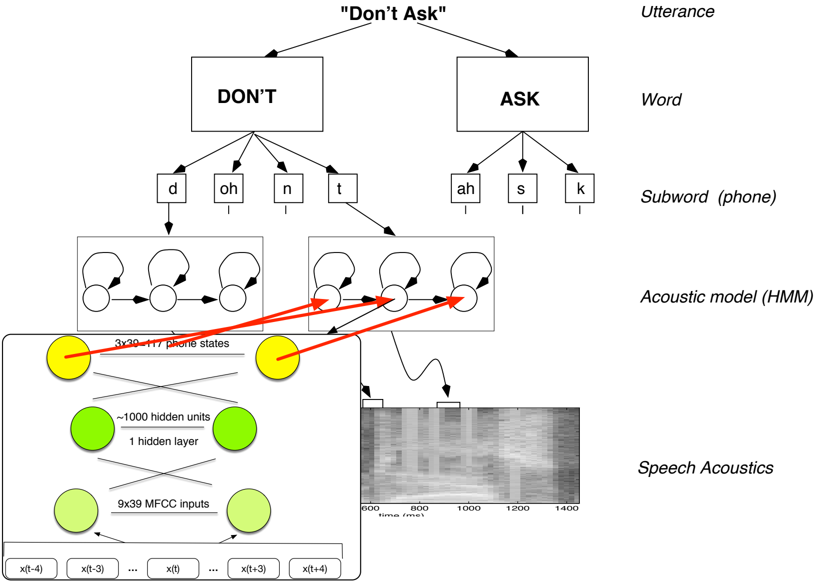
Deep Neural Network Acoustic Models

Steve Renals

Automatic Speech Recognition – ASR Lecture 12
25 February 2016

Recap

Hybrid NN/HMM



HMM/NN vs HMM/GMM

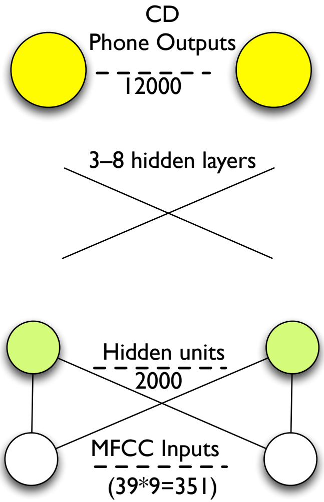
- Advantages of NN:
  - Can easily model **correlated features**
    - Correlated feature vector components (eg spectral features)
    - Input context – multiple frames of data at input
  - **More flexible** than GMMs – not made of (nearly) local components); GMMs inefficient for non-linear class boundaries
  - NNs can **model multiple events** in the input simultaneously – different sets of hidden units modelling each event; GMMs assume each frame generated by a single mixture component.
  - NNs can **learn richer representations** and learn 'higher-level' features (tandem, posteriorgrams, bottleneck features)

HMM/NN vs HMM/GMM

- Advantages of NN:
  - Can easily model **correlated features**
    - Correlated feature vector components (eg spectral features)
    - Input context – multiple frames of data at input
  - **More flexible** than GMMs – not made of (nearly) local components); GMMs inefficient for non-linear class boundaries
  - NNs can **model multiple events** in the input simultaneously – different sets of hidden units modelling each event; GMMs assume each frame generated by a single mixture component.
  - NNs can **learn richer representations** and learn 'higher-level' features (tandem, posteriorgrams, bottleneck features)
- Disadvantages of NN:
  - Until $\sim$ 2012:
    - Context-independent (monophone) models, weak speaker adaptation algorithms
    - NN systems less complex than GMMs (fewer parameters): RNN – $<$ 100k parameters, MLP – $\sim$ 1M parameters
  - Computationally expensive - more difficult to parallelise training than GMM systems

Deep Neural Network Acoustic Models

Deep neural networks (DNNs) — Hybrid system



DNNs — what's new?

