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

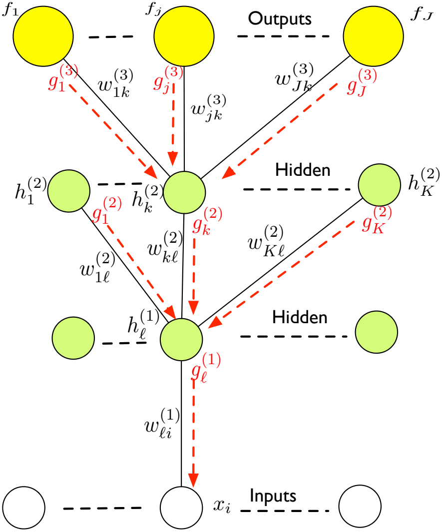
Peter Bell

Automatic Speech Recognition – ASR Lecture 11
28 February 2022

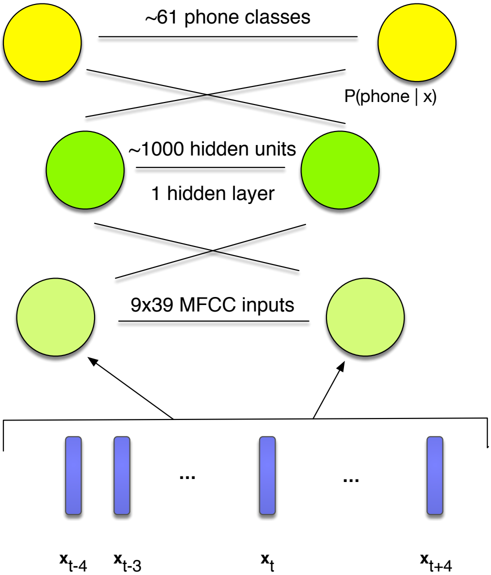
Training deep networks: Backprop and gradient descent

- 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 x_t, is an estimate of the posterior probability $P(q_t = j|\mathrm{x}_t)$. (This is because we have softmax outputs and use a cross-entropy loss function)
- Using Bayes Rule we can relate the posterior $P(q_t = j|\mathrm{x}_t)$ to the likelihood $p(\mathrm{x}_t|q_t = j)$ used as an output probability in an HMM:

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

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 $p(\mathrm{x}|q)$ – likelihoods.
  We can write *scaled likelihoods* as:

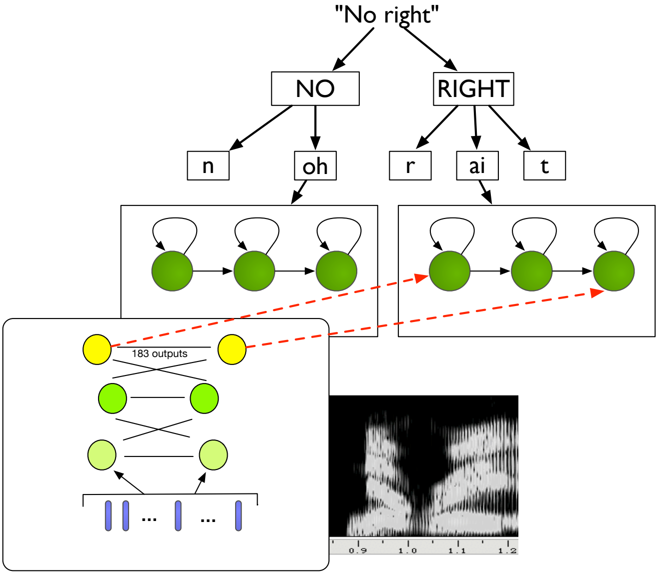
$$\frac{P(q_t = j|\mathrm{x}_t)}{p(q_t = j)} = \frac{p(\mathrm{x}_t|q_t = j)}{p(\mathrm{x}_t)}$$

- Scaled likelihoods can be obtained by "dividing by the priors" – divide each network output $P(q_t = j|\mathrm{x}_t)$ by $P(q_t)$, the relative frequency of class j in the training data
- Using $p(\mathrm{x}_t|q_t = j)/p(\mathrm{x}_t)$ rather than $p(\mathrm{x}_t|q_t = j)$ is OK since $p(\mathrm{x}_t)$ 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



