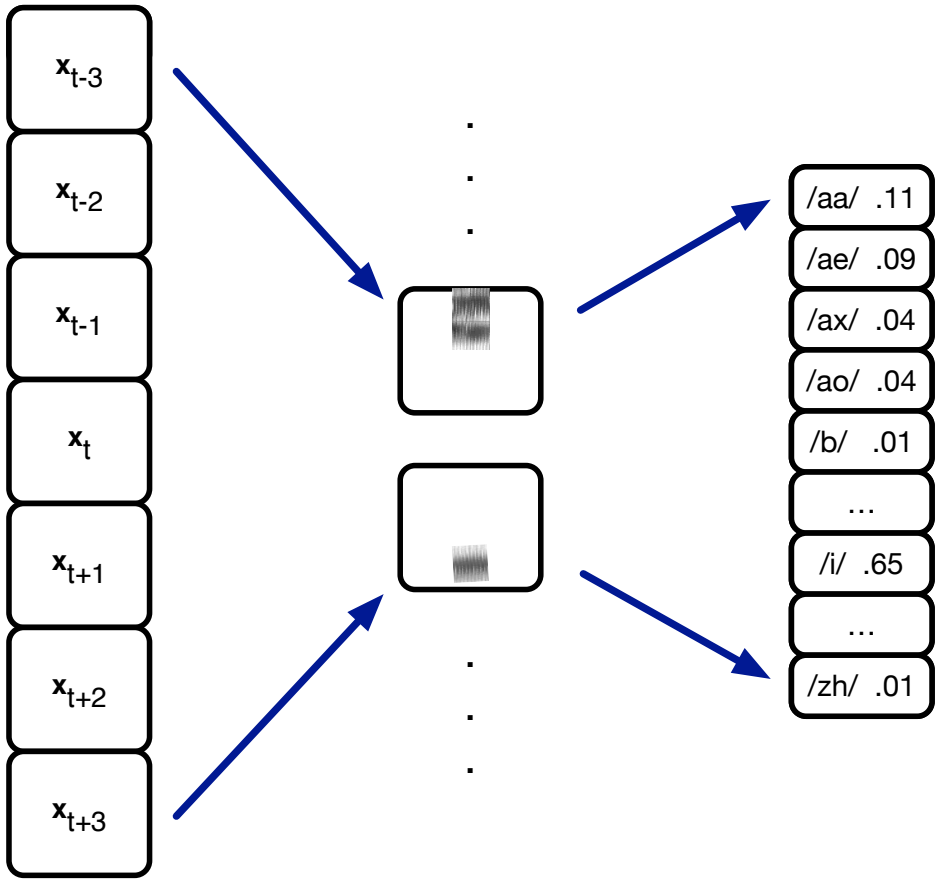
Neural Networks for Acoustic Modelling 2: Hybrid HMM/DNN systems

