

курс «Глубокое обучение»

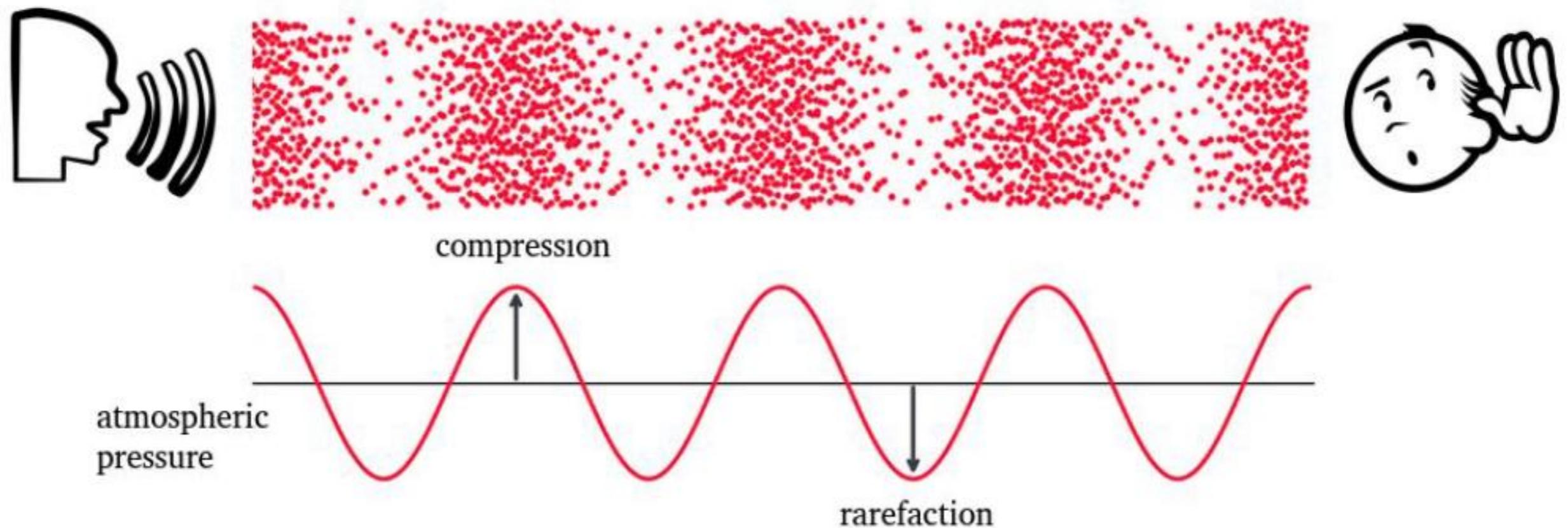
Распознавание речи

Александр Дьяконов

14 ноября 2022 года

