Deep Learning for Automatic Speech Recognition – Part I

Xiaodong Cui

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Fall, 2018

Outline

- A brief history of automatic speech recognition
- Speech production model
- Fundamentals of automatic speech recognition
- Context-dependent deep neural networks

Applications of Speech Technologies

- Speech recognition
- Speech synthesis
- Voice conversion
- Speech enhancement
- Speech coding
- Spoken term detection
- Speaker recognition/verification
- Speech-to-speech translation
- Dialogue systems
- ......

Some ASR Terminology

- Speaker dependent (SD) vs. speaker independent (SI)
- Isolated word recognition vs. continuous speech recognition
- Large-vocabulary continuous speech recognition (LVCSR)
  - naturally speaking style
  - $>$ 1000 words historically but way more nowadays (e.g. 30K - 50K, some may reach 100K)
- Speaker adaptation
  - supervised
  - unsupervised
- Speech input and channel
  - close-talking microphone
  - far-field microphone
  - microphone array
  - codec

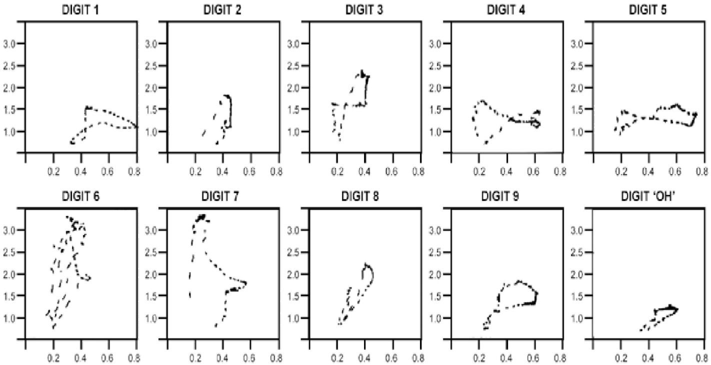
Four Generations of ASR

Roughly four generations:

- **1st generation (1950s-1960s)**:
  Explorative work based on acoustic-phonetics
- **2nd generation (1960s-1970s)**:
  ASR based on template matching
- **3rd generation (1970s-2000s)**:
  ASR based on rigorous statistical modeling
- **4th generation (late 2000s-present)**:
  ASR based on deep learning

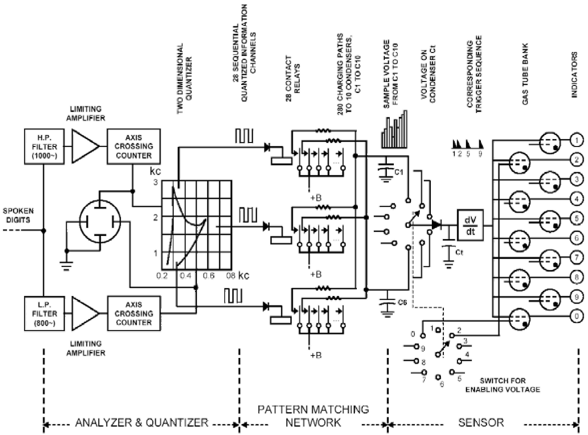
*adapted from S. Furui, "History and development of Speech Recognition."

1st Generation: Early Attempts (1)



K. H. Davis, R. Biddulph, and S. Balashek, "Automatic Recognition of Spoken Digits," J. Acoust. Soc. Am., vol 24, no. 6, pp. 627-642, 1952.

1st Generation: Early Attempts (2)

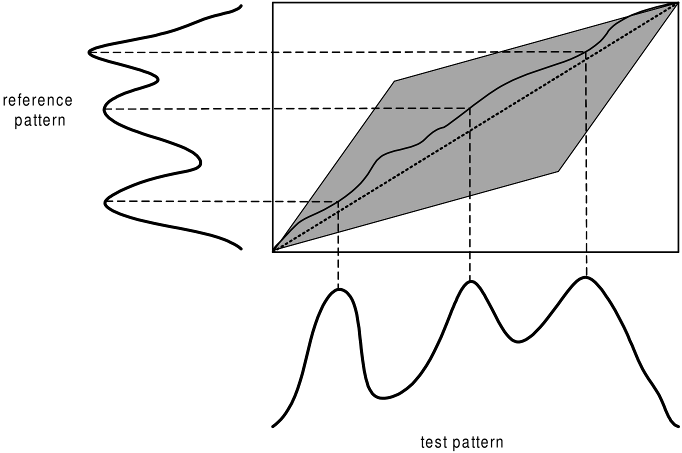


K. H. Davis, R. Biddulph, and S. Balashek, “Automatic Recognition of Spoken Digits," J. Acoust. Soc. Am., vol 24, no. 6, pp. 627-642, 1952.

