# Adaptation of Deep Neural Network Acoustic Models Using Factorised I-Vectors

**P. Karanasou**, Y.Q. Wang, M. Gales, P. Woodland
University of Cambridge

Natural Speech Technology
Edinburgh – Cambridge – Sheffield

UNIVERSITY OF CAMBRIDGE

## Overview

- Adaptation of Deep Neural Networks (DNNs)
- Adaptation using i-vectors
  - Speaker i-vectors as cluster adaptive training (CAT) weights
  - Factorised i-vectors
  - Factorisation with an orthogonality constraint
  - Integrating the i-vectors into a DNN
- Experiments on a noise distorted version of WSJ

## Adaptation of DNNs

- Difficulties
  - Difficult to find structure in weights of DNN and apply transforms as in GMM-HMMs [Seide et al., 2011]
  - Large number of parameters requires large amount of adaptation data, thus overfitting may occur [Liao, 2013]
- Separate the adaptation from the DNN weights
  - Append speaker codes to basic input features [Bridle and Cox, 1990, Abdel-Hamid and Jiang, 2013]
  - Append i-vectors [Saon et al., 2013]

## Speaker i-vectors

- One i-vector per speaker, estimated on all data of the speaker
- Equivalence between clusters in CAT [Gales, 1999], eigenvoices [Kuhn et al., 1998] and i-vector space [Glembek et al., 2011], where models are GMMs rather than HMMs

$$\boldsymbol{\mu}^{(sm)} = \boldsymbol{\mu}_0^{(m)} + \boldsymbol{M}^{(m)}\boldsymbol{\lambda}^{(s)} \quad (1)$$

,where $\boldsymbol{\mu}^{(sm)}$ the speaker-dependent supervector, $\boldsymbol{M}^{(m)}$ a low-rank matrix $D \times P$, $\boldsymbol{\lambda}^{(s)}$ the i-vector of size $P$ of speaker $s$.

- Model-based approach; ML estimation of model parameters and i-vectors using EM algorithm

## Factorised i-vectors

- Acoustic factorisation [Gales, 2001] : one i-vector per speaker and one i-vector per noise condition

$$\begin{aligned}\boldsymbol{\mu}^{(snm)} &= \boldsymbol{\mu}_0^{(m)} + \boldsymbol{M}_{\mathrm{s}}^{(m)}\boldsymbol{\lambda}^{(s)} + \boldsymbol{M}_{\mathrm{n}}^{(m)}\boldsymbol{\lambda}^{(n)} \\ &= \boldsymbol{\mu}_0^{(m)} + \boldsymbol{M}_{\mathrm{sn}}^{(m)}\boldsymbol{\lambda}^{(sn)}\end{aligned} \quad (2)$$

where

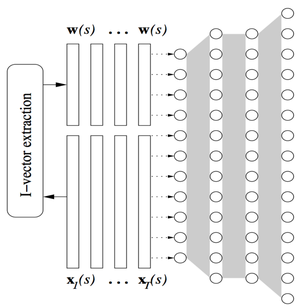
$$\boldsymbol{M}_{\mathrm{sn}}^{(m)}\boldsymbol{\lambda}^{(sn)} = \begin{bmatrix}\boldsymbol{M}_{\mathrm{s}}^{(m)} & \boldsymbol{M}_{\mathrm{n}}^{(m)}\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda}^{(s)} \\ \boldsymbol{\lambda}^{(n)}\end{bmatrix} \quad (3)$$

- ML training to iteratively update factorised i-vectors and model parameters following fCAT training [Latorre et al., 2012]

## Ensuring orthogonality in factorisation


- Represent wide range of speaker/noise pairs by combining i-vectors from particular speaker and particular noise
- Implicit orthogonality : noise balanced data
- Explicit orthogonality constraint [Wang and Gales, 2013]

## Adaptation of a hybrid DNN-HMM system with i-vectors




[Saon et al., 2013]

1. Estimate training i-vectors
2. Train DNN system : Baseline
3. Append the i-vectors and train adapted DNN system
4. Extract test i-vectors and Hybrid decoding on adapted DNN system

## Experimental setup

- WSJ Corpus: noise disrupted - 284 speakers and 7 noise conditions
- SI UBM GMM model to train the i-vectors
- Train a baseline hybrid DNN-HMM model
  - 5 hidden layers with 1000 units, output layer with 6000 targets
  - CD model built with discriminative pre-training and fine-tuning
- Decoding with a 5k bigram LM and test dictionary

## Results of hybrid decoding

Table : Hybrid decoding results (WER %)

| System | wsj1_dt | wsj1_et | wsj0_dt | wsj0_et |
|---|---|---|---|---|
| Hybrid | 12.0 | 9.7 | 8.9 | 6.2 |
| Hybrid-iv | 11.5 | 8.6 | 8.3 | 6.4 |
| Hybrid-fiv | 11.1 | 7.8 | 8.0 | 5.8 |

- *Hybrid-iv*: append speaker i-vectors to basic acoustic features
  *Hybrid-fiv*: append factorised i-vectors
- Factorised approach presents gains over speaker i-vectors approach

## Handling unseen speaker-environment pairs

Table : Decoding results (WER %) on held-out speaker-noise pairs

| System | WER (%) |
|---|---|
| Hybrid | 9.6 |
| Hybrid-fiv-heldout | 8.3 |

- Illustrative experiments on four speaker/noise pairs held-out from the set used to extract i-vectors → Generate factorised i-vectors combining existing speaker and noise i-vectors
- Gains over the Hybrid baseline: Indicates that factorised version can adapt DNN system to unseen speaker/noise pairs

## Conclusions

- Adaptation of a hybrid DNN-HMM system by appending factorised i-vectors as an additional input feature
- Gains over the standard speaker i-vectors approach
- Factorisation allows to handle unseen speaker-noise pairs

## Future work

- Investigate statistical smoothing
- Try transform-based approaches for training
- Apply on other sets of data, for ex. BBC 1-week data
- Apply to other tasks: speaker verification, speaker diarization or asynchronous labelling of speaker and noise conditions