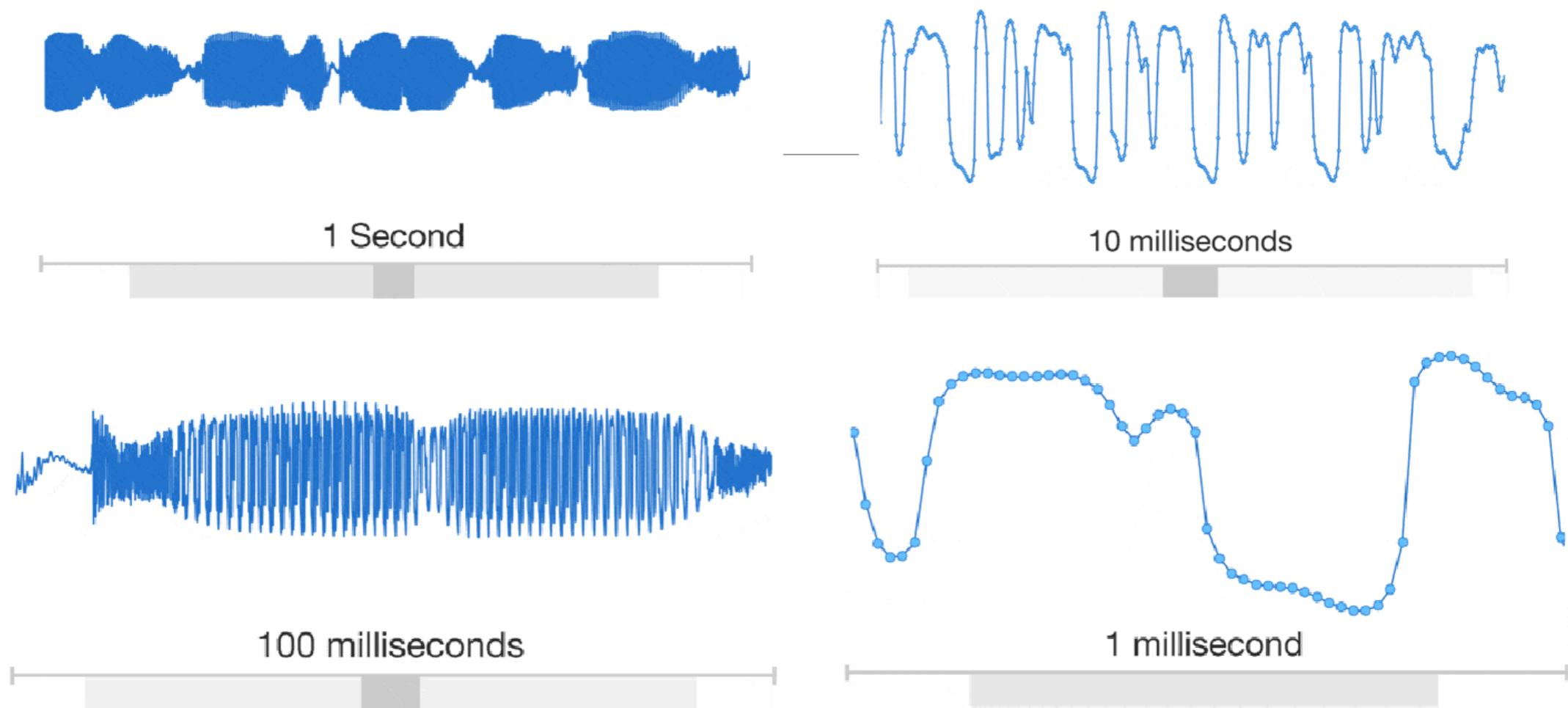
Что такое звук

измерение давления воздуха во времени



<https://github.com/musikalkemist/AudioSignalProcessingForML/>

