Algorithms for Speech and Natural Language Processing

The image part with relationship ID rId2 was not found in the file.

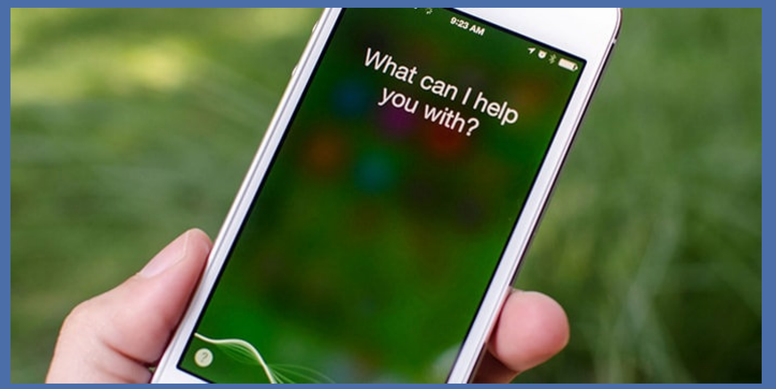
The image part with relationship ID rId2 was not found in the file.

Speech features and Acoustic Models

Neil Zeghidour

Phd Student, FAIR Paris, CoML (INRIA/ENS)

Speech recognition



Personal assistants

“FIND ME AN ALGERIAN RESTAURANT NEARBY”

Personal assistants

These devices can:

- Identify your voice
- Recognize what you say
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- **Identify your voice**
- Recognize what you say
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice
- **Recognize what you say**
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice
- Recognize what you say
- **Understand what you mean**
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice
- Recognize what you say
- Understand what you mean
- **Answer with a natural voice**

Personal assistants

These devices can:

- Identify your voice
- **Recognize what you say**
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice ➡️ Speaker identification
- **Recognize what you say ➡️ Speech recognition**
- Understand what you mean ➡️ Natural Language Processing
- Answer with a natural voice ➡️ Speech Synthesis

What is speech recognition?

RECORDING

TRANSCRIBING

“FIND ME AN ALGERIAN RESTAURANT NEARBY”

Our task: Speech ➡️ Text

Outline

“CAB”

Outline

“CAB”

Outline

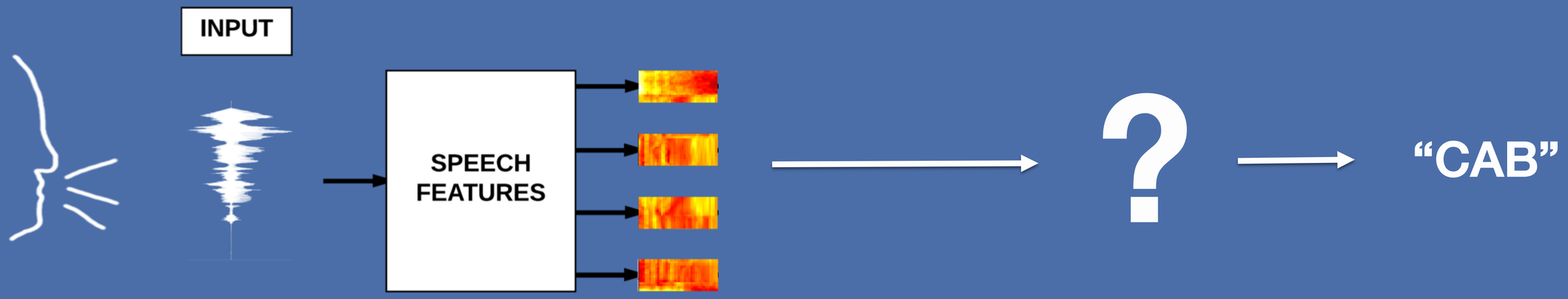
I. Anatomy of a speech recognition system

1. Speech and speech features

Outline

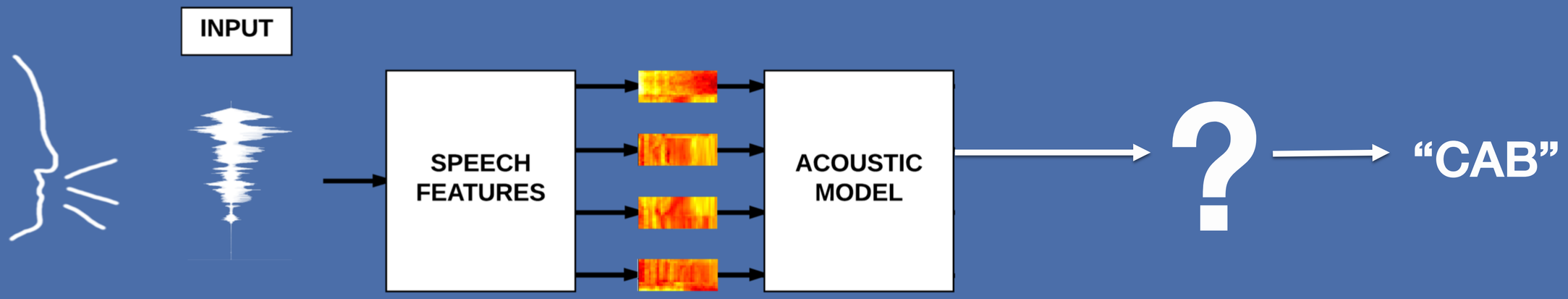
I. Anatomy of a speech recognition system

1. Speech and speech features



Outline

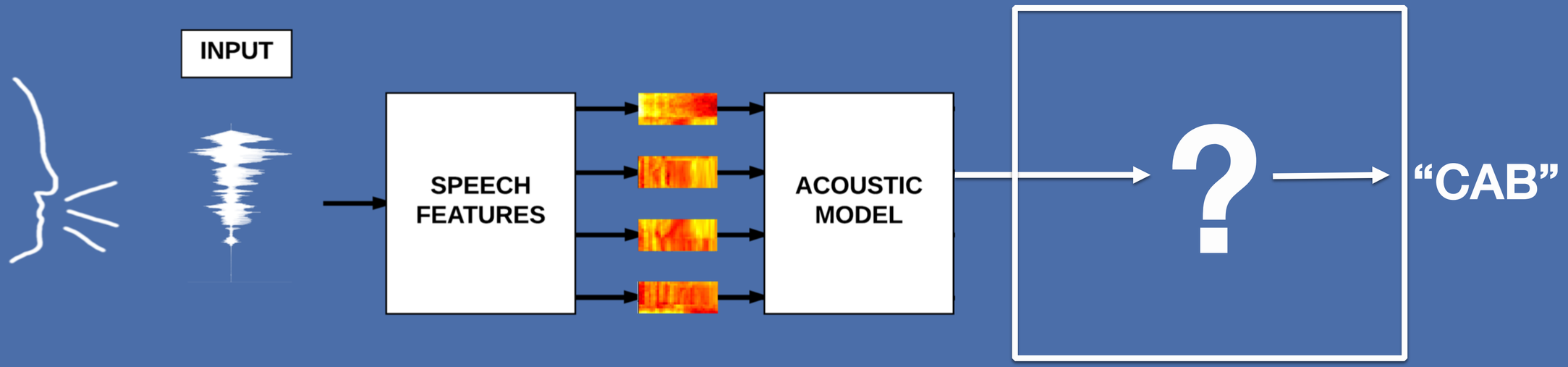
I. Anatomy of a speech recognition system
   1. Speech and speech features
   2. Acoustic model



Outline

I. Anatomy of a speech recognition system
   1. Speech and speech features
   2. Acoustic model
   3. Language Modelling

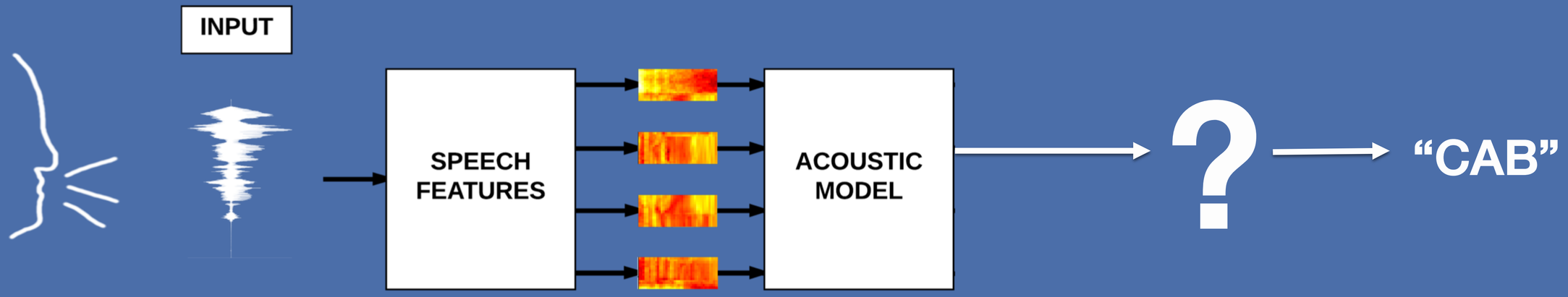
Next class on language modelling



Outline

I. Anatomy of a speech recognition system
   1. Speech and speech features
   2. Acoustic model
   3. Language Modelling

II. Handling variability in speech
   1. Gender
   2. Speaker identity
   3. Noise



Speech and speech features

- The speech waveform
- Spectrogram
- Speech production
- Matching human perception

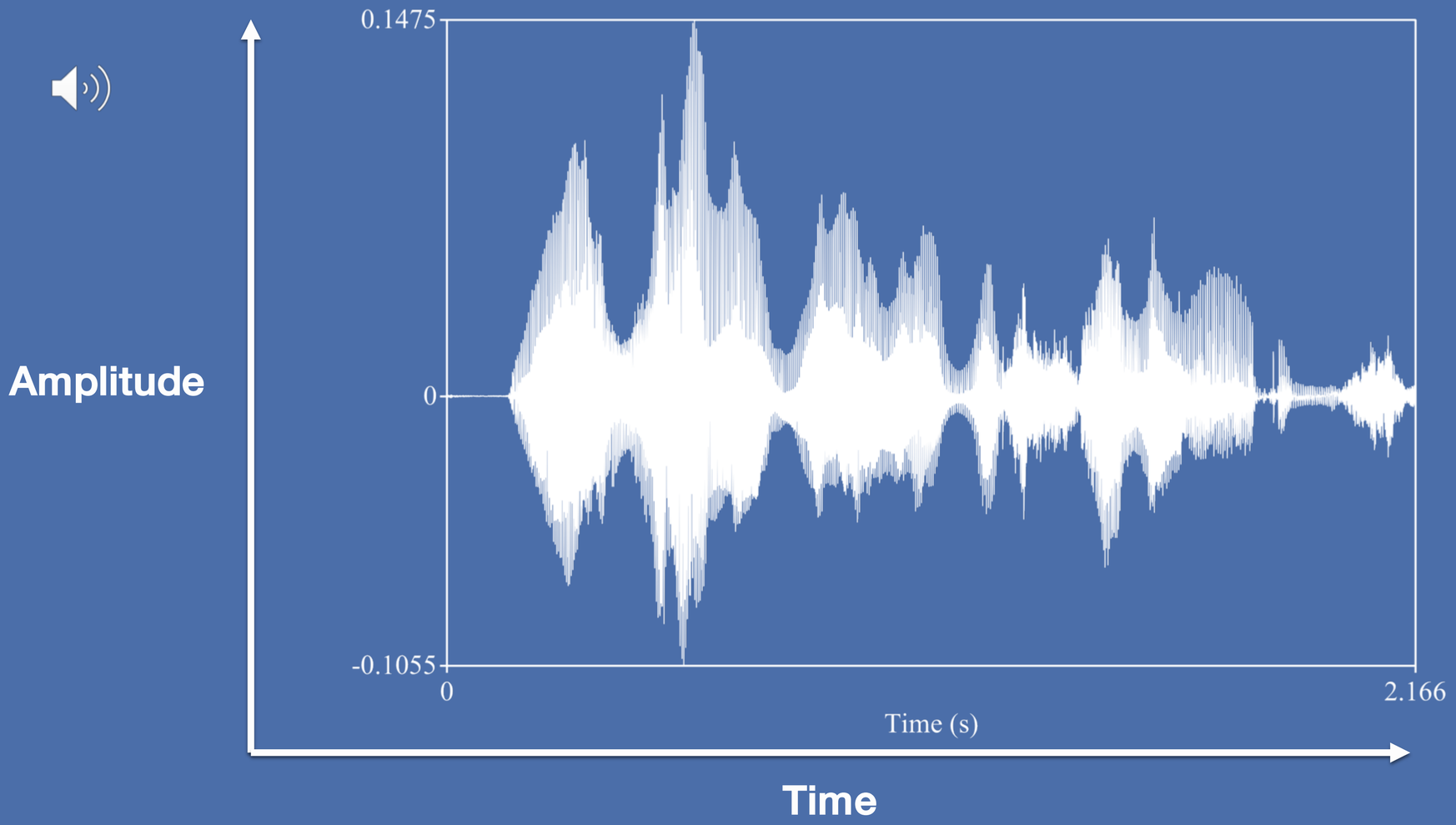
What does a microphone record ?

- A microphone measures variations in air pressure
- It collects a discretized (not continuous) signal
- Can record at different samplerate (8kHz, 16kHz, etc.)
- Codes the amplitude of the air pressure on 16 bits

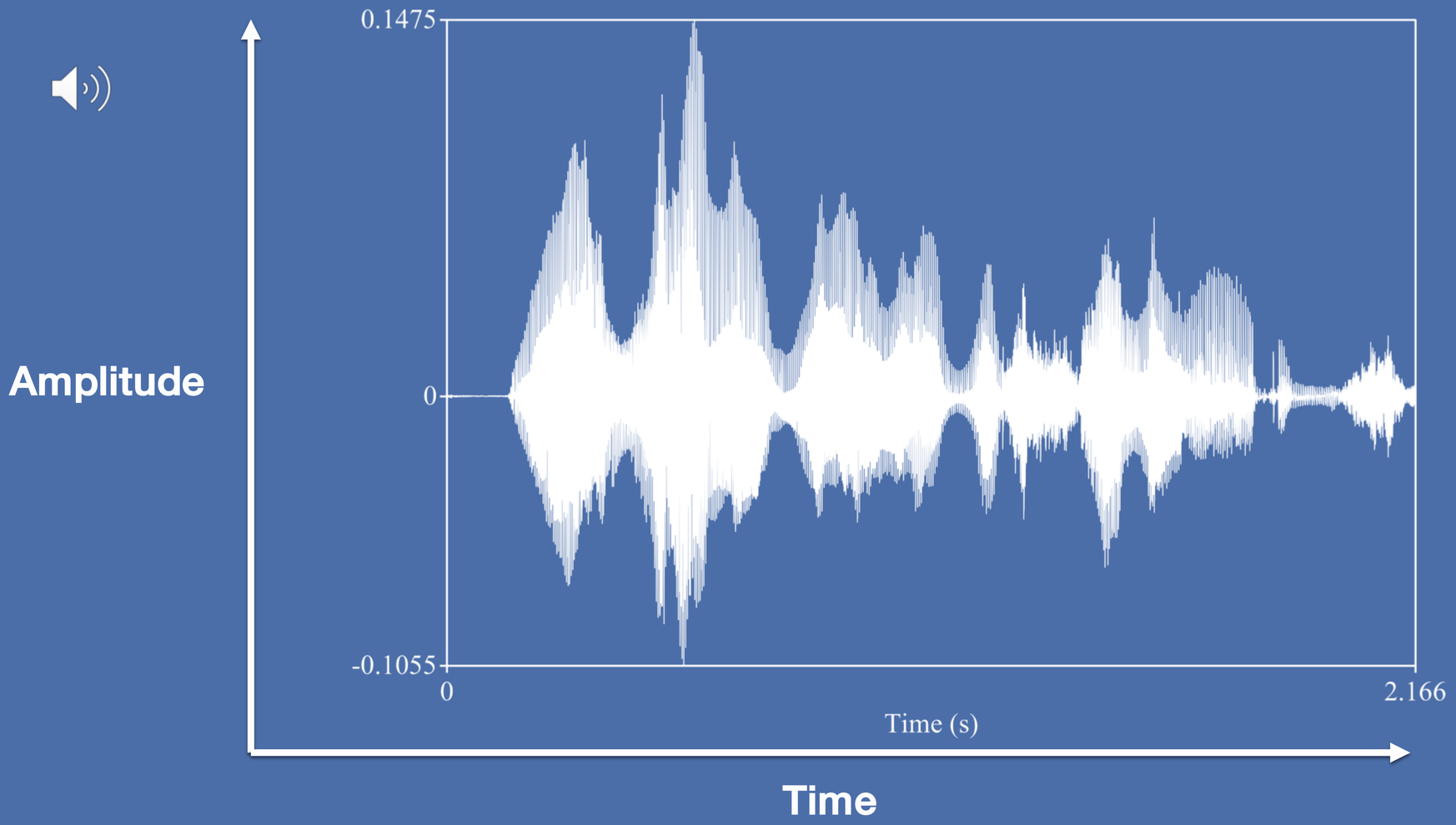
[-0.10, 0.01, -0.02,..., 0.05]