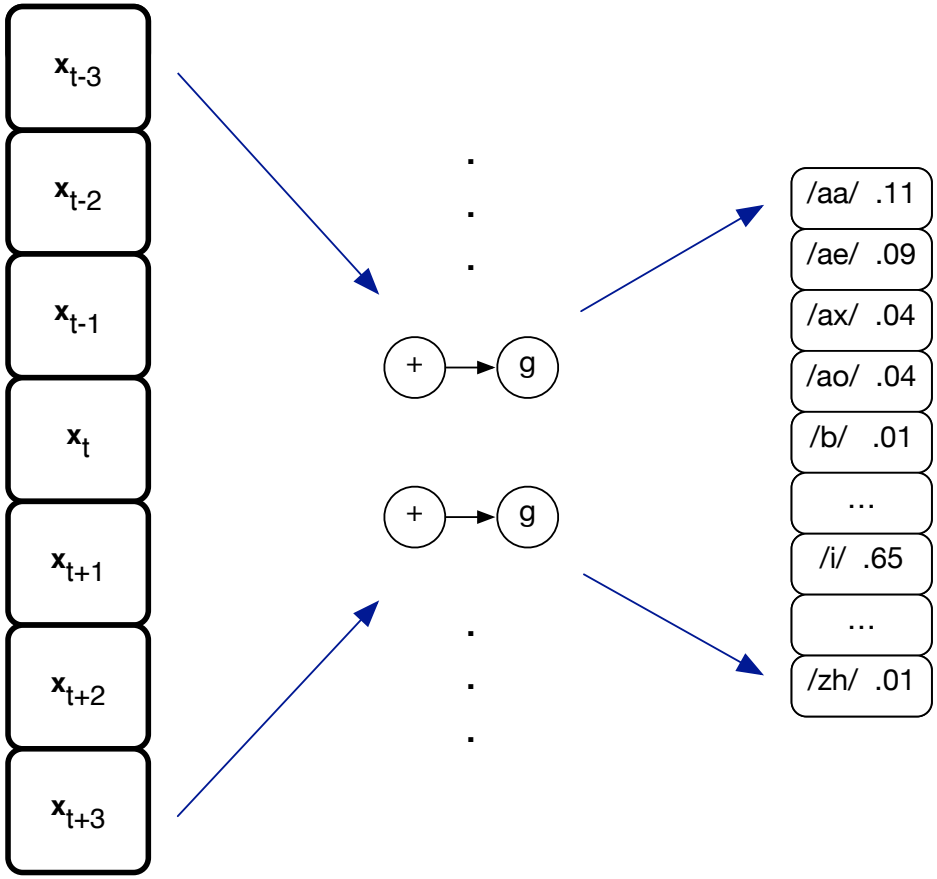
Peter Bell

Automatic Speech Recognition – ASR Lecture 8
6 February 2019

Recap: Hidden units extracting features



Recap: Hidden Units

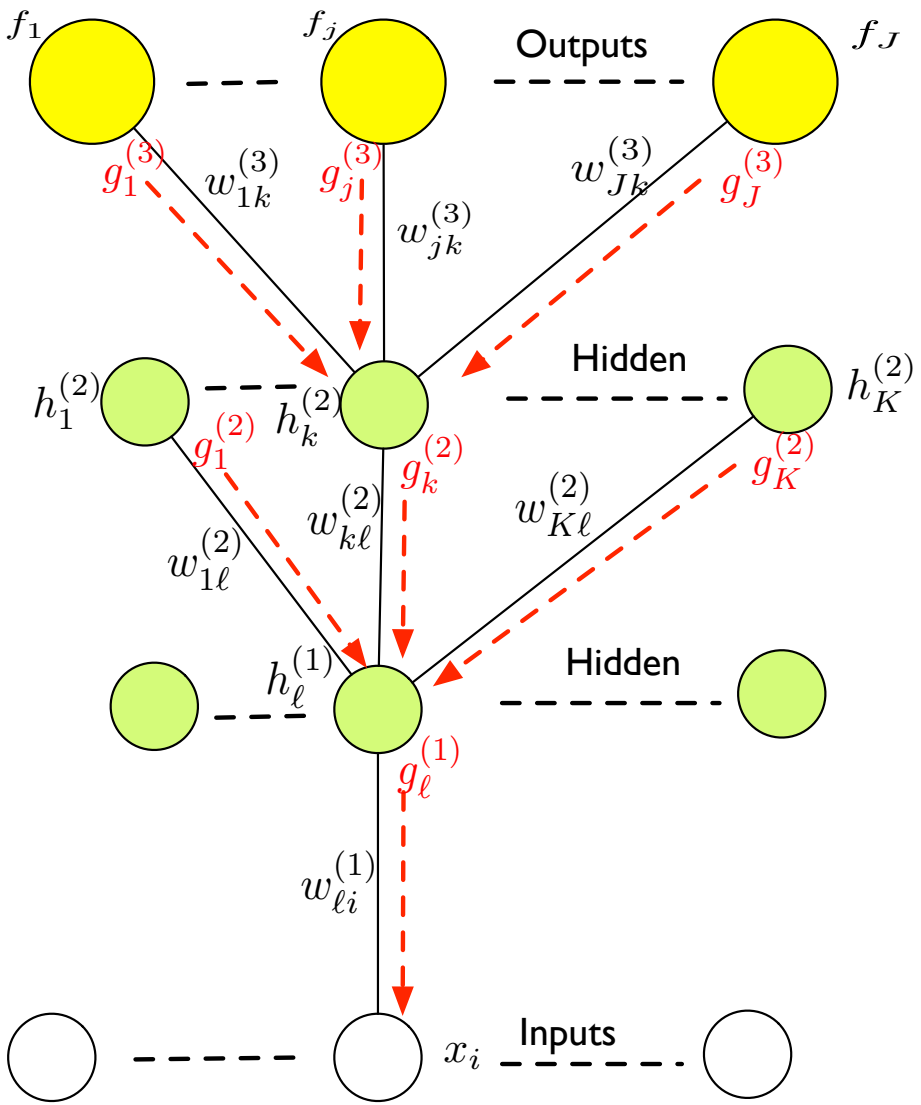


$$h_k = \text{relu}\left(\sum_{d=1}^{D} v_{kd} x_d + b_k\right) \qquad f_j = \text{softmax}\left(\sum_{k=1}^{K} w_{jk} h_k + b_j\right)$$

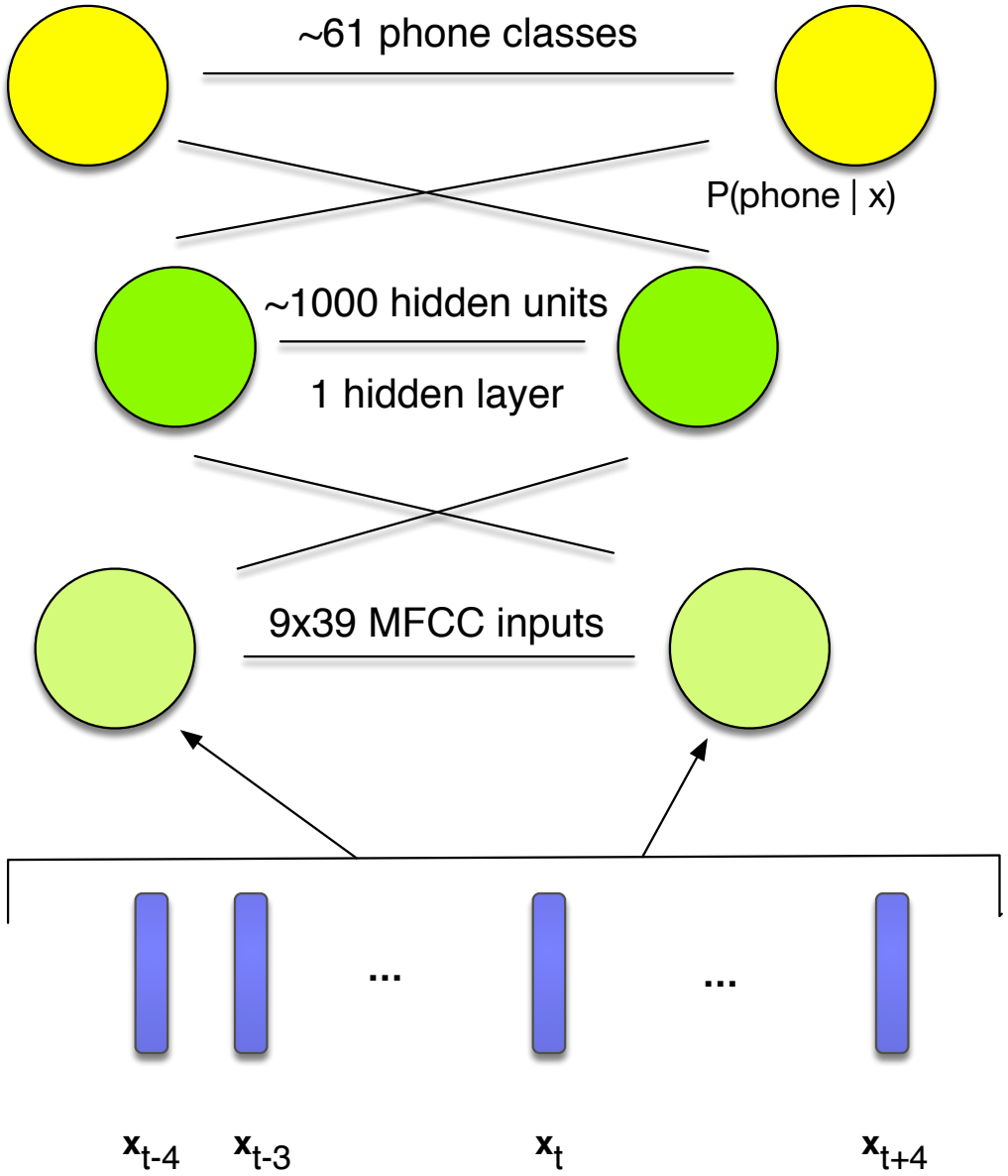
Training deep networks: Backprop and gradient descent

- Hidden units make training the weights more complicated, since each hidden units affects the error function indirectly via all the output units
- The credit assignment problem: what is the "error" of a hidden unit? how important is input-hidden weight v_{kd} to output unit j?
- Solution: *back-propagate* the gradients through the network – the gradient for a hidden unit output with respect to the error can be computed as the weighted sum of the deltas of the connected output units. (Propagate the g values backwards through the network)
- The *back-propagation of error* (*backprop*) algorithm thus provides way to propagate the error graidents through a deep network to allow gradient descent training to be performed

Training DNNs using backprop



Simple neural network for phone classification



Neural networks for phone classification

- Phone recognition task – e.g. TIMIT corpus
  - 630 speakers (462 train, 168 test) each reading 10 sentences (usually use 8 sentences per speaker, since 2 sentences are the same for all speakers)
  - Speech is labelled by hand at the phone level (time-aligned)
  - 61-phone set, often reduced to 48/39 phones
- Phone recognition tasks
  - Frame classification – classify each frame of data
  - Phone classification – classify each segment of data (segmentation into unlabelled phones is given)
  - Phone recognition – segment the data and label each segment (the usual speech recognition task)
- Frame classification – straightforward with a neural network
  - train using labelled frames
  - test a frame at a time, assigning the label to the output with the highest score

Neural networks for phone recognition

- Train a neural network to associate a phone-state label with a frame of acoustic data (+ context)
- Can interpret the output of the network as P(phone-state | acoustic-frame)
- Hybrid NN/HMM systems: in an HMM, replace the GMMs used to estimate output pdfs with the outputs of neural networks
- One-state per phone HMM system:
  - Train an NN as a phone-state classifier (= phone-state probability estimator)
  - Use NN to obtain output probabilities in Viterbi algorithm to find most probable sequence of phones (words)

