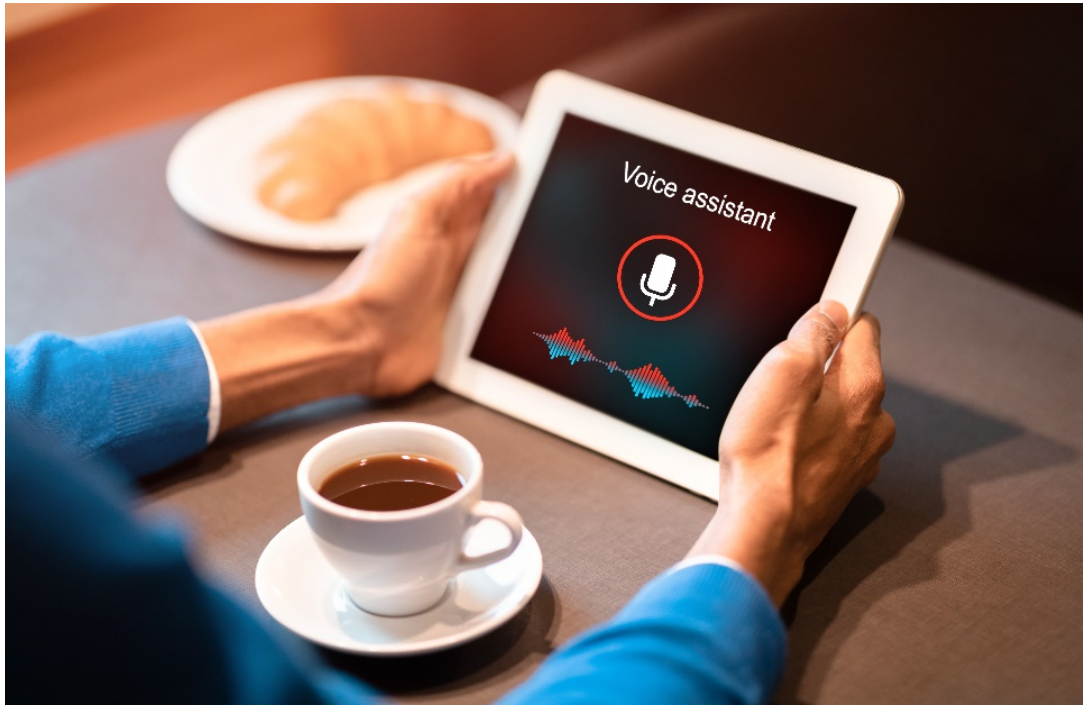
ML school audio track

Introduction

Dr. Paul Wallbott

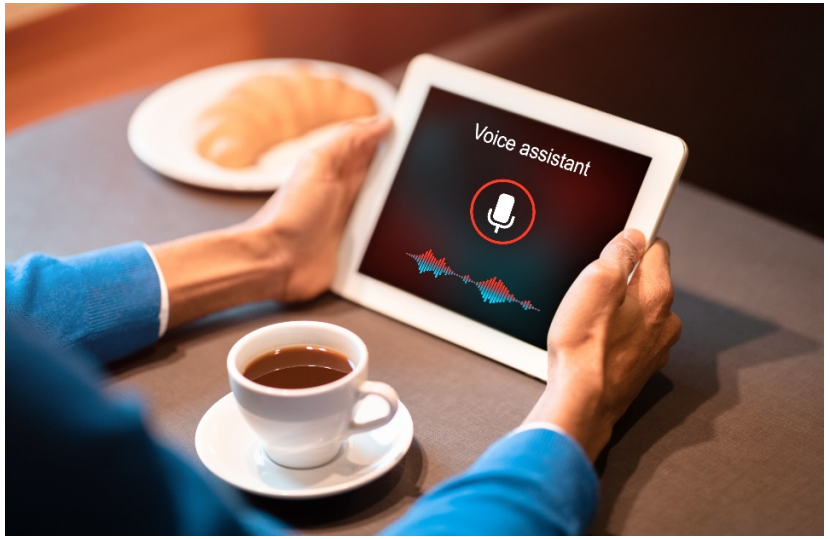


Who are you?

- Please introduce yourself and share your motivation with us!
  - Background
  - Why audio?
  - Expectations

Preview: keyword spotter

- Can detect spoken words!
  - E.g., „Ok google“, “yes”, “no”, “on”, “off”
- Combines
  - Audio processing
  - Deep Learning
- We will build one!



Goals

- Goals
  - Introduction to machine learning with focus on deep learning
  - Introduction to audio processing (for deep learning)
  - **A hands-on experience** combining both

- Remarks
  - You have different backgrounds learn from each other!
  - You might know some stuff already chance to refresh!
  - You might find the pace high interrupt me, ask questions!

- Questions please! I want to talk to you ☺

- Feedback please! Take down things you like/ dislike to help us improve

Remarks

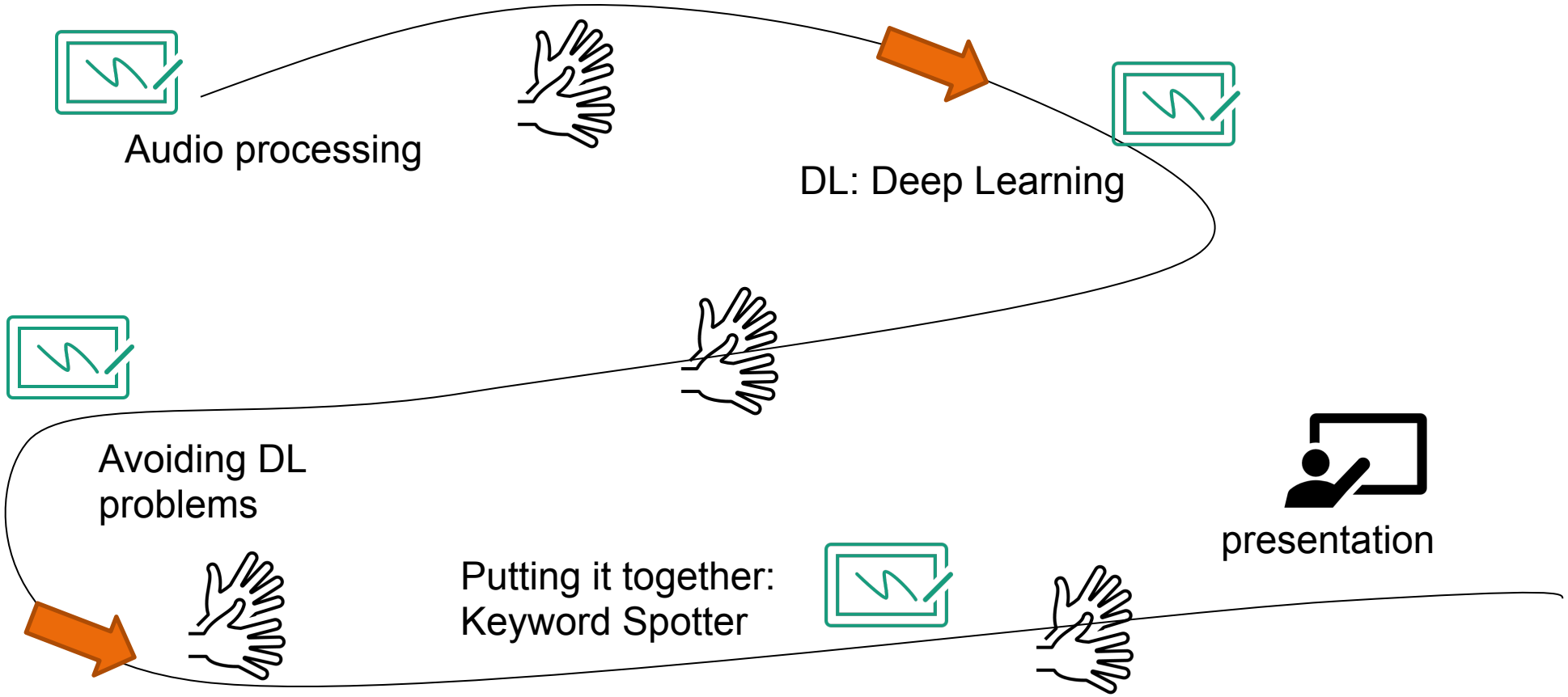
!