3 seconds * 16 000 Hz = vector of length 48000

The speech waveform

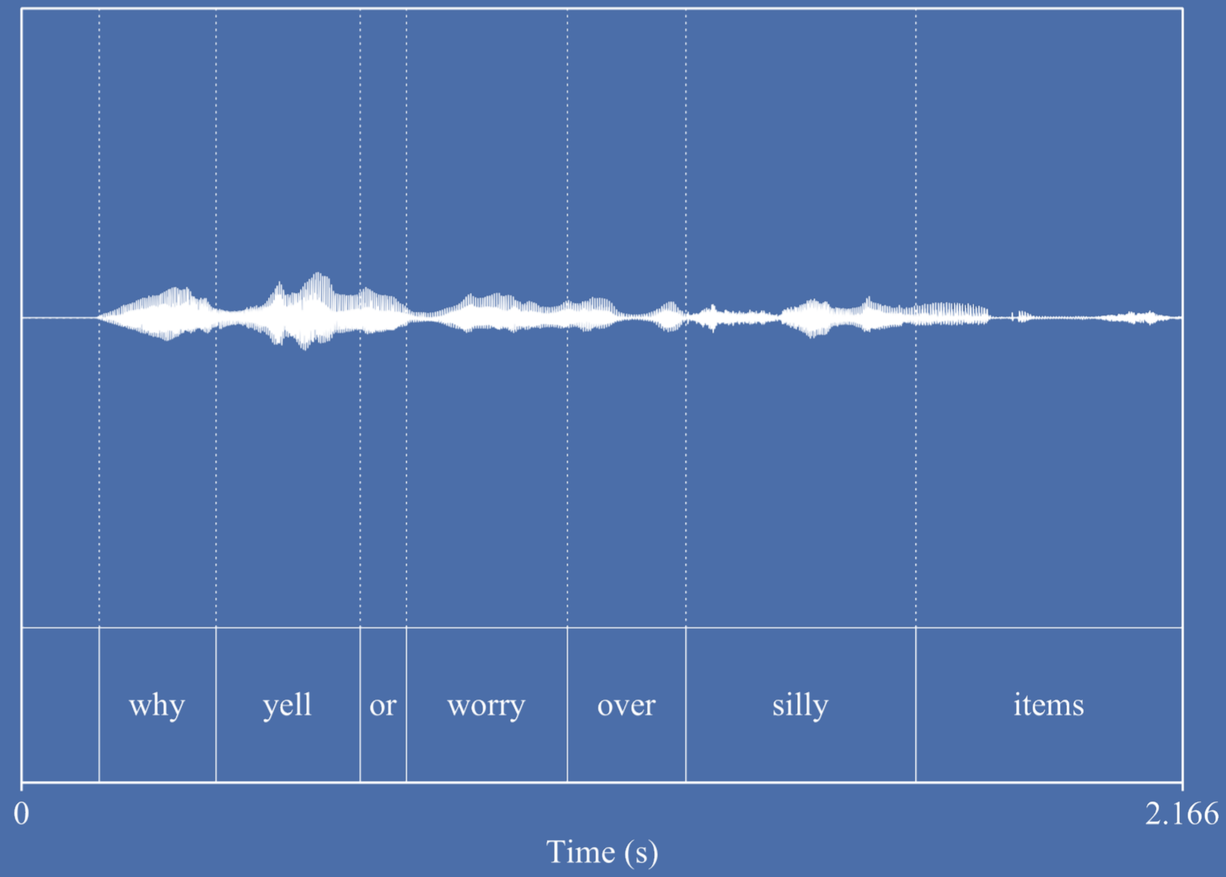


The speech waveform



The speech waveform

- Word transcription: what the sentence looks like when it’s written

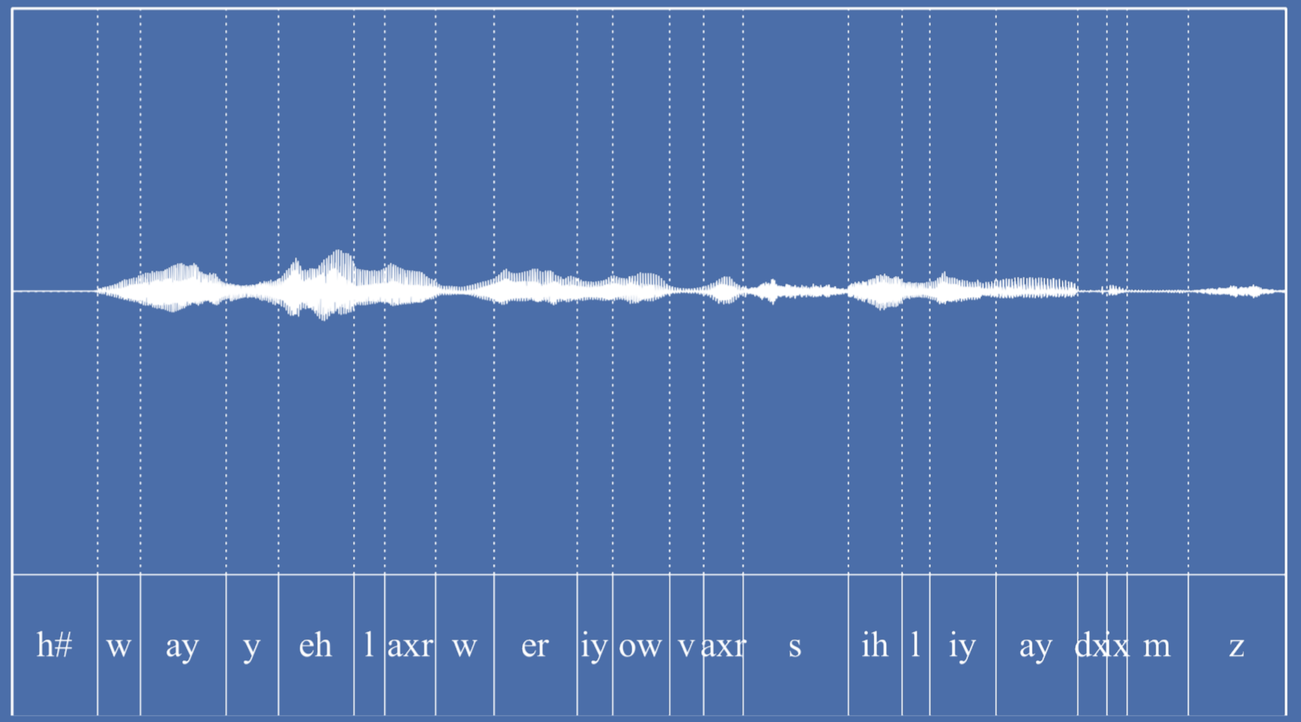


Phonetic transcription

- Thibault vs /tibo/
- Sean vs /ˈʃɔːn/
- A phonetic unit is called a **phoneme**

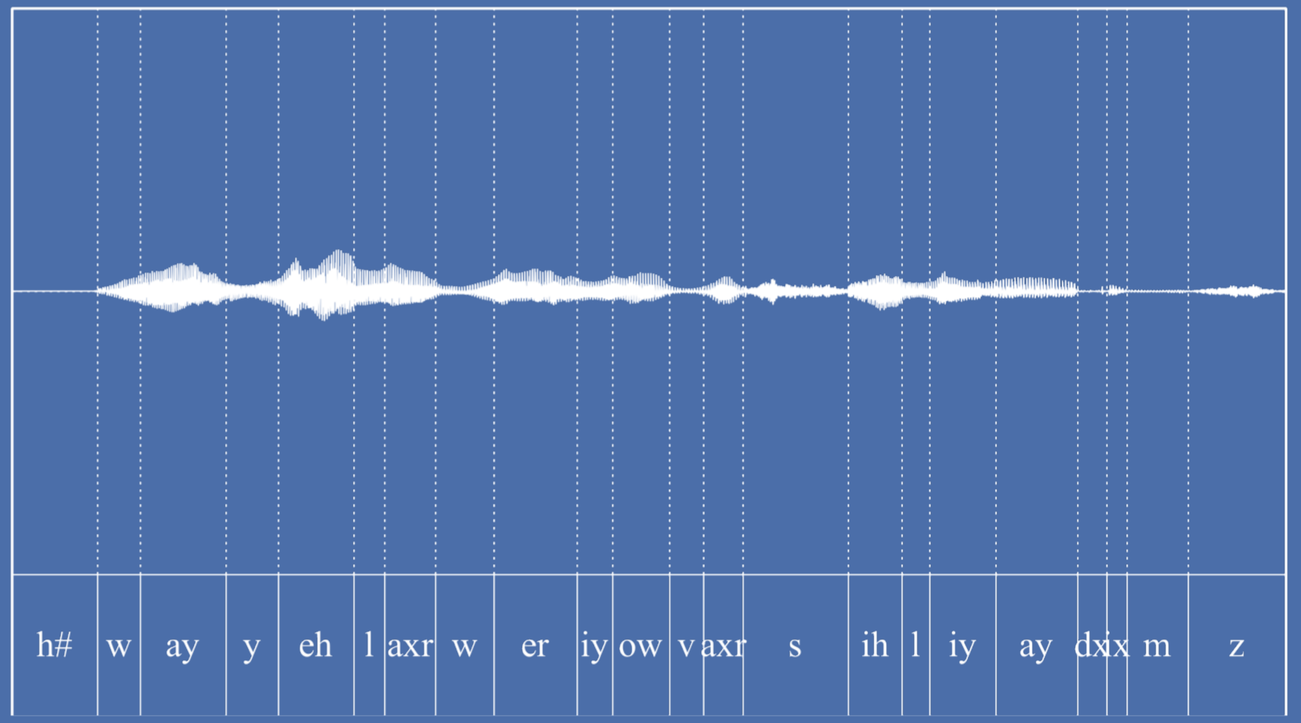
The speech waveform

- Phonetic transcription: what the sentence **sounds** like



Why we do not train directly on the waveform

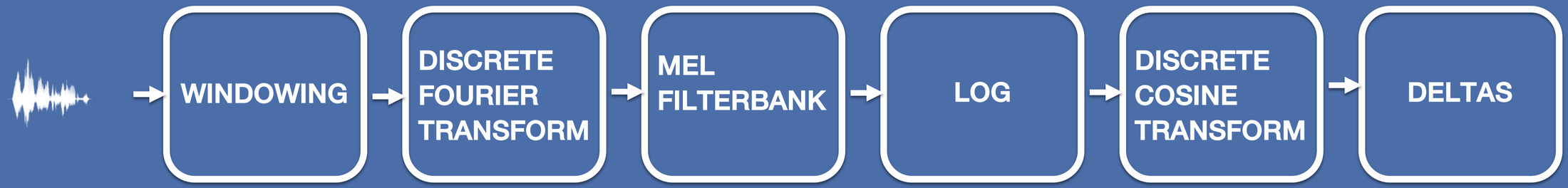
- Hard to correlate the phonetic content to a waveform
- Classifying/clustering the waveform is very hard
- We extract **features: representation of a signal that makes learning easier for an algorithm**



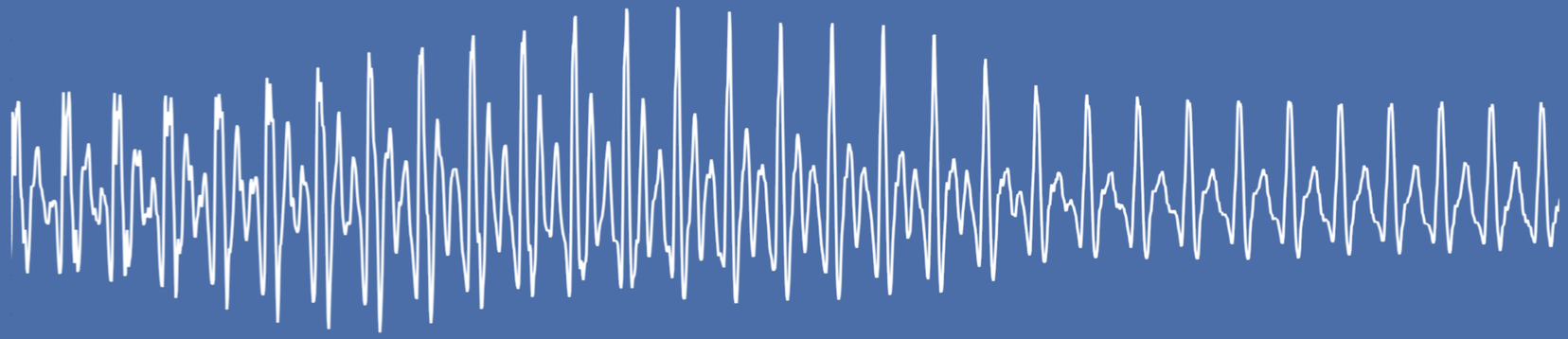
Desired properties of speech features

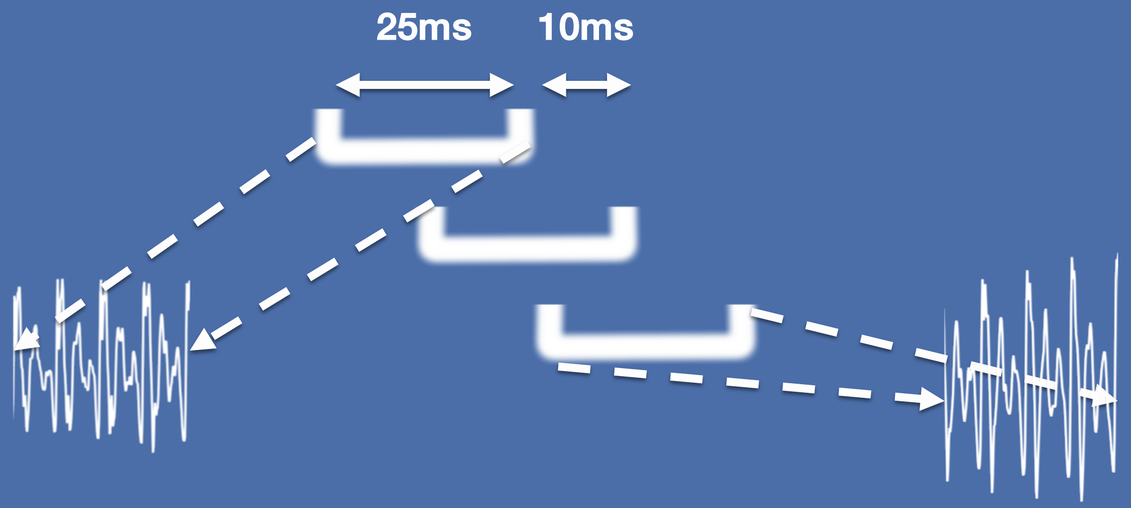
- The waveform is a highly non stationary signal ➡️ we need local features
- Phonemes are characterized by their spectral (frequency domain) signature ➡️ we need spectral features
- The speech signal is high dimensional (8k-16k values per second) ➡️ we need compact features
- Two types of speech features have these properties: **mel-filterbanks** and **MFCC**

The MFCC (Mel-frequency Cepstral Coefficients) Pipeline



Extracting local representations





Duration of a phoneme:
a few dozens miliseconds

The Discrete Fourier Transform

- The phonetic content is characterized by the spectral signature i.e. the frequency representation of the signal
- The waveform is in the time domain (amplitude along time)
- The Discrete Fourier Transform brings the signal into the frequency domain (amplitude along frequency)

$$X(\omega) = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi\omega \frac{n}{N}}$$

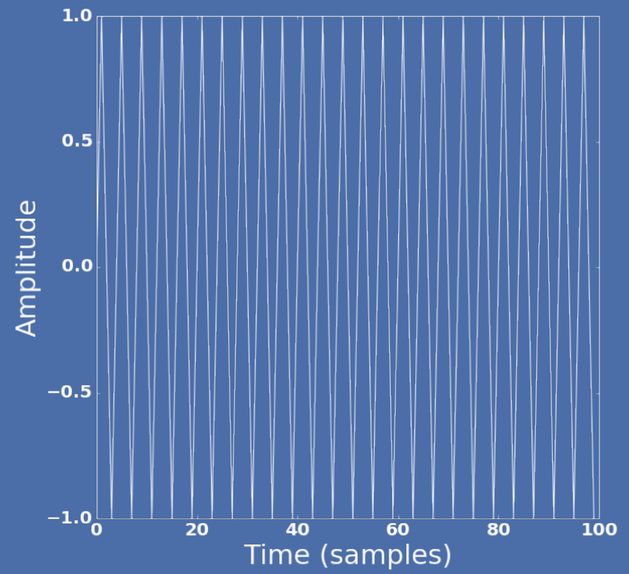
Frequency **Waveform**

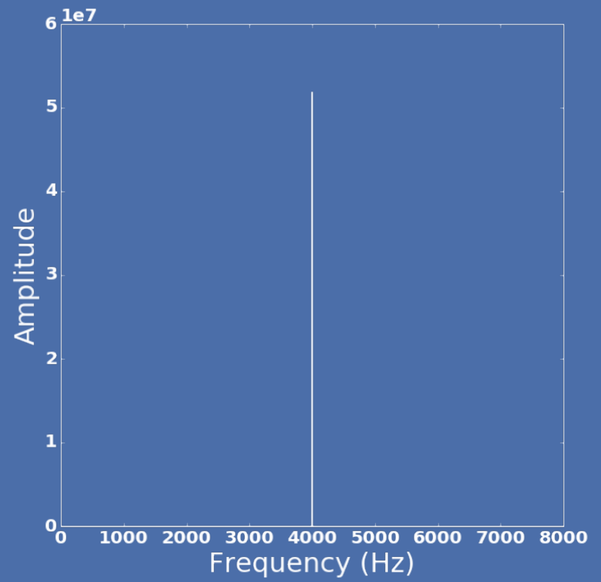
Frequency representation: The power spectrum

- The output of the Discrete Fourier Transform is a complex vector
- We obtain the spectrum by taking the squared modulus of the DFT
- Example: Sine wave of frequency 4000Hz

$$x[n] = \sin(2\pi * 4000 * n)$$

$$|X(\omega)|^2$$





Good for stationary signals, however this does not model variations in frequency along time

Time-frequency representation: The spectrogram

- Computes the spectrum over the windows, instead of the full signal (Short-Term Fourier Transform), to model variations in frequencies along time

- If the signal is made of 100 windows, each being centered in $c_{i=1..100}$:

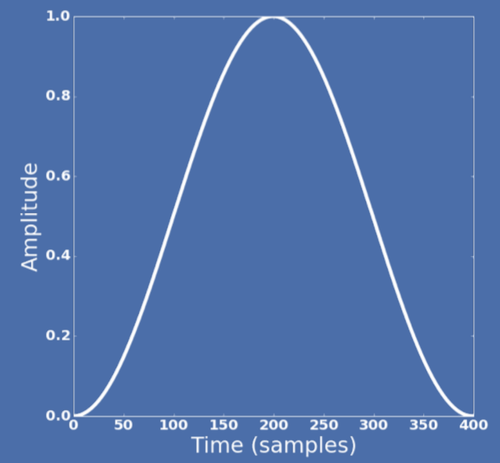
$$h[n]$$

$$|X(c_k, \omega)|^2 = |\sum_{n=-\infty}^{\infty} x[n]h[n - c_k]e^{-i\omega n}|^2$$

Spectrogram

Waveform

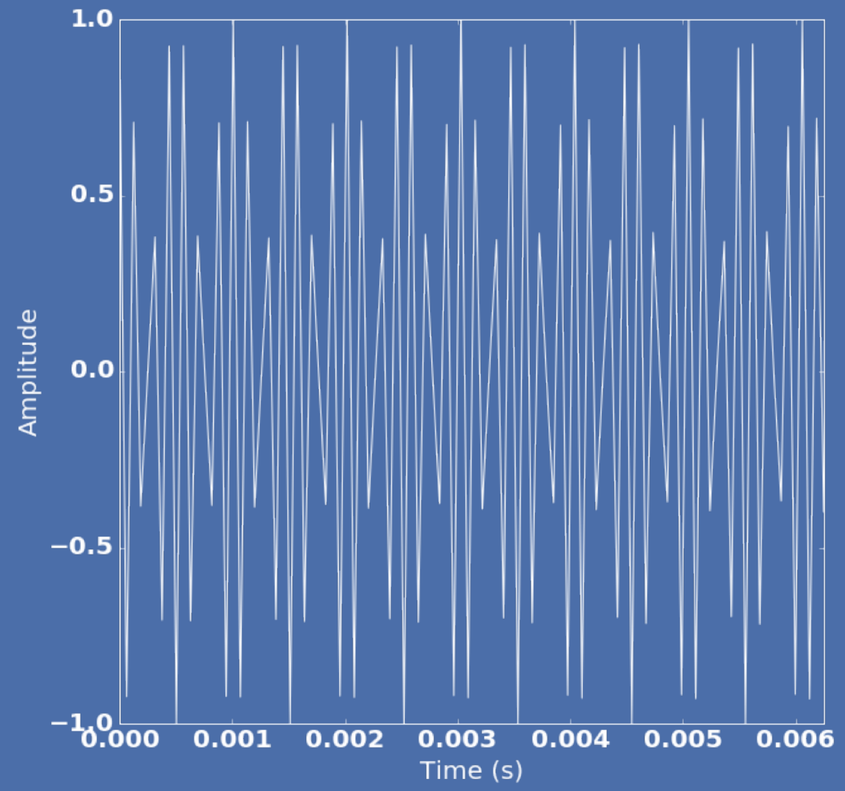
Window



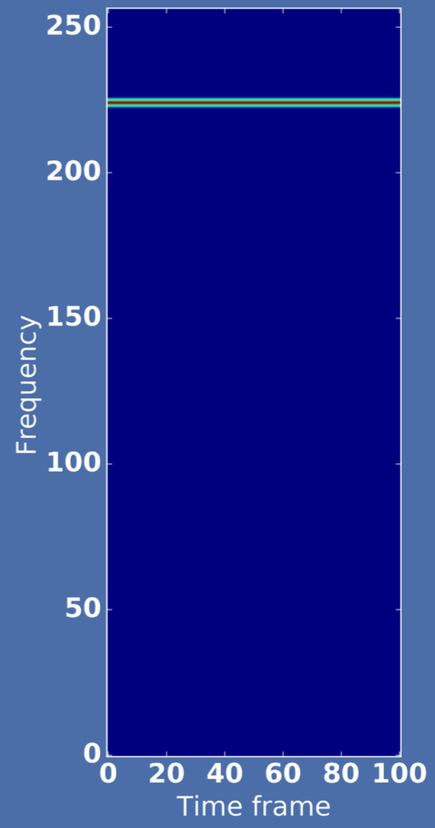
Hanning window

- Outputs a matrix of dim [n_windows, n_frequency_bins]

