Attention Weight Smoothing Using Prior Distributions for Transformer-Based End-to-End ASR

Takashi Maekaku¹, Yuya Fujita², Yifan Peng², Shinji Watanabe²

¹Yahoo Japan Corporation, Tokyo, JAPAN
²Carnegie Mellon University, PA, USA

{tmaekaku,yuyfujit}@yahoo-corp.jp, {yifanpen,swatanab}@andrew.cmu.edu

Abstract

Transformer-based encoder-decoder models have so far been widely used for end-to-end automatic speech recognition. However, it has been found that the self-attention weight matrix could be too peaky and biased toward the diagonal component. Such attention weight matrix contains little useful context information, which may result in poor speech recognition performance. Therefore, we propose the following two attention weight smoothing methods based on the hypothesis that an attention weight matrix whose diagonal components are not peaky can capture more context information. One is a method to linearly interpolate the attention weight using a learnable truncated prior distribution. The other uses the attention weight from a previous layer as a prior distribution given that lower-layer weight tends to be less peaky and diagonal. Experiments on LibriSpeech and Wall Street Journal show that the proposed approach achieves 2.9% and 7.9% relative improvement, respectively, over a vanilla Transformer model.

Index Terms: end-to-end speech recognition, transformer

1. Introduction

End-to-end (E2E) approaches for automatic speech recognition (ASR) have been gaining increasing attention recently. So many approaches have been proposed to improve the performance, which are mainly based on connectionist temporal classification (CTC) [1, 2, 3, 4], recurrent neural network transducer (RNN-T) [5, 6], and attention-based encoder-decoder (AED) [7, 8, 9, 10]. In recent years, many studies have reported that Transformer architecture [11] significantly improves ASR performance in CTC, RNN-T, and AED models [12, 13, 14, 15, 16, 17]. Unlike recurrent neural network (RNN) models, which convert a variable-length sequence into a single fixed-length vector, the self-attention mechanism used in Transformer can capture more detailed context information by considering global dependencies between elements in an input sequence.

However, the attention weights obtained from the self-attention mechanism tend to have larger values for the diagonal components as one moves to the upper layers [18, 19]. Those diagonally aligned weights indicate that only local context is considered, failing to extract global and rich contextual information from a sequence. Fig. 1 shows actual attention weights calculated in a Transformer model, each of which was obtained from the (a) 1st, (b) 6th, or (c) 12th layer. These indicate that the diagonal component of the attention weight matrices contain little context information, which may result in limiting the ASR performance.

One possible method to mitigate this problem is to assume a certain prior distribution for the attention weights and smooth with it inspired by regular attempts in acoustic and language model smoothing [20, 21, 22]. In this work, we investigate two novel attention smoothing methods. First, we propose smoothing using a learnable prior distribution. However, it is difficult to make the entire region of attention learnable because the size of the attention map depends on each utterance length. Hence, we use a $1 \times k$ learnable tensor to construct a band matrix covering a width of $k$ from diagonal components of the attention weight matrix. The band matrix with the softmax function can be regarded as a truncated prior distribution of the attention weights. For the attention weight corresponding to a query at a time step $t$, linear interpolation is performed between the weights in the range from $(t-k)$ to $(t+k)$ and the truncated prior.

Second, we propose a method of smoothing the attention weights using the previous layer’s attention weights as a prior. Based on our observations and reports [19, 18] that the attention weights of the previous layer are less diagonal and less peaky than those of the current layer, the smoothing method can naturally capture the context information via a prior while maintaining the general shape of the original distribution. In addition to the smoothing method that simply uses the previous layer’s attention weights, the use of the multiple priors obtained by previous layers’ attention weights recursively is also examined. We also investigate the case where the coefficients of linear interpolation are predictable by a neural network for each time step.

This study primarily investigates its application to the self-attention and it is inspired by [18, 19]. In [18], it is shown that diagonality, which is a metric to measure how a matrix component is concentrated on a diagonal component, increases the upper layer’s self-attention weights. In addition, replacing such self-attention layers with high diagonality to feed-forward network introduces little performance degradation or sometimes...
small improvement of performance. In [19], the previous work has extended to stochastically eliminate such a high diagonality head of self-attention and achieved performance improvement. Our methods are different from these works because we aim at avoiding high diagonality by smoothing the attention weight, not to replace/exclude such a high diagonality component.