Что такое звук – это волна



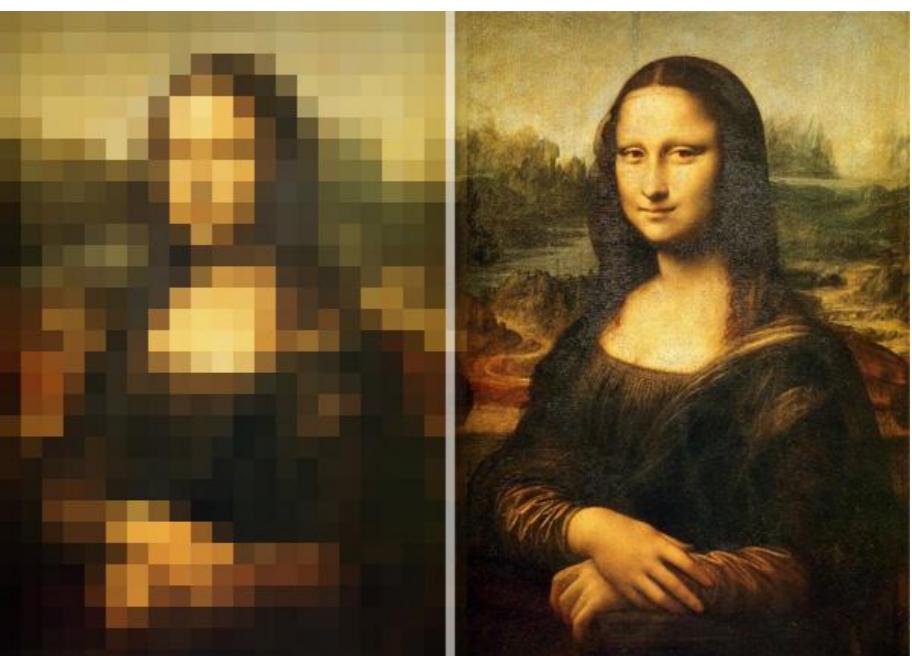
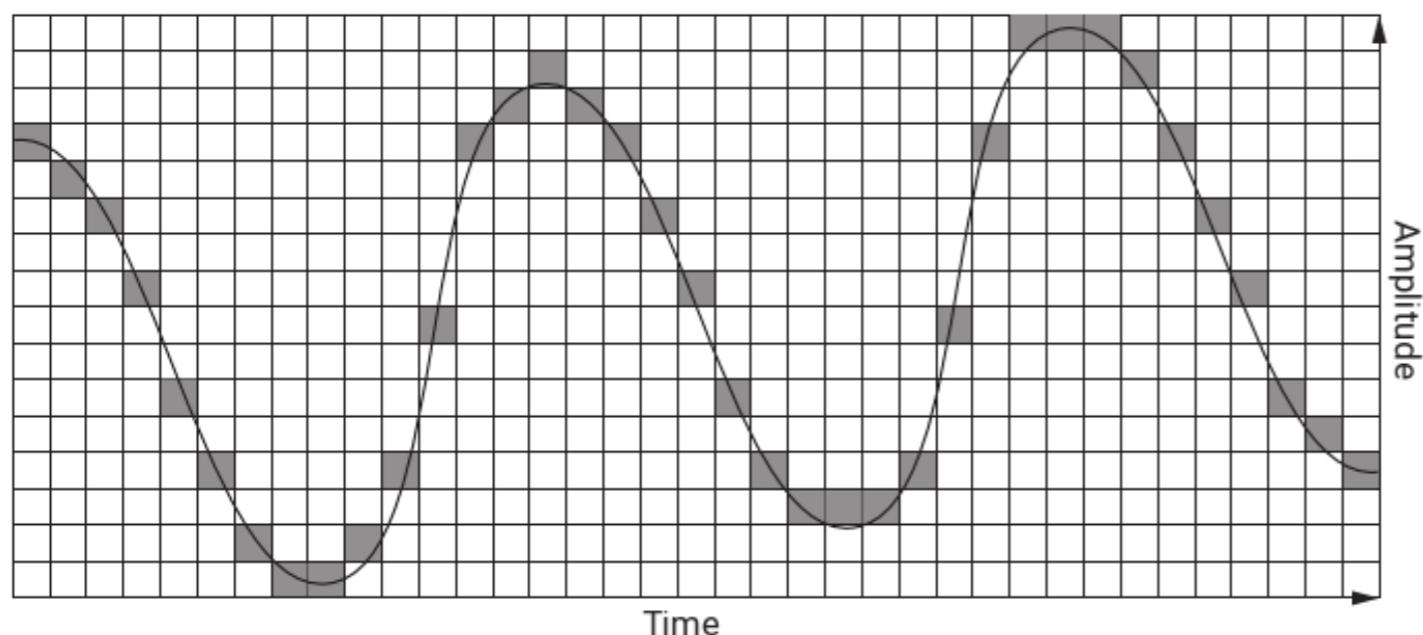
Tone Generator

