\*If you’re going to use this to study, please contribute by asking adding to the doc or asking questions! We’re here to support each other :’)

**Midterm Structure:**

* 8-10 Qs (each 10-15 subparts/pts?)

**Eqns you REALLY should know**

* Joint p(x, w) = p(x|w)\*p(w)
* Bayes p(X|W) = p(W|X) P(X) [/ P(W)] ← don’t really care about denominator (only used to normalize)
* Be able to map the problems that we have to these eqns!

More review sess stuff:

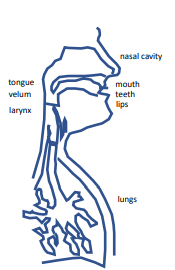
* Spectrograms:
  + Consonants tend to look like white noise
  + Vowels have more discrete signals
* **Forced alignment** = max probable sequence of states (“viterbi path”)

HLT Midterm 1

## 2. Auditory System

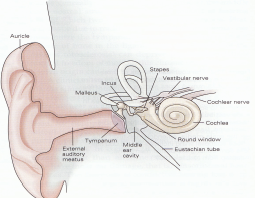
Speech - lots of features to classify sounds (e.g. loudness, timbre)

Human vocal tract

 Each part = means for generating diff sounds/features

**McGurk effect** - top down processing (visual inputs affect how you perceive a sound)

**Physiology of hearing**:



Eardrum → hammer/anvil/stirrup → tectorial and basilar membrane → inner/outer hair cell → auditory nerve

* Basilar membrane separates freqs mechanically
* Basilar membrane movement → hair cells bend → electrical pulses
* Inner hair cell - (information) connected to basilar membrane (~40 hairs/cell)
* Outer hair cell - (governs cochlear mechanics - cochlear amplifier: pos feedback) connected to both tectorial and basilar membranes (~140 hairs/cell)

Speech signal → (bottom-up connections) → brain → (top-down connections - feedback) → cochlea

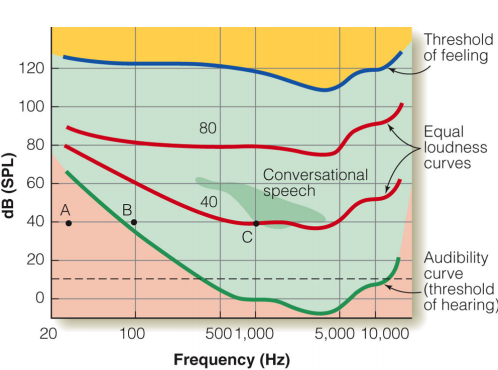
* Brain is slow - ISI ~100 ms (sound frequencies can be much faster)
* Massive ↑ in # neurons from lower processing levels to cortex \*\*can anyone explain the significance of these three facts? (I believe that Hynek mentioned this just for explaining the hardness of the speech perceptual task.)
* Decrease in avg spiking rates from periphery to cortex
* Spikes in cortex are sparse (<5% of cortical neurons active at any moment)

**Tonotopy**:

* Diff freqs excite diff parts of the cochlea due to freq separation by the basilar membrane in cochlea (base [entry] = thin, high freqs; apex [end] = thick, low freqs)
* These diff freqs then map to diff parts of auditory cortex

Psychophysics - Sensitivity of Hearing:

* Diff people req diff loudness (dB) to be able to perceive sounds at certain freqs - Nonuniform! Sensitive to diff loudness/freqs



* **Equal loudness curves**: perceive the sounds at same freq to be the same loudness even though diff dBs
* **Threshold of hearing**: dB req to perceive the sound at that freq
  + Grow older → lose hearing w/ higher freqs

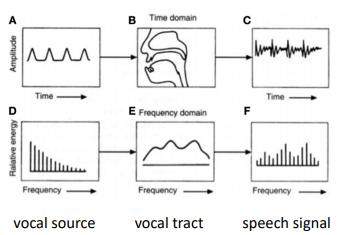
\*\*What’s simultaneous masking/freq selectivity of hearing? A frequency may have a spillover effect on nearby frequency components. For example, if 2kHz has a large energy, people may not perceive the information of 2050Hz.(MP3 is developed with this phenomenon)

Shape of Vocal tract changes freq:

