Conventional Training of CD-DNN-HMMs

- CD-DNN-HMMs rely on GMM-HMMs in two aspects:
  - Training labels — state-to-frame alignments
  - Tied CD state targets — GMM-HMM based decision tree state tying

- Is it possible to build CD-DNN-HMMs independently from any GMM-HMMs?

- Standalone training of CD-DNN-HMMs
Standalone Training of CD-DNN-HMMs

- The standalone training strategy can be divided into two parts:
  - Alignments — by CI- (monophone state) DNN-HMMs trained in a standalone fashion
  - Targets — by DNN-HMM based decision tree target clustering
Standalone Training of CI-DNN-HMMs

- The standalone CI-DNN-HMMs are trained with *flat initial alignments* (with averaged CI state duration)
- CI-DNN-HMMs training include:
  - Refine initial alignments in an iterative fashion
  - Train a CI-DNN-HMMs using *discriminative pre-training with realignment* and standard fine-tuning
Initial Alignment Refinement

Flat Initial Alignments

Refined Initial Alignments

Train for 1 epoch

Realign

Train for 1 epoch

Realign
Discriminative Pre-training with Realignment

Refined Initial Alignments

Alignments for Fine-tuning
DNN-HMM based Target Clustering

- Assume the output distribution for each target is Gaussian with common covariance matrix, i.e., \( p(z | C_k) = \mathcal{N}(z; \mu_k, \Sigma) \)
  - the \( k \)th target
  - sigmoidal activation vector from the last hidden layer

- \( \mathcal{N}(z; \mu_k, \Sigma) \) are estimated based on maximum likelihood criterion
  - the features are de-correlated with state-specific rotation
  - the left clustering process is the same as the original approach

- Next, we investigate the link between the Gaussian distributions and the DNN output layer
DNN-HMM based Target Clustering

• From Bayes’ theorem,

\[ p(C_k|z) = \frac{p(z|C_k)P(C_k)}{\sum_{k'} p(z|C_{k'})P(C_{k'})} \]

\[ = \frac{\exp\left\{ \mu_k^T \Sigma^{-1} z - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \ln P(C_k) \right\}}{\sum_{k'} \exp\left\{ \mu_{k'}^T \Sigma^{-1} z - \frac{1}{2} \mu_{k'}^T \Sigma^{-1} \mu_{k'} + \ln P(C_{k'}) \right\}} \]

• According to softmax output activation function,

\[ p(C_k|z) = \frac{\exp\left\{ w_k^T z + b_k \right\}}{\sum_{k'} \exp\left\{ w_{k'}^T z + b_{k'} \right\}} \]
Procedure of Building CD-DNN-HMMs

1. Estimate Distributions for all untied CD states
2. Add untied state to CI-DNN hidden layers to collect statistics
3. Perform DNN-HMM based Target Clustering
4. Add tied CD state targets to CI-DNN hidden layers
5. Fine-tuning
Experiments

- Wall Street Journal training set (SI-284), along with 1994 H1-dev (Dev) and Nov’94 H1-eval (Eval) testing sets were used.
  - utterance level CMN and global CVN
- MPE GMM-HMMs have 5981 tied triphone states and 12 Gaussian components per state
  - MPE GMM-HMMs were with $((13\text{PLP})_{D,A,T,Z})_{\text{HLDA}}$
- Every DNN had 5 hidden layers with 1000 nodes per layer
  - All DNN-HMMs were with $9 \times (13\text{PLP})_{D,A,Z}$
  - sigmoid/softmax hidden/output activation function
  - cross-entropy training criterion
- 65k dictionary and trigram language model
**CI-DNN-HMM Results**

**Table**: Baseline CI-DNN-HMM Results ($351 \times 1000^5 \times 138$).

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>DNN Alignments</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>G2</td>
<td>MPE GMM-HMMs</td>
<td>—</td>
<td>8.0</td>
</tr>
<tr>
<td>I1</td>
<td>CI-DNN-HMMs</td>
<td>G2</td>
<td>10.5</td>
</tr>
</tbody>
</table>

**Table**: Different CI-DNN-HMMs trained in a standalone fashion.

<table>
<thead>
<tr>
<th>ID</th>
<th>Training Route</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>I3</td>
<td>Realigned</td>
<td>12.2</td>
</tr>
<tr>
<td>I4</td>
<td>Realigned+Conventional</td>
<td>11.7</td>
</tr>
<tr>
<td>I5</td>
<td>Conventional</td>
<td>12.2</td>
</tr>
<tr>
<td>I6</td>
<td>Conventional+Conventional</td>
<td>12.0</td>
</tr>
</tbody>
</table>
CD-DNN-HMM Results

- Baseline CD-DNN-HMMs (D1) were trained with G2 alignments. The WER on Dev and Eval are 6.7 and 8.0, respectively.
- CD-DNN-HMMs with different clustered targets were listed in the table. The hidden layer and alignments were from I4.

Table: CD-DNN-HMM based state tying results ($351 \times 1000^5 \times 6000$).

<table>
<thead>
<tr>
<th>ID</th>
<th>Clustering</th>
<th>BP Layers</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>G3</td>
<td>GMM-HMM</td>
<td>Final Layer</td>
<td>7.6</td>
</tr>
<tr>
<td>G4</td>
<td></td>
<td>All Layers</td>
<td>6.8</td>
</tr>
<tr>
<td>D2</td>
<td>DNN-HMM</td>
<td>Final Layer</td>
<td>7.7</td>
</tr>
<tr>
<td>D3</td>
<td></td>
<td>All Layers</td>
<td>6.8</td>
</tr>
</tbody>
</table>

- The CD-DNN-HMMs (D3) trained without relying on any GMM-HMMs is comparable to baseline D1.
Conclusions

• We accomplish training CD-DNN-HMMs without relying on any pre-existing system
  ◦ train CI-DNN-HMMs by updating the model parameters and the reference labels in an interleaved fashion
  ◦ adapt decision tree tying to the sigmoidal activation vector space of a CI-DNN

• The experiments on WSJ SI-284 have shown
  ◦ the proposed training procedure gives state-of-the-art performance
  ◦ the methods are very efficient