Spectrogram: High pitch

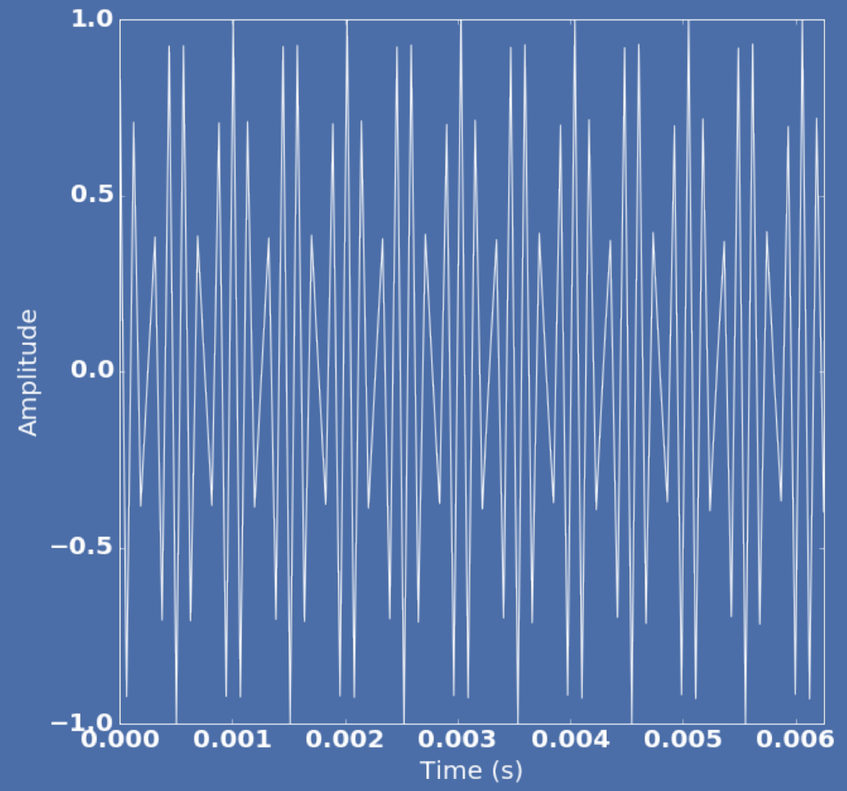


Waveform

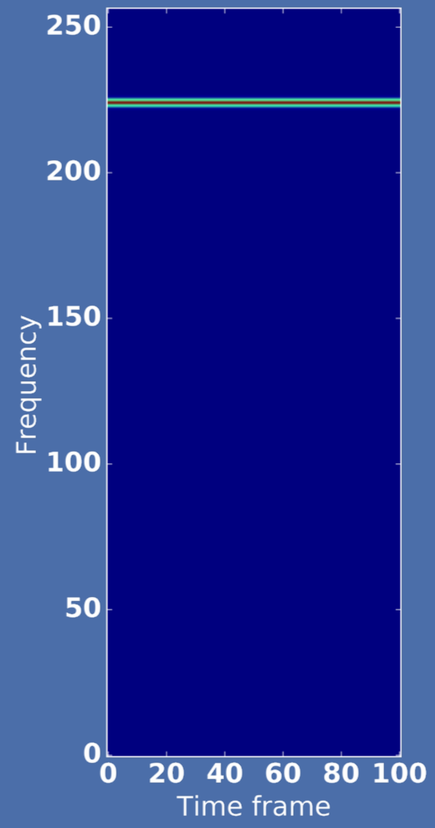


Spectrogram

Spectrogram: High pitch

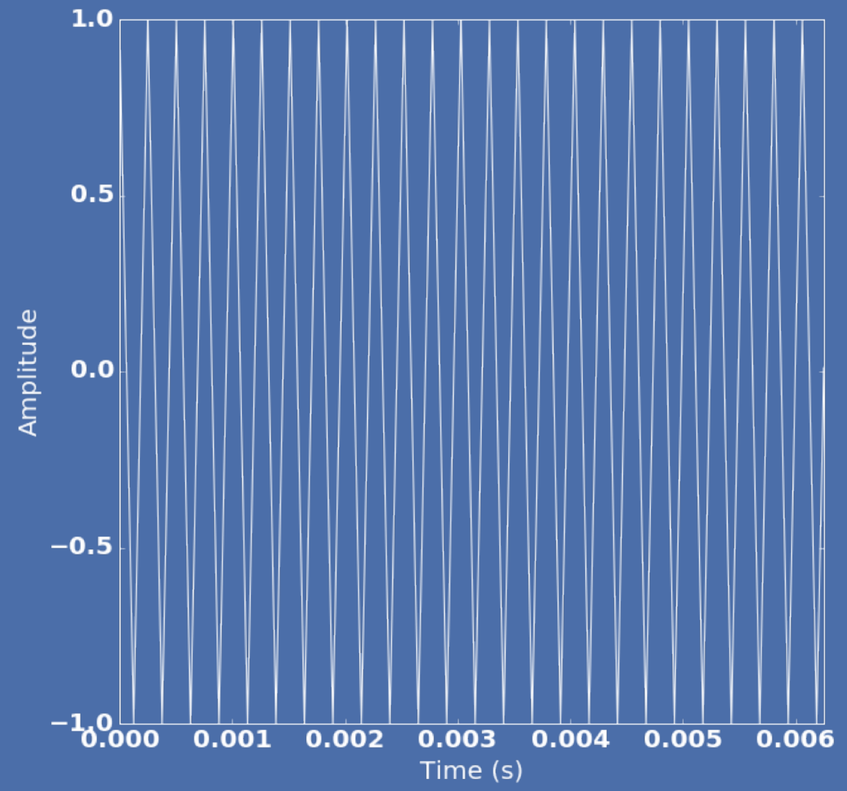


Waveform

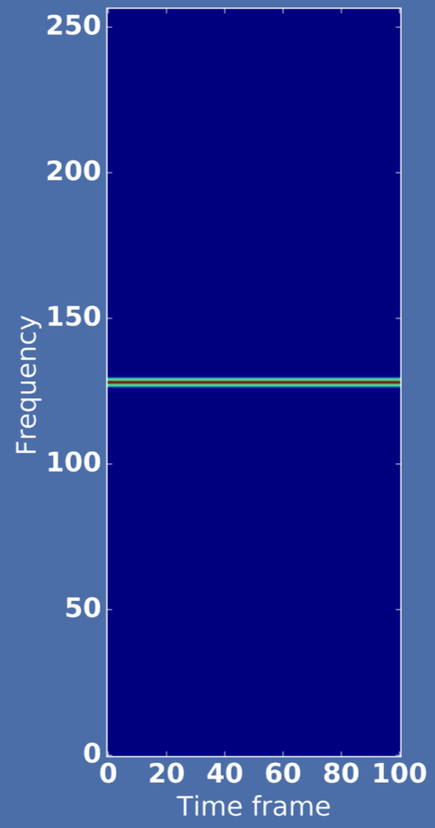


Spectrogram

Spectrogram: Middle pitch

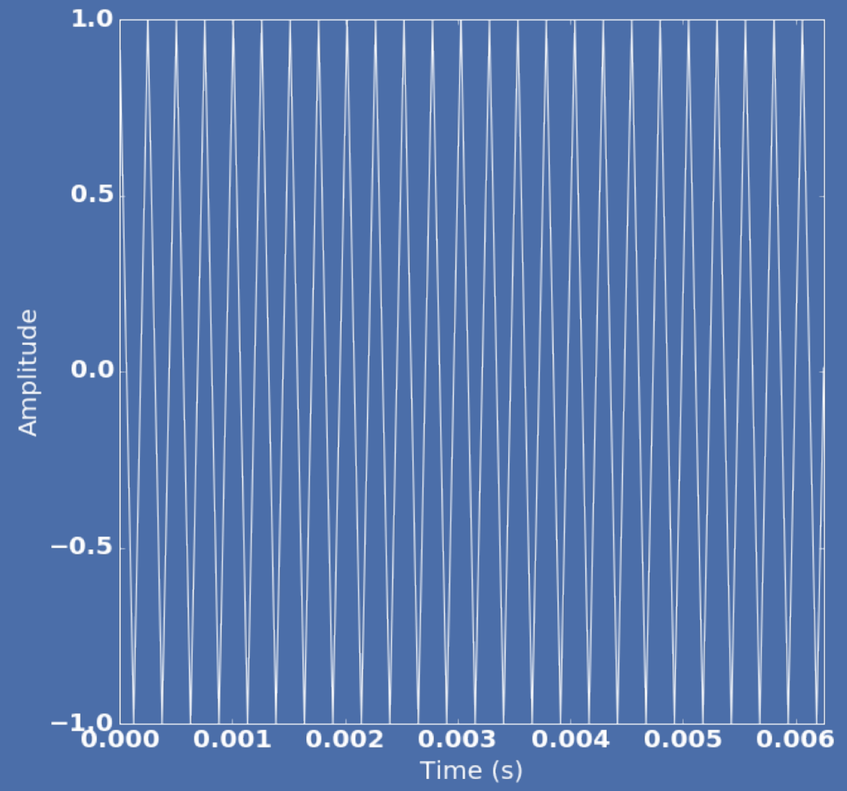


Waveform

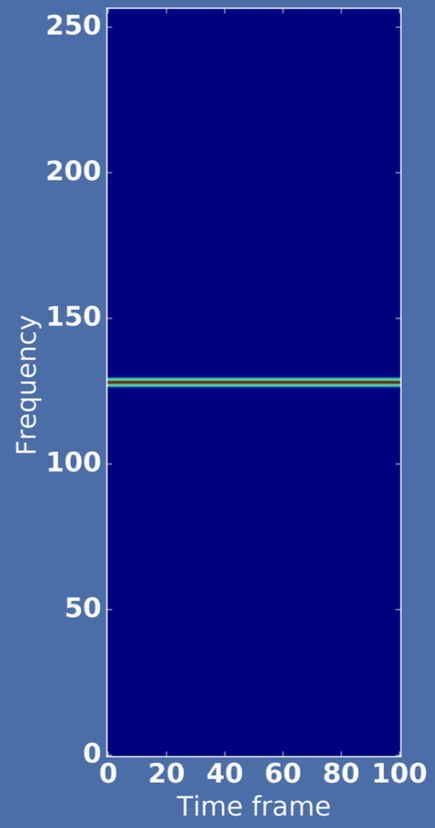


Spectrogram

Spectrogram: Middle pitch

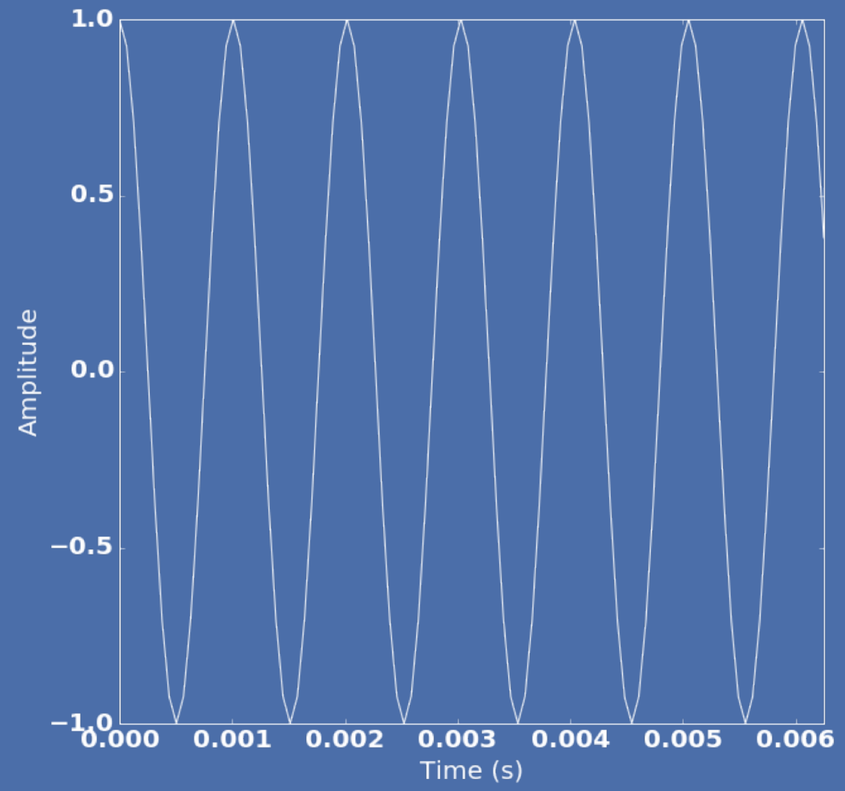


Waveform

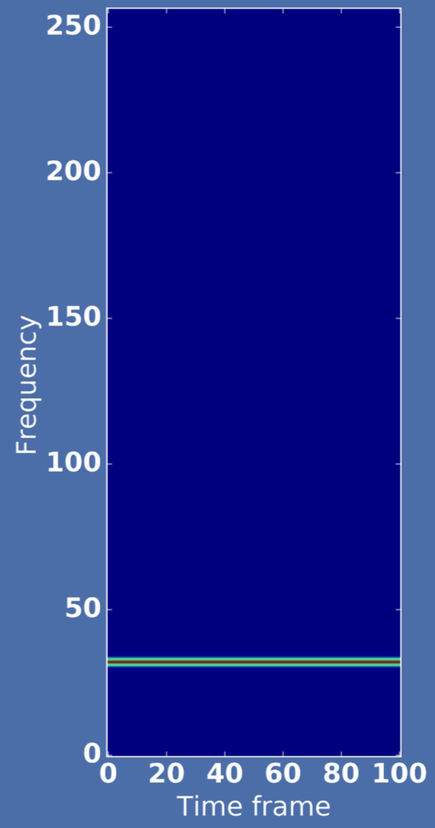


Spectrogram

Spectrogram: Low pitch

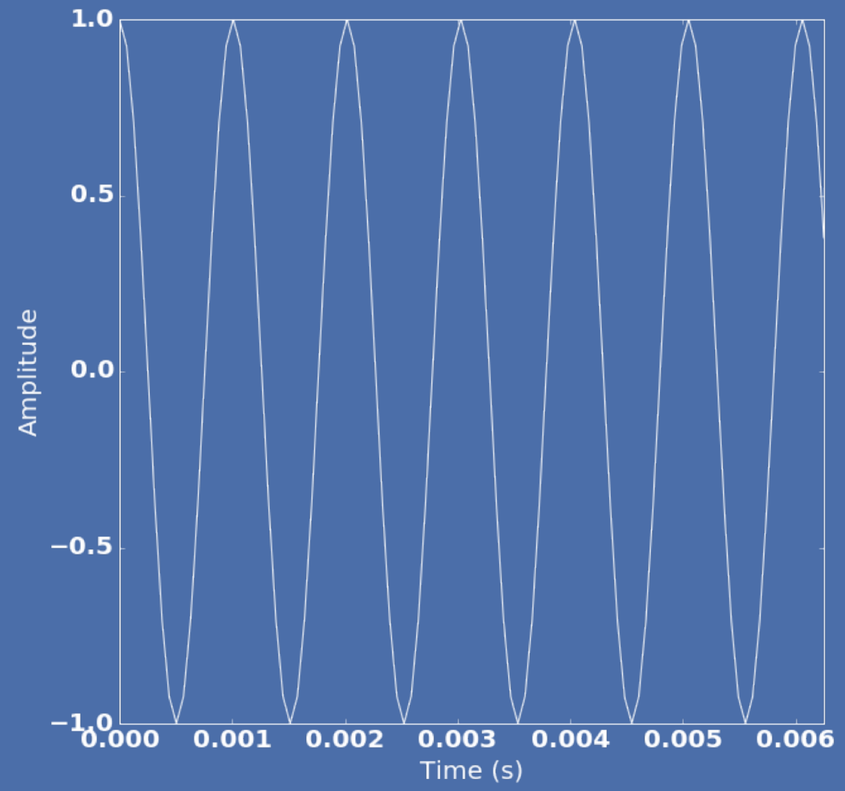


Waveform

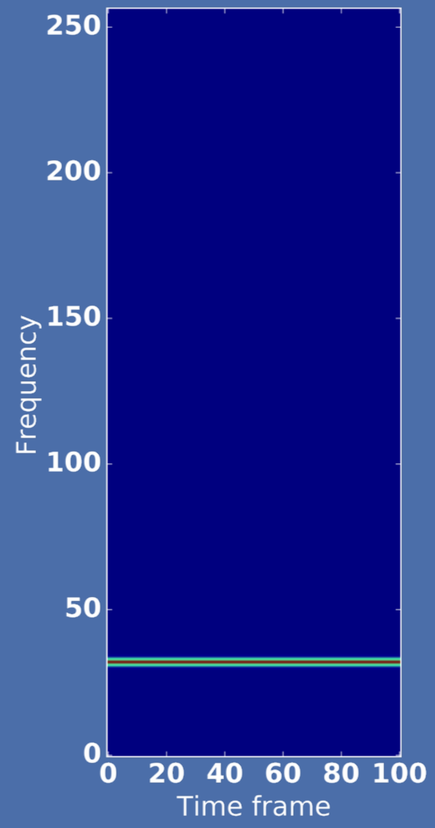


Spectrogram

Spectrogram: Low pitch

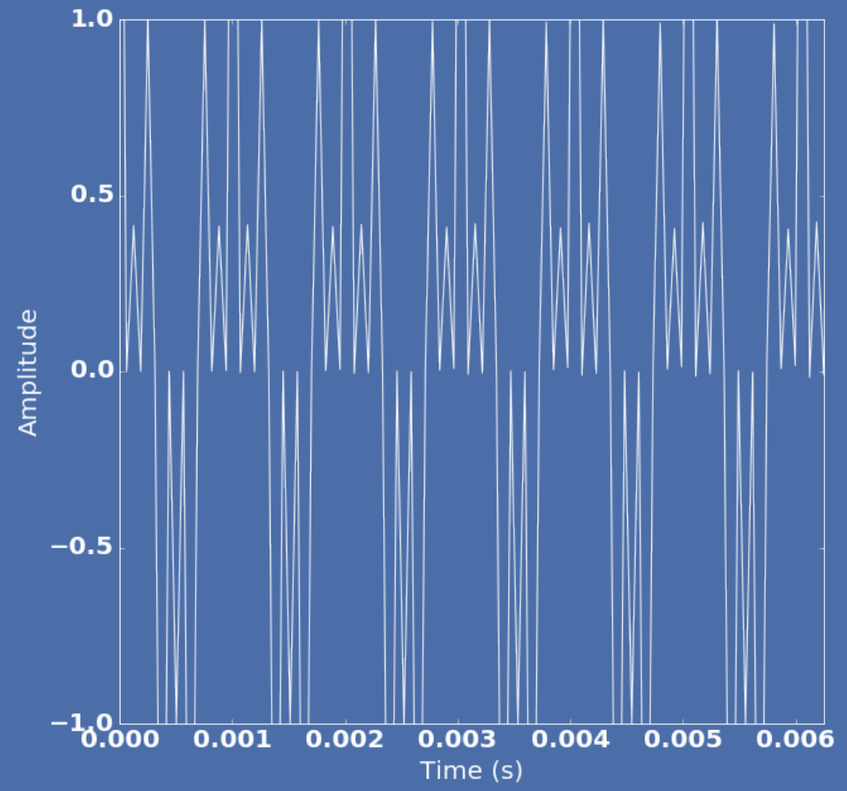


Waveform

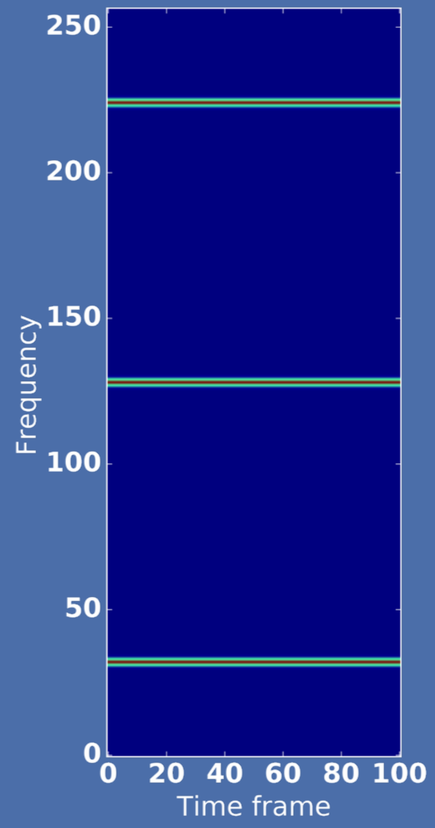


Spectrogram

Spectrogram: Sum of the three

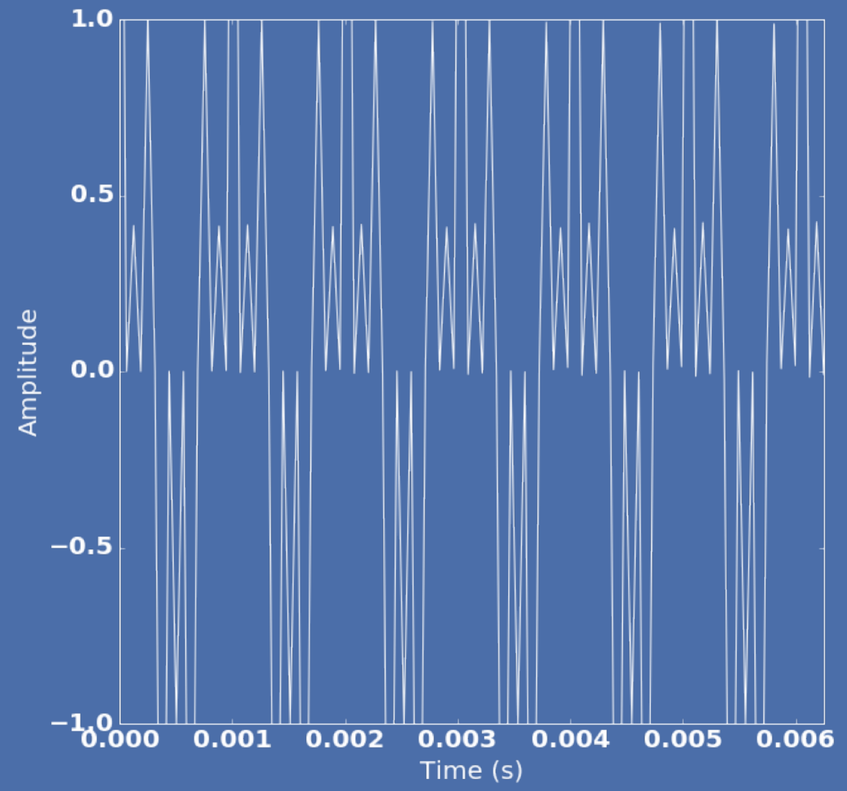


Waveform

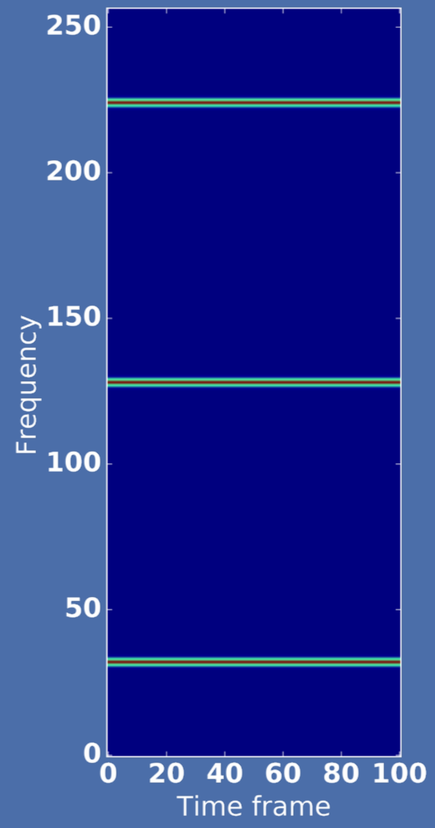


Spectrogram

Spectrogram: Sum of the three

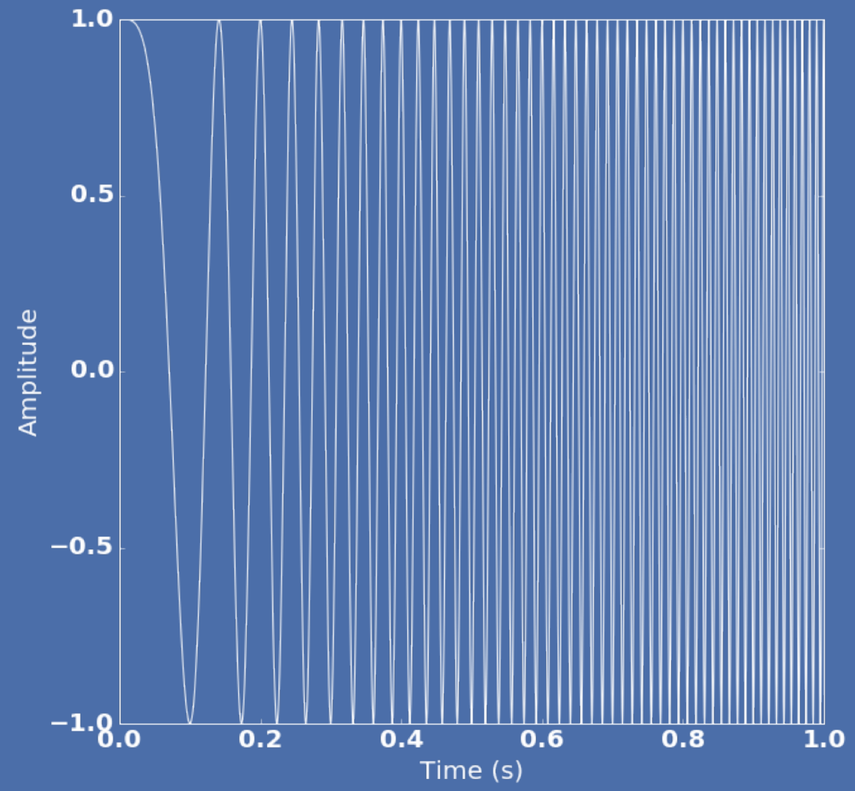


Waveform

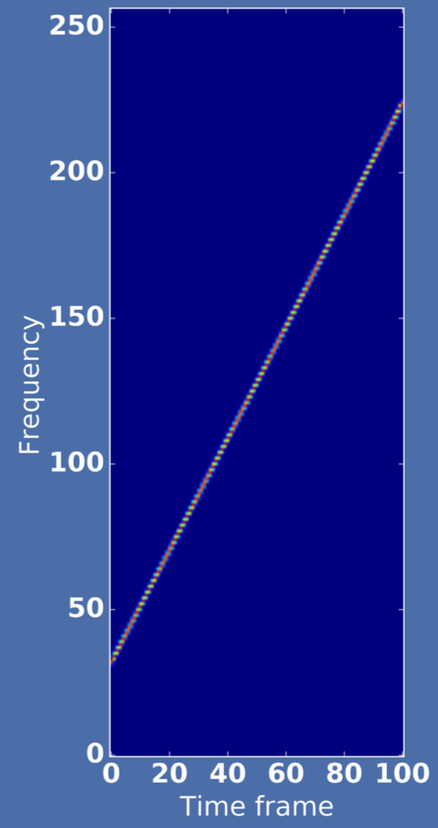


Spectrogram

Spectrogram: Chirp

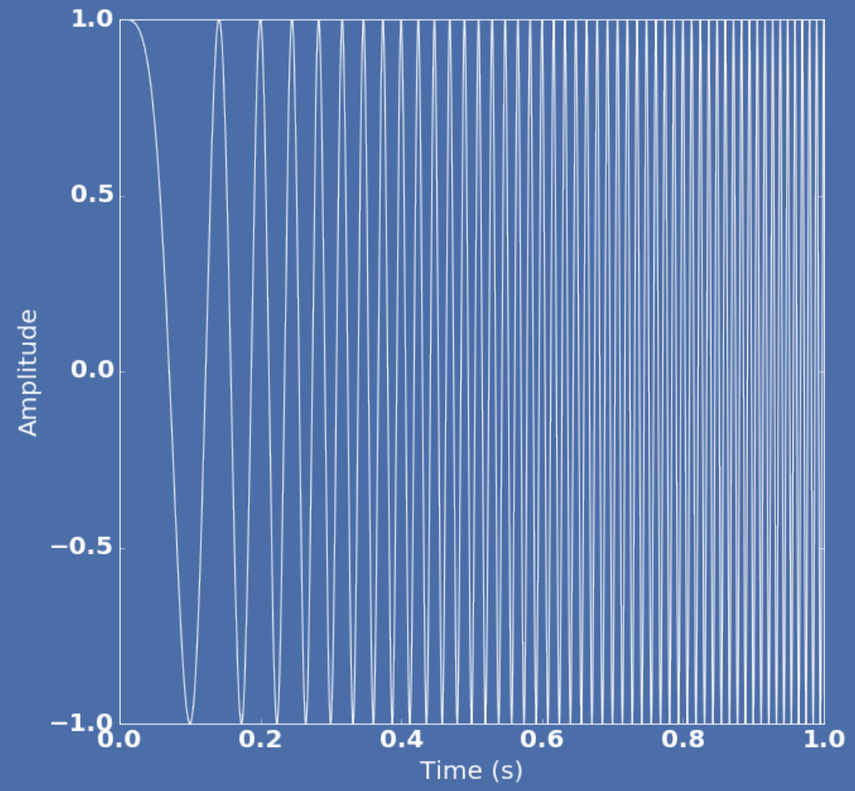


Waveform

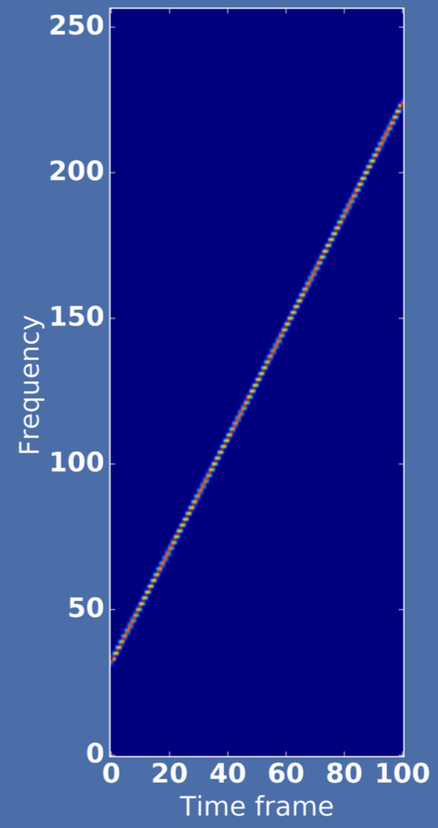


Spectrogram

Spectrogram: Chirp



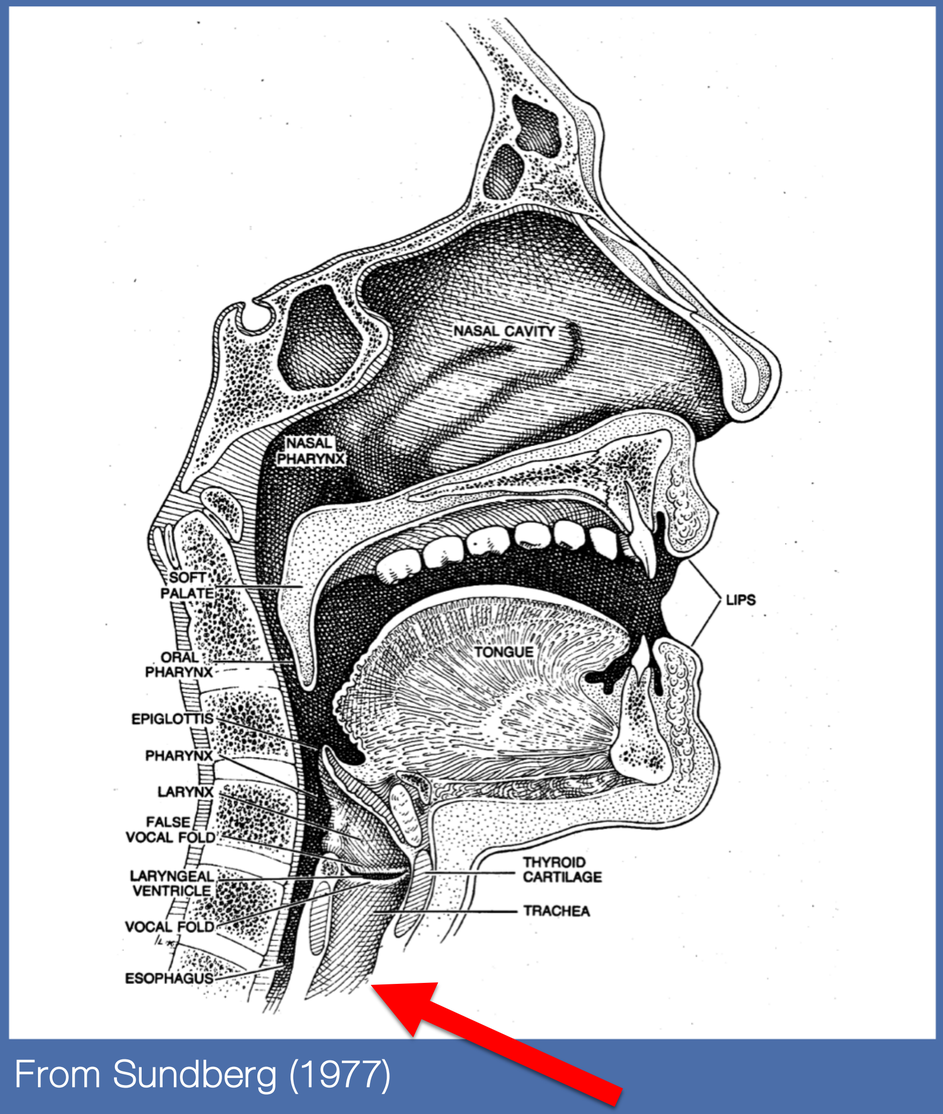
Waveform



Spectrogram