I might not have explained every word, ask if I did not do so and you do not know it

Training, epoch, batches, gradients, …

The mission

Road to our neural keyword spotter



Organisation

Table 1

Date // Time	Mo	Tu	We	Do
9.30 – 11.00		Exercise I	Exercise III	
11.15 – 12.45		**Deep Learning I**	**Keyword Spotting**	
13.00 – 14.15				
14.15 – 15.45	**Welcome**	Exercise II	Exercise IV + **Wrap Up**	
16.00 - 17.30	**Intro to Audio**	**Deep Learning II**		

- Rough schedule, might change during the days
- Exercises in small groups
  - You will fill in the missing code in jupyter notebooks

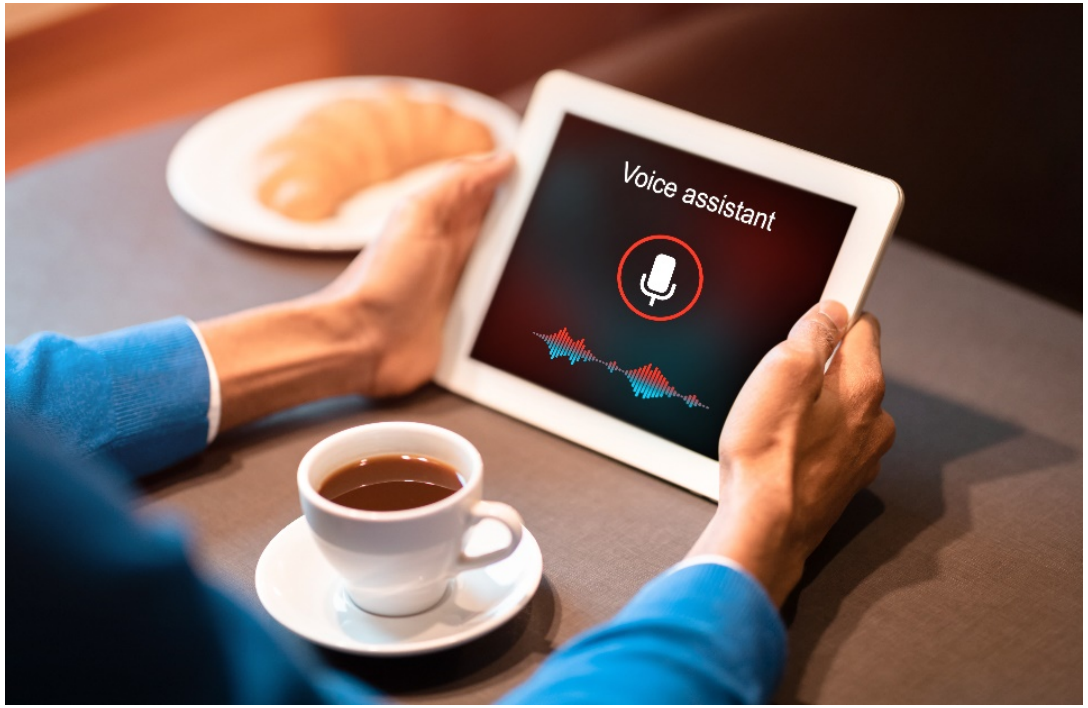
Reading

- Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow (Geron)
  - Easy to understand
  - Very practical approach
- Deep Learning (Goodfellow, Bengio, Courville)
  - Comprehensive in-depth explanations
  - For those who really want to understand what is going on
  - Free! (https://www.deeplearningbook.org/)
- ASR lecture by Peter Bell
- I read about modern topics on towards datascience / towards ai

ML school audio track

Introduction to Audio

Dr. Paul Wallbott, Fraunhofer IAIS



Motivation

- Speech is vital part of our daily life
  - Examples?

Motivation

- Speech is vital part of our daily life
- The list of applications is long
  - Creation of a speech manuscript
  - Dictate software
  - Subtitling of TV shows or videos
  - Searchabilty of media archives
  - Home automation and speech assistents/dialog systems
- Fraunhofer one of many players in the field

Example I: The Speaker Platform

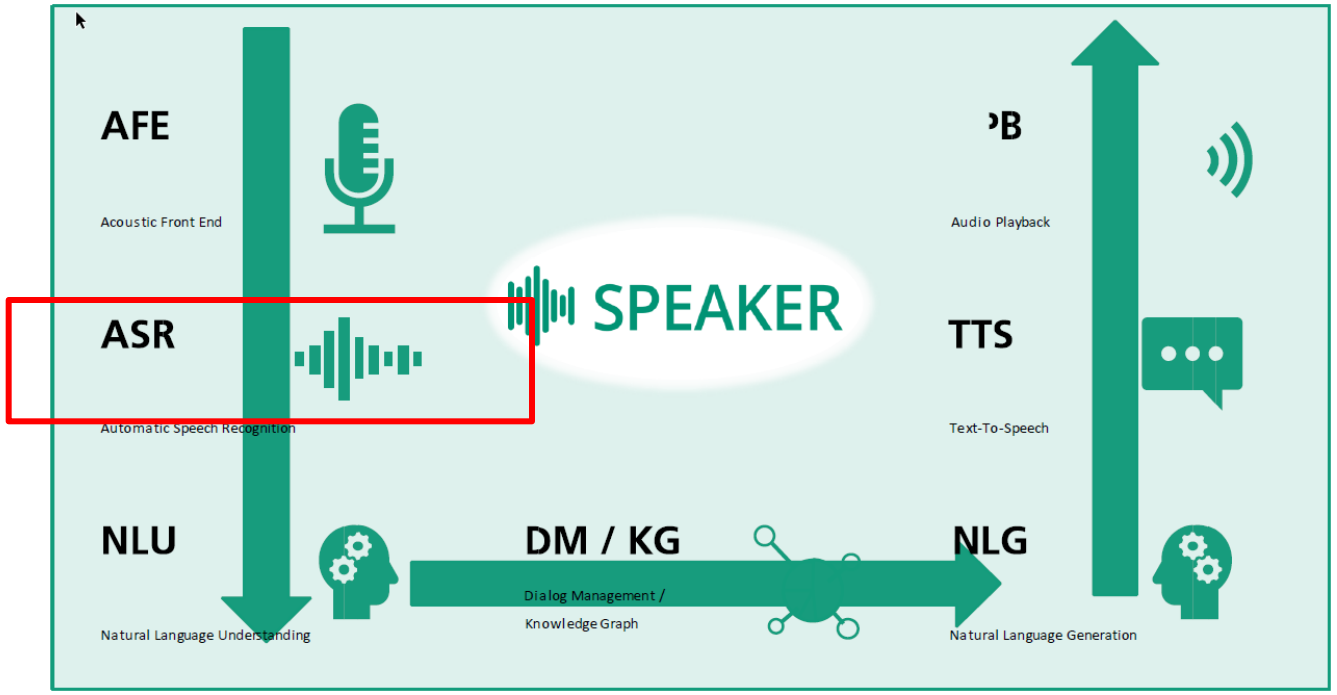
- Different components
  - Acoustic frontend
  - Automatic Speech Recognition
  - Natural Language Understanding
  - Dialogue Management
  - Natural Language Generation
  - Text-To-Speech
  - Playback
- Keyword Spotting / Wake Word detection is part of ASR

Example I: The Speaker Platform

- IAIS provides Automatic Speech Recognition (ASR)

Example I: The Speaker Platform

- IAIS provides Automatic Speech Recognition (ASR)

