(Towards) next generation acoustic models for speech recognition

Erik McDermott
Google Inc.
It takes a village

... and 250 more colleagues in the Speech team
Overview

• The past: some recent history

• The present: the “conventional” state-of-the-art, from the perspective of Farfield / Google Home.

• The future is already here? End2End.

• Longer-term: Deep Generative approach?
Google Speech Group
Early Days “Mobile”

• Speech group started in earnest in 2005

• Build up our own technology, first application launched in April 2007

• Simple directory assistance

• Early view of what a “dialer” could be
Launched early 2009 as part of Google Voice

Voicemail transcription:
- navigation
- search
- information extraction
Google Speech Group
Early Days YouTube

Launched early 2010
• automatic captioning
• translation
• editing, “time sync”
• navigation
The Revolution

• Early speech applications had some traction but nothing like the engagement we see today

• The 2007 launch of smartphones (iPhone and Android) was a revolution and dramatically changed the status of speech processing

• Our current suite of mobile applications is launched in 100+ languages and processes several centuries of speech each week
Mobile Application Overview

Context: contacts

Speech: A

Result: W, search, action, speech

HotWord: OK Google

Model

Recognizer

Result Processing

argmax P(W | A)

W

Web Search

Text-To-Speech
Recognition Models

- **Language Model**
- **Lexicon**
- **Acoustic Model**

**Deep Neural Networks**

**Multi-lingual**

- Domain/Text Norm: 7:15AM $3.22
- Dynamic Lexical Items: Contact Names
- Size/Generalization: goredforwomen.org

Acoustic Units/Context/Distribution Estimation

P(W)
P(A | W)

**Finite State Transducers**

Lexical

Acoustic
App Context vs. Technology

Mobile makes use of accurate speech recognition compelling

Large volume use improves statistical models

Accuracy Gains from Data and Modeling

- Initial results using DNNs in hybrid systems showed large gains (GMM 16.0% to DNN 12.2% with about 2k hours on VoiceSearch task)
- Additional gains from larger models
- Application of sequence models and sequence training

<table>
<thead>
<tr>
<th>Model Type</th>
<th>DNN</th>
<th></th>
<th>LSTM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objective</strong></td>
<td>CE</td>
<td>Sequence</td>
<td>CE</td>
<td>Sequence</td>
</tr>
<tr>
<td><strong>WER</strong></td>
<td>11.3</td>
<td>10.4</td>
<td>10.7</td>
<td>9.8</td>
</tr>
</tbody>
</table>
Long Short Term Memory

- Facilitates BPTT compared to vanilla RNNs.
- Trains efficiently.

\[ P(S | x_t) \]
Optimization with TensorFlow

• \{CE, CTC\} + \{sMBR, WMBR\}
• No observable differences between CE and CTC
• On-the-fly decoding for sMBR/WMBR on CPU driving LSTMs on GPU/TPU
• WMBR based on M. Shannon’s sampling-based approach (“EMBR”, Interspeech 2017).
• CTC can learn without alignments (FwdBkwd), but typically uses alignments as constraint for better latency.
• See “End-to-end training of acoustic models for LVCSR with TensorFlow”, Variani, Bagby, McDermott & Bacchiani, Interspeech 2017
Farfield

- A new way for people to interact with the internet
- More natural interface in the home
- More social

- Non-trivial engineering challenges: reverb, noise, level differences
Data Approach

- New application, no prior data that is
  - Multi-channel
  - Reverberant
  - Noisy
- Lots of data from phone launched applications (may be noisy/reverberant, but no control)
- Bootstrap approach to build a room simulator (IMAGE method) to generate “room data” from “clean data”
Room Simulator

$T60 = 500\text{ms}, \text{SNR} = 10\text{dB}$
Study on Multi-channel processing with deep learning

- T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra and C. Kim

Training Data

• 2000 hour set from our anonymized voice search data set
• Room dimensions sampled from 100 possible configurations
• T60 reverberation ranging from 400 to 900 ms. (600ms. ave)
• Simulate an 8-channel uniform linear mic array with 2cm mic spacing
• Vary source/target speaker locations, distances from 1 to 4 meters
• Noise corruption with “daily life” and YouTube music/noise data sets
• SNR distribution ranging from 0 to 20 dB SNR
Test Data

• Evaluate on a 30k voice search utterance set, about 20 hours

• One version simulated like the training set

• Another by **re-recording**
  • In a physical room, playback the test set from a mouth simulator
  • Record from an actual mic array
  • Record speech and noise from various (different) angles
  • Post mix to get SNR variations

• The baseline is MTR trained: early work with the room simulator (DNN models) showed
  16.2% clean-clean -> 29.4% clean-noisy -> 19.6% MTR-noisy
• Added accuracy improvements from combining layers of different types.

