CS224D: Deep Learning for Natural Language Processing

Andrew Maas
Spring 2016

Neural Networks in Speech Recognition
Outline

• Speech recognition systems overview
• HMM-DNN (Hybrid) acoustic modeling
• What’s different about modern HMM-DNNs?
• HMM-free RNN recognition
Are there any good robot movies I can rent tonight?
but it was really nice to get back with a telephone and the city and everything and you know yeah

well (i-) the only way i could bear it was to (pass) (some) to be asleep i was like well it is not gonna (be-) get over until you know (w-) (w-) yeah it (re-) really i (th-) i think that is what ruined it for us
Outline

• Speech recognition systems overview
• HMM-DNN (Hybrid) acoustic modeling
• What’s different about modern HMM-DNNs?
• HMM-free RNN recognition
Acoustic Modeling with GMMs

Transcription: Samson
Sub-phones: 942 – 6 – 37 – 8006 – 4422 …

Hidden Markov Model (HMM):

Acoustic Model:

Audio Input:

GMM models: P(x|s)
x: input features
s: HMM state
DNN Hybrid Acoustic Models

Transcription: Samson
Sub-phones: 942 – 6 – 37 – 8006 – 4422 ...

Hidden Markov Model (HMM):

Use a DNN to approximate:
\[ P(s|x) \]

Apply Bayes’ Rule:
\[ P(x|s) = P(s|x) \times P(x) / P(s) \]

DNN * Constant / State prior

Audio Input:
Not Really a New Idea

TRAINING

Speech → Front End → Features → MLP → Estimated Phone Labels → Targets

RECOGNITION

Speech → Front End → Features → MLP → Estimated Phone Probabilities → Viterbi Alignment (HMM) → Word Sequence

Modern Systems use **DNNs** and **Senones**

Comparison of Context-Independent Monophone State Labels and Context-Dependent Triphone Senone Labels

<table>
<thead>
<tr>
<th># Hidden Layers</th>
<th># Hidden Units</th>
<th>Label Type</th>
<th>Dev Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2K</td>
<td>Monophone States</td>
<td>59.3%</td>
</tr>
<tr>
<td>1</td>
<td>2K</td>
<td>Triphone Senones</td>
<td>68.1%</td>
</tr>
<tr>
<td>3</td>
<td>2K</td>
<td>Monophone States</td>
<td>64.2%</td>
</tr>
<tr>
<td>3</td>
<td>2K</td>
<td>Triphone Senones</td>
<td>69.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Dev Accuracy</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>62.9%</td>
<td>60.4%</td>
</tr>
<tr>
<td>MMI</td>
<td>65.1%</td>
<td>62.8%</td>
</tr>
<tr>
<td>MPE</td>
<td>65.5%</td>
<td>63.8%</td>
</tr>
</tbody>
</table>

Hybrid Systems now Dominate ASR

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

<table>
<thead>
<tr>
<th>TASK</th>
<th>HOURS OF TRAINING DATA</th>
<th>DNN-HMM</th>
<th>GMM-HMM WITH SAME DATA</th>
<th>GMM-HMM WITH MORE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWITCHBOARD (TEST SET 1)</td>
<td>309</td>
<td>18.5</td>
<td>27.4</td>
<td>18.6 (2,000 H)</td>
</tr>
<tr>
<td>SWITCHBOARD (TEST SET 2)</td>
<td>309</td>
<td>16.1</td>
<td>23.6</td>
<td>17.1 (2,000 H)</td>
</tr>
<tr>
<td>ENGLISH BROADCAST NEWS</td>
<td>50</td>
<td>17.5</td>
<td>18.8</td>
<td></td>
</tr>
<tr>
<td>BING VOICE SEARCH (SENTENCE ERROR RATES)</td>
<td>24</td>
<td>30.4</td>
<td>36.2</td>
<td></td>
</tr>
<tr>
<td>GOOGLE VOICE INPUT</td>
<td>5,870</td>
<td>12.3</td>
<td></td>
<td>16.0 (&gt;&gt; 5,870 H)</td>
</tr>
<tr>
<td>YOUTUBE</td>
<td>1,400</td>
<td>47.6</td>
<td>52.3</td>
<td></td>
</tr>
</tbody>
</table>

What’s Different in Modern DNNs?

