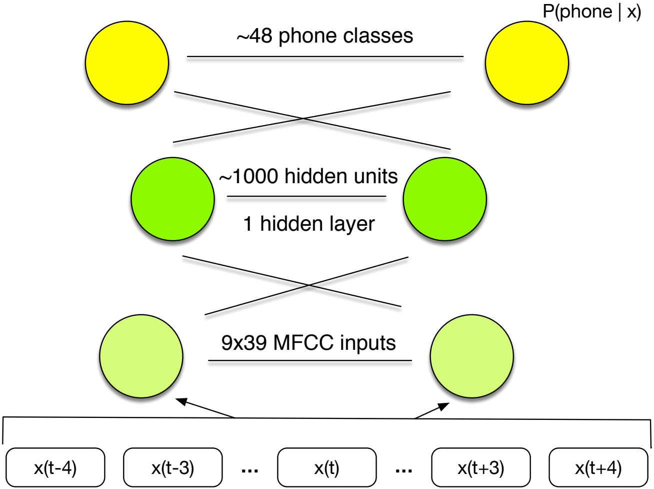
Neural Networks for Acoustic Modelling part 1

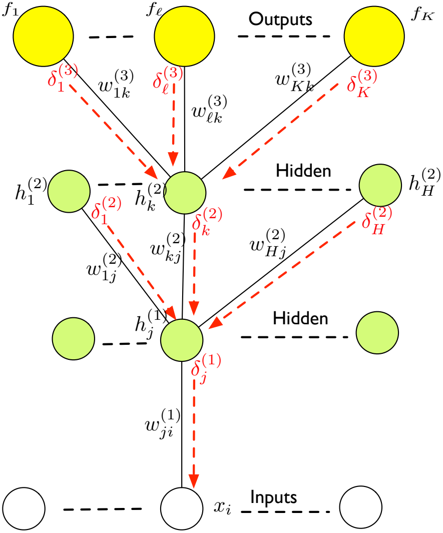
Steve Renals

Automatic Speech Recognition – ASR Lecture 8
13 February 2017

Recap: Neural networks for phone classification



Recap: Training NNs using back-propagation of error



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, usually 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 label with a frame of acoustic data (+ context)
- Can interpret the output of the network as P(phone | 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)

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 q given an input $\mathbf{x}$, is an estimate of the posterior probability $P(q|\mathbf{x})$
- 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:

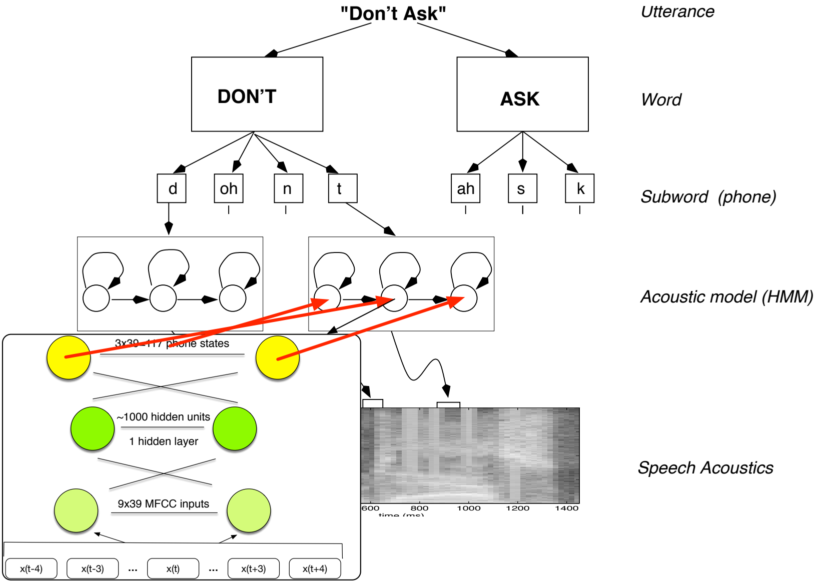
$$\frac{P(q|\mathbf{x})}{p(q)} = \frac{p(\mathbf{x}|q)}{p(\mathbf{x})}$$

- 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
- We can use the scaled likelihoods obtained from a neural network in place of the usual likelihoods obtained from a GMM

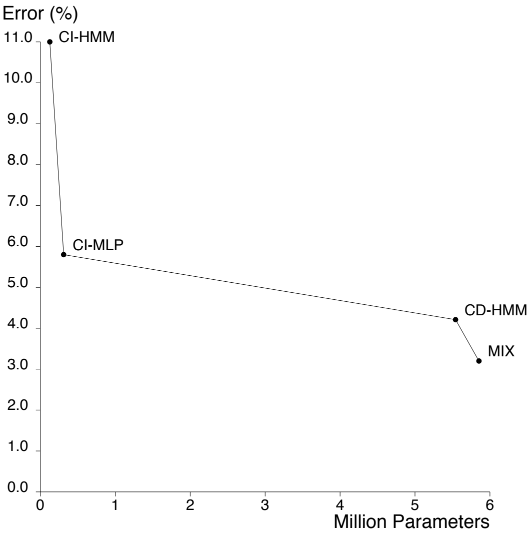
Hybrid NN/HMM

- 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 models
  - 3 state CI models (so we would have an NN with $39 \times 3 = 117$ outputs)
  - State-clustered models, with one NN output per tied state (this can lead to networks with many outputs!)
- Scaled likelihood and dividing by the priors
  - One can interpret computing the scaled likelihoods 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

