ADVANCING CONNECTIONIST TEMPORAL CLASSIFICATION

WITH ATTENTION MODELING

Amit Das, Jinyu Li, Rui Zhao, Yifan Gong

Microsoft AI and Research
Outline

• Overview: Connectionist Temporal Classification (CTC)

• Issues with CTC

• Proposed Solution: Blend Attention directly into CTC

• Experiments and Results

• Conclusions
Connectionist Temporal Classification (CTC)

• CTC is a sequence-to-sequence learning method used to map speech waveforms directly to characters, phonemes, or even words

\[ L_{\text{CTC}} = -\ln p(z | x) \]

\( z \) -- labels sequence  \( x \) -- observation frames

• CTC paths \( \pi \) differ from labels sequences in that:
  
  — Add the blank as an additional label, meaning no (actual) labels are emitted
  
  — Allow repetitions of non-blank/blank labels

\[
p(z | x) = \sum_{\pi \in \text{Desired}} p(\pi | x)
\]
End-to-End Modeling with CTC

• Greedy decoding: concatenate the non-blank tokens corresponding to the posterior spikes.

• Neither LM nor complex decoding is involved.
CTC Issues

**Issues:**

- Assumes **conditional independence (CI)** between outputs given input. Not true, in general, for sequential tasks like ASR, machine translation, language modeling.

\[
p(\pi|x) \overset{\text{CI}}{=} \prod_{t=1}^{T} p(\pi_t|x) \overset{\Delta}{=} \prod_{t=1}^{T} y_t(\pi_t) \quad \text{CI: } (\pi_t \perp \pi_{\neq t})|x
\]

- Assumes **hard alignment**. Output \( y_t \) dependent on input \( x_t \). Not true, in general, since neighboring inputs \( x_{<t}, x_{>t} \) also have an influence.

**Solution:** **Attention mechanism** relaxes hard alignment.
RNN-Encoder Decoder (No Attention)

Note: Fixed context vector ‘c’ at all times.
RNN-Encoder Decoder (Attention)

Note: Time-varying context vector
CTC Attention
Baseline CTC Network = RNN + CTC Loss

\[ h_t = \mathcal{H}(x_t, h_{t-1}) \]

\[ u = t \text{ for CTC modeling} \]
CTC Annotate

- Key: Compute the context vector $c_\text{u}$ as \textit{time convolved feature}.
Context Vector As Time Convolved Feature

- Time convolved feature is a special case of context vector with uniform attention.

**RNN-ED Annotate**

\[ c_u = \text{Annotate}(\alpha_u, h) \]
\[ = \sum_{t=1}^{T} \alpha_{u,t} h_t \]

Non-Uniform attention

**CTC Annotate**

\[ c_u = W' * h \]
\[ = \sum_{t=u-\tau}^{u+\tau} W'_{u-t} h_t \]
\[ \Delta = \sum_{t=u-\tau}^{u+\tau} g_t \]
\[ = \gamma \sum_{t=u-\tau}^{u+\tau} \alpha_{u,t} g_t. \]
\[ \alpha_{u,t} = \frac{1}{C} \]
\[ \gamma = C' \]

Uniform attention
CTC Attend

• Why need this? Ans: To move from uniform to non-uniform attention.
• In non-uniform attention, we weight the input features distinctively.
• How? Introduce an Attend block. No explicit decoder in CTC network. Replace the decoder state $s_{u-1}$ in RNN-ED Attend with the logits $z_{u-1}$ in CTC Attend.
CTC Attend

- Attend block is simply a single layer neural network.
- Scores $e_{u,t}$ are computed using $z_{u-1}, g_t$.
- Softmax over scores computed over a small context window $[u - \tau, u + \tau]$.