* Motor control → critical elements (tongue, lips, velum) → shape of whole vocal tract → generates speech signal spectrum (redundant contributions of movements of critical elements in diff freq bands)
* Vocal organs don’t move fast enough (sluggish) → blurs produced speech sounds (modulated)

Models:

* **Carrier nature of speech**: message in movements of vocal tract → modulator → voiced/unvoiced carrier to make tract movements audible → message modulated carrier
* **Linear model of speech production**: source → filter → filtered source signal

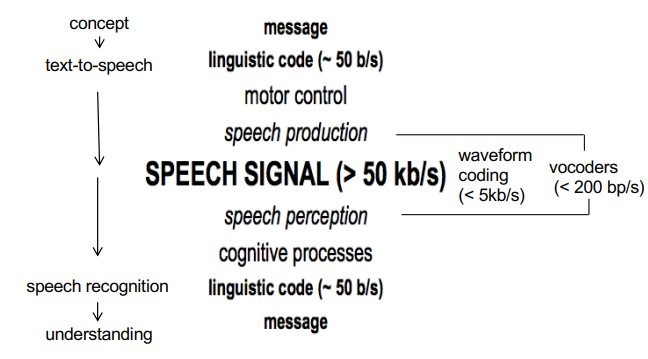
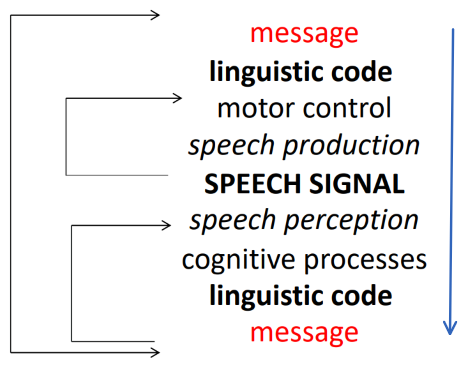


Redundancies:

* Frequency:
  + Production: tract acoustics distributes info to all freqs of the speech spectrum
  + Perception: hearing selectivity allows for decoding information in separate freq bands
* Time:
  + Production: tract sluggishness (coarticulation) distributes info about each speech signal in time
  + Perception: temporal sluggishness of hearing collects info distributed in time

## 3. Speech Basics

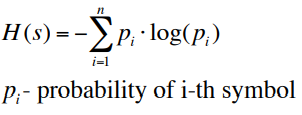
Words = ordered seqs of phonemes; phrases = ordered seqs of words



\* Does anyone understand exactly what this diagram means? I’m not sure what your question is. The right figure is comparing the information content. Waveform coding and vocoders are both methods adopted in speech encoding.

**Entropy** = measure of info in the source

* Property of alphabet (info source)
* Avg amount of info per alphabet symbol



**Phonemes** = perceptually distinct speech sounds that could distinguish one word from another

**Grapheme** = (combos of) letters representing phonemes

Vowels: mouth open; Consonants: mouth not so open

Typical syllable:

* cvc (onset - nucleus - coda)
* cv (onset - nucleus)

Words = ordered seq of speech sounds

* Represent objs, ideas, actions, relationships,... as agreed on by language (society)
* New words constantly invented and old words change meaning
* Learned using interventions and rewards from other humans
* Particular word meanings often depend on context

Word Sequences (sentences, phrases,...)

* Words ordered using **rules of language** (**syntax, grammar**)
* Order carries info

Word predictability: not super predictable! 100% predictable message has no info value

* But speech follows rules (grammar, use of words, word order,...) → somewhat predictable → easier communication

Variability: we want message (signal), but don’t want noise

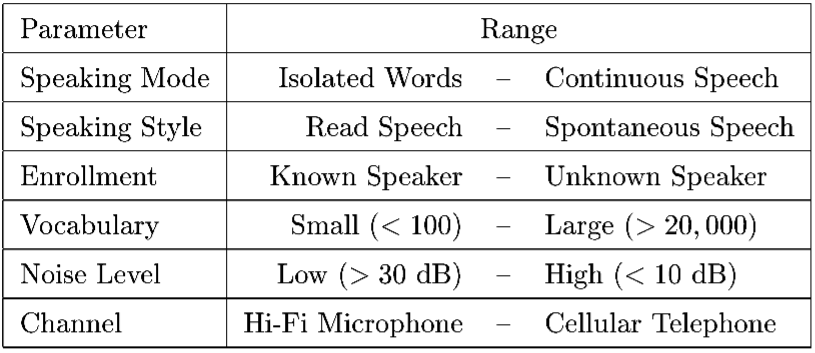
Noise: if we know effect of noise; if we know noise will be but effect unknown; unknown noise w/ unknown effect…

Why Speech?