Neural networks and posterior probabilities

Posterior probability estimation

- Consider a neural network trained as a classifier – each output corresponds to a class.
- When applying a trained network to test data, it can be shown that the value of output corresponding to class j given an input $\mathbf{x}_t$, is an estimate of the posterior probability $P(q_t = j|\mathbf{x}_t)$. (This is because we have softmax outputs and use a cross-entropy loss function)
- Using Bayes Rule we can relate the posterior $P(q_t = j|\mathbf{x}_t)$ to the likelihood $p(\mathbf{x}_t|q_t = j)$ used as an output probability in an HMM:

$$P(q_t|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|q_t = j)P(q_t = j)}{p(\mathbf{x}_t)}$$

Scaled likelihoods

- If we would like to use NN outputs as output probabilities in an HMM, then we would like probabilities (or densities) of the form $p(\mathbf{x}|q)$ – likelihoods.
  We can write *scaled likelihoods* as:

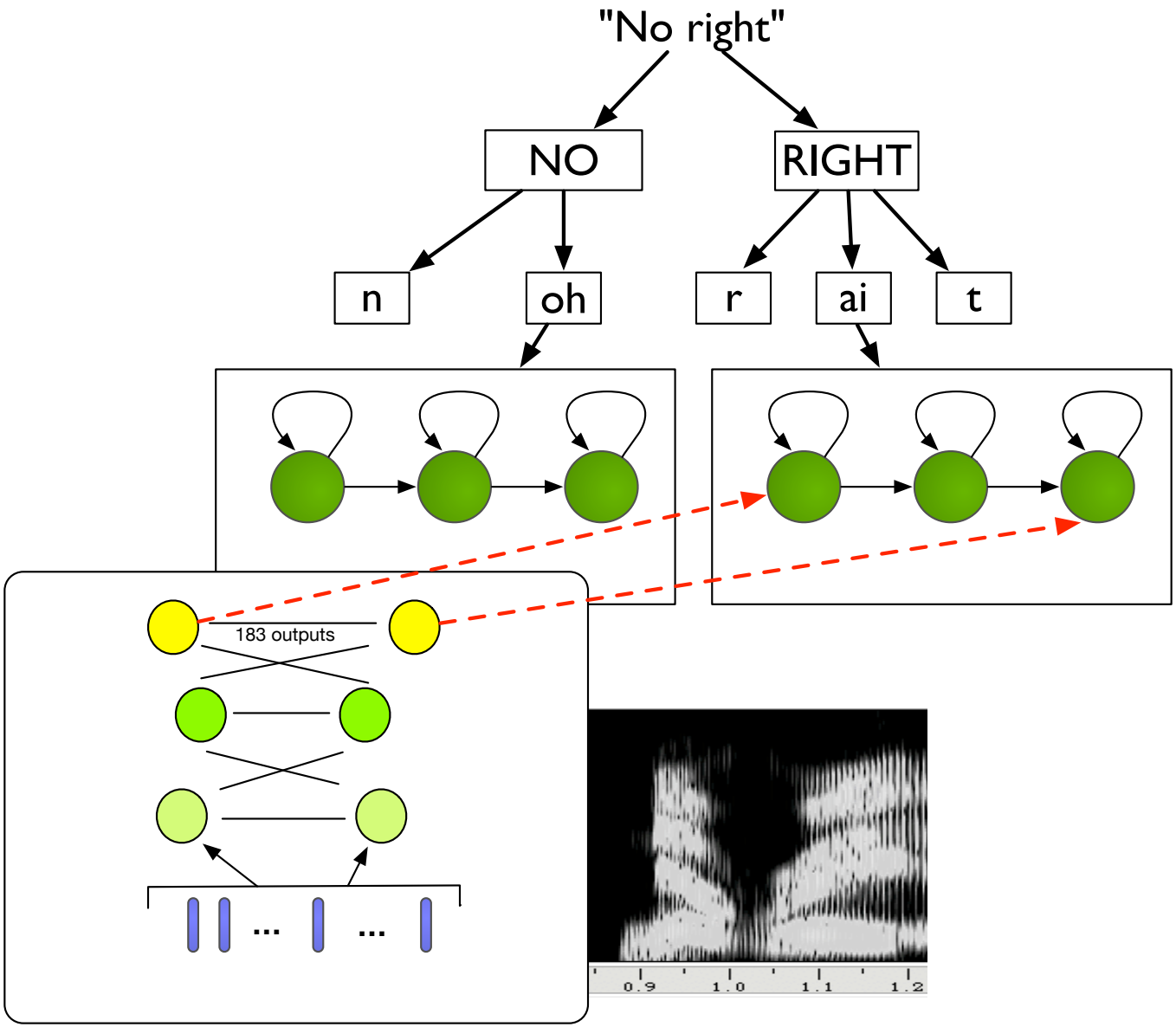
$$\frac{P(q_t = j|\mathbf{x}_t)}{p(q_t = j)} = \frac{p(\mathbf{x}_t|q_t = j)}{p(\mathbf{x}_t)}$$

- Scaled likelihoods can be obtained by "dividing by the priors" – divide each network output $P(q_t = j|\mathbf{x}_t)$ by $P(q_t)$, the relative frequency of class j in the training data
- Using $p(\mathbf{x}_t|q_t = j)/p(\mathbf{x}_t)$ rather than $p(\mathbf{x}_t|q_t = j)$ is OK since $p(\mathbf{x}_t)$ does not depend on the class j
- Use the scaled likelihoods obtained from a neural network in place of the usual likelihoods obtained from a GMM

