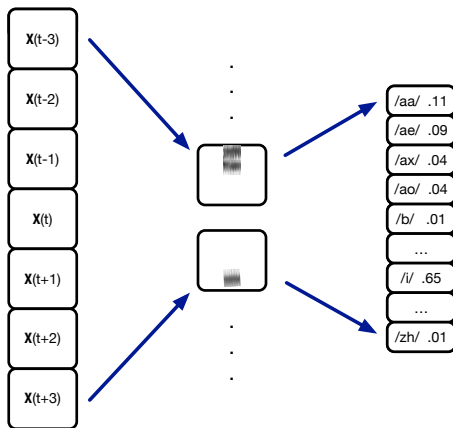


Neural Networks for Acoustic Modelling 2: Hybrid HMM/DNN systems

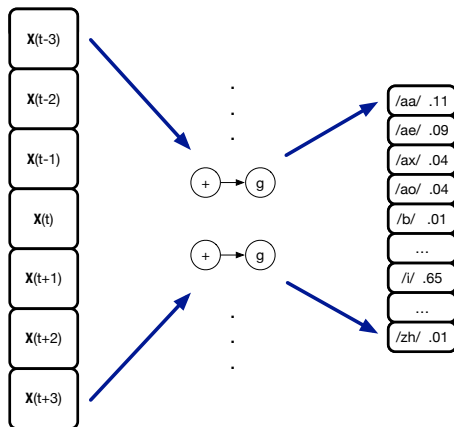
Steve Renals

Automatic Speech Recognition – ASR Lecture 8
7 February 2019

Recap: Hidden units extracting features



Recap: Hidden Units



$$h_j = \text{relu} \left(\sum_{i=1}^d v_{ji} x_i + b_j \right)$$

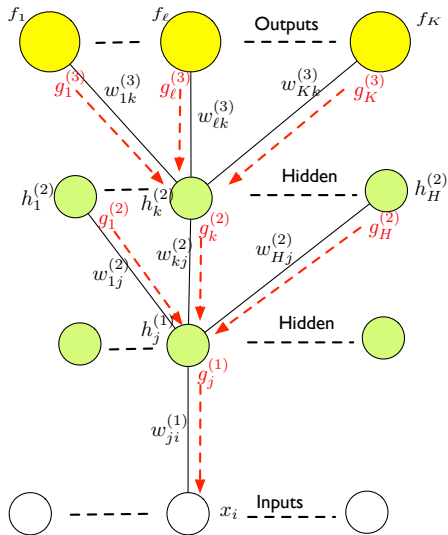
$$f_k = \text{softmax} \left(\sum_{j=1}^H w_{kj} h_j + b_k \right)$$

Training deep networks: Backprop and gradient descent

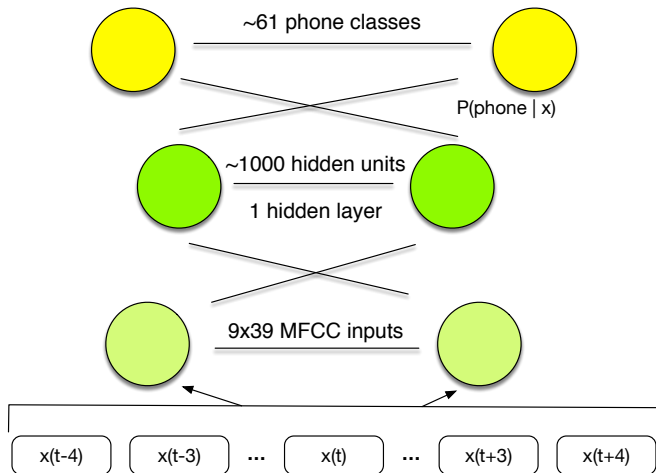
- Hidden units make training the weights more complicated, since each hidden unit affects the error function indirectly via all the output units
- The credit assignment problem: what is the “error” of a hidden unit? how important is input-hidden weight w_{ji} to output unit k ?
- Solution: *back-propagate* the gradients through the network – the gradient for a hidden unit output with respect to the error¹ 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 gradients through a deep network to allow gradient descent training to be performed

¹And this gradient can be easily used to compute the gradients of the error with respect to the weights into that hidden unit

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

Interim conclusions

- Neural networks using cross-entropy (CE) and softmax outputs give us a way of assigning the probability of each possible phonetic label for a given frame of data
- Hidden layers provide a way for the system to learn representations of the input data
- All the weights and biases of a network may be trained by gradient descent – back-propagation of error provides a way to compute the gradients in a deep network
- Acoustic context can be simply incorporated into such a network by providing multiples frame of acoustic input

Neural networks for phone recognition

- Train a neural network to associate a phone label with a frame of acoustic data (+ context)
- Can interpret the output of the network as $P(\text{phone} \mid \text{acoustic-frame})$
- **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 classifier (= phone probability estimator)
 - Use NN to obtain output probabilities in Viterbi algorithm to find most probable sequence of phones (words)

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

$$P(q|\mathbf{x}) = \frac{p(\mathbf{x}|q)P(q)}{p(\mathbf{x})}$$

(this is assuming 1 state per phone q)

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.

