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.

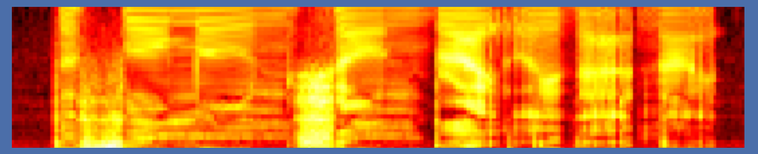
End-to-end speech recognition

Neil Zeghidour

Research scientist, Google Brain

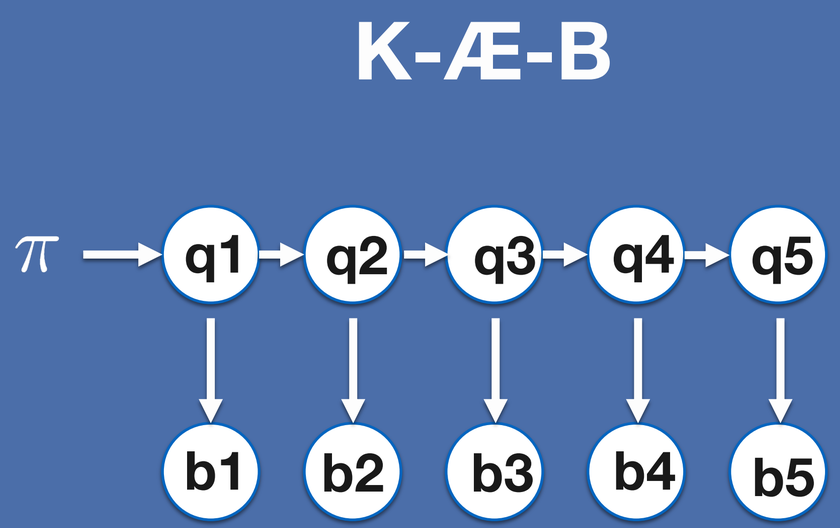
HMM-DNN Recap

- Extract speech features

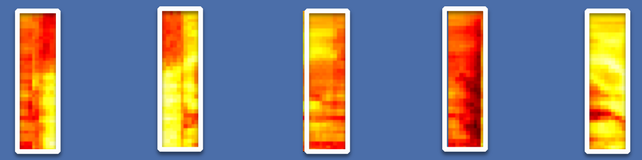


HMM-DNN Recap

- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions

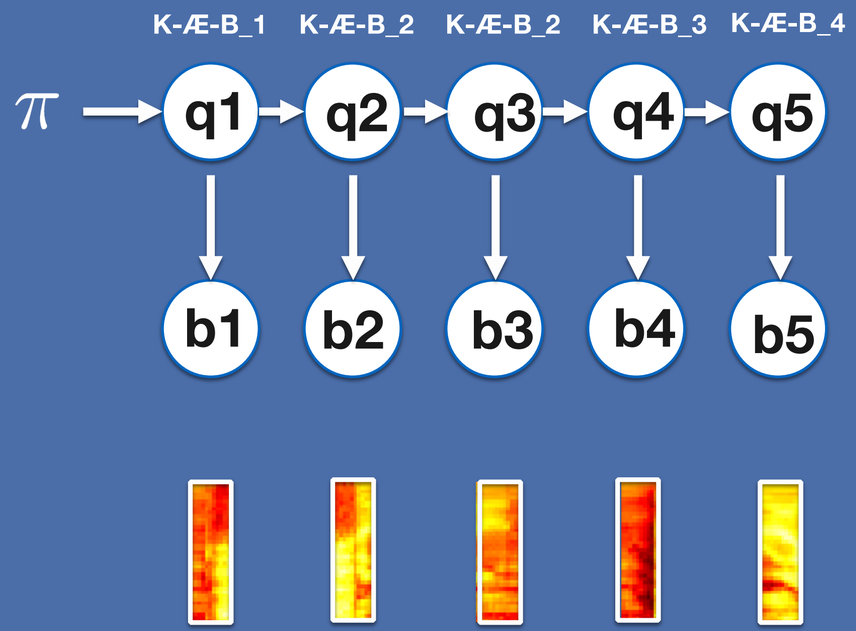


$$b_j \sim \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm})$$



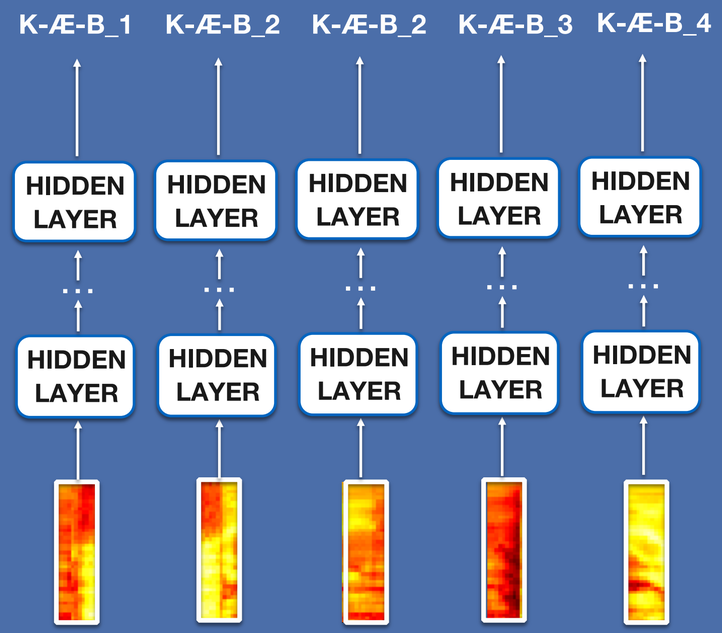
HMM-DNN Recap

- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions
- Use the Viterbi algorithm to extract an alignment of feature frames with states



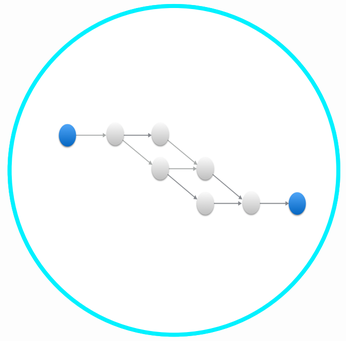
HMM-DNN Recap

- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions
- Use the Viterbi algorithm to extract an alignment of feature frames with states
- Train a Neural Network to predict each state from its frame



End-to-end training

- One neural network
- No alignment
- No speech features

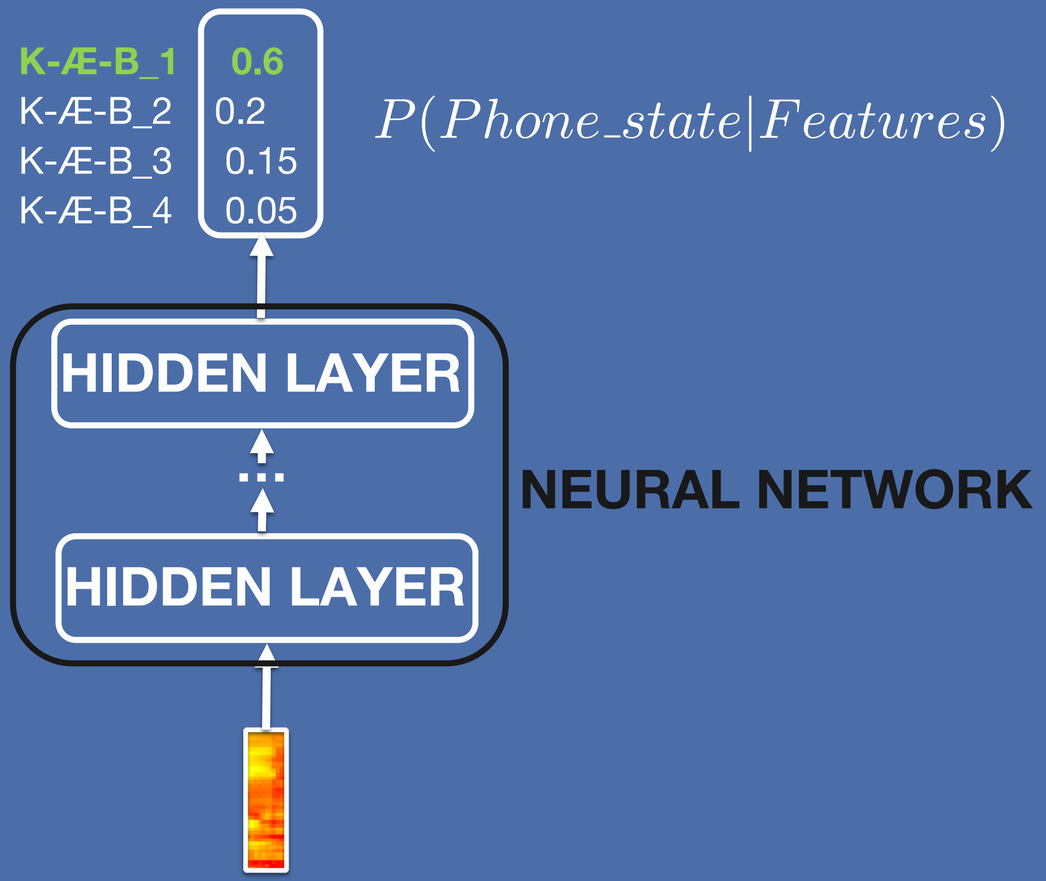


End-to-end speech recognition

- Connectionist Temporal Classification
- Deep Speech 2

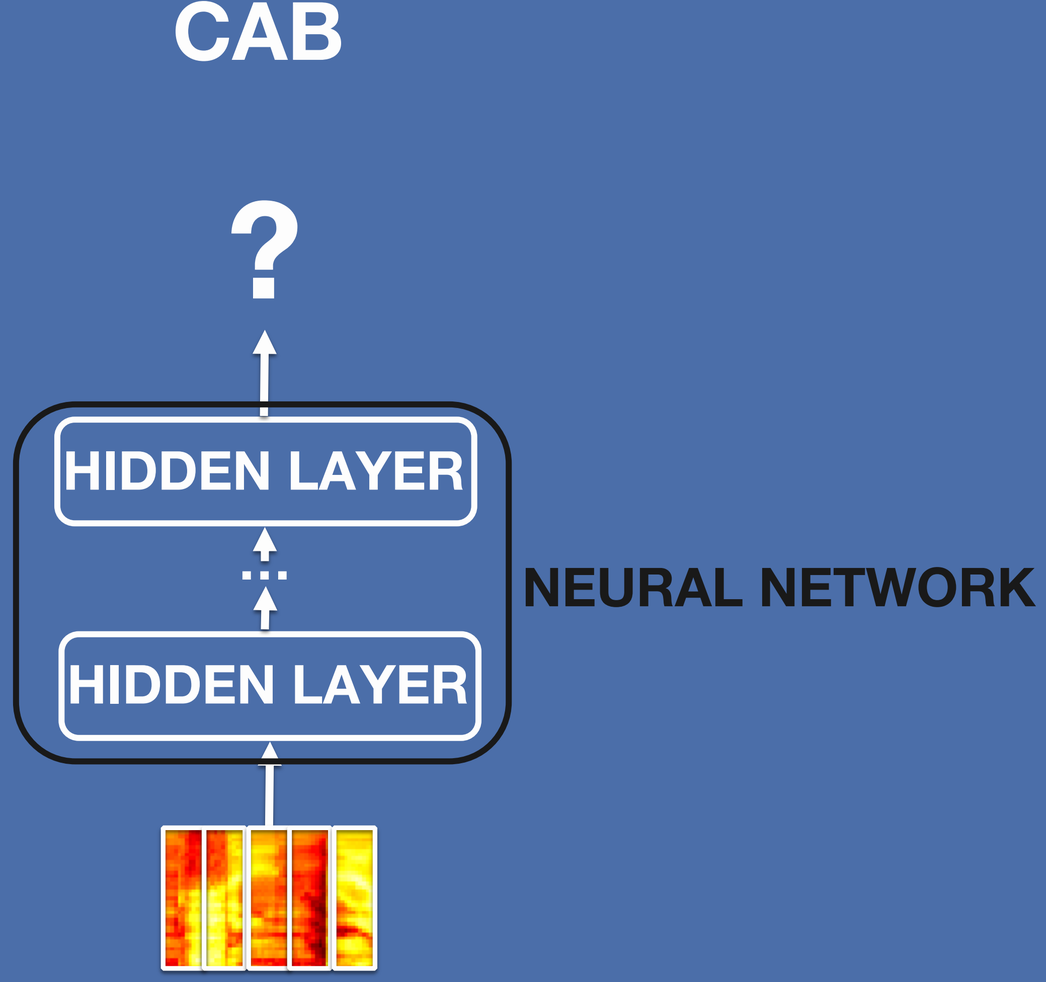
Training a DNN without states?