<https://www.szynalski.com/tone-generator/>

<https://pudding.cool/2018/02/waveforms/>

Что такое звук

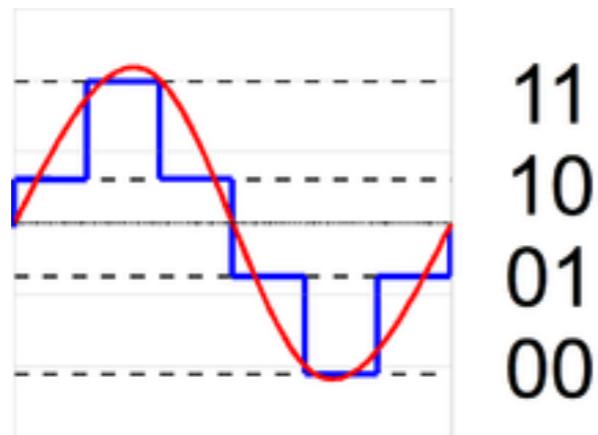
**измерение давления воздуха во времени
дискретизировано по времени ($1/T = 16\text{kHz}$, 8kHz для телефона, 44.1kHz CD)
дискретизировано по магнитуде (16bits, $2^{16} = 65536$)**



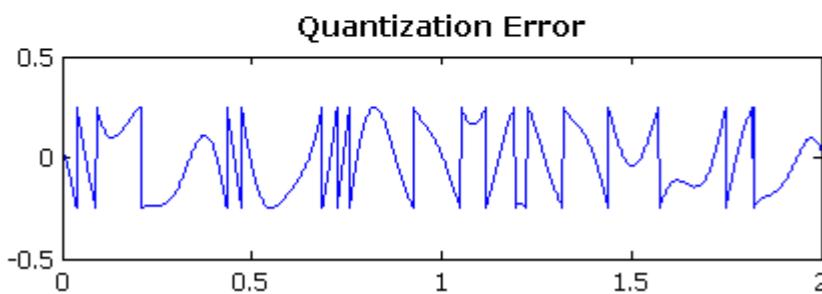
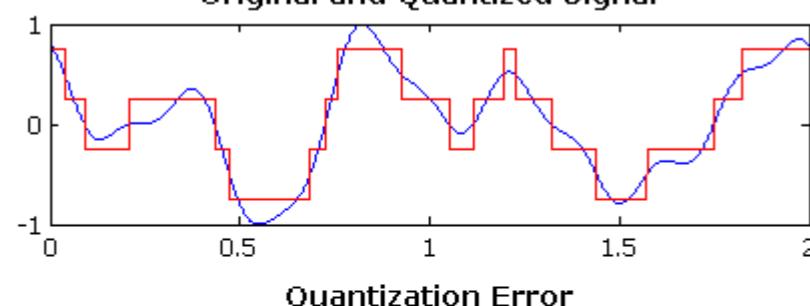
**сигнал – амплитуда / частота
его представление – квантование / частота**

Квантование (Quantization)

