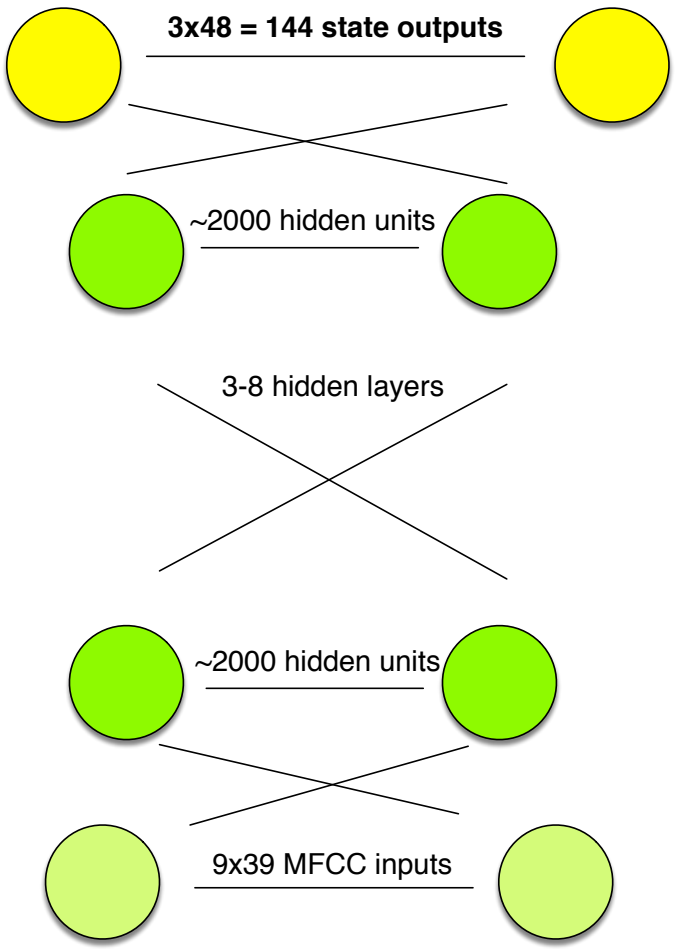
Neural Networks for Acoustic Modelling part 2; Sequence discriminative training