Speech production

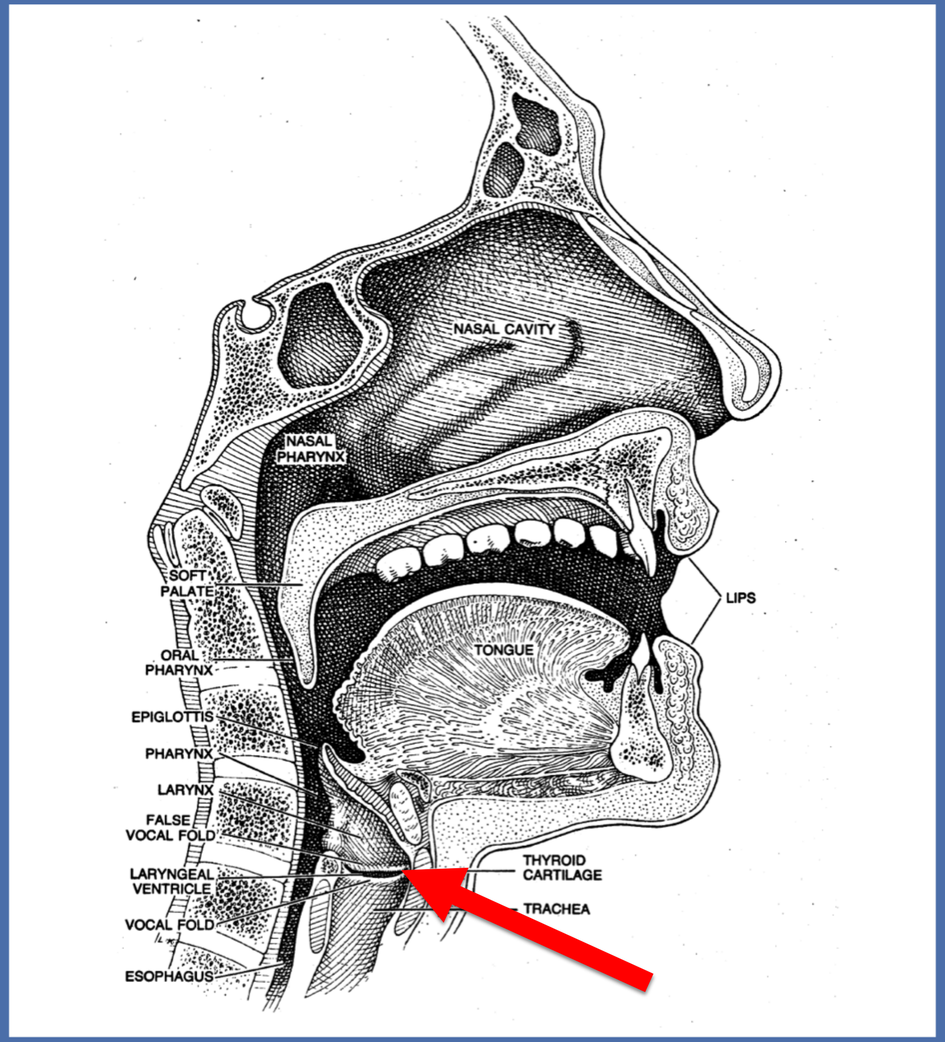
- Air is expelled from the lungs through the **trachea**



From Sundberg (1977)

Speech production

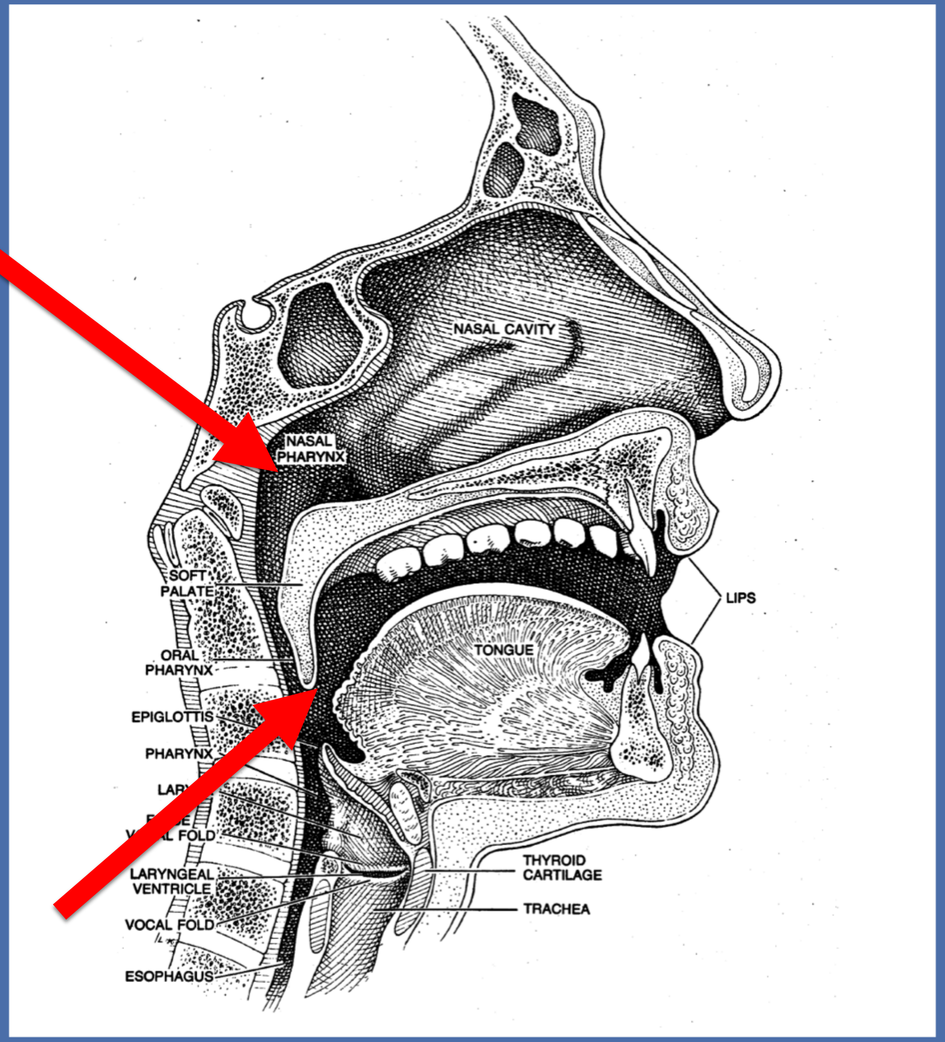
- Air is expelled from the lungs through the **trachea**
- Passes through **vocal folds**



From Sundberg (1977)

Speech production

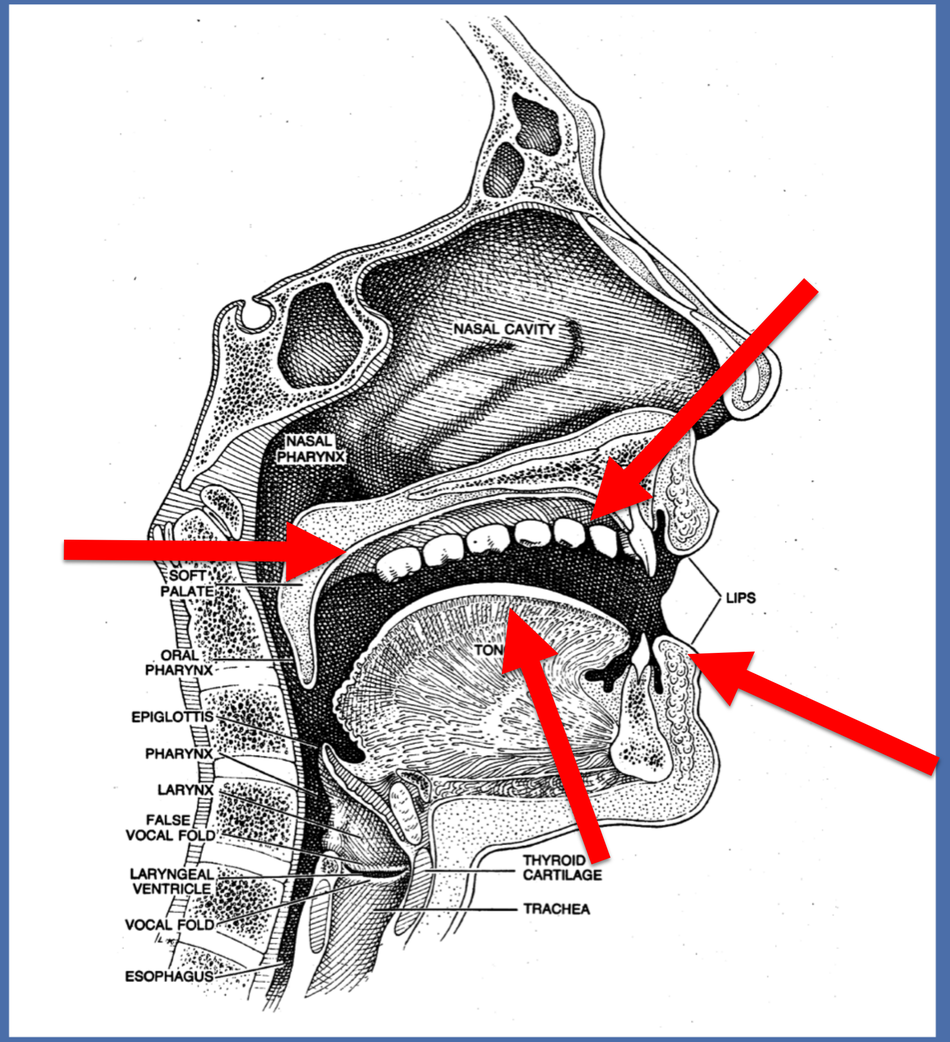
- Air is expelled from the lungs through the **trachea**
- Passes through **vocal folds**
- Then through the **nose** and/or the **mouth**



From Sundberg (1977)

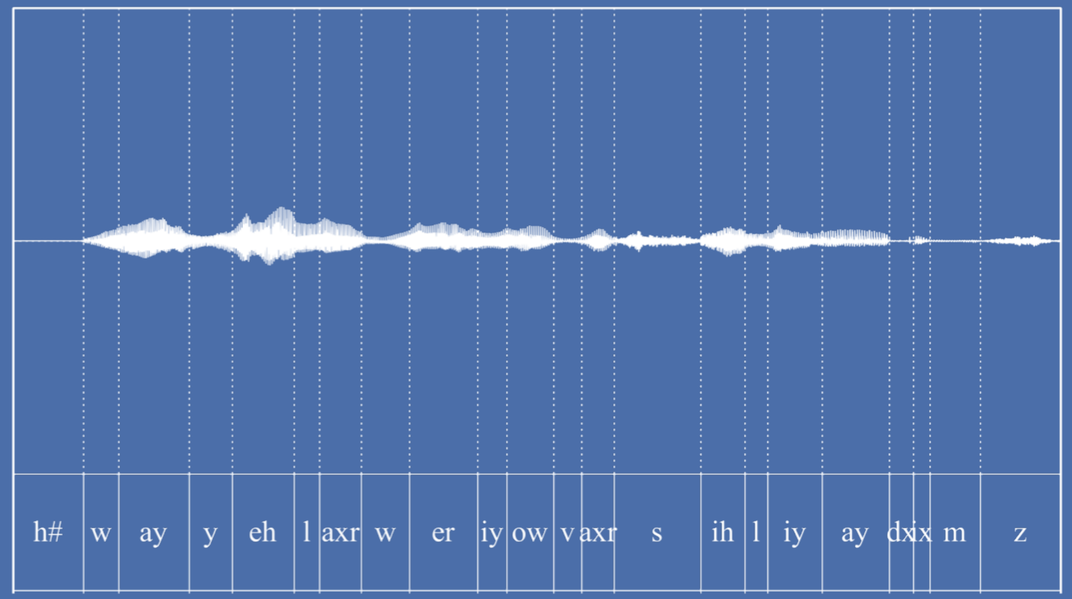
Speech production

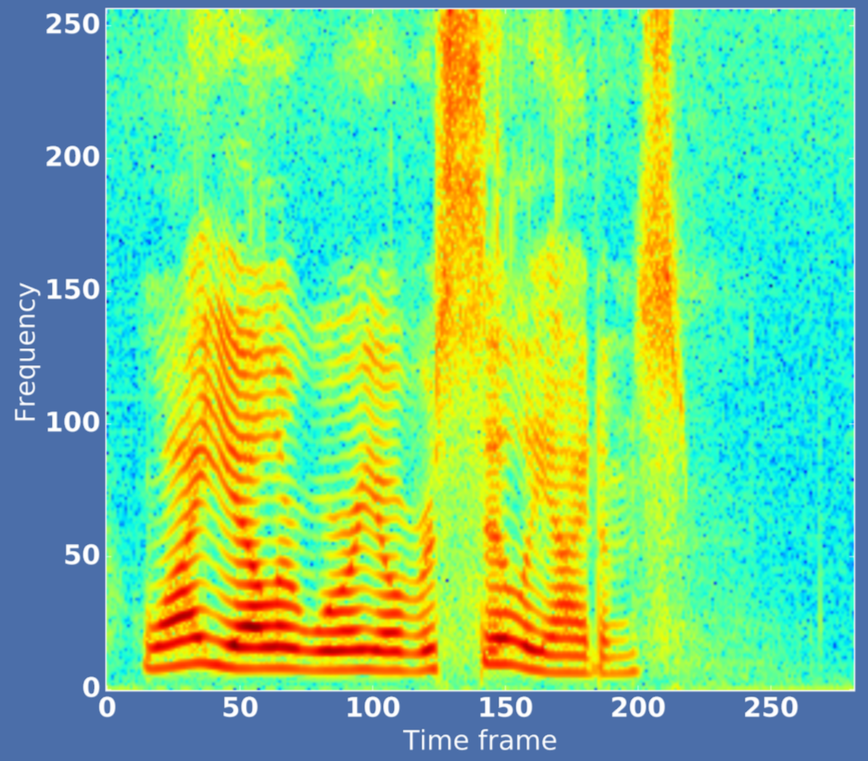
- Air is expelled from the lungs through the **trachea**
- Passes through **vocal folds**
- Then through the **nose** and/or the **mouth**
- Shaped by « articulators »: **lips**, **tongue**, **teeth**, **palate**, etc.
- The way we control the articulators defines the sound we produce



From Sundberg (1977)

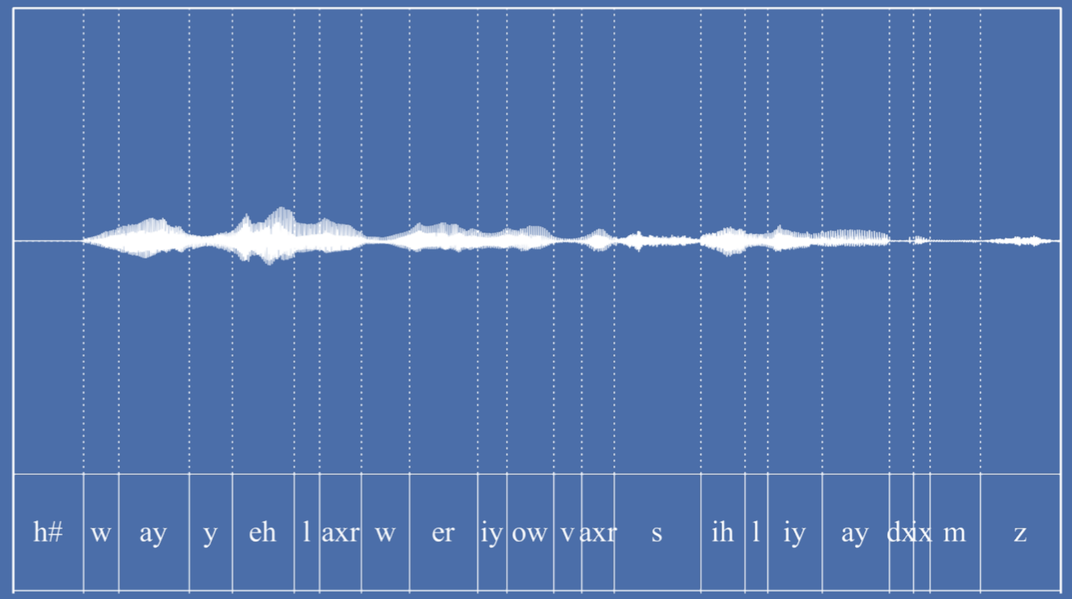
Spectrogram

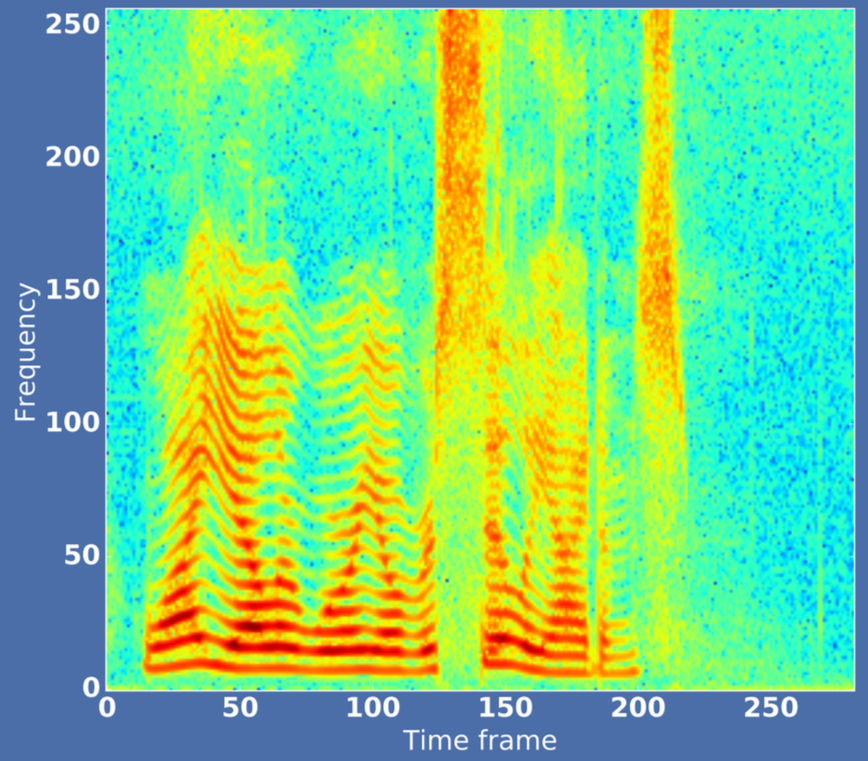




Correlated to phonetic content and local. With training, you can sometimes read spectrograms!

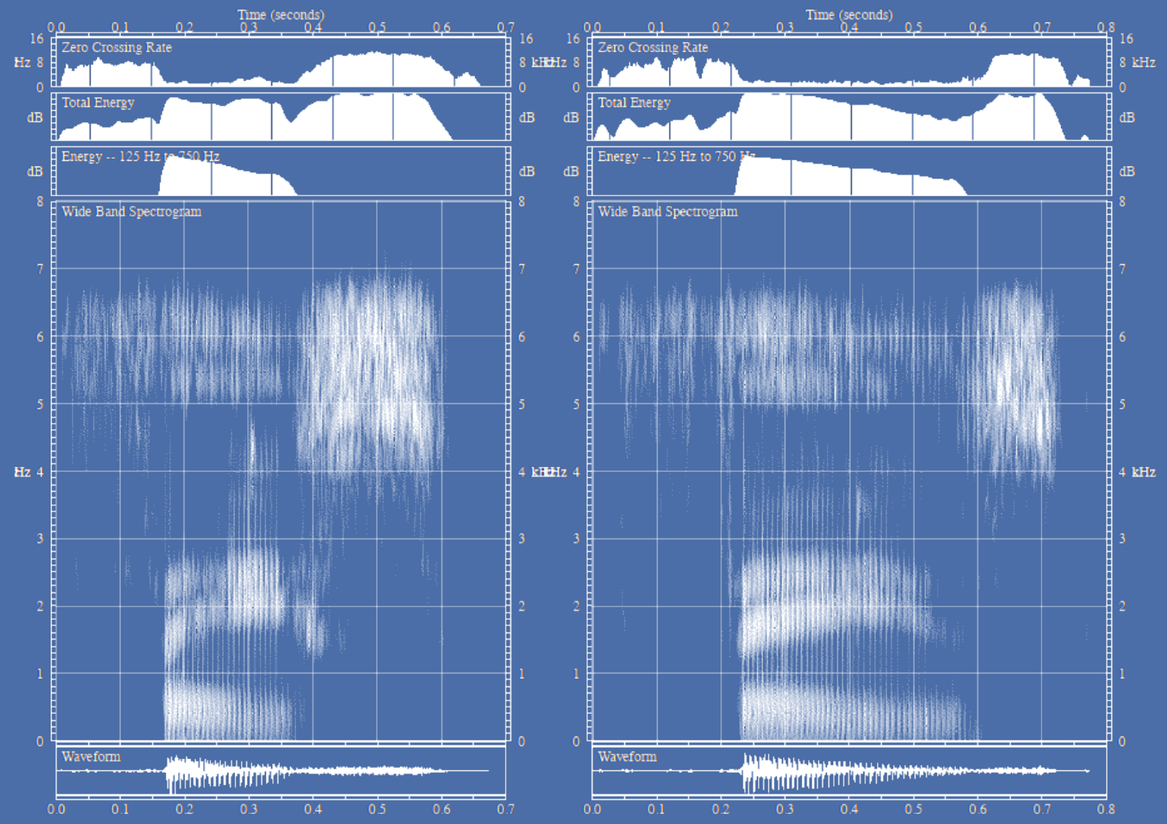
Spectrogram





Correlated to phonetic content and local. With training, you can sometimes read spectrograms!

