Deep learning for end-to-end speech recognition

Liang Lu

Centre for Speech Technology Research
The University of Edinburgh

11 March 2016
Golden age for speech technology?

Speech technology is around us
Golden age for speech technology?

Driven by data
Golden age for speech technology?

Driven by data

\[ \approx 5 \times 10^9 (5 \text{ billion}) \]
Golden age for speech technology?

Driven by deep learning

Feed-forward neural network

Convolutional neural network

Unfolding

Recurrent neural network
Golden age for speech technology?

Driven by deep learning

- 2010: CD-DNN-HMM (Microsoft & Toronto)
- 2011: Sequence training, Hessian-free (IBM, Google, Academics)
- 2012: RNNs, CNNs, Maxout, Dropout, ReLU ...
- 2013: LSTM-HMM (Alex Graves, and Google, etc)
- 2014: CTC, learning from waves, complex networks (CLDNN)
- ? : Further developments
But, what is next?

- Open challenges in speech recognition
  - Efficient **adaptation** to speakers, environment, etc
  - **Distant** speech recognition, from close-talk microphone to distant microphone(s)
  - **Small footprint** models, reduce the model size for mobile devices
  - **Open-vocabulary** speech recognition
  - **Low-resource** languages
  - ...

- In this talk, I would like to revisit the fundamental architecture for speech recognition
Speech recognition problem

- Speech recognition is a typical sequence to sequence transduction problem

- Given $y = \{y_1, \cdots, y_J\}$, $y \in \mathcal{Y}$ and $X = \{x_1, \cdots, x_T\}$, compute $P(y | X)$

- However, it is difficult
  - $T \gg J$ and $T$ can be large (> 1000)
  - Large size of vocabulary $|\mathcal{Y}| \approx 60K$
  - Uncertainty and variability in features
  - Context-dependency problem
  - ...

Channel distortion + noise

A bit signal processing

Sequence of features

Sequence of labels
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

LAWRENCE R. RABINER, FELLOW, IEEE

Although initially introduced and studied in the late 1960s and early 1970s, statistical methods of Markov source or hidden Markov modeling have become increasingly popular in the last several years. There are two strong reasons why this has occurred. First, the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Second, the models, when applied properly, work very well in practice for several important applications. In this paper we attempt to carefully and methodically review the tools of statistical modeling and show selected problems in machine recognition.

I. INTRODUCTION

Real-world processes generate signals which can be characterized by distributions with unknown parameters. One of these approaches is the knowledge engineering approach. While hidden Markov learning places learning entirely in the training algorithm, the knowledge engineering approach attempts to explicitly program human knowledge about acoustic/phonetic events into the recognizer. Whereas an HMM-based search is data driven, a knowledge engineering search is typically heuristically guided.

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 11, NOVEMBER 1989

Speaker-Independent Phone Recognition Using Hidden Markov Models

KAI-FU LEE, MEMBER, IEEE, AND HSIAO-WUEN HON

Abstract—In this paper, we extend hidden Markov modeling to speaker-independent phone recognition. Using multiple codebooks of various LPC parameters and discrete HMM's, we obtain a speaker-independent phone recognition accuracy of 58.8%—73.8 percent on the TIMIT database, depending on the type of acoustic and language models used. In comparison, the performance of expert speech recognition systems is only 69 percent without use of higher level knowledge. We also introduce the co-occurrence smoothing algorithm which enables accurate recognition even with very limited training data. Since our model is composed of two independent models, a phone recognition error is likely to be a single phone error, hence the large performance improvement.

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 11, NOVEMBER 1989
Hidden Markov Models

- Why the hidden Markov model works for speech recognition?
- It converts the sequence-level classification problem into a frame-level problem

\[ P(y \mid X) \propto p(X \mid y) \]
\[ \approx p(X_{1:T} \mid Q_{1:T})P(y) \]
\[ \approx P(y) \prod_{t} p(x_t \mid q_t)p(q_t \mid q_{t-1}) \]
Hidden Markov Models

- Problems of HMMs:
  - Loss function: minimise the word error $\mathcal{L}(y, \tilde{y})$ versus maximise the likelihood $p(X_{1:T} | Q_{1:T})$
  - Conditional independence assumption
  - Weak sequence model – first order Markov rule
  - System complexity: monophone $\rightarrow$ alignment $\rightarrow$ triphone $\rightarrow$ alignment $\rightarrow$ neural net $\rightarrow$ alignment $\rightarrow$ neural net
End-to-end speech recognition

- Can we train a model that directly computes $P(y \mid X)$?