• Fast computers = run many experiments
• Deeper nets improve on shallow nets
• Architecture choices (easiest is replacing sigmoid)
• Pre-training *matters very little*. Initially we thought this was the new trick that made things work
• Many more parameters
Warning! Depth can also act as a regularizer because it makes optimization more difficult. This is why you will sometimes see very deep networks perform well on TIMIT or other small tasks.
Replacing Sigmoid Hidden Units

(Glorot & Bengio. 2011)
Comparing Nonlinearities

Switchboard WER

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In Submission) Andrew Maas. Stanford CS224D. 2016
Scaling up NN acoustic models in 1999

0.7M total NN parameters

(Ellis & Morgan. 1999)
Adding More Parameters 15 Years Ago


Hybrid NN. 1 hidden layer. 54 HMM states. 74hr broadcast news task

“...improvements are almost always obtained by increasing either or both of the amount of training data or the number of network parameters ... We are now planning to train an 8000 hidden unit net on 150 hours of data ... this training will require over three weeks of computation.”
Adding More Parameters Now

• Comparing total number of parameters (in millions) of previous work versus our new experiments

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In Submission)  
Andrew Maas. Stanford CS224D. 2016
Combining Speech Corpora

Switchboard
- 300 hours
- 4,870 speakers

Fisher
- 2,000 hours
- 23,394 speakers

Combined corpus baseline system now available in Kaldi

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In Submission) Andrew Maas. Stanford CS224D. 2016
Scaling Total Parameters

![Graph showing scaling total parameters](image)

(Model Size)

- GMM
- 36M
- 100M
- 200M
- 400M

Frame Error Rate

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In Submission) Andrew Maas. Stanford CS224D. 2016
Scaling Total Parameters

Frame Error Rate

<table>
<thead>
<tr>
<th>Model Size</th>
<th>Frame Error Rate</th>
<th>Word Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>400M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. In Submission) Andrew Maas. Stanford CS224D. 2016
Outline

• Speech recognition systems overview
• HMM-DNN (Hybrid) acoustic modeling
• What’s different about modern HMM-DNNs?
• HMM-free RNN recognition
HMM-DNN Speech Recognition

Transcription: Samson
Sub-phones: 942 – 6 – 37 – 8006 – 4422 ...

Hidden Markov Model (HMM):

942

P(s|x₁)

942

P(s|x₂)

6

P(s|x₃)

Acoustic Model:

Features (x₁)

Features (x₂)

Features (x₃)

Use a DNN to approximate:
P(s|x)

Apply Bayes’ Rule:
P(x|s) = P(s|x) * P(x) / P(s)

DNN * Constant / State prior

Audio Input:

Andrew Maas. Stanford CS224D. 2016
HMM-Free Recognition

Transcription: Samson
Pronunciation: S – AE – N
Sub-phones: 942 – 6 – 8006 – 1422 ...

Hidden Markov Model (HMM):

\[
P(s|x_1) \quad P(s|x_2) \quad P(s|x_3)
\]

Acoustic Model:

Audio Input:

(Graves & Jaitly. 2014)
HMM-Free Recognition

Transcription: Samson

Characters: SAMSON

Collapsing function: SS___AA_M_S___O___NNNN

Acoustic Model:

Audio Input:

Use a DNN to approximate: $P(a|x)$
The distribution over characters

(Graves & Jaitly. 2014)
CTC Objective Function

Labels at each time index are conditionally independent (like HMMs)

\[
Pr(a|x) = \prod_{t=1}^{T} Pr(a_t, t|x)
\]

Sum over all time-level labelings consistent with the output label.