In principle, our smoothing methods can be applied to the source-target attention. As reported in [7, 23], attention weights of source-target attention tend to focus on a single acoustic frame which may lead to performance degradation. This phenomenon could be mitigated by applying our proposed method to source-target attention so we also investigated the effectiveness. Note that relaxed attention proposed in [23] can be viewed as a similar approach to ours because it applied a smoothing with a uniform prior distribution to source-target attention, and concatenated them. In the decoder, source-target attention is used in addition to self-attention. The source-target attention is calculated in the same way as in Eq. 1. The only difference is that the output of the previous layer of the decoder is used as the query input, and the output of the encoder $X_e$ is used for the key and value inputs.

### 2. Transformer-based End-to-End ASR

In this section, we briefly review the Transformer-based E2E ASR model [14], which is based on the attention-based encoder-decoder (AED) structure. Note that AED is just one instance of our proposed techniques and they can also be applied to RNN-T or CTC-based models using the Transformer encoder or its variants.

#### 2.1. Architecture

The Transformer-based ASR model predicts an $L$-length output sequence of token IDs $Y = (y_l \in \mathcal{Y}| l = 1, \ldots, L)$, where $\mathcal{Y}$ is a set of distinct tokens, from a $T^\prime$-length input sequence of speech features $X = (x_t \in \mathbb{R}^d| t = 1, \ldots, T^\prime)$, where $d$ is the dimension of an acoustic feature. Here we use an 80-dimensional log-Mel filterbank as the feature. The model consists of an encoder and a decoder network. First, the speech feature sequence $X$ is fed into a convolutional neural network (CNN) to obtain a subsampled sequence $X^\prime = (x^\prime_t| t = 1, \ldots, T)$, where $T^\prime < T$ is the length after subsampling. Then, the encoder converts $X^\prime$ into a sequence of latent representations $X_e$. The decoder predicts a token $\hat{y}_{(l+1)}$, given $X_e$ and prefix tokens $(y_1, \ldots, y_l)$.

#### 2.2. Self-attention mechanism

The encoder and the decoder have the self-attention mechanism that can capture long-term contextual information by calculating attention weights according to each time step of input.

The output of self-attention with input $X^\prime \in \mathbb{R}^{T \times d_{\text{att}}}$ is defined as:

$$\text{Att}(X^\prime) = \text{softmax} \left( \frac{(X^\prime W^v) (X^\prime W^k)^T}{\sqrt{d_{\text{att}}}} \right) X^\prime W^v$$

(1)

$$= AX^\prime W^v,$$

(2)

where $W^v, W^k, W^v \in \mathbb{R}^{d_{\text{att}} \times d_{\text{att}}}$ are linear transformations to produce a sequence of query, key, and values, respectively. $d_{\text{att}}$ is the dimension of the attention. $^T$ is a transpose operation. $A \in \mathbb{R}^{T \times T}$ is called the attention weight. We use multi-headed self-attention, which splits the input dimension by $H$, calculates self-attention independently, and concatenates them.

In the decoder, source-target attention is used in addition to self-attention. The source-target attention is calculated in the same way as in Eq. 1. The only difference is that the output of the previous layer of the decoder is used as the query input, and the output of the encoder $X_e$ is used for the key and value inputs.

#### 2.3. Training and decoding

For model training, the objective function is defined as

$$L = -\log p(Y|X_e)$$

(3)

$$= -\log \prod_{l=1}^{L-1} p(y_{(l+1)}|y_{1:l}, X_e),$$

(4)

where $p(Y|X_e)$ is decomposed into the product of the decoder’s emission probabilities at each time step. In the decoding stage, the decoder network outputs each token in an autoregressive manner, in which beam search decoding is often adopted to find the most likely hypothesis $\hat{Y}$ as follows:

$$\hat{Y} = \arg \max_{Y \in \mathcal{Y}^*} \log p(Y|X_e),$$

(5)

where $Y \in \mathcal{Y}^*$ is a set of output hypotheses.

### 3. Uniform Smoothing and Proposed Smoothing Methods

#### 3.1. Attention smoothing using uniform prior

As a simple smoothing method, we consider using a uniform distribution as a prior for source-target attention, which is similar to the additive smoothing technique for language modeling [21]. In E2E ASR, relaxed attention [23] has introduced a uniform distribution as a prior for source-target attention, which can capture long-term contextual information in a self-attention.