* Applications: voice recognition ~ keyboards, voice control, etc.
* Spinoffs: Digital signal processing, Seq classification (finance, DNA matching), Image processing techniques
* Problems:
  + Human speaking… distinct, noisy, fillers, interruptions, non-grammatical, dialects, emotions,...
  + Noisy environments
  + Require large amounts of annotated training data
* How to get there? Engineering + life sciences

## 4. Classic speech recognition

**Axes of Characterization**: factors of difficulty of speech recognition



**Speech recognition problem: Convert acoustic signal → seq of words**

* Using conditional probabilities and bayes to find argmax W

Prior Knowledge - linguistics

* **Grammar**: organization of words, phrases, clauses
* **Phonology**: words have canonical pronunciations (// used to indicate the organization of the speech sound)
* **Phonetics**: Phones have characteristic sounds (used to study how the sounds are generated)

Source-Channel Model

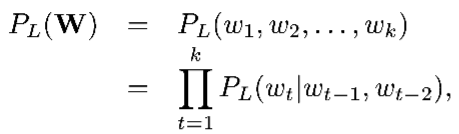
* Given A (speech signal) guess the phonemes/words to create the word W\_hat?

Signal Representation and the Acoustic Processor

* Pitch, formants, vowels, consonants
* **MEL cepstrum** - to represent the “tuning”/nerve cells in ear specific to diff freqs:
  + Get Fourier transform of a window of speech, *bin* the magnitude spectrum into a small number of coeffs, take log of those, decorrelate the feature vector via Cosine transform
* (discussed more in later lecture)

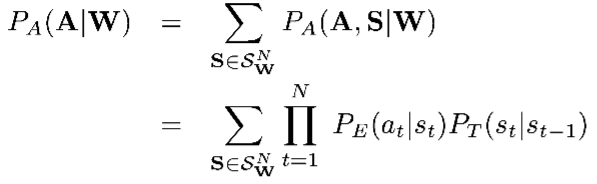
**The Component Problems of Speech Recognition**

1. **Language Modeling**: Assign probabilities to sequences of words in the language. Constructing P(W) given text.



1. **Acoustic Modeling**: Assign probabilities to acoustic realizations of a sequence of words. Constructing P(A|W) given set of acoustic signals and corresponding text.

* A separate model for **each word** **sequence or word** requires too many samples!
* But there are only ~50 phonemes. So we use **acoustic-phonetic models**.
* Construct the model for a **word** by **stringing together models for constituent phone**s.
* This makes P(A|W) a string of **HMM**



1. **Hypothesis Search**: Find the word sequence with maximum a posteriori probability. Argmax W.

* Use graph minimization (finite state transducer), heuristics

**HMMs**

* Markov chain st (state process) and output at
* Each transition has a prob of occurring given being on current state (only depend on current state, nothing before that)

Computational Issues w HMMs

* Exponentially many state seqs
* We don’t know S (the seq of states we go thru)
* Must search over all possible state seqs to compute most likely state seq for acoustic signal A given W

## 6. Speaker recognition

Example Q from Review Sess:

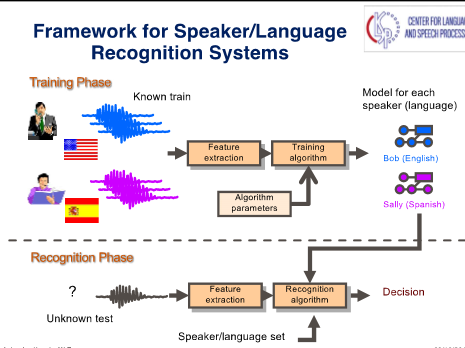
* What would happen if you took recordings from diff speakers and diff rooms and try to come up w/ model for it?
  + Microphones pick up diff room reverberations

Extracting Information from Speech

* Speech recognition: words
* Language recognition: language name
* **Speaker recognition**: speaker name (also applies to matching languages
  + **Closed-set identification**: determine whether a test speaker matches one of a set of known speakers (1-many)
  + **Open-set verification/authentication**: determine whether a test speaker matches a specific speaker (1-1)
    - Unknown speech comes from a large set of unknown speakers
* **Speaker diarization**: who speaks when
  + **Segmentation**: Determine when a speaker change has occurred in speech signal
  + **Clustering**: Group together speech segments corresponding to same speaker
  + \*\* prior speaker info may/may not be available!