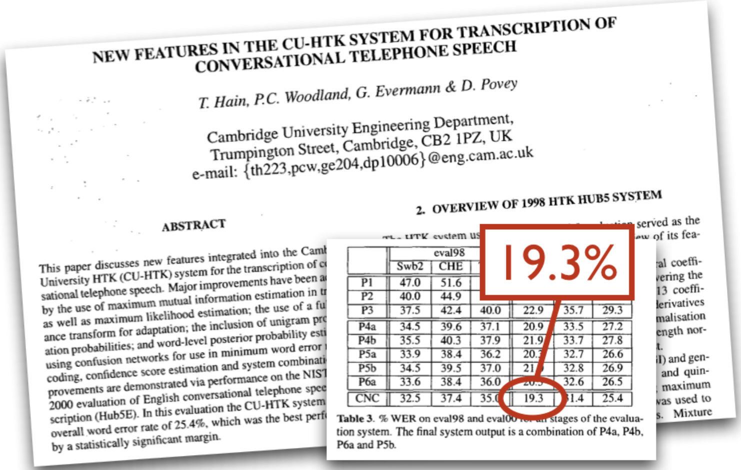
HMM/NN vs HMM/GMM

- 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 Camb University HTK (CU-HTK) system for the transcription of co sational telephone speech. Major improvements have been a by the use of maximum mutual information estimation in tr as well as maximum likelihood estimation; the use of a fu ance transform for adaptation; the inclusion of unigram pro ation probabilities; and word-level posterior probability esti using confusion networks for use in minimum word error coding, confidence score estimation and system combinati provements are demonstrated via performance on the NIST 2000 evaluation of English conversational telephone spee scription (Hub5E). In this evaluation the CU-HTK system overall word error rate of 25.4%, which was the best perf by a statistically significant margin.

2. OVERVIEW OF 1998 HTK HUB5 SYSTEM

The HTK system us ... served as the ... ew of its fea- ... al coeffi- ... vering the ... 13 coeffi- ... derivatives ... malisation ... ength nor- ... t. ... l) and gen- ... and quin- ... maximum ... was used to ... s. Mixture

	eval98					
	Swb2	CHE	[illegible]	[illegible]	[illegible]	[illegible]
P1	47.0	51.6	[illegible]	[illegible]	[illegible]	[illegible]
P2	40.0	44.9	[illegible]	[illegible]	[illegible]	[illegible]
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	[illegible]	32.7	26.6
P5b	34.5	39.5	37.0	[illegible]	32.8	26.9
P6a	33.6	38.4	36.0	[illegible]	32.6	26.5
CNC	32.5	37.4	[illegible]	19.3	[illegible]	25.4

Table 3. % WER on eval98 and eval00 [illegible] 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%**

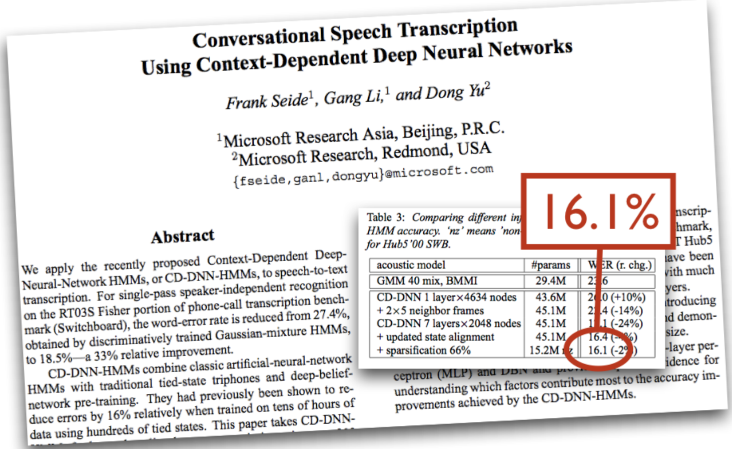
Features of the Cambridge system

	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

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 in[...] HMM accuracy. 'nz' means 'non[...] for Hub5'00 SWB.*

acoustic model	#params	WER (r. chg.)
GMM 40 mix, BMMI	29.4M	2[illegible]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 [illegible]	16.1 (-2[illegible])

