End-to-End Attention-Based Large Vocabulary Speech Recognition

Dzmitry Bahdanau  Jan Chorowski  Dmitriy Serdyuk  Philemon Brakel  Yoshua Bengio

University of Montreal (MILA)

2016 IEEE International Conference on Acoustics, Speech and Signal Processing
Motivation

Problem Statement

- Current SOTA contain separate modules for acoustic and language modelling, sequence decoding etc.

- Deep networks are popular acoustic models, but are hybrid systems (GMM + HMM) which require a two-stage training process.

- Acoustic model is not directly trained to minimize the final objective of interest.
Motivation

Contributions

- Train end-to-end models by replacing HMMs with Attention based Recurrent Sequence Generators (ARSG)
- Long sequences can be modeled by limiting the area of attention to a range of most promising locations.
- A recurrent architecture that reduces source sequence length by pooling frames neighbouring in time.
- Combine character level ARSG and n-gram model using Weighted Finite State Transducer framework
Attention-Based Recurrent Sequence Generators

Encoder-Decoder Architecture

Architecture of Encoder: Bi-Directional RNN with Pooling over Time
Attention-Based Recurrent Sequence Generators

Encoder-Decoder Architecture

\[ p(y_1, y_2, \ldots, y_T \mid h_1, h_2, \ldots, h_l) \]

Architecture of Decoder

\[
\begin{align*}
F &= Q \ast \alpha_{t-1} \\
\epsilon_{tl} &= w^T \tanh(Ws_{t-1} + Vh_l + Uf_l + b) \\
\alpha_{tl} &= \exp(\epsilon_{tl}) \left/ \sum_{l=1}^{L} \exp(\epsilon_{tl}) \right.
\end{align*}
\]
What is the complexity of the training procedure? (for attention weights)
Attention-Based Recurrent Sequence Generators

Windowing

- What is the complexity of the training procedure? (for attention weights)
- Complexity: $O(LT)$
- Is it possible to achieve a lower time complexity?
What is the complexity of the training procedure? (for attention weights)

- Complexity: $O(LT)$
- Is it possible to achieve a lower time complexity?
- Use Windowing: attention mechanism considers only positions within the context window
Integration with Language Model

- ARSG implicitly learns how an output symbol depends on previous ones.
- Transcripts of training sets not sufficient to learn a high-quality language model.
- Problem: Standard speech recognition tasks use word-base language models whereas ARSG is a character based model.
Integration with Language Model

- Use Weighted Finite State Transducer to build character level model

Sample WFST
Integration with Language Model

Let $L$ be the lexicon FST and $G$ be the language model.

\[
newFST = \min\left(\det(L \circ G)\right)
\]

\[
L = -\log p_{ED}(y|x) - \beta \log p_{LM}(y) - \gamma T
\]

To minimize $L$, approximate the value of $y$ using beam search.
Integration with Language Model

Beam Search

- Choose k most likely candidates at each step
- Expand on those candidates. Repeat.
- To avoid underflow, use natural logarithm
- Terminate when certain conditions are met (EOS, threshold etc)
Results
Data

- Wall Street Journal Corpus
- 81 hour long with a set of 37K sequences
- Feature Representation: MFCCs
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>CER</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder-Decoder</td>
<td>6.4</td>
<td>18.6</td>
</tr>
<tr>
<td>Encoder-Decoder + bigram LM</td>
<td>5.3</td>
<td>11.7</td>
</tr>
<tr>
<td>Encoder-Decoder + trigram LM</td>
<td>4.8</td>
<td>10.8</td>
</tr>
<tr>
<td>Encoder-Decoder + extended trigram LM</td>
<td>3.9</td>
<td>9.3</td>
</tr>
<tr>
<td>Graves and Jaitly (2014)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTC</td>
<td>9.2</td>
<td>30.1</td>
</tr>
<tr>
<td>CTC, expected transcription loss</td>
<td>8.4</td>
<td>27.3</td>
</tr>
<tr>
<td>Hannun et al. (2014)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTC</td>
<td>10.0</td>
<td>35.8</td>
</tr>
<tr>
<td>CTC + bigram LM</td>
<td>5.7</td>
<td>14.1</td>
</tr>
<tr>
<td>Miao et al. (2015),</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTC for phonemes + lexicon</td>
<td>-</td>
<td>26.9</td>
</tr>
<tr>
<td>CTC for phonemes + trigram LM</td>
<td>-</td>
<td>7.3</td>
</tr>
<tr>
<td>CTC + trigram LM</td>
<td>-</td>
<td>9.0</td>
</tr>
</tbody>
</table>
Discussion Questions

- Why is ARSG performing better than CTC, inspite of not using a language model?
- Would a small dataset (as used in the paper) affect ARSG’s performance?