2nd Generation: Template Matching

- Linear predictive coding (LPC)
  - formulated independently by Atal and Itakura
  - an effective way for estimation of vocal tract response
- Dynamic programming
  - widely known as dynamic time warping (DTW) in speech community
  - deal with non-uniformity of two patterns
  - first proposed by Vintsyuk from former Soviet Union which was not known to western countries until 1980s.
  - Sakoe and Chiba from Japan independently proposed it in late 1970s.
- Isolated-word or connected-word recognition based on DTW and LPC (and its variants) under appropriate distance measure.

An example illustrating DTW



3rd Generation: Hidden Markov Models

- Switched from template-based approaches to rigorous statistical modeling
- Path of HMMs becoming the dominant approach in ASR
  - Earliest research dated back to late 1960s by L. E. Baum and his colleagues at Institute for Defense Analyses (IDA)
  - James Baker followed it up at CMU when he was a Ph.D in 1970s
  - James and Janet Baker joined IBM and worked with Fred Jelinek on using HMMs for speech recognition in 1970s
  - Workshop on HMMs was held by IDA in 1980 which resulted in a so-called "The Blue Book" with the title "Applications of Hidden Markov Models to Text and Speech". But the book was never widely distributed.
  - A series of papers on the HMM methodology was published after the IDA workshop in the next few years including the well-known IEEE proceedings paper "A tutorial on hidden Markov models and selected applications in speech recognition" in 1989.
  - HMMs have since become the dominant approach for speech recognition.

A Unified Speech Recognition Model

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}$$

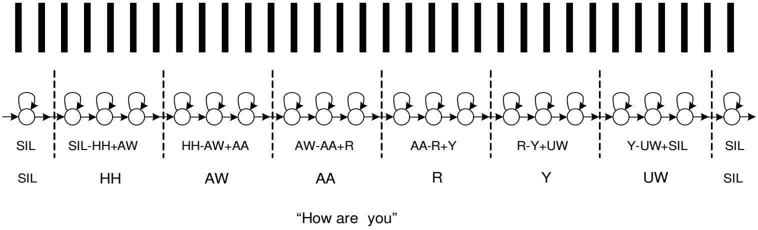
$X = \{x_1, x_2, \cdots, x_m\}$ is a sequence of speech features

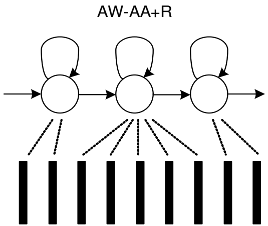
$W = \{w_1, w_2, \cdots, w_n\}$ is a sequence of words

$P(W) = P(w_1, w_2, \cdots, w_n)$ gives the probability of the sequence of the words – referred to as language model (LM)

$P(X|W) = P(x_1, x_2, \cdots, x_m | w_1, w_2, \cdots, w_n)$ gives the probability of the sequence of speech features given the word sequence – referred to as acoustic model (AM)

An Illustrative Example of HMMs





Two LVCSR Developments at IBM and Bell Labs

- IBM
  - focused on dictation systems
  - interested in seeking a probabilistic structure of the language model with a large vocabulary
  - n-grams

$$
\begin{aligned}
P(W) &= P(w_1 w_2 \cdots w_n) \\
&= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \cdots P(w_n|w_1 w_2 \cdots w_{n-1}) \\
&= P(w_1)P(w_2|w_1)P(w_3|w_2) \cdots P(w_n|w_{n-1})
\end{aligned}
$$

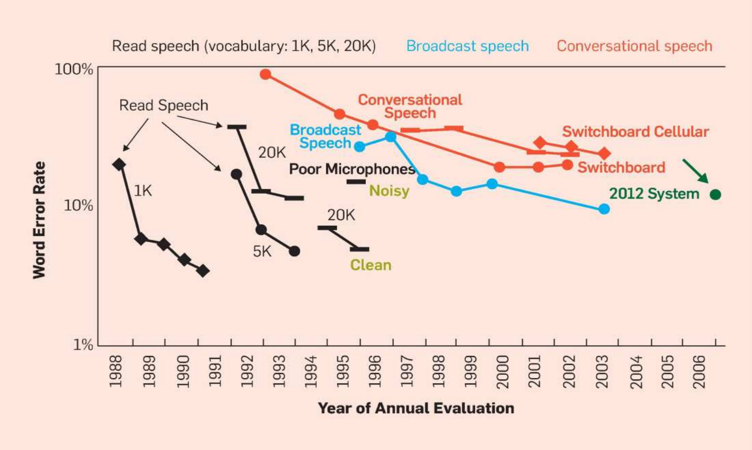
- Bell Labs
  - focused on voice dialing and command and control for call routing
  - interested in speaker independent systems that could handle acoustic variability from a large number of different speakers
  - Gaussian mixture models (GMMs) for state observation distribution

$$
P(x|s) = \sum_i c_i \, \mathcal{N}(x; \mu_i, \Sigma_i)
$$

3rd Generation: Further Improvements

- GMM-HMM acoustic model with n-gram language model have become the "standard" LVCSR recipe.
- Significant improvements in 1990s and 2000s.
  - Speaker adaptation/normalization
    - vocal tract length normalization (VTLN)
    - maximum likelihood linear regression (MLLR)
    - speaker adaptive training (SAT)
  - Noise robustness
    - parallel model combination (PMC)
    - vector Taylor series (VTS)
  - Discriminative training
    - minimum classification error (MCE)
    - maximum mutual information (MMI)
    - minimum phone error (MPE)
    - large margin
  - ......