Speech Modalities (tasks):

* **Text-dependent**: text spoken by person is known (e.g. fixed or prompted)
  + Strong control over user input
* **Text-independent**: text is unknown (e.g. conversational)
  + More flexible system but also more difficult problem!



Speech signal: **Spectrogram** (freq vs time)

Feature extraction (want time seq of features: via sliding window/Fourier Transform)

* Cepstral Features: FT → Magnitude → Mel Filter banks → log() → DCT(discrete cosine transform) (a similar form of Inverse FFT)→MFCCs

Modeling Seq of Features:

* **Gaussian Mixture Models (GMMs)**: Weighted sum of Gaussian dists
  + Building a GMM: (MLE)

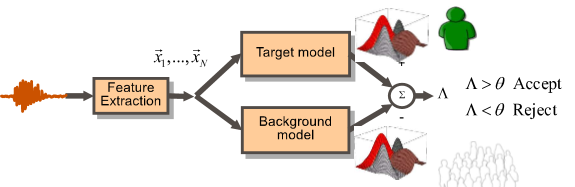
1. Compute likelihood of seq of features given a GMM
2. Estimate params of GMM given set of feature vectors

(Argmax log likelihood of features wrt GMM params)

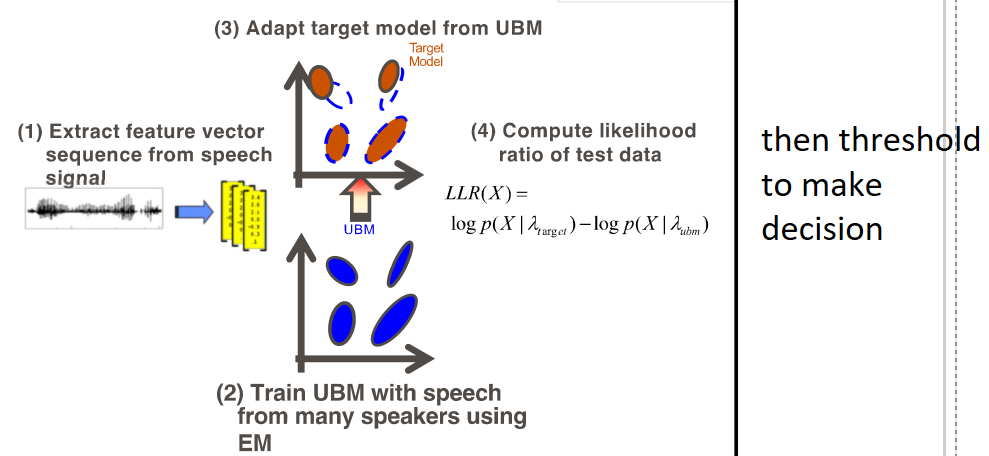
* + Train GMM w/ **Expectation Maximization (EM)** algo:

1. E step: Probabilistically align vectors to model (Compute stats) / function for the expectation of the log-likelihood evaluated using the current estimate for the parameters
2. M step: Update model params (normalize by frame) / computes parameters maximizing the expected log-likelihood found on the E step

* **GMM-UBM**:
  + Speech signal → Use GMMs for both target and background models
    - Target trained w/ enrollment speech (not likely to produce all phonemes in prior speech data)
    - Background trained w/ **Universal Background Model (UBM)**
  + → Compare and threshold to accept/reject

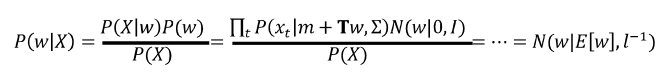
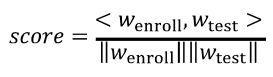