$$A^\text{uni}(l) = (1 - \gamma) A(l) + \gamma \frac{1}{T^\prime},$$

(6)

where $A(l)$ is the $l$-th layer’s attention weight in Eq. (2) and $\gamma$ is a tunable interpolation hyperparameter.
Recursive Smoothing: The other method is to apply the linear interpolation recursively to the attention weights as follows:

\[
\begin{align*}
A_{(1)}^{rec} &= (1 - \gamma)A_{(1)} + \gamma \cdot A_{(0)}, \\
A_{(l)}^{rec} &= (1 - \gamma)A_{(l)} + \gamma \cdot A_{(l-1)}^{rec} \quad \text{for} \quad l > 1.
\end{align*}
\]  

Prediction of Interpolation Coefficient: Instead of tuning \( \gamma \) as a hyperparameter, we propose to predict it depending on the input as follows:

\[
\begin{align*}
\mathbf{g}(i) &= \sigma\left(\mathbf{X}_l^{(i)}\mathbf{W}_g^{(i)}\mathbf{c}(i)\right), \\
A_{(l)}^{rec} &= (1 - \mathbf{g}(i)) \odot A_{(l)} + \mathbf{g}(i) \odot A_{(l-1)}^{rec},
\end{align*}
\]  

where \( \sigma(\cdot) \) denotes the sigmoid function, and \( \mathbf{c}(i) \in \mathbb{R}^{d_{att}} \) is a learnable vector. \( \odot \) represents row-wise product between \( \mathbf{g}(i) \) and \( i \)-th row component of the multiplied matrix. Making \( \gamma \) predictable in this way not only eliminates the need for tuning it but also allows the value to be adjusted for each time step. Note that \( \gamma \) is different for each head, though omitted for simplicity in Eq. (10).

4. Experiments

4.1. Experimental setup

We trained E2E ASR models using the 100-hour subset of clean audios from LibriSpeech [24] and the 81-hour training set of Wall Street Journal (WSJ) [25]. We build the models based on the Transformer architecture [11] implemented on the ESPnet toolkit [26]. All the models were trained with the joint CTC and attention objectives [27, 14] with a multi-task loss of weight of 0.3. The hyperparameters for the Transformer-related architecture are shown in Table 1. We followed the same setup as in the libriSpeech 100h or wsj recipes in the ESPnet for regularization hyperparameters (e.g., dropout rate, learning rate, label-smoothing weight, and optimizer). For data augmentation, we used speed perturbation [28] at a ratio of 0.9, 1.0, and 1.1. SpecAugment [29] was applied only when training on LibriSpeech. We trained the models for 70 and 100 epochs for LibriSpeech and WSJ, respectively. During inference, model averaging was performed using the model from the last 10 epochs. CTC weight is set to 0.3, and beam size was set to 10 for both corpora. We did not use any external language model (LM) during decoding to simplify the experimental investigations.

4.2. Results and discussion

4.2.1. Non-recursive vs. recursive smoothing

As a preliminary investigation, Table 2 compares the performance of the non-recursive (Eq. (8)) and the recursive (Eq. (9)) smoothing methods, as introduced in Section 3.3, using LibriSpeech 100h. We can see that the improvement in the “test-other” set was particularly large for the recursive smoothing. This indicates that a more reliable prior distribution is obtained by incorporating information from all layers while emphasizing information from the previous layer. Therefore, it is used in subsequent experiments.

4.2.2. Comparison of baseline and proposed methods

Table 3 shows the overall results on LibriSpeech 100h and WSJ. We compared three smoothing methods based on the uniform prior (uniform smoothing), truncated prior (see Section 3.2), and recursive smoothing (see Section 3.3). We followed the
procedure of relaxed attention [23] for the uniform smoothing and applied it during training only. Note that it is equivalent to relaxed attention when it is applied to the source-target attention. The proposed methods, in contrast, performed smoothing both during training and inference. From the results, it is clear that recursive smoothing outperformed the other systems. Compared with the vanilla Transformer, it achieved a 2.9% relative improvement on the "test-other" set, when applied to the encoder with $\gamma = 0.2$ in the case of LibriSpeech. We can also observe that the recursive smoothing is effective when applied to source-target attention as well as self-attention. In the case of WSJ, making $\gamma$ predictable, as in Eq. (10), was effective for improving performance. Compared with the vanilla Transformer, it achieved a 7.9% relative improvement on the "dev93" set when $\gamma$ was made predictable and applied to all attentions in the encoder and the decoder. The smoothing with the truncated prior also outperformed the two baseline systems in some cases even though the improvements were small. Therefore, it is confirmed that both of the proposed methods contributed to improving ASR performance. It is also suggested that our methods are able to obtain useful contextual information.