- CTC – Connectionist Temporal Classification

- Attention-based recurrent neural network (RNN) encoder-decoder

- Segmental recurrent neural networks
End-to-end speech recognition

- CTC – Connectionist Temporal Classification
  - Trick: \( \{y_1, \cdots, y_J\} \rightarrow \{\hat{y}_1, \cdots, \hat{y}_T\} \rightarrow \{x_1, \cdots, x_T\} \)
  - Replicate the labels \( (a \ b \ c \rightarrow a \ a \ b \ b \ b \ \odot \ c) \) with blank symbol \( \odot \)
  - Approximate the conditional probability

\[
P(\hat{y} \mid X) = \prod_{t=1}^{T} P(\hat{y}_t \mid x_t) \tag{1}
\]

End-to-end speech recognition

- Maximum Entropy Markov Model (MEMM)
- Still reply on the independence assumption
End-to-end speech recognition

ACOUSTIC MODELLING WITH CD-CTC-SMBR LSTM RNNs

Andrew Senior, Haşim Sak, Félix de Chaumont Quirty, Tara Sainath, Kanishka Rao

Google

{hasim, andrewsenior, fcq, tsainath, kanishkarao}@google.com

ABSTRACT
This paper describes a series of experiments to extend the application of Context-Dependent (CD) long short-term memory (LSTM) recurrent neural networks (RNNs) trained with Connectionist Temporal Classification (CTC) and sMBR loss. Our experiments, on a noisy, reverberant voice search task, include training with alternative pronunciations and the application to child speech recognition; combination of multiple models, and convolutional input layers. We also investigate the latency of CTC models and show that constraining forward-backward alignment in training can reduce the delay for a real-time streaming speech recognition system. Finally we investigate transferring knowledge from one network to another through alignments.

Index Terms: Long Short Term Memory, Recurrent Neural Networks, Connectionist Temporal Classification, sequence discriminative training, knowledge transfer.
End-to-end speech recognition

- Attention-based RNN encoder-decoder

\[ P(y \mid X) \approx \prod_j P(y_j \mid y_1, \cdots, y_{j-1}, c_j) \]  \hspace{1cm} (2)

\[ h_{1:T} = \text{RNN}(x_{1:T}) \]  \hspace{1cm} (3)

\[ c_j = \text{Attend}(h_{1:T}) \]  \hspace{1cm} (4)

End-to-end speech recognition

- Attention-based RNN encoder-decoder
End-to-end speech recognition

- Attention-based RNN encoder-decoder
End-to-end speech recognition

- Attention-based RNN encoder-decoder
End-to-end speech recognition

- Attention-based RNN encoder-decoder
End-to-end speech recognition

- Attention-based RNN encoder-decoder
End-to-end speech recognition

- Attention-based RNN encoder-decoder

\[
\begin{align*}
&\text{Encoder} & \h_1:T &= \text{RNN}(\mathbf{x}_{1:T}) \\
&\text{Attention} & \mathbf{c}_j &= \text{Attend}(\h_{1:T}) \\
&\text{Decoder} & P(y_j | y_1, \ldots, y_{j-1}, \mathbf{c}_j) \\
& & \\
\end{align*}
\]
End-to-end speech recognition

- Attention-based RNN encoder-decoder
  - A flexible sequence-to-sequence transducer
  - “Revolutionising” machine translation
  - Popularising the attention-based scheme
  - But it may not be a natural model for speech recognition, why?
End-to-end speech recognition

- Segmental recurrent neural network – segmental CRF + RNN


[3] Many many more on (segmental) CRFs
• CRF [Lafferty et al. 2001]

\[
P(y \mid X) = \frac{1}{Z(X)} \prod_j \exp \left( w^\top \Phi(y_j, X) \right) \tag{5}
\]

where the length of \( y \) and \( X \) should be equal.