RNN-ED Attend
\[
e_{u,t} = \begin{cases} 
    v^T \tanh(Us_{u-1} + Wh_t + b), & \text{(content)} \\
    v^T \tanh(Us_{u-1} + Wh_t + Vf_{u,t} + b), & \text{(hybrid)}
\end{cases}
\]

CTC Attend
\[
e_{u,t} = \begin{cases} 
    v^T \tanh(Uz_{u-1} + Wg_t + b), & \text{(content)} \\
    v^T \tanh(Uz_{u-1} + Wg_t + Vf_{u,t} + b), & \text{(hybrid)}
\end{cases}
\]

\[
\alpha_{u,t} = \frac{\exp(e_{u,t})}{\sum_{t'=1}^{T} \exp(e_{u,t'})}, \quad t = [1, T]
\]

\[
\alpha_{u,t} = \frac{\exp(e_{u,t})}{\sum_{t'=u-\tau}^{u+\tau} \exp(e_{u,t'})}, \quad t = [u - \tau, u + \tau]
\]
CTC Attention Network

- **Softmax**
- **LM**
- **Attention**
- **LSTM**

CTC Attention Network Diagram
Integration with Language Model (LM)

- Instead of using $\mathbf{z}_{u-1}$ in $\text{Attend}(.)$, use $\mathbf{z}^{\text{LM}}_{u-1}$ obtained from another RNN/LSTM $\mathcal{H}(.)$ modeling a pseudo-LM.

- The term $\mathbf{z}^{\text{LM}}_{u-1}$ captures long-term language information (n-gram like).

- However, because of blanks in CTC, it is only a pseudo-LM.

$$
\mathbf{z}^{\text{LM}}_{u-1} = \mathcal{H}(\mathbf{x}^{\text{LM}}_{u-1}, \mathbf{z}^{\text{LM}}_{u-2}), \quad \mathbf{x}^{\text{LM}}_{u-1} = \begin{bmatrix} \mathbf{z}_{u-1} \\ \mathbf{c}_{u-1} \end{bmatrix}
$$
Component-wise Attention (COMA)

- Instead of a single score per vector $g_t$, we obtain a score for every component of $g_t$.

**CTC Attend (w/o COMA)**

$$e_{u,t} = \begin{cases} 
    v^T \tanh(Uz_{u-1} + Wg_t + b), & \text{(content)} \\
    v^T \tanh(Uz_{u-1} + Wg_t + Vf_{u,t} + b), & \text{(hybrid)} 
\end{cases}$$

$$e_{u,t} \in \mathbb{R}$$

$$t = [u - \tau, u + \tau]$$

**CTC Attend (w/ COMA)**

$$e_{u,t} = \begin{cases} 
    \tanh(Uz_{u-1} + Wg_t + b), & \text{(content)} \\
    \tanh(Uz_{u-1} + Wg_t + Vf_{u,t} + b), & \text{(hybrid)} 
\end{cases}$$

$$e_{u,t} \in \mathbb{R}^n$$

$$t = [u - \tau, u + \tau]$$
Component-wise Attention (COMA)

- Keeping component fixed, take softmax across all time steps to get the COMA weights.

CTC Attend (w/o COMA)

\[ e_{u,t} \in \mathbb{R} \]

\[
\begin{bmatrix}
  e_{u,t-\tau} & e_{u,t-\tau+1} & \cdots & e_{u,t} & \cdots & e_{u,t+\tau}
\end{bmatrix}
\]

CTC Attend (w/ COMA)

\[ e_{u,t} \in \mathbb{R}^n \]

\[
\begin{pmatrix}
  e_{u,t-\tau}(1) & e_{u,t-\tau+1}(1) & \cdots & e_{u,t}(1) & \cdots & e_{u,t+\tau}(1) \\
  e_{u,t-\tau}(2) & e_{u,t-\tau+1}(2) & \cdots & e_{u,t}(2) & \cdots & e_{u,t+\tau}(2) \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  e_{u,t-\tau}(n) & e_{u,t-\tau+1}(n) & \cdots & e_{u,t}(n) & \cdots & e_{u,t+\tau}(n)
\end{pmatrix}
\]
Component-wise Attention (COMA)

- Keeping component fixed, take softmax across all time steps to get the COMA weights.