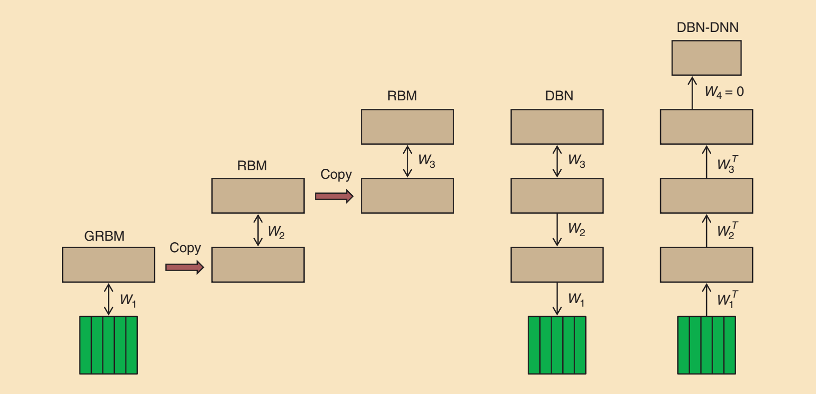
- Training multi-hidden layers directly with gradient descent is difficult — sensitive to initialisation, gradients can be very small after propagating back through several layers.

  Unsupervised pretraining
  - Train a stacked restricted Boltzmann machine generative model (unsupervised), then finetune with backprop
  - Contrastive divergence training

  Layer-by-layer training
  - Successively train deeper networks, each time replacing output layer with hidden layer and new output layer
- Many hidden layers
  - GPUs provide the computational power
- Wide output layer (context dependent phone classes)
  - GPUs provide the computational power

(Hinton et al 2012)

Unsupervised pretraining



Hinton et al (2012)

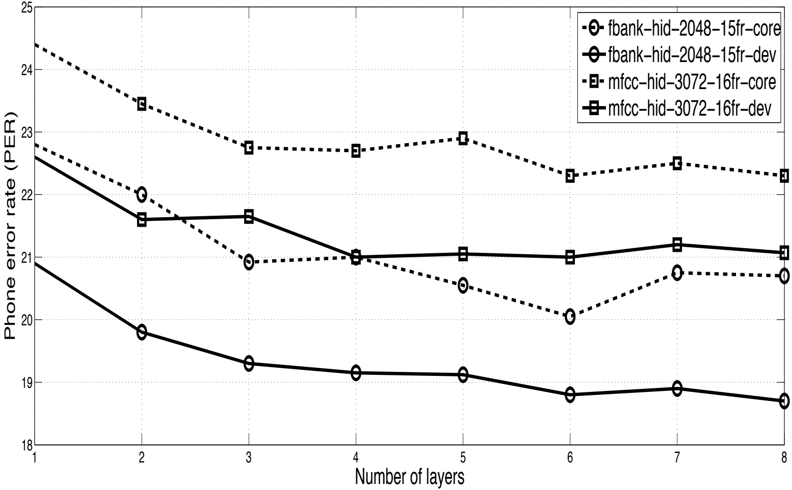
Example: hybrid HMM/DNN phone recognition (TIMIT)

- Train a 'baseline' three state monophone HMM/GMM system (61 phones, 3 state HMMs) and Viterbi align to provide DNN training targets (time state alignment)
- The HMM/DNN system uses the same set of states as the HMM/GMM system — DNN has 183 (61*3) outputs
- Hidden layers — many experiments, exact sizes not highly critical
  - 3–8 hidden layers
  - 1024–3072 units per hidden layer
- Multiple hidden layers always work better than one hidden layer
- Pretraining always results in lower error rates
- Best systems have lower phone error rate than best HMM/GMM systems (using state-of-the-art techniques such as discriminative training, speaker adaptive training)

Acoustic features for NN acoustic models

- GMMs: filter bank features (spectral domain) not used as they are strongly correlated with each other – would either require
  - full covariance matrix Gaussians
  - many diagonal covariance Gaussians
- DNNs do not require the components of the feature vector to be uncorrelated
  - Can directly use multiple frames of input context (this has been done in NN/HMM systems since 1990!)
  - Can potentially use feature vectors with correlated components (e.g. filter banks)
- Experiments indicate that filter bank features result in greater accuracy than MFCCs