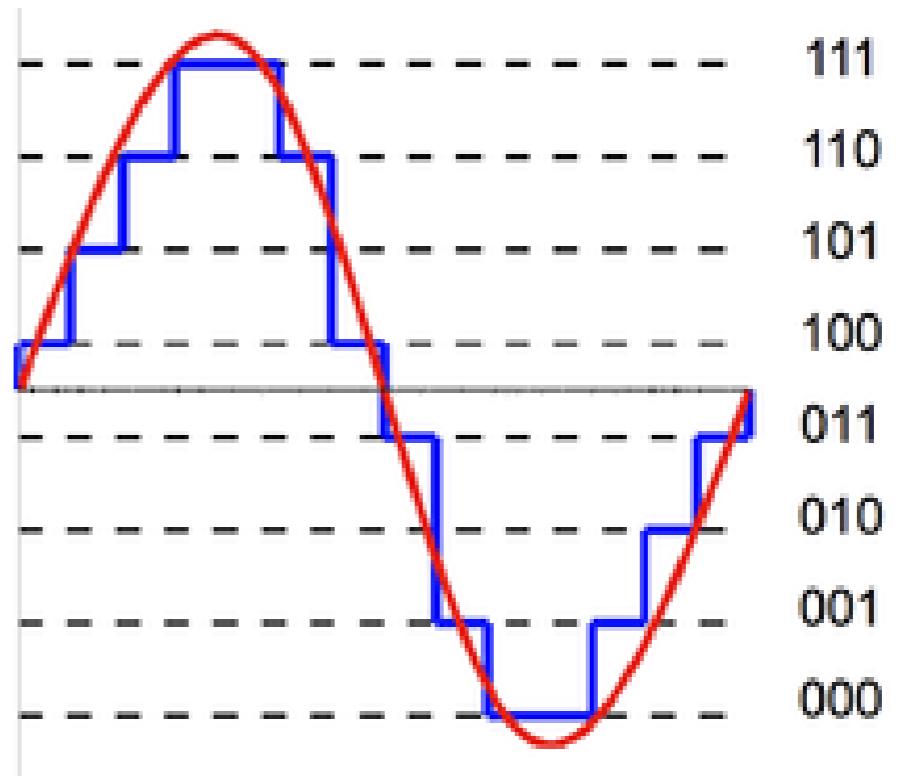
2 битное



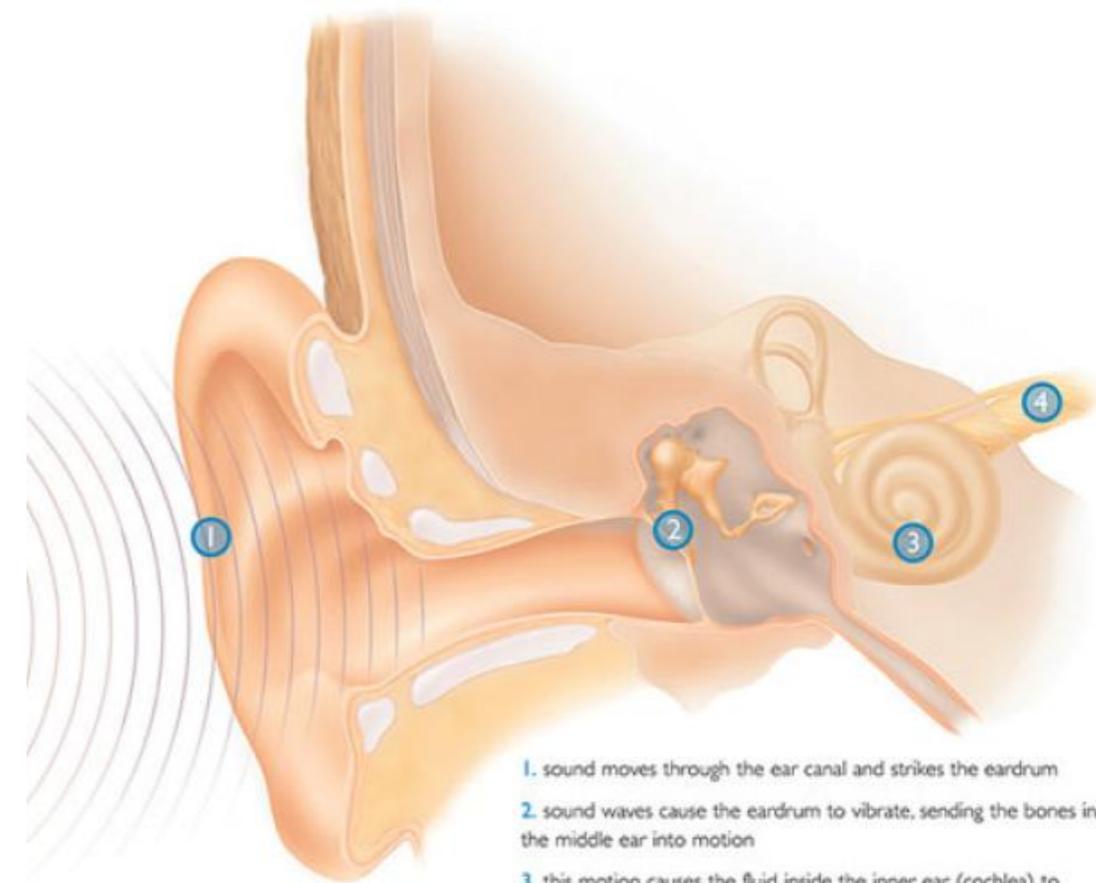