Progresses Made Before the Advent of Deep Learning



*X. Huang, J. Baker and R. Reddy, "A historical perspective of speech recognition."

4th Generation: Deep Learning

- G. E. Hinton, S. Osindero and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, 18, pp 1527-1554, 2006.
  - the ground-breaking paper on deep learning.
  - layer-wise pre-training in an unsupervised fashion
  - fine-tune afterwards with supervised training
- Changed people's mindset that deep neural networks are not good and can never be trained
- Microsoft, Goolge and IBM around 2011 and 2012 all reported significant improvements over their then state-of-the-art GMM-HMM-based ASR systems.

4th Generation: Deep Learning

- Deep acoustic modeling
  - deep feedforward neural networks (DNN, CNN, $\cdots$)
  - deep recurrent neural networks (LSTM, GRU, $\cdots$)
  - end-to-end neural networks
- Deep language modeling
  - deep feedforward neural networks (DNN, CNN)
  - deep recurrent neural networks (RNNs, LSTM $\cdots$)
  - word embedding

DNN-HMM vs. GMM-HMM

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

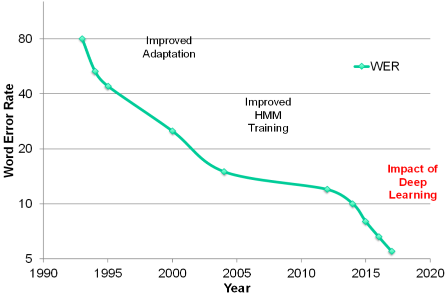
TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

*G. Hinton et. al., "Deep neural networks for acoustic modeling in speech recognition – the shared views of four research groups."

Impact of Deep Learning on SWB2000

Switchboard database

- a popular public benchmark in speech recognition community
- human-human landline telephone conversations on directed topics
- 300 hours/ 2000 hour



Progresses Made at IBM on SWB2000

Model	Word Error Rate	Described in IBM Publication
1. CNN	**10.4**	2013
2. RNN	9.9	2014, 2015
3. VGG	9.4	2016
4. RNN+VGG+LSTM	8.6	2016
5. (4) +More Ngrams+ModelM	7.0	2009, 2016
6. (4) +More Ngrams+ModelM+NNLM	6.6	2007, 2009, 2016
7. Adversarial Learning + Resnet + LSTM	6.7	2017
8. (7) + (6) + LSTM LMs + Wavenet LM	5.5	2017

*estimate of human performance: 5.1%

Achieving "Human Parity" in ASR

IBM Developer
Watson | Watson services | Dev tools | Documentation | Forum | Blog | Try free in Bluemix

Announcement

IBM Watson announces breakthrough in Conversational Speech Transcription

Michael Picheny
Published on May 26, 2015 / Updated on May 31, 2015

→ **WER = 8.0, 05/2015**

IBM Developer
Watson | Watson services | Dev tools | Documentation | Forum | Blog | Try free in Bluemix

General

Recent Advances in Conversational Speech Recognition

George Saon
Published on April 28, 2016 / Updated on April 29, 2016

→ **WER = 6.9, 04/2016**

ZDNet

MUST READ: Amazon announces everything Alexa but the kitchen sink

Microsoft's newest milestone? World's lowest error rate in speech recognition

Microsoft has leapfrogged IBM to claim a significant test result in the quest for machines to understand speech better than humans.

By Liam Tung | September 14, 2016 -- 11:41 GMT (04:41 PDT) | Topic: Artificial Intelligence

→ **WER = 6.3, 09/2016**
WER = 5.9, 10/2016

ZDNet

MUST READ: Amazon announces everything Alexa but the kitchen sink

IBM vs Microsoft: 'Human parity' speech recognition record changes hands again

Artificial intelligence can do many things better than humans, but speech recognition isn't one of them. Yet.