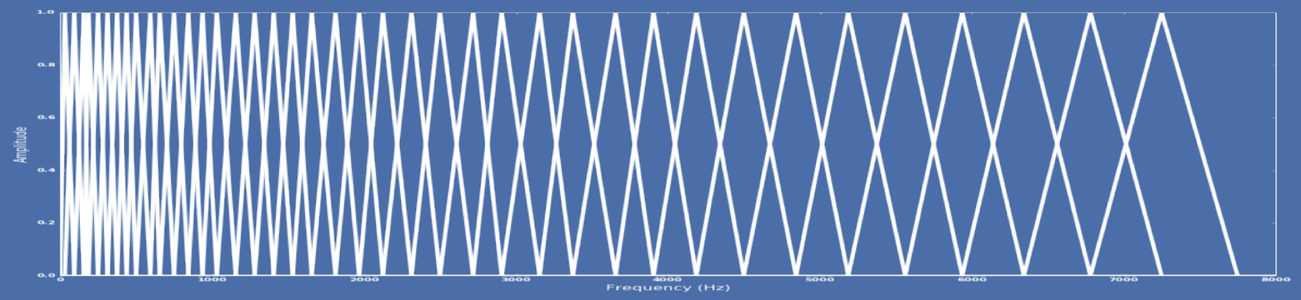
Spectrogram



FACE **PHASE**

Matching human perception: the mel-filterbanks

- Human ear not equally sensitive to all frequency bands -> we warp the frequencies to a scale which is **linear below 1000hz** and **logarithmic above 1000hz**.



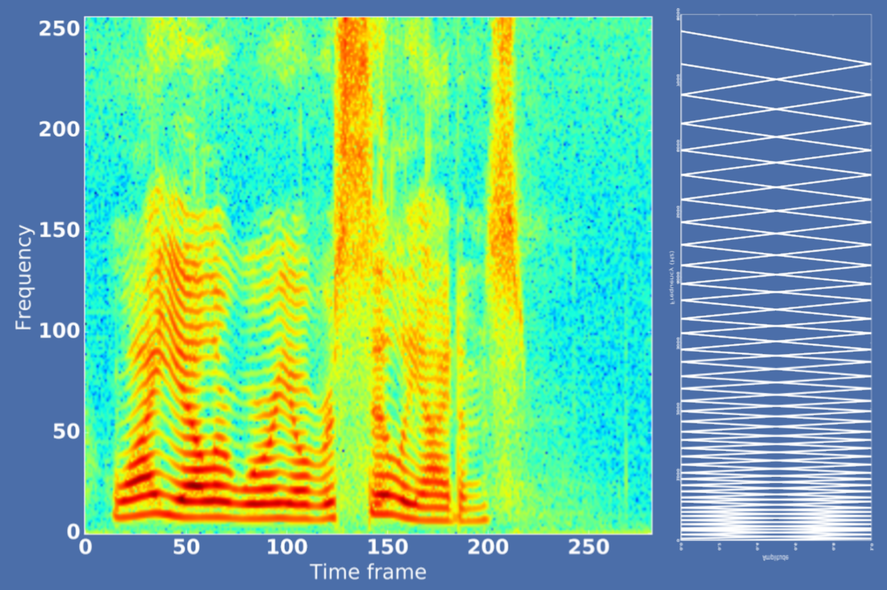
$$m = 2595 \cdot \log_{10}(1 + \frac{f}{700})$$

- We compute the product of our spectrum with each filter, then sum the results -> our features are now a vector of length n = number of filters
- We then take the log of this vector to reduce the dynamic range and variability linked to acquisition

Matching human perception: the mel-filterbanks

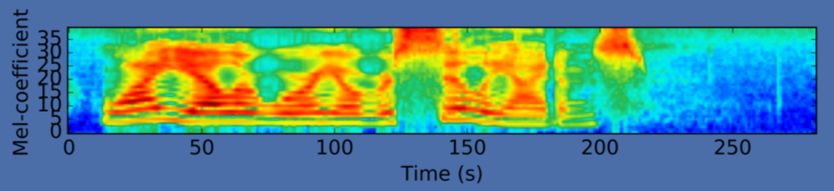
- Even more compact representation: ~6x fewer dimensions than spectrograms

$$Melfbank_j(k) = \sum_{\omega=0}^{256} Spectrogram(k,\omega) Melfilter_j(\omega)$$



Each frame has 257 dimensions

Mel-averaging

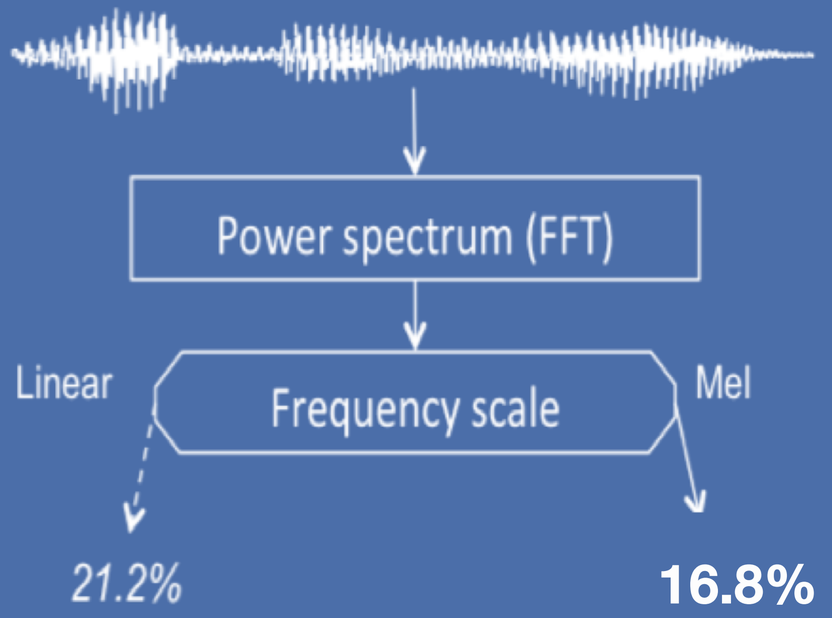


Each frame has 40 dimensions

The impact on phonetic classification

We compare error rates on a phonetic classification task, between linear scale spectrogram and mel-filterbanks.

0% means that the features allow a perfect separation of phonemes, 50% means that they are random for phonetic classification.



From mel-filterbanks to MFCC: the source-filter model

- The source-filter model makes the assumption that speech is the convolution between a glottal source and the vocal tract

$$x[n] = s[n] * v[n]$$

- A convolution in time domain is an elementwise product in frequency domain

$$X[\omega] = S[\omega]V[\omega]$$

$$|X(\omega)|^2 = |S(\omega)|^2|V(\omega)|^2$$

$$\log(|X(\omega)|^2) = \log(|S(\omega)|^2) + \log(|V(\omega)|^2)$$

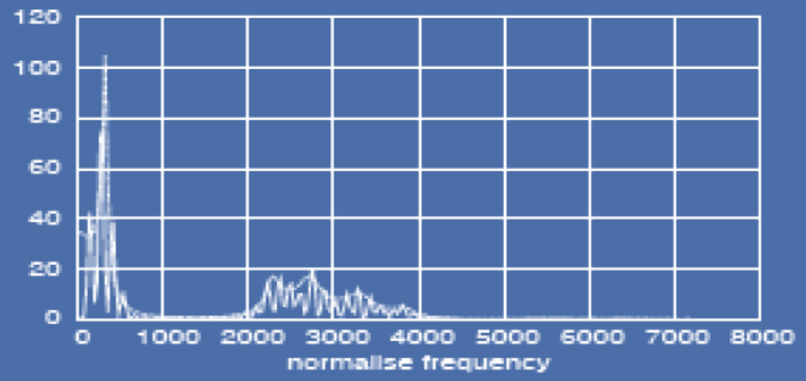
From spectrum to cepstrum: the Discrete Cosine Transform

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

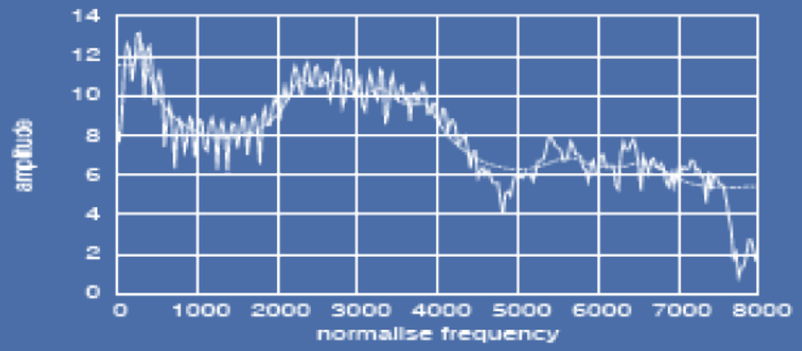
- Somehow similar to a Fourier Transform: projects the signal from the spectral to the **cepstral** domain, where glottal source and vocal tract are well separated
- Uncorrelated coefficients (important property for the modelling)

Cepstral representation

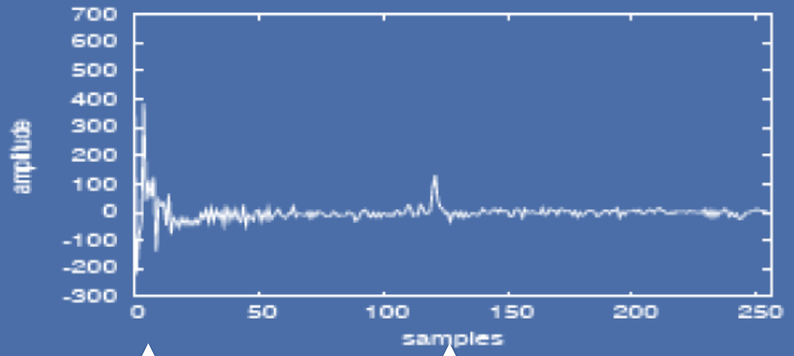
Spectrum



Log-spectrum

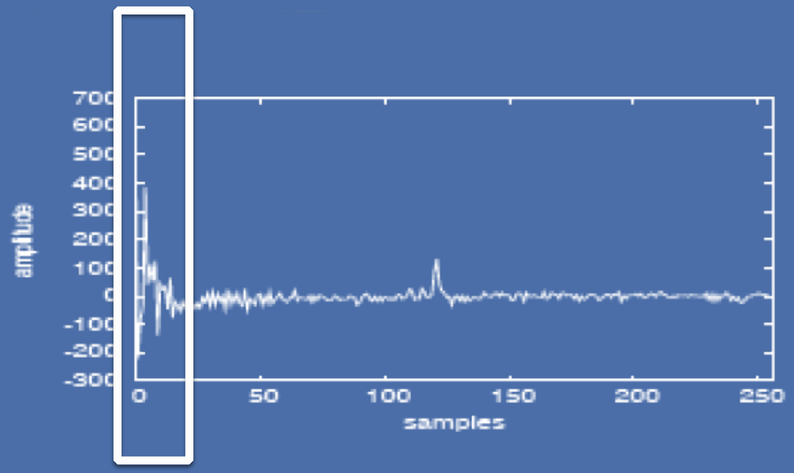


Cepstrum



Vocal tract Glottal source

The Mel-Frequency Cepstral Coefficients



- Take the first 12 cepstral coefficients
- Concatenate the log energy (13 coefficients)
- Concatenate first and second derivatives
- 39-dimensional coefficients

$$\Delta(n) = \frac{c(n+1) - c(n-1)}{2}$$

$$\Delta\Delta(n) = \frac{\Delta(n+1) - \Delta(n-1)}{2}$$

Question

- Some languagues such as Mandarin are **tonal**: depending on the pitch at which it is pronounced, a phoneme is different.

Question

- Some languagues such as Mandarin are **tonal**: depending on the pitch at which it is pronounced, a phoneme is different.

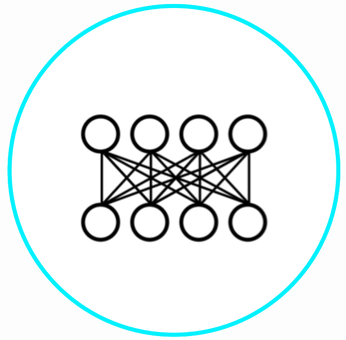
- **To do speech recognition in such languages, which features would you use?**
- **1. Mel-filterbanks**
- **2. MFCC**
- **3. Other**

Go to: https://api.socrative.com/rc/KBcrE4 and login with first name name mail

Speech features in practice

- Mel-filterbanks: speech recognition
- MFCC: speaker recognition
- Praat: http://www.fon.hum.uva.nl/praat/
- Kaldi: http://kaldi-asr.org/
- HTK: http://htk.eng.cam.ac.uk/
- spectral.py: https://github.com/bootphon/spectral
- Librosa: https://librosa.github.io/librosa/

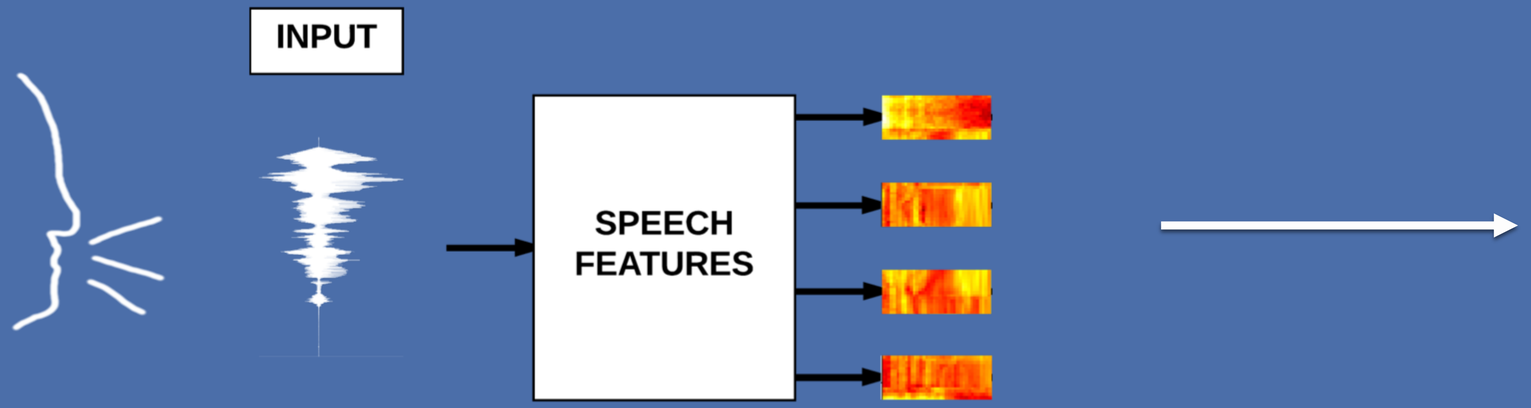
- Further reading: « Speech and Language Processing » Jurafsky and Martin, Chapter 9.3



Acoustic modelling

- Statistical modelling of ASR
- Gaussian Mixture Models
- Deep Neural Networks

Speech recognition as a statistical problem



“MY NAME IS NEIL”

$$X_T = x_1, ..., x_T$$

A sequence of features frames

$$W_M = w_1, ..., w_M$$

A sequence of words

Speech recognition as a statistical problem

$$X_T = x_1, ..., x_T$$

A sequence of features frames

$$W_M = w_1, ..., w_M$$

A sequence of words

- Goal of ASR:

$$\hat{W} = \arg\max_W P(W|X)$$

Dividing the problem

$$\hat{W} = \arg\max_{W} P(W|X)$$

- By Bayes formula:

$$\hat{W} = \arg\max_{W} P(X|W)P(W)$$

Summing on the pronunciations

- Final goal: Modelling the likelihood of features frames X conditioned on the transcription

$$P(X|W)$$

- We can condition on sequences of phonemes instead of sequences of words

$$q_{1:K_w}^{(w)} = q_1, \ldots, q_{K_w}$$

Valid pronunciation for word w: sequence of phonemes

$$P(X|W) = \sum_Q P(X|Q)P(Q|W)$$

Sum over all valid pronunciations

What model do we choose?

$$P(X|Q)$$

- We want to model a sequence of features, conditioned on a sequence of symbols
- A natural way of modelling it is by using a **Hidden Markov Model**

Hidden Markov Model: quick recap

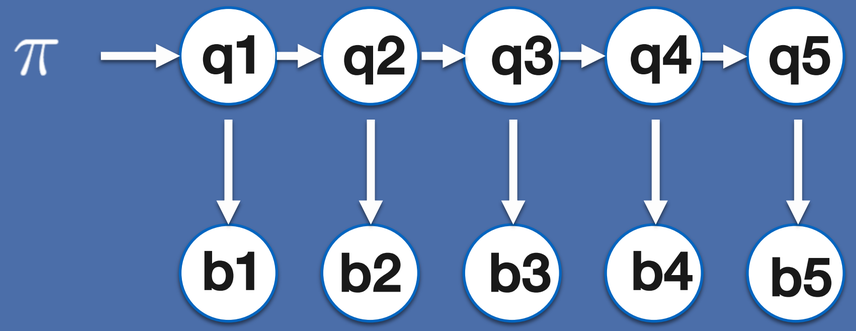
$$S = \{s_1, ..., s_N\} \text{ with } N\text{: number of states} \quad q_t \in S$$

$$A = \{a_{ij}\} \qquad a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$$

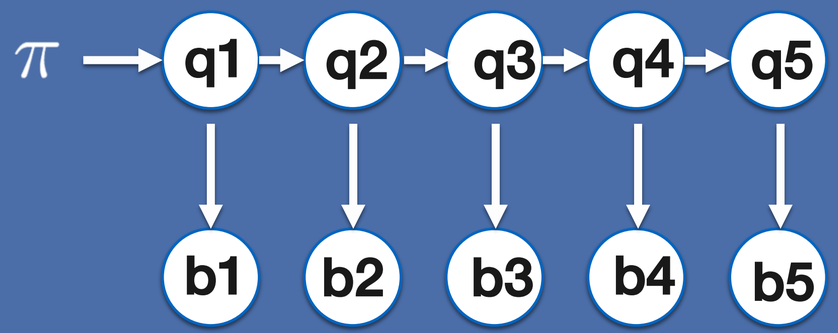
$$B = \{b_j\}$$

$$\pi_i = P(q_1 = s_i) \qquad \pi = \{\pi_i\}$$

$$\theta = \{A, B, \pi\}$$



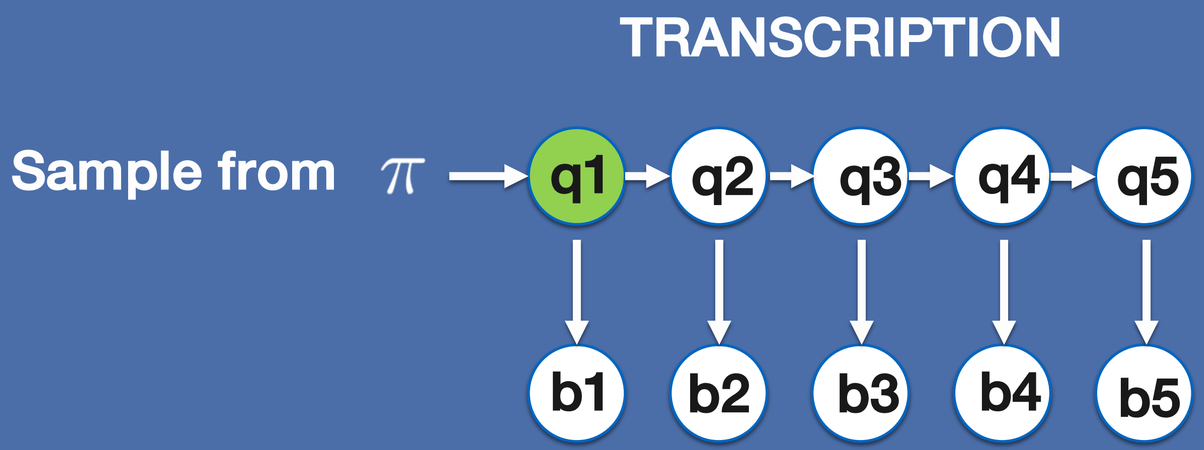
Hidden Markov Models for Speech Recognition



