Overview

❖ Task Definition
❖ DNNs for Text-Dependent Speaker Verification.
❖ Frame-level Vs Utterance-level Features
❖ Recurrent Neural Networks
❖ Experimental Results
❖ Conclusion
Task

- Single Pass-phrase Text-Dependent Speaker Verification.
  - Allows us to focus on speaker modelling without worrying about phonetic variability.
  - Previous work based on this speaker verification paradigm study the same task (but with much more data)
- **Biggest Challenge:** Amount of background data available to train neural networks (~ 100 speakers)
DNNs for Text-Dependent Speaker Verification

- A Feedforward DNN is trained to learn a mapping from speech frames to speaker labels.
- Once trained, the network can be used as a feature extractor for the runtime speech data.
- Utterance-level speaker features can be fed to a backend classifier like cosine distance.
Frame-level Features

- DNN is trained to learn a mapping from 300 ms frame of speech to a speaker label [Variani et. al].
  - d-vector approach
- After training is complete, the network can be used as a feature extractor by forward propagating speech frames through the network and collecting the output of the hidden layer.
- Utterance-level speaker features are generated by averaging all the (forward-propagated) frames of a recording.
Recently Google introduced a framework for utterance-level modelling using both DNNs and RNNs [Heigold et. al].

Core idea is to learn a mapping from a global, utterance-level representation to a speaker label.

This can be done with a DNN or RNN - RNN does better.

They evaluate the approach for two kinds of loss functions:

- Softmax loss
- End-to-End loss: Big deal!

The authors note that the main reason for the improvement over d-vectors is the use utterance-level modelling vs frame-level modelling.
Utterance-level Features

- The end-to-end loss performs slightly better than the softmax loss.
- Does not require a separate backend.

**Dataset**
- Small: 4000 speakers, 2 Million Recordings
- Large: 80,000 speakers, 22 Million Recordings

- End-to-end loss performs better than softmax loss on both datasets, and the improvement is more pronounced on the larger training set.
- It is worth noting that the utterance-level modelling approach uses a much larger training set than the original d-vector paper. This suggests that the d-vector approach may be more suitable in a low-data regime.

- We focus on the softmax loss and using RNNs for utterance level modelling.
Recurrent Neural Networks

- Extend feedforward neural networks to sequences of arbitrary length with the help of a recurrent connection.
- Have enjoyed great success in sequential prediction tasks like speech recognition and machine translation.
- Can be viewed as a feedforward network by unrolling the computational graph - `Deep in Time`
  - RNNs can be trained in essentially the same way as DNNs, i.e. using a gradient descent based algorithm and backpropagation (through time).
- For a sequence $X = \{x_1, x_2, \ldots, x_T\}$, a RNN produces a sequence of hidden activations $H = \{h_1, h_2, \ldots, h_T\}$
  - $h_T$ can be interpreted as a summary of the sequence [Sutskever et. al].
Speaker Modelling: Utterance Summary

Forward Pass:

- $h_{forward} = f(W_1.x_t + W_2.h_{t-1} + b_1)$ \hspace{1cm} t = 1,2,…,T
- $h_{backward} = f(W_3.x_t + W_4.h_{t-1} + b_2)$

Hidden Activations:

- $h_t = [h_{t-forward}; h_{t-backward}]$

Summary Vector:

- $h_{summary} = h_T$

Classification:

- $O = \text{softmax}(h_{summary})$
  \hspace{1cm} f = \text{Non-linearity}
  \hspace{1cm} O = \text{Network output}$
Speaker Modelling: Averaged Speaker Representation

- Summary vector approach discards potentially useful information.
- A simple approach is to average all the hidden activations.

Hidden Activations:

\[ h_t = [h_t{}^{\text{forward}}; h_t{}^{\text{backward}}] \quad t = 1,2,\ldots, T \]

Utterance-level feature:

\[ h_{avg} = \frac{1}{T} \sum_{i=1}^{T} h_i \]

Classification:

\[ O = \text{softmax}(h_{avg}) \]
Speaker Modelling: Learning a weighted speaker feature