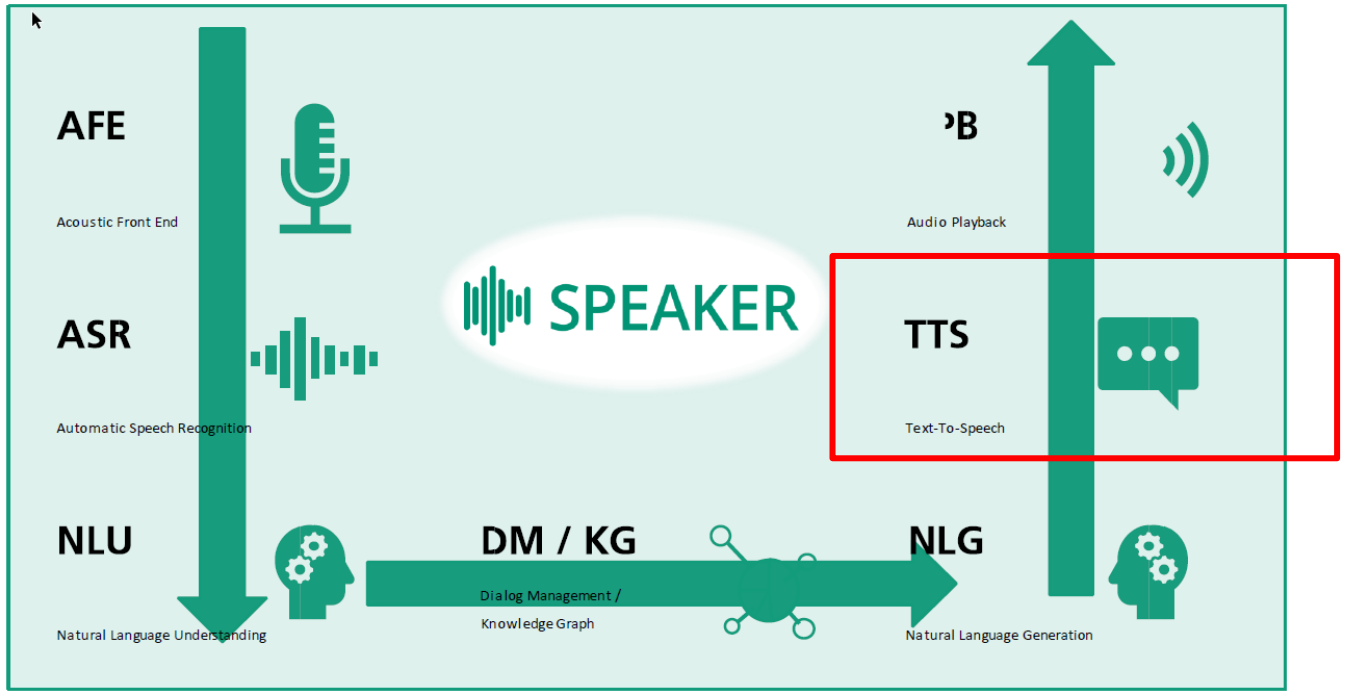


Example I: The Speaker Platform

- LLMs

Example I: The Speaker Platform

- Text2Speech

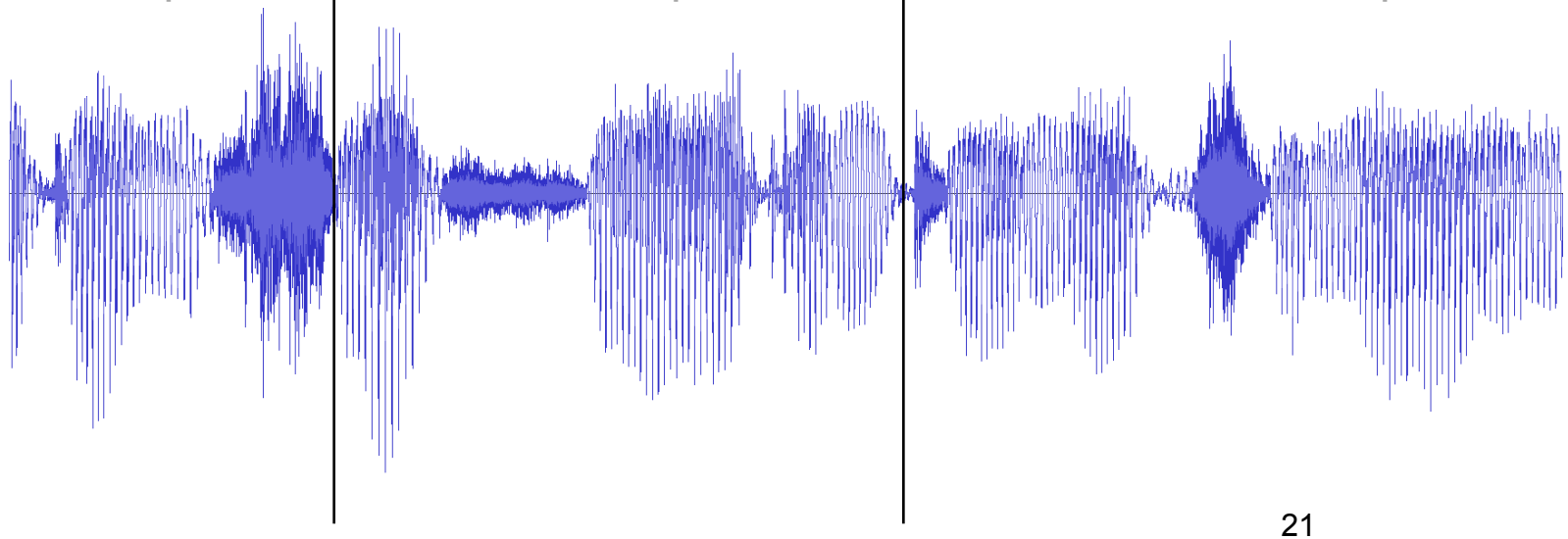


Example II: Audio mining

0:00 – 0:10
Speech
Studio
German
Male
Speaker A
John Doe
Transcript

0:10 – 0:35
Speech
Telephone
German
Female
Speaker B
Jane Doe
Transcript

0:35 – 1:00
Speech
Studio
German
Male
Speaker A
John Doe
Transcript



Need Automatic Speech Recognition (ASR)

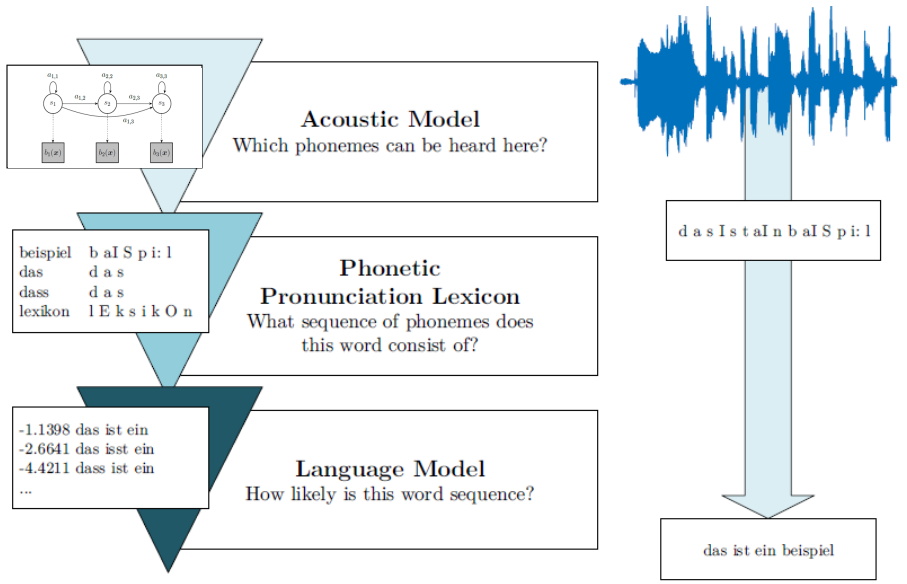
- The problem is formulated as finding the most likely transcript for the given audio
- Classically the problem is separated into
  - Acoustic models that creates a sequence of acoustic units
    - Words (need to know what Im looking for in advance)
    - Sub-Words (can be useful for compound rich languages like German)
    - Phones (good choice usually)
  - A lexicon that translates the acoustic units into words
  - A language model combines the words into sentences

Conventional ASR

Trained on annotated speech

Manually created or trained using G2P

Trained using textual data



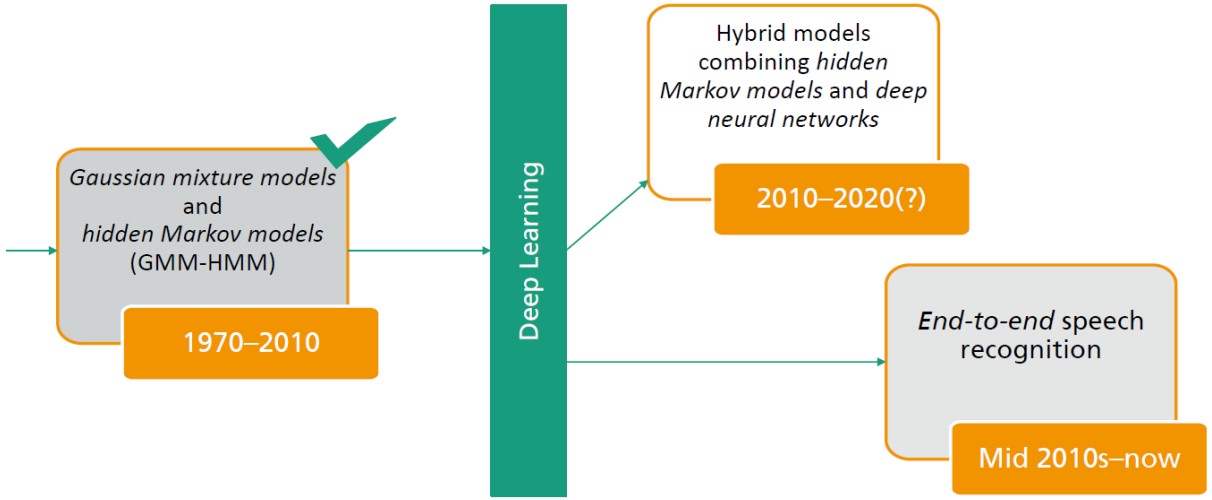
HMM

N-grams

Figure 2.23: Simplified, informal schematic structure of components in a large-vocabulary automatic speech recognition system.

