End-to-end Neural Architectures
For ASR

Lecture 15

Instructor: Preethi Jyothi
Connectionist Temporal Classification (CTC): Recap

- CTC objective function is the probability of an output label sequence $y$ given an utterance $x$ (by summing over all possible alignments for $y$ provided by $B^{-1}(y)$):

$$CTC(x, y) = \Pr(y \mid x) = \sum_{a \in B^{-1}(y)} \Pr(a \mid x)$$

$$= \sum_{a \in B^{-1}(y)} \prod_{t=1}^{T} \Pr(a_t \mid x)$$

- Efficient forward+backward algorithm to compute this loss function and its gradients
Illustration: Forward Algorithm to compute $\alpha_t(j)$

$$\alpha_t(j) = \sum_{i=j-2}^{j} \alpha_{i-1}(i)a_{ij}b_t(y'_j)$$

where

$$b_t(y'_j)$$ is the probability given by NN to the symbol $y'_j$ for $t = 1 \ldots T$, when $|x| = T$

$$y'_j = \begin{cases} y_{j/2} & \text{if } j \text{ is even} \\ \epsilon & \text{otherwise} \end{cases} \quad (j = 1 \ldots 2l + 1 \text{ when } |y| = l)$$

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \text{ or } i = j - 1 \\ 1 & \text{if } i = j - 2 \text{ and } y'_j \neq y'_{j-2} \\ 0 & \text{otherwise} \end{cases}$$

$$CTC(x, y) = \sum_{a \in B^{-1}(y)} Pr(a | x) = \alpha_T(2l) + \alpha_T(2l + 1)$$

Image from: https://distill.pub/2017/ctc/
CTC vs. LAS

- Works well for end-to-end ASR systems
- CTC makes an assumption that the network outputs at different time steps are conditionally independent given the inputs
- The Listen, Attend and Spell [LAS] network makes no independence assumptions about the probability distribution of the output sequences given the input
  \[
P(y|x) = \prod_i P(y_i|x, y_{<i})
\]
- Based on the sequence-to-sequence with attention framework

---

[LAS]: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016
Sequence to sequence models

Encoder-decoder architecture
Sequence to sequence models
Encoder-decoder architecture
Sequence to sequence models
Encoder-decoder with attention
Sequence to sequence models
Encoder-decoder with attention

How do we compute $\alpha_j$?

$c_i = \sum_j \alpha_{ij} h_j$

$s_i$
Sequence to sequence models

Encoder-decoder with attention

\[
e_{i1} = f(s_i, h_1)
\]

\[
e_{ij} = f(s_i, h_j)
\]

\[
e_{iM} = f(s_i, h_M)
\]

\[
a_i = \frac{\sum_j a_{ij} h_j}{\sum_j a_{ij}}
\]

\[
a_{ij} \leftarrow \text{normalized}(e_{ij})
\]

\[
a_{ij} \in [0,1]
\]

\[
f \text{ could be a neural n/w}
\]
The Model

- The Listen, Attend & Spell (LAS) architecture is a sequence-to-sequence model consisting of
- a Listener (Listen): An acoustic model encoder. Deep BLSTMs with a pyramidal structure: reduces the time resolution by a factor of 2 in each layer.
- a Speller (AttendAndSpell): An attention-based decoder. Consumes $h$ and produces a probability distribution over characters.

$$h = \text{Listen}(x)$$

$$P(y_i | x, y_{<i}) = \text{AttendAndSpell}(y_{<i}, h)$$
Attend and spell

- Produces a distribution over characters conditioned on all characters seen previously
  \[
  c_i = \text{AttentionContext}(s_i, h) \\
  s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \\
  P(y_i|x, y_{<i}) = \text{CharacterDistribution}(s_i, c_i)
  \]

- At each decoder time-step i, AttentionContext computes a score for each encoder step u, which is then converted into softmax probabilities that are linearly combined to compute \( c_i \)
  \[
  e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle \\
  \alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_{u'} \exp(e_{i,u'})} \\
  c_i = \sum_u \alpha_{i,u} h_u
  \]
Training and Decoding

- Training
  - Train the parameters of the model to maximize the log probability of the training instances
    \[
    \tilde{\theta} = \max_{\theta} \sum_i \log P(y_i|x, \tilde{y}_{<i}; \theta)
    \]
- Decoding
  - Simple left-to-right beam search
  - Beams can be rescoring with a language model
Experiments

Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring can be beneficial.

<table>
<thead>
<tr>
<th>Model</th>
<th>Clean WER</th>
<th>Noisy WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLDNN-HMM [22]</td>
<td>8.0</td>
<td>8.9</td>
</tr>
<tr>
<td>LAS</td>
<td>14.1</td>
<td>16.5</td>
</tr>
<tr>
<td>LAS + LM Rescoring</td>
<td>10.3</td>
<td>12.0</td>
</tr>
</tbody>
</table>

- Listen function used 3 layers of BLSTM (512 nodes); AttendAndSpell used a 2-layer LSTM (256 nodes)
- Constraining the beam search with a dictionary had no impact on WER
Analysis

Alignment between the Characters and Audio

The model proposed here is based on the sequence-to-sequence framework. This distinguishes it from end-to-end training systems, rather than end-to-end models. While CTC has shown tremendous promise in end-to-end systems, rather than end-to-end models.

We have presented Listen, Attend and Spell (LAS), a neural speech recognizer that can transcribe acoustic signals to characters directly without using any of the traditional components of a speech recognition system - the acoustic, pronunciation and language models are all encoded within its parameters. We argue that this makes it not only an end-to-end trained system, rather than end-to-end models.

We demonstrated that such an end-to-end model can be trained and to accomplish this. We showed how this model learns an implicit language model blending (see section 2.4). We note that

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>how</td>
<td>1</td>
</tr>
<tr>
<td>much</td>
<td>2</td>
</tr>
<tr>
<td>would</td>
<td>3</td>
</tr>
<tr>
<td>wood</td>
<td>4</td>
</tr>
<tr>
<td>chuck</td>
<td>5</td>
</tr>
</tbody>
</table>

The alignment produced is generally monotonic without a need for any location based priors. The model proposed here is based on the sequence-to-sequence framework. This distinguishes it from end-to-end training systems, rather than end-to-end models.
Attention Distributions

Spelling Variants of "aaa" vs. "triple a"

<table>
<thead>
<tr>
<th>Beam</th>
<th>Text</th>
<th>log $P$</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth</td>
<td>call aaa roadside assistance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>call aaa roadside assistance</td>
<td>-0.57</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>call triple a roadside assistance</td>
<td>-1.54</td>
<td>50.00</td>
</tr>
<tr>
<td>3</td>
<td>call trip way roadside assistance</td>
<td>-3.50</td>
<td>50.00</td>
</tr>
<tr>
<td>4</td>
<td>call xxx roadside assistance</td>
<td>-4.44</td>
<td>25.00</td>
</tr>
</tbody>
</table>

Figure 7: The spelling variants of "st" vs. "saint" produces different attention distributions, both spelling variants appear in our top beams. The ground truth is: "st mary's animal clinic".