• Segmental (semi-Markov) CRF [Sarawagi and Cohen 2004]

\[
P(y, E, \mid X) = \frac{1}{Z(X)} \prod_j \exp \left( w^\top \Phi(y_j, e_j, X) \right) \tag{6}
\]

where \( e_j = \langle s_j, n_j \rangle \) denotes the beginning \((s_j)\) and end \((n_j)\) time tag of \( y_j \); \( E = \{e_1, \cdots, e_J\} \) is the latent segment label.
Segmental recurrent neural network

- Segmental recurrent neural network – using neural networks to learn the feature function $\Phi(\cdot)$. 

![Diagram of a segmental recurrent neural network](image-url)
Segmental recurrent neural network

- Training criteria
  - Conditional maximum likelihood
    \[ L(\theta) = \log P(y \mid X) = \log \sum_{E} P(y, E \mid X) \] (7)
  - Max-margin – maximising the distance between the ground truth and negative labels
    \[ L(\theta) = \sum_{\hat{y} \in \Omega} \mathcal{D}(y, \hat{y}) \] model distance (8)

Segmental recurrent neural network

- Viterbi decoding
  - Partially Viterbi decoding
    \[ y^* = \arg \max_y \log \sum_E P(y, E | X) \]  
  - Fully Viterbi decoding
    \[ y^*, E^* = \arg \max_{y,E} \log P(y, E | X) \]  

Experiment 1

- **TIMIT dataset**
  - 3696 training utterances (∼ 3 hours)
  - core test set (192 testing utterances)
  - trained on 48 phonemes, and mapped to 39 for scoring
  - log filterbank features (FBANK)
  - using LSTM as an implementation of RNN
Experiment 1

- Speed up training
## Experiment 1

**Table:** Results of hierarchical subsampling networks.

<table>
<thead>
<tr>
<th>System</th>
<th>LSTM layers</th>
<th>hidden</th>
<th>PER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>skip</td>
<td>3</td>
<td>128</td>
<td>21.2</td>
</tr>
<tr>
<td>conc</td>
<td>3</td>
<td>128</td>
<td>21.3</td>
</tr>
<tr>
<td>add</td>
<td>3</td>
<td>128</td>
<td>23.2</td>
</tr>
<tr>
<td>skip</td>
<td>3</td>
<td>250</td>
<td>20.1</td>
</tr>
<tr>
<td>conc</td>
<td>3</td>
<td>250</td>
<td>20.5</td>
</tr>
<tr>
<td>add</td>
<td>3</td>
<td>250</td>
<td>21.5</td>
</tr>
</tbody>
</table>
# Experiment 1

**Table**: Results of tuning the hyperparameters.

<table>
<thead>
<tr>
<th>Dropout</th>
<th>layers</th>
<th>hidden</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>3</td>
<td>128</td>
<td>21.2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>250</td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>250</td>
<td><strong>19.3</strong></td>
</tr>
<tr>
<td>0.1</td>
<td>3</td>
<td>128</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>250</td>
<td>20.9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>250</td>
<td>20.4</td>
</tr>
<tr>
<td>×</td>
<td>6</td>
<td>250</td>
<td>21.9</td>
</tr>
</tbody>
</table>
Experiment 1

Table: Results of three types of acoustic features.

<table>
<thead>
<tr>
<th>Features</th>
<th>Deltas</th>
<th>$d(x_t)$</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-dim FBANK</td>
<td>√</td>
<td>72</td>
<td>19.3</td>
</tr>
<tr>
<td>40-dim FBANK</td>
<td>√</td>
<td>120</td>
<td>18.9</td>
</tr>
<tr>
<td>Kaldi</td>
<td>×</td>
<td>40</td>
<td>17.3</td>
</tr>
</tbody>
</table>

Kaldi features – 39 dimensional MFCCs spliced by a context window of 7, followed by LDA and MLLT transform and with feature-space speaker-dependent MLLR
### Experiment 1

Table: Comparison to related works.

