Recap: DNN for TIMIT

- **Deeper**: Deep neural network architecture – multiple hidden layers
- **Wider**: Use HMM state alignment as outputs rather than hand-labelled phones – 3-state HMMs, so $3 \times 48$ states
- Training many hidden layers is computationally expensive – use GPUs to provide the computational power
Context Dependent DNN Acoustic Models
DNN acoustic model for Switchboard

- 9304 CD state outputs
- 2048 hidden units
- 9x39 = 351 PLP inputs
- 7 hidden layers
- 2048 hidden units

(Hinton et al (2012))
Context-dependent hybrid HMM/DNN

- First train a context-dependent HMM/GMM system on the same data, using a phonetic decision tree to determine the HMM tied states.
- Perform Viterbi alignment using the trained HMM/GMM and the training data.
- Train a neural network to map the input speech features to a label representing a context-dependent tied HMM state.
  - So the size of the label set is thousands (number of context-dependent tied states) rather than tens (number of context-independent phones). Each frame is labelled with the Viterbi aligned tied state.
- Train the neural network using gradient descent as usual.
- Use the context-dependent scaled likelihoods obtained from the neural network when decoding.
Example: hybrid HMM/DNN large vocabulary conversational speech recognition (Switchboard)

- Recognition of American English conversational telephone speech (Switchboard)
- Baseline context-dependent HMM/GMM system
  - 9,304 tied states
  - Discriminatively trained (BMMI — similar to MPE)
  - 39-dimension PLP (+ derivatives) features
  - Trained on 309 hours of speech
- Hybrid HMM/DNN system
  - Context-dependent — 9304 output units obtained from Viterbi alignment of HMM/GMM system
  - 7 hidden layers, 2048 units per layer
- DNN-based system results in significant word error rate reduction compared with GMM-based system
DNN vs GMM on large vocabulary tasks (Experiments from 2012)

[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</td>
<td>24</td>
<td>30.4</td>
<td>36.2</td>
<td></td>
</tr>
<tr>
<td>(SENTENCE ERROR RATES)</td>
<td></td>
<td></td>
<td></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>

(Hinton et al (2012))
TDNNs
Time-delay Neural Networks
Modelling acoustic context

- DNNs allow the network to model acoustic context by including neighbouring frame in the input layer – the output is thus estimating the phone or state probability using that contextual information
- Richer NN models of acoustic context
  - **Time-delay neural networks (TDNNs)**
    - each layer processes a context window from the previous layer
    - higher hidden layers have a wider receptive field into the input
  - **Recurrent neural networks (RNNs)**
    - hidden units at time $t$ take input from their value at time $t - 1$
    - these recurrent connections allow the network to learn state
- Both approaches try to learn invariances in time, and form representations based on compressing the history of observations
TDNNs – first hidden layer receptive field

ASR Lecture 9

NNs for Acoustic Modelling 3: CD DNNs and TDNNs
TDNNs – first hidden layer receptive field
TDNNs – first hidden layer receptive field

Hidden Units

Features

Hidden Layer 1

Input Layer

Time
TDNNs – first hidden layer receptive field

Hidden Units

Features

Hidden Layer 1

Input Layer

Time

ASR Lecture 9  
NNs for Acoustic Modelling 3: CD DNNs and TDNNs
TDNNs – first hidden layer receptive field
TDNNs – first hidden layer receptive field
TDNNs – first hidden layer receptive field

```
<table>
<thead>
<tr>
<th>Time</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Layer</td>
<td>Hidden Layer 1</td>
</tr>
<tr>
<td>Hidden Units</td>
<td></td>
</tr>
</tbody>
</table>
```

ASR Lecture 9  NNs for Acoustic Modelling 3: CD DNNs and TDNNs
Higher hidden layers take input from a time window over the previous hidden layer

Lower hidden layers learn from narrower contexts, higher hidden layers from wider acoustic contexts

Receptive field increases for higher hidden layers
Higher hidden layers take input from a time window over the previous hidden layer.

Lower hidden layers learn from narrower contexts, higher hidden layers from wider acoustic contexts.

Receptive field increases for higher hidden layers.
Higher hidden layers take input from a time window over the previous hidden layer.

Lower hidden layers learn from narrower contexts, higher hidden layers from wider acoustic contexts.

Receptive field increases for higher hidden layers.
Higher hidden layers take input from a time window over the previous hidden layer.

Lower hidden layers learn from narrower contexts, higher hidden layers from wider acoustic contexts.

Receptive field increases for higher hidden layers.
Example TDNN Architecture

- View a TDNN as a 1D convolutional network with the transforms for each hidden unit tied across time.
- TDNN layer with context [-2,2] has 5x as many weights as a regular DNN layer.
- More computation, more storage required!
Example TDNN Architecture

- View a TDNN as a 1D convolutional network with the transforms for each hidden unit tied across time.
- TDNN layer with context [-2,2] has 5x as many weights as a regular DNN layer.
- More computation, more storage required!
View a TDNN as a 1D convolutional network with the transforms for each hidden unit tied across time.