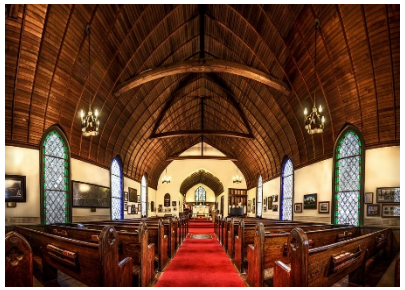
ASR development

- Recent developments include neural networks into the classical pipeline
- Even more recent developments use only neural networks (no HMMs)
  - Often there is still a common way of extracting features spectrograms
  - Examples: Wav2Vec, Whisper



ASR Challenges

Hardware



Room Acoustics

Multiple Speakers

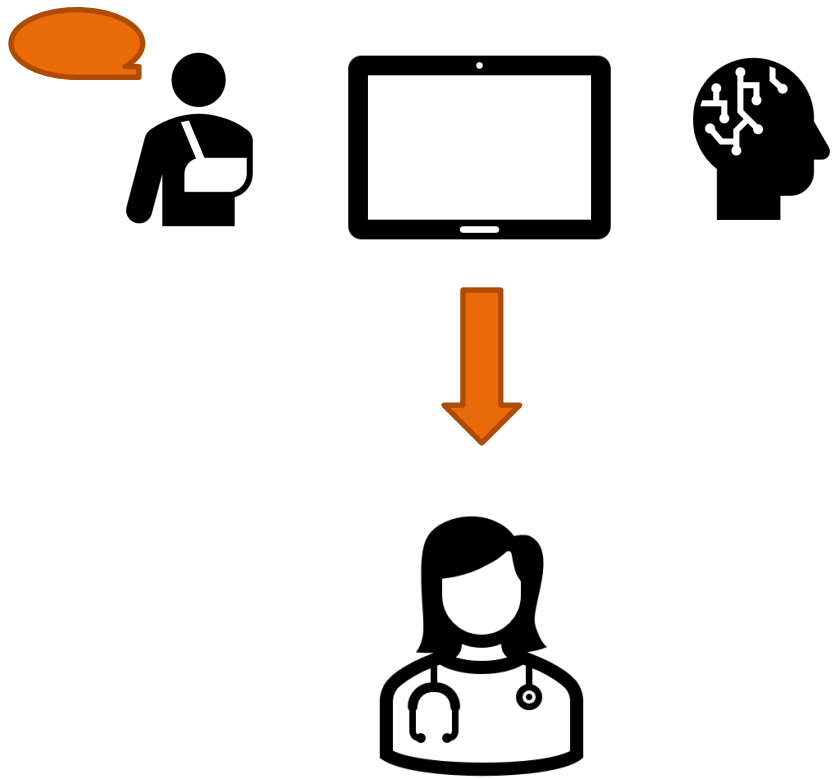
Background Noise

Spontaneous Speech

https://pixabay.com/de/mikrofon-ton-stimme-singen-laut-305503/
https://pixabay.com/de/kirche-kapelle-haus-der-anbetung-581061/
https://pixabay.com/de/r%C3%BCckmeldung-gruppe-kommunikation-2044701
https://pixabay.com/de/manhattan-konzert-solo-klavier-1674404/
https://pixabay.com/de/oliver-kahn-mann-mensch-fussball-406393/
https://pixabay.com/de/freunde-m%C3%A4nnlich-m%C3%A4nner-au%C3%9Ferhalb-1209740/

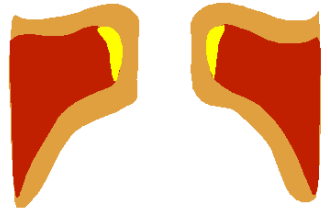
Example III: Telemed

- Voice contains more information than just speech content
- Idea: Medical diagnosis from voice recordings
  - Patient records at home
  - Algorithm recognises problems

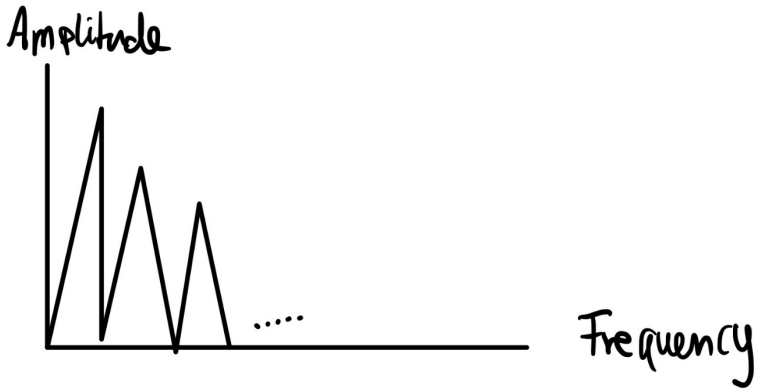


Voice production (voals)

- Audio (speech) is a wave packet
  - Frequencies
  - Intensity
- Generated in our body
  - Air stream from your lunges
  - Oscillation of **vocal folds** create wave with characteristic frequencies

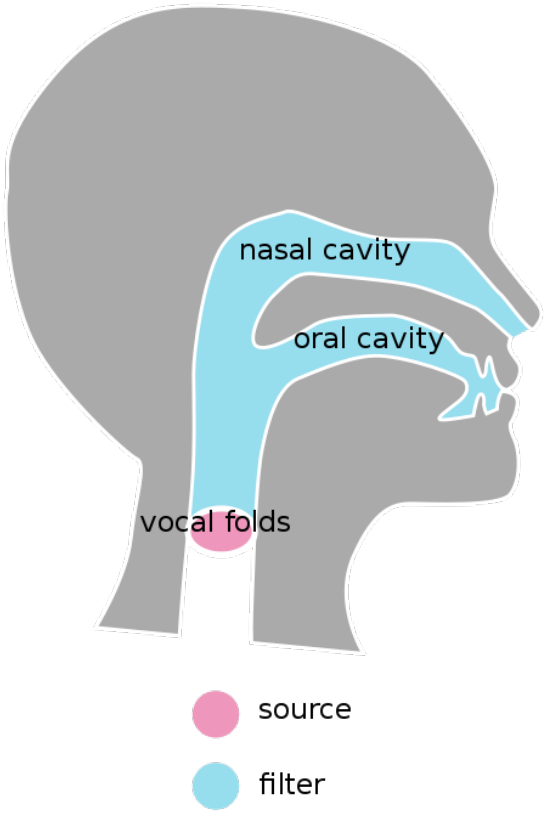


https://commons.wikimedia.org/wiki/File:Vocal_fold_animated.gif



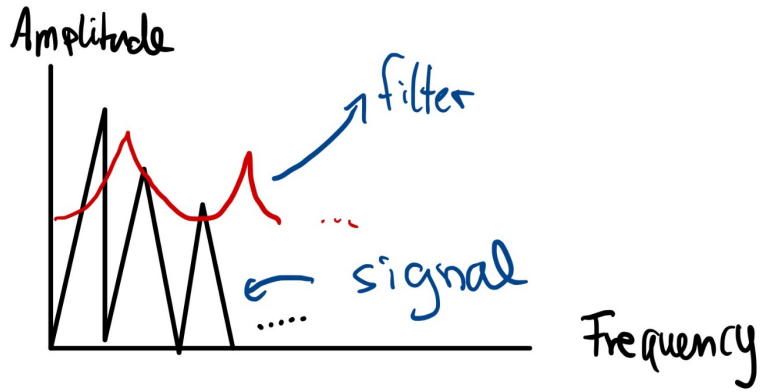
Voice production

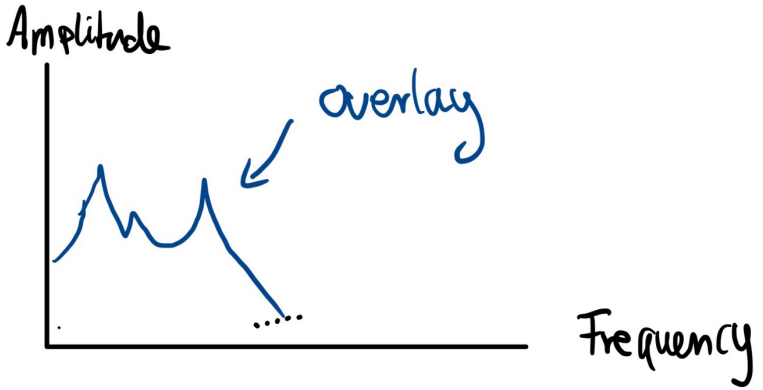
- Audio (speech) is a wave packet
  - Frequencies
  - Intensity
- Generated in our body
  - Air stream from your lunges
  - Oscillation of **vocal folds** create wave with characteristic frequencies
  - **Vocal tract** is a cavity that acts as a filter