- This model takes a weighted-sum of the hidden activations.
- The weights are learned using a single-layer neural network that outputs a sigmoid.
- Approach is motivated by neural attention models [Badhanu et. al]

\[ h_i = [h_{i\text{-forward}}, h_{i\text{-backward}}] \quad i = 1, 2, ..., T \]

\[ a_i = c(h_i) \]

\[ a = [a_1, a_2, ..., a_N] \]

\[ s = \sum_i e^{a_i} \]

\[ h_{\text{weighted}} = \sum_{i=1}^{T} h_i \cdot s_i \] Utterance-level feature

\[ O = \text{softmax}(h_{\text{weighted}}) \] Classification
Experimental Setup

❖ DATA

❖ Single Passphrase (German)
Each background speaker is recorded multiple times on 3 channels - Data, land-line and cellular

❖ Training: 1547 recording, 98 speakers (male + female)
Enrolment: 230 models (multiple recordings)
Test: 1164 recordings

❖ SPEECH FEATURES
20-dimensional MFCC (static)
DNN Results

All DNN models perform substantially worse than a GMM-UBM system.

Regularization and special purpose units (Maxout) help performance.
RNN Results

RNN models perform worse than the DNN models. However, the RNN models are exposed to a smaller number of training data points.

The weighted-sum RNN model achieves the best speaker verification performance of the RNN models, with an EER of 8.84%.

We did not use dropout or any other regularization while training RNNs. This may also contribute to the worse performance of the RNNs.

<table>
<thead>
<tr>
<th>Network</th>
<th>Architecture</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unidirectional</td>
<td>spk-softmax</td>
<td>13.53</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>spk-softmax</td>
<td>11.60</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>avg-softmax</td>
<td>10.45</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>attn-softmax</td>
<td>8.84</td>
</tr>
</tbody>
</table>
Conclusions

❖ DNNs are able to outperform RNNs on the single pass-phrase task. This is contrary to Google’s results that show that utterance-level features are clearly superior, provided a very large training set is available.

❖ One possible reason for this is we attempt to train DNN and RNN models to discriminate between 98 speakers.

❖ The RNN appears to overfit the training data too easily, especially without any regularization.

❖ On the other hand the DNN learns to map individual frames to speaker labels, which is a harder task. This allows it to learn a somewhat more robust speaker representation.

❖ Regularization methods have been shown to be helpful/necessary in conjunction with a softmax loss.

❖ In closed-set speaker identification experiments (on the validation set), the weighted feature RNN model achieved 82% accuracy while the DNN achieved 98%.

❖ This suggests that neural network models can normalize out channel effects but not model new speakers effectively, given the data constraints of this study.
Ongoing & Future work
Why have DNN approaches that have been so successful in face verification not translated to speaker verification?

- **Diversity of Training Data**
  - Face verification is most similar to the text-dependent speaker verification paradigm. The main difference is that while the number of examples per class is similar (10-15), the number of classes (unique faces) is a few thousand. Compare this to the 98 classes (speakers) used in this work.

- **Variable Length Problem**
  - Variation in terms of recording length is a major problem in speaker verification. At shorter time-scales it becomes important to control for phonetic variability.
Why have DNNs only worked when applied indirectly to Speaker Verification?

- Speech Recognition DNN is used to collect sufficient statistics for i-vector training.
- The speech recognizer can be used to produce both senone posteriors and bottle-neck features.
- When the same approach is applied using a speaker discriminative DNN, the results are much worse.
  - We performed such an experiment using the RSR part-3 dataset. While this is a text-dependent task, there is a mismatch between enrolment and test recording regarding the order of phonetic events. The results we obtained were not publishable.
- A major difference between face speaker verification is the variable-duration problem. In face verification images are normalized to be the same size.
Experiment: Full length Utterances

- A DNN was trained to learn a mapping from i-vectors to speaker labels.
- After training the network is used as a feature extractor.
- Training was done a subset of Mixer and Switchboard speakers.
- Model achieves 2.15% EER as compared to 1.73% achieved by a PLDA classifier trained on the same set.
- DNNs can directly be applied to speaker verification - when long utterances are available.
What architecture would be suitable for shorter time-scales?

- The order of phonetic events is a major source of variability at shorter time-scales.
- Ideally we would like a model that that learns a representation that is invariant to this ordering.
- This is one of the most prominent features of the representations learnt by a Convolutional Neural Networks (CNNs).
  - CNNs have been successfully been applied to language identification [Lozano et. al].
  - CNNs have been used to process images of arbitrary size [Long et. al].
What should be done about the backend?
References


Thank You