By Liam Tung | March 13, 2017 -- 13:42 GMT (06:42 PDT) | Topic: Artificial Intelligence

→ **WER = 5.5, 03/2017**

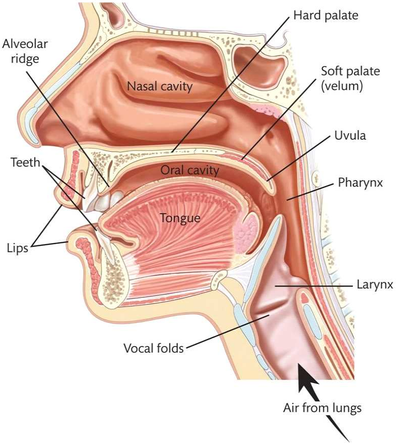
LANGUAGE MODELING WITH HIGHWAY LSTM

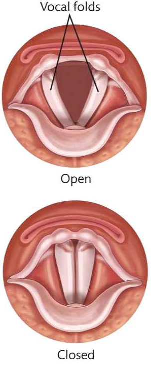
Gakuto Kurata[1], *Bhuvana Ramabhadran*[2], *George Saon*[2], *Abhinav Sethy*[2]

IBM Research AI
gakuto@jp.ibm.com, {bhuvana,gsaon,asethy}@us.ibm.com

→ **WER = 5.1, 09/2017**

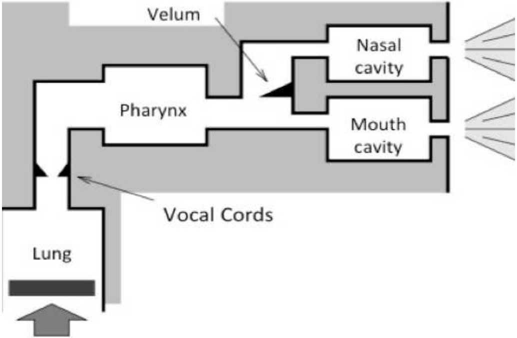
Speech Production Model





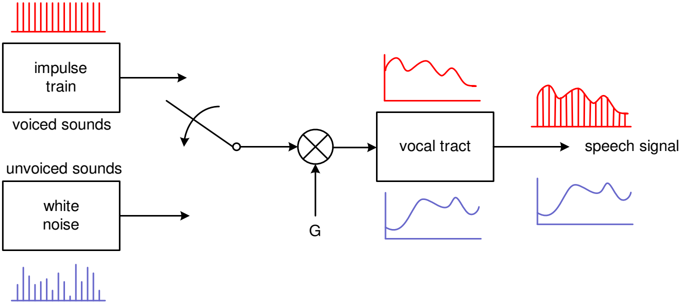
*from internet

Speech Production Model



*L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition".

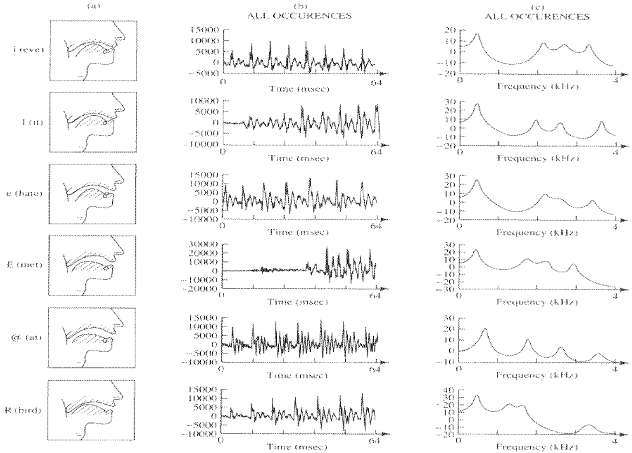
Speech Production Model



time domain: $s_t = e(t) * v(t)$

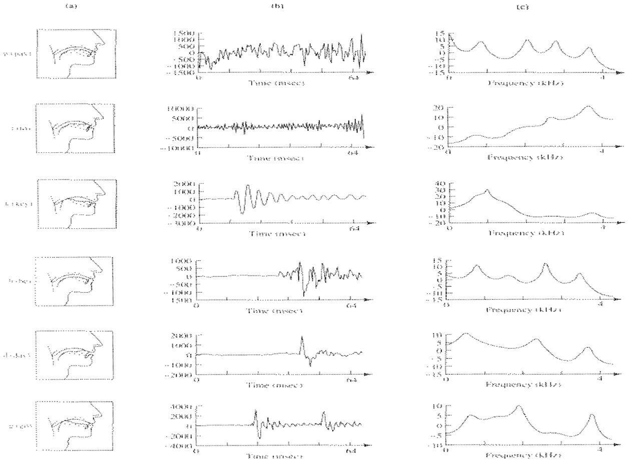
frequency domain: $S(\omega) = E(\omega)V(\omega)$

Speech Production Model



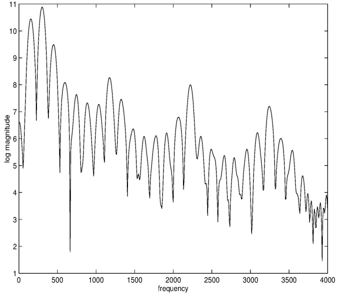
*J. Picone, "Fundamentals of speech recognition: a short course".