- When a state is attributed to each frame it is trivial to define the loss function: classification loss



Training a DNN without states?

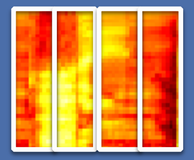
- When a state is attributed to each frame it is trivial to define the loss function
- Without alignment, we have the word transcription and the feature frames
- Train a speech recognition system directly from the features to the transcription (no phonetic transcription, no pronunciation dictionary, no alignment)



Problem: no alignment available

- Phonemes last dozens to hundreds of miliseconds, while frames are sampled every 10ms
- Many more frames than phonemes/letters
- We need to learn **classification** and **alignment** jointly
- A loss function allows learning both at the same time: Connectionist Temporal Classification (CTC)

CAB



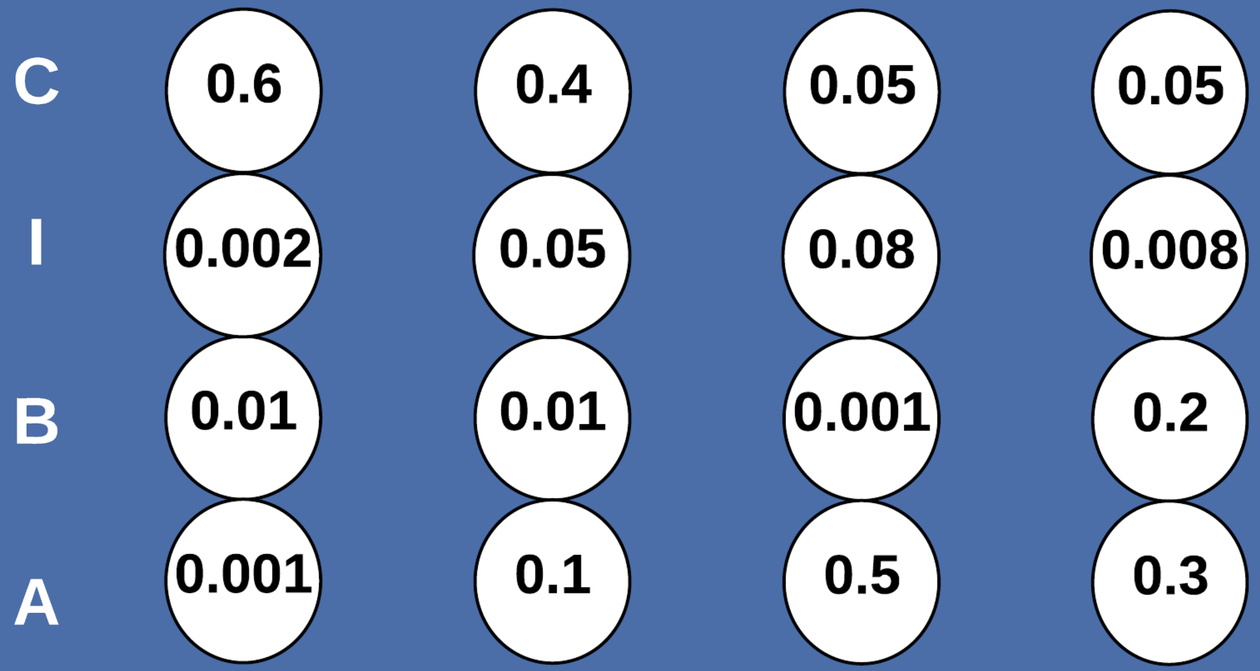
Connectionist Temporal Classification

- Given that the final layer of your network is a softmax with as many dimensions as there are characters, it can be seen as a probability distribution over characters

y_k^t the probability of character k at time t

$$y_k^t = Softmax(h_{N-1}(x_t))$$

$$= \frac{e^{w_k^T h_{N-1}(x_t)+b}}{\sum_{k=1}^{K} e^{w_k^T h_{N-1}(x_t)}}$$



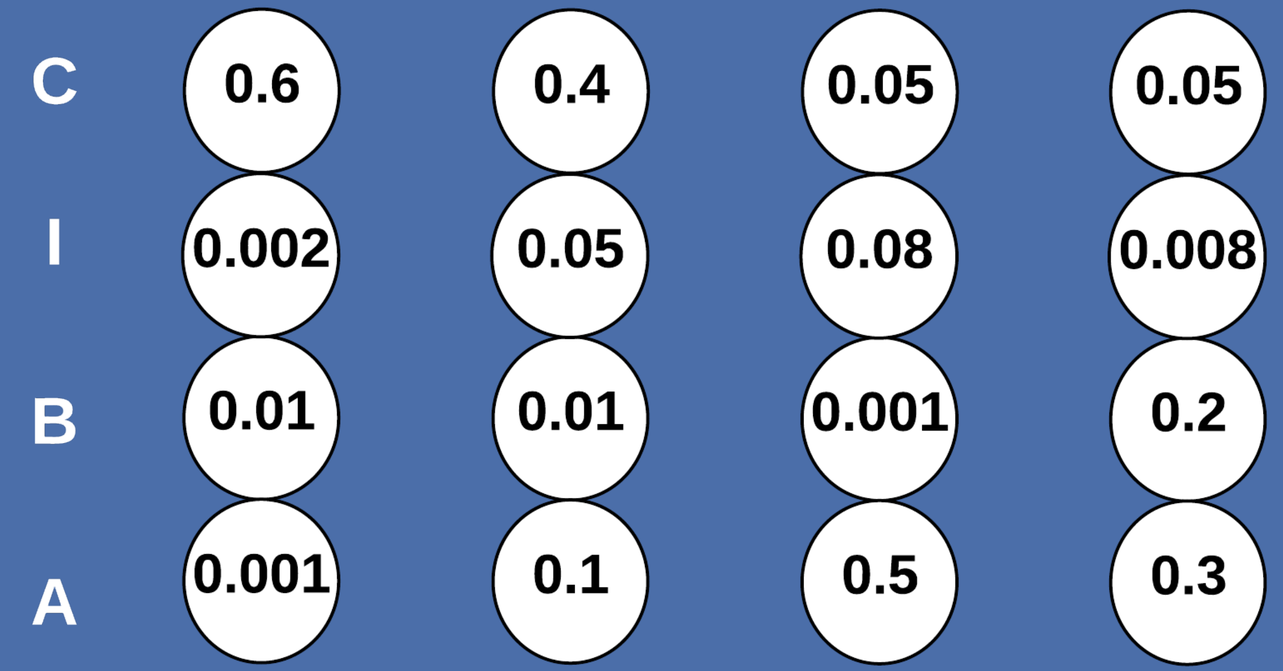
« Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks », Graves et al.

Connectionist Temporal Classification

- Given that the final layer of your network is a softmax with as many dimensions as there are characters, it can be seen as a probability distribution over characters

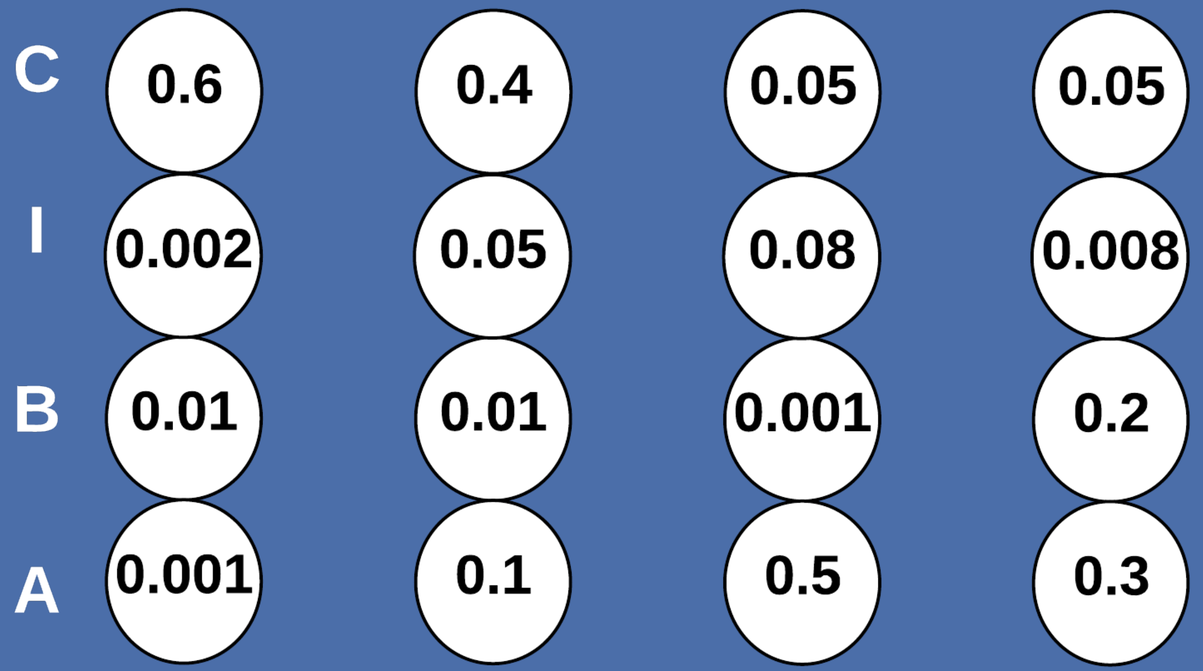
y_k^t the probability of character k at time t

$Y = \{A, B, C, ..., Z, apostrophe, space, blank\}$



Connectionist Temporal Classification

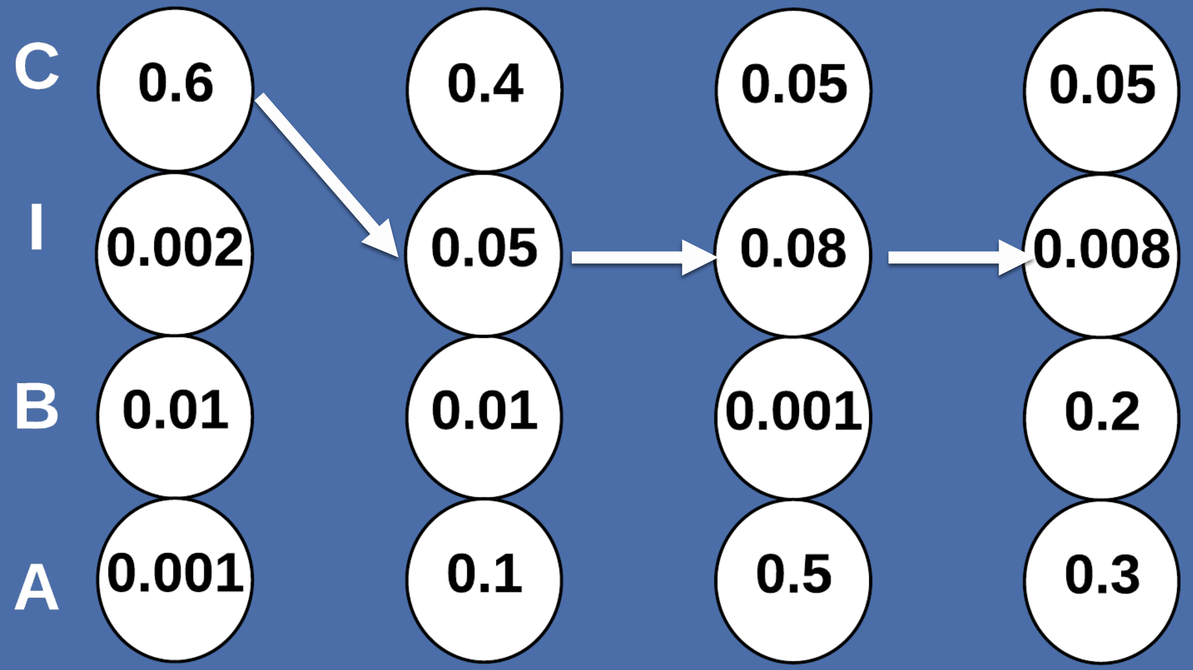
- Each output dimension is a **node**
- A **path** is a sequence of nodes
- Probability of a path is the product of the probability of the nodes (independence of time steps)



Connectionist Temporal Classification

$\pi = \pi_1, .., \pi_T$ a path in the graph

$$p(\pi|x) = \prod_{t=1}^{T} y^t_{\pi_t}, \forall \pi \in L^T$$

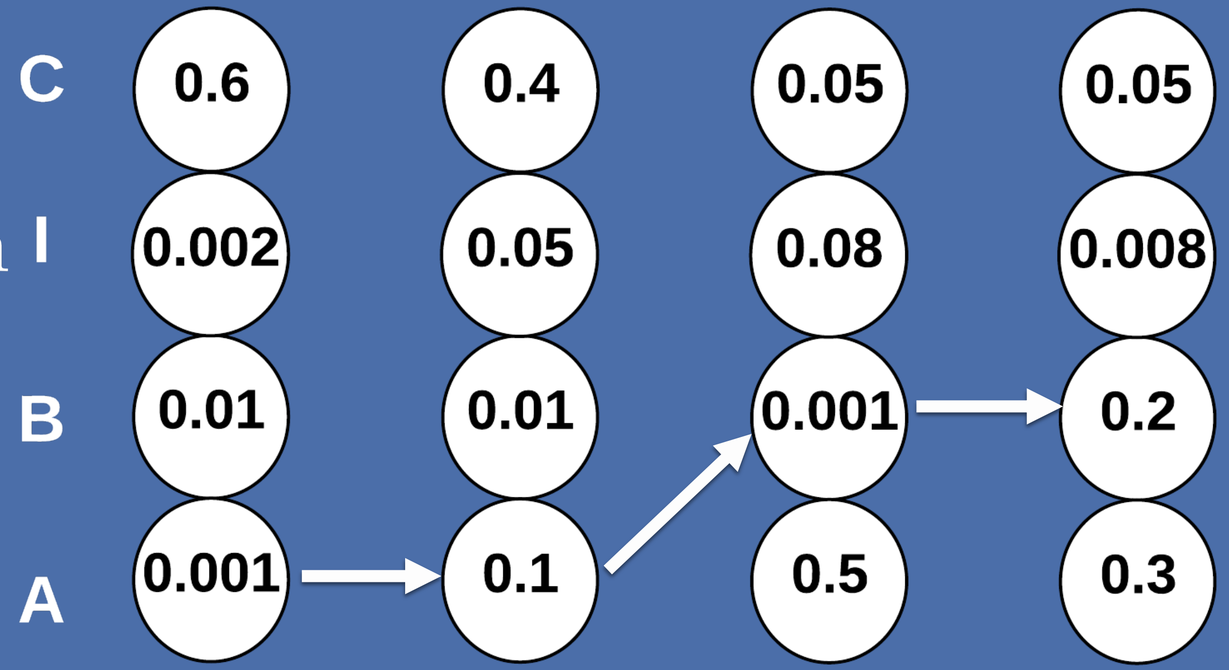


Probability(CIII) = 0.6*0.05*0.08*0.008 = 0.00000192

Connectionist Temporal Classification

$\pi = \pi_1, .., \pi_T$ a path in the graph

$$p(\pi|x) = \prod_{t=1}^{T} y^t_{\pi_t}, \forall \pi \in L^T$$



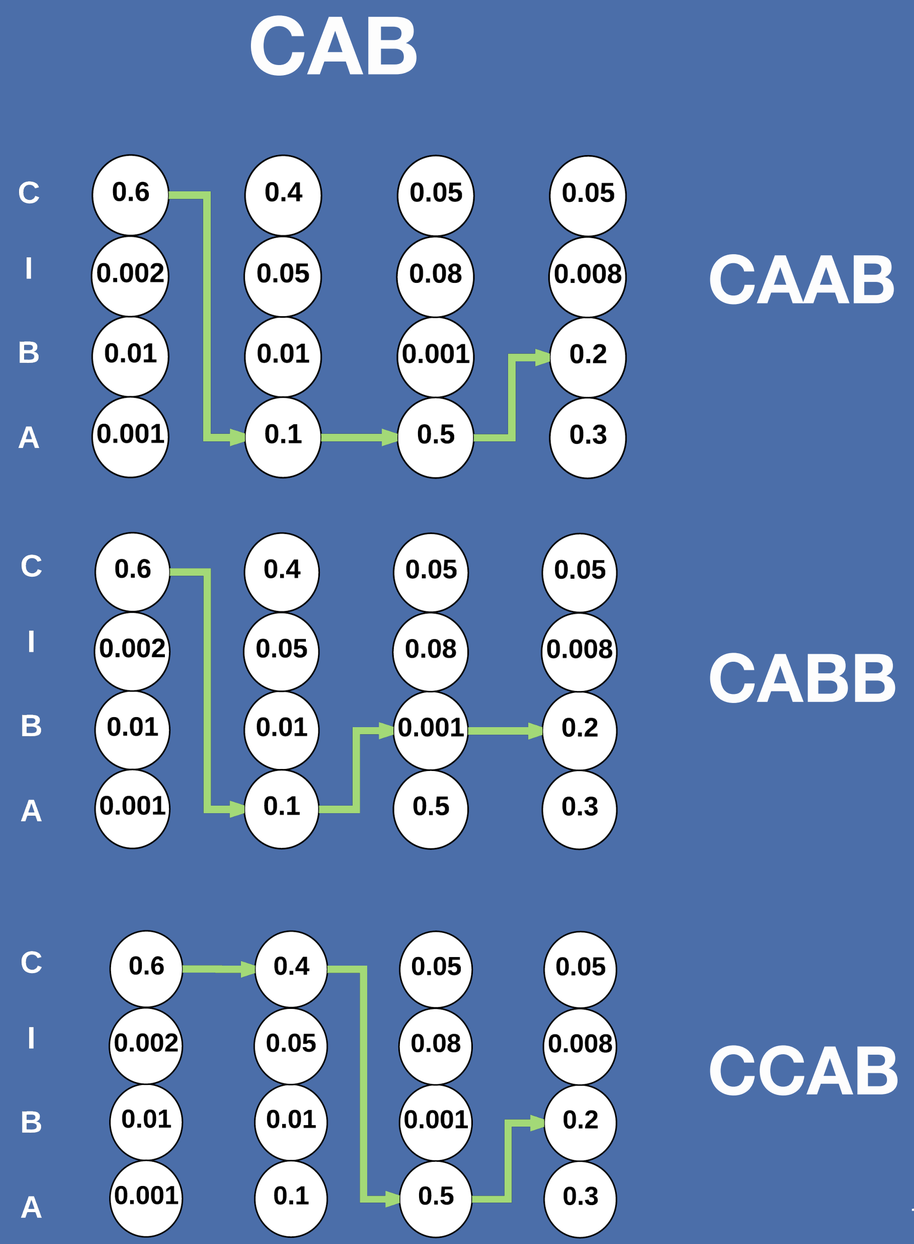
Probability(AABB) = 0.001*0.1*0.001*0.2 = 0.00000002

Training criterion

- Loss function:

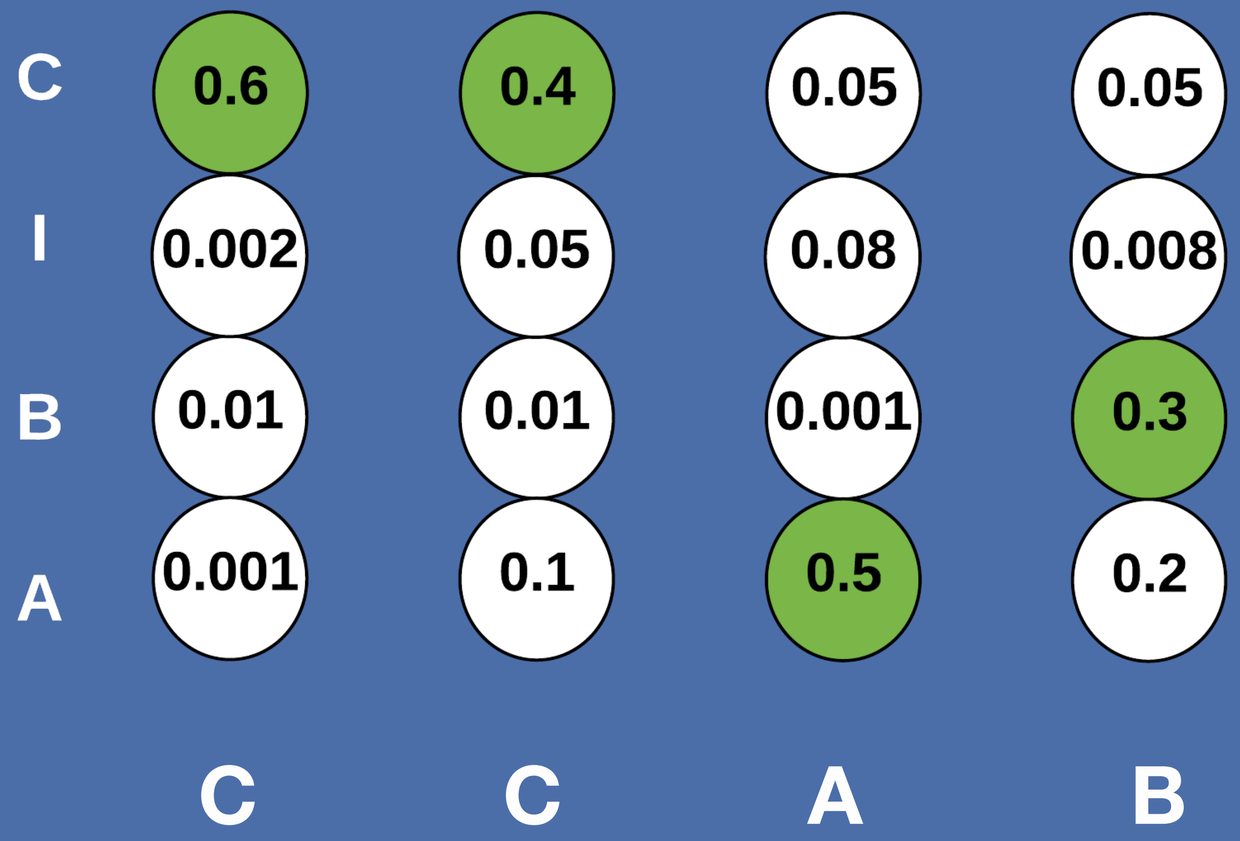
$$-\sum_{\pi \in \text{Valid paths}} P(\pi|x)$$

- Maximize the valid paths
- Nothing for the other paths
- Train the entire network with backpropagation



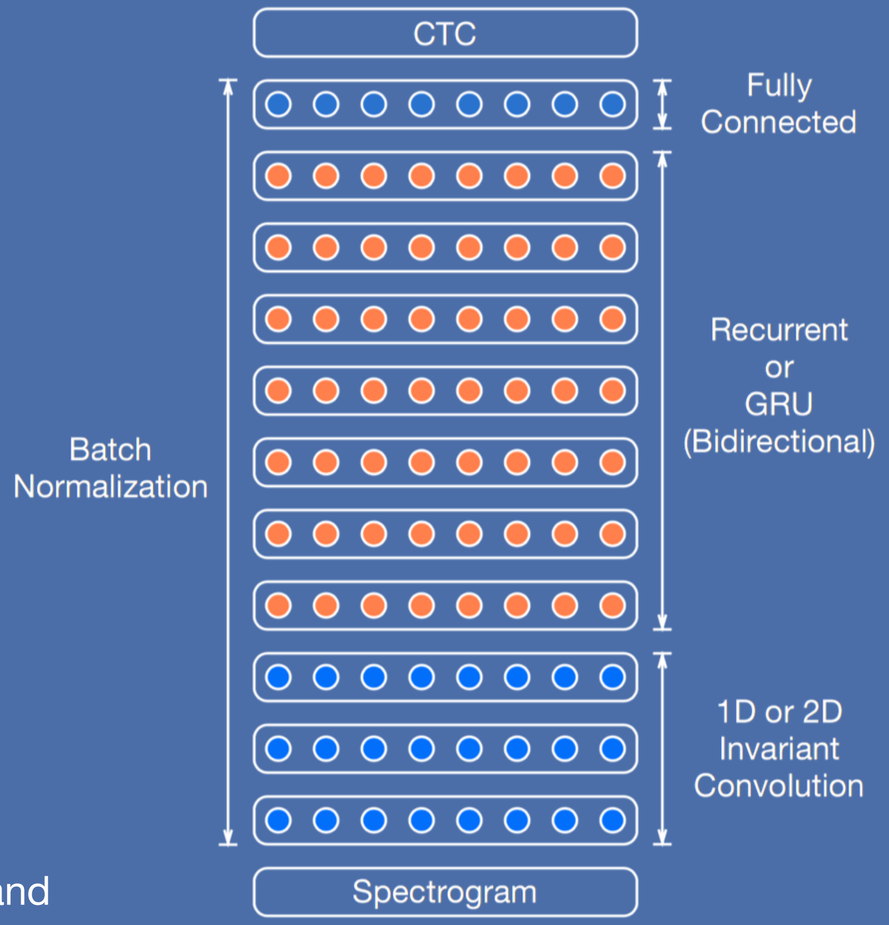
Decoding

- The basic decoding is just to take the most likely character at each step
- It can incorporate a language model by adding the probability of words



Deep Speech 2

- Baidu system
- Combines convolutions on spectrograms, bi-directional RNN and CTC
- 100m parameters (best models)
- 16 GPUs (50TFlops)
- 3 to 5 days of training
- No speaker normalization (enough data to learn invariances!)



« Deep Speech 2: End-to-End Speech Recognition in English and Mandarin », Almodei et al., 2015

Current state-of-the-art

- The measure of Performance in speech recognition is called WER (Word Error Rate): how many words the system failed to recognize (in %) –> **the lower the better**
- Best systems are now on par with humans on conversational speech in English (~5-10% WER)

Resources

- Kaldi: http://kaldi-asr.org/
- Wav2letter: https://github.com/facebookresearch/wav2letter

Current challenges

- Challenging types of speech
- Need of data: humans vs machines
- Reducing supervision in speech recognition

What about accented/noisy speech?

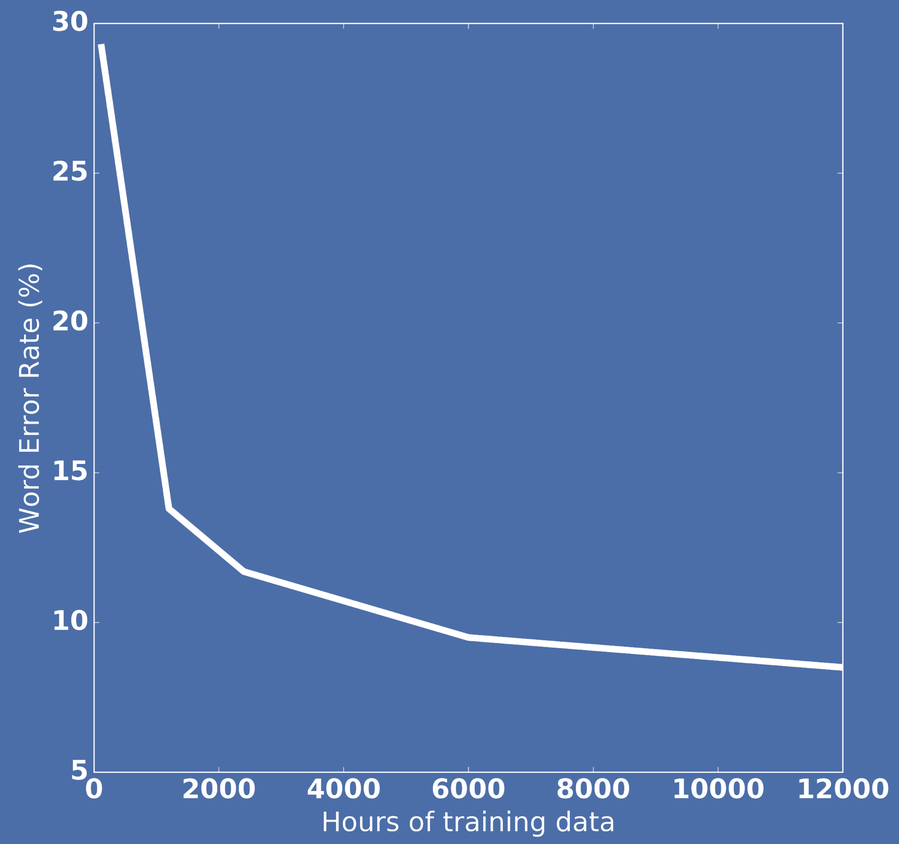
Accented Speech			
Test set	DS1	DS2	Human
VoxForge American-Canadian	15.01	7.55	4.85
VoxForge Commonwealth	28.46	13.56	8.15
VoxForge European	31.20	17.55	12.76
VoxForge Indian	45.35	22.44	22.15

Word error rate of DeepSpeech 2 on accented speech

Noisy Speech			
Test set	DS1	DS2	Human
CHiME eval clean	6.30	3.34	3.46
CHiME eval real	67.94	21.79	11.84
CHiME eval sim	80.27	45.05	31.33

Word error rate of DeepSpeech 2 on noisy speech

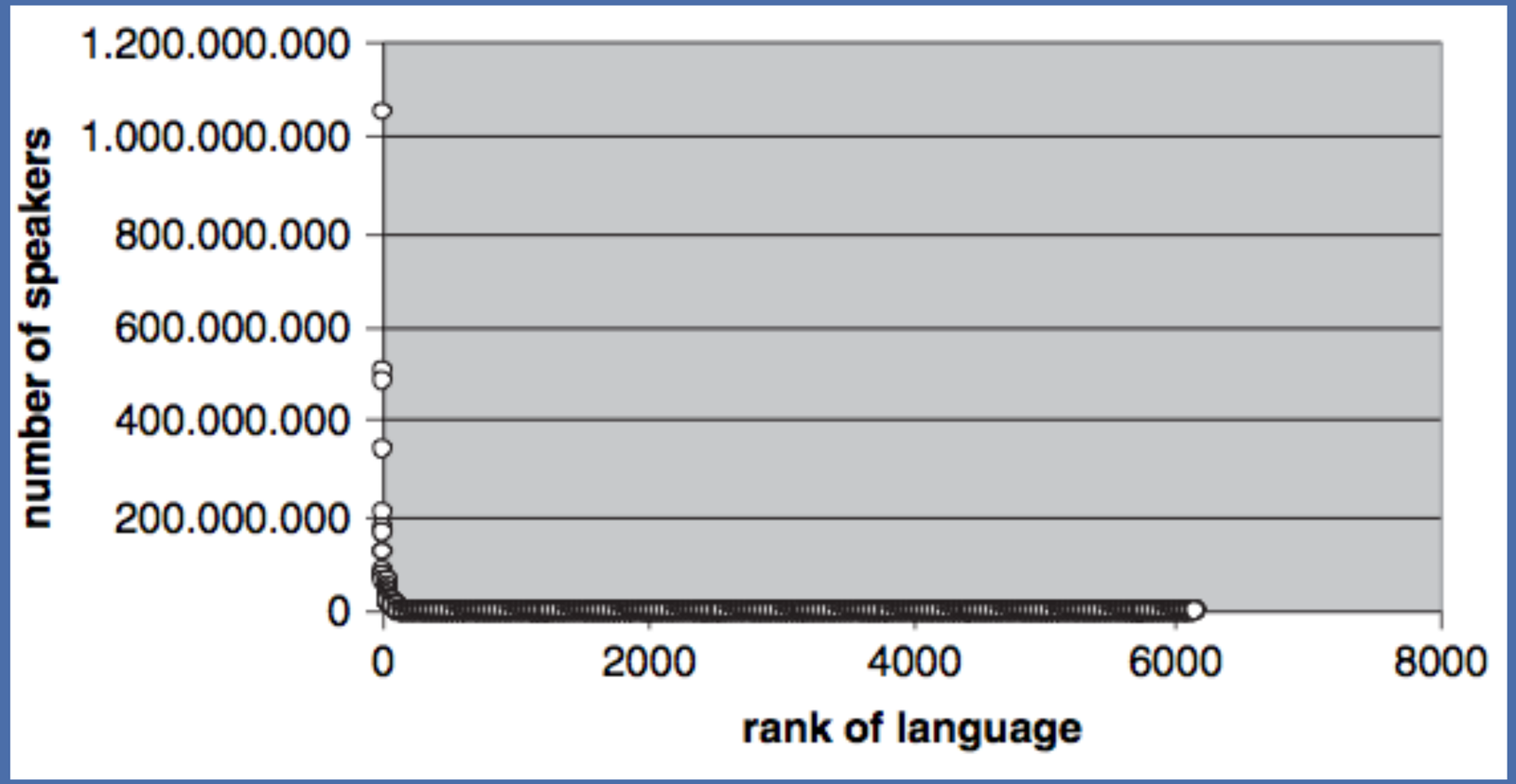
How much data do we need?



Challenges

- Annotating data is time-consuming and expensive
- Low-resource languages
- Languages without standard orthography
  Half of the ~7000 languages in the world
- Evolving vocabulary, new categories of users, new accents

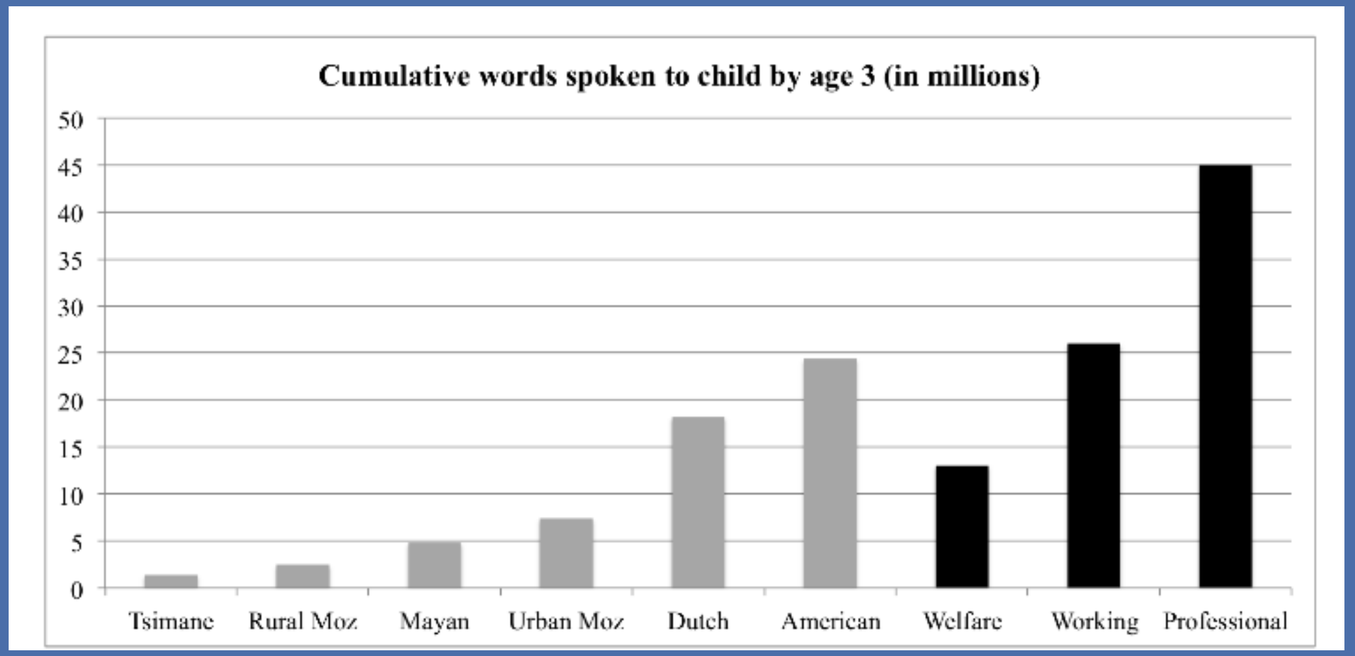
Low-resource languages



Efficiency of human learning

- Small and varying quantity of data
- No annotations!
- Two speakers!
- We learn speech recognition, robustness to noise, quick adaptation to new accents

From Cristia et al. (2017) Child Development.



66 h/year

800 h/year

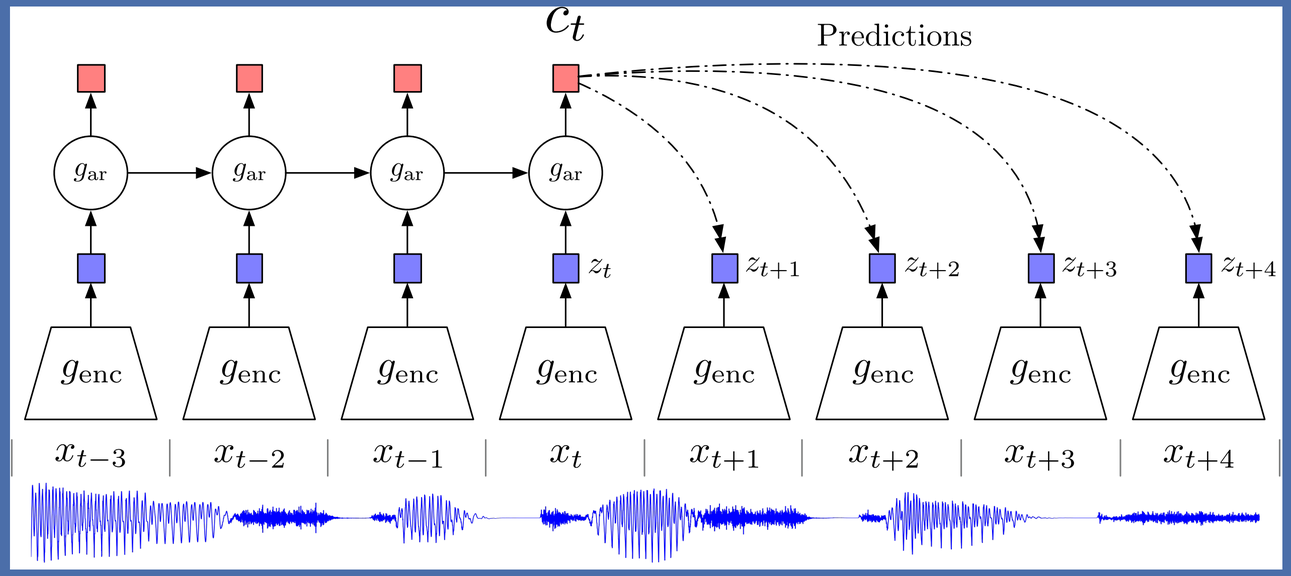
1200 h/year

Deep Speech 2
12000h

...

Semi-supervised speech recognition

- Pre-train a model with an unsupervised loss
- Contrastive predictive coding: predict future of an audio sequence
- No need for annotations, only audio (lots of audio)

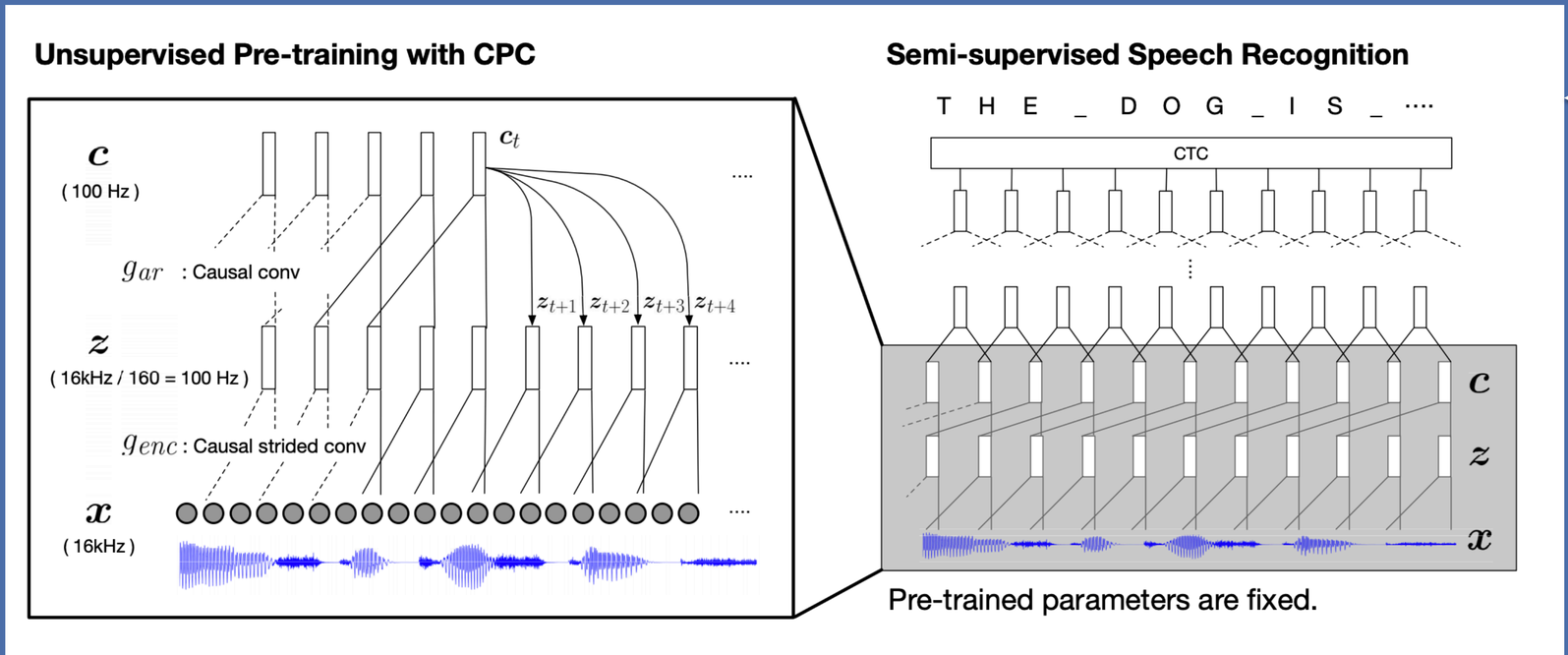


« **Representation Learning with Contrastive Predictive Coding** », Oord et al., 2018

Semi-supervised speech recognition

- After pre-training, train on a low-resource language with only few annotated data

ENGLISH →



← AMHARIC

« **Unsupervised Learning of Efficient and Robust Speech Representations** », Kawakami et al., 2019

Semi-supervised speech recognition

- Word Error Rate (%), the lower the better

MODEL	AMHARIC	FONGBE	SWAHILI	WOLOF
NO PRE-TRAINING	78.85	65.34	77.18	69.93
CPC (8K HOURS OF ENGLISH)	**66.10**	**57.20**	**69.23**	**55.41**

« **Unsupervised Learning of Efficient and Robust Speech Representations** », Kawakami et al., 2019

Self training

- Train a model for speech recognition on dataset A using true transcriptions
- Use this trained model to transcribe dataset B (pseudo-transcriptions)
- Retrain the model with both the true and pseudo-transcriptions
- Kind of a « magic tool » of deep learning, even though some theoretical undertanding emerges [1]

[1] « **Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data** », Wei et al., 2020

Self training

- Since the neural acoustic model outputs a distribution over characters, we can evaluate the confidence of the model by using the predicted probabilities (low entropy = highly confident, high entropy = low confidence)

	460 hours labelled	100 hours labelled	100 hours labelled + 360 hours unlabelled
WER (%)	4.23	8.06	5.79

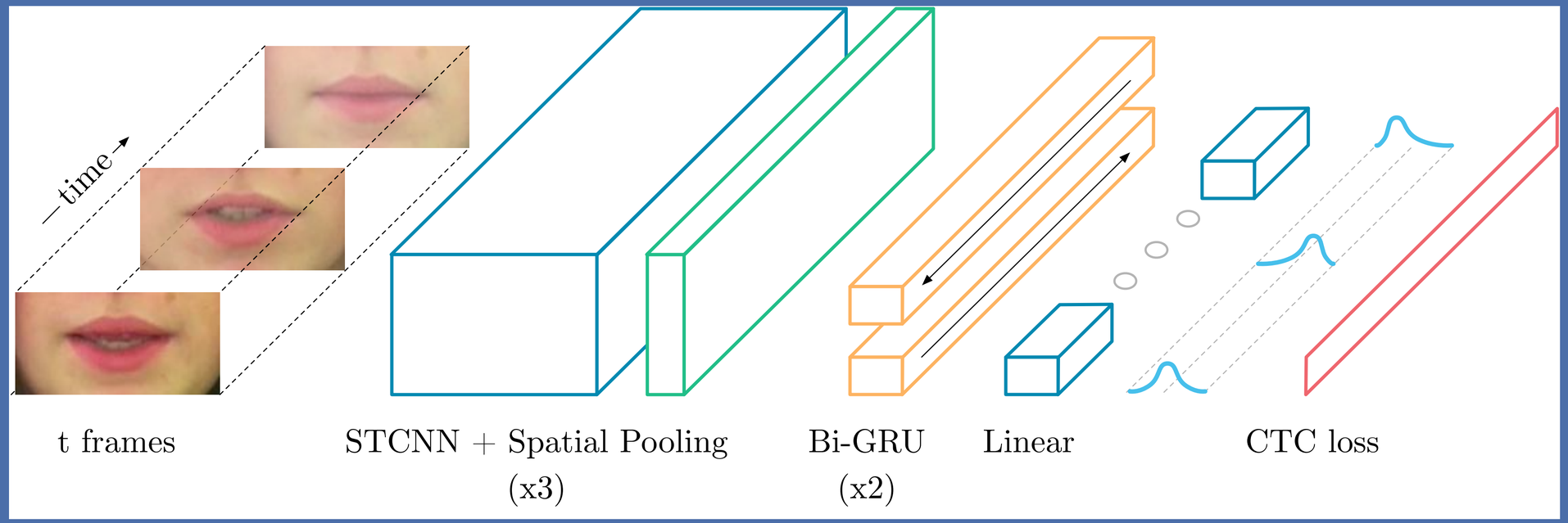
« **Self-Training for End-to-End Speech Recognition** », Kahn et al., 2020

End-to-end everything?

- End-to-end lip-reading
- End-to-end speech synthesis
- End-to-end speech separation

End-to-end lip-reading

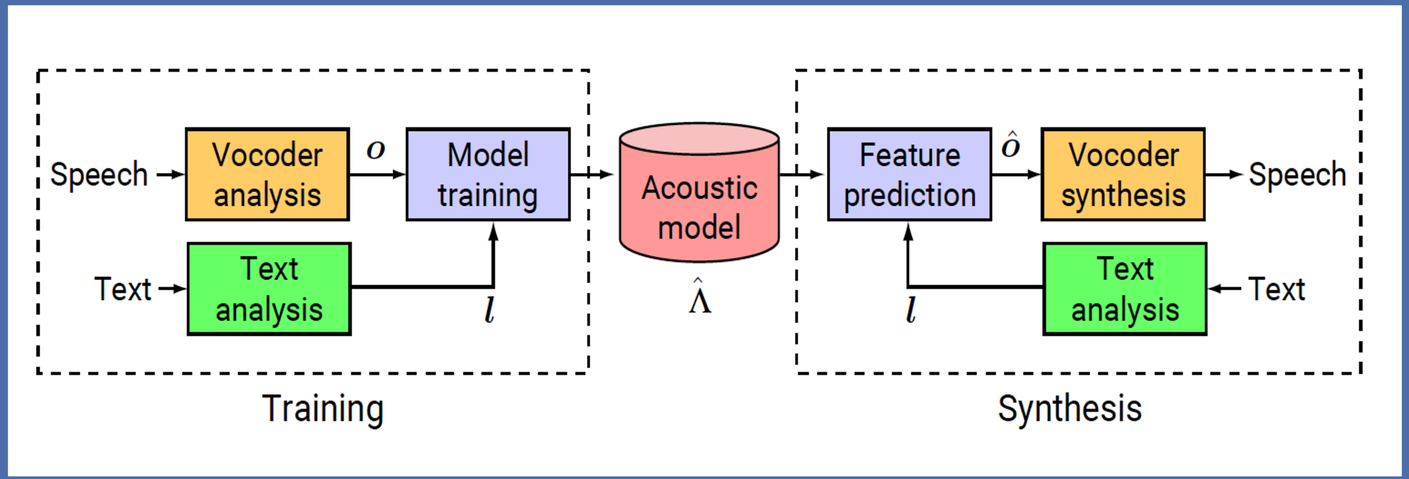
- An end-to-end speech recognition system takes frames of audio features as inputs and outputs a transcription
- Idea: Replace audio frames by photographs of the mouth and train with CTC
- 11.4% WER!



« **LipNet: End-To-End Sentence-Level Lipreading** », Assael et al., 2020

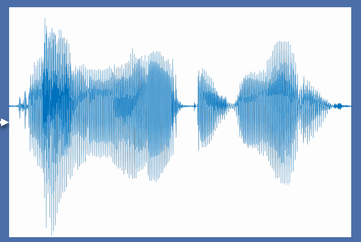
Speech synthesis

- Given a text sequence and information about a speaker, generate speech
- Traditionally not end-to-end (at all)
- Parametric synthesis: control a vocoder (synthesizer) with parameters (pitch, MFCCs, etc.)



- Concatenative speech synthesis: concatenate chunks of audio from a database

End-to-end speech synthesis

« **MY NAME IS NEIL** » → **NEURAL NETWORK** → 

- From a sequence of characters, we want to generate a sequence of waveform values

Wavenet: Speech synthesis as language modelling

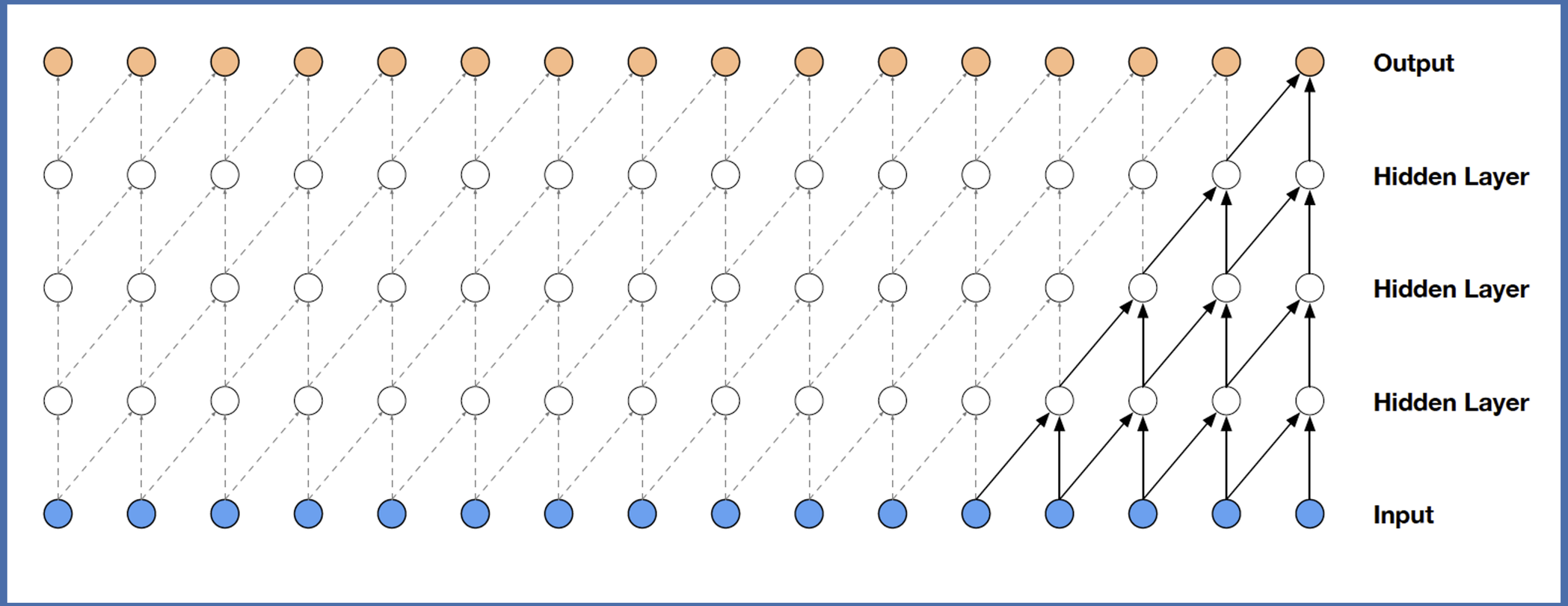
- Idea: autoregressive synthesis of waveform values from previous values

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

- Similar to a language model (predicts next word based on previous ones)
- Problem: a waveform is continuous (infinite number of possible values, one cannot compute the probability of x_t)
- In practice: waveforms are digital and typically on 16 bits = 65536 values
- Can we train a model to predict one of 65536 classes?
- Yes but too big! We can quantize these 65536 to 256 values with limited loss of quality

« **WaveNet: A Generative Model for Raw Audio** », van den Oord et al., 2016

Causal convolutions



« **WaveNet: A Generative Model for Raw Audio** », van den Oord et al., 2016

Dilated causal convolutions

- At 16kHz, using 1s of previous context requires a receptive field of 16000!
- Double the size of dilation at each layer (up to 512 then restart)
  to have a spread that grows exponentially with the number of layers
- Allows using large spreads efficiently (between 240 and 300 ms in experiments)

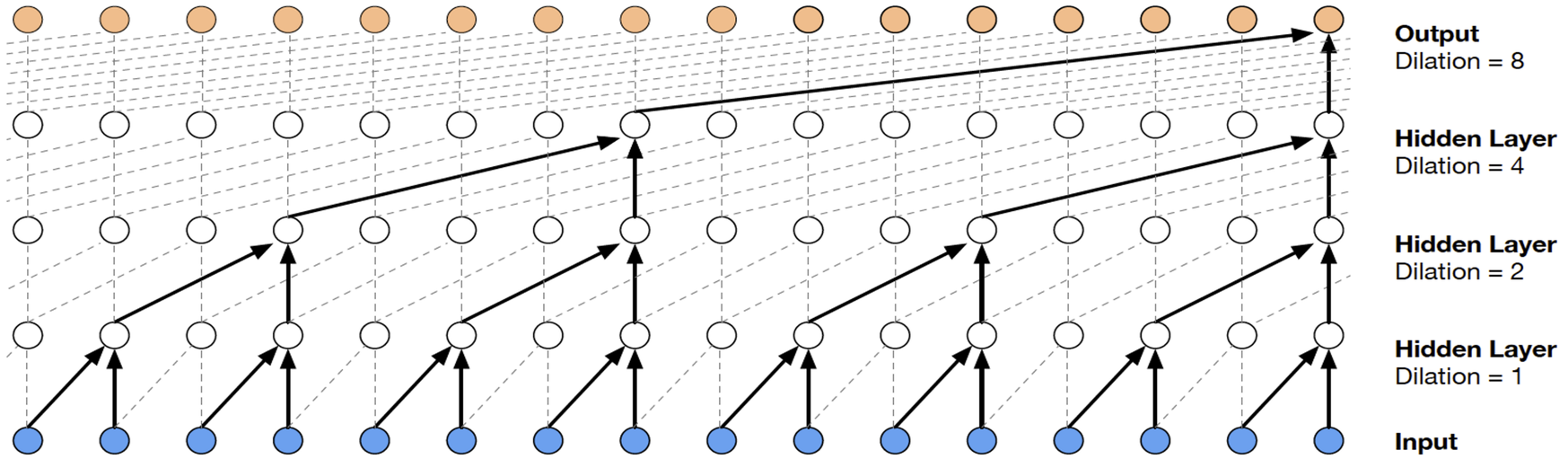


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Generating babbling

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

- If you train this model on a dataset of speech it just learns the distribution of speech and generates « babbling »: sounds like speech but is not real speech

Generating babbling

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

- If you train this model on a dataset of speech it just learns the distribution of speech and generates « babbling »: sounds like speech but is not real speech

Generating actual speech

- Condition the model on linguistic features upsampled (with a CNN) to have the same resolution as the waveform
- Phonetic, syllable, word, phrase, and utterance-level features (e.g. number of syllables in a word, phoneme duration, etc.)

Generating actual speech

- Condition the model on linguistic features upsampled (with a CNN) to have the same resolution as the waveform
- Phonetic, syllable, word, phrase, and utterance-level features (e.g. number of syllables in a word, phoneme duration, etc.)

Generating speech from a particular speaker

- We can also condition on other variables, including speaker identity, here represented as a one-hot encoding
- 44 hours of data for 109 speakers => ~24 minutes per speaker
- Cannot generalize to other speakers

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h})$$

Generating speech from a particular speaker

- We can also condition on other variables, including speaker identity, here represented as a one-hot encoding
- 44 hours of data for 109 speakers => ~24 minutes per speaker
- Cannot generalize to other speakers

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h})$$

Evaluating speech synthesis: Mean Opinion Score

- Unlike WER there is good enough computational metric for synthesis
- Needs to rely on human evaluations
- Mean Opinion Score (MOS): How natural does it sound to you (1 to 5)?

	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

Evaluating speech synthesis: Subjective comparison

- Among these two samples, which one do you prefer?

	Subjective preference (%) in naturalness					
Language	**LSTM**	**Concat**	**WaveNet** (L)	**WaveNet** (L+F)	No preference	p value
North American English	23.3	**63.6**			13.1	$\ll 10^{-9}$
	18.7		**69.3**		12.0	$\ll 10^{-9}$
	7.6			**82.0**	10.4	$\ll 10^{-9}$
		32.4	**41.2**		26.4	0.003
		20.1		**49.3**	30.6	$\ll 10^{-9}$
			17.8	**37.9**	44.3	$\ll 10^{-9}$
Mandarin Chinese	**50.6**	15.6			33.8	$\ll 10^{-9}$
	25.0		23.3		51.8	0.476
	12.5			**29.3**	58.2	$\ll 10^{-9}$
		17.6	**43.1**		39.3	$\ll 10^{-9}$
		7.6		**55.9**	36.5	$\ll 10^{-9}$
			10.0	**25.5**	64.5	$\ll 10^{-9}$

Limitations of WaveNet : linguistic features

- It’s end-to-end in a sense, but still needs linguistic features computed separately
- Tacotron generates speech directly from characters

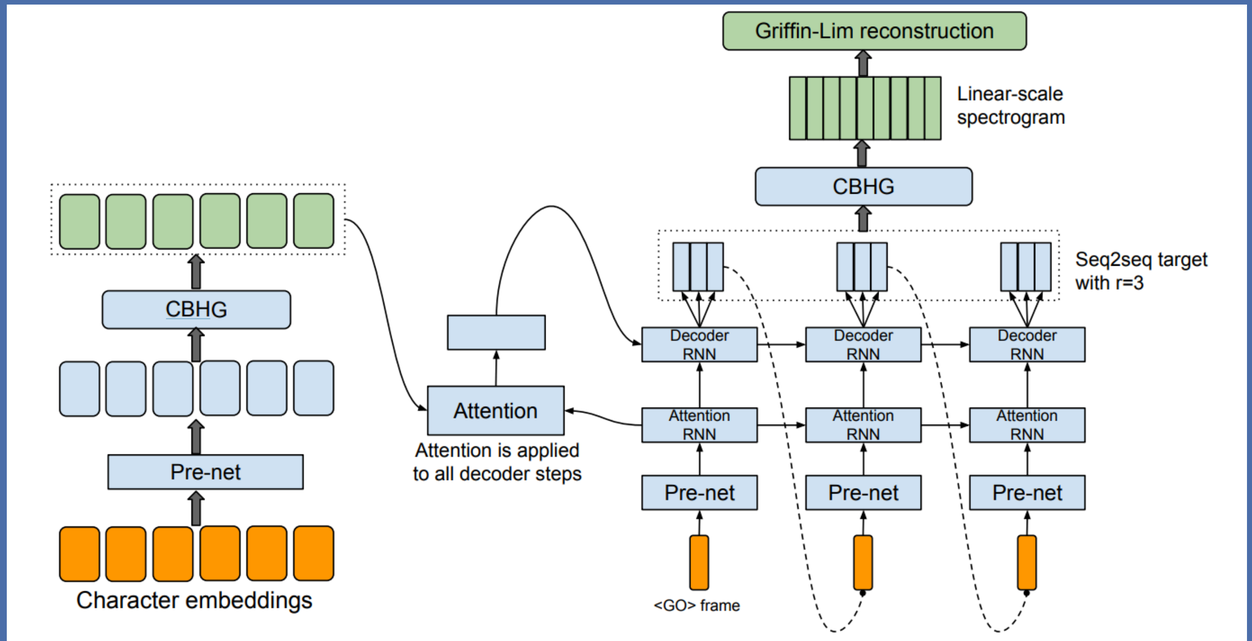


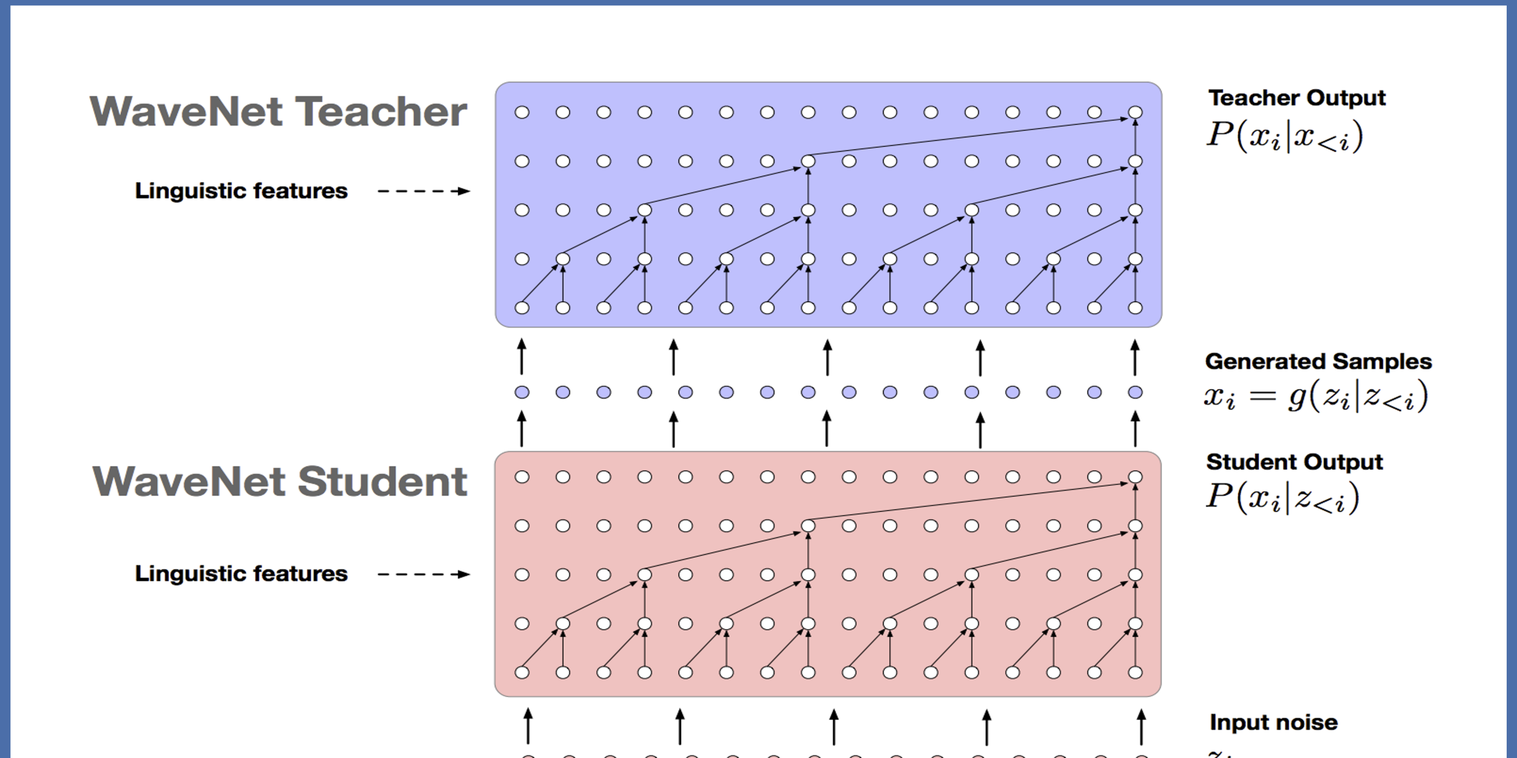
Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

Limitations of WaveNet : autoregressive synthesis

- Autoregressive synthesis is not parallelizable: generating x_t requires having generated previous steps
- Extremely slow (can take minutes to generate 1s of speech)

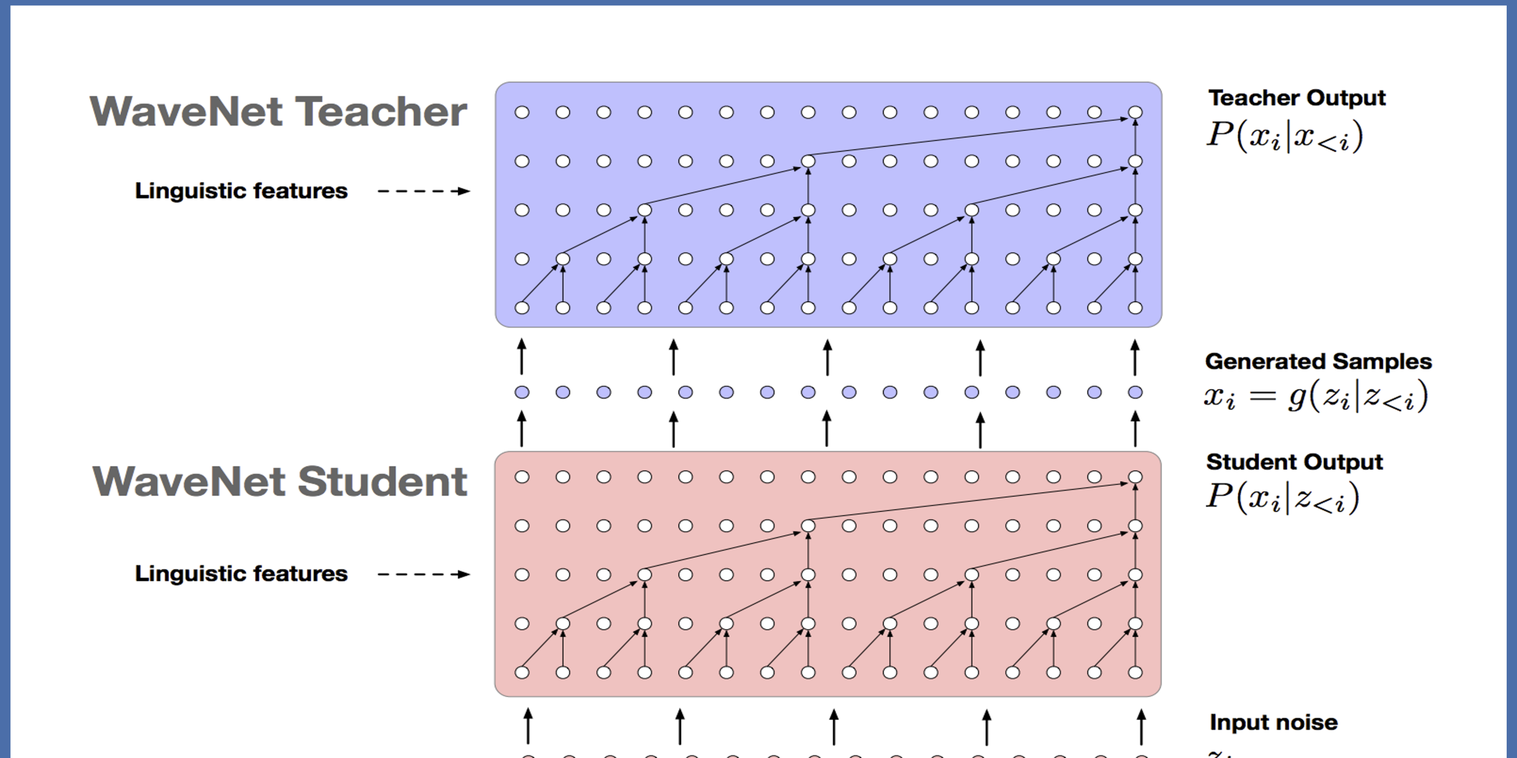
Parallel WaveNet

- Student trained to minimize the KL-divergence between its distribution and the teacher's
- The Student is not autoregressive!
- As good as the autoregressive, 1000x faster, deployed in Google Assistant



Parallel WaveNet

- Student trained to minimize the KL-divergence between its distribution and the teacher's
- The Student is not autoregressive!
- As good as the autoregressive, 1000x faster, deployed in Google Assistant



Voice technology « in the wild »

IDEAL SETTING

COMMON SETTING

- Voice technology (voice search, voice identification, youtube subtitling) works well in an ideal setting: one person speaking, no noise.
- Unfortunately, these conditions are often idealistic.

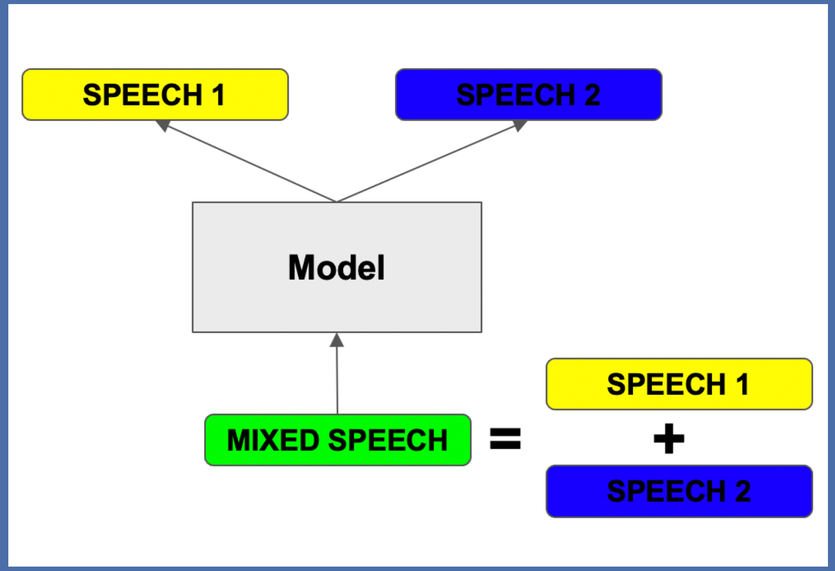
Speech separation

- Speech separation task:
- **inputs**: a single recording with several people speaking at the same time
- **outputs**: the voice of each speaker

Speech separation and the permutation problem

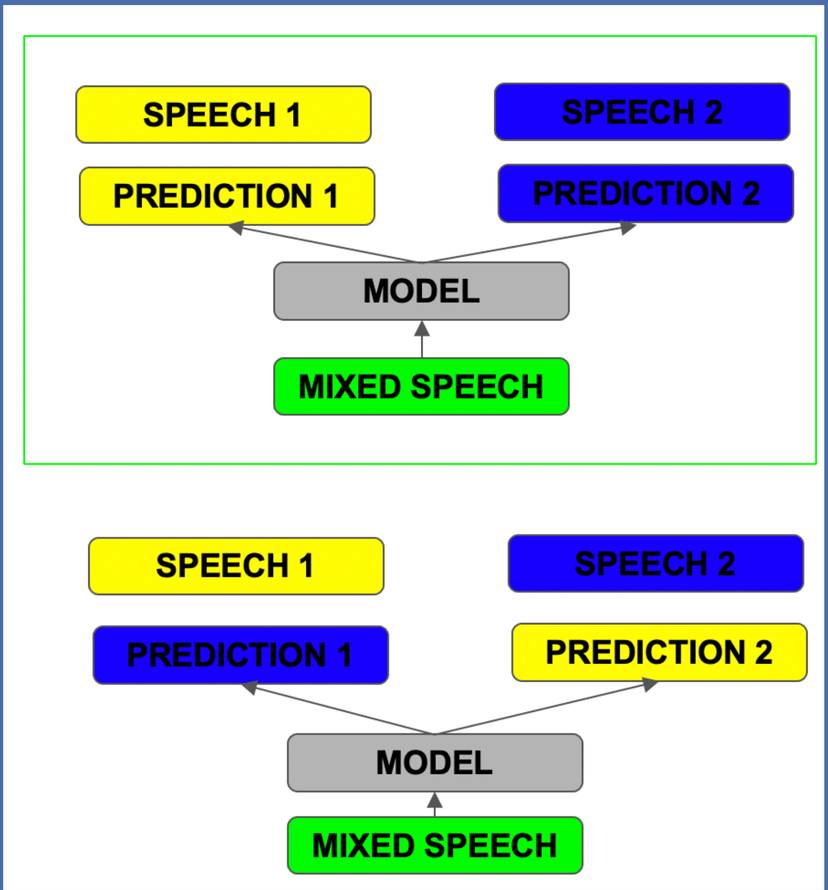
- **Task: given a mix of K unknown voices, extract the speech of each person**

- **Main challenge: Speaker permutation**
  - During training, the target speaker assigned to each channel is necessarily arbitrary (no obvious split e.g. female/male, british/us)
  - Result: channels are inconsistent
  - Main solution: **permutation invariant training**



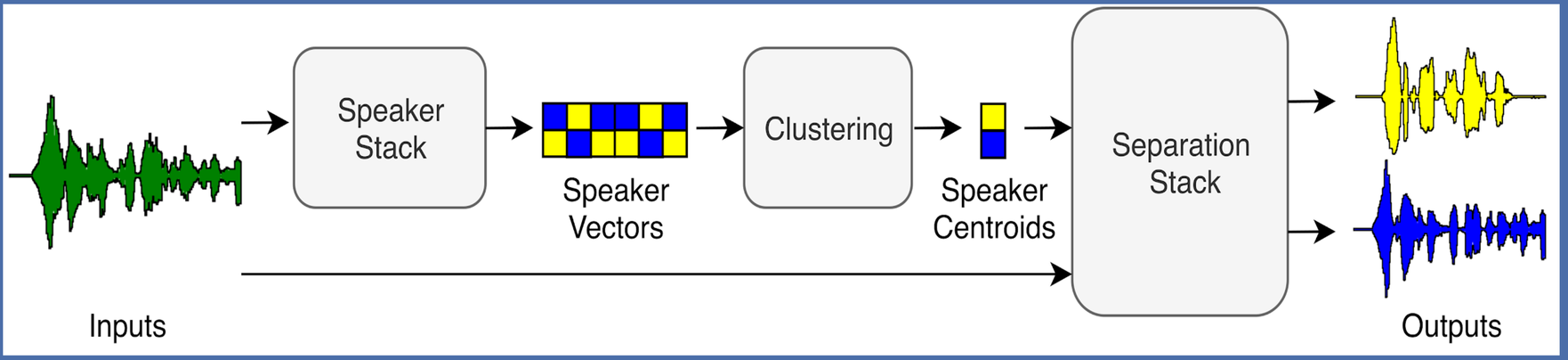
Permutation invariant training

- **Compute loss (MSE, NLL, etc.) over all permutations and backprop the minimum**
- **Problems:**
  - To learn to be consistent along a sentence, need to compute P-I loss along long windows (~4s) for one gradient step
  - Need to compute k! losses (k=nb speakers)
- **Solution?:**
  - Listen to the mixture and identify who is speaking
  - Then separate the speech of each speaker



Wavesplit: end-to-end speech separation by speaker clustering

- The speaker stack listens to mixed speech and extract N vectors per time step, each represents one speaker
- A clustering algorithm groups these vectors to have N vectors for the entire sequence
- The separation stack uses each vector to extract the speech of the corresponding speaker



« **Wavesplit: End-to-End Speech Separation by Speaker Clustering**», Zeghidour and Grangier, 2020

Dynamic mixing: generating new example mixtures on the fly

- To train a speech separation system, clean speech of N speakers are summed to create an artificial mixture (in standard datasets)
- Instead of relying on a fixed number of artificial mixtures, we generate an infinity from clean speech, by resampling randomly which sequence we mix and the volume of each speaker
- We can also randomly add background noise and reverberation to increase diversity of examples

« **Wavesplit: End-to-End Speech Separation by Speaker Clustering**», Zeghidour and Grangier, 2020

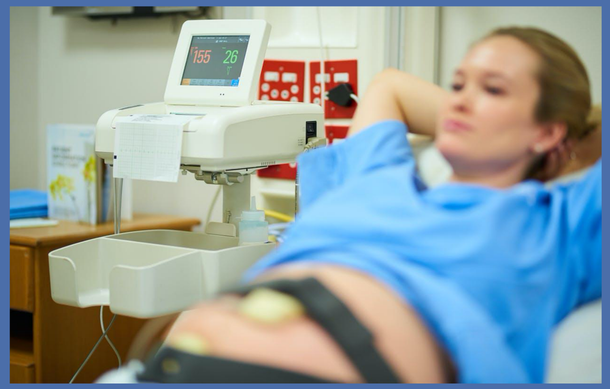
Results

- The main metric is Signal-to-Distortion Ratio, which measures the separation quality (the higher the better), in a logarithmic scale

MODEL	SDR (2 speakers)	SDR (3 speakers)	SDR (2 speakers w/ noise)
Previous state-of-the-art	20.3	16.9	12.4
Wavesplit	21.2	17.3	15.4
Wavesplit w/ Dynamic mixing	**22.3**	**17.8**	**16.0**

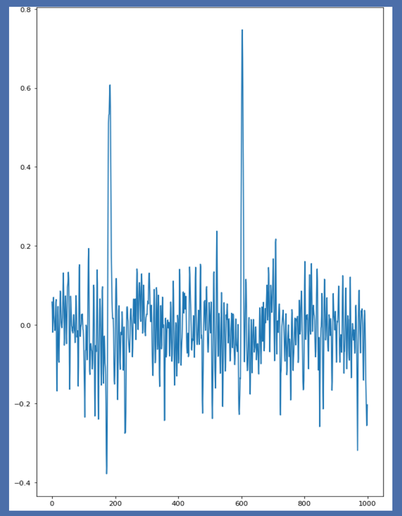
« **Wavesplit: End-to-End Speech Separation by Speaker Clustering**», Zeghidour and Grangier, 2020

Beyond speech: separating fœtal and maternal heart rate from electrocardiograms

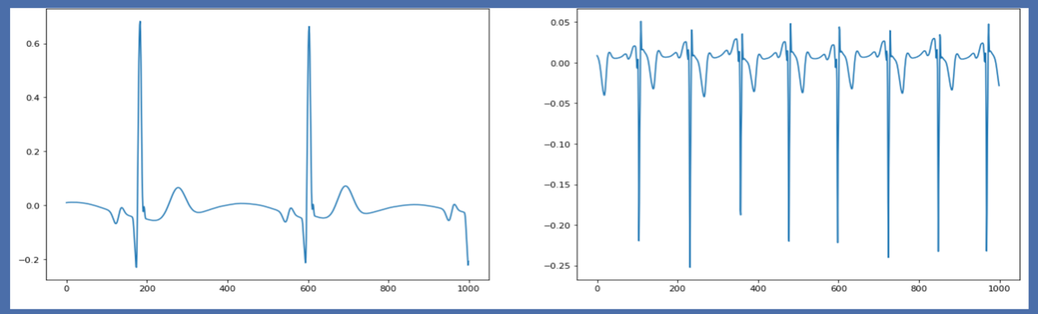


- Monitoring foetal health from its heart rate is common at various stages of pregnancy
- Hard task: the sensor gets the mother heart beats, the foetus', and noise
- **Wavesplit can separate both from a single sensor**

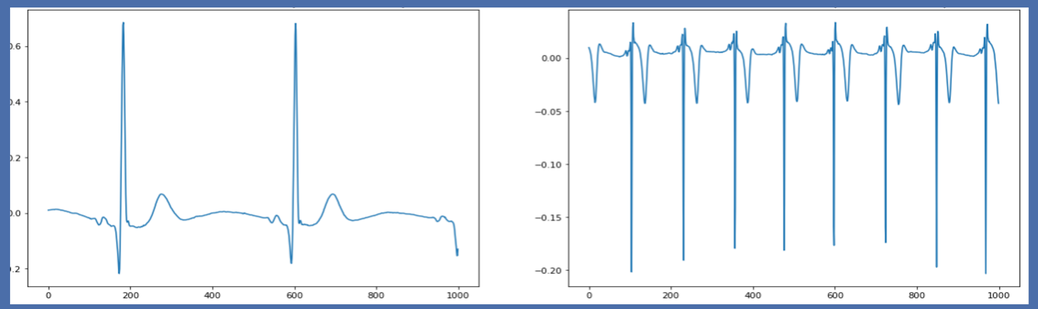
ABDOMINAL SENSOR



MOTHER **FOETUS**



REAL



SEPARATED WITH WAVESPLIT