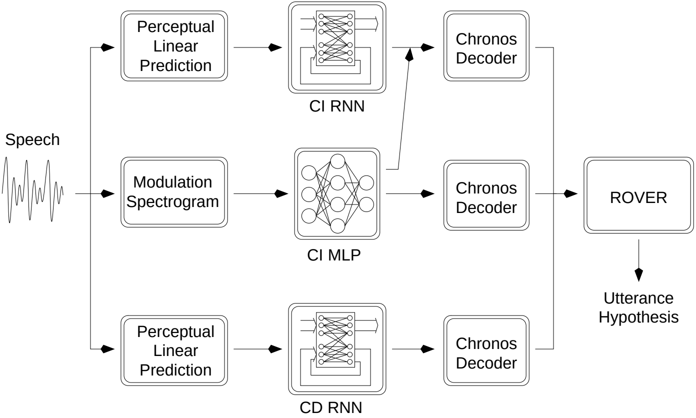


Monophone HMM/NN hybrid system (1993)



Renals, Morgan, Cohen & Franco, ICASSP 1992

Monophone HMM/NN hybrid system (1998)

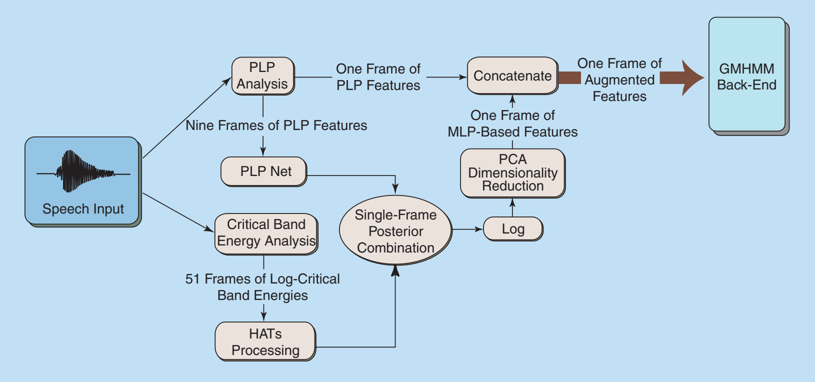


- Broadcast news transcription (1998) – 20.8% WER
- (best GMM-based system, 13.5%)
- Cook et al, DARPA, 1999

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

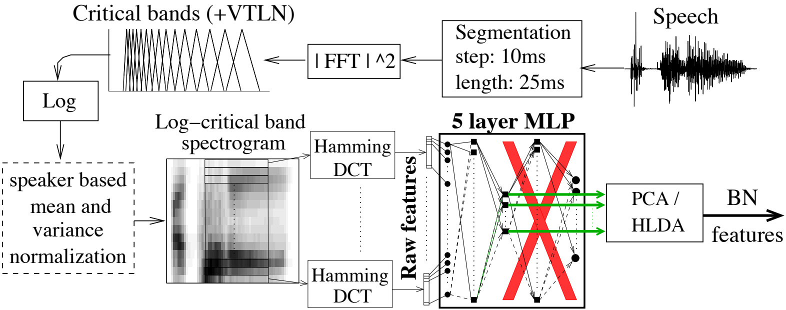
Tandem features



Speech Input
PLP Analysis
One Frame of PLP Features
Concatenate
One Frame of Augmented Features
GMHMM Back-End
Nine Frames of PLP Features
One Frame of MLP-Based Features
PLP Net
PCA Dimensionality Reduction
Critical Band Energy Analysis
Single-Frame Posterior Combination
Log
51 Frames of Log-Critical Band Energies
HATs Processing

Morgan et al (2005)

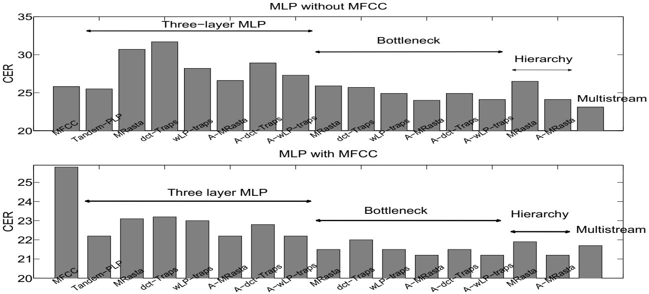
Bottleneck features



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)

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

Summary

- Hybrid neural network / HMM systems – using NN acoustic models to compute the output probabilities for HMMs
  - NNs trained as a phone classifier estimate posterior probabilities P(phone | acoustic-frame)
  - Scaled likelihoods – divide by the phone priors to obtained (scaled) likelihoods to use as HMM output probabilities
- Neural network features – append features obtained from a trained NN to acoustic features (e.g. MFCCs)
  - Tandem / posteriorgram: use the (transformed) output of an NN trained as a phone classifier as additional features for a GMM system
  - Bottleneck features: use a the (transformed) hidden layer output of an NN trained as a phone classifier as additional features for a GMM system

Next lecture: Deep neural network acoustic models

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
- N Morgan et al (Sep 2005). "Pushing the envelope – aside", *IEEE Signal Processing Mag.*, **22**(5), 81–88. http://ieeexplore.ieee.org/document/1511826
- F Grezl and P Fousek (2008). "Optimizing bottleneck features for LVCSR", Proc ICASSP–2008. http://ieeexplore.ieee.org/document/4518713
- F Valente et al (2011). "Analysis and Comparison of Recent MLP Features for LVCSR Systems", Proc Interspeech–2011. http://www.isca-speech.org/archive/interspeech_2011/i11_1245.html