16.1%

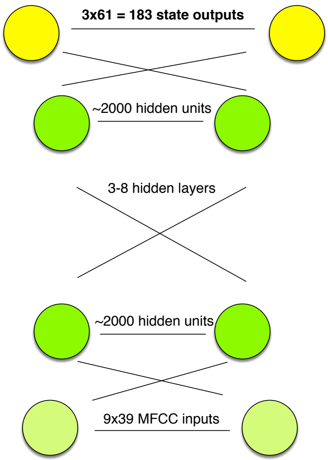
...nscrip- ...hmark, ...T Hub5 ...ave been ...vith much ...yers. ...troducing ...d demon- ...size. ...-layer per- ...idence for ...ceptron (MLP) and DBN ... understanding which factors contribute most to the accuracy improvements achieved by the CD-DNN-HMMs.

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

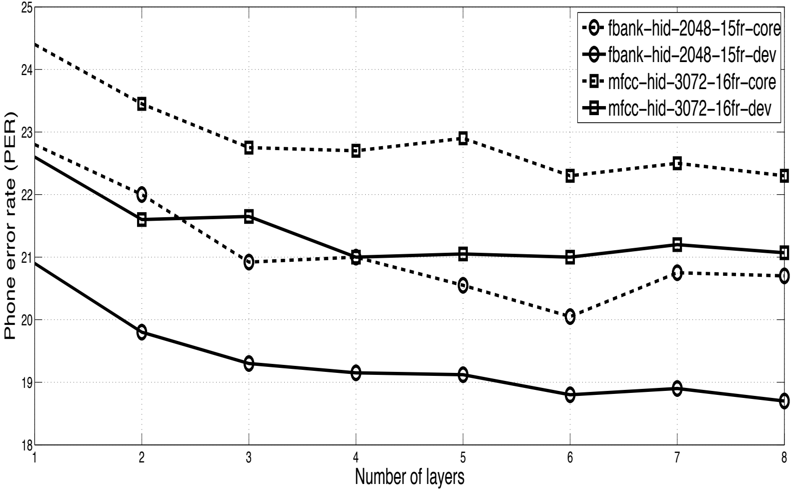
Hybrid HMM/DNN phone recognition (TIMIT)

- 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

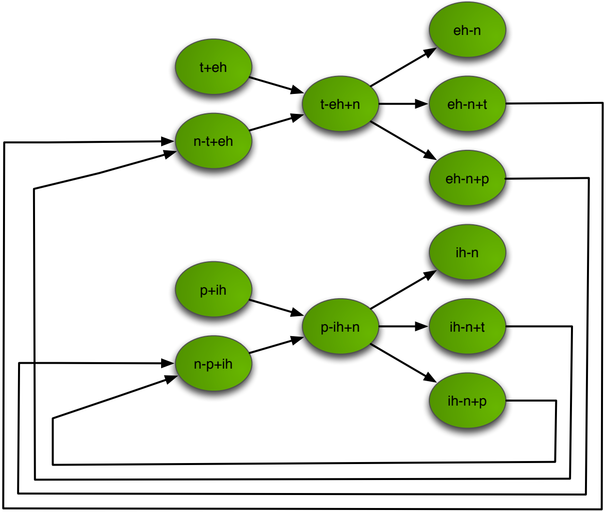
TIMIT phone error rates: effect of depth and feature type



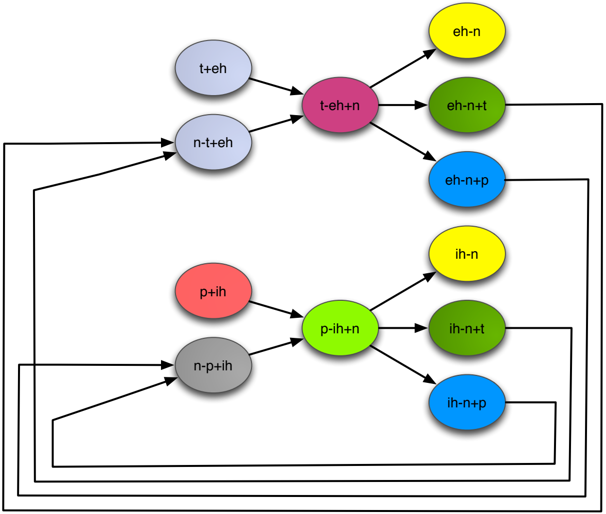
(Mohamed et al (2012))