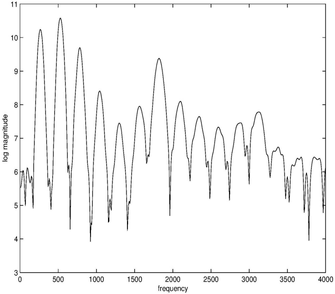
Speech Production Model



*J. Picone, "Fundamentals of speech recognition: a short course".

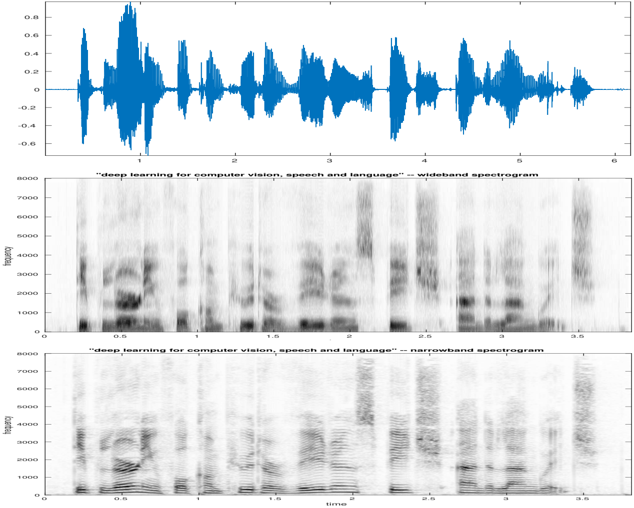
An example of speech spectra



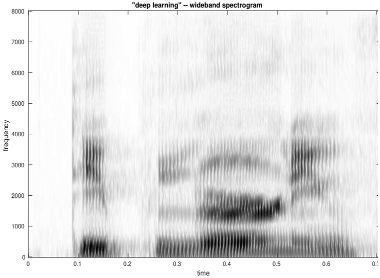


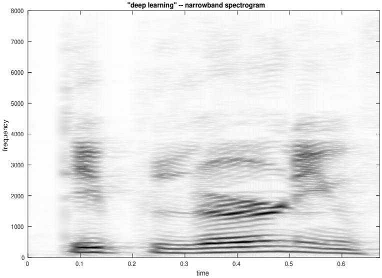
- /uw/ sound from an adult male and a boy
- pitch
- vocal tract

Waveforms and Spectrograms

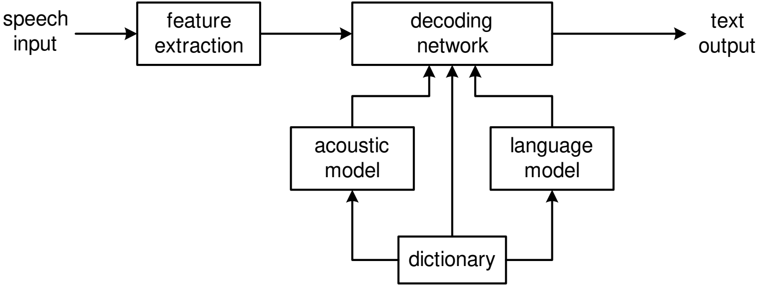


Wideband and Narrowband Spectrograms



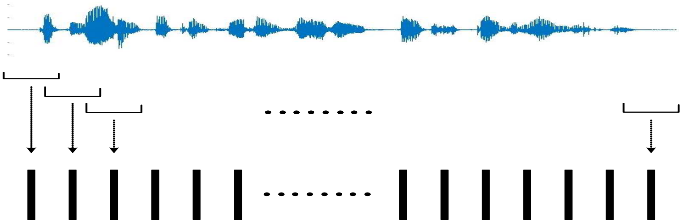


A Quick Walkthrough of GMM-HMM Based Speech Recognition



- training
- decoding

Feature Extraction



- Frame window length $\sim$20ms with a shift of $\sim$10ms
- Commonly used hand-crafted features
  - Linear Predictive Cepstral Coefficients (LPCCs)
  - Mel-frequency Cepstral Coefficients (MFCCs)
  - Perceptual Linear Predictive (PLP) analysis
  - Mel-frequency Filter banks (FBanks) (widely used in DNNs/CNNs)

Cepstral Analysis

Why cepstral analysis? deconvolution!

$$\longrightarrow \boxed{\text{windowing}} \longrightarrow \boxed{\text{DFT}} \longrightarrow \boxed{\log|\cdot|} \longrightarrow \boxed{\text{IDFT}} \longrightarrow \boxed{\text{liftering}} \longrightarrow$$