Steve Renals

Automatic Speech Recognition – ASR Lecture 9
16 February 2017

DNN Acoustic Models

Deep neural network for TIMIT



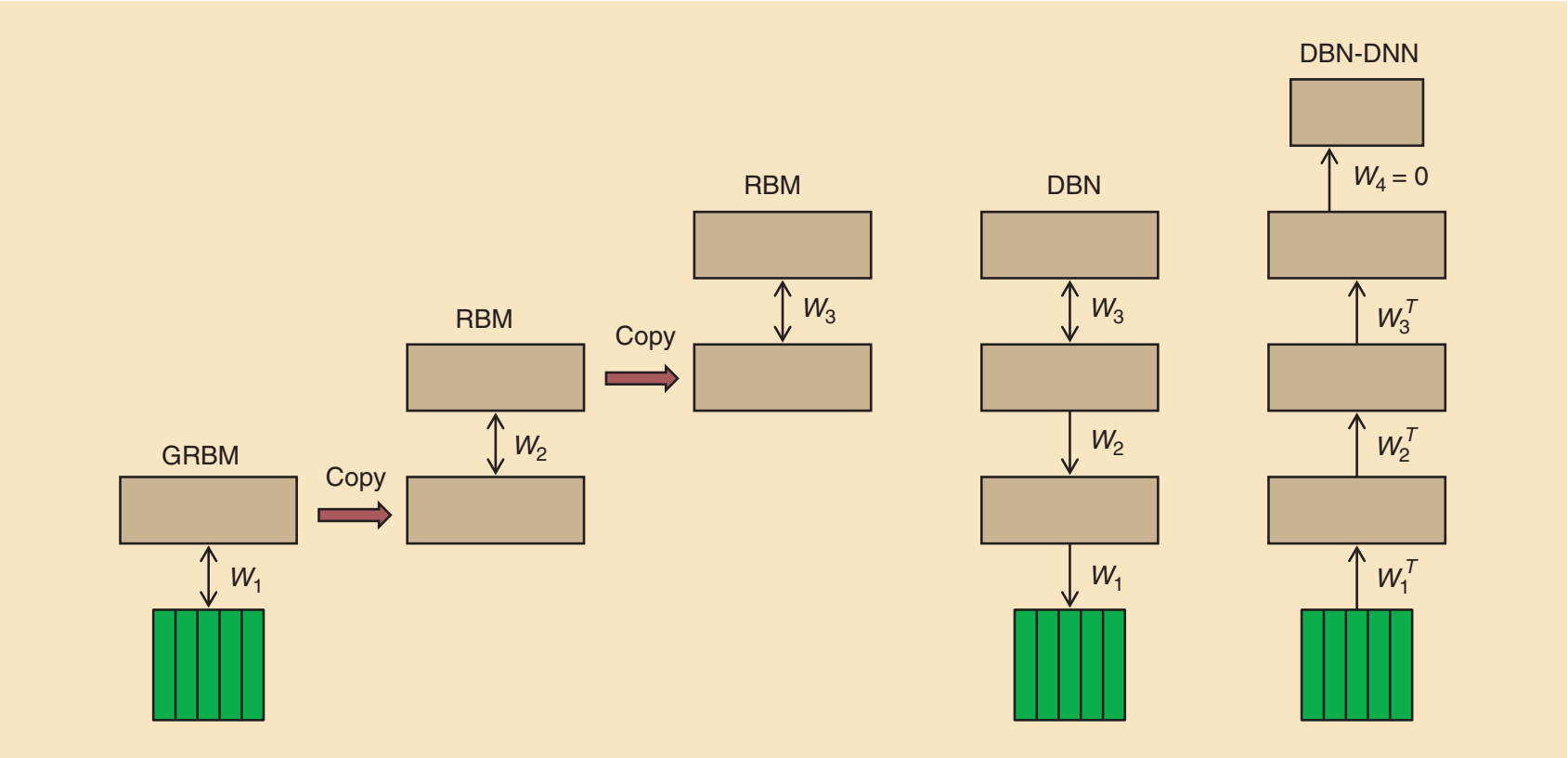
- **Deeper**: Deep neural network architecture – multiple hidden layers
- **Wider**: Use HMM state alignment as outputs rather than hand-labelled phones – 3-state HMMs, so 3×48 states
- Can use *pretraining* to improve training accuracy of models with many hidden layers
- Training many hidden layers is computationally expensive – use GPUs to provide the computational power

Pretraining

- 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, contrastive divergence training), then finetune with backprop
  - Train a stacked autoencoder, then finetune with backprop

  Layer-by-layer training
  - Successively train deeper networks, each time replacing output layer with hidden layer and new output layer

Unsupervised pretraining



Hinton et al (2012)