2000 hour clean training set, 20 hour clean test set

<table>
<thead>
<tr>
<th></th>
<th>CE</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>14.6</td>
<td>13.7</td>
</tr>
<tr>
<td>CLDNN</td>
<td>13.0</td>
<td>13.1</td>
</tr>
</tbody>
</table>

2000 hour MTR training set, 20 hour noisy test set

<table>
<thead>
<tr>
<th></th>
<th>CE</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>20.3</td>
<td>18.8</td>
</tr>
<tr>
<td>CLDNN</td>
<td>19.4</td>
<td>17.4</td>
</tr>
</tbody>
</table>
Raw Waveform Models

**Input**
- M samples

**Convolution**
- N x P weights

**Max pooling**
- M+N-1 window

**Nonlinearity**
- log(ReLU(...))

**Output Targets**
- raw waveform
- M samples

**fConv**
- $x_t \in \mathbb{R}^P$

**tConv**
- raw waveform
- M samples
Raw Waveform Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Log Mel</th>
<th>Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1L3D1</td>
<td>16.2</td>
<td>16.2</td>
</tr>
<tr>
<td>L3D1</td>
<td>16.5</td>
<td>16.5</td>
</tr>
<tr>
<td>D6</td>
<td>22.3</td>
<td>23.2</td>
</tr>
</tbody>
</table>

- mel ($f_{break} = 700$ Hz)
- gammatone untrained
- random init, MTR train
- gammatone init, MTR train
- gammatone init, clean train
Multi-channel Enhancement

Localization

\[ \tau_{ij} = \frac{d(i - j) \cos(\theta)}{c} \]

\[ \hat{\tau}_{ij} = \arg\max_\tau \sum_{t=0}^{L} x_i[t] x_k[t - \tau] \]

Delay-and-Sum Beamforming

\[ y(t, \theta) = \frac{1}{M} \sum_{i} x_i[t - \tau_i(\theta)] \]
Multi-channel ASR

- Common approach separates enhancement and recognition

- Enhancement commonly done in localization, beamforming and postfiltering stages

- Filter-and-sum beamforming takes a steering delay from localization for the c-th channel $\tau_c$

$$y[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h_c[n]x_c[t - n - \tau_c]$$

- Estimation is commonly based on Minimum Variance Distortionless Response (MVDR) or Multi-channel Wiener Filtering (MWF)
Raw Waveform & Multi-Channel

- Implicitly model steering delay with P multi-channel filters
- Optimize the filter parameters directly on ASR objective akin to raw waveform single channel model.

\[
y^p[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h^p_c[n] x_c[t - n]
\]

- Implicitly model steering delay with P multi-channel filters
- Optimize the filter parameters directly on ASR objective akin to raw waveform single channel model.
Learned Filters

Filterbank center frequencies

<table>
<thead>
<tr>
<th>Filters</th>
<th>2ch (14cm)</th>
<th>4ch (4-6-4cm)</th>
<th>8ch (2cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>21.8</td>
<td>21.3</td>
<td>21.1</td>
</tr>
<tr>
<td>256</td>
<td>21.7</td>
<td>20.8</td>
<td>20.6</td>
</tr>
<tr>
<td>512</td>
<td>-</td>
<td>20.8</td>
<td>20.6</td>
</tr>
</tbody>
</table>
Removing Phase

Train a baseline system with Log-mel features and feed these as feature maps into the CLDNN

<table>
<thead>
<tr>
<th>Filters</th>
<th>2ch (14cm)</th>
<th>4ch (4-6-4cm)</th>
<th>8ch (2cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>22.0</td>
<td>21.7</td>
<td>22.0</td>
</tr>
<tr>
<td>256</td>
<td>21.8</td>
<td>21.6</td>
<td>21.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filters</th>
<th>2ch (14cm)</th>
<th>4ch (4-6-4cm)</th>
<th>8ch (2cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>21.8</td>
<td>21.3</td>
<td>21.1</td>
</tr>
<tr>
<td>256</td>
<td>21.7</td>
<td>20.8</td>
<td>20.6</td>
</tr>
</tbody>
</table>
Localization

- The multi-channel raw waveform model does both beam forming as well as localization.

