Motivation

- Traditional Automatic Speech Recognition (ASR) systems are complex with many moving parts: acoustic model, language model, lexicon, etc.
- End-to-end ASR maps acoustics directly to text, jointly optimizing for the recognition task
- End-to-end models do not require explicit phonetic supervision (e.g., phonemes)

Research questions:

- Do end-to-end models implicitly learn phonetic representations (“g” in “bought”)?
- Which components capture more phonetic information?
- Do more complicated ASR models learn better representations for phonology?

Methodology and Data

- Methodology
  - Train ASR model on transcribed speech
  - Extract features from the pre-trained model on a supervised dataset with phonetic segmentation
  - Train a simple classifier on a frame classification task: predict phones using the extracted features
- Classifier
  - One hidden layer, dropout, ReLU, softmax
  - Adam optimizer, cross-entropy loss

Data

- ASR training: LibriSpeech, 1000 hours of read speech
- Frame classifier: TIMIT, time segmentation of phones

<table>
<thead>
<tr>
<th>Data</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterances</td>
<td>3,692</td>
<td>400</td>
<td>192</td>
</tr>
<tr>
<td>Frames</td>
<td>988K</td>
<td>108K</td>
<td>50K</td>
</tr>
</tbody>
</table>

Analysis

- Effect of blank symbols
  - With strides, better representations at blanks
  - Without strides, better representations at non-blanks

ASR Model

- DeepSpeech2 (Amodei et al. 2017):
  - Map spectrograms to characters (or blanks)
  - Stack of CNNs and RNNs

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Input Size</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cnn1</td>
<td>161</td>
<td>41x11</td>
</tr>
<tr>
<td>2</td>
<td>cnn2</td>
<td>41x11</td>
<td>21x11</td>
</tr>
<tr>
<td>3</td>
<td>rnn1</td>
<td>1312</td>
<td>1760</td>
</tr>
<tr>
<td>4</td>
<td>rnn2</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>5</td>
<td>rnn3</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>6</td>
<td>rnn4</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>7</td>
<td>rnn5</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>8</td>
<td>rnn6</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>9</td>
<td>rnn7</td>
<td>1760</td>
<td>1760</td>
</tr>
<tr>
<td>10</td>
<td>fc</td>
<td>1760</td>
<td>29</td>
</tr>
</tbody>
</table>

- CTC loss (Graves 2006)
  - Map spectrograms x to characters / by considering all possible alignments π
  
  \[ p(l|x) = \sum_{\pi \in B^{-1}(l)} p(\pi | x) = \sum_{\pi \in B^{-1}(l)} \prod_{t=1}^{T} \phi_t(x)[\pi_t] \]
  
  - where \( \phi_t(x) \in \mathbb{R}^V \) - output at time t

Results

- Main results
  - Conv1 improves the input representation, but conv2 degrades it
  - RNN layers initially improve, then drop
  - Higher layers capture more global information like dependencies between characters (e.g., “bought”)  
  - Similar trends in different configurations (layers, phone classes, input futures)

- Model complexity
  - LSTM layer representations are better than RNN, but the respective conv layers are worse
  - Deeper model has better WER (12 vs 15) but worse representations for phonology

- Effect of strides
  - Similar overall trend
  - Less spiky shape without strides, possibly thanks to higher time resolution

- CTC models learn substantial phonetic information
  - Phonetic information persists until mid-layers, but the top layers lose phonetic information
  - Separability in vector space corresponds to representation quality

Conclusion

Acknowledgements: This work was supported by the Qatar Computing Research Institute (QCRI).