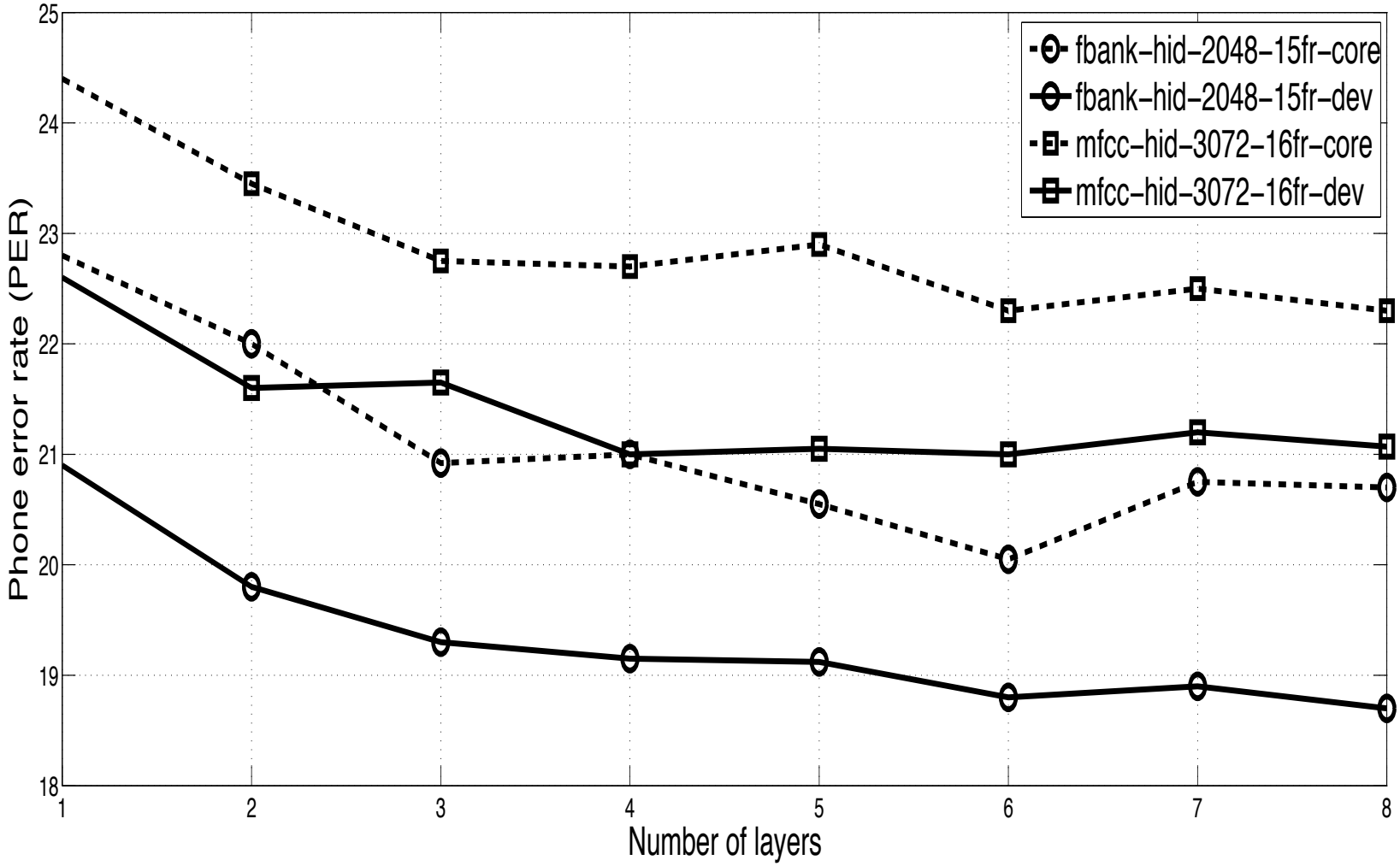
Hybrid HMM/DNN phone recognition (TIMIT)

- 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, and is crucial to make them work)
  - 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

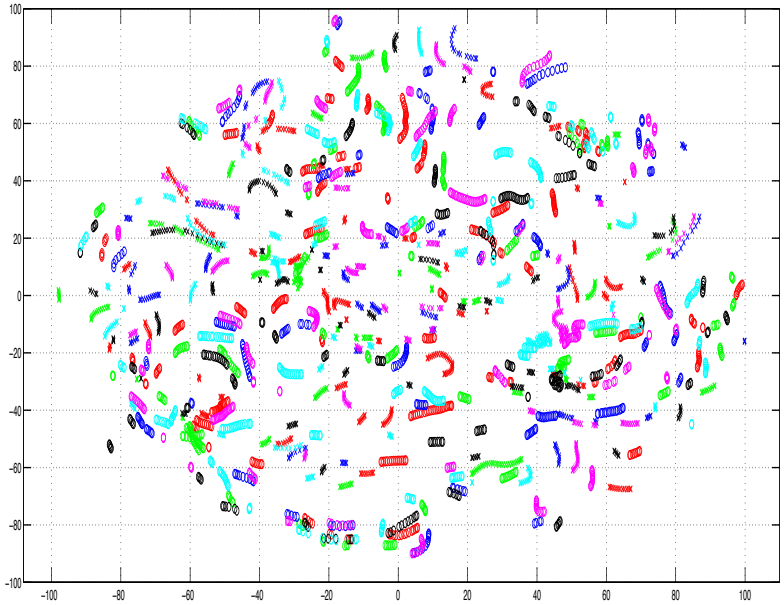


(Mohamed et al (2012))