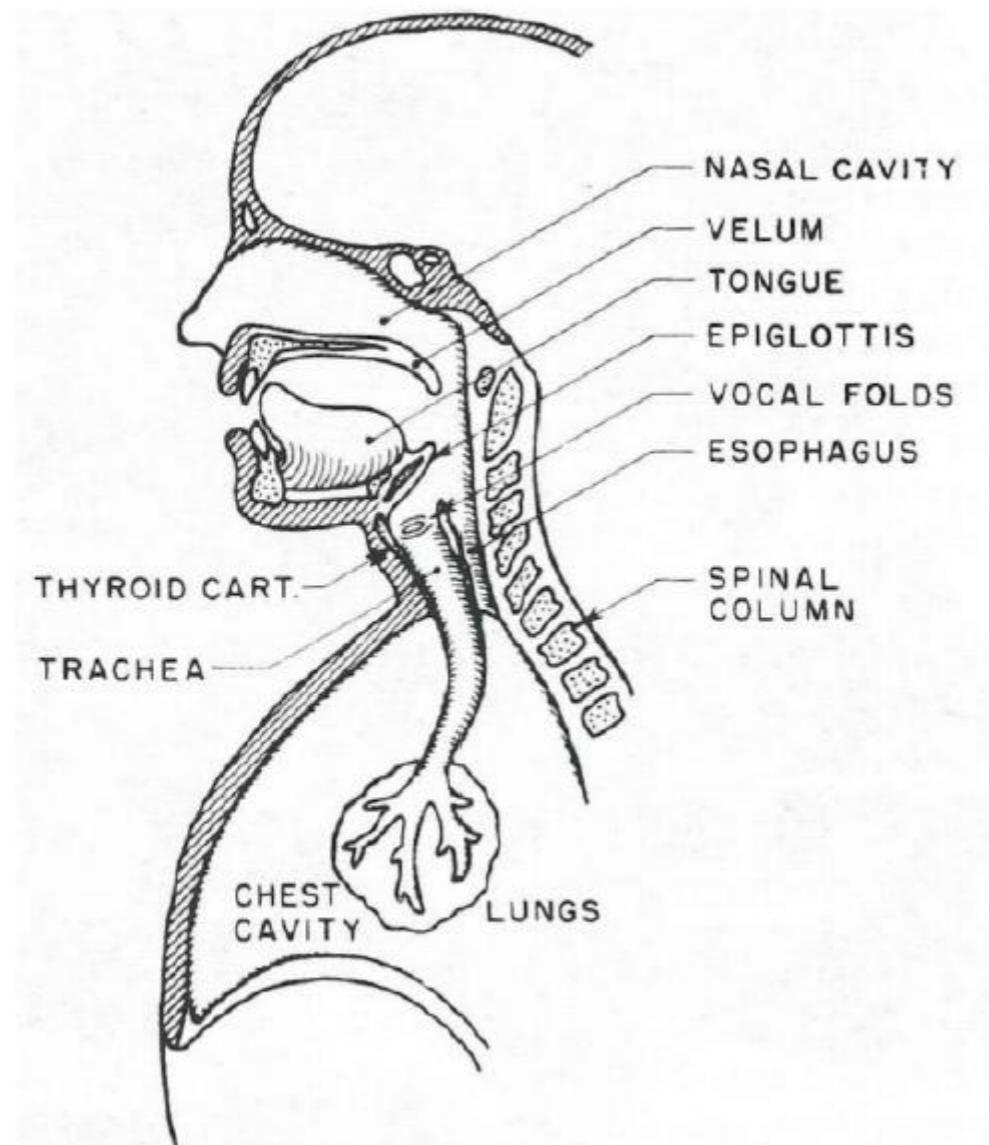
Original and Quantized Signal