4.2.3. Attention Analysis

Fig. 3 shows the attention weights that depict the weight patterns obtained by applying the proposed recursive smoothing. Fig. 3 (a) represents the weights corresponding to a certain head for the vanilla Transformer. Fig. 3 (b) and (c) represent the case where the recursive smoothing is applied to the self-attention in the encoder with $\gamma = 0.2$, and the case where $\gamma$ is made predictable and applied to all attentions in the encoder and the decoder, respectively. It is observed that Fig. 3 (a) shows extremely sharp attention weights that are concentrated on the diagonal component. This indicates that this weight matrix does not have useful context information. On the other hand, no such problems have occurred with respect to the other two proposed methods. The attention weight in Fig. 3 (b) and (c) have diagonal components that are not as sharp as the attention weight in (a). This result suggests that the recursive smoothing works to suppress diagonal components of the attention weight matrix from becoming peaky.

5. Conclusions

This paper proposed two novel smoothing methods of attention weight using its prior distribution for the Transformer-based end-to-end automatic speech recognition. The truncated prior distribution using the band matrix or attention weight from the previous layer’s attention weight could be used to mitigate diagonal components of the current layer’s attention from becoming peaky. Experimental results showed that relative improvements of up to 2.9% and 7.9% in LibriSpeech and Wall Street Journal, respectively, are achieved compared to a vanilla Transformer model. The evaluation of the performance with an external LM is left to a future work since the relaxed attention has been reported to improve the performance with LM.

---

Table 3: WER results on LibriSpeech 100h and WSJ. "Enc.", "Dec.", "SA", and "STA" denote encoder, decoder, self-attention, and source-target attention, respectively. $k$ is the size of the tensor that constructs the band matrix $B$, and $\gamma$ is the coefficient of linear interpolation. "pred." means "predictable".

<table>
<thead>
<tr>
<th>System</th>
<th>$k$</th>
<th>$\gamma$</th>
<th>WER (LibriSpeech)</th>
<th>WER (WSJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enc. SA</td>
<td>Dec. SA</td>
<td>Dec. STA</td>
<td>dev-clean</td>
</tr>
<tr>
<td>Baseline: Vanilla Transformer</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.9</td>
</tr>
<tr>
<td>Baseline: Uniform Smoothing</td>
<td>- 0.2</td>
<td>-</td>
<td>-</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>- 0.3</td>
<td>-</td>
<td>-</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td>- 0.2</td>
<td>0.2</td>
<td>-</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>- 0.3</td>
<td>0.3</td>
<td>-</td>
<td>8.4</td>
</tr>
<tr>
<td>Proposed: Truncated Prior</td>
<td>11 0.2</td>
<td>-</td>
<td>-</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>11 0.3</td>
<td>-</td>
<td>-</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>31 0.3</td>
<td>-</td>
<td>-</td>
<td>8.2</td>
</tr>
<tr>
<td>Proposed: Recursive Smoothing</td>
<td>- 0.2</td>
<td>-</td>
<td>-</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>- 0.3</td>
<td>-</td>
<td>-</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>- 0.1</td>
<td>0.1</td>
<td>-</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>- 0.2</td>
<td>0.1</td>
<td>-</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>- pred.</td>
<td>-</td>
<td>-</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>- pred.</td>
<td>-</td>
<td>-</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>- pred.</td>
<td>pred.</td>
<td>-</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>- pred.</td>
<td>pred.</td>
<td>pred.</td>
<td>8.4</td>
</tr>
</tbody>
</table>

---

Note that we did not observe the improvement with relaxed attention when it is applied to the source-target attention. The proposed methods, in contrast, performed smoothing both during training and inference. From the results, it is clear that recursive smoothing outperformed the other systems. Compared with the vanilla Transformer, it achieved a 2.9% relative improvement on the "test-other" set, when applied to the encoder with $\gamma = 0.2$ in the case of LibriSpeech. We can also observe that the recursive smoothing is effective when applied to source-target attention as well as self-attention. In the case of WSJ, making $\gamma$ predictable, as in Eq. (10), was effective for improving performance. Compared with the vanilla Transformer, it achieved a 7.9% relative improvement on the "dev93" set when $\gamma$ was made predictable and applied to all attentions in the encoder and the decoder. The smoothing with the truncated prior also outperformed the two baseline systems in some cases even though the improvements were small. Therefore, it is confirmed that both of the proposed methods contributed to improving ASR performance. It is also suggested that our methods are able to obtain useful contextual information.

---

As a preliminary experiment, we also evaluated the proposed methods when applied only during training, but found no improvement.
6. References