Modelling phonetic context

Context-dependent units



Tied context-dependent units



Acoustic and phonetic context

One solution (Bourlard et al, 1992) –separate the modelling of the primary class, y, and its context, c, with two neural networks:

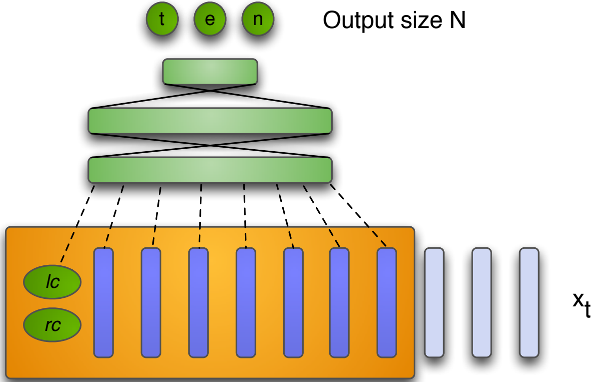
$$p(y, c|x) = p(c|y, x)p(y|x)$$

or

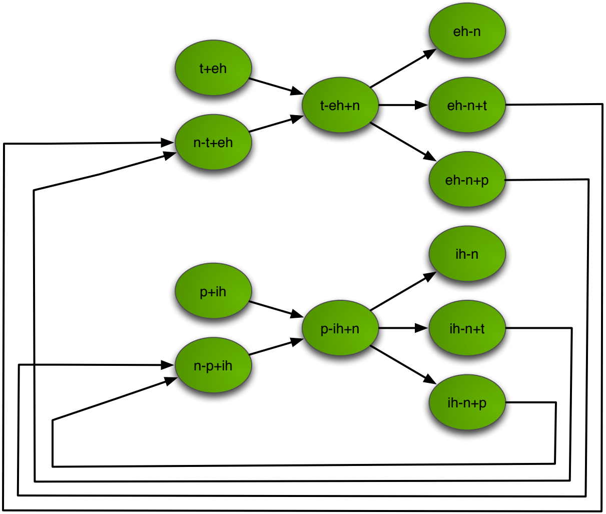
$$p(y, c|x) = p(y|c, x)p(c|x)$$

During decoding, we need separate forward passes for each context

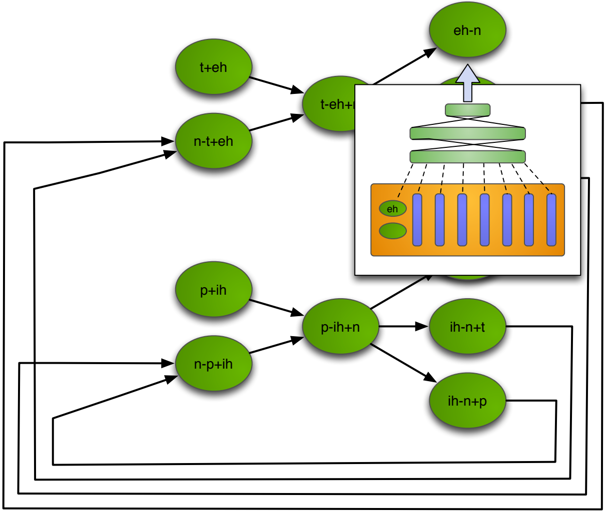
Using context as input for $p(y|c, x)$



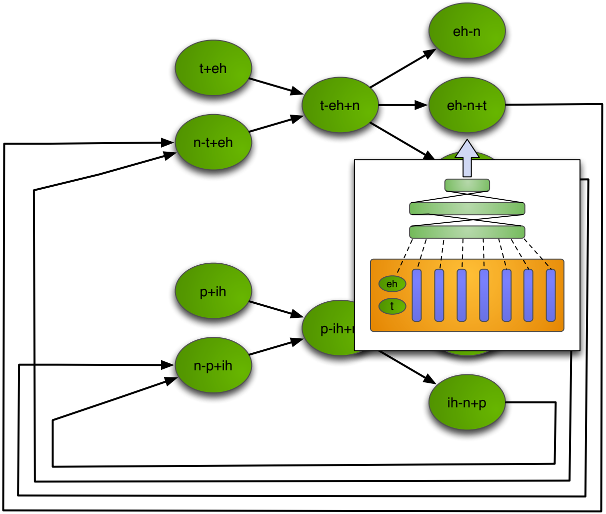
Context-dependent decoding



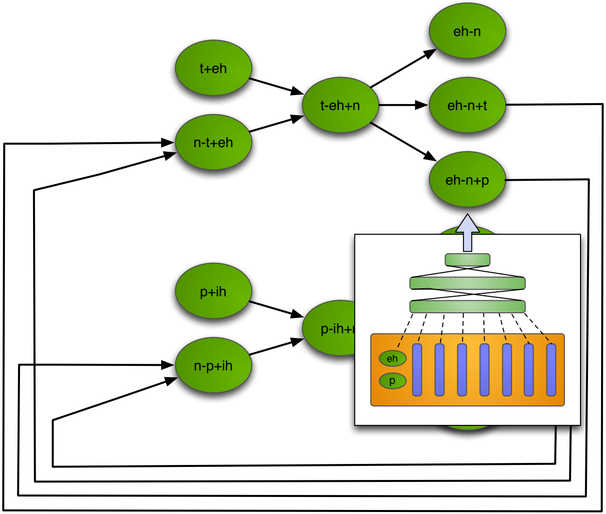
Context-dependent decoding



Context-dependent decoding