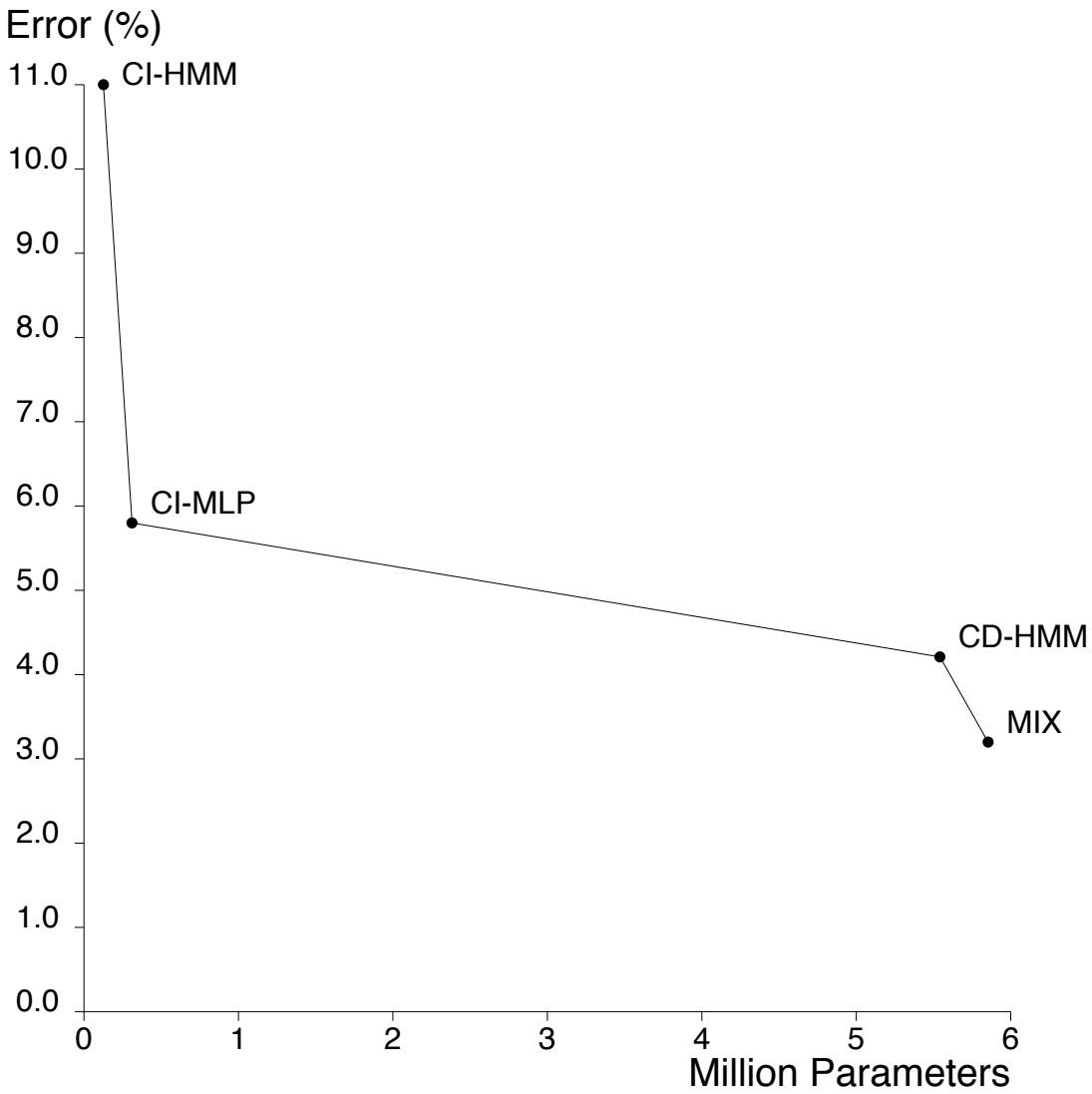
Hybrid NN/HMM

- Generally, if we have a J-state HMM system, then we train a J-output NN to estimate the scaled likelihoods used in a hybrid system.
- For continuous speech recognition we can use:
  - 1 state per phone (61 NN outputs, if we have 61 phone classes)
  - 3 state context-independent (CI) models ($61 \times 3 = 183$ NN outputs)
  - State-clustered context-dependent (CD) models, with one NN output per tied state (this can lead to networks with many outputs!)
- Scaled likelihood and dividing by the priors
  - Computing the scaled likelihoods can be interpreted as factoring out the prior estimates for each phone based on the acoustic training data. The HMM can then integrate better prior estimates based on the language model and lexicon.

Hybrid NN/HMM

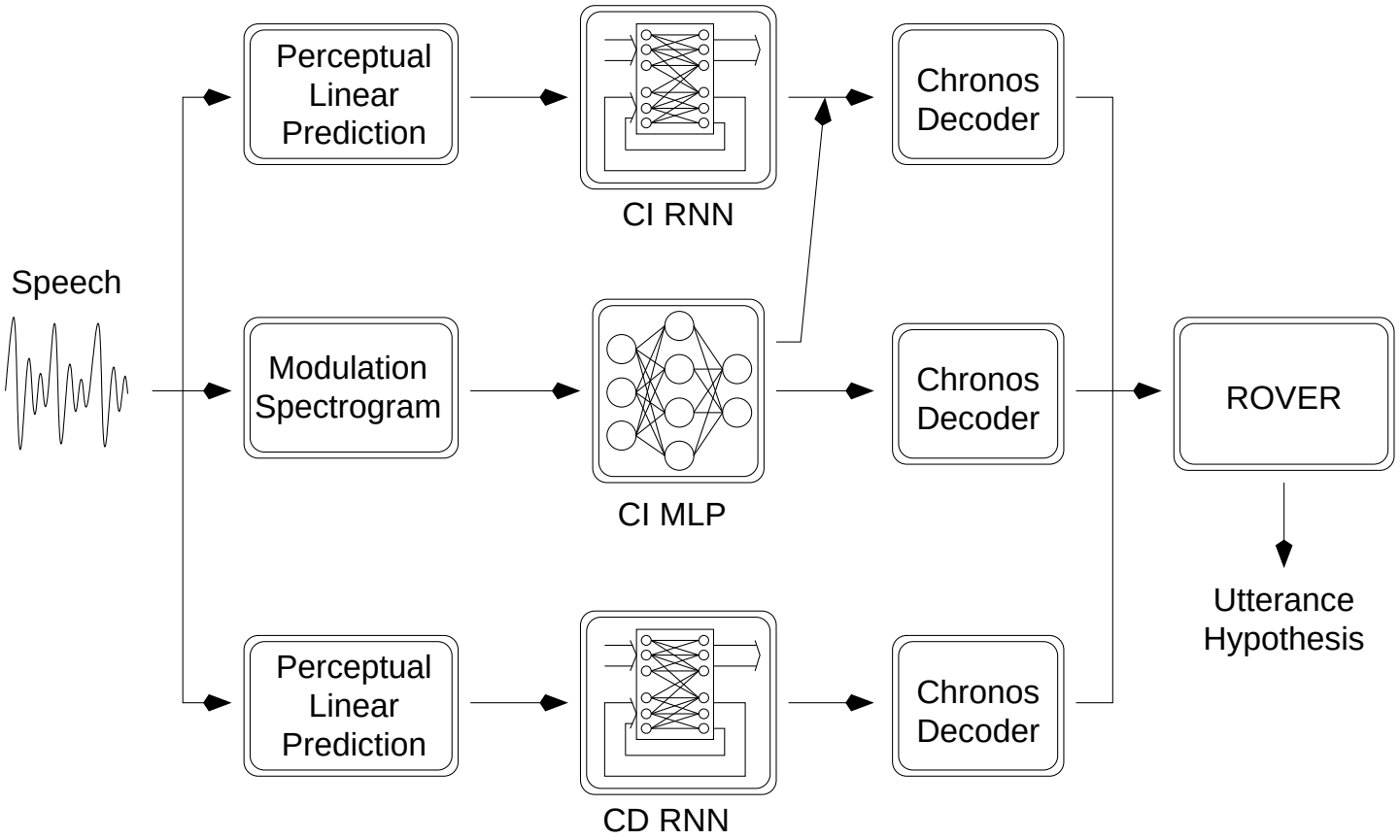


Monophone HMM/NN hybrid system (1993)



Renals, Morgan, Cohen & Franco, ICASSP 1992

Monophone HMM/NN hybrid system (1998)



- Broadcast news transcription (1998) – 20.8% WER
- (best GMM-based system, 13.5%)
- Cook et al, DARPA, 1999