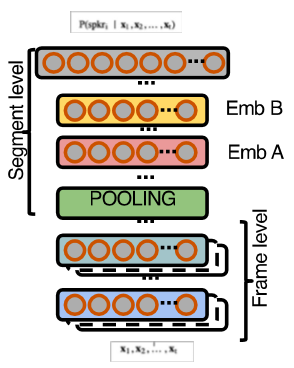
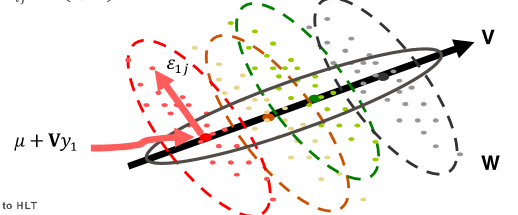
* **GMM-UBM Mean-only adaptation**

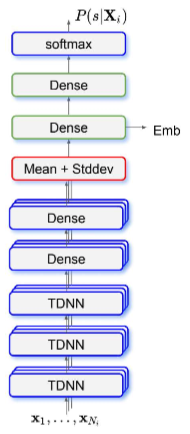


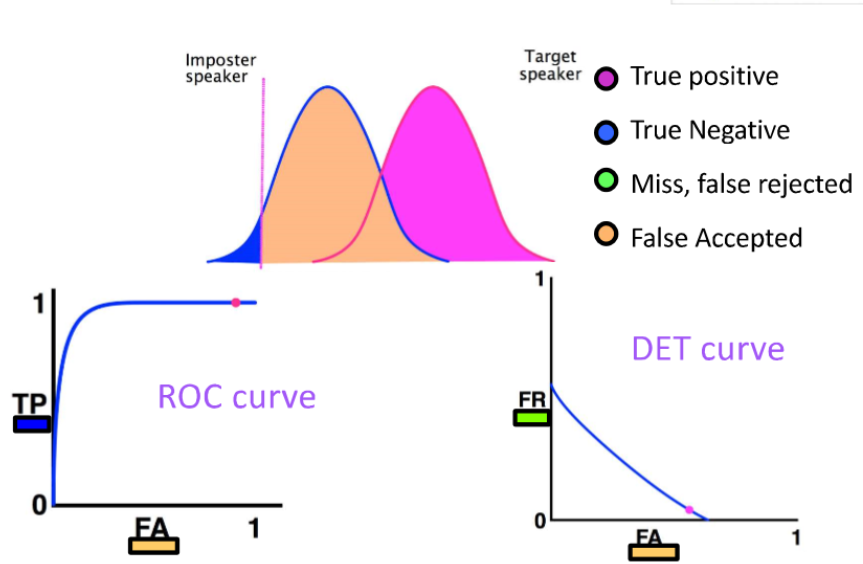
* **i-vectors** (intermediate vector) (Probabilistic principal component analysis):total variability model (i = intermediate-size vector) - Represent GMM supervector by a single total-variability space
  + Produced from UBM via PCA



* + mu = utterance specific mean supervector, m = UBM-GMM supervector mean, T = low rank total variability matrix (high variability subspace), w = standard normal random vector
  + In practice, computed as:
    - , where posterior distribution
  + (unsupervised, PCA)
  + Advantages:
    - Limits adaptation directions of UBM → robust to noise, etc
    - Reqs less data than GMM-UBM - don’t need data for all the Gaussians
      * Use data from a few Gaussians to estimate w
    - Compression - summarize speech recording into small vector
  + Processing i-vectors: channel compensation and scoring
    - Cosine scoring 
    - Channel compensation techniques
      * Linear Discriminant Analysis (LDA) - linear classification and maximize the margins in between!
      * Probabilistic Linear discriminant Analysis (PLDA) - LDA but with a probability attached to it being in a particular class
        + i-vector j of class i is decomposed as sum: 



* x-vectors - use nonlinear models
  + DNN trained to discriminate between speakers to produce better embeddings
  + Input: feature seqs of variable length (MFCCs, Mel filter-banks, bottleneck features)
  + Output: posterior prob for the speaker labels
  + 3 parts:
    - Encoder: extracts frame level representations
    - Pooling: pooling layer that computes mean and stdev
      * Mean+Stdev
      * Learnable dictionary encoder (LDE)
      * Multi-head attention
    - Classification: predicts posterior probs for the target speakers
  + Once trained:
    - Softmax layer removed
    - Embeddings extracted from layers after pooling layer
  + Time Delay NN (TDNN) x-vector (comes from TDNN encoder)
    - Can capture features in wider window as it gets deeper
    - Dilation makes temporal context to grow faster as the info travels thru the layers of the network
  + (Factorized) F-TDNN x-vector w/ skip connections
  + ResNet x-vector