Output label: AB

Time-level labelings: AB, _AB, A_B, ... _A_B_

Final objective maximizes probability of true labels:

\[
CTC(x) = -\log Pr(y^*|x)
\]

(Graves & Jaitly. 2014)
Collapsing Example

Per-frame argmax:

After collapsing:
yet a rehabilitation crew is on hand in the building lugging bricks plaster and blueprints for forty two new bedroom apartments

Reference:
yet a rehabilitation crew is on hand in the building lugging bricks plaster and blueprints for forty two new bedroom apartments

(Hannun, Maas, Jurafsky, & Ng. 2014)  Andrew Maas. Stanford CS224D. 2016
Recurrence Matters!

\[
P(a|\mathbf{x}_1) \

\]

\[
P(a|\mathbf{x}_2) \

\]

\[
P(a|\mathbf{x}_3) \

\]

(Hannun, Maas, Jurafsky, & Ng. 2014)

Andrew Maas. Stanford CS224D. 2016
Decoding with a Language Model

Lexicon: [a, ..., zebra]

Language Model: \( p(\text{“yeah”} \mid \text{“oh”}) \)

Character Probabilities: __oo_h__y_e_aa_h

(Hannun, Maas, Jurafsky, & Ng. 2014)

Andrew Maas. Stanford CS224D. 2016
Rethinking Decoding

Out of Vocabulary Words
- syriza
- bae
- abo--
- schmidhuber
- sof--

Lexicon
- [a, …, zebra]

Languages
- (“yeah” | “oh”)
Lexicon-Free & HMM-Free on Switchboard

(Maas*, Xie*, Jurafsky, & Ng. 2015)

Andrew Maas. Stanford CS224D. 2016
Transcribing Out of Vocabulary Words

Truth: yeah i went into the i do not know what you think of fidelity but
HMM-GMM: yeah when the i don’t know what you think of fidelity it even them
CTC-CLM: yeah i went to i don’t know what you think of fidelity but um

Truth: no no speaking of weather do you carry a altimeter slash barometer
HMM-GMM: no i’m not all being the weather do you uh carry a uh helped emitters last brahms her
CTC-CLM: no no beating of whether do you uh carry a uh a time or less barometer

Truth: i would ima- well yeah it is i know you are able to stay home with them
HMM-GMM: i would amount well yeah it is i know um you’re able to stay home with them
CTC-CLM: i would ima- well yeah it is i know uh you’re able to stay home with them

(Maas*, Xie*, Jurafsky, & Ng. 2015)
Comparing Alignments

HMM-GMM phone probabilities

CTC character probabilities

(HMM slide from Dan Ellis)
Learning Phonemes and Timing

(Maas*, Xie*, Jurafsky, & Ng. 2015)

Andrew Maas. Stanford CS224D. 2016
Learning Phonemes and Timing

(Maas*, Xie*, Jurafsky, & Ng. 2015)
Pushing Performance with HMM-Free
CTC now powers Google search ASR

- Context-dependent states rather than characters
- Uni-directional LSTM for faster streaming
- CTC + sequence discriminative loss

http://googleresearch.blogspot.com/2015/09/google-voice-search-faster-and-more.html
(Sak, Senior, Rao, & Beaufays. 2015)
Deep Speech 2: Scaling up CTC

• Efficient GPU training
• Some recurrent architecture variants
• Data augmentation
• Works on both English and Mandarin

Table 10: Comparison of English WER for Regular and Noisy development sets on increasing training dataset size. The architecture is a 9-layer model with 2 layers of 2D-invariant convolution and 7 recurrent layers with 68M parameters.

(Amodei et al. 2015)
Listen, attend, and spell

Figure 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence $x$ into high level features $h$, the speller is an attention-based decoder generating the $y$ characters from $h$.

(Chan, Jaitly, Le, & Vinyals. 2015)
Listen, attend, and spell

Alignment between the Characters and Audio

Figure 2: Alignments between character outputs and audio signal produced by the Listen, Attend and Spell (LAS) model for the utterance “how much would a woodchuck chuck”. The content based attention mechanism was able to identify the start position in the audio sequence for the first character correctly. The alignment produced is generally monotonic without a need for any location based priors.
Conclusion

• HMM-DNN systems are now the default, state-of-the-art for speech recognition
• We roughly understand why HMM-DNNs work but older, shallow hybrid models didn’t work as well
• HMM-Free approaches are rapidly improving and making their way to production systems
• It’s a very exciting time for speech recognition
End

• More on spoken language understanding:
  – cs224s.stanford.edu

• Open source speech recognition toolkit (Kaldi):
  – Kaldi.sf.net

• Multiple open source implementations of CTC available