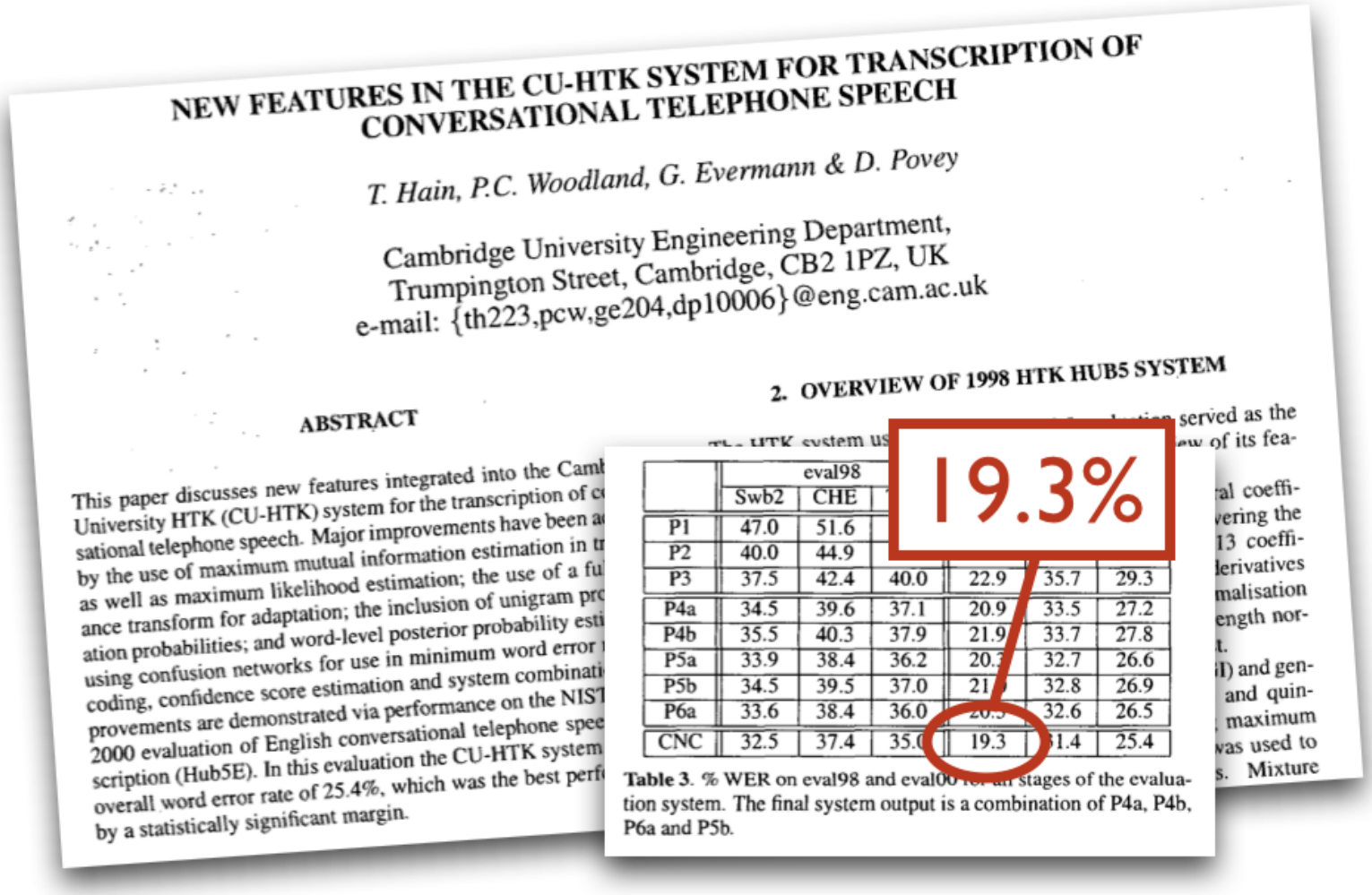
HMM/NN vs HMM/GMM

- 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 NNs in the 1990s:
  - 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

State of the art in the year 2000



NEW FEATURES IN THE CU-HTK SYSTEM FOR TRANSCRIPTION OF CONVERSATIONAL TELEPHONE SPEECH

T. Hain, P.C. Woodland, G. Evermann & D. Povey

Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, UK
e-mail: {th223,pcw,ge204,dp10006}@eng.cam.ac.uk

ABSTRACT

This paper discusses new features integrated into the Cambridge University HTK (CU-HTK) system for the transcription of conversational telephone speech. Major improvements have been achieved by the use of maximum mutual information estimation in training as well as maximum likelihood estimation; the use of a full matrix transform for adaptation; the inclusion of unigram probabilities; and word-level posterior probability estimation using confusion networks for use in minimum word error decoding, confidence score estimation and system combination. Improvements are demonstrated via performance on the NIST 2000 evaluation of English conversational telephone speech transcription (Hub5E). In this evaluation the CU-HTK system achieved an overall word error rate of 25.4%, which was the best performance by a statistically significant margin.

2. OVERVIEW OF 1998 HTK HUB5 SYSTEM

	eval98					
	Swb2	CHE				
P1	47.0	51.6				
P2	40.0	44.9				
P3	37.5	42.4	40.0	22.9	35.7	29.3
P4a	34.5	39.6	37.1	20.9	33.5	27.2
P4b	35.5	40.3	37.9	21.9	33.7	27.8
P5a	33.9	38.4	36.2	20.3	32.7	26.6
P5b	34.5	39.5	37.0	21.[illegible]	32.8	26.9
P6a	33.6	38.4	36.0	20.[illegible]	32.6	26.5
CNC	32.5	37.4	35.[illegible]	19.3	[illegible].4	25.4

19.3%

Table 3. % WER on eval98 and eval00 for all stages of the evaluation system. The final system output is a combination of P4a, P4b, P6a and P5b.

Features of the Cambridge system

	CU-HTK 2000
Base model	HMM-GMM
Acoustic context	Δ, $\Delta\Delta$ features, HLDA projection
Phonetic context	Tied state triphones & quinphones
Speaker adaptation	Gender-dependent models, VTLN, MLLR
Training criterion	ML + MMI sequence training
System architecture	6-pass system
Other features	Multi-system combination
Hub 2000 WER	**19.3%**