TIMIT phone error rates: effect of depth and feature type

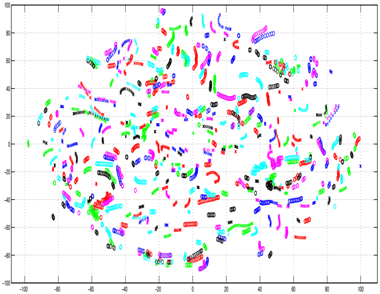


(Mohamed et al (2012))

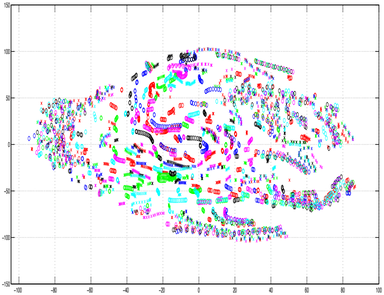
Visualising neural networks

- How to visualise NN layers? "t-SNE" (stochastic neighbour embedding using t-distribution) projects high dimension vectors (e.g. the values of all the units in a layer) into 2 dimensions
- t-SNE projection aims to keep points that are close in high dimensions close in 2 dimensions by comparing distributions over pairwise distances between the high dimensional and 2 dimensional spaces – the optimisation is over the positions of points in the 2-d space

Feature vector (input layer): t-SNE visualisation



MFCC



FBANK

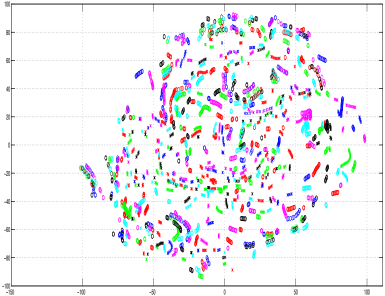
(Mohamed et al (2012))

Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

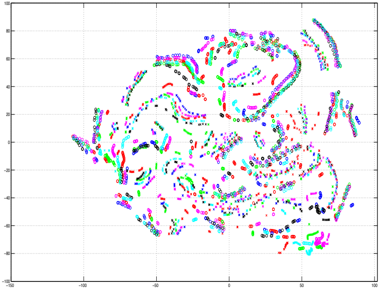
MFCCs are more scattered than FBANK

FBANK has more local structure than MFCCs

First hidden layer: t-SNE visualisation



MFCC



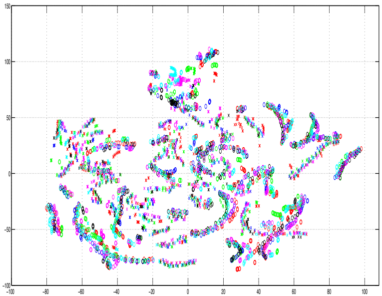
FBANK

(Mohamed et al (2012))

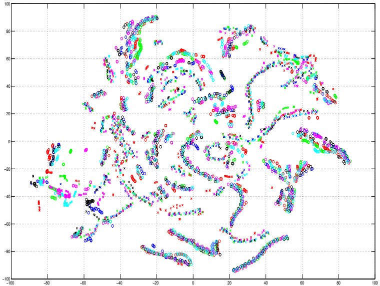
Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

Hidden layer vectors start to align more between speakers for FBANK

Eighth hidden layer: t-SNE visualisation



MFCC



FBANK

(Mohamed et al (2012))

Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

In the final hidden layer, the hidden layer outputs for the same phone are well-aligned across speakers for both MFCC and FBANK – but stronger for FBANK

Visualising neural networks

- How to visualise NN layers? "t-SNE" (stochastic neighbour embedding using t-distribution) projects high dimension vectors (e.g. the values of all the units in a layer) into 2 dimensions
- t-SNE projection aims to keep points that are close in high dimensions close in 2 dimensions by comparing distributions over pairwise distances between the high dimensional and 2 dimensional spaces – the optimisation is over the positions of points in the 2-d space

Are the differences due to FBANK being higher dimension ($41 \times 3 = 123$) than MFCC ($13 \times 3 = 39$)?