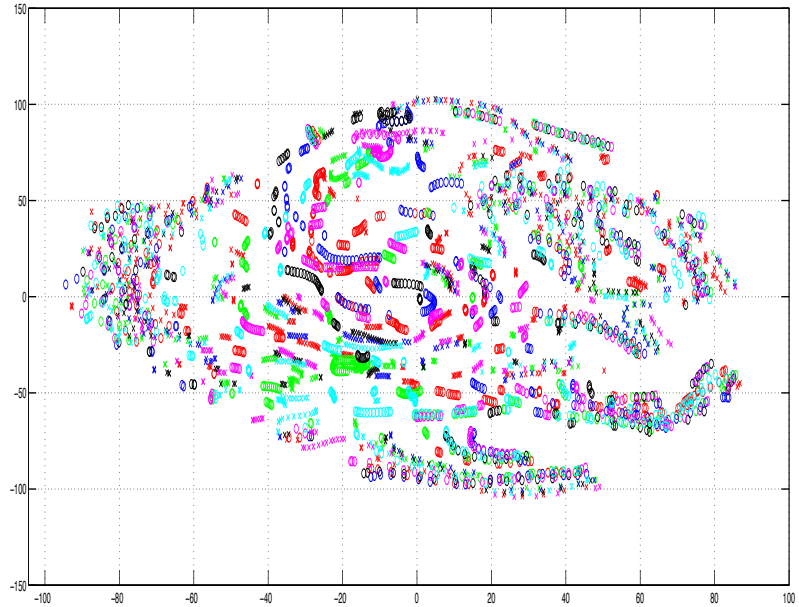
Visualising neural networks

- 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

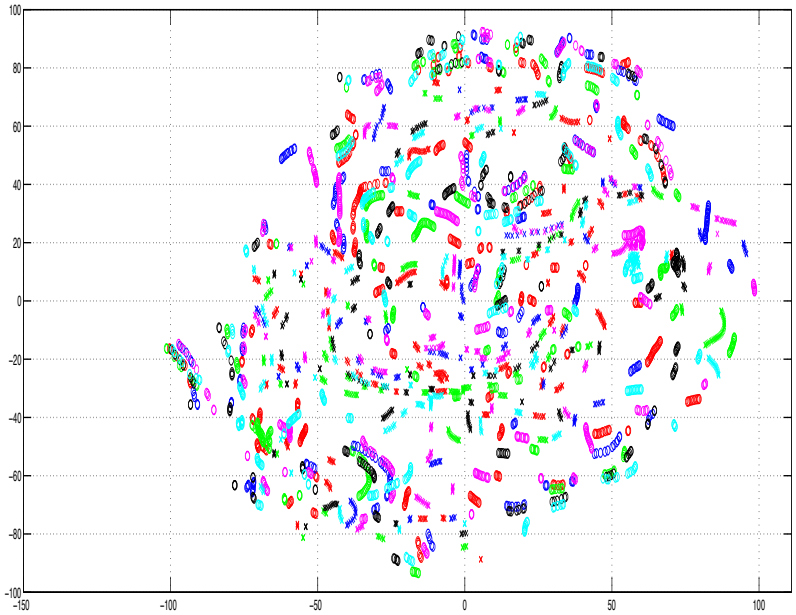
(Mohamed et al (2012))