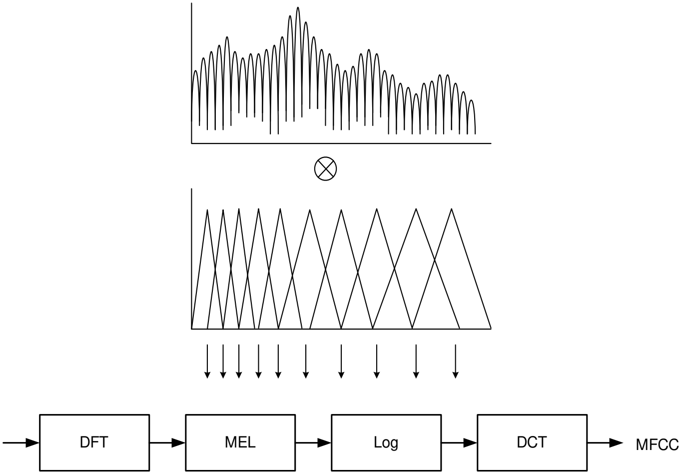
$$|S(\omega)| = |E(\omega)| \cdot |V(\omega)|$$

$$\log|S(\omega)| = \log|E(\omega)| + \log|V(\omega)|$$

$$c(n) = \mathsf{IDFT}(\log|S(\omega)|) = \mathsf{IDFT}(\log|E(\omega)| + \log|V(\omega)|)$$

- cepstrum, quefrency and liftering
- vocal tract components are represented by the slowly varying components concentrated at lower quefrency
- excitation components are represented by the fast varying components concentrated at the higher quefrency

Mel-Frequency Filter Bank and MFCC



Acoustic Units and Dictionary

- Dictionary

  HOW HH AW

  ARE AA R

  YOU Y UW

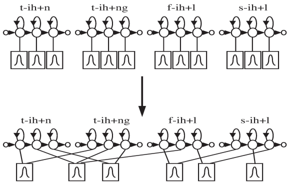
  ......
- Context-Independent (CI) phonemes

  HH AW AA R Y UW $\cdots$
- Context-dependent (CD) phonemes
  - coarticulation
  - phonemes are different if they have different left and right contexts

    $$P_l - P_c + P_r$$
  - e.g. HH-AW+AA, P-AW+AA, HH-AW+B are different CD phonemes
  - each CD phoneme is model by an HMM.
- Other acoustic units
  - syllables, words, $\cdots$
  - a tradeoff between modeling accuracy and data coverage

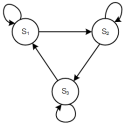
Context-Dependent Phonemes

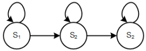
- Advantages:
  - modeling subtle acoustic characteristics given the acoustic contexts
- Disadvantages:
  - giving rise to a substantial number of CD phonemes
  - e.g. $45^3 = 91125$, $45^5 \approx 1.8 \times 10^8$
- Solution:
  - parameter tying (vowels, stops, fricatives, nasals...)
  - widely used for targets of DNN/CNN systems



*after HTK.

GMM-HMM: Mathematical Formulation





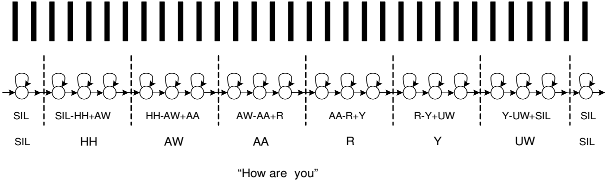
$$\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$$

- State transition probability $\mathbf{A}$: $a_{ij} = P(s_{t+1} = j | s_t = i)$
- State observation PDF $\mathbf{B}$: $b_i(o_t) = p(o_t | s_t = i)$
- Initial state distribution $\boldsymbol{\pi}$: $\pi_i = P(s_1 = i)$

HMM is an extension of Markov chain

- a doubly embedded stochastic process
- an underlying stochastic process which is not directly observable
- another observable stochastic process generated from the hidden stochastic process

GMM-HMM: Mathematical Formulation



Three fundamental problems for HMMs

- Given the observed sequence $O = \{o_1, \cdots o_\mathsf{T}\}$ and a model $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, how to evaluate the probability of the observation sequence $P(O|\lambda) = \sum_S P(O, S|\lambda)$?
- How do we adjust the model parameters $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ to maximize $P(O|\lambda)$?
- Given the observed sequence $O = \{o_1, \cdots o_\mathsf{T}\}$ and the model $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, how to choose the most likely state sequence $S = \{s_1, \cdots s_\mathsf{T}\}$?

GMM-HMM: Acoustic Model Training

How to compute the feature sequence likelihood given the acoustic model?
The forward-backward algorithm

- Forward Computation: $\alpha_t(i) = P(o_1 \cdots o_t, s_t = i|\lambda)$
  - Initialization
  $$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \leq i \leq M$$
  - Induction
  $$\alpha_{t+1}(j) = \sum_{i=1}^{M} \alpha_t(i) a_{ij} b_j(o_{t+1}), \quad 1 \leq j \leq M, \quad 1 \leq t \leq T-1$$
  - Termination
  $$P(\mathcal{O}|\lambda) = \sum_{i=1}^{M} \alpha_T(i)$$
- Backward Computation: $\beta_t(i) = P(o_{t+1} \cdots o_T | s_t = i, \lambda)$
  - Initialization
  $$\beta_T(i) = 1, \qquad 1 \leq i \leq M$$
  - Induction
  $$\beta_t(i) = \sum_{i=1}^{M} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad 1 \leq j \leq M, \quad T-1 \geq t \geq 1$$
- Using both forward and backward variables: $P(\mathcal{O}|\lambda) = \sum_{i=1}^{M} \alpha_t(i)\beta_t(i)$

GMM-HMM: Acoustic Model Training

How to estimate model parameters λ given the data?