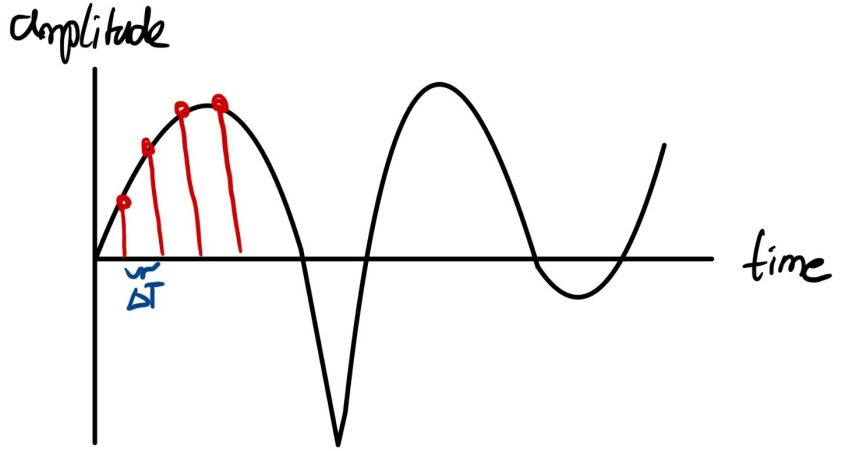
Voice production

- Audio (speech) is a wave packet
  - Frequencies
  - Intensity
- Generated in our body
  - Air stream from your lunges
  - Oscillation of **vocal folds** create wave with characteristic frequencies
  - **Vocal tract** is a cavity that acts as a filter
- The result is a characteristic frequency pattern
- Is influenced by lung stream, patients liquidity state etc.
  - **Might detect changes in voice!**



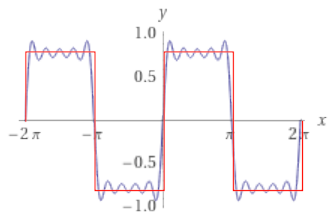


How to understand speech?



- Digital speech representation
  - Signal is sampled in time e.g., with 16 kHz --> series of floats
  - Usually saved as .wav file Series of floats
- Speech recognition starts here
  - Want to understand **which frequencies are present at what time**
- Fourier Transform
  - Decompose the **wav into a sum of waves, each term has single frequency** (sin functions in time)
  - Visual introduction --> watch this
  - **Amplitude of each single frequency wave related to how much energy** is stored in that frequency
- **Goal: Extract a set of numbers (features) that describe the content of an audio signal in terms of frequency and energies stored in those frequencies**

Discrete Fourier Transform



$$\sin(x) + \frac{1}{3}\sin(3x) + \frac{1}{5}\sin(5x) + \frac{1}{7}\sin(7x) + \frac{1}{9}\sin(9x)$$

Values that characterise each component:

- The frequency
- The amplitude
- Phase

Discrete Fourier Transform

These coefficients represents the amplitude and phase as complex number

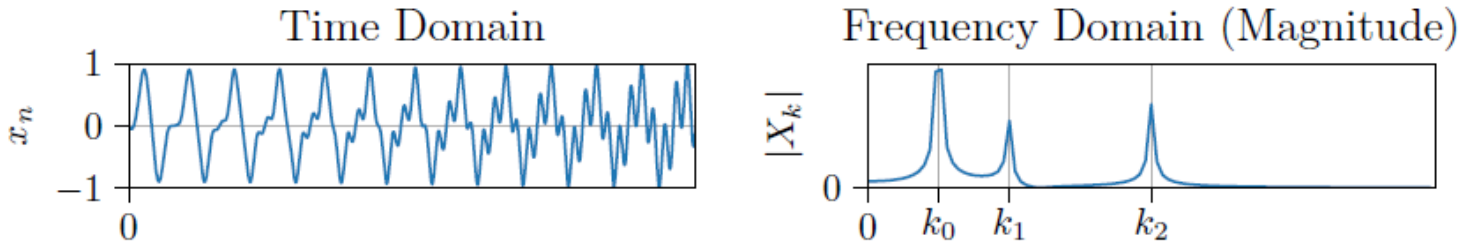
$$\mathcal{DFT}(x) := \left(\sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi n \frac{k}{N}}\right)_{k\in\mathbb{Z}}$$

discrete and periodic spectrum

Periodic signal: We can plot the coefficients of the entire signal and understand it!

Time dependent (almost periodic) signals

- We can see that the signal consists of three main frequencies
- However, cannot see time dependence in the frequency domain
- Applying the DFT on the entire signal is not very helpful

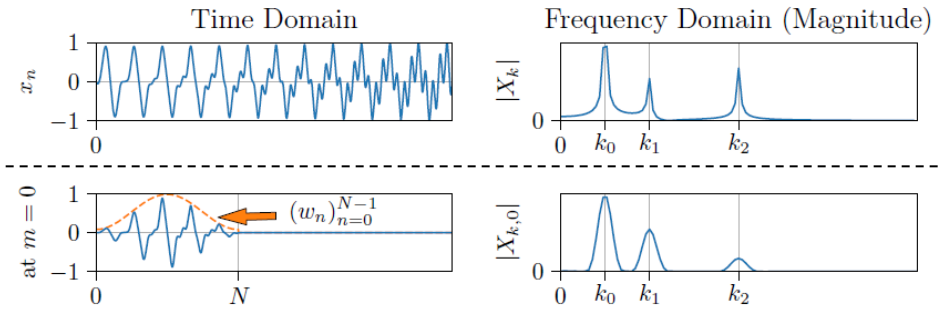


Discrete Fourier Transform

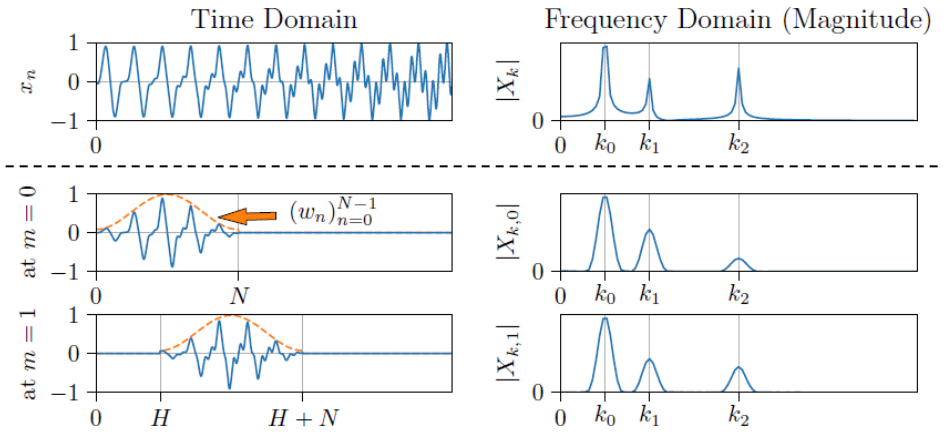
$$\mathcal{DFT}(\boldsymbol{x}) := \left(\sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi n \frac{k}{N}} \right)_{k \in \mathbb{Z}}$$