Applications:

* Speaker verification: Accept/reject a user and their claimed identity based on their speech signal
  + Target model and imposter model
  + Likelihood ratio to get curves (as seen on right)
  + Hypotheses: null hypothesis (legitimate speaker); alternative hypothesis (imposter)
    - False negative, false positive (false alarm), or true positive, true negative
  + DetCurve: Prob of False neg vs Prob of False Positive
  + ROC Curve: True positive vs False Positive

## 7. Deep Learning I

**Word error rate (WER)** = word-level edit distance (insertions, substitutions, deletions)

* Speech recognition evaluation metric
* WER (%) = edit distance / # reference words

History of speech recognition:

* 2001: no applications, no breakthrough tech, criticized by everyone outside speech research, little public understanding
* Now: applications -- voice search, smart speakers, breakthrough tech -- deep nn’s, widespread recognition and appreciation

Bayesian model (before 2011):

* **Acoustic model** = , **Language model** =

**GMM/HMM (before 2011):**

* Likelihood function given HMM state j:
* Limited performance because it’s a linear model
* For GMMs, need to think about correlation (delta, LDA, covariance)
  + GMMs are sum over lots of gaussians
* **Frequencies can be broken up by moment in time (i.e. model type of speech seen in each state)**
* Input features: MFCC, LPC

**Deep Neural Network (after 2011):**

* Can use DNN to solve the same problem as GMM/HMM
* Input = MFCC vector
* Output = phoneme (or HMM state)
* Training = find using pair data
  + Can use linear classifier or use probability with the sigmoid function
  + Can use combinations of linear classifiers for more complicated patterns, e.g. for speech recognition

**Sigmoid function**

* Squashes input from all real numbers to [0,1]
* Derivative:

Why was adaptation of nn’s slow:

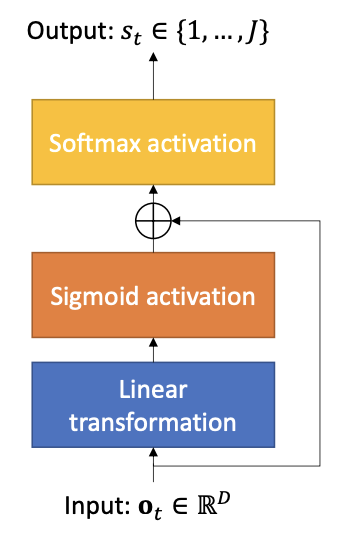
* Very difficult to train
  + Batch (slow convergence but efficient computation)? On-line (fast convergence but inefficient computation)? Mini-batch (between batch and online)?
  + SGD? Learning rate for it?
  + Network topology?
  + Large computation cost
* Need large amounts of training data
* CPU → GPU

**Kaldi** = Feed forward nn w/ deep belief network-based pre-training & sequence-discriminative training

**Basics of Feed-Forward Neural Networks:**

* Configs for feed-forward nn for acoustic model:
  + Input features: context expansion
    - Don’t need to care about correlation nor use markov assumptions, just concatenate left and right contexts
  + Output class: softmax, training criterion
    - Phoneme or HMM state ID
    - Use Viterbi algorithm to get the state sequence for all *t*

* + - Treat as multiclass classification by predicting all HMM state IDs given observations
  + # layes, # hidden states
  + Type of non-linear activation (need to be able to take the derivatives of them!)
* Training process:

1. Affine transformation and non-linear activation function (sigmoid)

)

1. Apply the transformation L times
2. Softmax to get the probability distribution

p(j | h) = [**Softmax** (h)

Why do we feed to the softmax?

