Speech Processing with Neural Networks

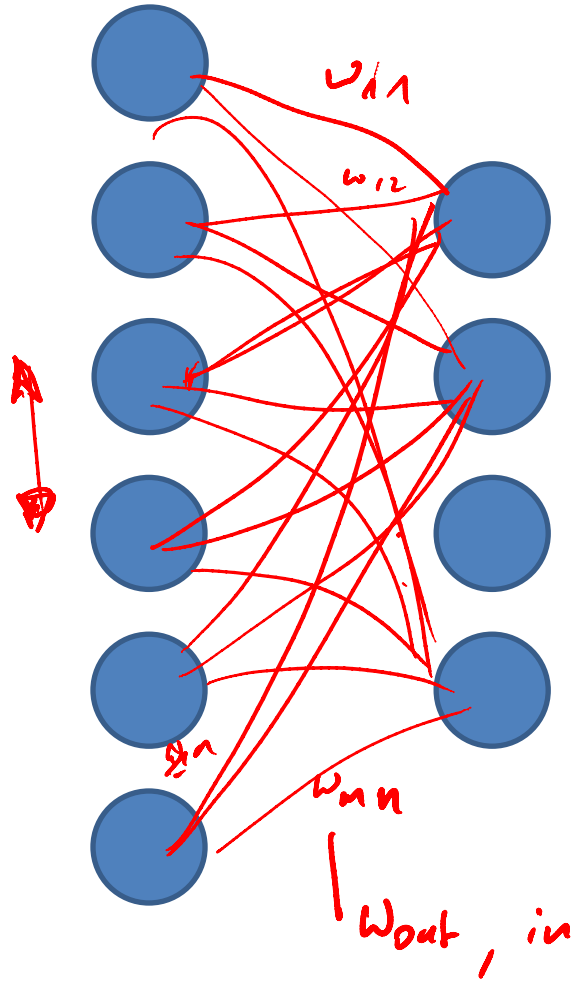
Jan Chorowski

RECAP: DIFFERENT NEURON TYPES

Different Layers for Different Problems

Dense layers:

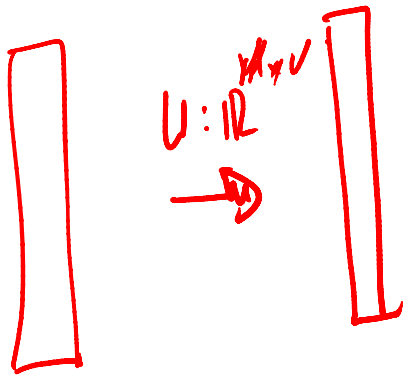
- Global connectivity:
  All outputs use all inputs
- No weight sharing
  $O(N^2)$ weights
- Trains the same when inputs/outputs permuted
- Any combination of input/output dimension possible



Dense Layers

Dense!

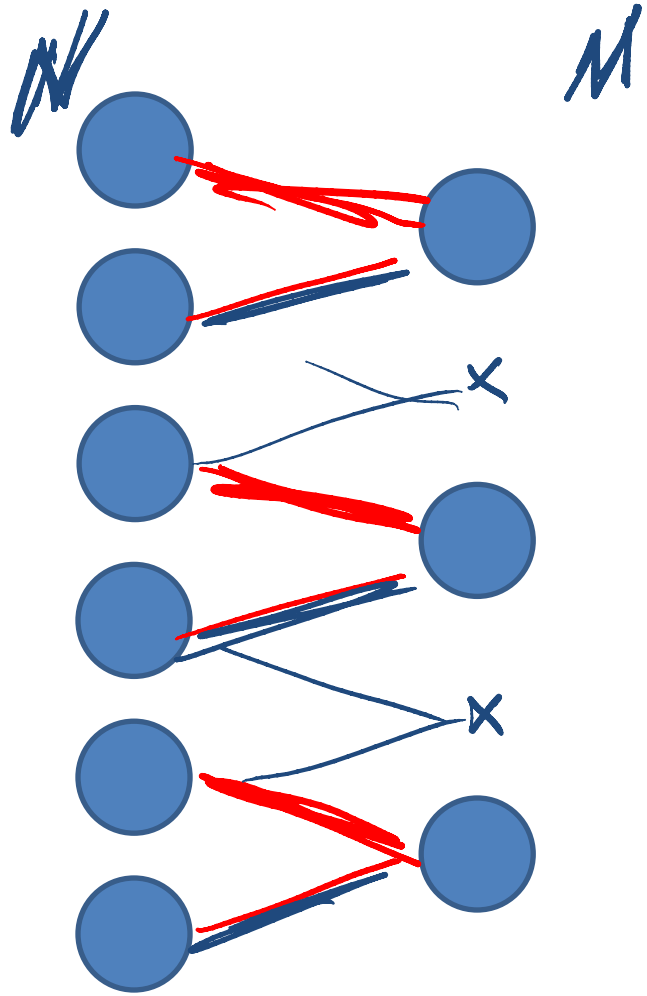
$\mathbb{R}^N \to \mathbb{R}^M$



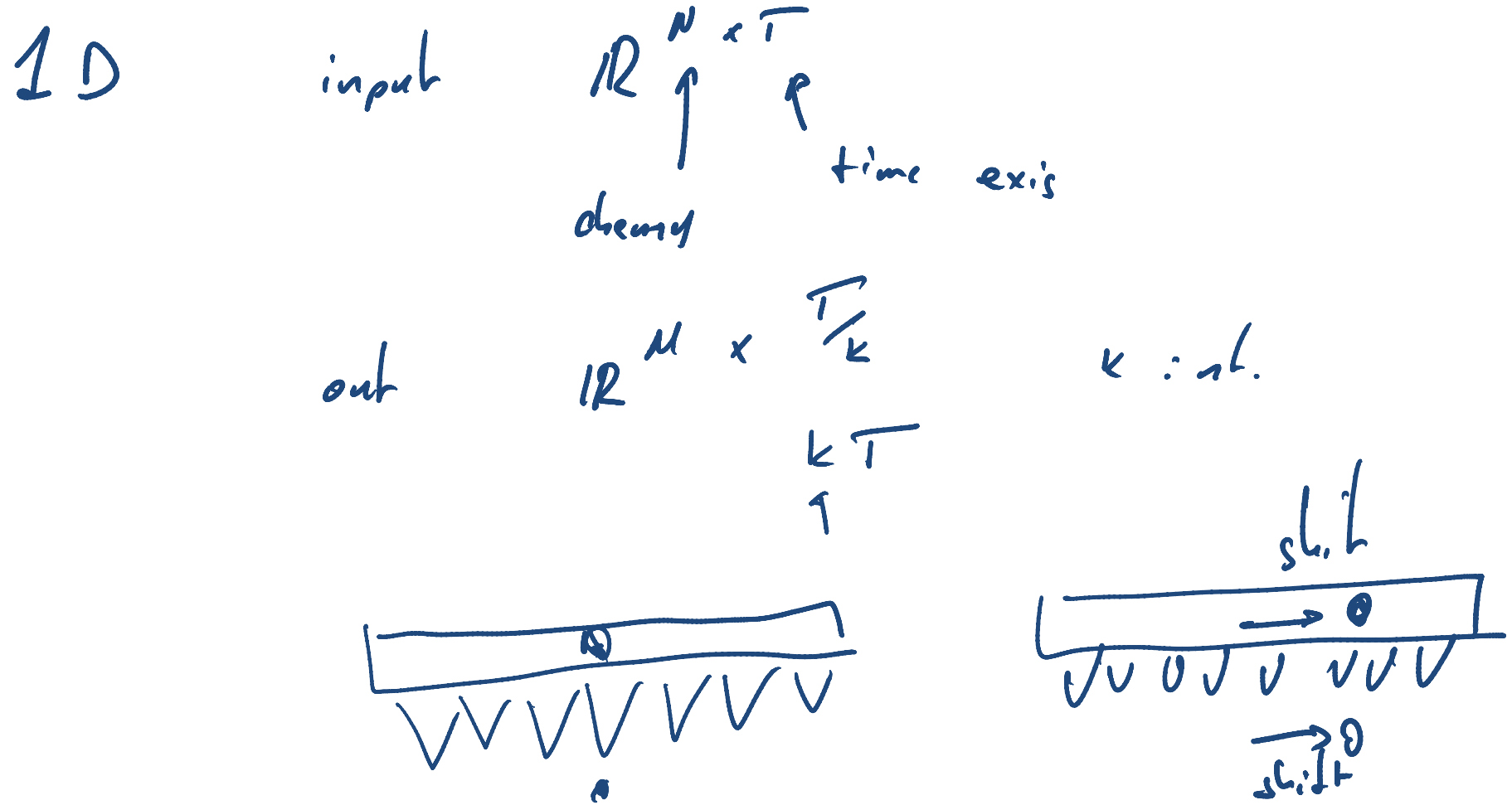
Different Layers for Different Problems

Convolutional layers

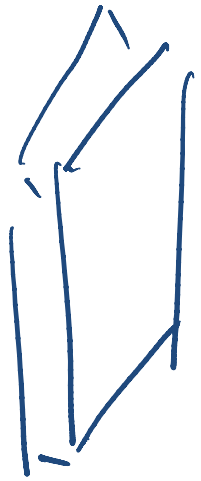
- Local connectivity – each output depends on small set of inputs
- Weight sharing – number of weights independent of input size
- Depends on sensible input/output ordering (e.g. can't shuffle pixels in an image)
- Number of output proportional to number of inputs (give-or-take padding)



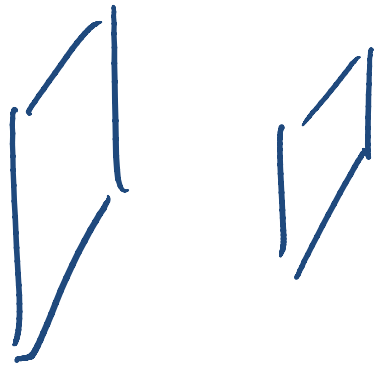
Conv layers


1 D
input
$\mathbb{R}^{N \times T}$
time axis
channels
out
$\mathbb{R}^{M \times \frac{T}{k}}$
k : wl.
kT
T
sh. l
sh. l

Conv nets



$\mathbb{R}^{N \times W \times H} \rightarrow \mathbb{R}^{M \times 1 \times 1} \rightarrow$ image recognition

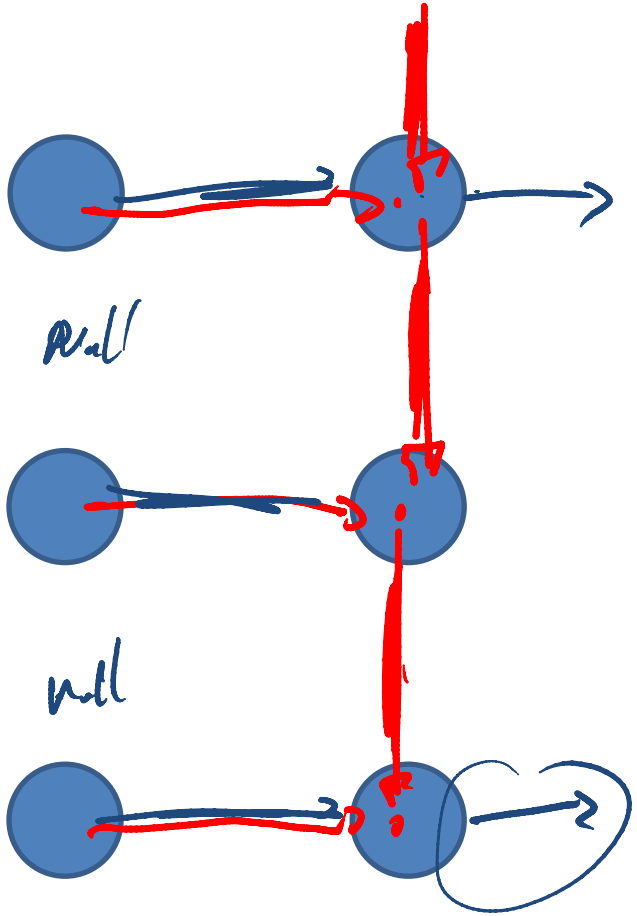


$\mathbb{R}^{N \times W \times H} \rightarrow \mathbb{R}^{M \times \frac{W}{k} \times \frac{H}{k}} \rightarrow$ detection, segmentation

Different Layers for Different Problems

Recurrent Layers

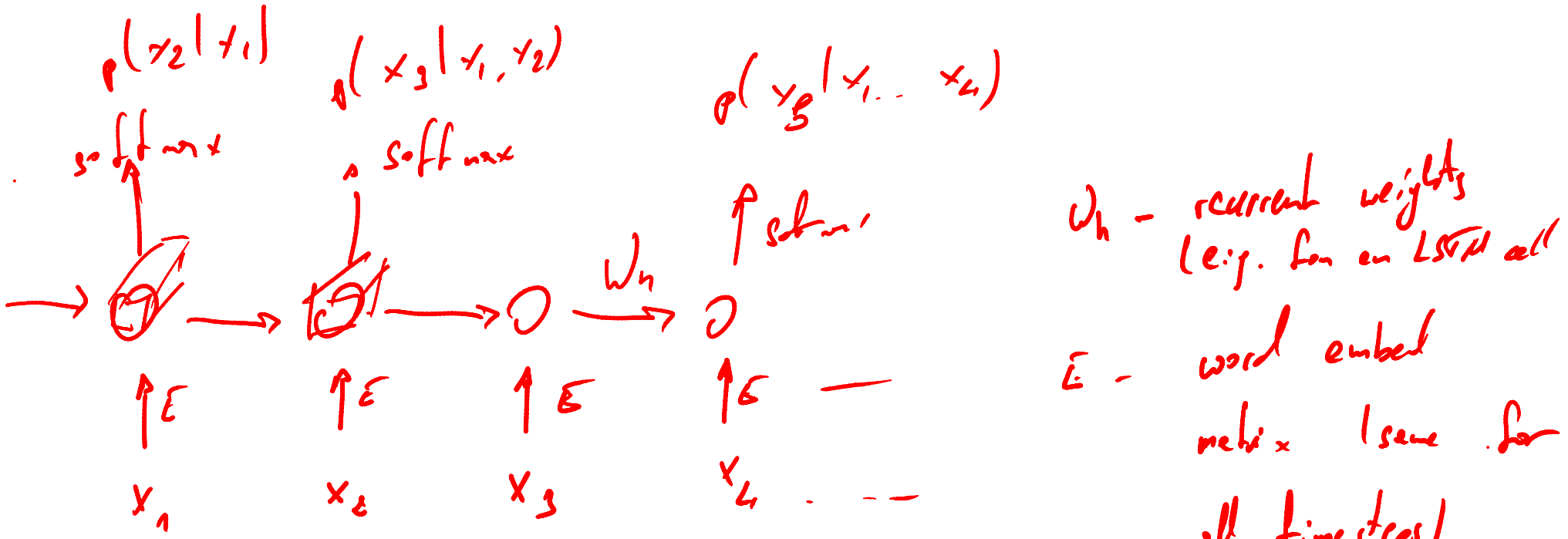
- Local connectivity
- Weight sharing
  all steps use same function
- Order dependent
  sequential computation
- Number of output proportional to number of inputs
  or forces only one input or one output



Recurrent Language Model

Eg. Recurrent LM

Task: predict $p(y_n \mid x_1 \dots x_{n-1})$



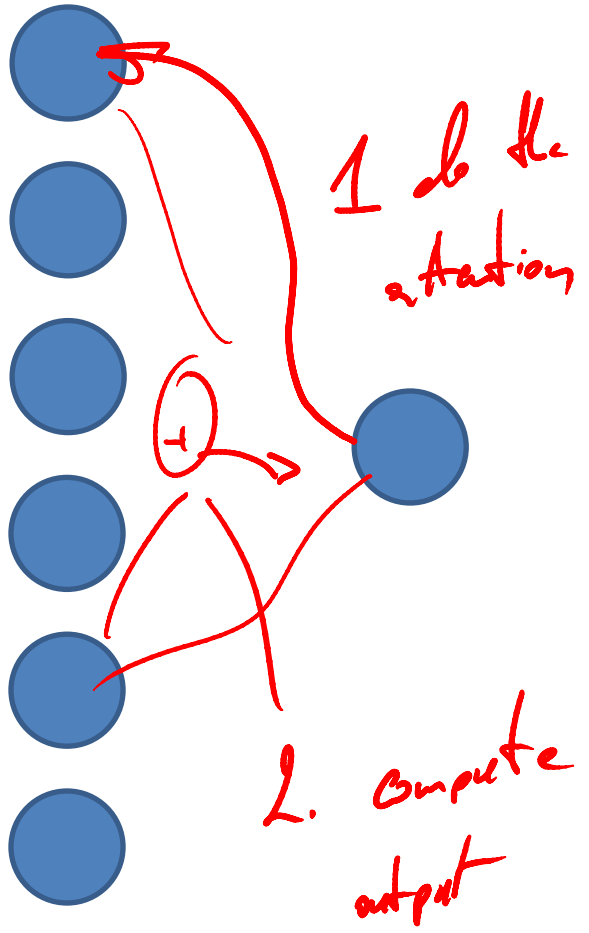
W_h - recurrent weights (e.g. for an LSTM all)

E - word embed matrix (same for all timesteps)

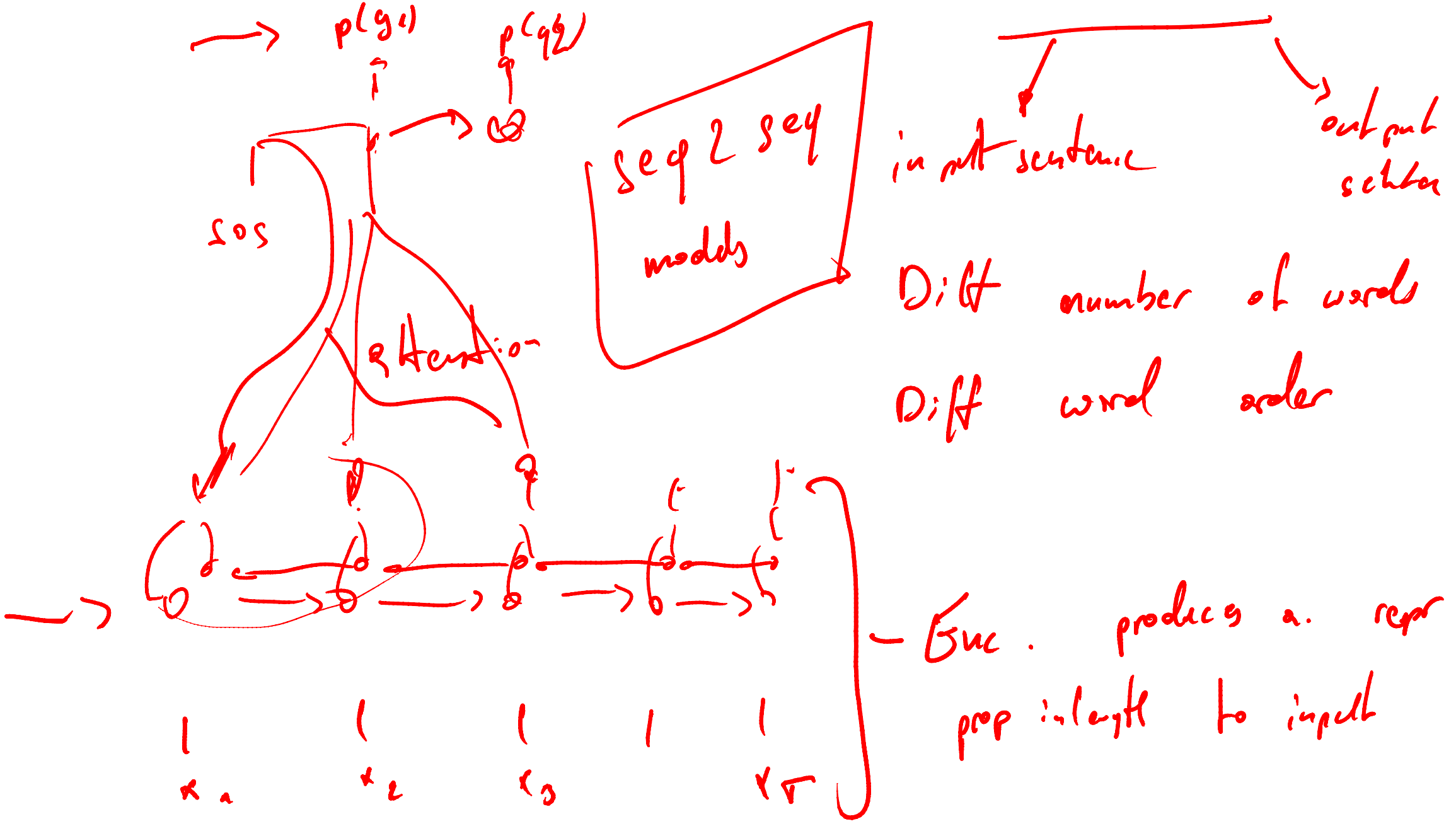
Different Layers for Different Problems

Attention neuron

- Global connectivity:
  The neuron selects some or all input units
- Works the same when inputs permuted
- Works for dynamically changing number of inputs



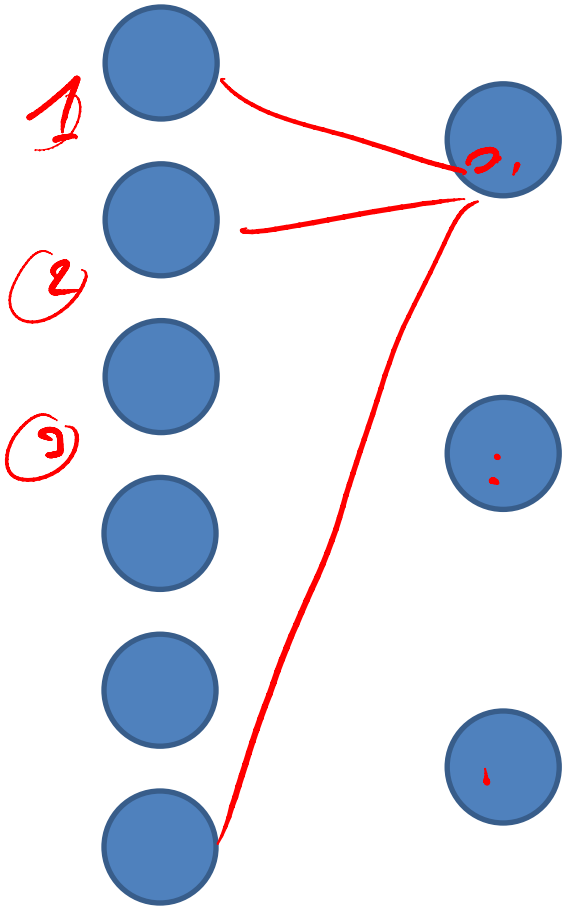
Recurrent Attention-Based Translation



Different Layers for Different Problems

Transformer (self-attention) layers

- Global connectivity – each output accesses all inputs using attention
- Weight sharing – all outputs use the same weights (but choose different subsets of inputs)
- Optionally can depend on input/output ordering (can use positional embeddings)
- Number of outputs independent from number of inputs



Transformer (self-attention) Translation

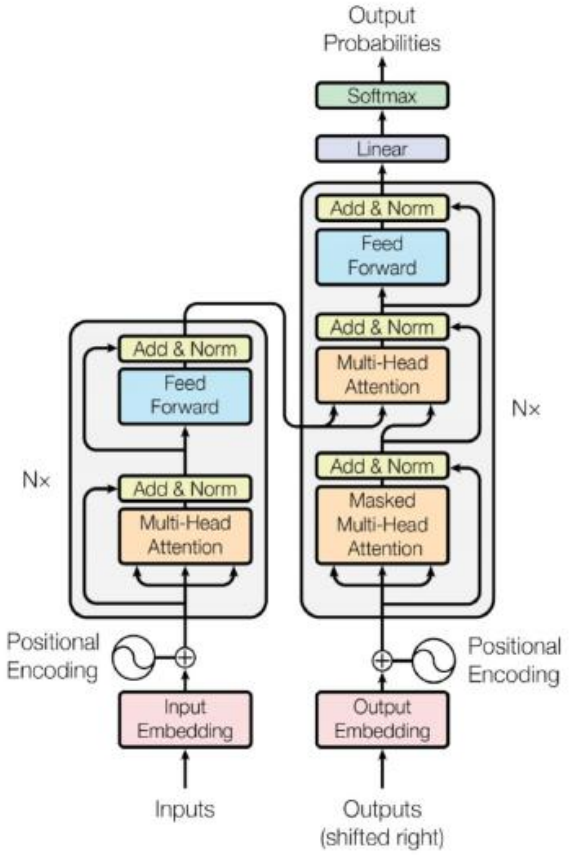
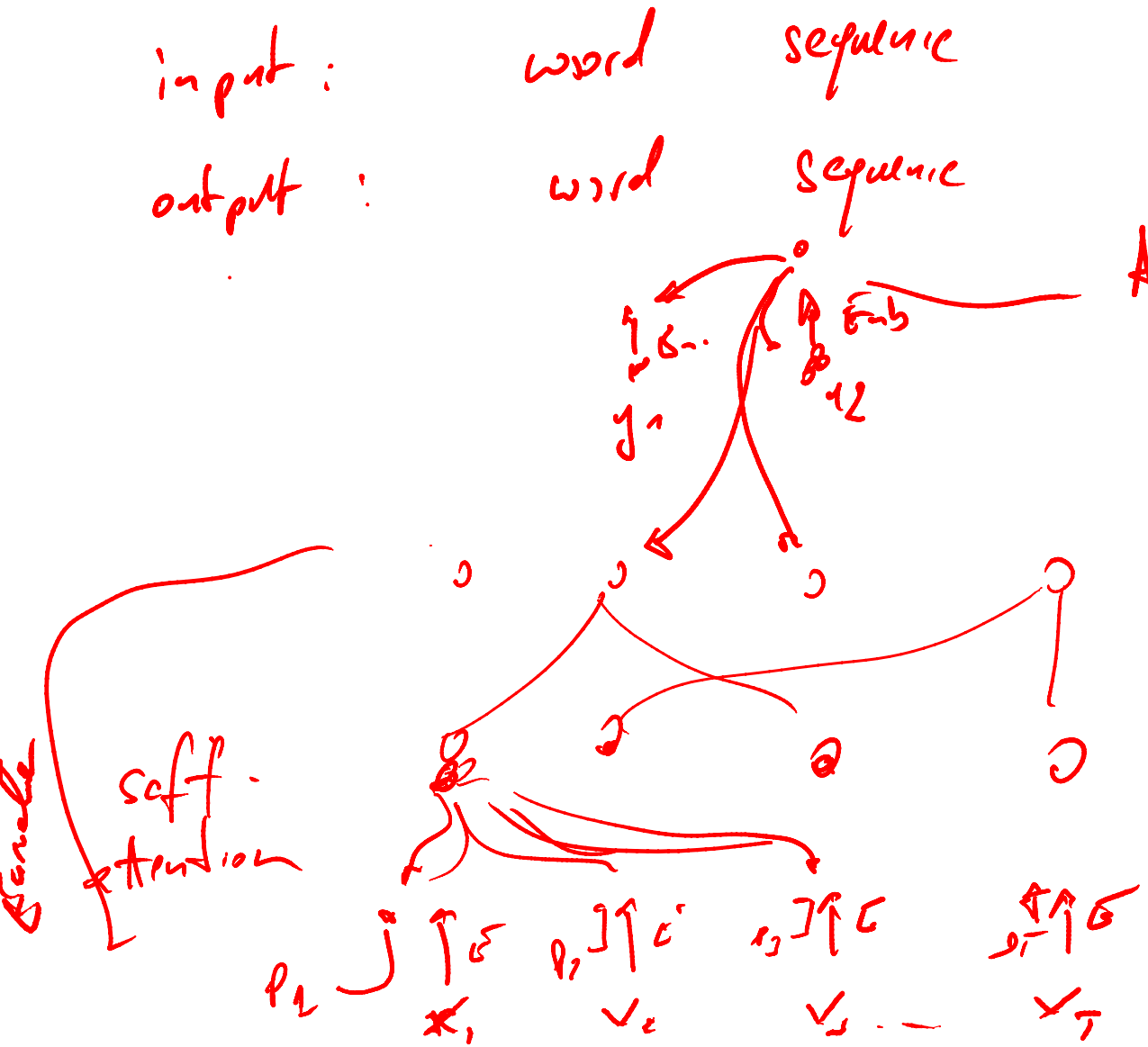


Figure 1: The Transformer - model architecture.

Transformer (self-attention) Translation



Architecture summary

Neural architectures exploit problem domain properties:

- Local vs global connectivity, match data layout
  1D (words, audio), 2D (images), 3D (movies, tomography)
- Weight sharing:
  - All image location use same set of filters
  - All words use same embedding matrix

SPEECH AND SOUND PROCESSING

Sounds

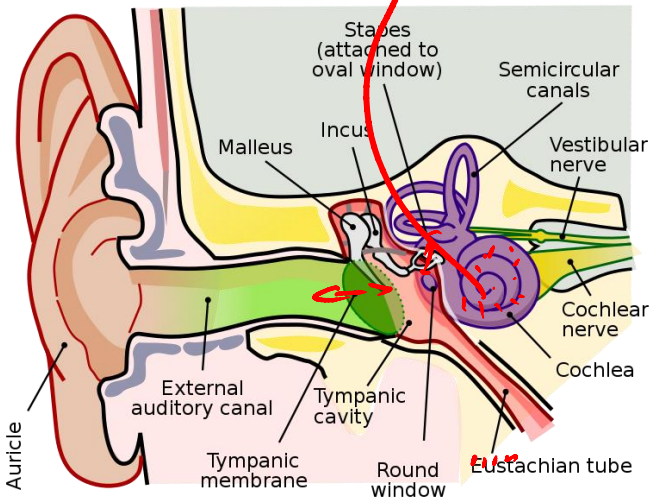
vibration

air starts to vibrate

propagate

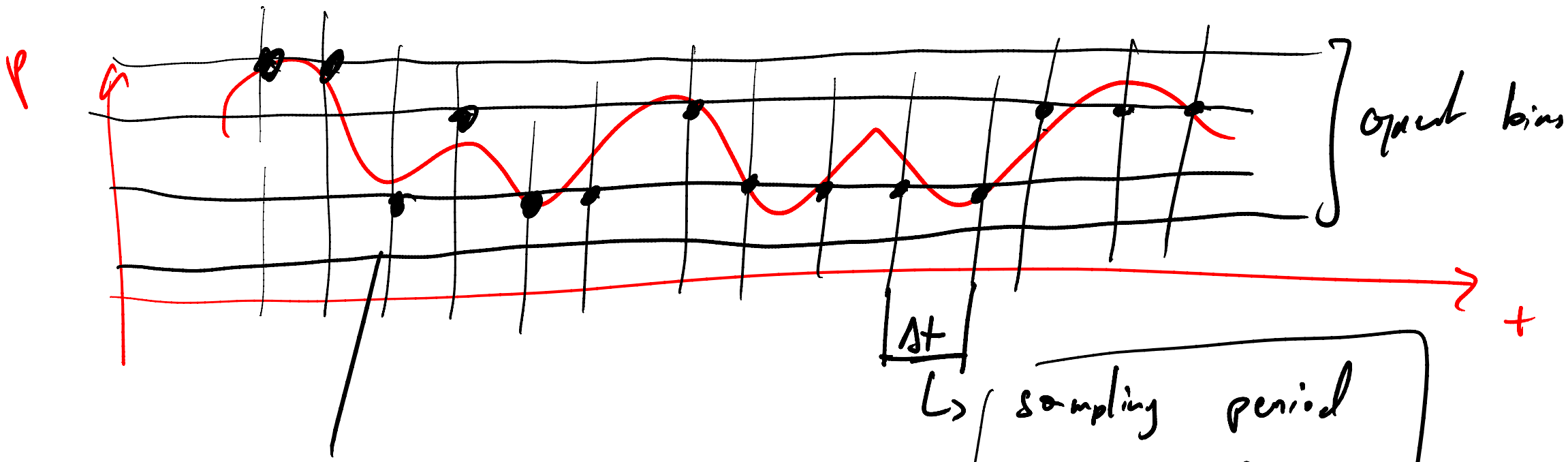
wave

we feel the movement of these tiny hairs



modulating an electrical property

Digital sound: discrete in time and magnitude



$\mathbb{R}^T$

seq of floating point values

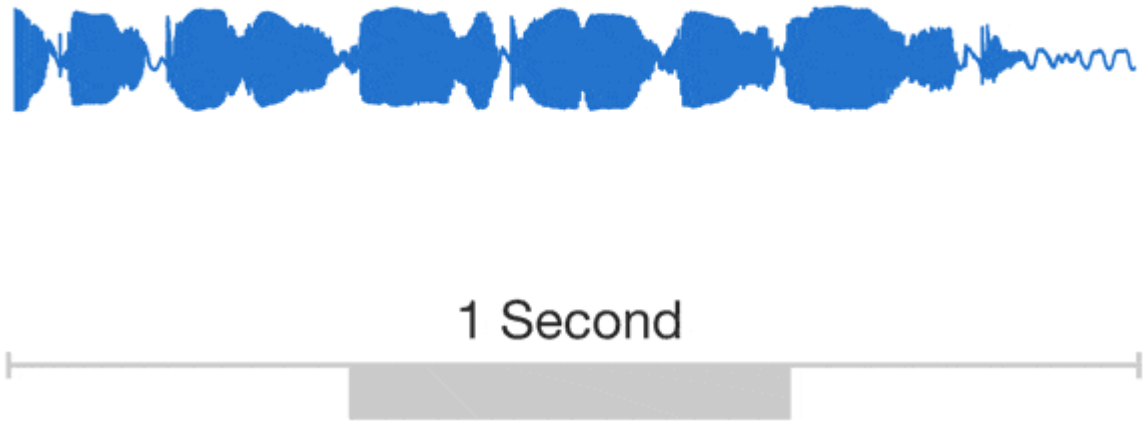
sampling period

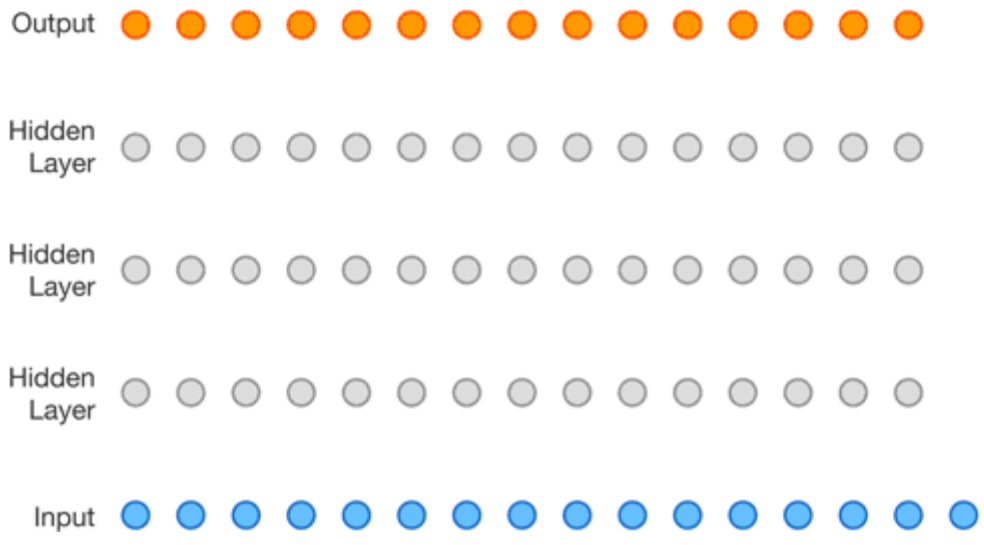
sampling frequency

depend on the signal

44kHz

Nnets for direct sound sample processing
WaveNet: a convnet for raw audio





Conditoned on text

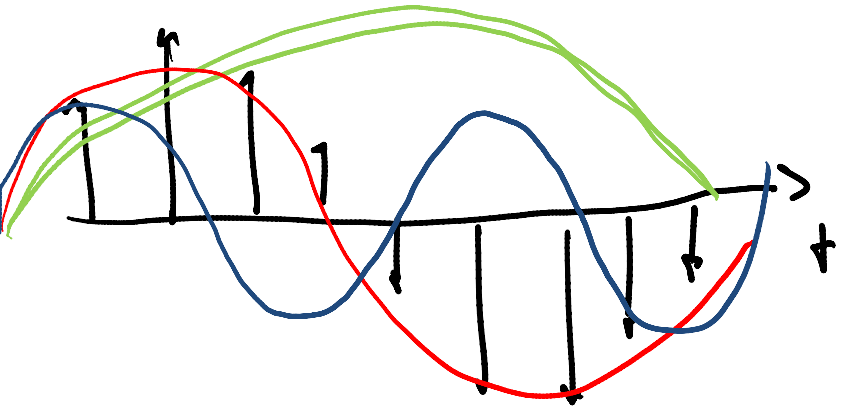
Unconditional

Piano

Wavenet: https://arxiv.org/pdf/1609.03499.pdf https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Fourier Transform: Time <-> Frequency

Sampled signal

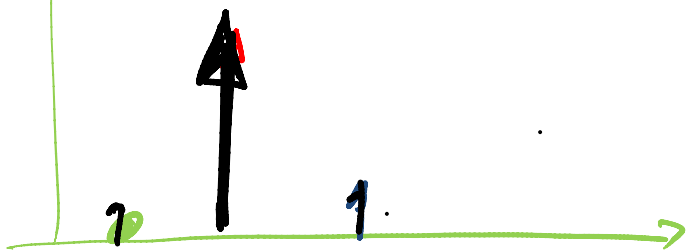


sequence of floats

correlate with sin waves with different frequencies

Fourier Transform

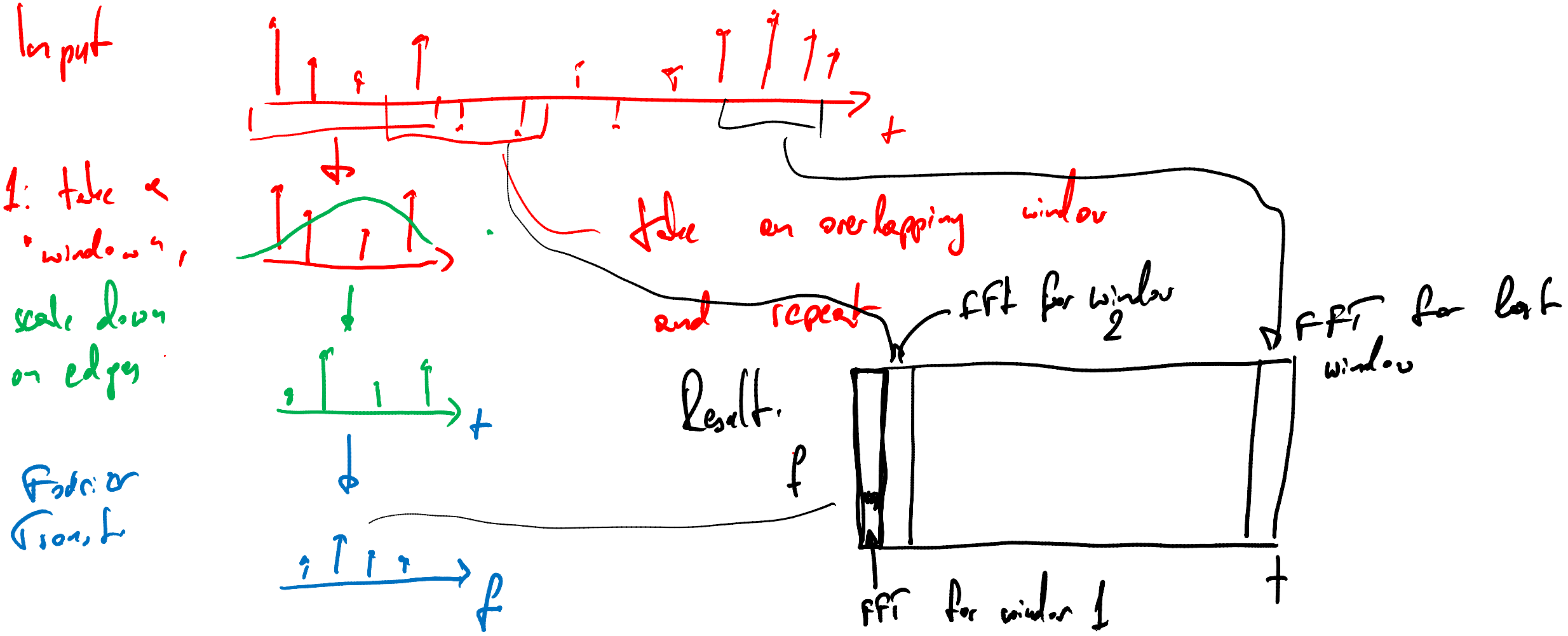
amplitude + phase - complex number



we will only look at amplitudes and drop the phase

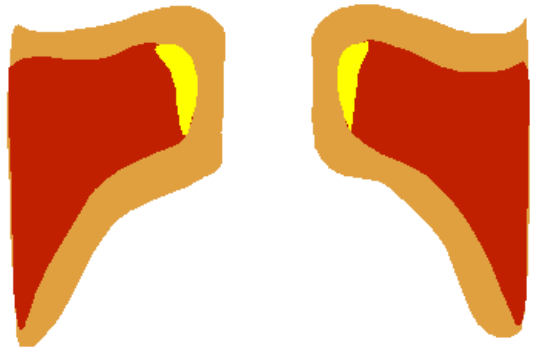
Short Time Fourier Transform STFT

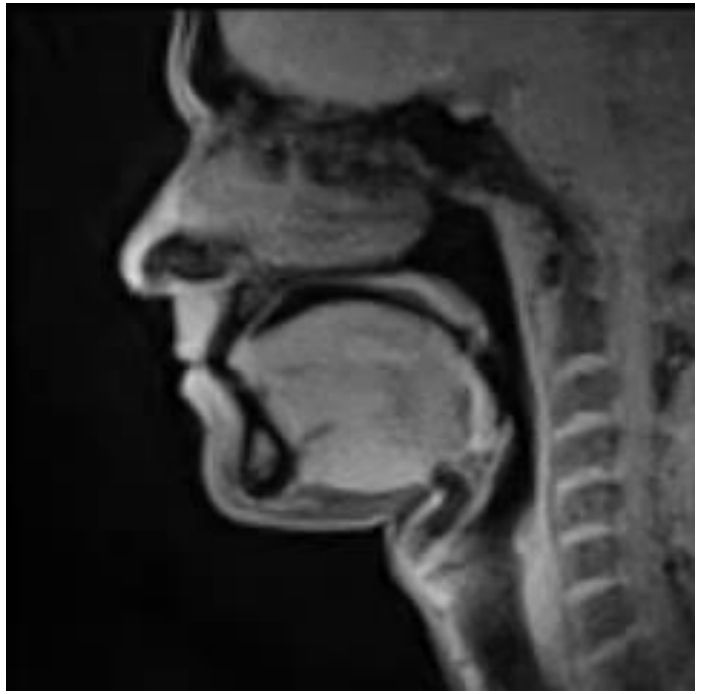
Intuition: time signal <-> time x frequency representation



Speech

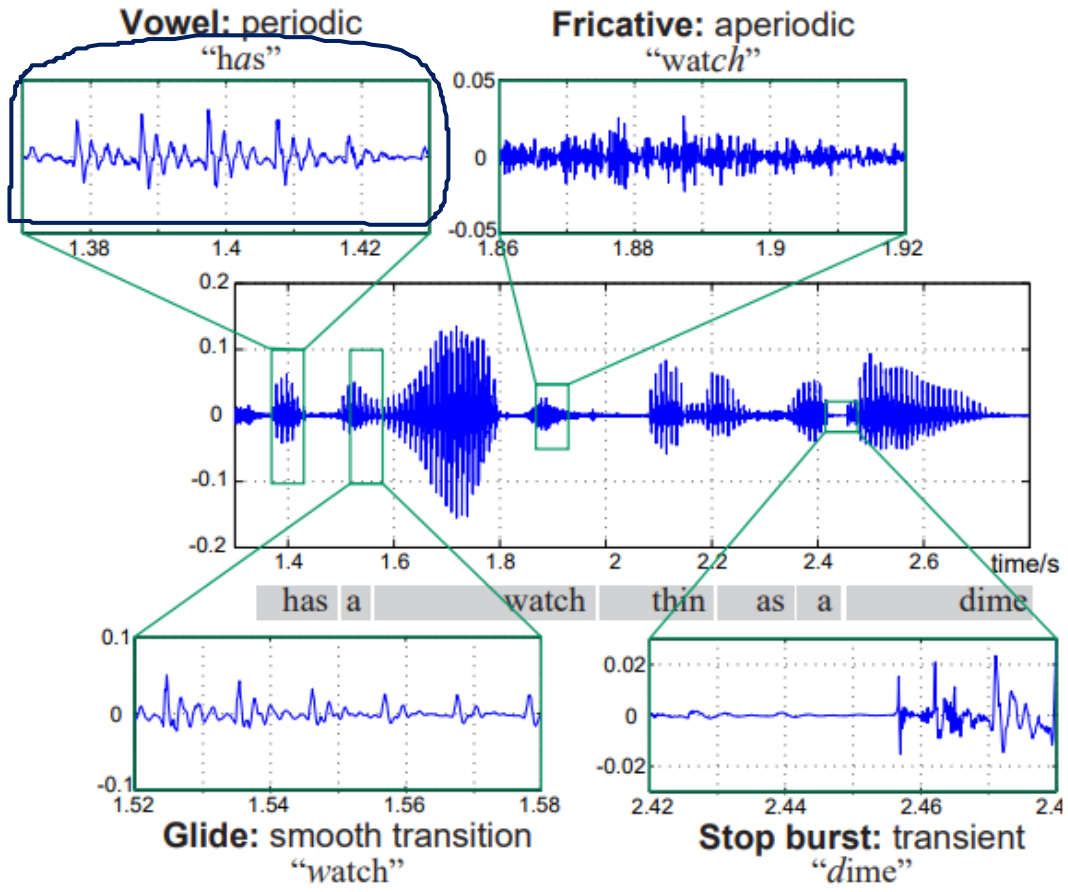
- Vocal cords vibrate (think of saying aaaa)
- The throat, mouth and tongue filter out some frequencies,





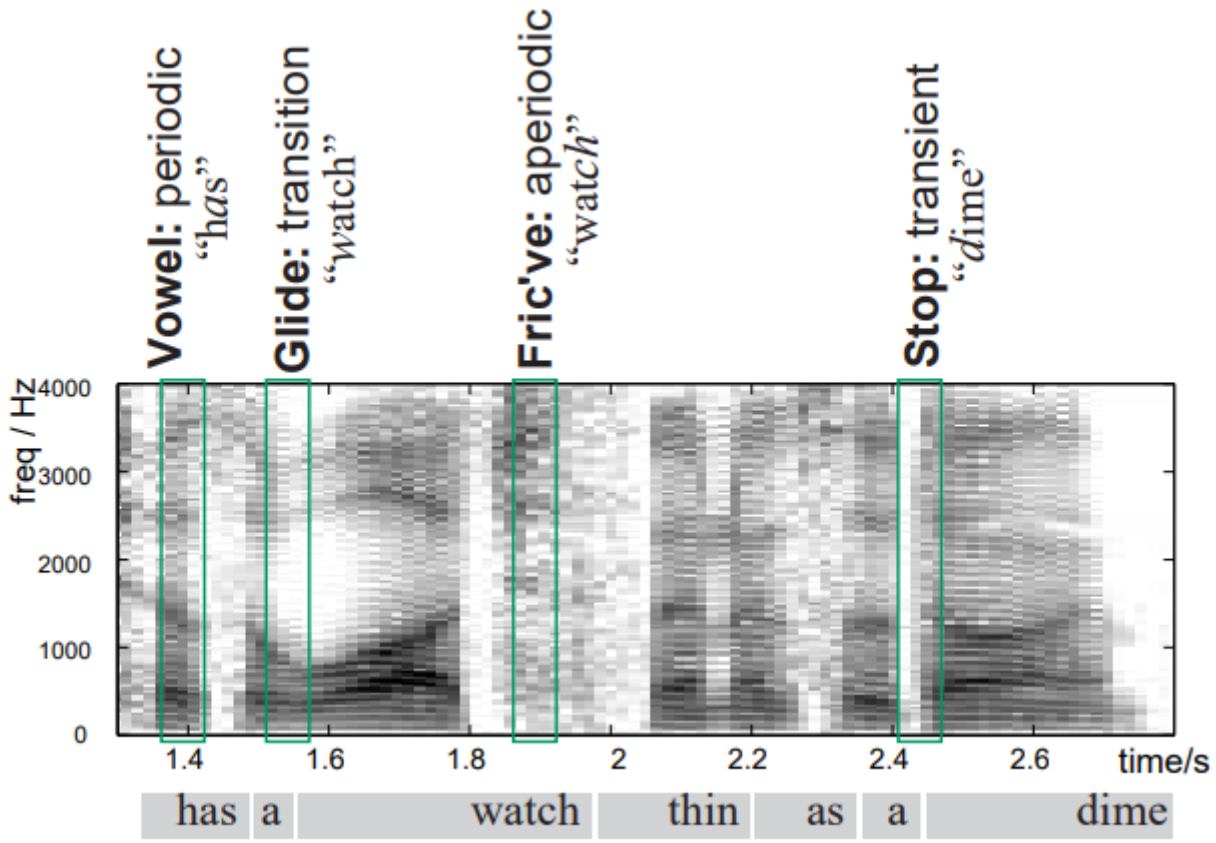
Media source: Wikipedia

Speech time domain signal



Slide from: http://www.ee.columbia.edu/~dpwe/e6820/lectures/L01-intro+dsp.pdf

Speech STFT



Slide from: http://www.ee.columbia.edu/~dpwe/e6820/lectures/L01-intro+dsp.pdf

SPEECH RECOGNITION

Speech recognition: prepare a transcription

“Speech to text”:
find a word sequence that matches the recorded utterance

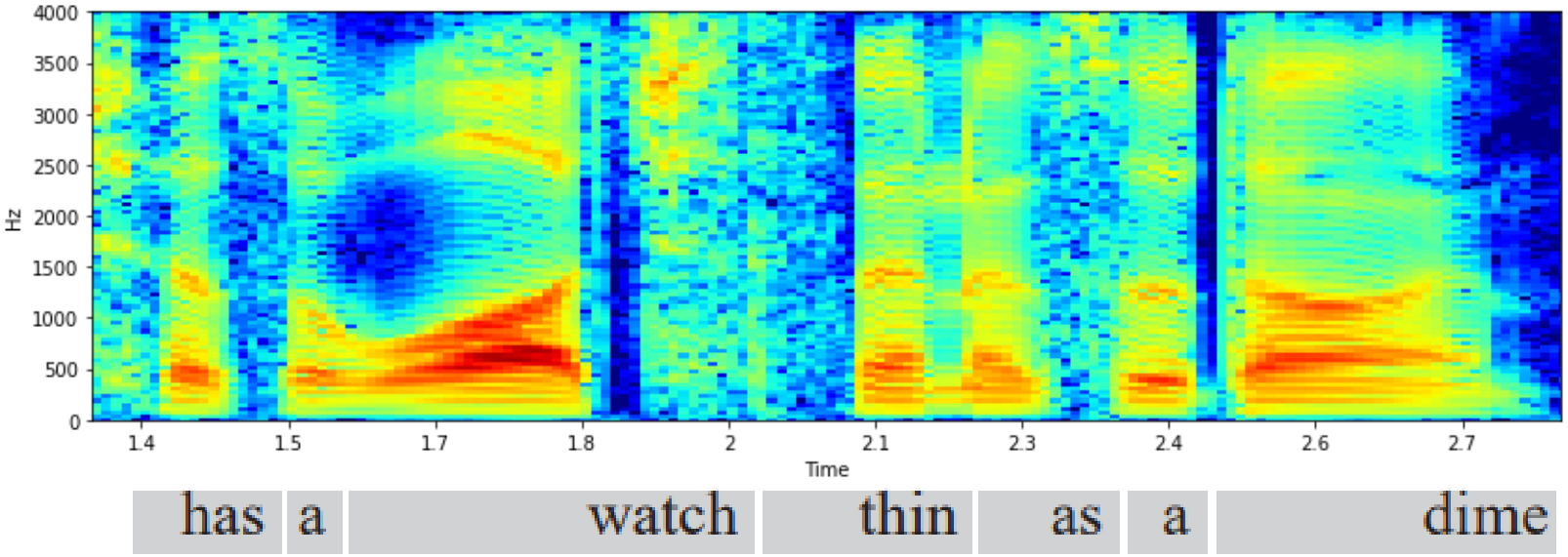
Error measure: Word Error Rate (WER)

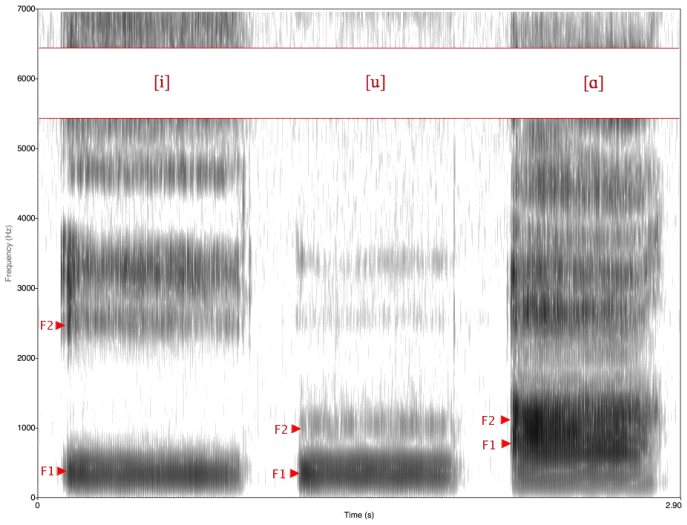
Reference:	It	is	a	sunny		day	
Recognized:	It	was		sunny	all	day	
Word status:	OK	S		D	OK	I	OK

WER = (S + D + I) / len(ref)

What the net should discard/recognize?

- Discard: vocal cord frequency (f0)
- Keep: formant frequency (set position of the mouth, tongue…)





Speech Differences Between Individuals

- Differences between individuals:
  - f0 (base vocal cord frequency)
  - Speed, prosody
  - accents

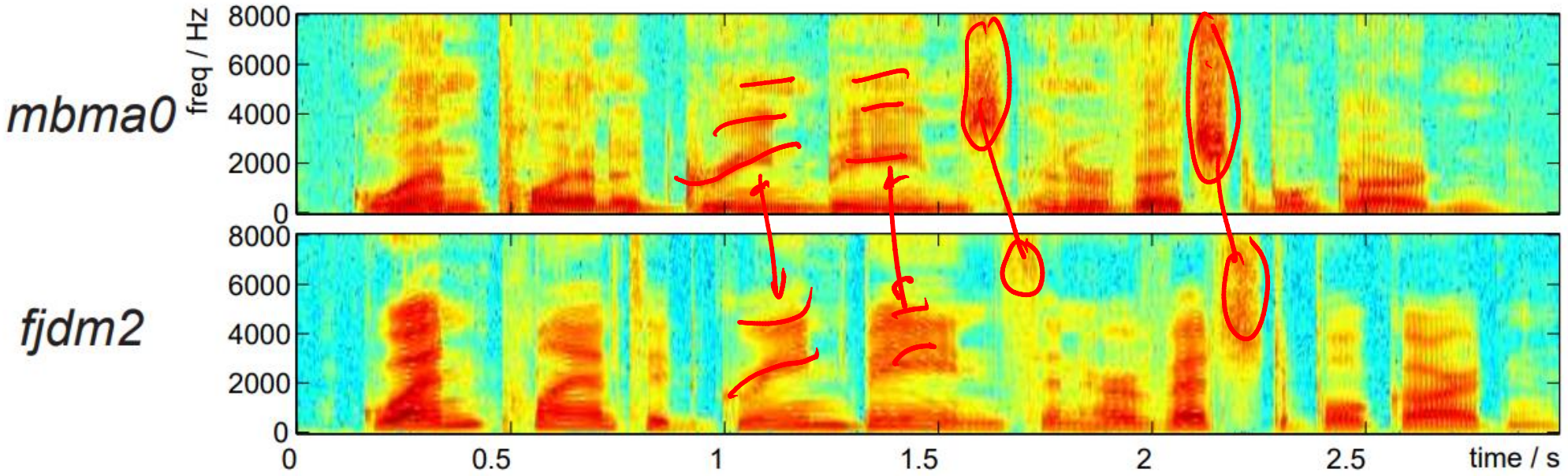
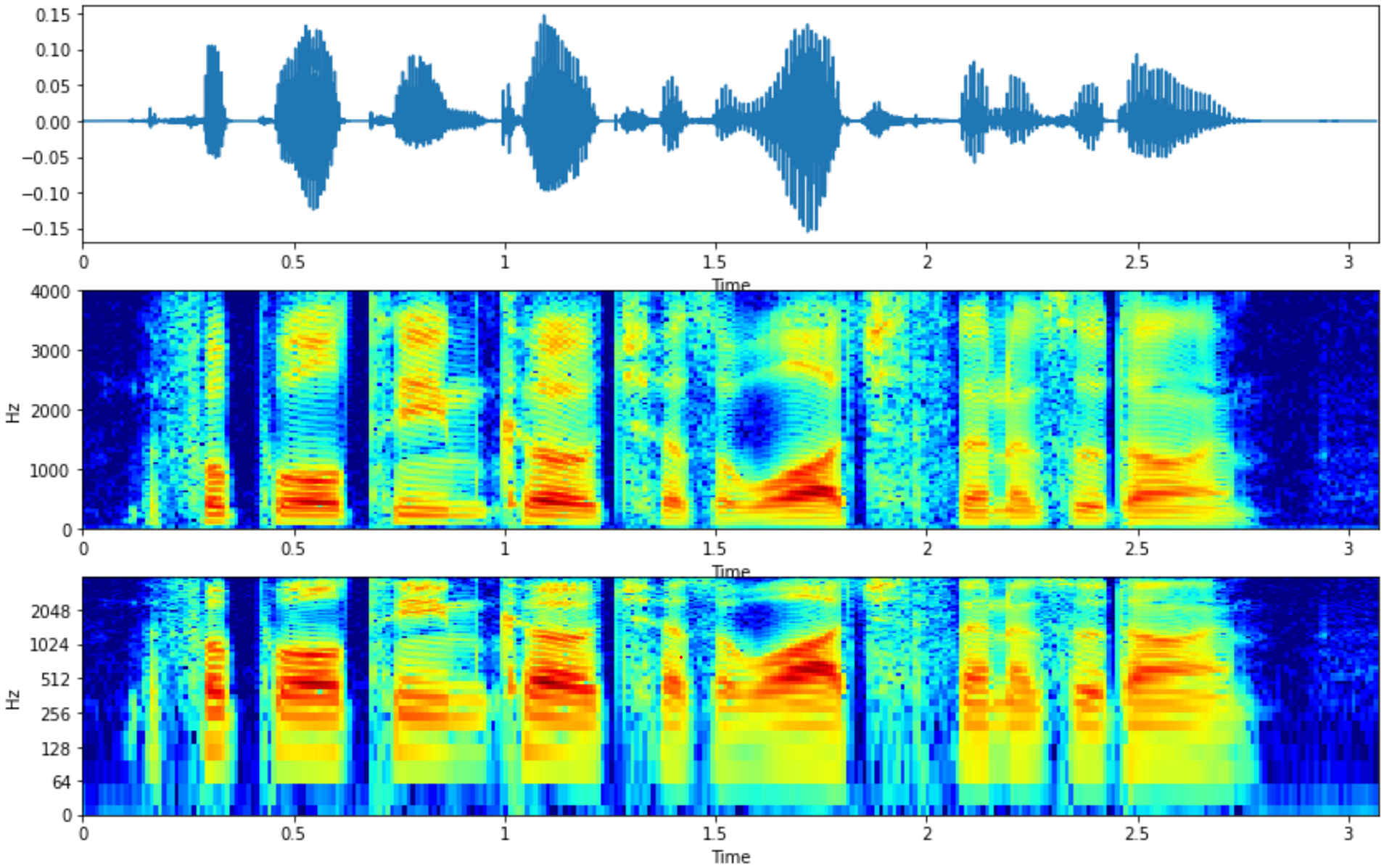


Image from: http://www.ee.columbia.edu/~dpwe/e6820/lectures/L09-asr.pdf

Speech Features for Neural Networks



- Raw

STFT

Log compress Freq axis

MEL scale

Speech Features: Intuitions

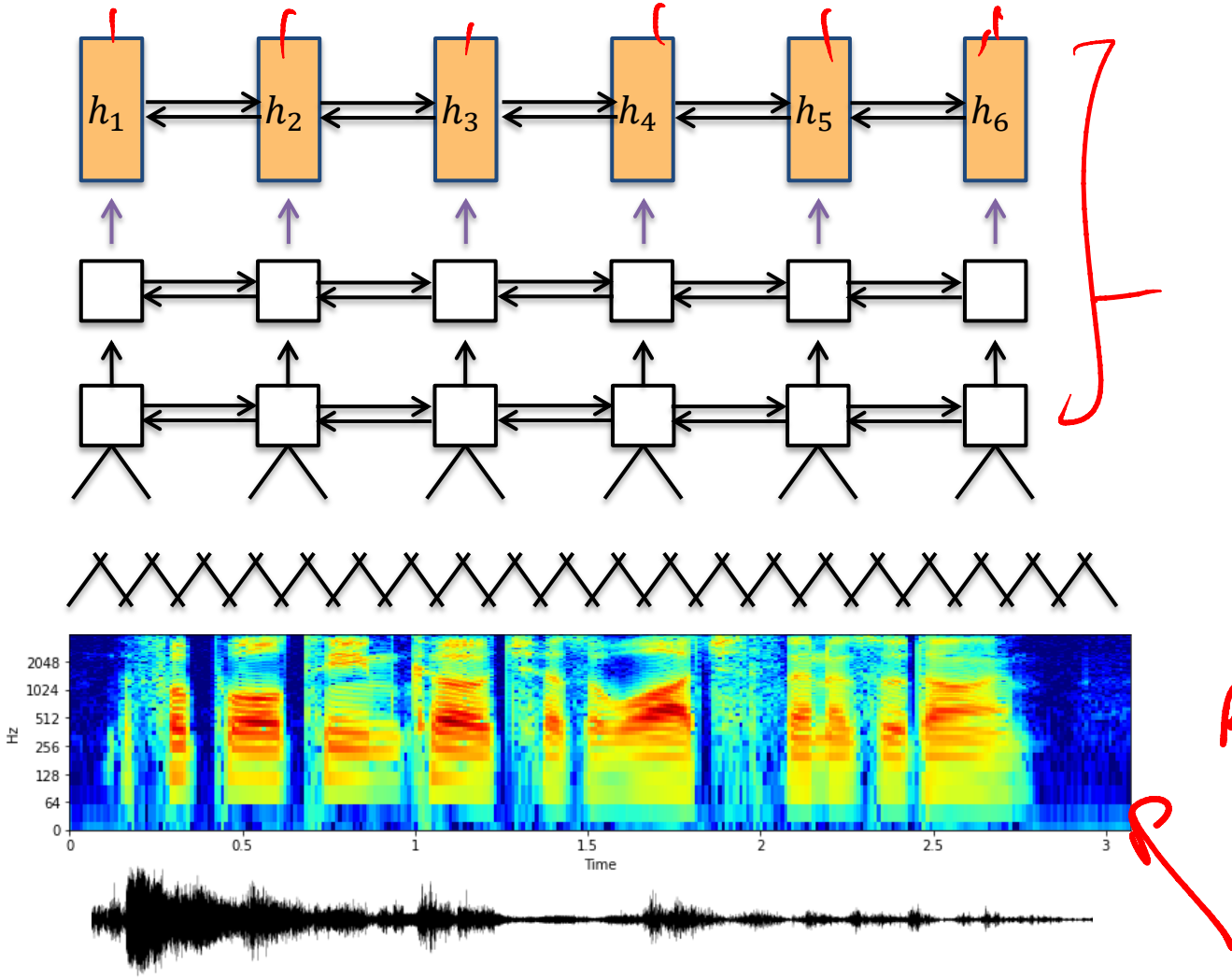
An utterance is represented as a $Time \times Frequency$ array of amplitudes.

Length along time axis varies between utterances.

The number of frequencies is fixed:

- Treat as a 1D signal with $|F|$ channels, apply 1D convolutions.
- Treat as a 2D signal (image) and apply 2D convolutions.

A typical speech processing architecture



(Bi) LSTMs

2D convs, per-freq bias

Nets: NBL

Speech – Time Alignment

Need alignments between:

- transcript (sequence of words, or phonemes, or graphemes)
- audio (sequence of frames, sampled uniformly in time)

Some possible solutions:

- Attention, treat as sequence-to-sequence task

- Design a loss function that handles the alignment (CTC)