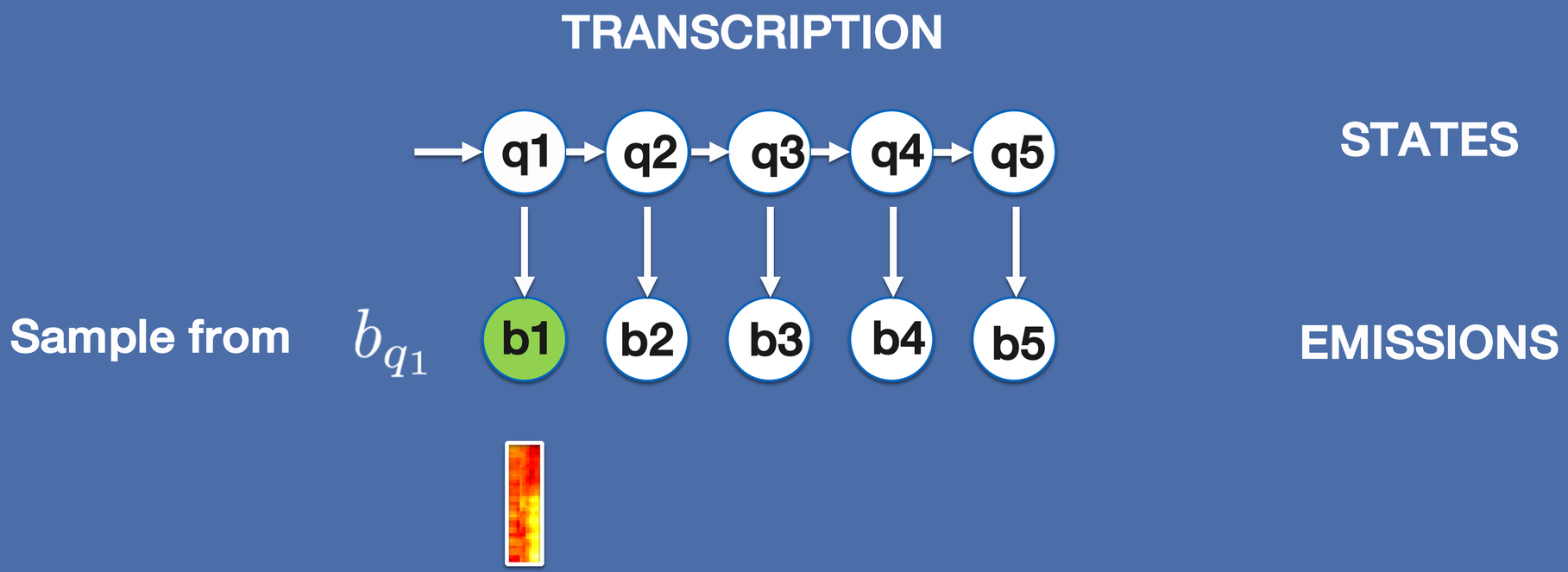
STATES = TRANSCRIPTION

EMISSIONS = FEATURES

Hidden Markov Models for Speech Recognition

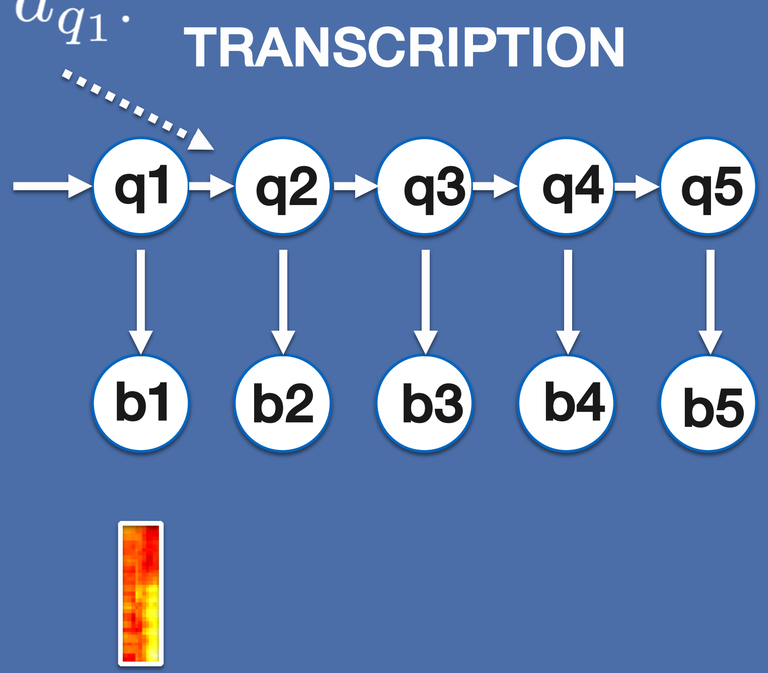


Hidden Markov Models for Speech Recognition

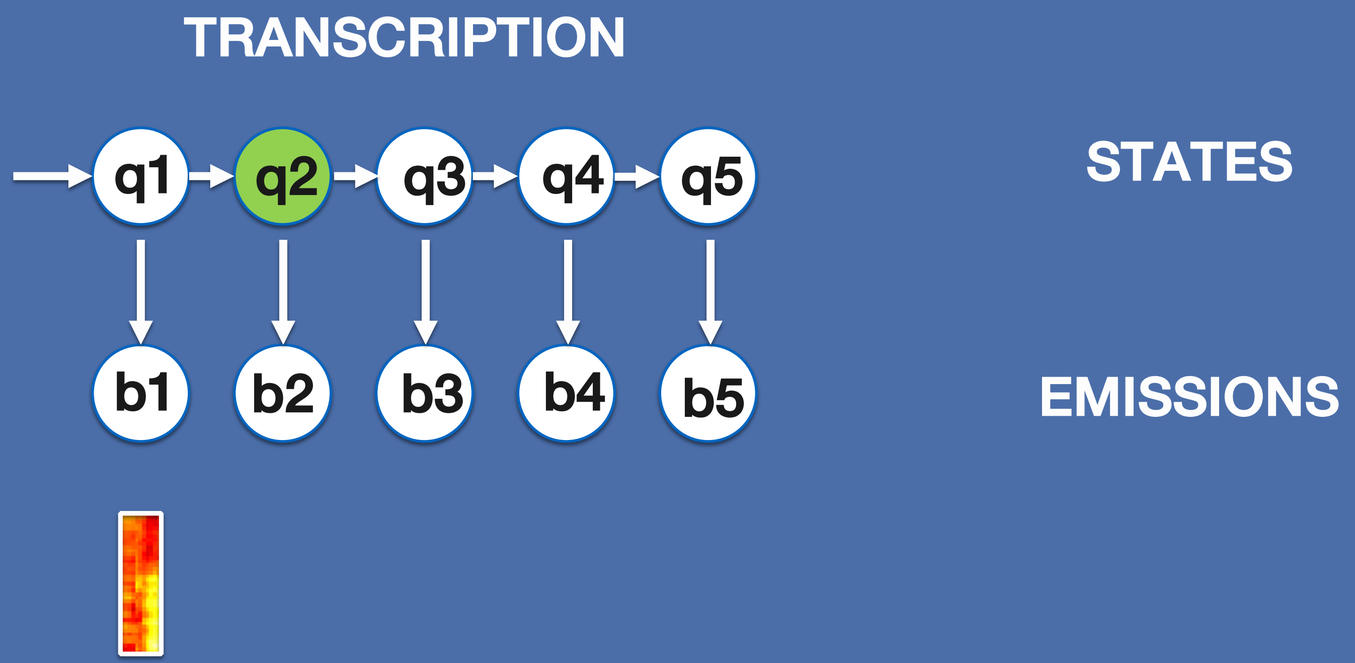


Hidden Markov Models for Speech Recognition

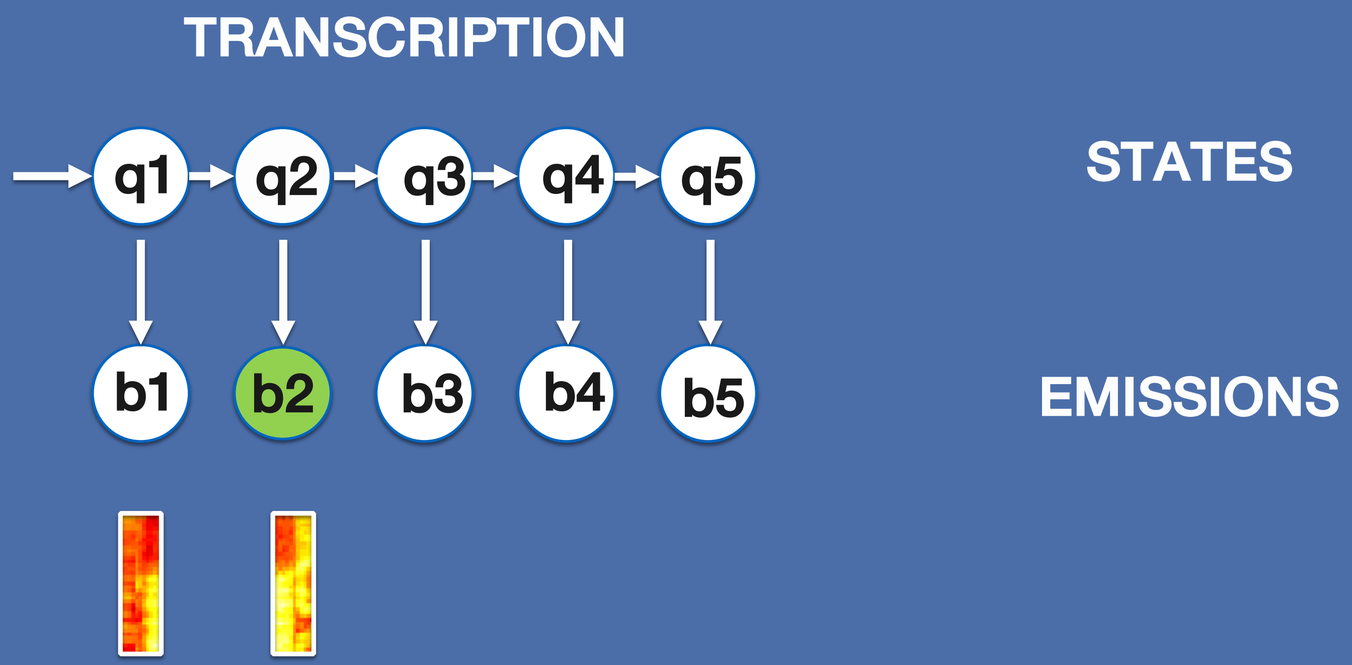
Sample from a_{q_1}.



Hidden Markov Models for Speech Recognition



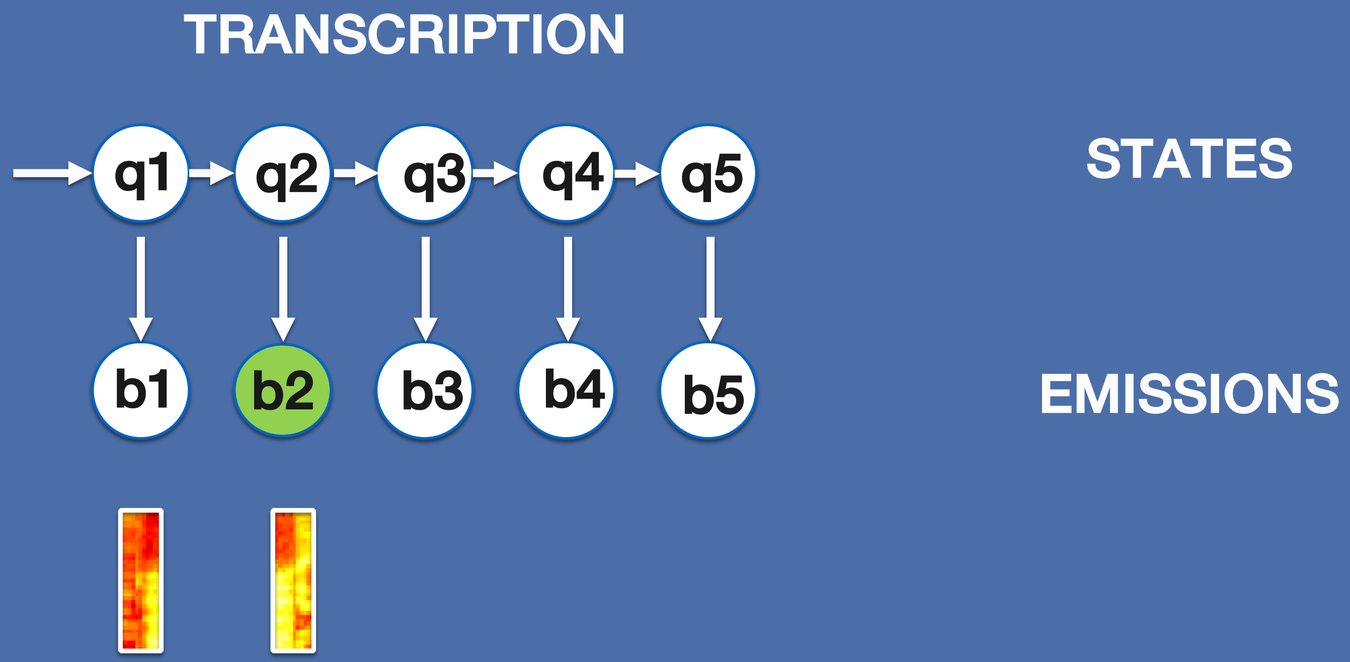
Hidden Markov Models for Speech Recognition



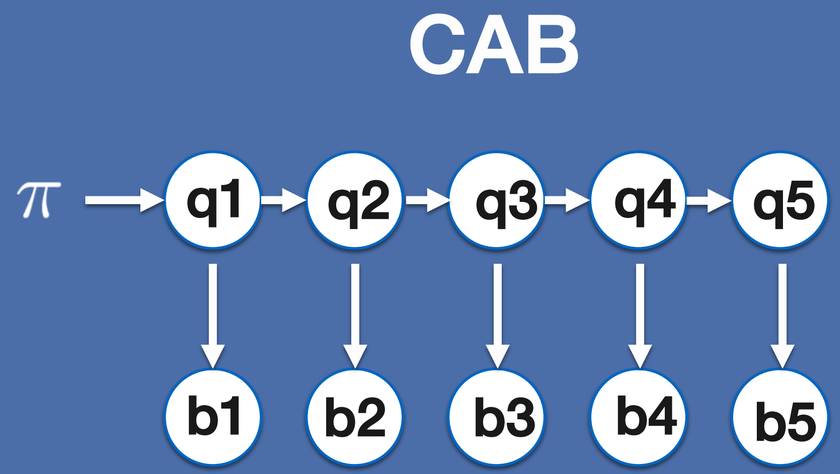
Hidden Markov Models for Speech Recognition

What transcription should we choose? Words, phonemes?

What distribution can model the speech features?

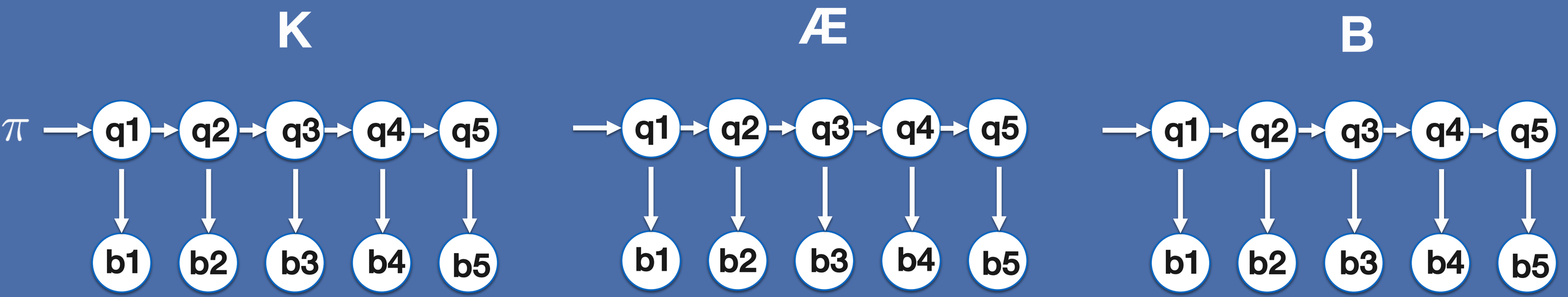


Word-based HMMs



- **Adapted for small vocabulary (like voice commands), but not in general**
- **Rare words will only see a few examples**

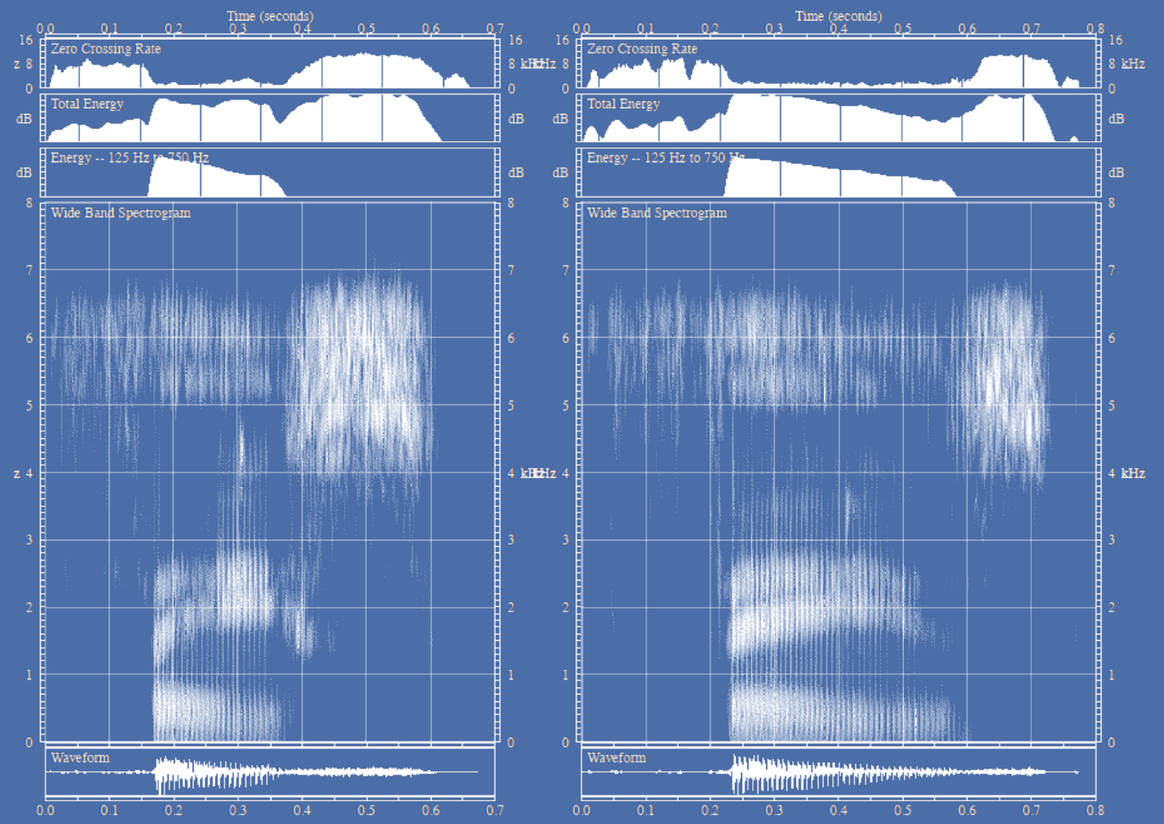
Phone-based HMMs



- **Lots of examples for each phoneme**
- **For a given pronunciation, a word is a concatenation of phone HMMs**

The problem of co-articulation

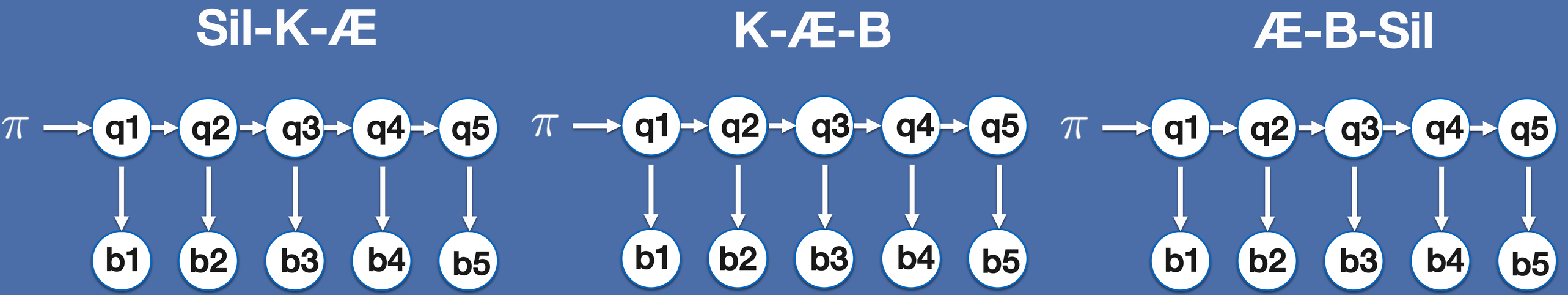
- The way phones are pronounced depends on the previous phone **but also on the next one**
- Breaks the conditional independence assumption of HMMs
- Solution: triphone-based HMM



FACE
/feɪs/

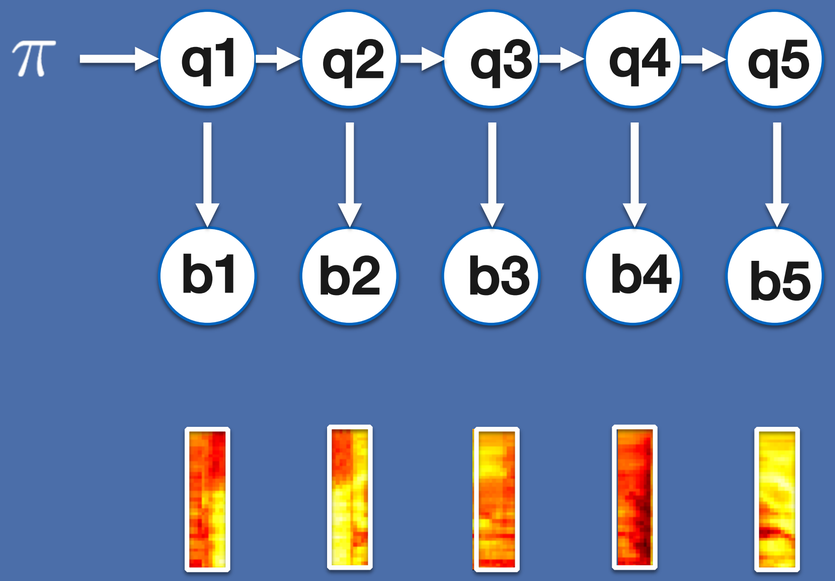
PHASE
/feɪz/

Triphone-based HMMs



- **Takes into account co-articulation with previous and next phone**

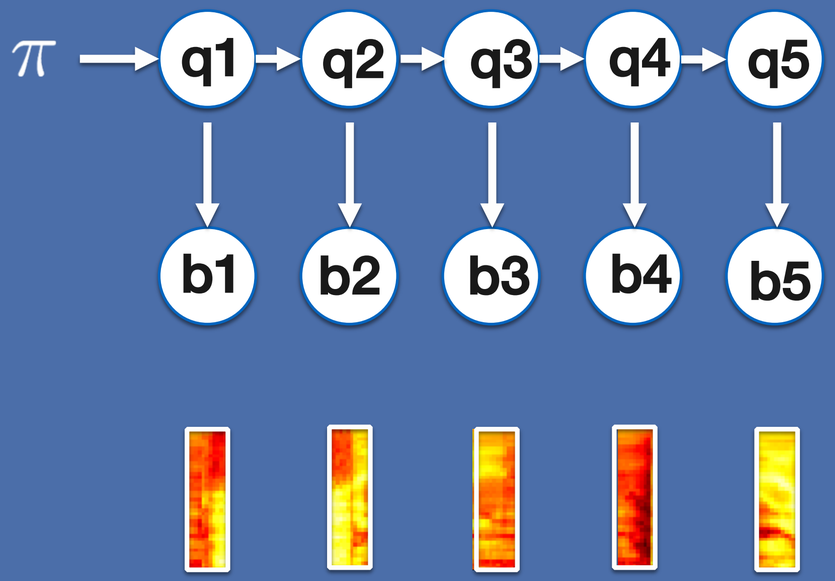
Modelling the feature space



$$B = \{b_j\}$$

is a set of distributions over the feature space. Here j is the value taken by the state. What family of distributions should we choose?

Gaussian Acoustic Model

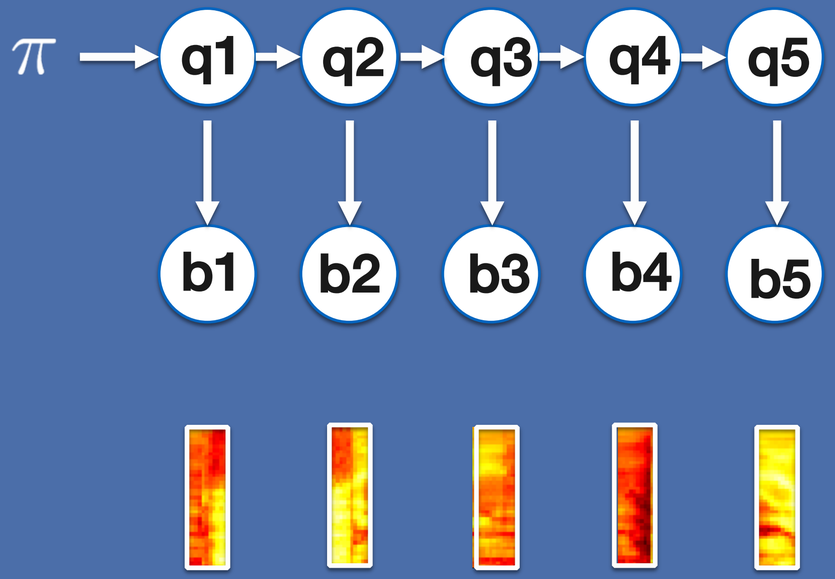


$$b_j \sim \mathcal{N}(\mu_j, \Sigma_j)$$

Models each state as a Gaussian distribution, general model but constrained by assumptions that the distribution is unimodal.

Since each dimension of MFCCs is independent from the others, we can use diagonal covariance matrices which considerably reduces the computational complexity.