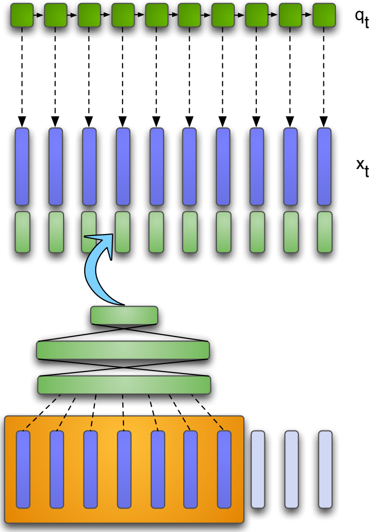
Context-dependent decoding



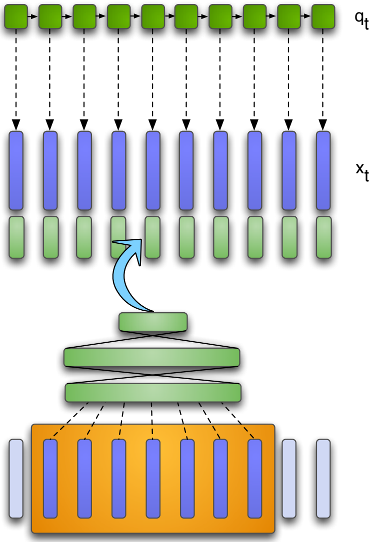
Tandem scheme

- Basic idea: use the output probabilities from the NN as input features to standard CD-HMM-GMM system
- Combines the benefits of both:
  - NNs good at modelling wide acoustic contexts, correlated input features
  - HMM-GMMs good for speaker adaptation, modelling phonetic context, sequence-training
- NN output probabilities are *Gaussianised* by taking logs and decorrelating with PCA
- Early variants used purely NN features; later variants augmented the feature vector with standard acoustic features
- Can also use "bottleneck features" (narrow, intermediate NN layers)

