Self-attention Networks for CTC in Speech Recognition

Julian Salazar[1] **Katrin Kirchhoff**[1,2] **Zhiheng Huang**[1]

[1]Amazon AI [2]University of Washington

International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2019

Motivation

CTC [Gra06] is a sequence-level objective function for end-to-end monotonic sequence transduction. Frame labels are viewed as conditionally independent; paths are collapsed to give outputs:

$$P(\boldsymbol{\pi} \mid \mathbf{X}) = \prod_{t=1}^{T} P(\pi_t, t \mid \mathbf{X})$$

$$\mathcal{B} : \mathbb{L}'^{T} \to \mathbb{L}^{\leq T}, \text{ e.g., } (a, b, -, -, b, b, -, a) \mapsto (a, b, b, a)$$

$$P(\mathbf{y} \mid \mathbf{X}) = \textstyle\sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} P(\boldsymbol{\pi} \mid \mathbf{X})$$

- **BLSTMs:** good at temporal modeling, bad at local feature extraction; recurrent.
- **CNNs:** good at time/frequency translation-invariant features at multiple levels; parallel, but bad at temporal modeling (requires depth).

Self-attention [Vas17] has benefits of both; it is a non-autoregressive layer that sees the whole sequence and attentively pools new representations. We stack self-att. layers (Transformer encoder) and train **the first non-autoregressive, fully self-attentive, ASR model** at scale (**end-to-end character SOTA on LibriSpeech**).

Table 1. Computational complexity of various layer types

Model	Operations per layer	Sequential operations	Maximum path length
Recurrent	$O(Td^2)$	$O(T)$	$O(T)$
Convolutional	$O(kTd^2)$	$O(1)$	$O(T/k)$
Convolutional (strided/dilated/pooled)	$O(kTd^2)$	$O(1)$	$O(\log_k(T))$
Self-attention	$O(T^2d)$	$O(1)$	$O(T)$
Self-attention (restricted)	$O(kTd)$	$O(1)$	$O(T/k)$

Self-attention

This generalizes content-based attention by **having each input concurrently query all inputs**:

$$\mathbf{HdAtt}^{(i)} = \sigma\left(\boldsymbol{Q}^{(i)}\boldsymbol{K}^{(i)\top}/\sqrt{d_{\text{h}}}\right)\boldsymbol{V}^{(i)}$$

Here Q, K, V are learned position-wise transforms of the input matrix. Multiple sets ("heads") of transforms are learned then concatenated:

$$\mathbf{MltHdAtt} = [\mathbf{HdAtt}^{(1)}, \ldots, \mathbf{HdAtt}^{(n_{\text{hds}})}]$$

Features are extracted by position-wise dense layers after every multi-head attention:

$$\mathbf{MidLyr}(\boldsymbol{H}) = \text{LN}(\mathbf{MltHdAtt}(\boldsymbol{H}) + \boldsymbol{H}),$$

$$\mathbf{SelfAttLyr}(\boldsymbol{H}) = \text{LN}(\mathbf{FFN}(\mathbf{MidLyr}(\boldsymbol{H})) + \mathbf{MidLyr}(\boldsymbol{H}))$$

We stack a la deep CNNs (self-attention ≅ parameterized convs.), w/ full temporality per layer.

Embeddings

Table 1: Self-attention creates an $O(T^2d)$ attention matrix. **Audio has large T** (e.g., 1500-frame utterances), **so we downsample/reshape** by $k = 3$ so that $T/k = d$ (embedding dim.).

Table 2: Location is lost when all frames are viewed at once. [Vas17] introduced position encodings added to each input vector:

$$\text{PE}(t, 2i) = \sin \frac{t}{10000^{2i/d_{\text{emb}}}}, \; \text{PE}(t, 2i+1) = \cos \frac{t}{10000^{2i/d_{\text{emb}}}}$$

but [Spe18] converged for audio only with concatenation. CTC's inductive bias is strong (encodings optional). **Adding pos. encs. did better for WSJ, but concat. scaled better (LibriSpeech)**