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

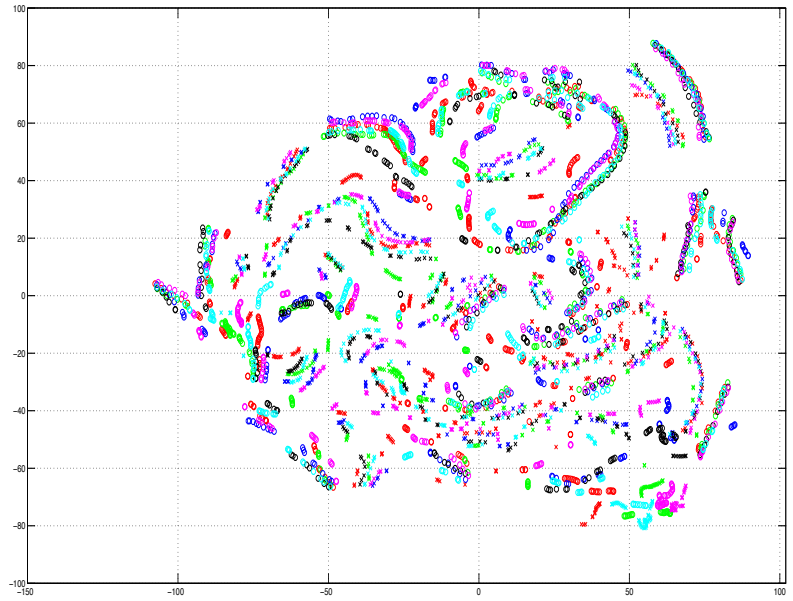
MFCCs are more scattered than FBANK

FBANK has more local structure than MFCCs

First hidden layer: t-SNE visualisation



MFCC



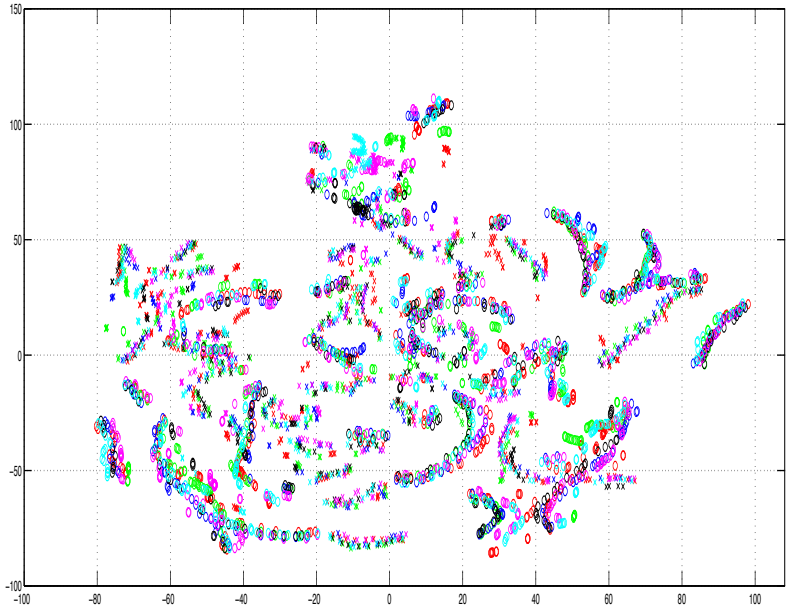
FBANK

(Mohamed et al (2012))

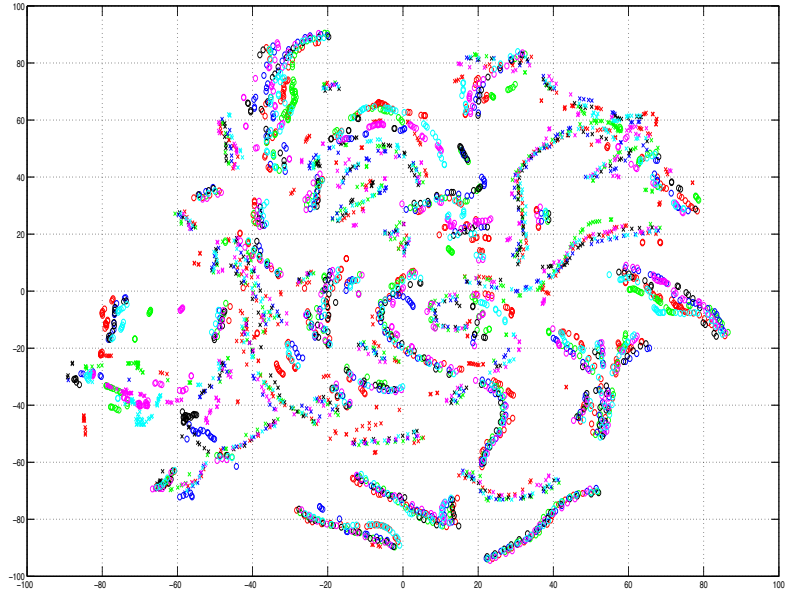
Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