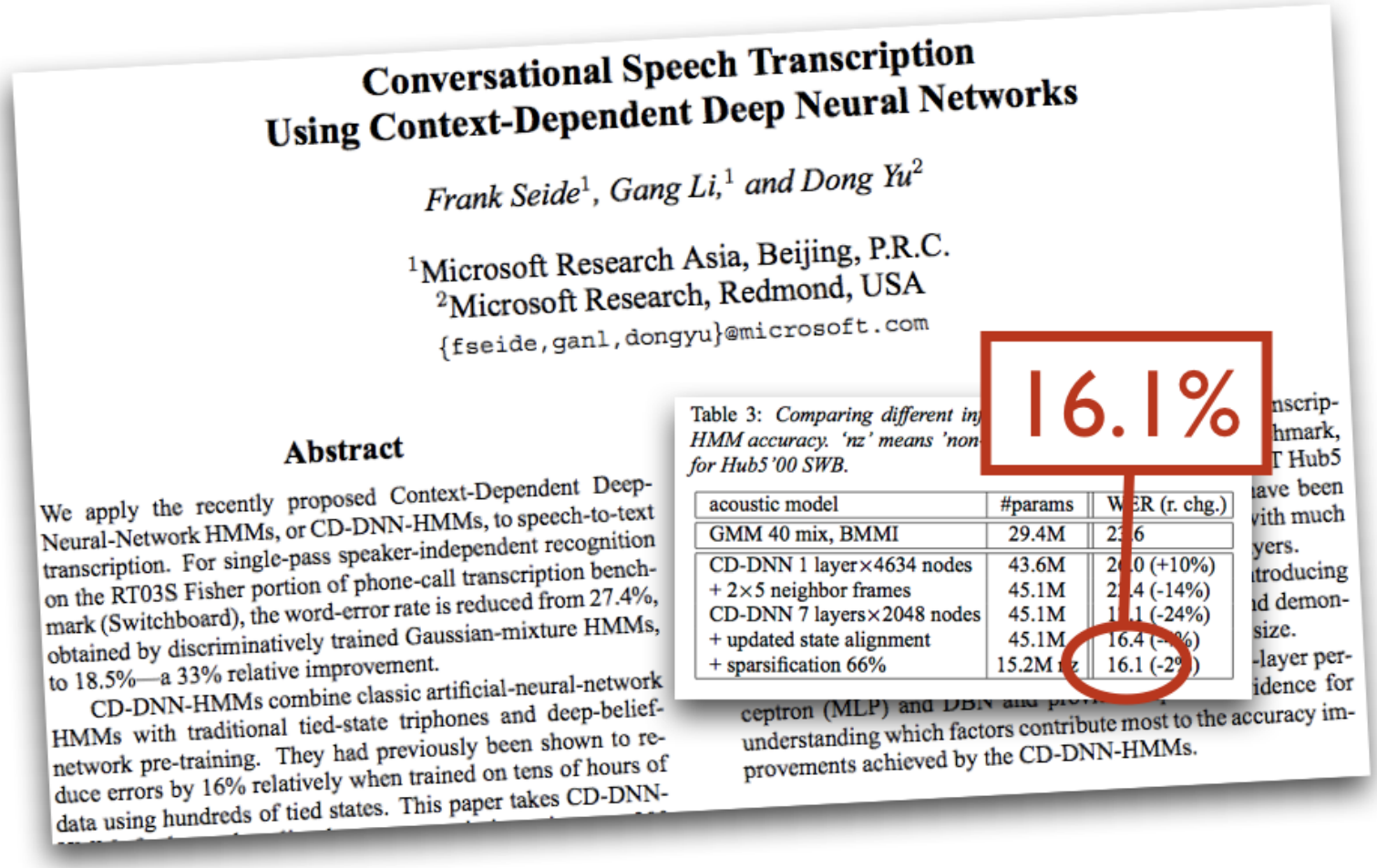
Features of the Cambridge system

	CU-HTK 2000
Base model	HMM-GMM
Acoustic context	Δ, $\Delta\Delta$ features, HLDA projection
Phonetic context	Tied state triphones & quinphones
Speaker adaptation	Gender-dependent models, VTLN, MLLR
Training criterion	ML + MMI sequence training
System architecture	6-pass system
Other features	Multi-system combination
Hub 2000 WER	**19.3%**

No neural networks!

Why were neural networks uncompetitive in 2000?

Ten years later



Conversational Speech Transcription Using Context-Dependent Deep Neural Networks

Frank Seide[1], *Gang Li*,[1] *and Dong Yu*[2]

[1]Microsoft Research Asia, Beijing, P.R.C.
[2]Microsoft Research, Redmond, USA
{fseide,ganl,dongyu}@microsoft.com

16.1%

Abstract

We apply the recently proposed Context-Dependent Deep-Neural-Network HMMs, or CD-DNN-HMMs, to speech-to-text transcription. For single-pass speaker-independent recognition on the RT03S Fisher portion of phone-call transcription benchmark (Switchboard), the word-error rate is reduced from 27.4%, obtained by discriminatively trained Gaussian-mixture HMMs, to 18.5%—a 33% relative improvement.

CD-DNN-HMMs combine classic artificial-neural-network HMMs with traditional tied-state triphones and deep-belief-network pre-training. They had previously been shown to reduce errors by 16% relatively when trained on tens of hours of data using hundreds of tied states. This paper takes CD-DNN-

Table 3: *Comparing different inj… HMM accuracy. 'nz' means 'non… for Hub5'00 SWB.*

acoustic model	#params	WER (r. chg.)
GMM 40 mix, BMMI	29.4M	23.6
CD-DNN 1 layer×4634 nodes	43.6M	2[illegible].0 (+10%)
+ 2×5 neighbor frames	45.1M	2[illegible].4 (-14%)
CD-DNN 7 layers×2048 nodes	45.1M	1[illegible].1 (-24%)
+ updated state alignment	45.1M	16.4 (-[illegible]%)
+ sparsification 66%	15.2M nz	16.1 (-2[illegible])

…nscrip-…hmark, …T Hub5 …ave been …vith much …yers. …troducing …d demon-…size. …-layer per-…idence for

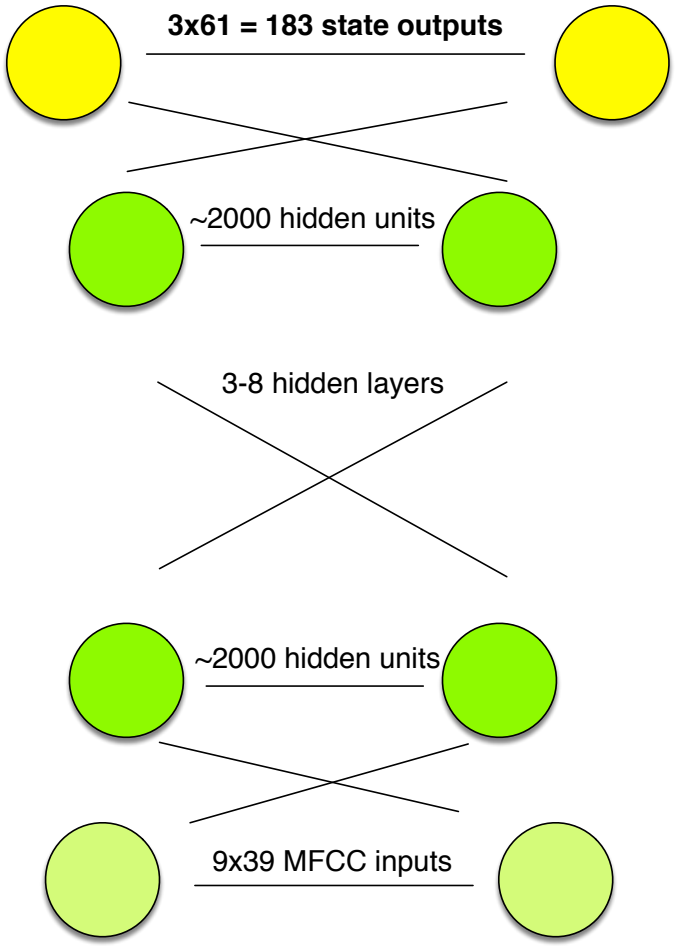
ceptron (MLP) and DBN and pre… understanding which factors contribute most to the accuracy improvements achieved by the CD-DNN-HMMs.