- Given the feature sequence $O = \{o_1, \cdots o_{\mathsf{T}}\}$

$$\lambda^* = \operatorname*{argmax}_{\lambda} P(O|\lambda)$$

where GMM is used for $\mathbf{B}$:

$$P(x|s) = \sum_i c_i \, \mathcal{N}(x; \mu_i, \Sigma_i)$$

- the Expectation-Maximization (EM) algorithm (also known as Baum-Welch algorithm)

$$c_{ik} = \frac{\sum_t \gamma_{ik}(t)}{\sum_t \gamma_i(t)}$$
$$\mu_{ik} = \frac{\sum_t \gamma_{ik}(t) o_t}{\sum_t \gamma_{ik}(t)}, \quad \Sigma_{ik} = \frac{\sum_t \gamma_{ik}(t)(o_t - \mu_{ik})(o_t - \mu_{ik})^T}{\sum_t \gamma_{ik}(t)}$$

where

$$\gamma_t(i) = P(s_{\mathsf{t}} = i|\mathcal{O}, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{M} \alpha_t(i)\beta_t(i)}$$
$$\gamma_{ik}(t) = P(s_t = i, \zeta_t = k|O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(o_{\mathsf{t}+1})\beta_{t+1}(j)}{\sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_t(i) a_{ij} b_j(o_{\mathsf{t}+1})\beta_{t+1}(j)}$$

GMM-HMM: Language Model Training

- n-gram:
  - Approximate the conditional probability with n history words

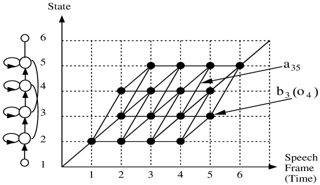
$$P(w_1, w_2, \cdots, w_m) \approx \prod_{i=1}^{m} P(w_i | w_{i-n+1}, \cdots, w_{i-1})$$

  - Counting events on context using training data

$$P(w_i | w_{i-n+1}, \cdots, w_{i-1}) = \frac{C(w_{i-n+1}, \cdots, w_{i-1}, w_i)}{C(w_{i-n+1}, \cdots, w_{i-1})}$$

- unigram, bigram, trigram, 4-grams, $\cdots$
- data sparsity issue
- back-off strategy and interpolation

GMM-HMM: Decoding



- How to find the path of a feature sequence that gives the maximum likelihood given the model?
  - dynamic programming (Viterbi decoding)
  - let $\phi_j(t)$ represent the maximum likelihood of observing partial sequence from o_1 to o_t and being in state j at time t

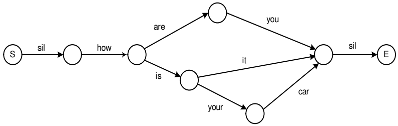
$$\phi_j(t) = \max_{s_1, s_2, \cdots s_{t-1}} P(s_1, s_2, \cdots s_{t-1}, s_t = j, o_1, o_2, \cdots o_t | \lambda)$$

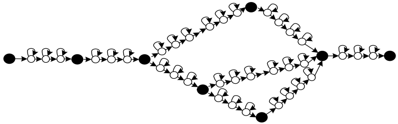
by induction,

$$\phi_j(t) = \max_i \{\phi_i(t-1) a_{ij}\} b_j(o_t)$$

*after HTK.

GMM-HMM: Decoding

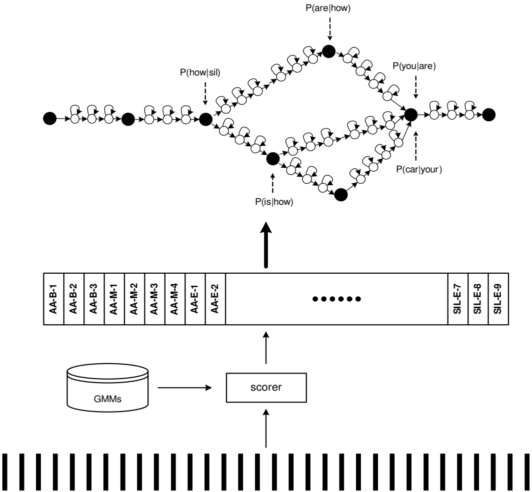




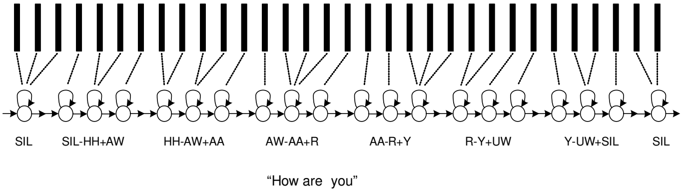
- Inject language model
- Inject dictionary
- Inject CD-HMMs with each CD-HMM state having a GMM distribution
- Compute (log-)likelihood of each feature vector in each CD-HMM state