discrete and periodic spectrum

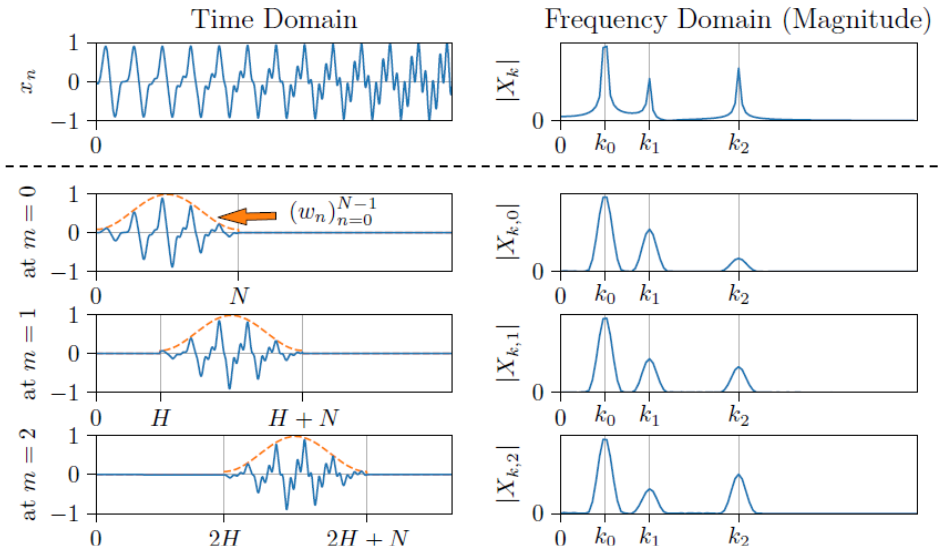
Short-Time Fourier Transform



Short-Time Fourier Transform

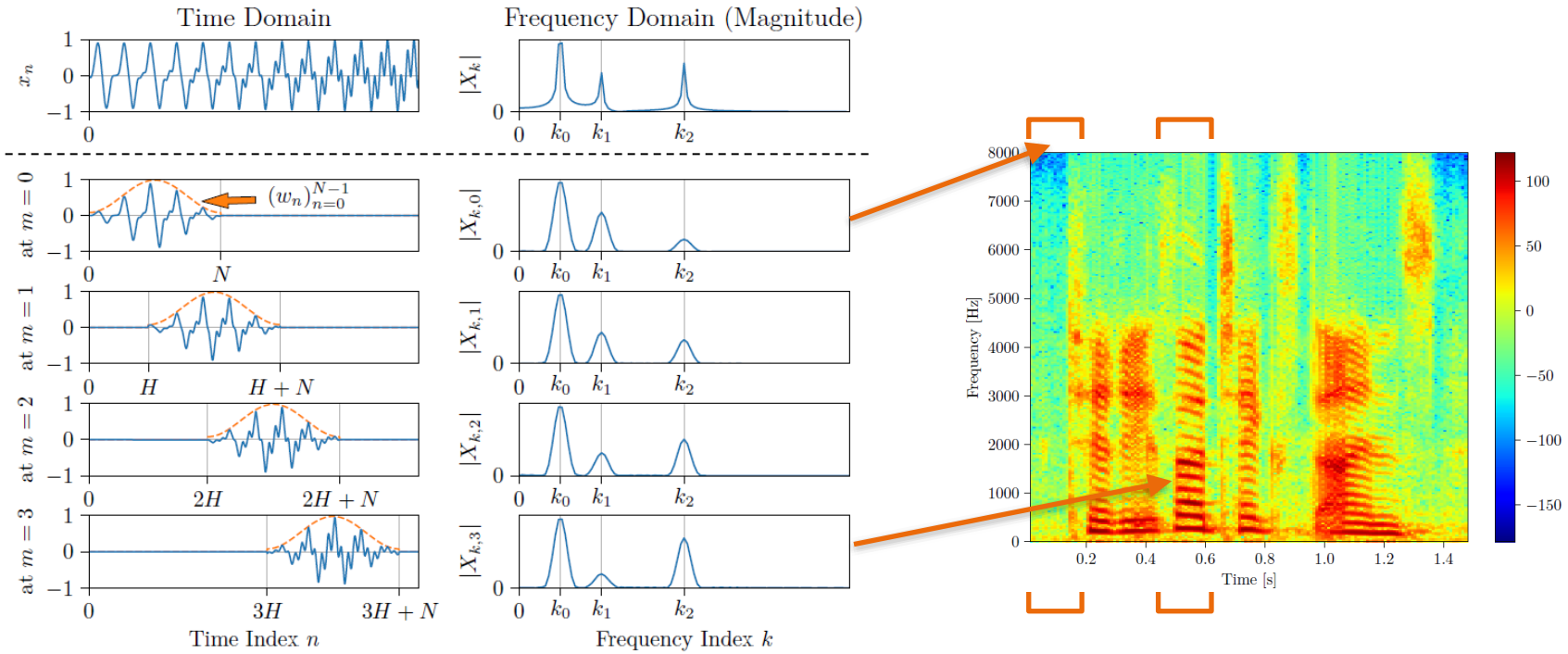


Short-Time Fourier Transform



Short-Time Fourier Transform (STFT)

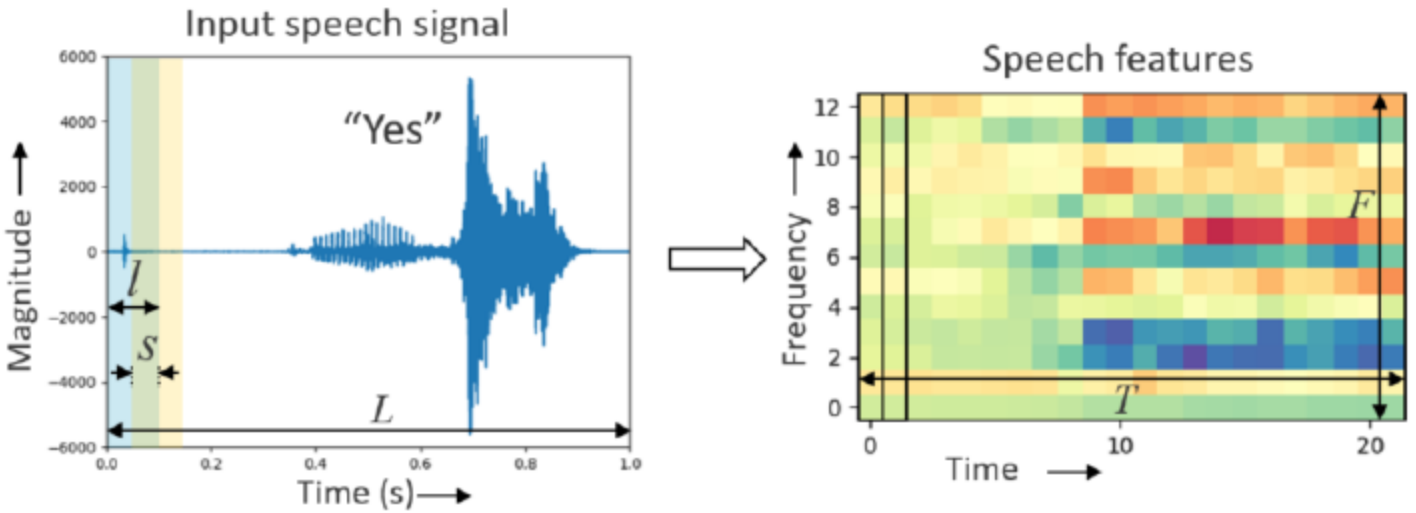
- left: signal. Middle: Fourier coefficient over frequency bin
- Each FT is one column in the spectrogram (color = log of amplitude prop. to energie)
- This is plotted as a continuous function, but it is actually a histogram with discrete bins in k
- The signal on the left does not necessarily correspond to the spectrogram on the right. It is just an illustration



STFT sliding window

- Notation
  - Stride S is overlap of two consecutive windows
  - Window length L is length of the window
  - Hopsize H is L-S

Remark: sometimes the number of points used for the Fourier trafo is not equal to the number of samples per window!



Zhang et al 2018

Exercise 1:

- What is the expected number of windows / features for a 1 second long audio recording with f = 16 kHz when we usa a sliding window of L = 40 ms and S = 20 ms?
- Hints
  - Start with calculating the hopsize and window length in samples
  - Draw a scatch of the signal with hopsize and window length similar to the one on the last slide
  - Think about how many windows fit into the signal

Exercise 1:

- You can discuss this in more detail later today in the smaller groups
- What is the expected dimension of a T = 1 second long audio recording with f = 16 kHz thats sampled with a sliding window of l = 40 ms and s = 20 ms?

The Hopsize H in ms is given by:

$$H = L - S = 20\,ms$$

The hopsize h in samples, is:

$$h = H\ f_s = 320$$

the number of frames is:

$$N_f = [\frac{N_s - s}{h}] = 49$$

where N_s is the number of samples in the 1s audio signal.

```
(16000 - 320)/320
```

```
49.0
```

Mel-Frequency Cepstral Coefficients
Mel Filter Bank

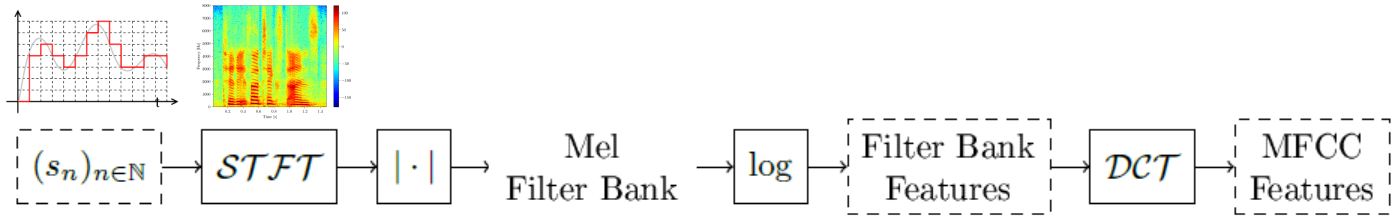


Figure 2.24: Workflow to obtain MFCC features.