Features of the Microsoft NN system

	Microsoft 2011
Base model	HMM-DNN
Acoustic context	11 frames directly modelled
Phonetic context	Tied state triphones
Speaker adaptation	None
Training criteria	Frame-level cross-entropy
System architecture	Single pass
Other features	Deep network architecture
Hub 2000 WER	**16.1%**

DNN acoustic Models

Deep neural networks 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 3×61 states
- Training many hidden layers is computationally expensive – use GPUs to provide the computational power

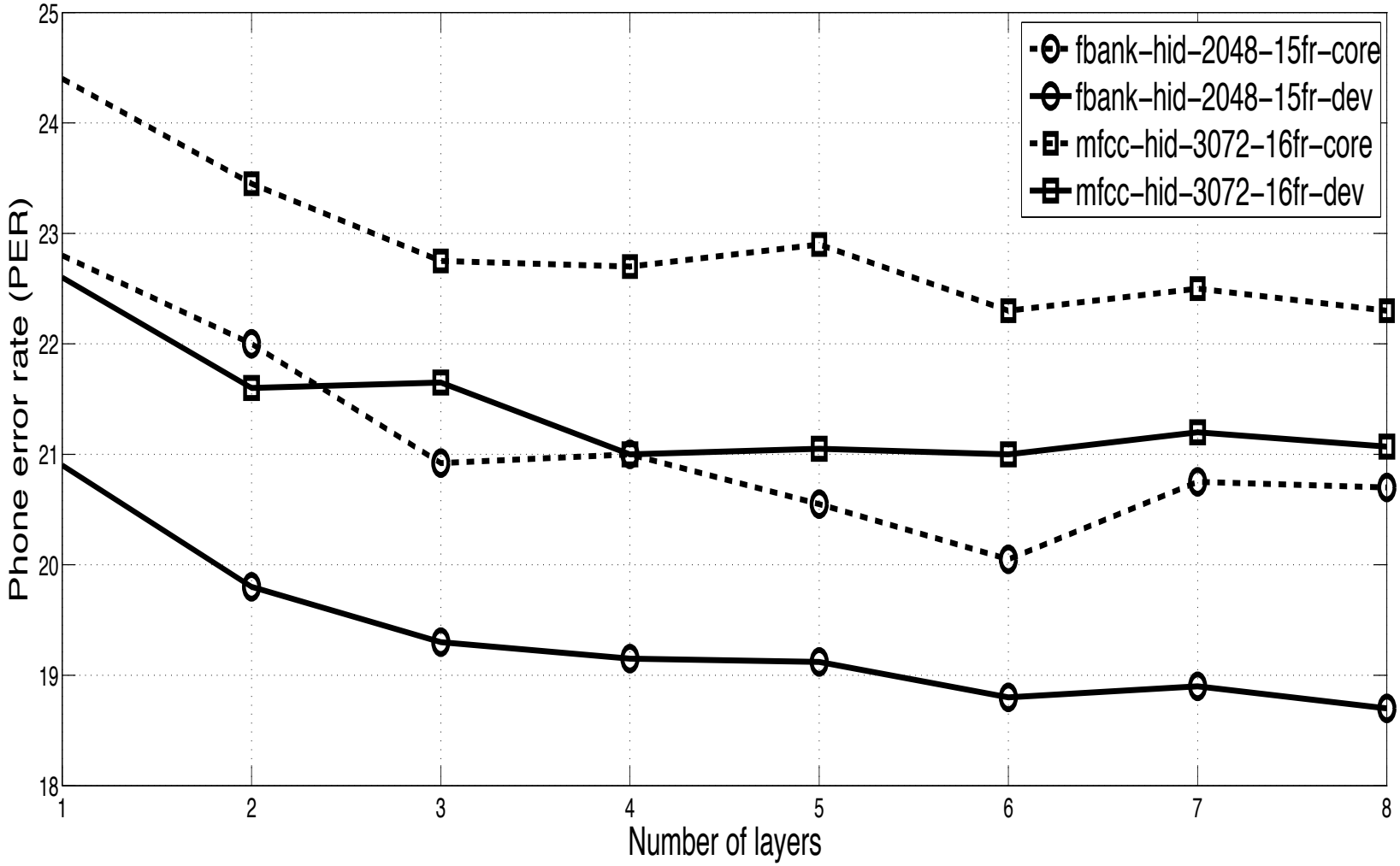
Hybrid HMM/DNN phone recognition (TIMIT)

- 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
- 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 well)
  - Can potentially use feature vectors with correlated components (e.g. filter banks)
- Experiments indicate that mel-scaled filter bank features (FBANK) result in greater accuracy than MFCCs

TIMIT phone error rates: effect of depth and feature type

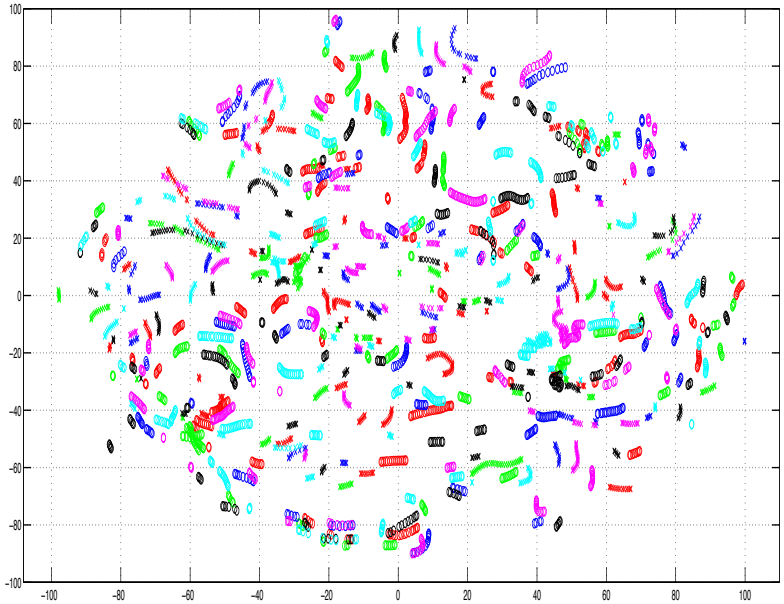


(Mohamed et al (2012))

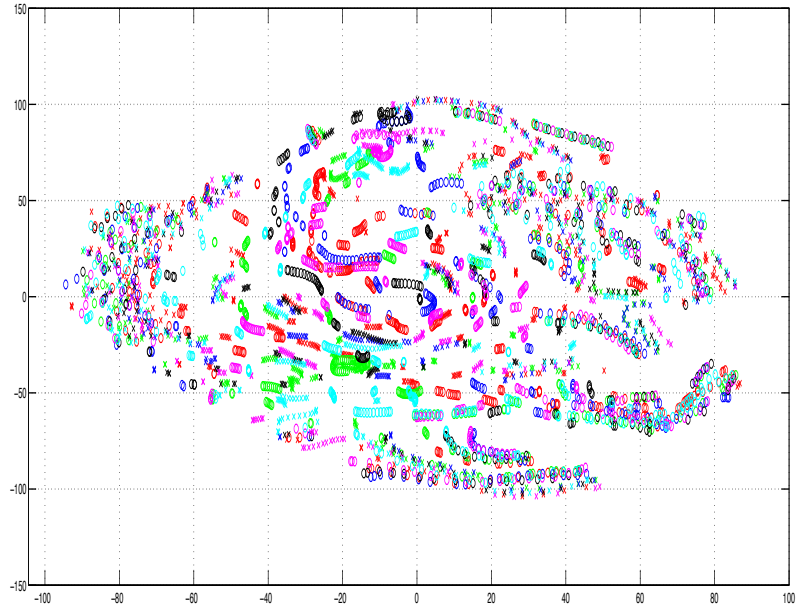
Visualising neural networks

- Visualise NN hidden layers to better understand the effect of different speech features (MFCC vs FBANK)
- 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