- NO!
- Using higher dimension MFCCs, or just adding noise to MFCCs results in higher error rate
- Why? – In FBANK the useful information is distributed over all the features; in MFCC it is concentrated in the first few.

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
- Pretraining not necessary on larger tasks (empirical result)

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.

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

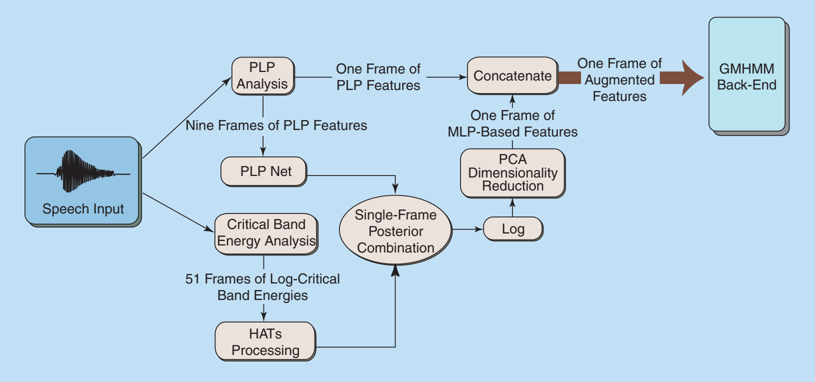
(Hinton et al (2012))

Neural Network Features

Tandem features (posteriorgrams)

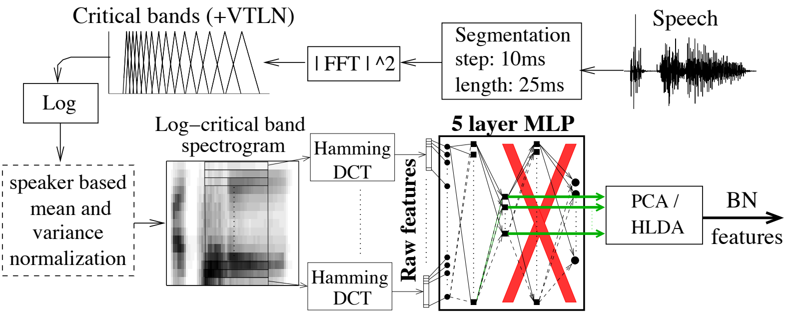
- Use NN probability estimates as an additional input *feature stream* in an HMM/GMM system —- (*Tandem* features (i.e. NN + acoustics), posteriorgrams)
- Advantages of tandem features
  - can be estimated using a large amount of temporal context (eg up to ± 25 frames)
  - encode phone discrimination information
  - only weakly correlated with PLP or MFCC features
- Tandem features: reduce dimensionality of NN outputs using PCA, then concatenate with acoustic features (e.g. MFCCs)
  - PCA also decorrelates feature vector components – important for GMM-based systems

Tandem features



Morgan et al (2005)

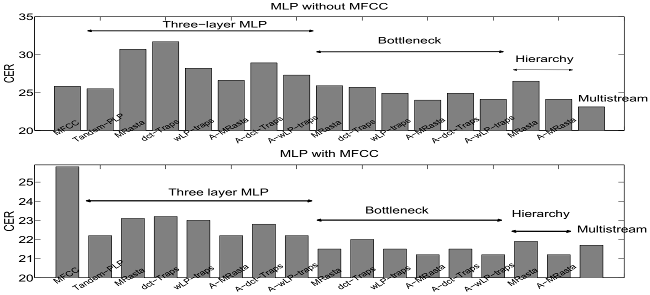
Bottleneck features



Grezl and Fousek (2008)

- Use a “bottleneck” hidden layer to provide features for a HMM/GMM system
- Decorrelate the hidden layer using PCA (or similar)