Gaussian Mixture Acoustic Model

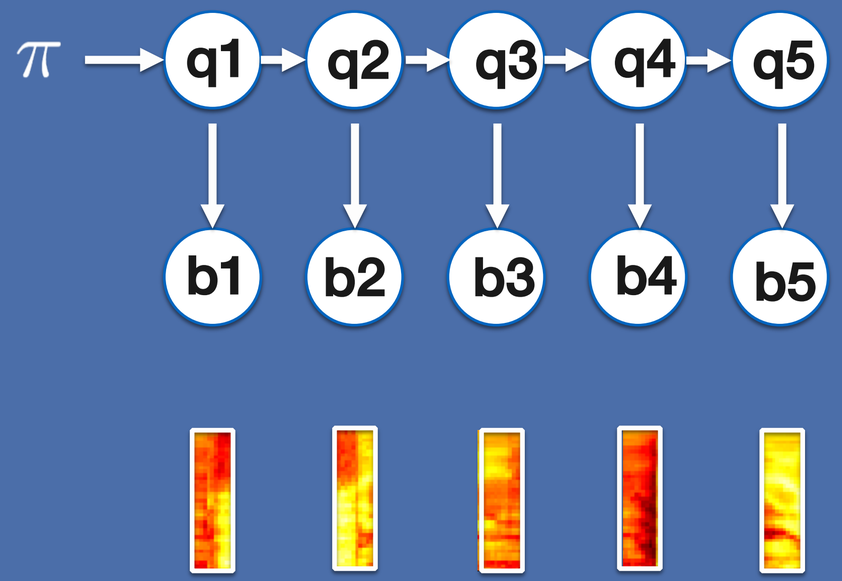


$$b_j \sim \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm})$$

Models each state as a Gaussian Mixture, universal approximator

Summary of the model

K-Æ-B



$$b_j \sim \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm})$$

Three Hidden Markov Model algorithms

- **Scoring**: Given a sequence of features X1,…,Xt how can we evaluate P(X|Theta) -> The forward-backward algorithm
- **Matching**: Given a sequence of features X how can we find the optimal sequence of states (that will yield the transcription)? -> the Viterbi algorithm
- **Training**: How do we adjust the model parameters Theta (transitions and emissions distributions) to maximize P(X|Theta)? -> the Baum-Welch algorithm

See more in the Probabilistic Graphical Models class

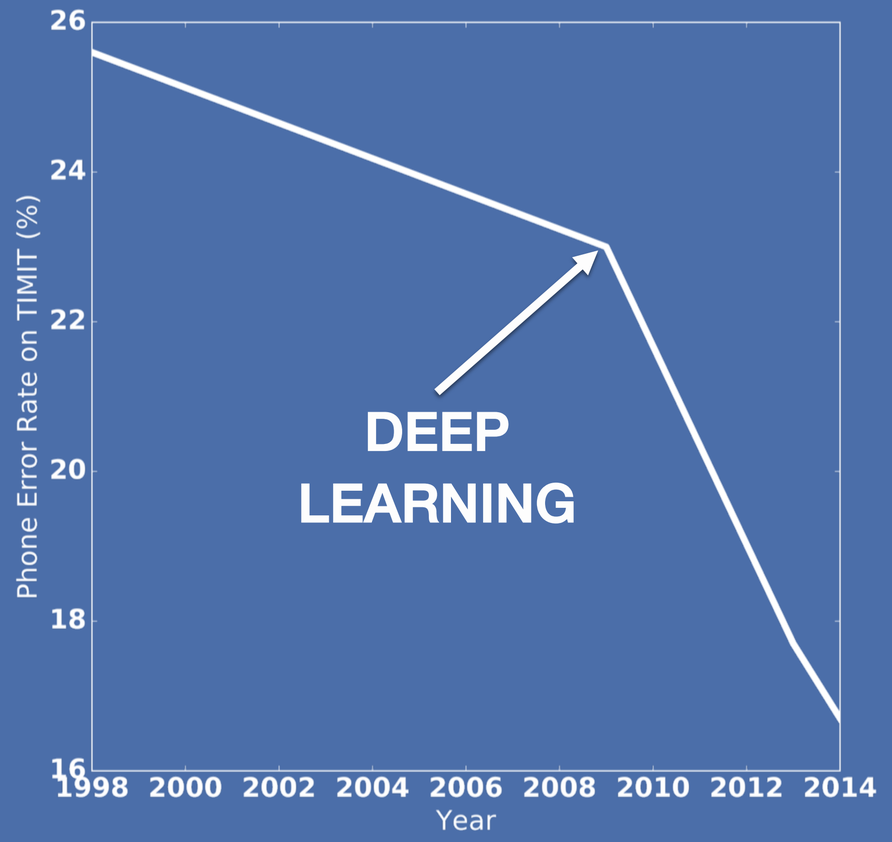
Hidden Markov Models in practice

- Train your model on a big annotated corpus to learn the transitions and the parameters of the distributions
- When given a new, unwritten, sentence at test time: use Viterbi to find the most likely symbol sequence

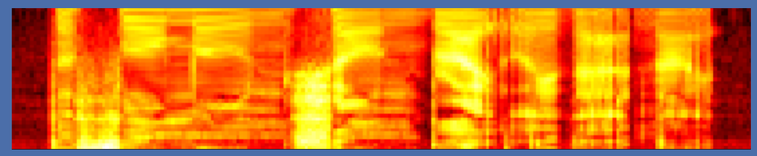
From Gaussian Mixture Models to Neural Networks

- After training your model you can use Viterbi to find the optimal sequence of states conditioned on the features
- You can train a deep neural network to predict the state from the features
- **Universal approximator**
- **No need of uncorrelated dimensions**
- **No need of modelling feature frames individually, you can use a wide context around the frame you want to classify**

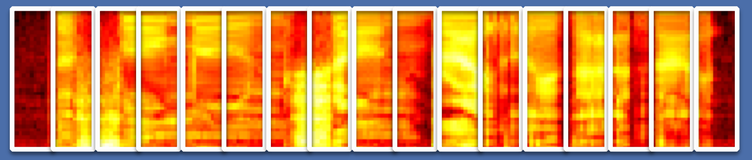
Deep speech recognition



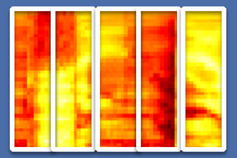
Deep Neural Network Acoustic Model



Deep Neural Network Acoustic Model

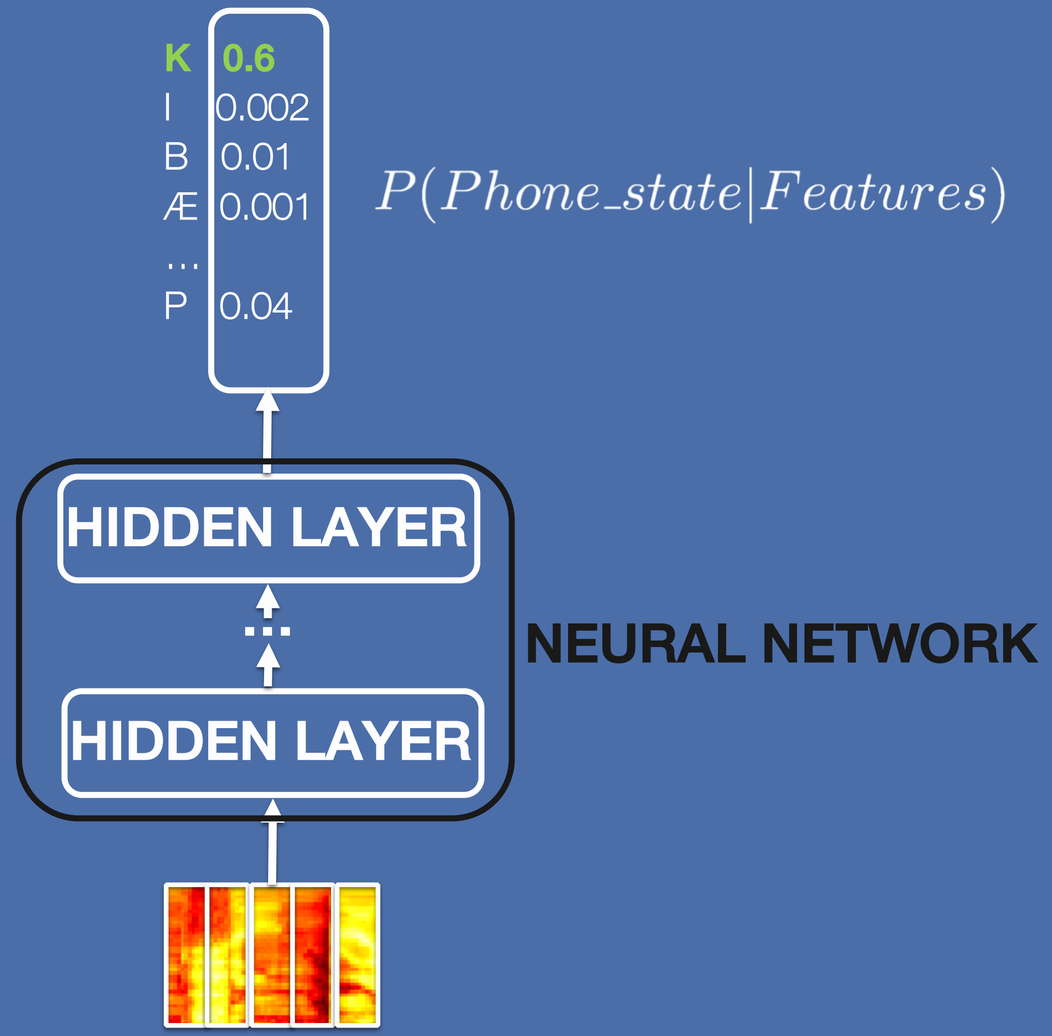


Deep Neural Network Acoustic Model

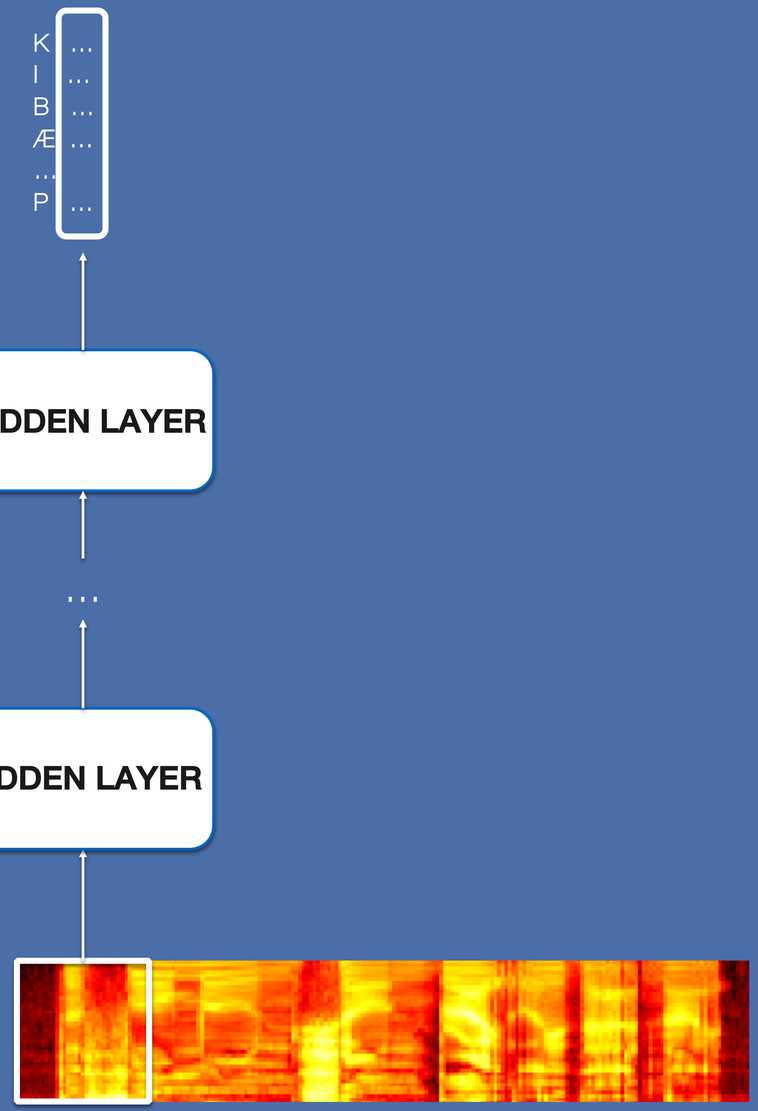


Deep Neural Network Acoustic Model

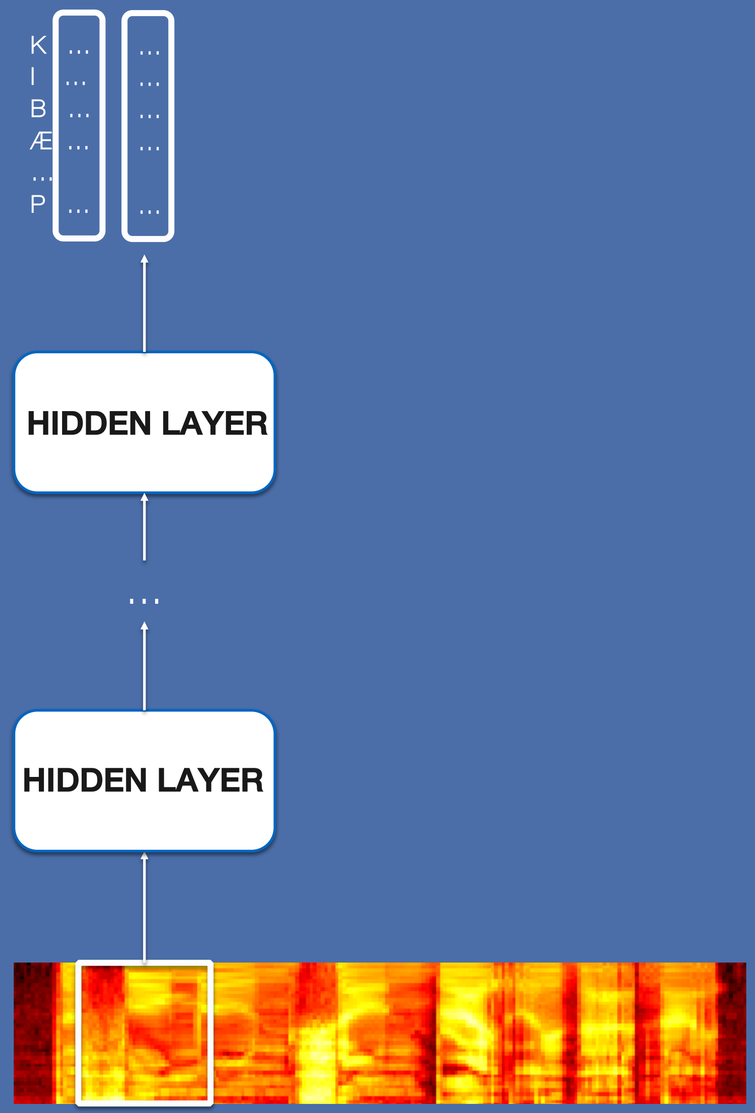
- **Deep acoustic models:**
  The neural network takes features as input and outputs probabilities over phone states



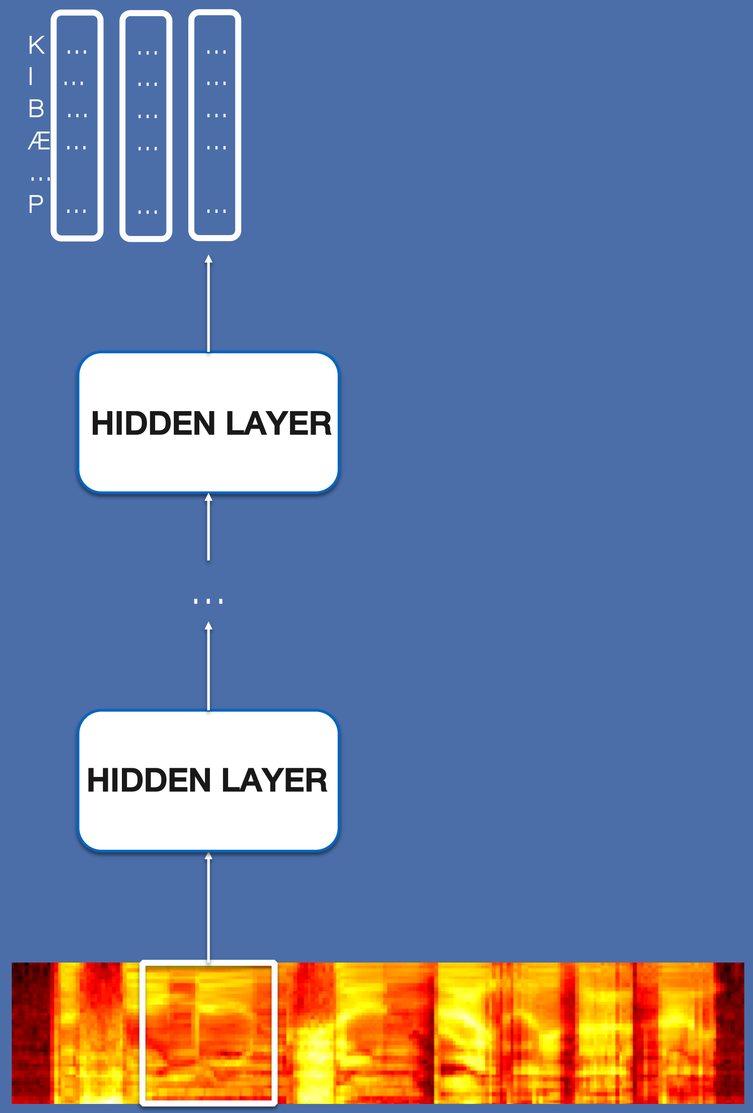
Deep Neural Network Acoustic Model



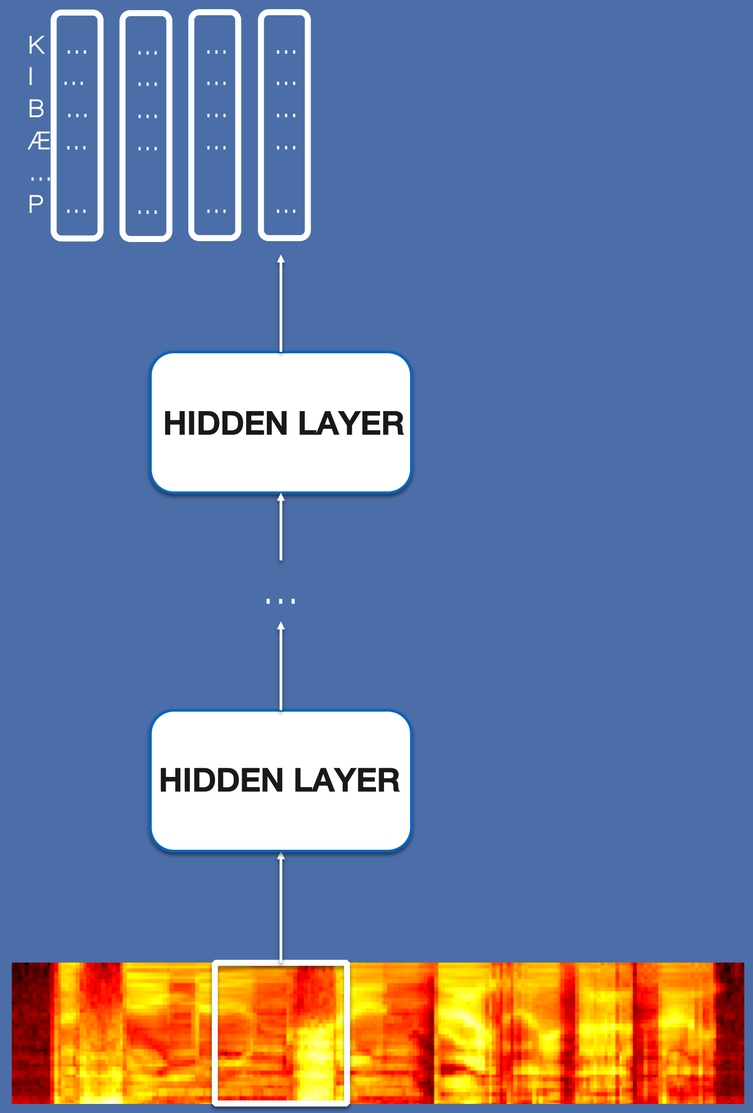
Deep Neural Network Acoustic Model



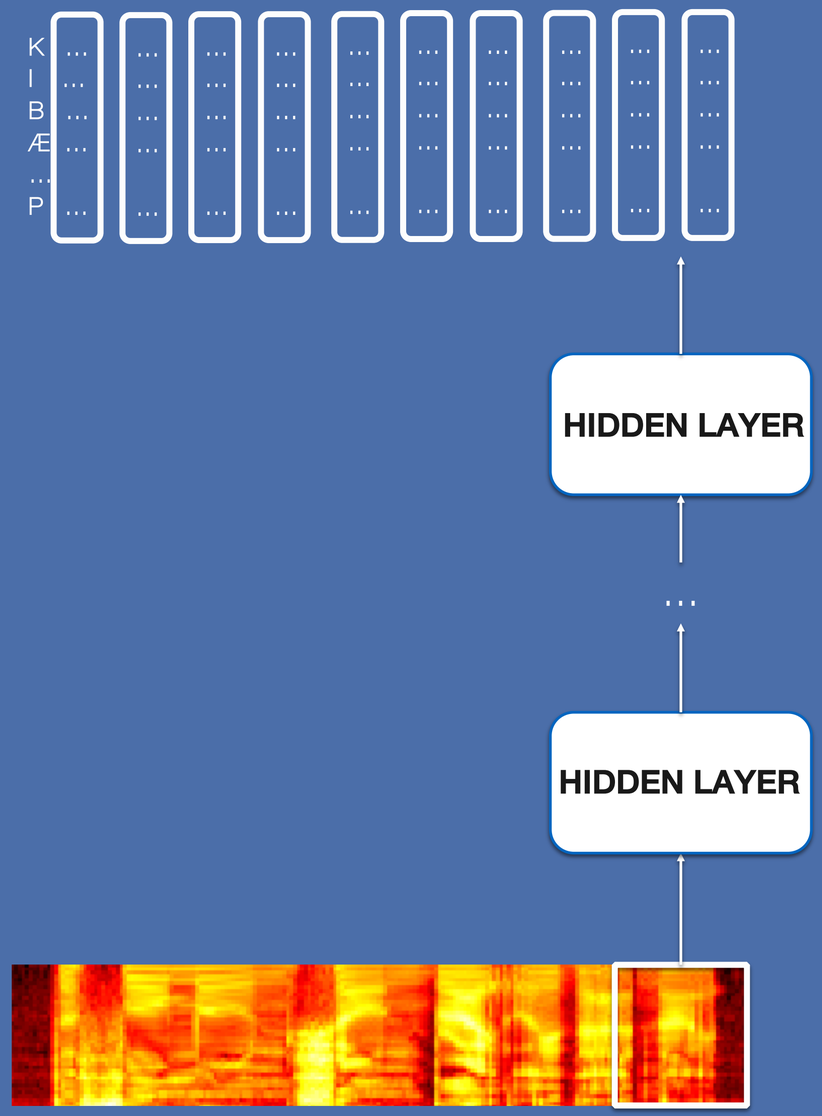
Deep Neural Network Acoustic Model



Deep Neural Network Acoustic Model



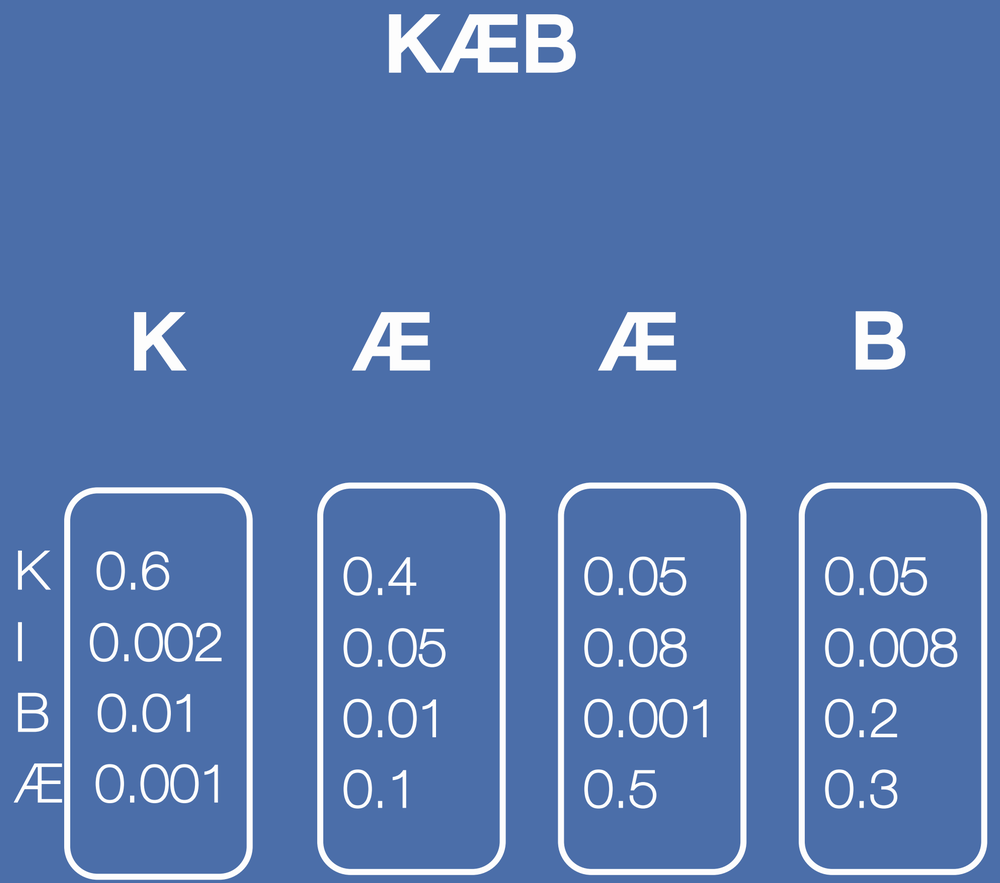
Deep Neural Network Acoustic Model



Classification loss

Labels are the phone states given by Viterbi

Train with a cross-entropy loss by backpropagation



Training pipeline

- Train a Hidden Markov Model with Gaussian Mixture Acoustic Model
- Use Viterbi to align every feature frame in your data with a state
- Train a deep neural network to predict the state from the features
- **Next class: no need of this anymore -> End-to-end training**

