25]</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>10</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>
In the main experiments, we only show the performance of our proposed KD method, Teacher, Student-baseline, using KL-divergence directly to KD [13], and Guiding-CTC [25] on Aishell-1 and Tedlium2 corpus. As shown in Table 3, we present the results with and without an external language model in the decoding stage.

1) Aishell-1: Table 3 lists experimental results on Aishell-1 corpus. In the case without LM, we can see that the result of using KL-divergence directly in KD is only slightly improved over the student-baseline. Moreover, its results are even worse in the case of using LM, which shows that KL-divergence cannot be directly applied to the CTC posterior distribution. And the results show that our proposed KD method greatly improves the performance of the student model. Our KD method achieves 18.4% and 16.1% relative CER reduction on dev-set and test-set compared to Student-baseline when decoding without LM. And significant performance improvement can also be observed when the LM is employed in the decoding stage, which reduces CER by 13.2% and 11.3% relatively.

2) Tedlium2: To demonstrate the generalization of our KD method, we also conduct experiments on Tedlium2 corpus, and the results are shown in Table 3. The conclusion we get is similar to that of the Aishell-1 task. Whether or not the LM is employed in the decoding process, our KD method can significantly improve performance compared to the student-baseline. The results of our KD method show 15.8% and 14.4% relative WER reduction on dev-set and test-set with LM, and achieve 19.7% and 26.0% relative WER reduction on dev-set and test-set without LM compared to student-baseline.

### 3.4. Ablation studies on feature-level KD (stage1)

In the main experiments, we only show the performance of our proposed KD method. In this part, we conduct ablation studies to show the effect of the first stage: feature-level KD. The results on Aishell-1 and Tedlium2 are shown in Table 4.

For Aishell-1, the w/o FKD in the table refers to skipping the feature-level KD and directly training 55 epochs for softmax-level KD and 60 epochs for Tedlium2. The w/ FKD in the table refers to train 5 epochs for feature-level KD and 50 epochs for softmax-level KD for Aishell-1 and for Tedlium2, w/ FKD refers to train 10 epochs for feature-level KD and 50 epochs for softmax-level KD. The rest of them are the same.

The experiments show that training with feature-level KD improves the student performance. We argue that using cosine similarity as a constraint in stage1 can make the orientations of teacher’s and student’s features close in the vector space before student learning with labels [32], which can make it easier for the student to learn from the teacher’s output in stage2.

### 3.5. Ablation studies on CARL loss

In previous experiments, we have demonstrated the effectiveness of the two-stage KD process. In this part, we conduct ablation studies to show the respective effects of the two KD loss functions in stage2, \( L_{BE} \) and \( L_{CARL} \). The results of ablation studies on \( L_{CARL} \) on Aishell-1 and Tedlium2 corpus are shown in Table 5. We remove the \( L_{CARL} \) in stage2, preserving other settings unchanged to verify the role of \( L_{CARL} \). We argue that \( L_{CARL} \) can enable the student and the teacher to represent acoustic features in a consistent way, which can improve the student’s representation ability and benefit the recognition performance of student models. Experimental results show that the addition of \( L_{CARL} \) leads to better results.

### 4. Conclusions

In this paper, we propose a new frame-level KD method that significantly improves the student CTC-model’s performance. First, we design a blank-frame-elimination (BE) mechanism to address the difficulty of applying KL-divergence on CTC posterior distribution. Second, we propose a consistent-acoustic-representation-learning (CARL) method which uses the teacher’s classifier to make teacher and student represent acoustic features in a consistent way, thus improving the representation ability of the student model. Third, we introduce a two-stage process to further improve the accuracy, which performs feature-level KD via cosine-similarity in stage1 and softmax-level KD via CARL in stage2. Experimental results show that our proposed method reduces CER by 16.1% and WER by 26.0% relatively on Aishell-1 and Tedlium2, compared to the vanilla CTC-baseline model.
5. References