Mel-Frequency Cepstral Coefficients
Mel Filter Bank

- Human hearing has non-linear frequency resolution
- 4 kHz not perceived as “double as high” as 2 kHz
- Stevens et al. 1937: “Mel-Scale”
- lower frequency more important for speech than high frequency
- Arrange it so, that higher frequencies move closer together

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

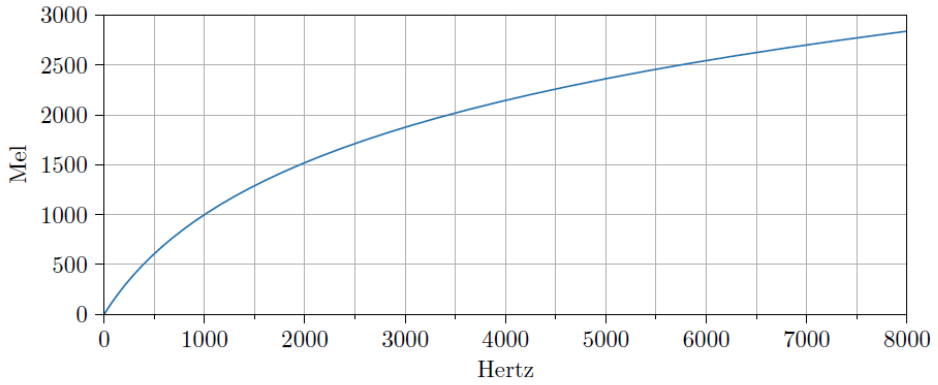
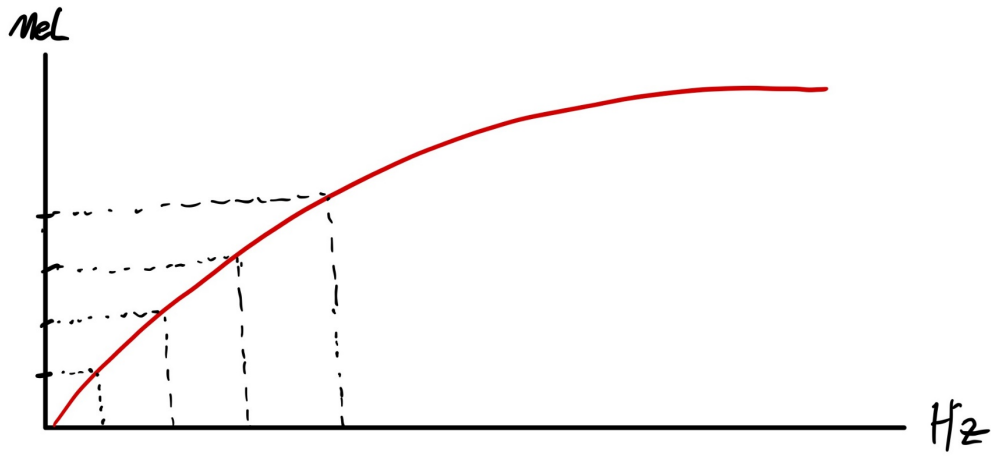


Figure 2.25: Commonly used MEL scale (shown from 0 to 8000 Hz) that maps frequencies in Hertz to respective MEL frequencies.

Mel-Frequency Cepstral Coefficients
Mel Filter Bank

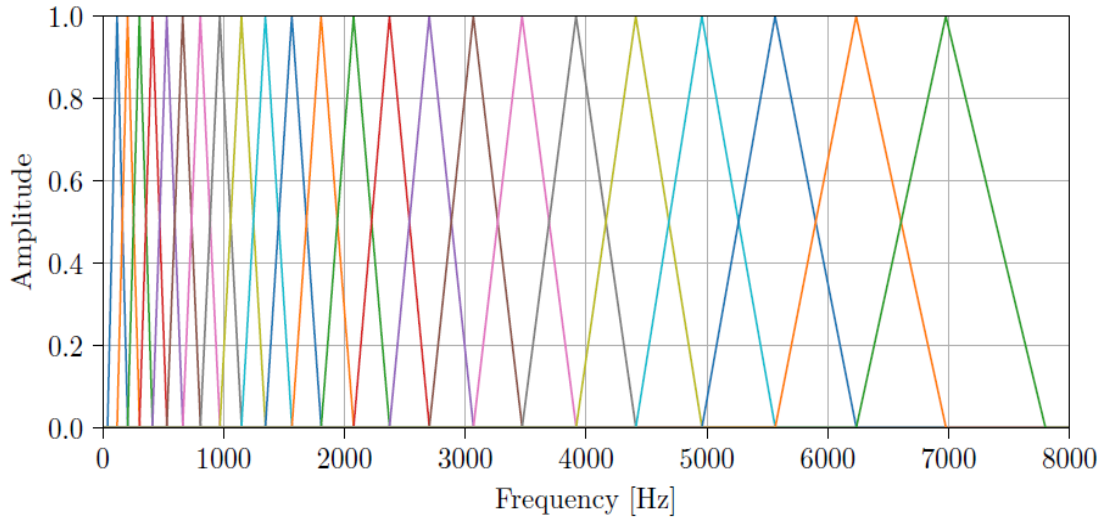
- Human hearing has non-linear frequency resolution
- 4 kHz not perceived as “double as high” as 2 kHz
- Stevens et al. 1937: “Mel-Scale”
- lower frequency more important for speech than high frequency

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



Mel-Frequency Cepstral Coefficients
Mel Filter Bank

- The Mel scale is used to design a filter-bank that summarizes frequency intervals
- These intervals are considered equally distant in human perception based on the Mel scale
- In practice, for example 23 intervals are chosen



Mel-Frequency Cepstral Coefficients
Short-Time Fourier Transform

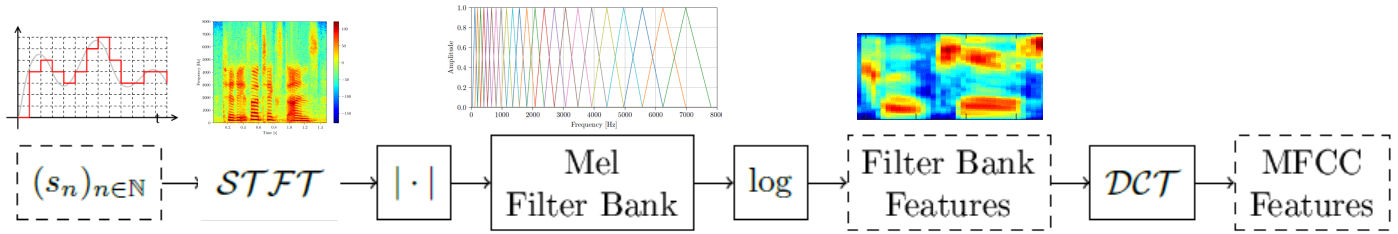


Figure 2.24: Workflow to obtain MFCC features.

Mel-Frequency Cepstral Coefficients
Discrete Cosine Transform

- The features we want to get to should not be correlated
  - => Covariance matrix of features is diagonal
- Decorrelation of features with discrete cosine transformation

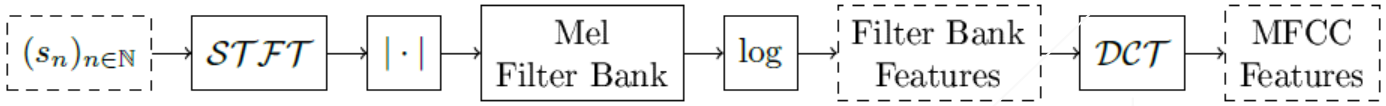


Figure 2.24: Workflow to obtain MFCC features.

$$\boldsymbol{c} := \mathcal{DCT}(\boldsymbol{z}) := \left(\sum_{n=0}^{N_F-1} z_n \cdot \cos\left(\frac{\pi \cdot k}{N_F} \left(n - \frac{1}{2} \right) \right) \right)_{k=0}^{N_C-1}$$

Mel-Frequency Cepstral Coefficients
Short-Time Fourier Transform

- We can now build an audio pipeline
  - Read .wav file
  - Calculate the spectrogram and mfcc features

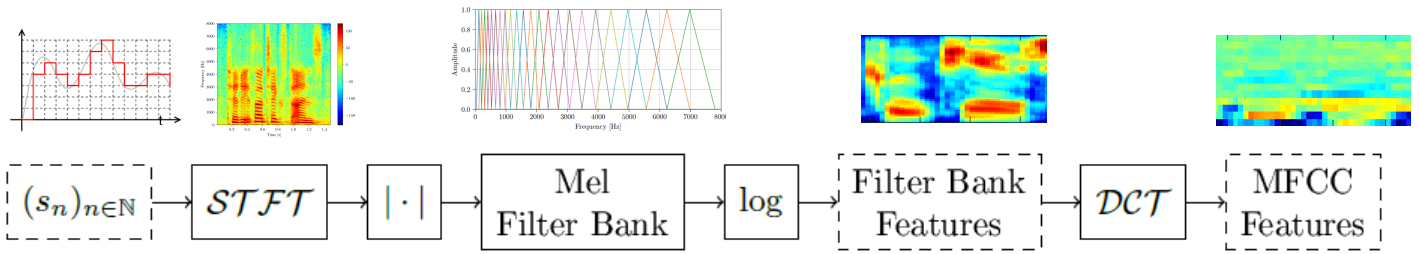


Figure 2.24: Workflow to obtain MFCC features.