- Train a Delay-and-Sum (D+S) single channel signals with the oracle Time Delay of Arrival (TDOA)

- Train a Time Aligned Multi-channel (TAM) system where we oracle TDOA align the channel inputs.

<table>
<thead>
<tr>
<th>Filters</th>
<th>1ch</th>
<th>2ch (14cm)</th>
<th>4ch (4-6-4cm)</th>
<th>8ch (2cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle D+S</td>
<td>23.5</td>
<td>22.8</td>
<td>22.5</td>
<td>22.4</td>
</tr>
<tr>
<td>Oracle TAM</td>
<td>23.5</td>
<td>21.7</td>
<td>21.3</td>
<td>21.3</td>
</tr>
<tr>
<td>Raw, not tdoa</td>
<td>23.5</td>
<td>21.8</td>
<td>21.3</td>
<td>21.1</td>
</tr>
</tbody>
</table>
WER and Filter Analysis

![Graphs and charts showing WER analysis across different parameters such as SNR, reverb time, and target to mic distance.]

Filter coefficients and beam patterns are also illustrated, indicating the impact of these parameters on WER.
Multi-Channel Raw Waveform Summary

- Performance improvements remain after sequence training
- The raw waveform models without any oracle information do better than an MVDR model that was trained with oracle TDOA and noise

<table>
<thead>
<tr>
<th>Model</th>
<th>WER-CE</th>
<th>WER-Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw 1ch</td>
<td>23.5</td>
<td>19.3</td>
</tr>
<tr>
<td>D+S, 8ch, oracle</td>
<td>22.4</td>
<td>18.8</td>
</tr>
<tr>
<td>MVDR, 8ch, oracle</td>
<td>22.5</td>
<td>18.7</td>
</tr>
<tr>
<td>raw, 2ch</td>
<td>21.8</td>
<td>18.2</td>
</tr>
<tr>
<td>raw, 4ch</td>
<td>20.8</td>
<td>17.2</td>
</tr>
<tr>
<td>raw, 8ch</td>
<td>20.6</td>
<td>17.2</td>
</tr>
</tbody>
</table>

All systems 128 filters
Factored Multi-Channel Raw Waveform

- In a first convolutional layer, apply filtering for P look-directions.
- Small number of taps to encourage learning of spatial filtering
- In a second convolutional layer, use a larger number of taps for frequency resolution. Tie filter parameters between look directions
Learned Filters

Impulse responses

Beampattern
Performance of Factored Models

- Factored performance improves on unfactored with increasing number of spatial filters
- Fixing the spatial filters to be D+S shows inferior performance

<table>
<thead>
<tr>
<th># Spatial Filters</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>2ch, unfactored</td>
<td>21.8</td>
</tr>
<tr>
<td>1</td>
<td>23.6</td>
</tr>
<tr>
<td>3</td>
<td>21.6</td>
</tr>
<tr>
<td>5</td>
<td>20.7</td>
</tr>
<tr>
<td>10</td>
<td>20.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tConv1</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>21.9</td>
</tr>
<tr>
<td>trained</td>
<td>20.9</td>
</tr>
</tbody>
</table>

P=5 “look directions”
Multi-Channel Factored Raw Waveform Summary

- Performance improvements remain after sequence training

<table>
<thead>
<tr>
<th>Model</th>
<th>WER-CE</th>
<th>WER-Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>unfactored, 2ch</td>
<td>21.8</td>
<td>18.2</td>
</tr>
<tr>
<td>factored, 2ch</td>
<td>20.4</td>
<td>17.2</td>
</tr>
<tr>
<td>unfactored 4ch</td>
<td>20.8</td>
<td>17.2</td>
</tr>
<tr>
<td>factored 4ch</td>
<td>19.6</td>
<td>16.3</td>
</tr>
</tbody>
</table>
Time-Frequency Duality

• So far, all models have been formulated in the time domain

• Given the computational cost of a convolutional operator in time, the frequency dual of elementwise multiplication is of interest.