Handling variability in speech

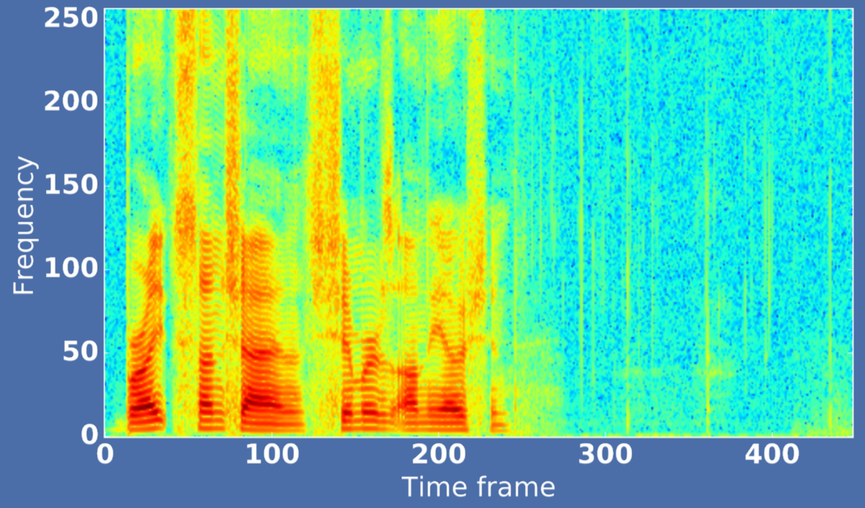
- Gender
- Speaker identity
- Noise

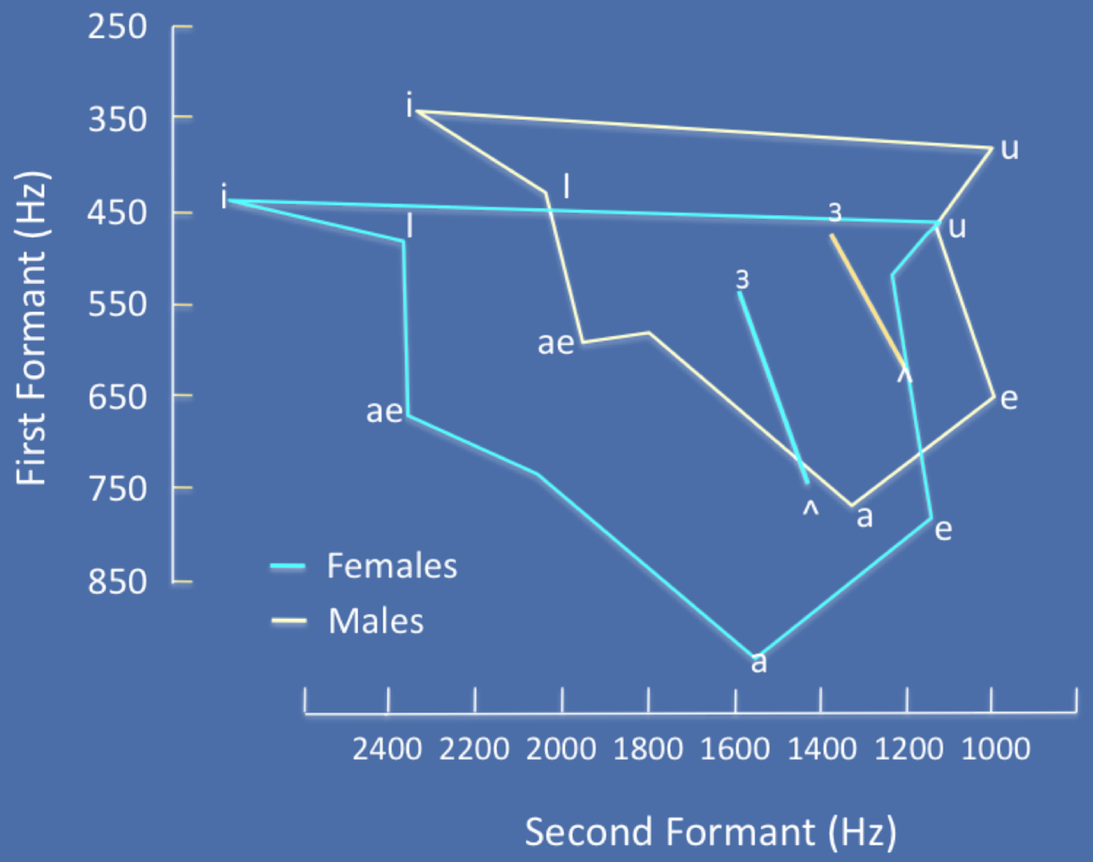
Dealing with variations in gender and age

- Female voices have a higher pitch than male voices, and children's voices have a higher pitch than female voices
- There are also deformations in the formants (spectrogram patterns that characterize phonemes)

Dealing with variations in gender and age

- Formants: main resonances that characterize the phonetic content

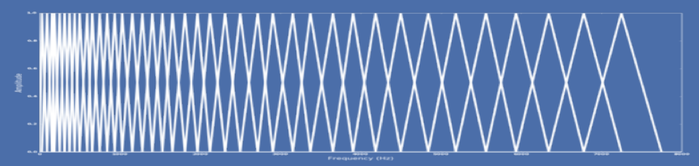


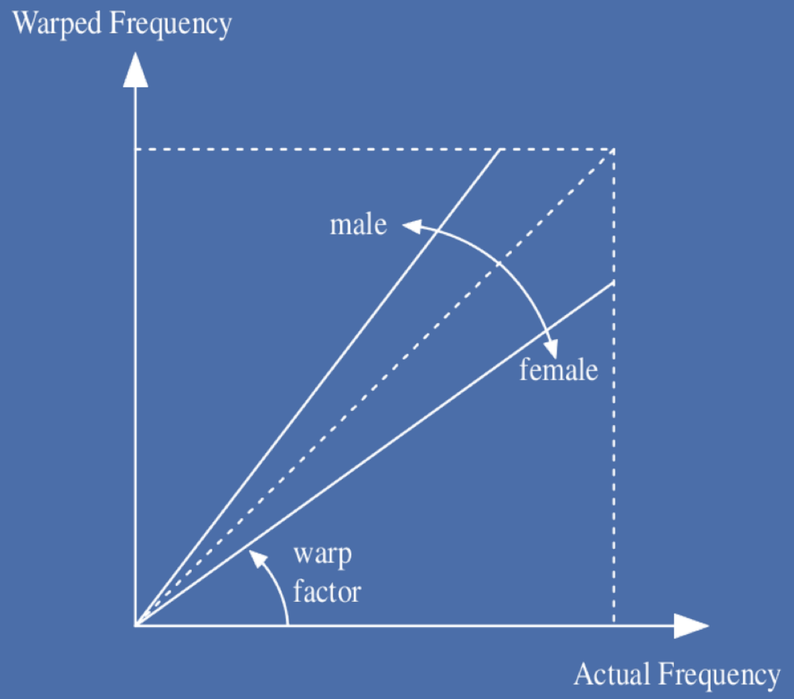


Learning invariance from the data

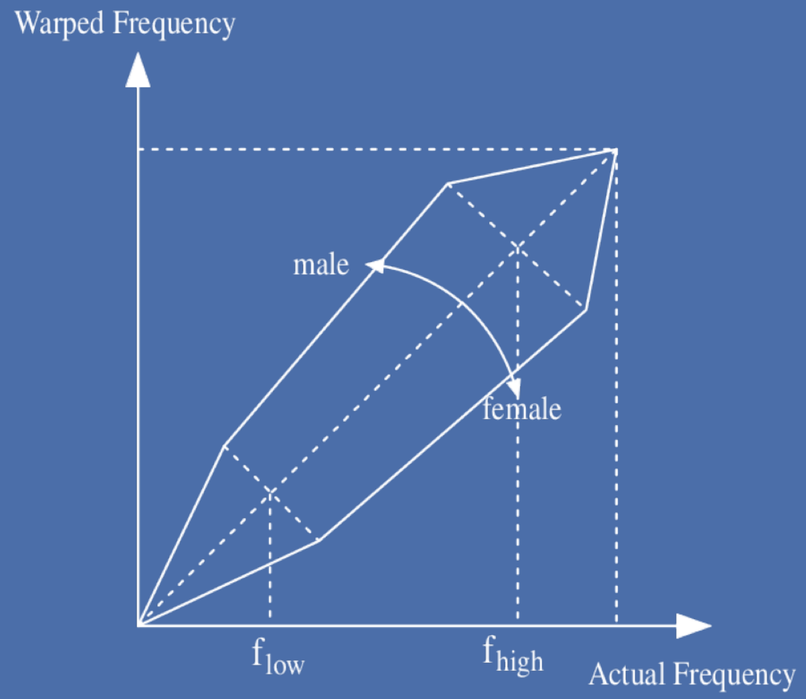
- The main way of dealing with this variability is to have a balanced dataset: Half male, half female
- This way your classifier will not be biased towards males of female voices
- Same solution for the system to work on children’s voices

Vocal Tract Length Normalization





Linear scaling



Piecewise-linear scaling

Question

- You build a dataset to train a speech recognizer for a home assistant. You have hundreds of hours, made of several speakers' speech. When splitting your training and validation/test set, should you:

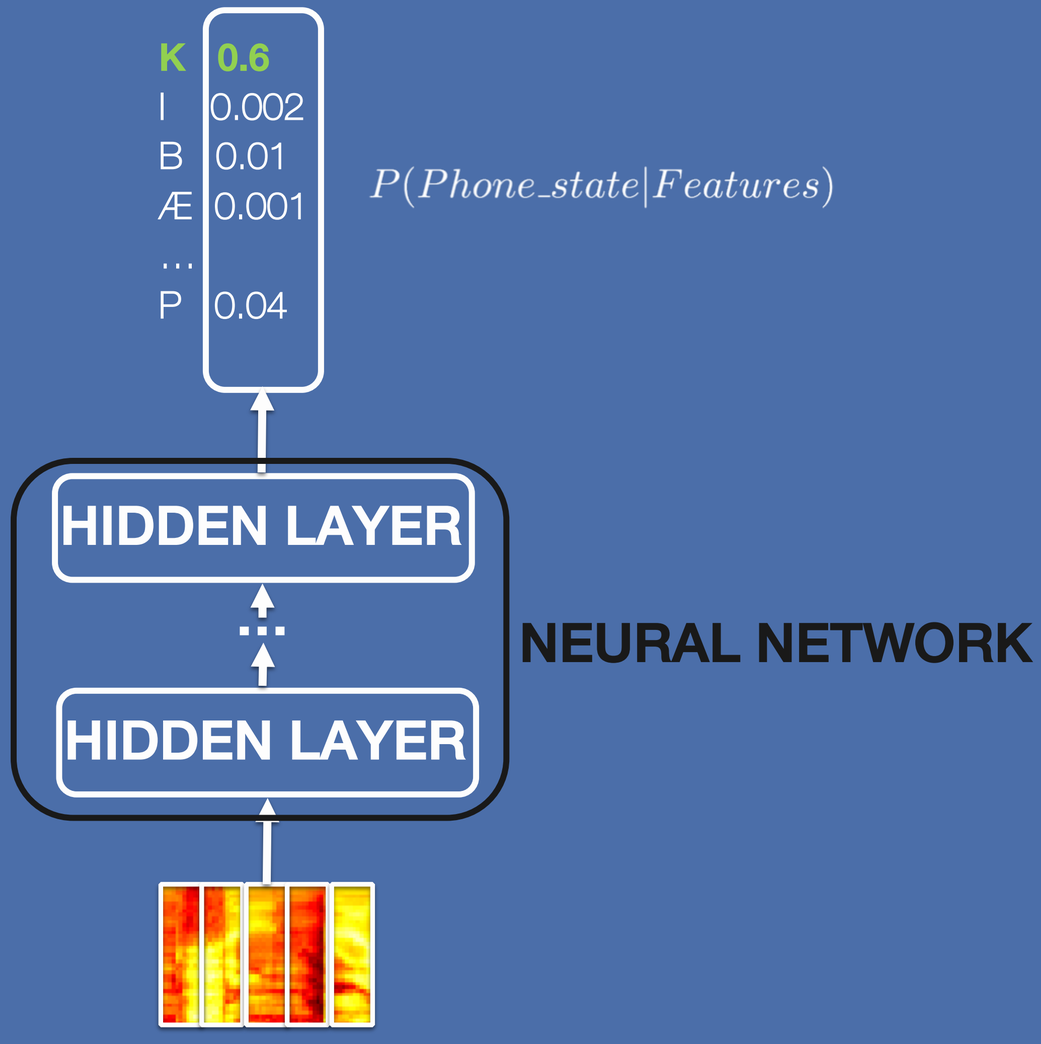
- **1. Split it randomly**
- **2. Split it such that every speaker appears in the training set**
- **3. Split it such that speakers in the training and validation/test are different**

Go to: https://api.socrative.com/rc/KBcrE4 and login
with first name name mail

Speaker variability

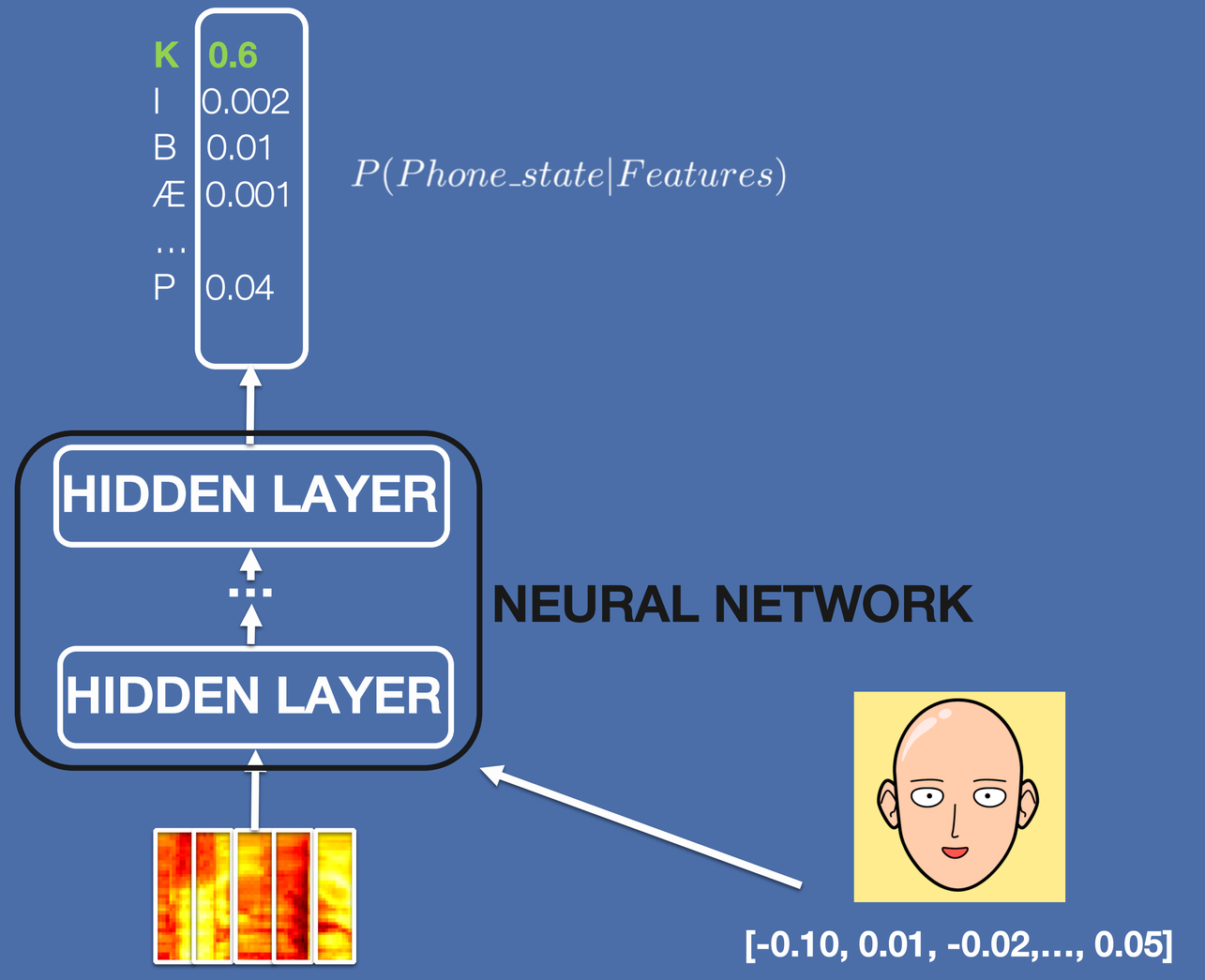
- Variability in speakers is one of the main challenges in speech recognition
- Traditional systems where « speaker-dependent »: they had to fit your voice before working
- Current systems are « speaker-independent »: they are expected to work directly on any voice

Adding speaker features

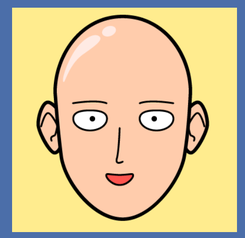


Adding speaker features

i-vectors are features that represent the identity of the speaker, regardless of what they say



Multi-task learning



[0, 0, 1,..., 0]

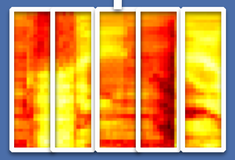
K	0.6
I	0.002
B	0.01
Æ	0.001
…	
P	0.04

$P(phoneme|features)$

HIDDEN LAYER

HIDDEN LAYER

NEURAL NETWORK



Speech recognition « in the wild »

- Speech recognition is « easy » when the speech is clean
- For many applications, the speech recorded by the microphone is noisy
- Example: GPS (road noise, people speaking in the car)
- Example: Bar (people speaking, music, background noise)

Data augmentation

- If you have access to a training set that is clean, but want to test your system on noisy speech
- Add various noises to your dataset: white noise, natural noises, reverberation

Data augmentation

- If you have access to a training set that is clean, but want to test your system on noisy speech
- Add various noises to your dataset: white noise, natural noises, reverberation

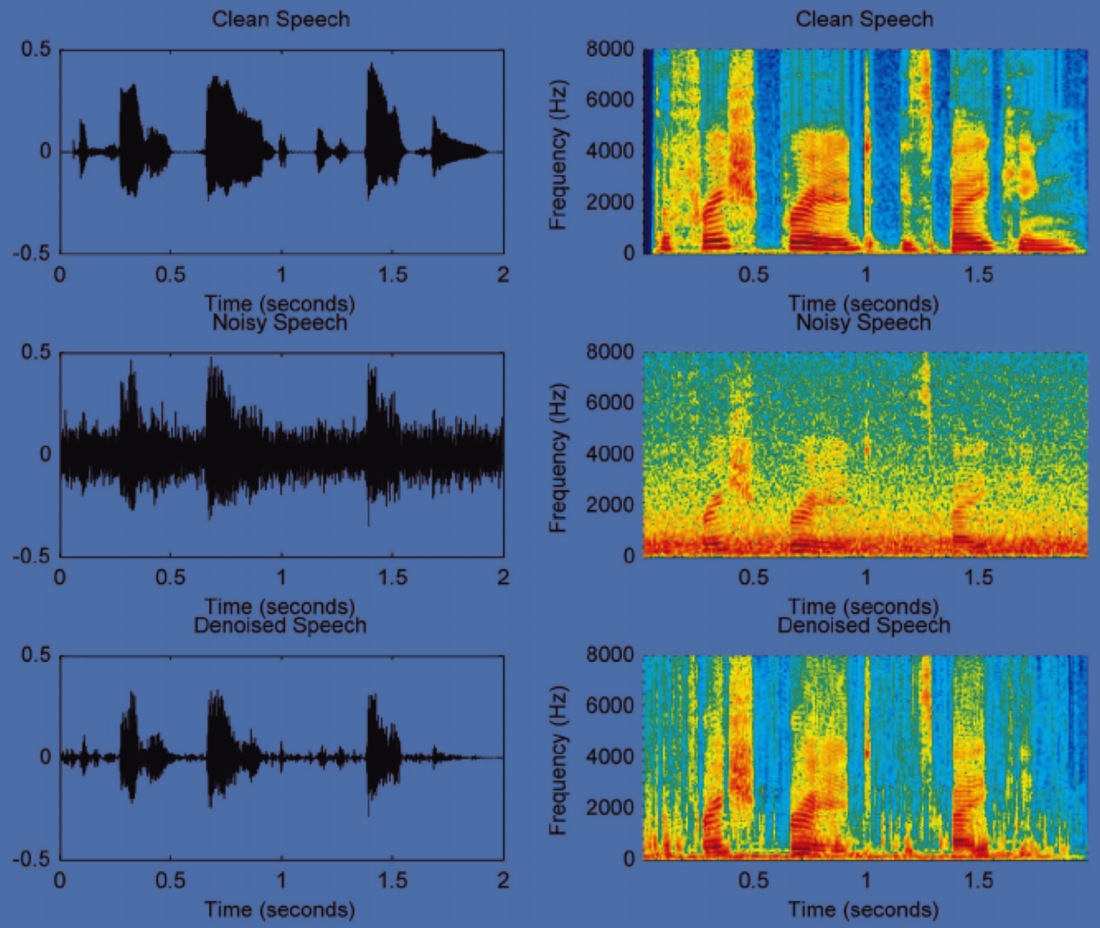
Data augmentation

- If you have access to a training set that is clean, but want to test your system on noisy speech
- Add various noises to your dataset: white noise, natural noises, reverberation

Data augmentation

- If you have access to a training set that is clean, but want to test your system on noisy speech
- Add various noises to your dataset: white noise, natural noises, reverberation

Denoising


Clean Speech
Noisy Speech
Denoised Speech
Time (seconds)
Frequency (Hz)
0.5
0
-0.5
8000
6000
4000
2000
0
1
1.5
2

Conclusion

- Speech and speech features
- Modeling speech with Hidden Markov Models
- Modeling speech features with Gaussian Mixture Models and Neural Networks
- Handling variability in gender, speaker identity, and noise conditions

- **Next class: Language modelling, end-to-end speech recognition, current state-of-the-art and frontiers in speech recognition**

Practical work: Recognizing speech commands