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

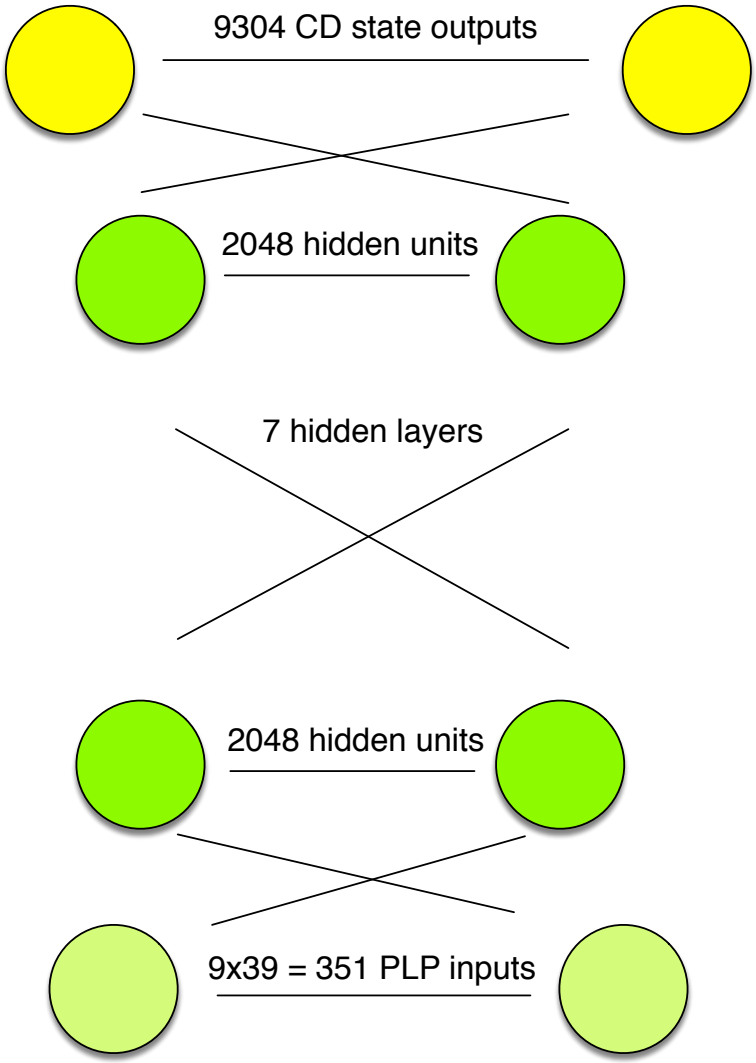
Visualising neural networks

- 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 noisy dimmensions 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.

DNN acoustic model for Switchboard



(Hinton et al (2012))

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))

Sequence Discriminative Training

Training HMM/GMM acoustic models

- Use forward-backward algorithm to estimate the state occupation probabilities (E-step), which are used to re-estimate the parameters (M-step)
- Maximum likelihood estimation: estimate the parameters so that the model reproduces the training data with the greatest probability (maximum likelihood)
- Discriminative training: directly estimate the parameters so as to make the fewest classification errors (optimize the word error rate)
  - Focus on learning *boundaries* between classes
  - Consider incorrect word sequences as well as correct word sequences
  - This is related to direct optimisation of the posterior probability of the words given the acoustics $P(W \mid \mathbf{X})$

Hybrid HMM/NN acoustic models

- Neural networks are discriminatively trained at the **frame** level
- Consider a context-dependent DNN
  - Output is a softmax over HMM states
  - Training involves increasing the probability of the correct state – and hence decreasing the probabilities of the others, since probabilities sum to 1
  - Frame-level discrimination – the network learns to optimise discrimination at the frame level by choosing the best state at each time frame
- **Sequence discrimination** – train the system to select the best sequence of frames by increasing the probability of the best sequence and decreasing the probability of all competing sequences
- Can train both GMM and DNN based models using sequence discrimination

Hybrid HMM/NN acoustic models

- Neural networks are discriminatively trained at the **frame** level
- Consider a context-dependent DNN
  - Output is a softmax over HMM states
  - Training involves increasing the probability of the correct state – and hence decreasing the probabilities of the others, since probabilities sum to 1
  - Frame-level discrimination – the network learns to optimise discrimination at the frame level by choosing the best state at each time frame
- **Sequence discrimination** – train the system to select the best sequence of frames by increasing the probability of the best sequence and decreasing the probability of all competing sequences
- Can train both **GMM** and DNN based models using sequence discrimination

Recall: Maximum likelihood estimation (MLE)