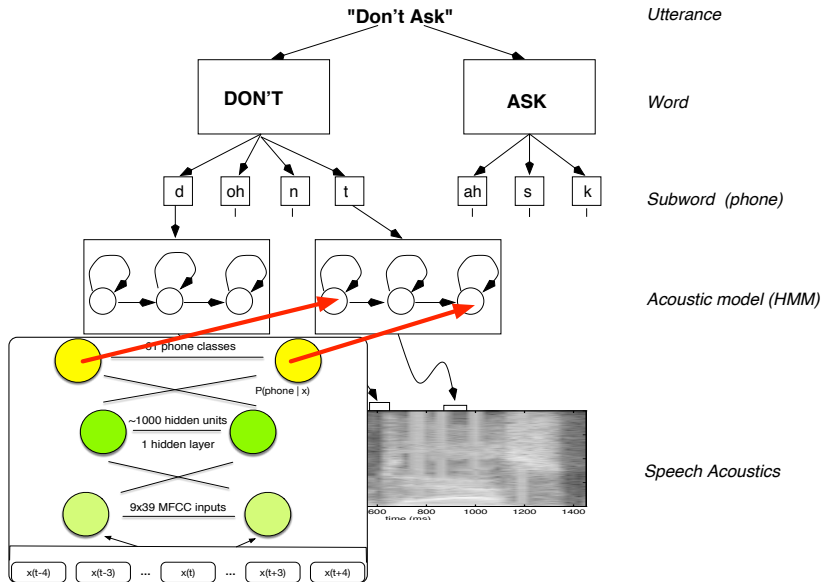
We can write *scaled likelihoods* as:

$$\frac{P(q|\mathbf{x})}{p(q)} = \frac{p(\mathbf{x}|q)}{p(\mathbf{x})}$$

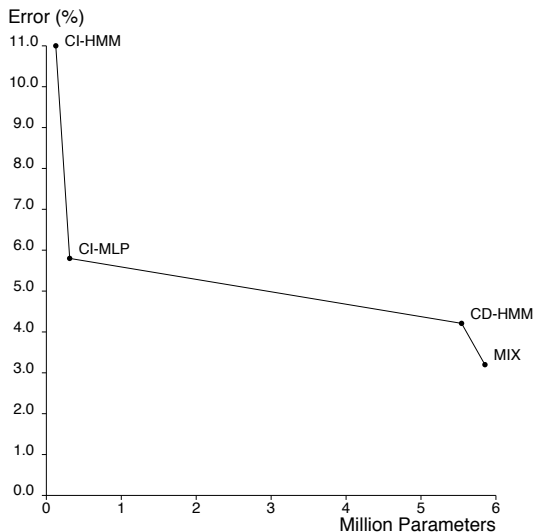
- Scaled likelihoods can be obtained by “dividing by the priors” – divide each network output $P(q|\mathbf{x})$ by $P(q)$, the relative frequency of class q in the training data
- Using $p(\mathbf{x}|q)/p(\mathbf{x})$ rather than $p(\mathbf{x}|q)$ is OK since $p(\mathbf{x})$ does not depend on the class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

- If we have a K -state HMM system, then we train a K -output NN to estimate the scaled likelihoods used in a hybrid system.
- For TIMIT, using a 1 state per phone systems, we obtain scaled likelihoods from a NN trained to classify phones.
- For continuous speech recognition we can use:
 - 1 state per phone (61 NN outputs, if we have 61 phone classes)
 - 3 state CI models ($61 \times 3 = 183$ NN outputs)
 - State-clustered models, with one NN output per tied state (this can lead to networks with many outputs!) (next lecture)
- 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

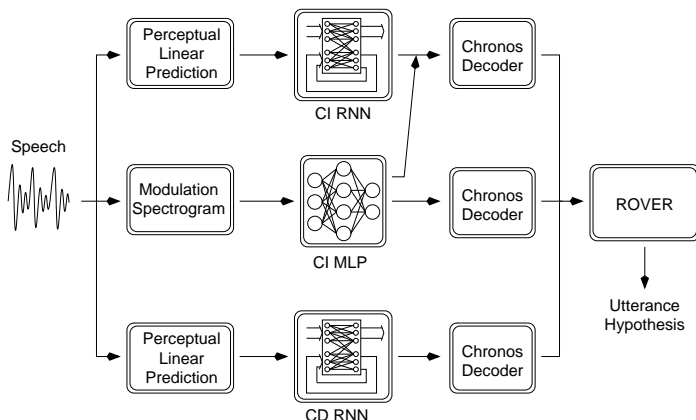


Monophone HMM/NN hybrid system (1993)