- TDNN layer with context $[-2,2]$ has 5x as many weights as a regular DNN layer.
- More computation, more storage required!
Comparison with DNN with input window

Input Features

Output HMM states

Input Features

ASR Lecture 9
NNs for Acoustic Modelling 3: CD DNNs and TDNNs
Comparison with DNN with input window

Input Features

Output HMM states

Hidden layer
~700 ReLU hidden units

ASR Lecture 9
NNs for Acoustic Modelling 3: CD DNNs and TDNNs
Comparison with DNN with input window

- **Input Features**
  - $t-4$, $t$, $t+4$

- **Hidden layer**
  - ~700 ReLU hidden units

- **Incoming weights**
  - 1x700 units

- **Output HMM states**
Sub-sampled TDNN

- Sub sample window of hidden unit activations
- Large overlaps between input contexts at adjacent time steps – likely to be correlated
- Allow gaps between frames in a window (cf. dilated convolutions)
- Sub-sampling saves computation and reduces number of model size (number of weights)
Example sub-sampled TDNN

- Increase the context for higher layers of the network
- Subsampled so that difference between sampled hidden units is multiple of 3 to enable “clean” sub-sampling
- Asymmetric contexts
- MFCC features in this case

**Peddinti (2015)**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Context</th>
<th>Sub-sampled Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[-2,2]</td>
<td>[-2,2]</td>
</tr>
<tr>
<td>2</td>
<td>[-1,2]</td>
<td>{-1,2}</td>
</tr>
<tr>
<td>3</td>
<td>[-3,3]</td>
<td>{-3,3}</td>
</tr>
<tr>
<td>4</td>
<td>[-7,2]</td>
<td>{-7,2}</td>
</tr>
<tr>
<td>5</td>
<td>{0}</td>
<td>{0}</td>
</tr>
</tbody>
</table>
### Table 2: Performance comparison of DNN and TDNN with various temporal contexts

<table>
<thead>
<tr>
<th>Model</th>
<th>Network Context</th>
<th>Layerwise Context</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>DNN-A</td>
<td>[-7, 7]</td>
<td>[-7, 7]</td>
<td>{0}</td>
</tr>
<tr>
<td>DNN-A₂</td>
<td>[-7, 7]</td>
<td>[-7, 7]</td>
<td>{0}</td>
</tr>
<tr>
<td>DNN-B</td>
<td>[-13, 9]</td>
<td>[-13, 9]</td>
<td>{0}</td>
</tr>
<tr>
<td>DNN-C</td>
<td>[-16, 9]</td>
<td>[-16, 9]</td>
<td>{0}</td>
</tr>
<tr>
<td>TDNN-A</td>
<td>[-7, 7]</td>
<td>[-2, 2]</td>
<td>{-2, 2}</td>
</tr>
<tr>
<td>TDNN-B</td>
<td>[-9, 7]</td>
<td>[-2, 2]</td>
<td>{-2, 2}</td>
</tr>
<tr>
<td>TDNN-C</td>
<td>[-11, 7]</td>
<td>[-2, 2]</td>
<td>{-1, 1}</td>
</tr>
<tr>
<td>TDNN-D</td>
<td>[-13, 9]</td>
<td>[-2, 2]</td>
<td>{-1, 2}</td>
</tr>
<tr>
<td>TDNN-E</td>
<td>[-16, 9]</td>
<td>[-2, 2]</td>
<td>{-2, 2}</td>
</tr>
</tbody>
</table>

Peddinti (2015)
### DNN vs TDNN on other datasets

<table>
<thead>
<tr>
<th>Database</th>
<th>Size</th>
<th>WER</th>
<th>Rel. Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DNN</td>
<td>TDNN</td>
</tr>
<tr>
<td>Res. Management</td>
<td>3h hrs</td>
<td>2.27</td>
<td>2.30</td>
</tr>
<tr>
<td>Wall Street Journal</td>
<td>80 hrs</td>
<td>6.57</td>
<td>6.22</td>
</tr>
<tr>
<td>TedLIUM</td>
<td>118 hrs</td>
<td>19.3</td>
<td>17.9</td>
</tr>
<tr>
<td>Switchboard</td>
<td>300 hrs</td>
<td>15.5</td>
<td>14.0</td>
</tr>
<tr>
<td>Librispeech</td>
<td>960 hrs</td>
<td>5.19</td>
<td>4.83</td>
</tr>
<tr>
<td>Fisher English</td>
<td>1800 hrs</td>
<td>22.24</td>
<td>21.03</td>
</tr>
</tbody>
</table>

Peddinti (2015)
Summary and Conclusions

- Scaling DNNs for large vocabulary speech recognition
- Context-dependent DNNs – use state clusters from CD HMM/GMM as output labels – results in significant improvements in accuracy for DNNs over GMMs
- Richer temporal modelling – time-delay neural networks (TDNNs)
- Sub-sampled TDNNs
Reading


Background Reading: