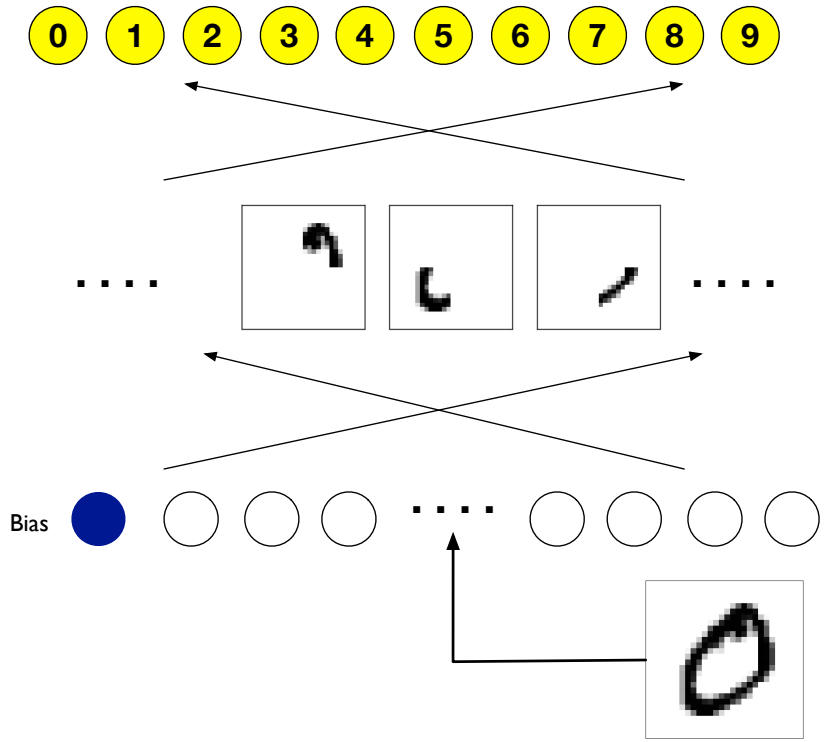
Neural Networks for Acoustic Modelling

Steve Renals

Automatic Speech Recognition – ASR Lecture 11
22 February 2016

Neural Networks

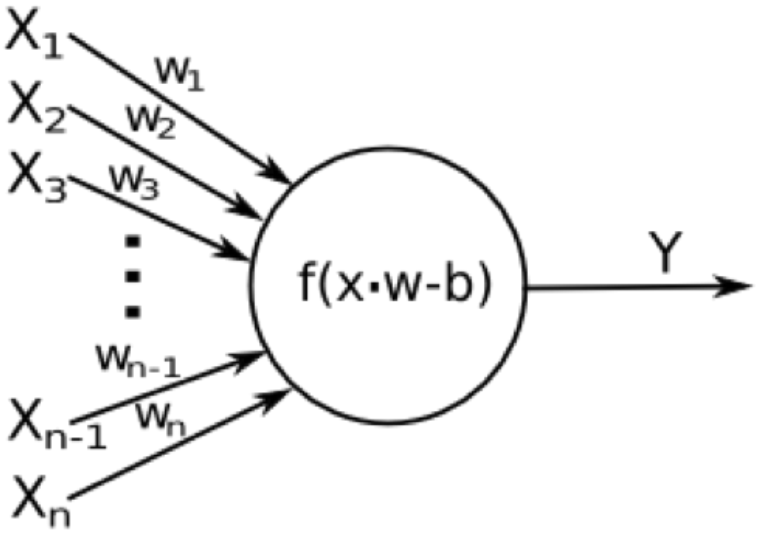


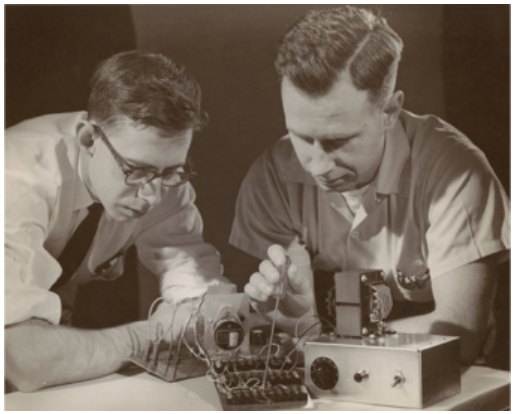
- Supervised training (stochastic gradient descent), classification (associating features with classes)
- Multi-layered
  - Hidden layer - weighted sum of inputs, followed by sigmoid transfer function
  - Output layer - softmax (1-from-N classification)

Historical Perspective

The Perceptron

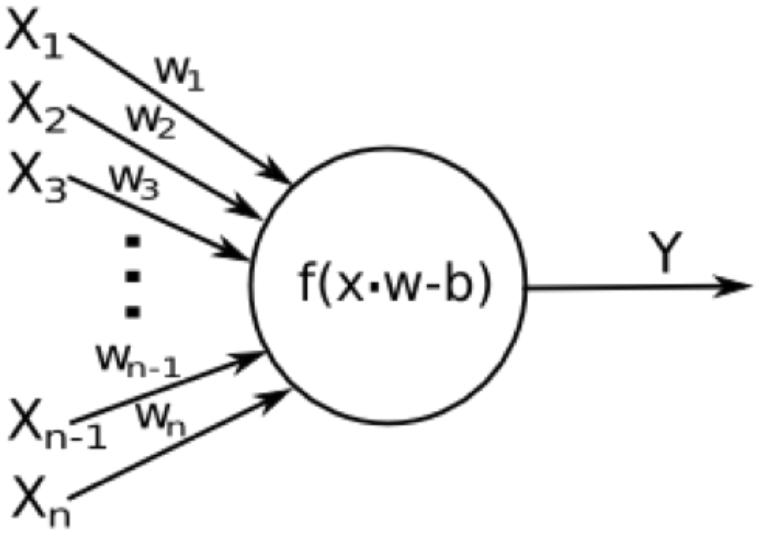
F Rosenblatt – late 1950s, early 1960s

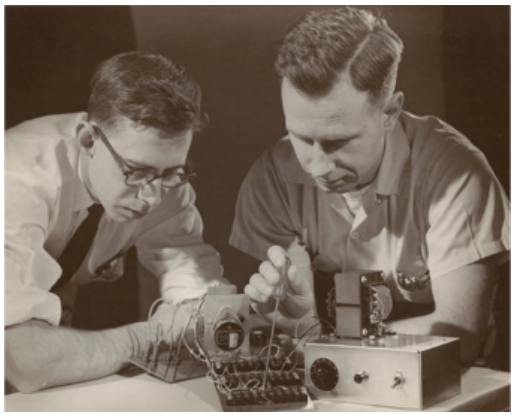


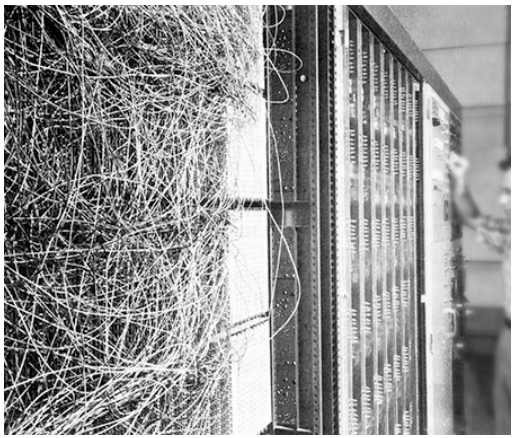


The Perceptron

F Rosenblatt – late 1950s, early 1960s

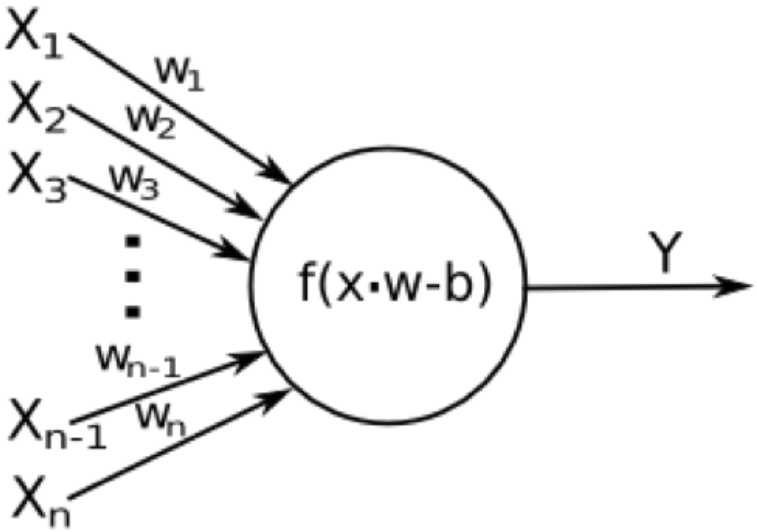


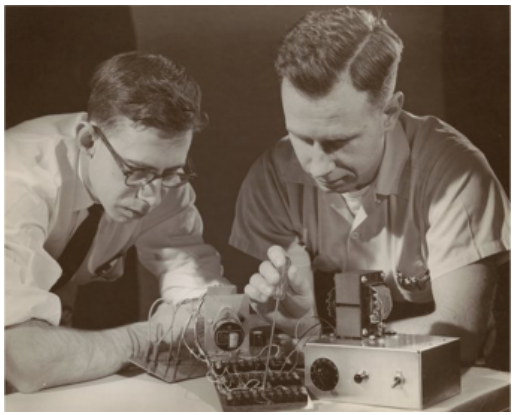


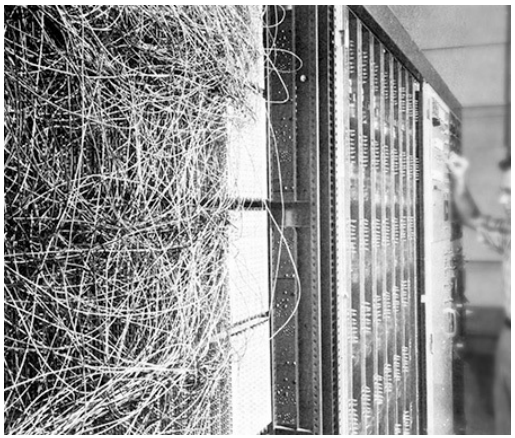


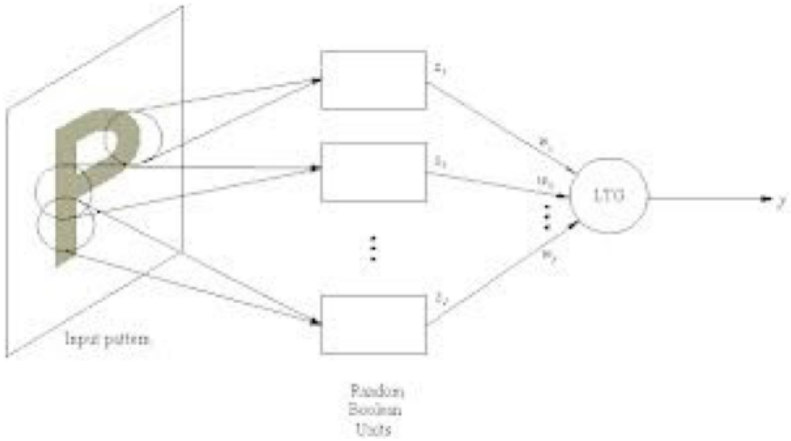
The Perceptron

F Rosenblatt – late 1950s, early 1960s



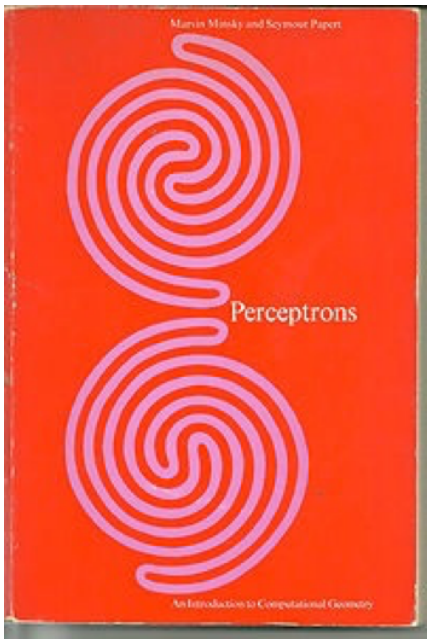


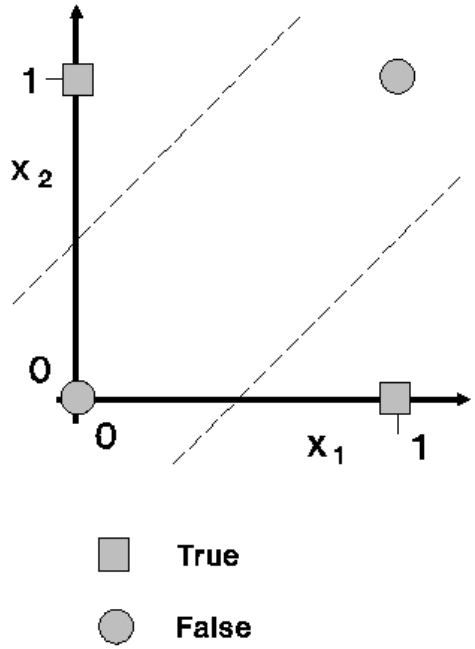




The first "neural network winter" (late 1960s – mid 1980s)

Work in AI focused on rule-based systems and logic from the late 1960s to the mid 1980s





xor

Minsky and Papert's book (*Perceptrons*) explored the capabilities and limitations of neural networks mathematically – but was perceived to show that perceptrons were unable to model some functions and were hence unsuitable for AI

Hidden Markov Models

However, the "cybernetic underground" developed statistical models during the 1970s...

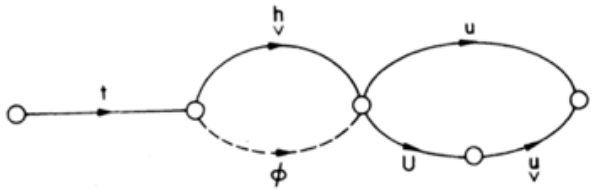


Fig. 9. A phonetic Markov subsource.

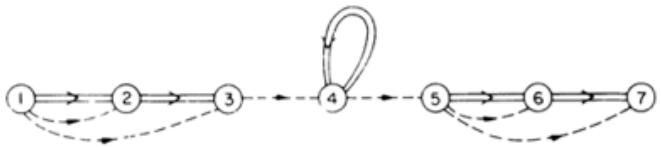


Fig. 10. An acoustic Markov subsource.

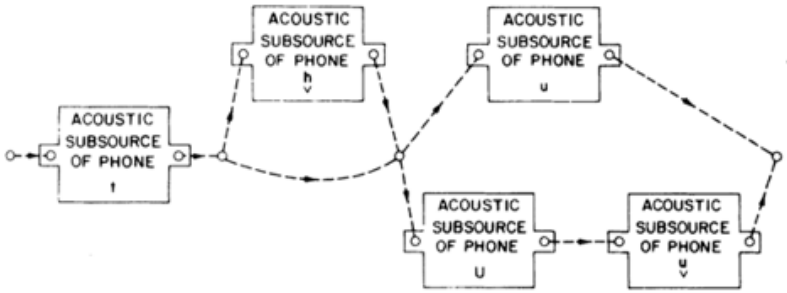


Fig. 11. A phone-based Markov source based on the phonetic subsource of Fig. 9.

GENERAL MODEL

Let the sequence $X(1),X(2),X(3),\cdots,X(T)$ be the sequence of states of a Markov process [3] with transition matrix $A = (a_{i,j})$. Let $Y(1),Y(2),Y(3),\cdots,Y(T)$ be a sequence of random variables such that, for all t, $\Pr(Y(t) = k \mid X(t-1) = i,\ X(t) = j) = b_{i,j,k}$. Use a bracket and colon notation to abbreviate sequences. Thus $X[1:T] = X(1),X(2),X(3),\cdots,X(T)$ and $Y[1:T] = Y(1),Y(2),Y(3),\cdots,Y(T)$. The assumptions of the model are that

$$\begin{aligned}\Pr(Y(t) = y(t) \mid X[1:t] &= x[1:t], Y[1:t-1] = y[1:t-1]) \\ &= \Pr(Y(t) = y(t) \mid X(t-1) = x(t-1), X(t) = x(t)) \\ &= b_{x(t-1)x(t),y(t)} \qquad (1)\end{aligned}$$

and

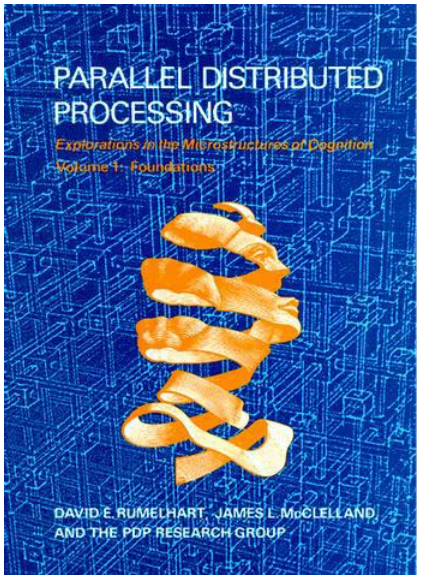
$$\begin{aligned}\Pr(X(t) = x(t) \mid X[1:t-1] = x[1:t-1]) &= \Pr(X(t) = x(t) \mid X(t-1) = x(t-1)) \\ &= a_{x(t-1),x(t)}. \qquad (2)\end{aligned}$$

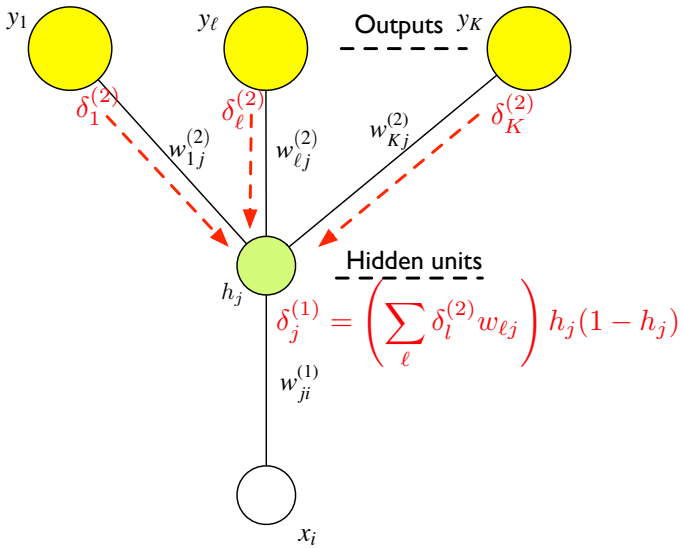
Under these assumptions,

$$\begin{aligned}\Pr(X[1:T] = x[1:T], Y[1:T] = y[1:T]) &= \prod_{t=1,T} a_{x(t-1),x(t)} b_{x(t-1),x(t),y(t)} \qquad (3)\end{aligned}$$

where a special extra state $x(0)$ is introduced and $a_{x(0),j}$ and $b_{x(0),j,k}$ are defined appropriately.

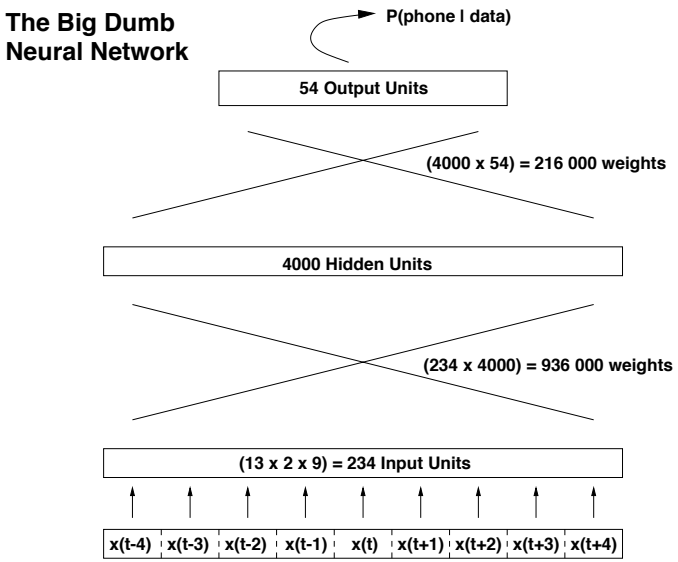
MLPs and backprop (mid-late 1980s)





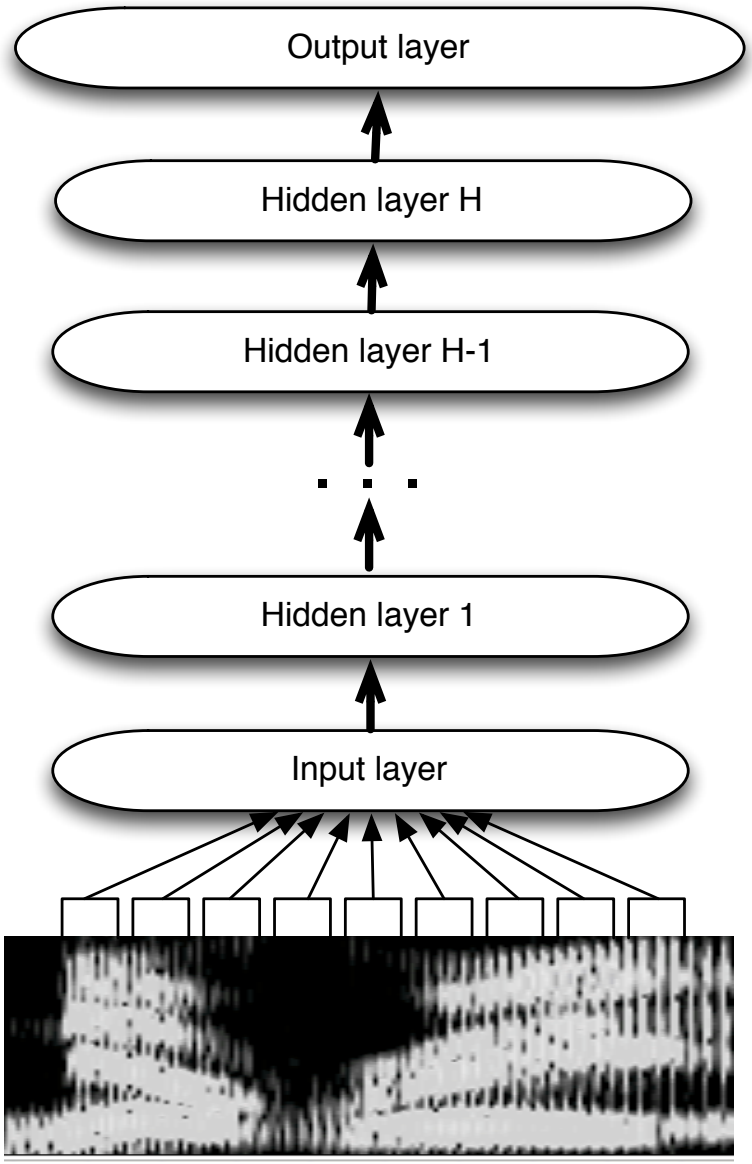
- Train multiple layers of non-linear units – nested non-linear functions ($f(g(\ldots))$)
  - Powerful feature detectors
  - Estimate class-conditional posterior probabilities
  - Theorem: network with a single hidden layer can approximate any function

Neural network acoustic models (1990s)



- Similar performance to context-dependent HMM/GMM systems on read speech (WSJ)
- More errors on more complex tasks (broadcast news, conversational telephone speech)
- Lacked effective approaches for context-dependent modelling and adaptation
- Training not easily parallelisable
  - Slower experimental turnaround
  - Smaller, less complex systems
- "Second NN winter" – focus on GMMs, support vector machines, conditional random fields, ...

(Deep) neural network acoustic models (2010s)



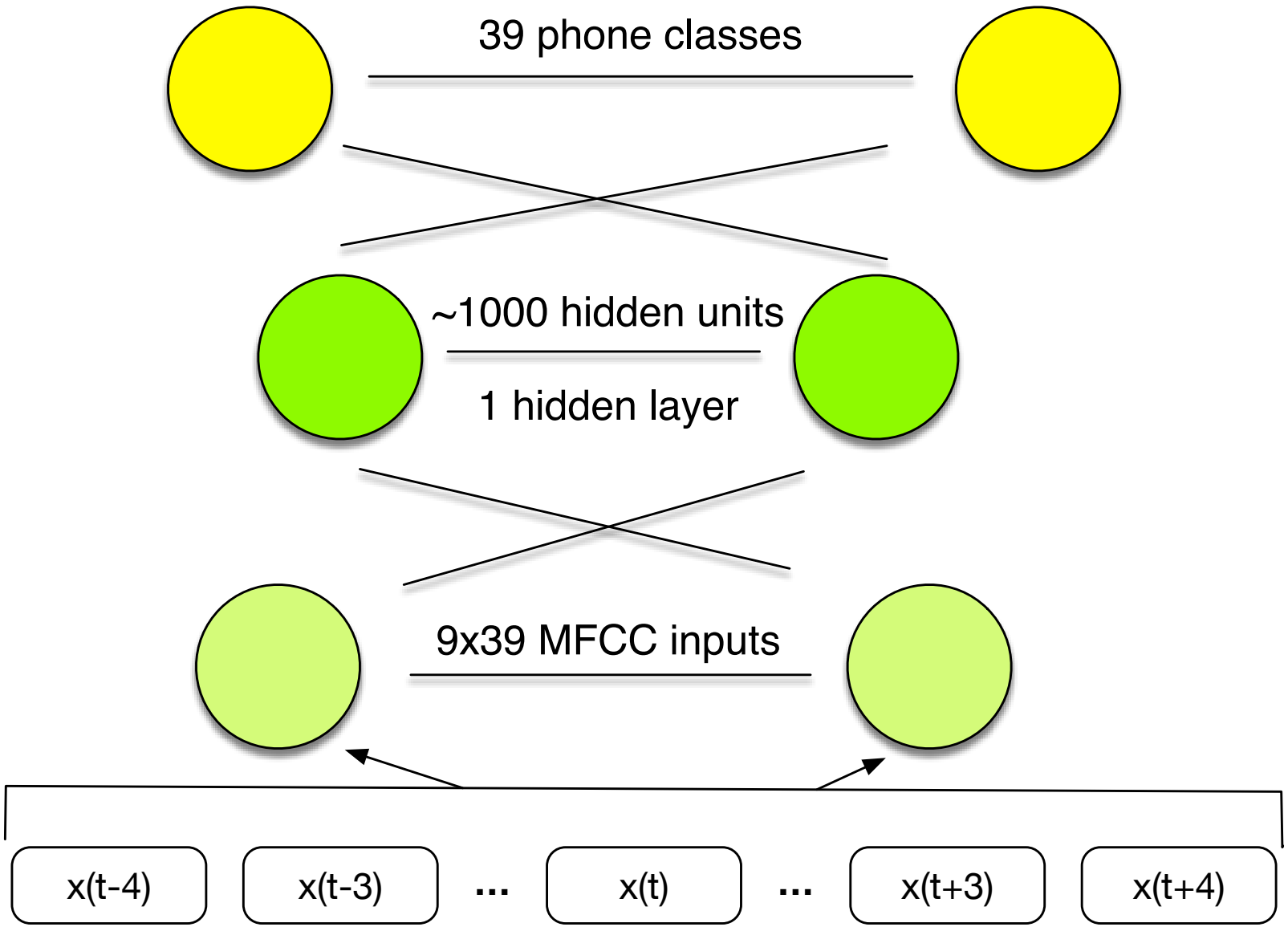
- **Multiple frames of input context:** Input layer takes several consecutive frames of acoustic features
- **One big network for everything, rather than multiple HMMs:** Output layer corresponds to classes (e.g. phones, HMM states)
- **Potential deep structure:** Multiple non-linear hidden layers between input and output

Hybrid NN/HMM Systems

Using neural networks for acoustic modelling

- **Hybrid** NN/HMM systems
  - Basic idea: in an HMM, replace the GMMs used to estimate output pdfs with the outputs of neural networks
  - NN outputs correspond to phone classes or HMM states, and estimate the probability of the class
- **Tandem** features
  - Use NN probability estimates as an additional input feature stream in an HMM/GMM system (posteriograms / bottleneck features)

Neural networks 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, usually reduced to 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 and posterior probabilities

- Moving from frame classification to phone recognition
  - Interpret the NN outputs as frame scores
  - Use dynamic programming (Viterbi) to compute the most likely sequence of phone segments
- **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:

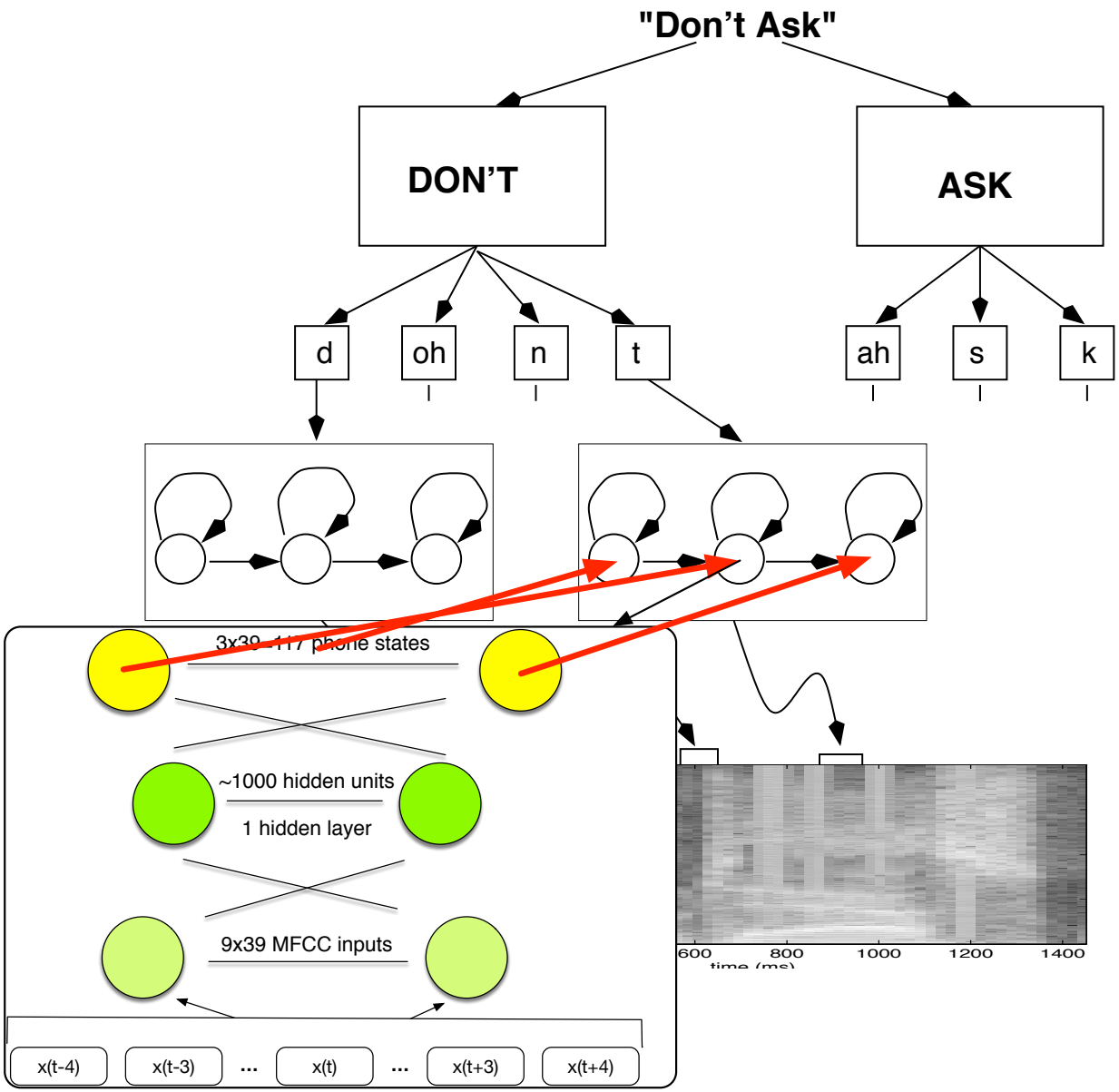
$$\frac{P(q|\mathbf{x})}{p(q)} = \frac{p(\mathbf{x}|q)}{p(\mathbf{x})}$$