- Maximum likelihood estimation (MLE) sets the parameters so as to maximize an objective function F_{MLE}:

$$F_{\text{MLE}} = \sum_{u=1}^{U} \log P_\lambda(\mathbf{X}_u \mid M(W_u))$$

for training utterances $\mathbf{X}_1 \ldots \mathbf{X}_U$ where W_u is the word sequence given by the transcription of the uth utterance, $M(W_u)$ is the corresponding HMM, and λ is the set of HMM parameters

Maximum mutual information estimation

- Maximum mutual information estimation (MMIE) aims to directly maximise the posterior probability (sometimes called conditional maximum likelihood). Using the same notation as before, with $P(w)$ representing the language model probability of word sequence w:

$$
\begin{aligned}
F_{\text{MMIE}} &= \sum_{u=1}^{U} \log P_\lambda(M(W_u) \mid \mathbf{X}_u) \\
&= \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}
\end{aligned}
$$

Maximum mutual information estimation

- Maximum mutual information estimation (MMIE) aims to directly maximise the posterior probability (sometimes called conditional maximum likelihood). Using the same notation as before, with $P(w)$ representing the language model probability of word sequence w:

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log P_\lambda(M(W_u) \mid \mathbf{X}_u)$$

$$F_{\text{MLE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}$$

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}$$

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)
- **Denominator**: total likelihood of the data given all possible word sequences – equivalent to summing over all possible word sequences estimated by the full acoustic and language models in recognition. ("free")
  Estimate by generating lattices, and summing over all words in the lattice

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid M(w'))P(w')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)
- **Denominator**: total likelihood of the data given all possible word sequences – equivalent to summing over all possible word sequences estimated by the full acoustic and language models in recognition. ("free")
  Estimate by generating lattices, and summing over all words in the lattice
- The objective function F_{MMIE} is optimised by making the correct word sequence likely (maximise the numerator), and all other word sequences unlikely (minimise the denominator)

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MPE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid M(W))P(W)A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid M(W_{\underline{u}}))P(W_{\underline{u}})A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MPE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid M(W))P(W)A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u
- F_{MPE} is a weighted average over all possible sentences w of the raw phone accuracy
- Although MPE optimizes a phone accuracy level, it does so in the context of a word-level system: it is optimized by finding probable sentences with low phone error rates

Sequence training of hybrid HMM/DNN systems

- It is possible to train HMM/NN systems using a MMI-type objective function
- Forward- and back-propagation equations are structurally similar to forward and backward recursions in HMM training
- Initially train DNN framewise using cross-entropy (CE) error function
  - Use CE-trained model to generate alignments and lattices for sequence training
  - Use CE-trained weights to initialise weights for sequence training
- Train using back-propagation with sequence training objective function (e.g. MMI)

Sequence training results on Switchboard (Kaldi)

Results on Switchboard "Hub 5 '00" test set, trained on 300h training set, comparing maximum likelihood (ML) and discriminative (BMMI) trained GMMs with framewise cross-entropy (CE) and sequence trained (MMI) DNNs. GMM systems use speaker adaptive training (SAT).
All systems had 8859 tied triphone states.
GMMs – 200k Gaussians
DNNs – 6 hidden layers each with 2048 hidden units

	SWB	CHE	Total
GMM ML (+SAT)	21.2	36.4	28.8
GMM BMMI (+SAT)	18.6	33.0	25.8
DNN CE	14.2	25.7	20.0
DNN MMI	12.9	24.6	18.8

Veseley et al, 2013.

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)
- Sequence training: discriminatively optimise GMM or DNN to a sentence (sequence) level criterion rather than a frame level criterion

Next lecture: Speaker adaptation

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/document/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
- HMM discriminative training: Sec 27.3.1 of: S Young (2008), "HMMs and Related Speech Recognition Technologies", in *Springer Handbook of Speech Processing*, Benesty, Sondhi and Huang (eds), chapter 27, 539–557. http://www.inf.ed.ac.uk/teaching/courses/asr/2010-11/restrict/Young.pdf
- NN sequence training: K Vesely et al (2013), "Sequence-discriminative training of deep neural networks", Interspeech-2013, http://homepages.inf.ed.ac.uk/aghoshal/pubs/is13-dnn_seq.pdf