$$P(x|s) = \sum_i c_i \, \mathcal{N}(x; \mu_i, \Sigma_i)$$

GMM-HMM: Decoding

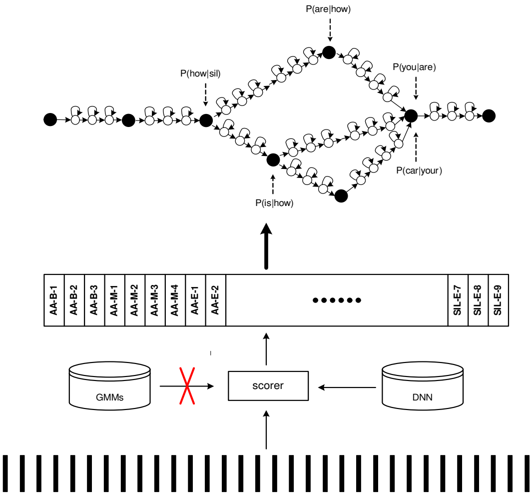


GMM-HMM: Forced Alignment

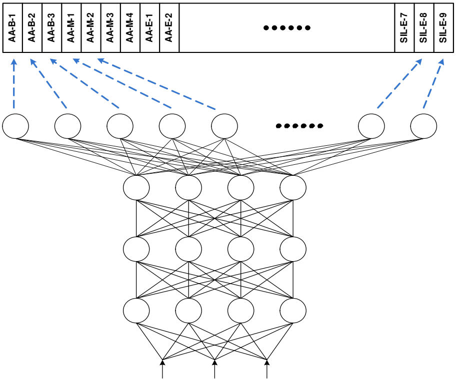


- Given the text label, how to find the best underlying state sequence?
  - same as decoding except the label is known
  - Viterbi algorithm (again)
- Often referred to as Viterbi alignments in speech community
- Widely used in deep learning ASR to generate the target labels for the training data

Context-Dependent DNNs

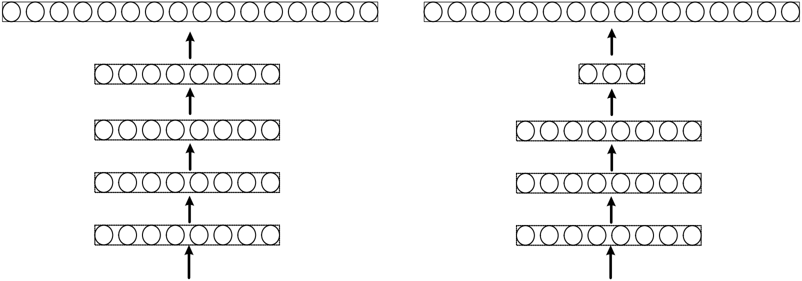


Context-Dependent DNNs



$$P(o|s) = \frac{p(s|o)p(o)}{p(s)} \propto \frac{p(s|o)}{p(s)}$$

Two Families of DNN-HMM Acoustic Modeling



- Hybrid systems
  - directly connected to HMMs for acoustic modeling
- Bottleneck tandem systems
  - used as feature extractors
  - bottleneck features extracted can be used to train GMM-HMMs or DNN-HMMs

Modeling with Hidden Variables

- Hidden variables (or latent variables) are crucial in acoustic modeling (also true for computer vision and NLP)
  - hidden state or phone sequence in acoustic modeling
  - hidden speaker transformation in speaker adaptation
  - latent topic and word distribution in Latent Dirichlet allocation in NLP
- It reflects your belief how a system works (internal working mechanism)
- Deep neural networks
  - using a large number of hidden variables
  - hidden variables are organized in a hierarchical fashion
  - usually lack of straightforward interpretation (the well-known interpretability issue)

DNN-HMMs: An Typical Training Recipe

- Preparation
  - 40-dim FBank features with ± 4 adjacent frames (input dim = 40x9)
  - use an existing model to generate alignments which are then converted to 1-hot targets from each frame
  - create training and validation data sets
  - estimate priors $p(s)$ of CD states
- Training
  - set DNN configuration (multiple hidden layers, softmax output layers and cross-entropy loss function)
  - initialization
  - optimization based on back-prop using SGD on the training set while monitoring loss on the validation set
- Test
  - push input features from test utterances through the DNN acoustic model to get their posteriors
  - convert posteriors to likelihoods
  - Viterbi decoding on the decoding network
  - measure the word error rate

Training A Hybrid DNN-HMM System