Figure 1. Combining self-attention with CTC

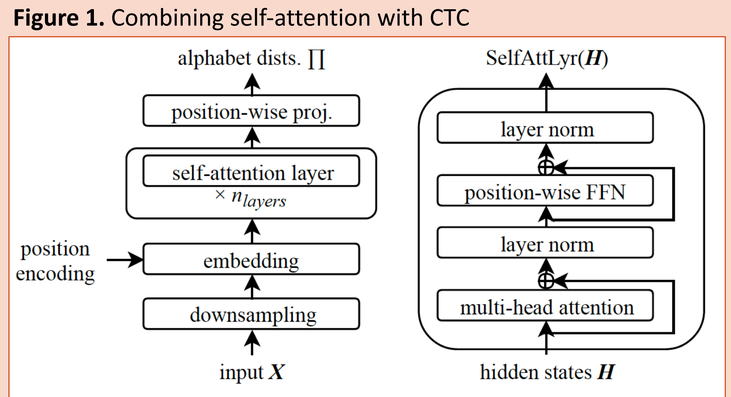


Table 2. WSJ performance wrt. downsampling/embedding choice

Downsampling	Position embedding	dev93 CER	dev93 WER
reshape	content-only	7.62	9.57
reshape	additive	7.10	**9.27**
reshape	concatenative	7.10	9.97
pooling (maximum)	additive	7.15	10.72
pooling (average)	additive	**6.82**	9.41
subsample	additive	none	none

Training

We take 10 layers, 512 hidden dimensions, and 8 heads per layer. Our features are CMVN MFCCs + Δ + ΔΔ. We use the **same 30-35M parameter model for WSJ and LibriSpeech** (compare with 100M+ param. CNN models for the latter [Col16]). We use [Vas17]'s LR schedule (**linear warmup, inverse square root decay**), with two further fixed decays:

$$\text{LR}(n) = \frac{\lambda}{\sqrt{d_{\text{h}}}} \min\left(\frac{n}{n_{\text{warmup}}^{1.5}}, \frac{1}{\sqrt{n}}\right)$$

Experiments

We use character labels and compare with past end-to-end works. **Character error rate (CER)** is from greedy decoding without an LM. **Word error rate (WER)** is after their default n-gram LMs, which we apply via WFST. We get **SOTA CERs for CTC-based models**, and outperform past end-to-end MLE-trained models on LibriSpeech test-clean.

Table 3. Results on the Wall Street Journal (WSJ) corpus

Model	dev93 CER	dev93 WER	eval92 CER	eval92 WER
CTC (BRDNN) [5]	—	—	10.0	—
CTC (BLSTM) [4]	—	—	9.2	—
CTC (BLSTM) [17]	11.5	—	9.0	—
Enc-Dec (4-1) [17]	12.0	—	8.2	—
Enc-Dec+CTC (4-1) [17]	11.3	—	7.4	—
Enc-Dec (4-1) [39]	—	—	6.4	9.3
CTC/ASG (Gated CNN) [40]	6.9	9.5	4.9	6.6
Enc-Dec (2,1,3-1) [41]	—	—	**3.6**	—
CTC (SAN), reshape, additive	7.1	9.3	5.1	6.1
+ label smoothing, $\lambda = 0.1$	6.4	8.9	4.7	5.9

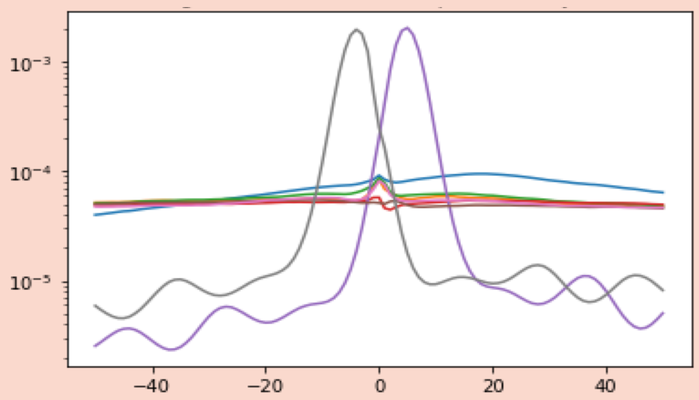
Table 4. Results on the LibriSpeech corpus