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

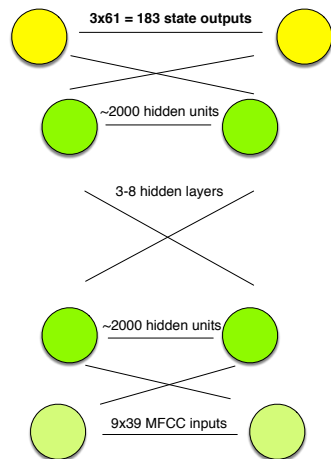
- Advantages of NN:
 - Can easily model **correlated features**
 - Correlated feature vector components (eg spectral features)
 - Input context – multiple frames of data at input
 - **More flexible** than GMMs – not made of (nearly) local components); GMMs inefficient for non-linear class boundaries
 - 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 ~ 2012:
 - Context-independent (monophone) models, weak speaker adaptation algorithms
 - NN systems less complex than GMMs (fewer parameters):
RNN – < 100k parameters, MLP – ~ 1M parameters
 - Computationally expensive - more difficult to parallelise training than GMM systems

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

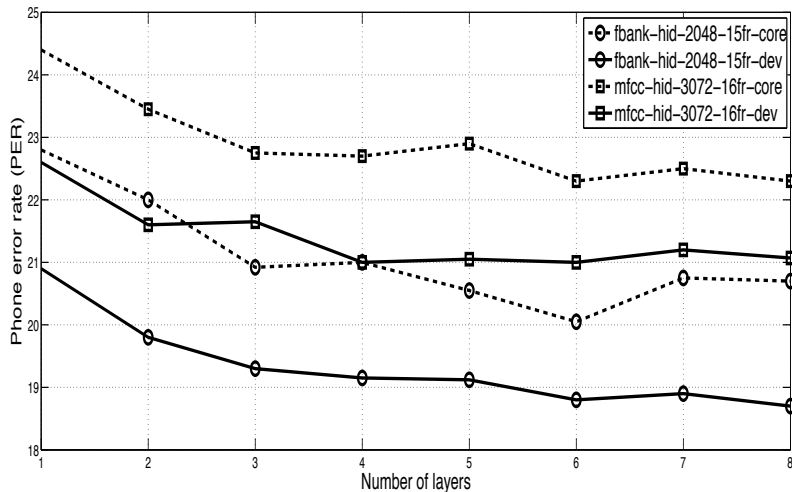
Hybrid HMM/DNN phone recognition (TIMIT)

- 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

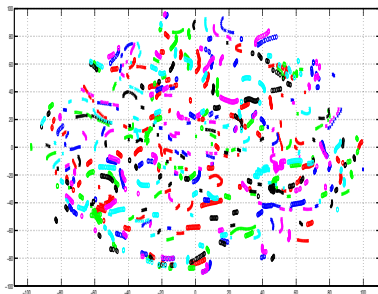


(Mohamed et al (2012))

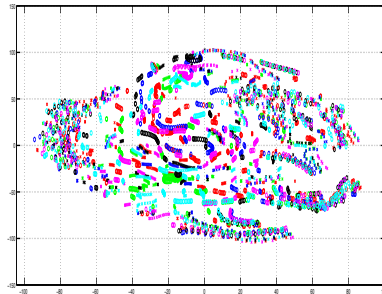
Visualising neural networks

- 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

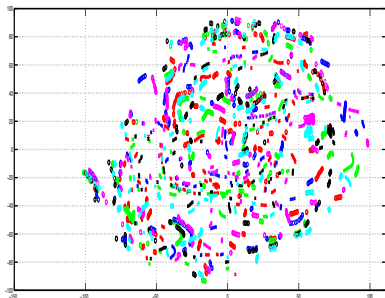
(Mohamed et al (2012))

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

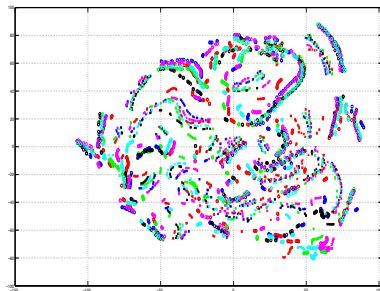
MFCCs are more scattered than FBANK

FBANK has more local structure than MFCCs

First hidden layer: t-SNE visualisation



MFCC



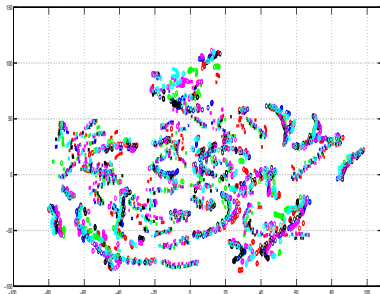
FBANK

(Mohamed et al (2012))

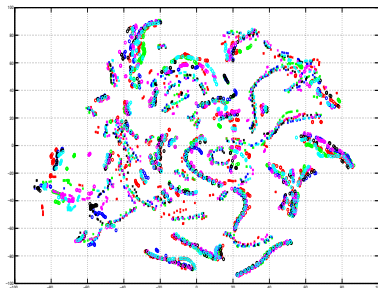
Visualisation of 2 utterances (cross and circle) spoken by 6 speakers (colours)

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 dimensions 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