Speech to Text Translation Using Google Speech Commands

Hyung Lee and Amita C. Patil
hyunglee@stanford.edu, amita2@stanford.edu

Audio Commands to Text Translation on the Edge
- We use three ML techniques: Multinomial Logistic Regression, Hidden Markov Models with Mixture of Gaussians, CNN and compare with oracle

Google speech commands[1]: 105,829 audio files of people speaking single-word commands
- 20 core words e.g. “zero”, “one”, “yes”, “no”, “stop”, “go” which most speakers say 5 times
- 10 auxiliary words e.g. “bird”, “sheila” which most speakers say once

Objective
- Audio Commands to Text Translation on the Edge

Dataset and Features
- Google speech commands: 105,829 audio files of people speaking single-word commands
- 20 core words e.g. “zero”, “one”, “yes”, “no”, “stop”, “go” which most speakers say 5 times
- 10 auxiliary words e.g. “bird”, “sheila” which most speakers say once

Results On Test Set

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Accuracy (weighted)</th>
<th>Precision (weighted)</th>
<th>Recall (weighted)</th>
<th>F1-score (weighted)</th>
<th># of test examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial LR</td>
<td>GSC v2 numbers 0 to 9</td>
<td>0.23</td>
<td>0.39</td>
<td>0.17</td>
<td>0.23</td>
<td>7782</td>
</tr>
<tr>
<td>HMM-GMM</td>
<td>GSC v2 numbers 0 to 9</td>
<td>0.66</td>
<td>0.67</td>
<td>0.66</td>
<td>0.65</td>
<td>4107</td>
</tr>
<tr>
<td>HMM-GMM</td>
<td>command words</td>
<td>0.65</td>
<td>0.65</td>
<td>0.65</td>
<td>0.65</td>
<td>2791</td>
</tr>
<tr>
<td>HMM-GMM</td>
<td>Our voice numbers 0 to 9</td>
<td>0.83</td>
<td>0.80</td>
<td>0.83</td>
<td>0.79</td>
<td>30</td>
</tr>
<tr>
<td>CNN</td>
<td>GSC v2 numbers 0 to 9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>7782</td>
</tr>
</tbody>
</table>

Summary / Future Work
- Multinomial Logistic Regression is too simplistic, does not identify speech commands based on MFCC features
- HMM with Mixture of Gaussians has better accuracy uses relatively small number of model parameters and works better on small dataset (one speaker, 130 examples), thus more suitable for edge devices
- Our CNN model was able to achieve accuracy of 90% on 10 command words
- Given more time we would try to improve accuracy on ALL google speech commands
  - By combining probabilistic HMM model with NN, and
  - Using RNN/LSTM

References

Input is a tensor with shape (number of examples) x (width) x (height) x (depth)
Convolutional layers turn the input into a feature map using kernels and receptive fields
Pooling employed as a means of downsampling
Fully connected layer and loss layer produce classes

As of late 2017, Google boasted a 95% word accuracy rate for U.S. English; the highest out of all the voice-assistants currently out there. This translates to a 4.9% word error rate – making Google the first to fall below the 5% threshold.
Link to YouTube video

https://youtu.be/vq1I5nr3zWs