To normalize the output into probability (in [0,1])

* Objective function
  + **Cross entropy**
  + Others: square error, binary cross entropy
* Use gradient descent to optimize
  + Use backpropagation with chain rule to get derivative recursively

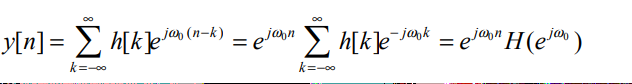
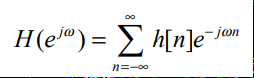
## 8. Deep Learning II

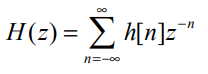
**Bottleneck feature** = intermediate representations that can be used by DNN instead of original features

* Generated by forcing a small layer to learn a lot of info

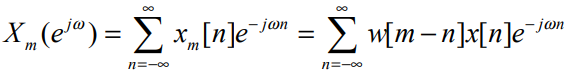
## 9. Signal Processing

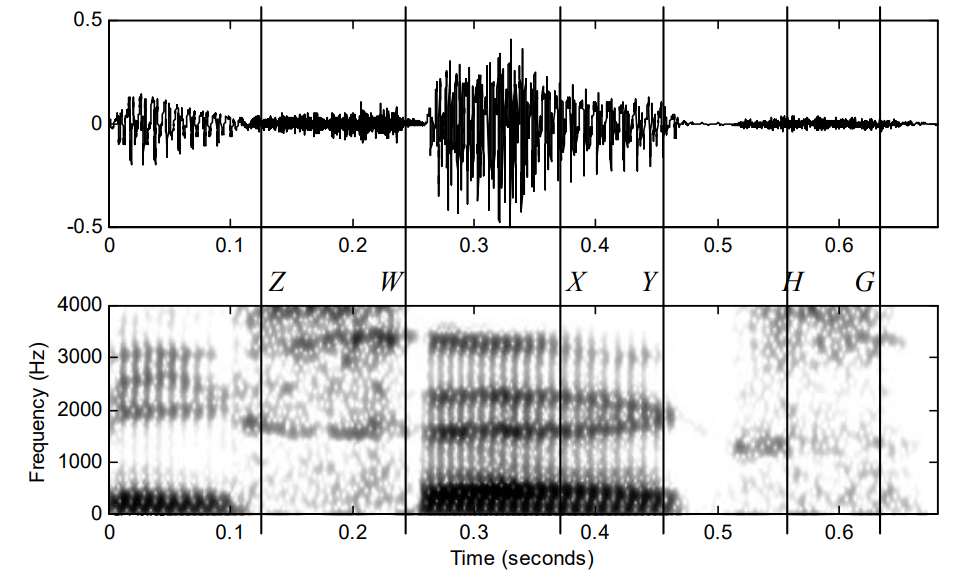
Discrete Signal x[n], Continuous x(t)

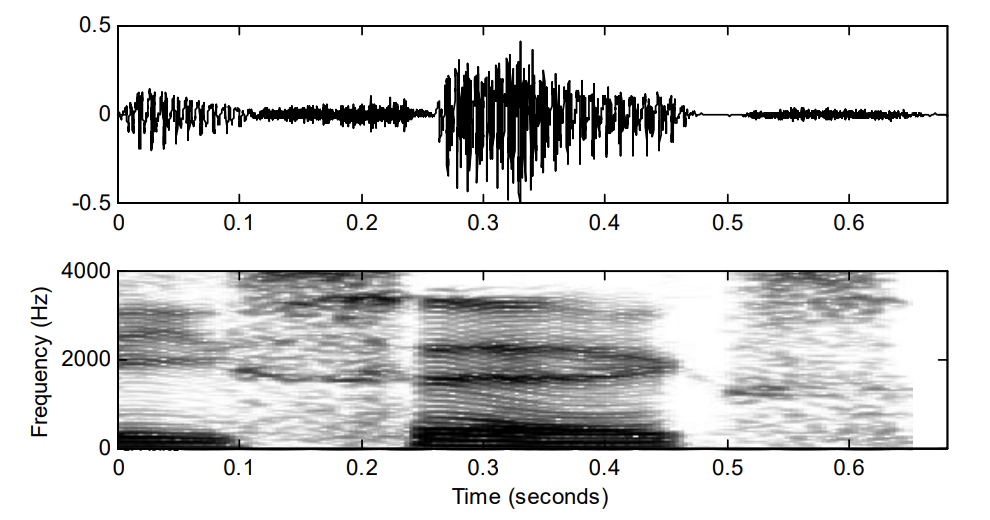
**Fourier Transform** , 

**Z-transform** (generalization of FT, where z=ejw) 

**6.1 Short-Time Fourier Analysis**

* Given spectrogram (freq vs time), decompose speech signal into a series of short segments (*analysis frames*), and analyze each independently
* Short-time signal , wm[n] window function (0 everywhere except windowing region)
* Short-time Fourier rep for frame m: 
  + FT prop: multiplication in freq domain ↔ convolution in time domain