Tools

- A great way to display audio is the **Ipython.display** class
  - „Ipd.Audio“ can play audio in jupyter notebboks
- A great python library for signal processing is **librosa**
  - Load wav files with resampling with „load“
  - Short time FFT
  - Mel spectrograms
  - Mfcc coefficients
- To visualize spectrograms we can use **matplotlib.imshow**

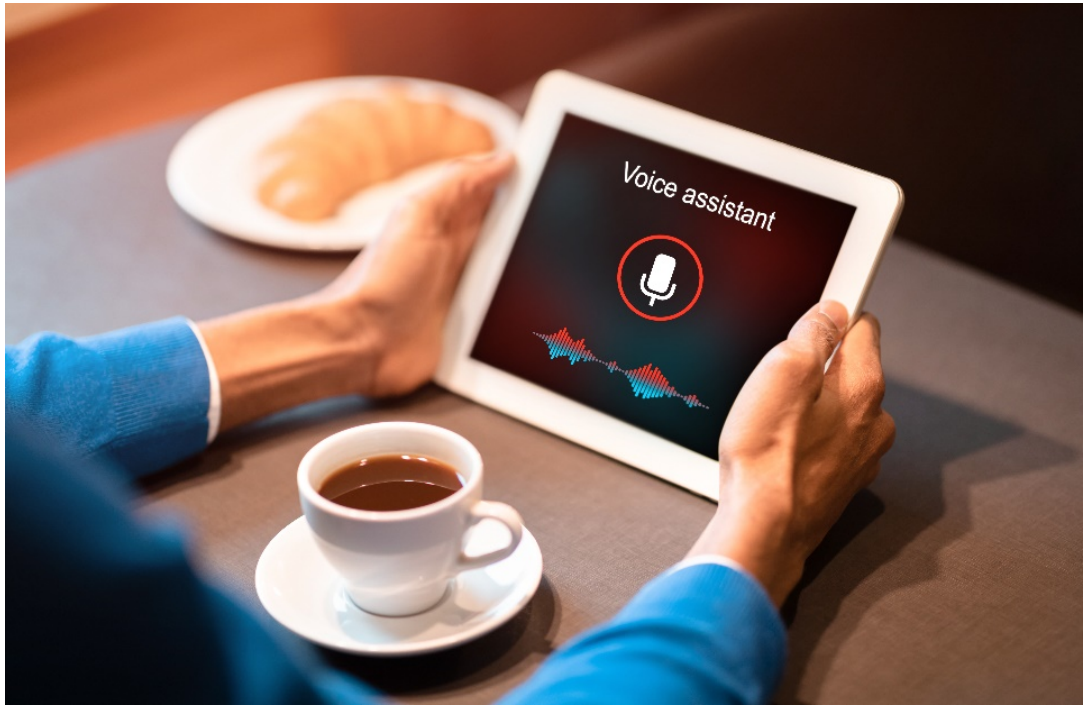
Licensing

- Please note that the slides are only for your personal use and not meant to share (at least for now) ☺

ML school audio track

Introduction to Audio II: Preparing our dataset

Dr. Paul Wallbott, Fraunhofer IAIS



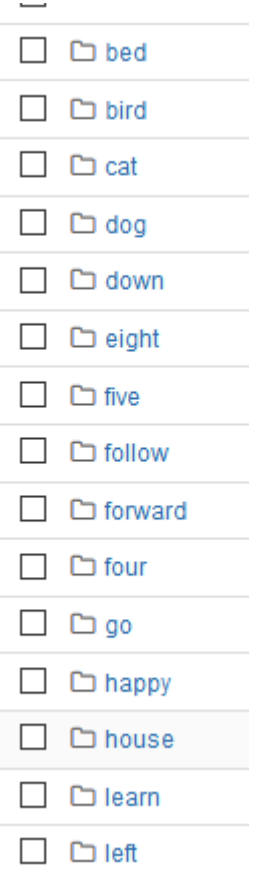
iStock.com | monsitj

Practical tips: Data Sources

- Deep Learning community often organises challenges
  - Datasets and metrics are provided: best model wins
- Kaggle can be a good data source for some problems
- Some benchmarking sets for (multilingual) ASR I know about
  - Libri Speech
  - Common voice
  - Babel
  - Ted Lium
- Keyword spotting
  - **Google Speech Commands: we will use this today**

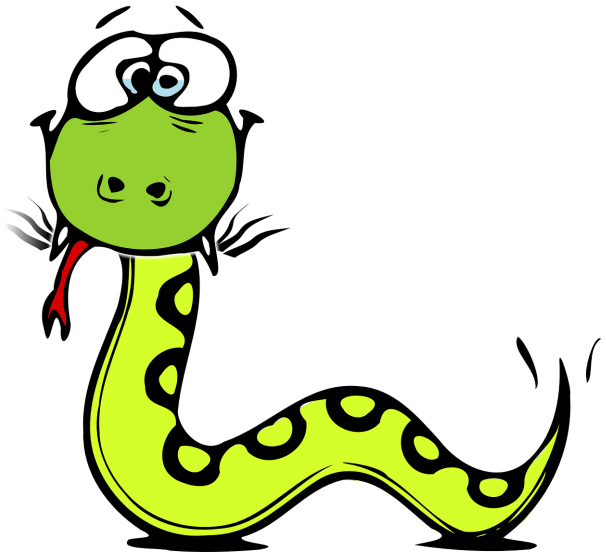
Practical tips: The GSC dataset

- „Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition“ Warden 2018
- Collection of .wav files for different words
  - Few thousand recordings per word
  - Many speakers
  - Different recording conditions
- Set already split into train, test, validation subsets (more later)



Python library overview

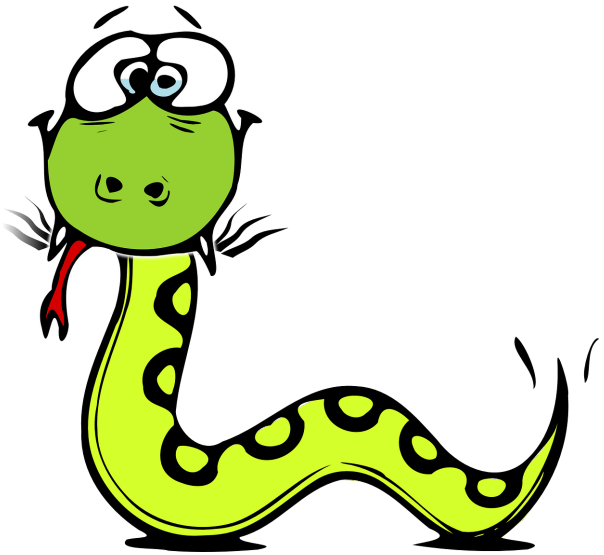
- Python is the most widely used language for data science
  - Easy to learn
  - Convenient package manager
  - Large community —> many packages available
  - Many great libraries



https://pixabay.com/de/vectors/schlange-python-gr%c3%bcn-reptil-312561/

Python library overview

- Machine Learning
  - Scikit-Learn
    - Great for data pipelines, data splitting, ML model training
  - Pandas
    - Great for reading and processing data
- Deep Learning
  - Pytorch (facebook)
  - Tensorflow (google)
    - Keras is a python layer that operates on tensorflow
    - Keras is easy to use, powerful, free —> we will use keras



https://pixabay.com/de/vectors/schlange-python-gr%c3%bcn-reptil-312561/

Pandas

- The central element of pandas is the dataframe class
- There are specific methods that operate on it
  - Get infos: info, describe
  - I/O: to_csv, to_pickle
  - …
- Columns are “series” objects and can easily be manipulated

```
## create a dataframe
df = pd.DataFrame.from_dict({'name':['Hans','Peter','Ulrike'], 'age':[55,63,59]})
df
```

	name	age
0	Hans	55
1	Peter	63
2	Ulrike	59

	age
count	3.0
mean	59.0
std	4.0
min	55.0
25%	57.0
50%	59.0
75%	61.0
max	63.0

Exercise

- Who is good in Python? Audio? ML? Backgrounds?
- Come together in small groups! (at least 2 per team)
- Make sure you can log in with your account and find the jupyter notebooks (next slide)
  - Folders with keywords and audio samples inside
  - Make sure you have a data directory that contains the google speech command dataset
- Copy all the notebooks so you can go back to the original if necessary. You can delete the solutions in your working copy if you want (not so tempting)
- Work through the first notebooks: exercise_01.ipynb, exercises 2a and 2b.
  - Go through the notebooks cell by cell
  - Stop when you encounter an exercise section (there are hints provided as well)
  - Work on the exercises within your team
  - Look at the solutions only if really necessary
- !!! Execute only one notebook at a time, they are independent of each other !!!

Setup

- Each participant has a unique username and password
- Login on
  - http://vss-text.rrz.uni-koeln.de:8080/USERNAME or
  - http://vss-speech.rrz.uni-koeln.de:8080/USERNAME (replace USERNAME with your username, e.g., dgx7001)
- Enter your password as a token
- Done.