• Early layers of the network, to be phase sensitive use complex weights.
Factored Models in Frequency

**Complex Linear Projection**

\[
Z^p_f[l] = \log \left| \sum_{k=1}^{N} W^p_f[l, k] \right|
\]

**Linear Projection of Energy**

\[
Z^p_f[l] = G_f \times (\hat{Y}^p[l])^\alpha
\]

\[
W^p_f[l] = Y^p_f[l] \cdot G_f
\]

\[
\hat{Y}^p[l, k] = |Y^p[l, k]|^2
\]

\[
Y^p[l] = \sum_{c=1}^{C} X^p_c[l] \cdot H^p_c
\]
Frequency Model Performance

Factored

<table>
<thead>
<tr>
<th>Model</th>
<th>Spatial M+A</th>
<th>Spectral M+A</th>
<th>Total M+A</th>
<th>WER Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLP</td>
<td>10.3k</td>
<td>655.4k</td>
<td>19.6M</td>
<td>17.2</td>
</tr>
<tr>
<td>LPE</td>
<td>10.3k</td>
<td>165.1k</td>
<td>19.1M</td>
<td>17.2</td>
</tr>
</tbody>
</table>

Factored increasing the model to 64ms/1024FFT

<table>
<thead>
<tr>
<th>Model</th>
<th>Spatial M+A</th>
<th>Spectral M+A</th>
<th>Total M+A</th>
<th>WER Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>906.1k</td>
<td>33.8M</td>
<td>53.6M</td>
<td>17.1</td>
</tr>
<tr>
<td>CLP</td>
<td>20.5k</td>
<td>1.3M</td>
<td>20.2M</td>
<td>17.1</td>
</tr>
<tr>
<td>LPE</td>
<td>20.5k</td>
<td>329k</td>
<td>19.3M</td>
<td>16.9</td>
</tr>
</tbody>
</table>
Time vs. Frequency Filters

(a) Factored model, time

(b) Factored model, frequency
Re-recorded Sets

• Two test sets from re-recording with the mic array “on the coffee table” or “on the TV stand”

• Only use 2-channel models as mic array configuration changed (circular vs. linear)

<table>
<thead>
<tr>
<th>Model</th>
<th>Rev I</th>
<th>Rev II</th>
<th>Rev I Noisy</th>
<th>Rev II Noisy</th>
<th>Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ch raw</td>
<td>18.6</td>
<td>18.5</td>
<td>27.8</td>
<td>26.7</td>
<td>22.9</td>
</tr>
<tr>
<td>2ch raw, unfactored</td>
<td>17.9</td>
<td>17.6</td>
<td>25.9</td>
<td>24.7</td>
<td>21.5</td>
</tr>
<tr>
<td>2ch raw, factored</td>
<td>17.1</td>
<td>16.9</td>
<td>24.6</td>
<td>24.2</td>
<td>20.7</td>
</tr>
<tr>
<td>2ch CLP, factored</td>
<td>17.4</td>
<td>16.8</td>
<td>25.2</td>
<td>23.5</td>
<td>20.7</td>
</tr>
<tr>
<td>2ch raw, NAB</td>
<td>17.8</td>
<td>18.1</td>
<td>27.1</td>
<td>26.1</td>
<td>22.3</td>
</tr>
</tbody>
</table>
Google Home recent setup

- “Acoustic modeling for Google Home”, Li et al., Interspeech 2017
- 100 MTR room configurations $\rightarrow$ 4 million room configurations (Kim et al., Interspeech 2017)
- 2000 hours $\rightarrow$ 18,000 hours Voice Search training data
- Use of 4000 hours of Home real world traffic.
- Online Weighted Prediction Error (WPE) (based on Yoshioka & Nakatani)
- factored CLP; CLDNN $\rightarrow$ GridLSTM
Google Home recent results

WERs on Home eval set

<table>
<thead>
<tr>
<th>Model</th>
<th>Full</th>
<th>Clean</th>
<th>Noise Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Speech</td>
</tr>
<tr>
<td>prod</td>
<td>6.1</td>
<td>5.1</td>
<td>8.5</td>
</tr>
<tr>
<td>home</td>
<td>5.1</td>
<td>4.9</td>
<td>6.3</td>
</tr>
<tr>
<td>home(adapt)</td>
<td>4.9</td>
<td>4.7</td>
<td>6.1</td>
</tr>
</tbody>
</table>

Most utterances are simple/low-perplexity:
- weather
- play XYZ
- change volume
- etc.
End-to-End Models

- Modeling string to string directly avoids any independence assumptions and allows joint optimization of the whole model.

\[
P(y_t | x_1, ..., x_T)
\]

CTC

\[
P(y_t | y_{t-1}, x_1, ..., x_T)
\]

RNN-T

\[
P(y_i | y_1, ..., y_{i-1}, x_1, ..., x_T)
\]

LAS
Implications/Limitations

• **PROS**
  
  • Simplicity: no lexicon design, no tuning
  
  • No independence assumptions, joint optimization

• **CONS**
  
  • Need “complete data”; speech/text pairs
  
  • Not an online/streamable model
  
  • No clear input for manual design/“biasing”
  
  • Performance is poor on proper nouns / rare words.
The new state-of-the art?