Experimental comparison of tandem and bottleneck features

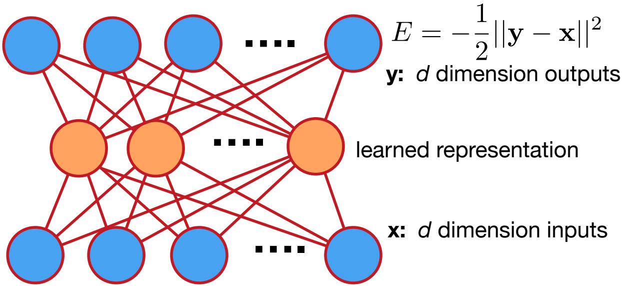


(Valente et al (2011))

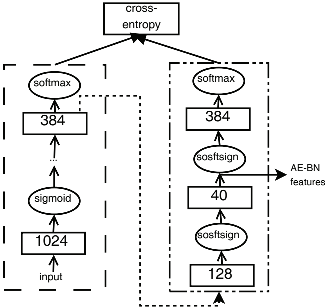
- Results on a Madarin broadcast news transcription task, using an HMM/GMM system
- Explores many different acoustic features for the NN
- Posteriorgram/bottleneck features alone (top)
- Concatenating NN features with MFCCs (bottom)

Autoencoders

- An autoencoder is a neural network trained to map its input into a distributed representation from which the input can be reconstructed
- Example: single hidden layer network, with an output the same dimension as the input, trained to reproduce the input using squared error cost function



Autoencoder Bottlneck (AE-BN) Features



(1) Deep Belief Network

(2) Auto-encoder

- First train a "usual" DNN classifying acoustic input into 384 HMM states
- Then train an autoencoder that maps the predicted output vector to the target output vector
- Use the bottleneck hidden layer in the autoencoder as features for a GMM/HMM system

Results using Autoencoder Bottlneck (AE-BN) Features

[TABLE 4] WER IN % ON ENGLISH BROADCAST NEWS.

	50 H		**430 H**	
LVCSR STAGE	**GMM-HMM BASELINE**	**AE-BN**	**GMM/HMM BASELINE**	**AE-BN**
FSA	24.8	20.6	20.2	17.6
+fBMMI	20.7	19.0	17.7	16.6
+BMMI	19.6	18.1	16.5	15.8
+MLLR	18.8	**17.5**	16.0	**15.5**
MODEL COMBINATION	16.4		15.0	

Hinton et al (2012)

Summary

- DNN/HMM systems (hybrid systems) give a significant improvement over GMM/HMM systems
- Compared with 1990s NN/HMM systems, DNN/HMM systems
    - model context-dependent tied states with a much wider output layer
    - are deeper – more hidden layers
    - can use correlated features (e.g. FBANK)
- DNN features obtained from output layer (posteriorgram) or hidden layer (bottleneck features) give a significant reduction in WER when appended to acoustic features (e.g. MFCCs)

Reading

- G Hinton et al (Nov 2012). "Deep neural networks for acoustic modeling in speech recognition", *IEEE Signal Processing Magazine*, **29**(6), 82–97. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6296526
- A Mohamed et al (2012). "Unserstanding how deep belief networks perform acoustic modelling", Proc ICASSP-2012. http://www.cs.toronto.edu/~asamir/papers/icassp12_dbn.pdf
- N Morgan et al (Sep 2005). "Pushing the envelope – aside", *IEEE Signal Processing Magazine*, **22**(5), 81–88. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1511826
- F Grezl and P Fousek (2008). "Optimizing bottleneck features for LVCSR", Proc ICASSP–2008. http://noel.feld.cvut.cz/speechlab/publications/068_icassp08.pdf
- F Valente et al (2011). "Analysis and Comparison of Recent MLP Features for LVCSR Systems", Proc Interspeech–2011. https://www.sri.com/sites/default/files/publications/analysis_and_comparison_of_recent_mlp_features_for_lvcsr_systems.pdf