CTC Attend (w/o COMA)

$$\alpha_{u,t} = \frac{\exp(e_{u,t})}{\sum_{t'=u-\tau}^{u+\tau} \exp(e_{u,t'})}$$

$$c_u = \gamma \sum_{t=u-\tau}^{u+\tau} \alpha_{u,t} g_t$$

CTC Attend (w/ COMA)

$$\alpha_{u,t,j} = \frac{\exp(e_{u,t,j})}{\sum_{t'=u-\tau}^{u+\tau} \exp(e_{u,t',j})}, \quad j = 1, \cdots, n$$

$$c_u = \gamma \sum_{t=u-\tau}^{u+\tau} \alpha_{u,t} \odot g_t$$
Experimental Set-Up

• Training Data: Cortana (Microsoft Voice Assistant)
  • 3400 hours (3.3 million utterances)

• Test Data: Cortana
  • 6 hours (5600 utterances)

• Model:
  • **Letter CTC (28 or 83 characters)**
  • 5 layers Uni-LSTM with 1024 memory cells or Bi-LSTM with 512 memory cells in each direction. Layer output is linearly projected to 512 dimensions.

• Greedy decoding
  • **No lexicon, No LM.** (Purest E2E)

• Log Mel Filterbank Energy (LMFE) Features:
  • base frame: 10 ms, Dim = 80
  • Input for Uni-LSTM: 8 base frames, shift = 3 base frames, Dim = 640
  • Input for Bi-LSTM: 3 base frames, shift = 3 base frames, Dim = 240
WER of CTC models Using One-Sided Context Window Size ($\tau$) = 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Char</th>
<th>Abs %</th>
<th>Rel %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni-LSTM</td>
<td>28</td>
<td>5.55</td>
<td>18.75</td>
</tr>
</tbody>
</table>

CTC Models:
- Vanilla CTC: 29.6
- Time Convolution: 27.36
- + Content Attention: 25.41
- + Hybrid Attention: 25.62
- + LM: 24.74
- + Component Attention: 24.05
### WER of CTC Models Using One-Sided Context Window Size ($\tau = 4$)

<table>
<thead>
<tr>
<th>Model</th>
<th>Char</th>
<th>Abs %</th>
<th>Rel %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni-LSTM</td>
<td>28</td>
<td>5.55</td>
<td>18.75</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>28</td>
<td>5.55</td>
<td>21.06</td>
</tr>
</tbody>
</table>

**CTC Models**

- Vanilla CTC
- Time Convolution
- + Content Attention
- + Hybrid Attention
- + LM
- + Component Attention
### WER of CTC Models Using One-Sided Context Window Size (τ) = 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Char</th>
<th>Abs %</th>
<th>Rel %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni-LSTM</td>
<td>28</td>
<td>5.55</td>
<td>18.75</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>28</td>
<td>5.55</td>
<td>21.06</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>83</td>
<td>4.80</td>
<td>20.61</td>
</tr>
</tbody>
</table>

**CTC Models**

- Vanilla CTC
- Time Convolution
- + Content Attention
- + Hybrid Attention
- + LM
- + Component Attention
Gain on Larger Units

<table>
<thead>
<tr>
<th>CTC Models</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-phone CTC (with LM)</td>
<td>9.28</td>
</tr>
<tr>
<td>E2E CTC with 3-letter units</td>
<td>13.28</td>
</tr>
<tr>
<td>E2E CTC with 3-letter units + Attention</td>
<td>11.36</td>
</tr>
<tr>
<td>E2E CTC with mixed units (word + 3-letter)</td>
<td>9.32</td>
</tr>
<tr>
<td>E2E CTC with mixed units + Attention</td>
<td>8.65</td>
</tr>
</tbody>
</table>

Conclusions

• Soft-alignment training in CTC using
  • Time Convolution
  • Hybrid Attention
  • Implicit LM
  • Component Attention

• Reduction in WER:
  • 3400 hrs: ~ 20% relative with single letter unit. Significant gain with larger unit.
  • Similar improvement no matter whether we used weaker (Uni-LSTM CTC) or stronger baseline (Bi-LSTM CTC).
Thank You