<table>
<thead>
<tr>
<th>System</th>
<th>LM</th>
<th>SD</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-DNN</td>
<td>✓</td>
<td>✓</td>
<td>18.5</td>
</tr>
<tr>
<td>first-pass SCRF [Zweig 2012]</td>
<td>✓</td>
<td>×</td>
<td>33.1</td>
</tr>
<tr>
<td>Boundary-factored SCRF [He 2012]</td>
<td>×</td>
<td>×</td>
<td>26.5</td>
</tr>
<tr>
<td>+ 2nd pass with various features</td>
<td>✓</td>
<td>×</td>
<td>19.9</td>
</tr>
<tr>
<td>CTC [Graves 2013]</td>
<td>×</td>
<td>×</td>
<td>18.4</td>
</tr>
<tr>
<td>RNN transducer [Graves 2013]</td>
<td>-</td>
<td>×</td>
<td>17.7</td>
</tr>
<tr>
<td>Attention-based RNN baseline [Chorowski 2015]</td>
<td>-</td>
<td>×</td>
<td>17.6</td>
</tr>
<tr>
<td>Segmental RNN</td>
<td>×</td>
<td>×</td>
<td>18.9</td>
</tr>
<tr>
<td>Segmental RNN</td>
<td>×</td>
<td>✓</td>
<td>17.3</td>
</tr>
</tbody>
</table>
Experiment 2

- Switchboard dataset ($\sim 300$ hours $\approx 100$ million frames)
- Attention-based RNN systems (EncDec)
- No language model in baseline systems

**Table:** Attention-Based RNN vs. CTC and DNN-HMM hybrid systems.

<table>
<thead>
<tr>
<th>System</th>
<th>Output</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-DNN sMBR [Vesely 2013]</td>
<td>-</td>
<td>18.4</td>
</tr>
<tr>
<td>CTC no LM [Maas 2015]</td>
<td>char</td>
<td><strong>47.1</strong></td>
</tr>
<tr>
<td>+ 7-gram</td>
<td>char</td>
<td>35.9</td>
</tr>
<tr>
<td>+ RNNLM (3 hidden layers)</td>
<td>char</td>
<td>30.8</td>
</tr>
<tr>
<td>Deep Speech [Hannun 2014]</td>
<td>char</td>
<td>25.9</td>
</tr>
<tr>
<td>EncDec no LM</td>
<td>word</td>
<td>36.4</td>
</tr>
<tr>
<td>EncDec no LM</td>
<td>char</td>
<td><strong>37.8</strong></td>
</tr>
</tbody>
</table>
Experiment 2

- Long memory decoder

a) Baseline decoder

b) LongMem decoder
Experiment 2

**Table:** Results of language model rescoring and using long memory decoder.

<table>
<thead>
<tr>
<th>System</th>
<th>Output</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncDec no LM</td>
<td>word</td>
<td>37.6</td>
</tr>
<tr>
<td>+ LongMem</td>
<td>word</td>
<td>36.4</td>
</tr>
<tr>
<td>+ 3-gram rescoring</td>
<td>word</td>
<td><strong>36.0</strong></td>
</tr>
<tr>
<td>EncDec no LM</td>
<td>char</td>
<td>42.8</td>
</tr>
<tr>
<td>+ LongMem</td>
<td>char</td>
<td>41.3</td>
</tr>
<tr>
<td>+ 5-gram rescoring</td>
<td>char</td>
<td>40.5</td>
</tr>
</tbody>
</table>


Summary

- End-to-end speech recognition is a new and exiting research area
- Three new models have been discussed
  - Connectionist Temporal Classification (CTC)
  - Attention-based recurrent neural network
  - Segmental recurrent neural network
Acknowledgement

- Joint work with
  - Xingxing Zhang (Ph.D student at Edinburgh)
  - Lingpeng Kong (Ph.D student at CMU)

- Funded by the NST project
Thank you! Questions?