• Reaching/surpassing results for standard hybrid model, e.g. CE + LSTM

• But issues with comparing results, details matter…

• .. and ongoing issues with streamability, LM biasing, rare words.

• Large number of topics to explore.
The path not (yet) taken: Waking up from the supervised, discriminative training dream?

- Is training on vast amounts of labelled training data really the future? Cost, freshness issues.

- Clearly a far vaster amount of unlabeled data is out there.

- Cf. Yan Le Cun’s plenary at ICASSP: use of predictive models, getting ground truth from the world.
ASR & TTS have grown closer, but are still quite distinct

• ASR: Limited generative models & discriminative training → Much richer discriminative models
  
  [ Though Hybrid Model fakes generative character at some level ]

• TTS: Limited generative models → Much richer generative models

• How about a deep generative model for ASR?
Discriminative vs. generative models for ASR

- Discriminative “end-to-end” model, e.g. LAS

\[ P(w|x) = \prod_k P(w_k|w_1, ..., w_{k-1}, A_k(x)) \]  

(1)

- Combine with separate language model & sequence training:

\[ Blend(x, w) = P(w|x)^\alpha \times P(w)^{1-\alpha} \]  

(2)

- Cf. generative model:

\[ p(x, w) = p(x|w) \times P(w) \]  

(3)

\[ P(w) = \prod_k P(w_k|w_1, ..., w_{k-1}) \]  

(4)

\[ p(x|w) = \prod_t p(x_t|x_1, ..., x_{t-1}, w) \]  

(5)
• WaveNet (van den Oord et al. 2016):
  – Probability of a waveform (unconditioned):
    \[ p(x) = \prod_t p(x_t|x_1, \ldots, x_{t-1}), \]  
    where observed samples \( x_t \) are targets of \( N \)-way quantized softmax trained with CE, using e.g. a DNN with dilated convolutions.
  – Conditional WaveNet:
    \[ p(x|h) = \prod_t p(x_t|x_1, \ldots, x_{t-1}, h), \]  
    where the input \( h \) represents e.g. speaker and text info.

• Mixture density networks (Zen & Senior, 2014; Schuster 1997)
  \[ p(x_t|h) = \sum w(x_{1:t-1}, h) \mathcal{N}(x_t|\mu(x_{1:t-1}, h), \sigma(x_{1:t-1}, h)) \]
Deep generative model for ASR

- Define predictive, generative likelihood of observation feature vector $x_t$ conditioned on all previous $x_t$ and symbol sequence $w$:
  \[
  p(x|w) = \prod_t p(x_t|x_1, \ldots, x_{t-1}, w), \tag{9}
  \]

- Combine with LM for decoding & sequence training:
  \[
  p(x, w) = p(x|w) \ast P(w) \tag{10}
  \]
  \[
  P(w) = \prod_k P(w_k|w_1, \ldots, w_{k-1}) \tag{11}
  \]

- Cf. hybrid model for LSTMs:
  \[
  p(x|w) = \prod_t P(w_t|x_1, \ldots, x_t)/P(w_t) \tag{12}
  \]

- Cf. ideal discriminative model
  \[
  P(w|x) = \prod_k P(w_k|w_1, \ldots, w_{k-1}, x_1, \ldots, x_T) \tag{13}
  \]
Deep Mixture Density Nets for TTS, Zen & Senior, 2014
Speech Remains Exciting

• Speech technology is becoming remarkably mainstream

• Many opportunities and research questions remain to be answered to make it truly ubiquitous: devices, languages, people, applications

• Thinking is not dead: model structure vs. parameter optimization

• Wide adoption means large data opening a very large opportunity for research using machine learning
Selected References

• C.-C. Chiu et al., “State-of-the-art speech recognition with sequence-to-sequence models”, in Proc. ICASSP 2018
• R. Prabhavalkar et al., “Minimum word error rate training for attention-based sequence-to-sequence models”, in Proc. ICASSP 2018
Selected References

Selected References

- B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, K. Rao, “Multi-Dialect Speech Recognition with a Single Sequence-to-Sequence Model,” submitted to ICASSP 2018