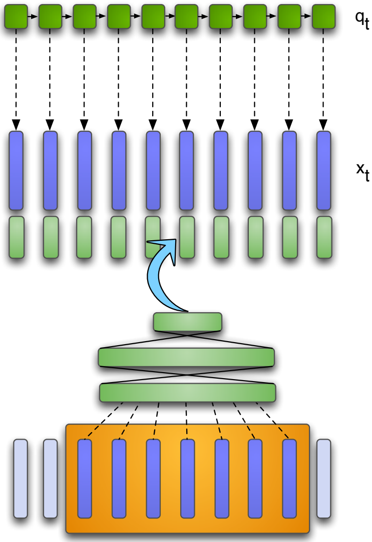
Tandem scheme



Tandem scheme



Tandem scheme

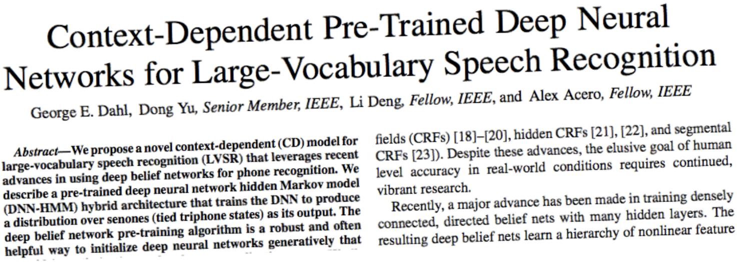


Modelling phonetic context with DNNs

- In the 1990s, this was considered hard (see Bourlard et al, 1992)
- But in 2011, a simple solution emerged: use state-tying from a GMM system

Modelling phonetic context with DNNs

- In the 1990s, this was considered hard (see Bourlard et al, 1992)
- But in 2011, a simple solution emerged: use state-tying from a GMM system



Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition

George E. Dahl, Dong Yu, *Senior Member, IEEE*, Li Deng, *Fellow, IEEE*, and Alex Acero, *Fellow, IEEE*

***Abstract*—We propose a novel context-dependent (CD) model for large-vocabulary speech recognition (LVSR) that leverages recent advances in using deep belief networks for phone recognition. We describe a pre-trained deep neural network hidden Markov model (DNN-HMM) hybrid architecture that trains the DNN to produce a distribution over senones (tied triphone states) as its output. The deep belief network pre-training algorithm is a robust and often helpful way to initialize deep neural networks generatively that**

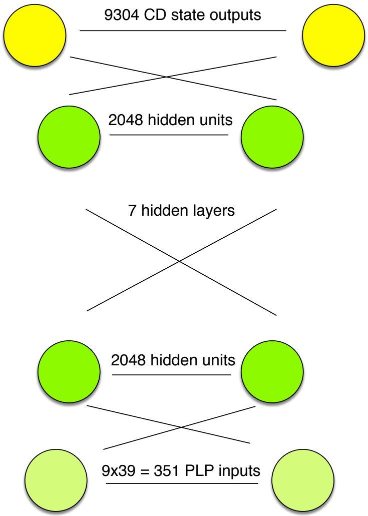
fields (CRFs) [18]–[20], hidden CRFs [21], [22], and segmental CRFs [23]). Despite these advances, the elusive goal of human level accuracy in real-world conditions requires continued, vibrant research.

Recently, a major advance has been made in training densely connected, directed belief nets with many hidden layers. The resulting deep belief nets learn a hierarchy of nonlinear feature

Context-dependent hybrid HMM/DNN

- First train a context-dependent HMM/GMM system on the same data, using a phonetic decision tree to determine the HMM tied states
- Perform Viterbi alignment using the trained HMM/GMM and the training data
- Train a neural network to map the input speech features to a label representing a context-dependent tied HMM state
  - So the size of the label set is thousands (number of context-dependent tied states) rather than tens (number of context-independent phones) Each frame is labelled with the Viterbi aligned tied state
- Train the neural network using gradient descent as usual
- Use the context-dependent scaled likelihoods obtained from the neural network when decoding

Example: HMM/DNN acoustic model for Switchboard



(Siede et al (2011))

Example: HMM/DNN acoustic model for Switchboard

- Alignments generated from context-dependent HMM/GMM system
- Hybrid HMM/DNN system
  - Context-dependent — 9304 output units obtained from Viterbi alignment of HMM/GMM system
  - 7 hidden layers, 2048 units per layer
  - 11 frames of acoustic context
- DNN-based system results in significant word error rate reduction compared with GMM-based system
- Note: still no speaker adaptation or sequence-level training

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