 ←**wide-band spectrogram**- short windows, <10ms, good time res but lower freq res - see by vertical bars (artifacts of window)

*  ←**narrow-band spectrogram**- long windows, >20ms, filters w/ narrow bandwidth (<100Hz); time res is lower - see by horizontal stripes (**harmonics**); but now window too big to localize a phoneme

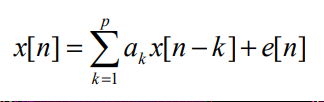
**6.2 Acoustical model of speech production**

Model propagation of sound in vocal tract:

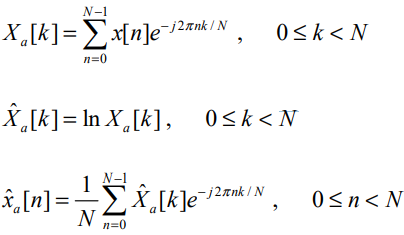
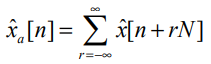
* Lungs → glottis → vocal cords → (mouth, lips, tongue) → speech signal

Given speech signal, try to reverse engineer shape of mouth!

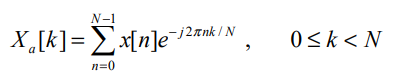
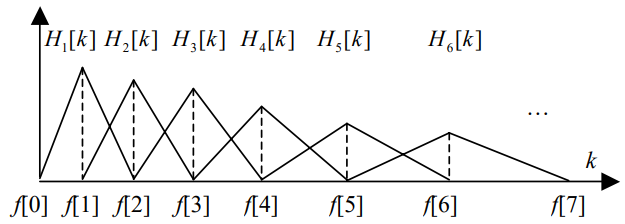
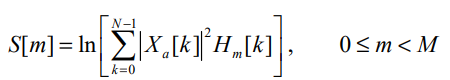
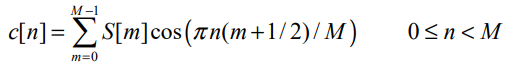
**6.3 Linear Predictive Coding (LPC)** aka LPC analysis aka auto-regressive (AR) modeling

* **Estimate parameters of speech signals, predicts current sample as a lincomb of its past p samples (given previous 20 speech signals, try to predict 21st)**:
* 
* Solving LPC Eqns - 3 Methods:
  + Covariance method
  + Autocorrelation method (this method is stressed in the lecture, it’s solving by a special matrix)
  + Lattice formulation

**6.4 Cepstral Processing**: represent a convolution as a sum (another method for representing speech signal)

* Real cepstrum of signal x[n]: 
* Complex cepstrum of x[n]: , where and phase 
* **6.4.4 Cepstrum of Speech Signals**:
  + Window signal w/ window of length N, then compute cepstrum through DFT\*\*What is DFT?\*\*:
  + 
  + 
  + For unvoiced speech, only real cepstrum has meaning
    - Speech can have freqs at which noise dominates (typically very low/high freqs) that result in a phase that changes drastically from frame to frame

**6.5 Perceptually-Motivated Representations**:

* Sensitivity to loudness/frequency is nonlinear - logarithmic
* **6.5.2 Mel-Frequency Cepstrum Coefficients (MFCC)** - representation of the short-term power spectrum of a sound
  + (real cepstrum of a windowed short-time signal derived from FFT of that signal w/ nonlinear freq scale)
  + **MEL cepstrum** - to represent the “tuning”/nerve cells in ear specific to diff freqs:
    - Get Fourier transform of a window of speech, *bin* the magnitude spectrum into a small number of coeffs, take log of those, decorrelate the feature vector via Cosine transform
  + DFT of input signal 
  + Filterbank of triangles w/ increasing bandwidths 
  + Then compute log-energy at output of each filter 
  + Mel freq cepstrum: 
  + Typically only first 13 cepstrum coeffs are used
* **Perceptual Linear Prediction (PLP)**: LPC coeffs → LPC-cepstrum ???????? it doesn’t have much?

**6.7 Role of Pitch**

* E.g. Chinese tone recognition
* Short term analysis techniques: for every frame xm we get score f(T|xm) → ™ = argmaxT f(T|xm)
* **6.7.4 Pitch Tracking**

## ~~10. End-to-end neural speech recognition~~ -- NOT ON EXAM