Model	Tok.	test-clean CER	test-clean WER	test-other CER	test-other WER
CTC/ASG (Wav2Letter) [9]	chr.	6.9	7.2	—	—
CTC (DS1-like) [33,43]	chr.	—	6.5	—	—
Enc-Dec (4-4) [44]	chr.	6.5	—	18.1	—
Enc-Dec (6-1) [45]	chr.	4.5	—	11.6	—
CTC (DS2-like) [8,32]	chr.	—	5.7	—	15.2
Enc-Dec+CTC (6-1, pretr.) [20]	10k	—	4.8	—	15.3
CTC/ASG (Gated CNN) [23]	chr.	—	4.8	—	14.5
Enc-Dec (2,6-1) [41]	10k	2.9	—	**8.4**	—
CTC (SAN), reshape, additive	chr.	3.2	5.2	9.9	13.9
+ label smoothing, $\lambda = 0.05$	chr.	3.5	5.4	11.3	14.5
CTC (SAN), reshape, concat.	chr.	**2.8**	4.8	9.2	13.1

Alphabets and attention

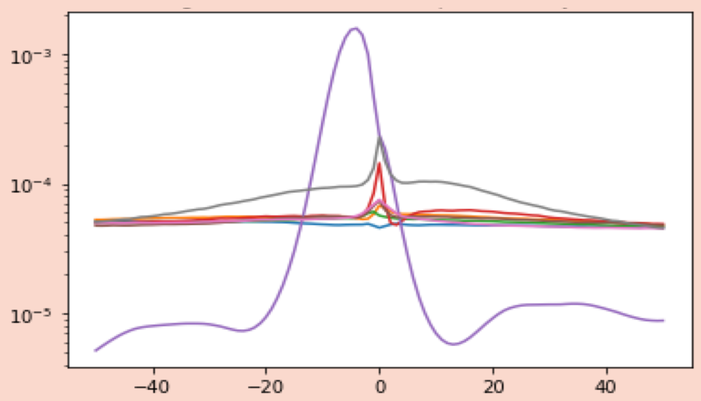
[Pov18, Spe18] consider the context attended by single self-attention layers. We do so **per head** and **across layers**, as a **function of label alphabet**. Character (and subword, not pictured) models learn directional heads. Phonemes & lexicon improve WSJ from 5.9 WER to 4.8, learns sharp backward head (more cond. independent). See Fig. 2, 3.

Figure 2. Differentiated attention heads of our WSJ character model



Character: g o o d [sp.] m o r n i n g

Figure 3. Differentiated attention heads of our WSJ phoneme model



CI-phonemes: g ʊ d [sil.] ˈm ɔː n ɪ ŋ

Table 5. Phoneme model on WSJ (via the CMU lexicon)

Model	dev93 PER	dev93 WER	eval92 PER	eval92 WER
CTC (BRDNN) [7]	—	—	—	7.87
CTC (BLSTM) [11]	—	9.12	—	5.48
CTC (ResCNN) [11]	—	9.99	—	5.35
Ensemble of 3 (voting) [11]	—	**7.65**	—	**4.29**
CTC (SAN), reshape, additive	7.12	8.09	5.07	4.84
+ label smoothing, $\lambda = 0.1$	6.86	8.16	4.73	5.23

Future work

- The success of BERT (which uses the same encoder) suggests opportunities for pre-training. We attain **SOTA on NIST LRE07 and a Fisher speaker rec. task by finetuning the encoder** of a Fisher-trained SAN-CTC model in [Lin19].
- Figure 2 validates [Pov18] by showing most attention heads use local/directed context. One could **use restricted attention heads** to speed up inference and/or **enable online decoding**.

Contact: Julian Salazar (julsal@amazon.com)

References:

[Gra06] A. Graves et al, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." ICML 2006

[Col16] R. Collobert et al, "Wav2Letter: an end-to-end ConvNet based speech recognition system." CoRR 2016

[Vas17] A. Vaswani et al., "Attention is all you need." NeurIPS 2017

[Pov18] D. Povey et al., "A time-restricted self-attention layer for ASR." ICASSP 2018

[Spe18] M. Sperber et al., "Self-attentional acoustic models." INTERSPEECH 2018

[Lin19] S. Ling et al., "Contextual phonetic pre-training for end-to-end utterance-level language and speaker recognition." Submitted to INTERSPEECH 2019