- 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



Utterance

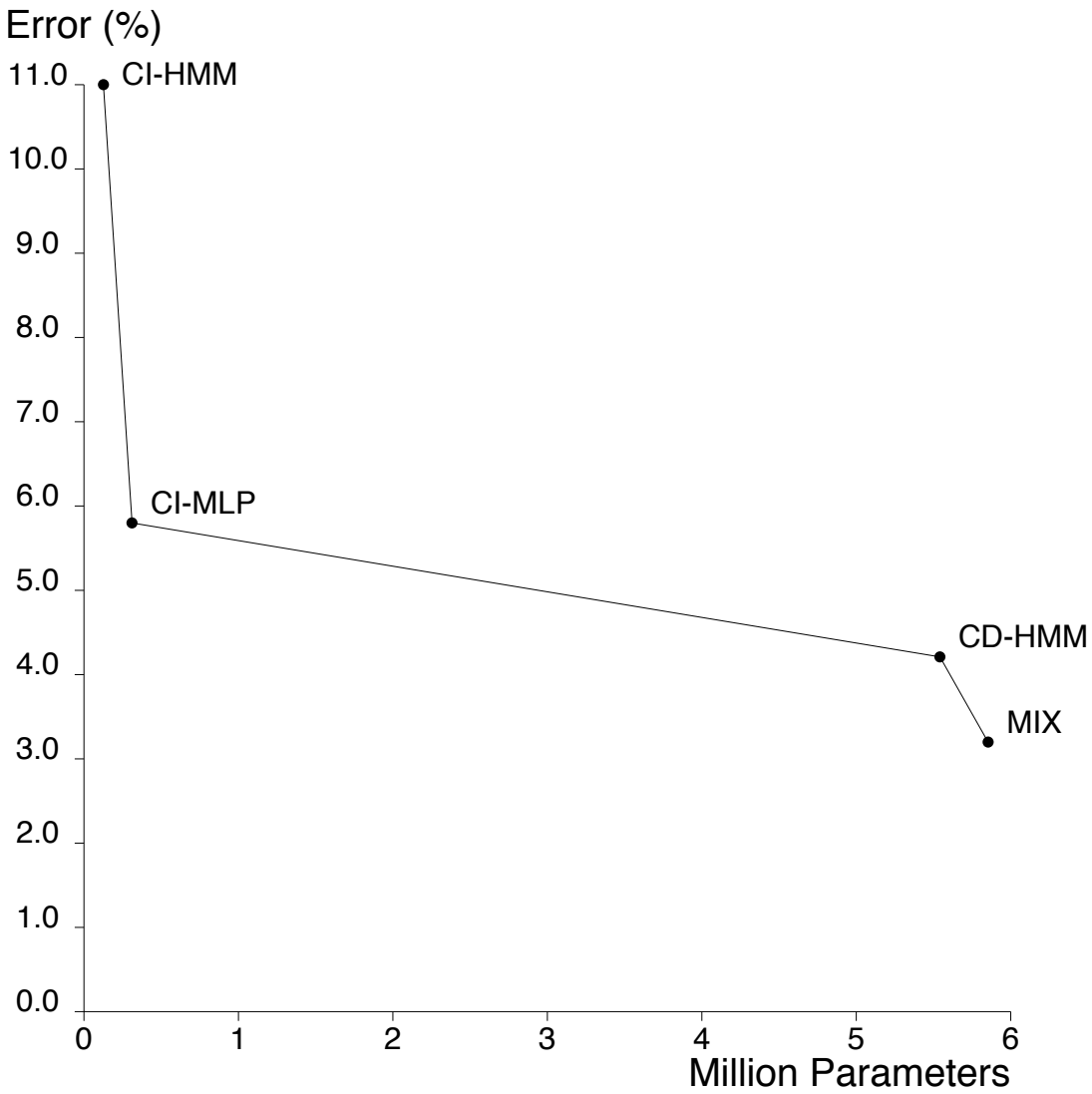
Word

Subword (phone)

Acoustic model (HMM)

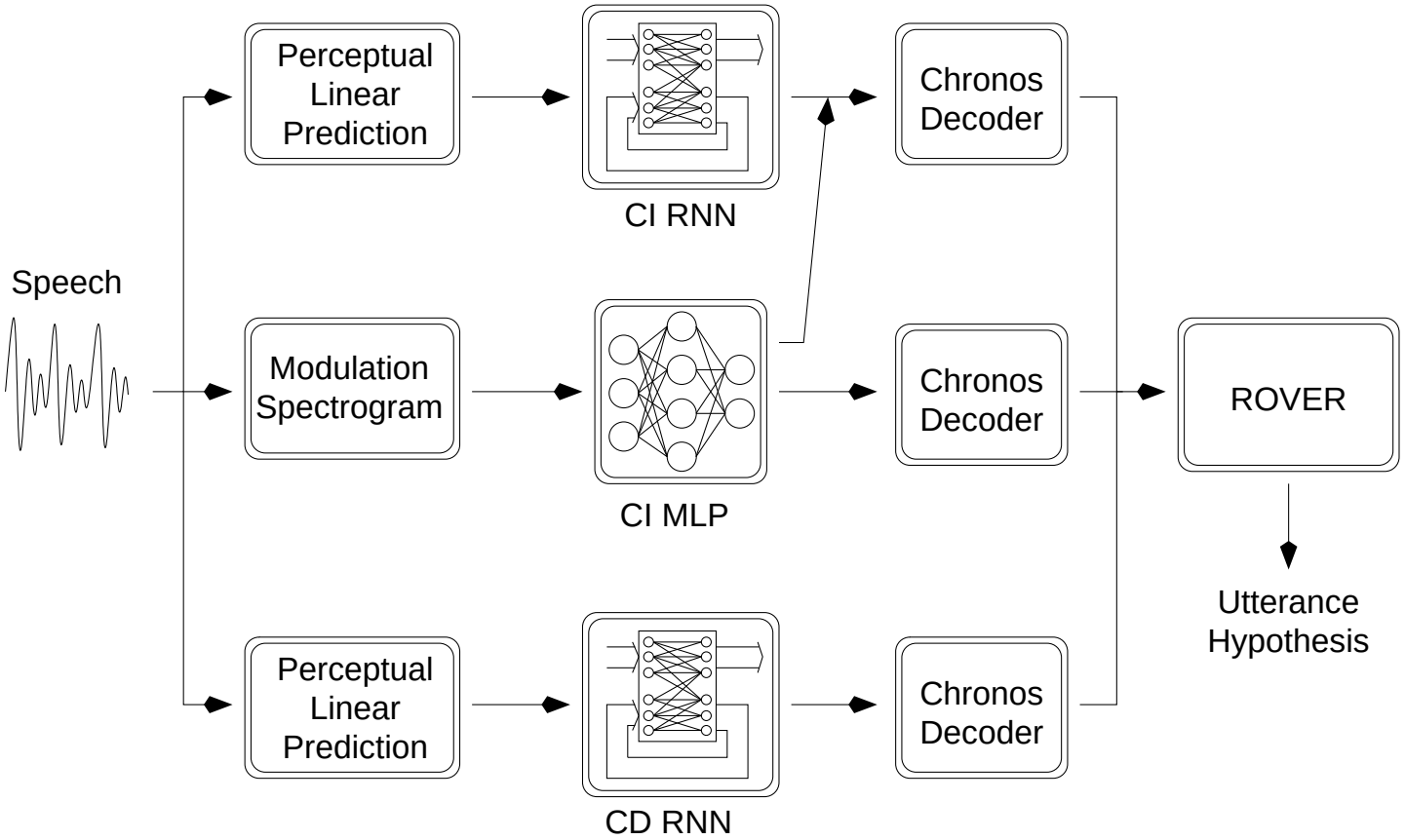
Speech Acoustics

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 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

Reading

- M Nielsen, *Neural Networks and Deep Learning*, http://neuralnetworksanddeeplearning.com
- N Morgan and H Bourlard (May 1995). "Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach", *IEEE Signal Processing Magazine*, **12**(3), 24–42. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=382443

Next lecture: Deep neural network acoustic models