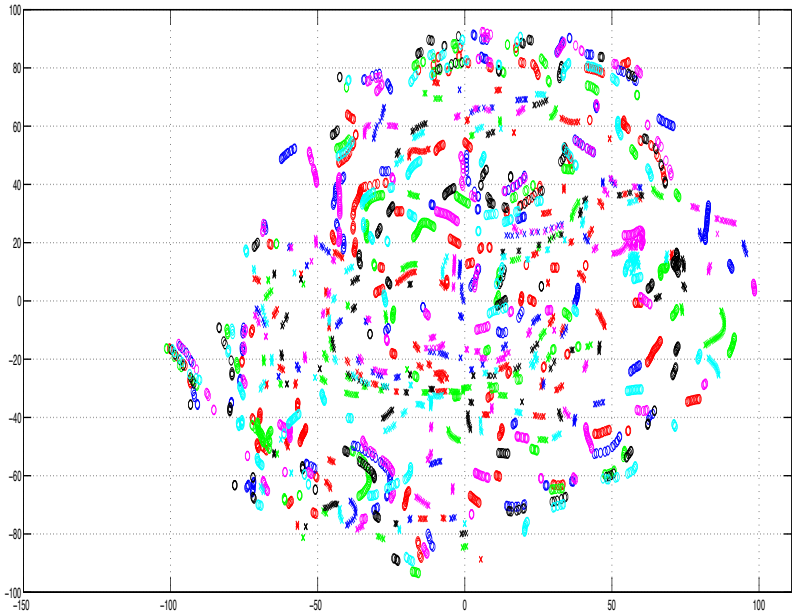
(Mohamed et al (2012))

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

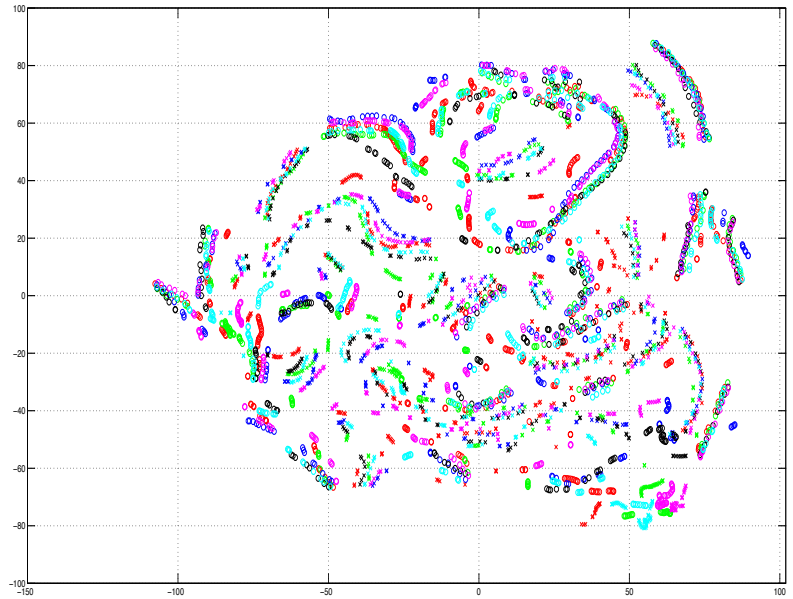
MFCCs are more scattered than FBANK

FBANK has more local structure than MFCCs

First hidden layer: t-SNE visualisation



MFCC



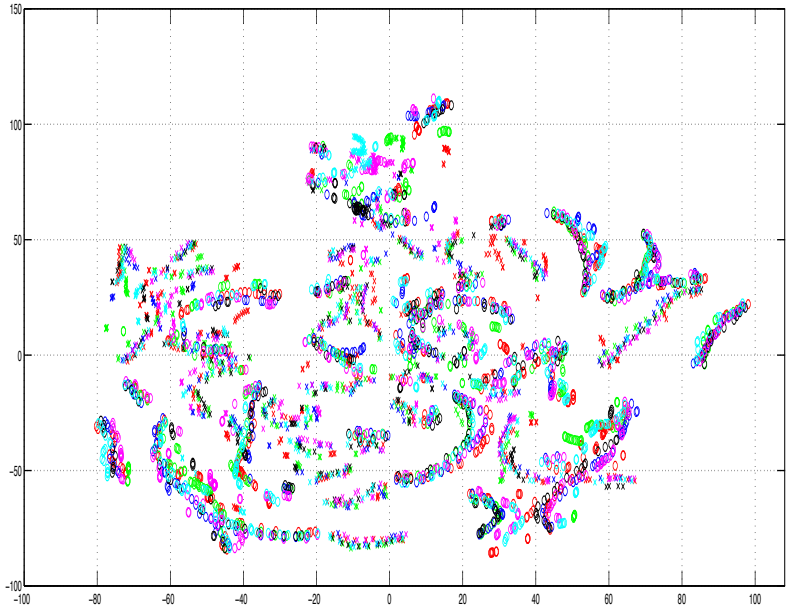
FBANK

(Mohamed et al (2012))

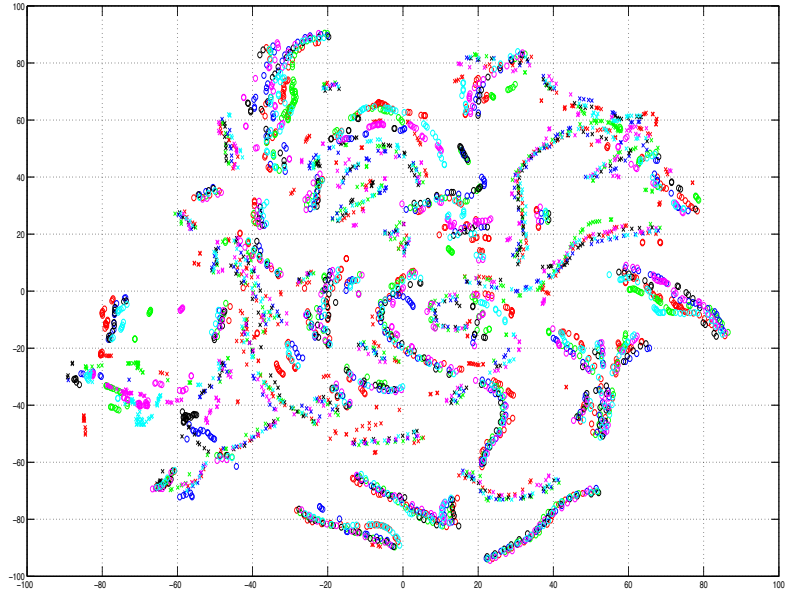
Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

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.

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)
- Background reading:
  - N Morgan and H Bourlard (May 1995). "Continuous speech recognition: Introduction to the hybrid HMM/connectionist approach", *IEEE Signal Processing Mag.*, **12**(3), 24–42. http://ieeexplore.ieee.org/document/382443
  - A Mohamed et al (2012). "Understanding how deep belief networks perform acoustic modelling", Proc ICASSP-2012. http://www.cs.toronto.edu/~asamir/papers/icassp12_dbn.pdf