3 битное

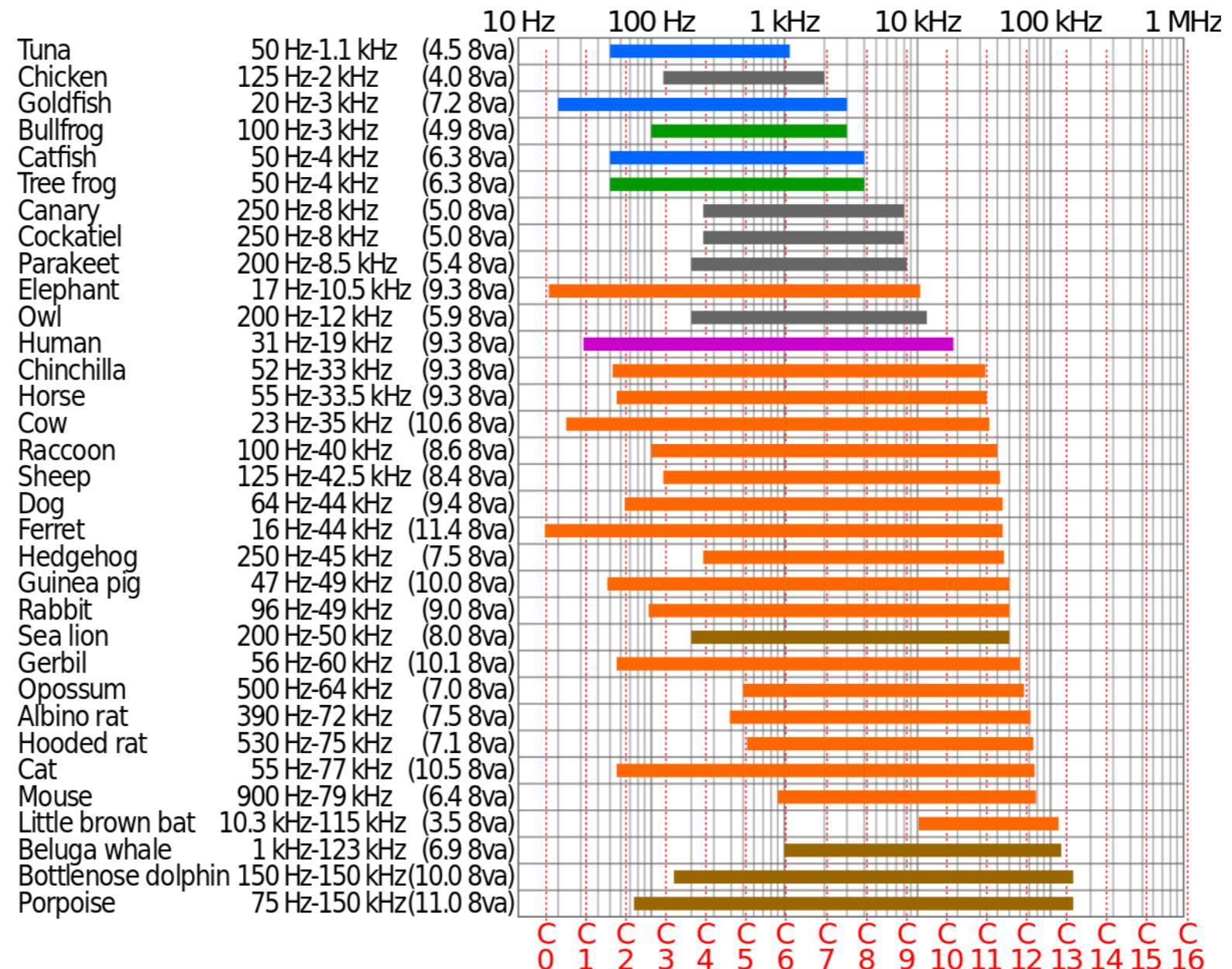


Звук / речь



1. sound moves through the ear canal and strikes the eardrum
2. sound waves cause the eardrum to vibrate, sending the bones in the middle ear into motion
3. this motion causes the fluid inside the inner ear (cochlea) to move the hair cells
4. hair cells change the movement into electric impulses, which are sent to the hearing nerve into the brain; you hear sound

Диапазон слышимости (Hearing range)



Откуда взялось 44100 kHz

**Человеческое ухо может слышать 20 kHz
По теореме Котельникова нужно 40 kHz**

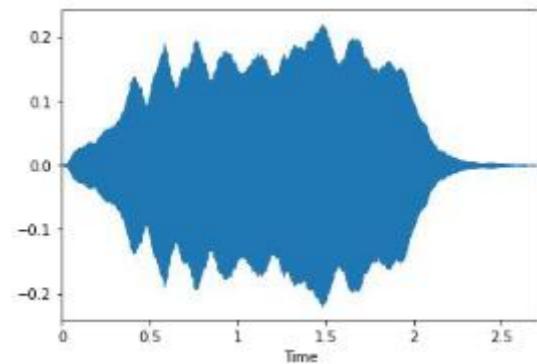
**Из-за специфики первых систем записи (использовали оборудования для видео):
NTSC 60 Гц, остаётся 245 строк, при записи 3 сэмплов в строку
дискретизации будет:
 $60 \times 245 \times 3 = 44100$**

**PAL 50 Гц =*=
 $50 \times 249 \times 3 = 44100$**

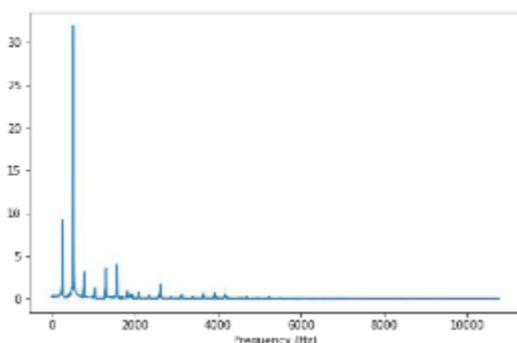
<https://www.masteringonline.ru/articles/44100/>

Описание сигнала

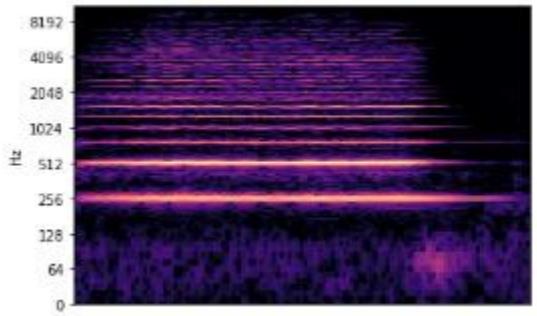
По времени



По частотам



По времени и частотам



Сложности в анализе звука... ... как у текста

- **входная информация – сигнал**
 - **одинаковый смысл объектов**
- **размер (size) «словаря» – число слов / фонем**
- **разнообразие спикеров (источников звука / акценты, диалекты)**
- **разнообразие среды (шум, фоновые звуки, способ записи)**
 - **«стиль» – монолог / диалог**
 - **разнообразие языков**

... в отличие от текста

- **сигнал вещественнозначный, а не дискретный**
- **сигнал непрерывен во времени (нет пауз между словами)**
 - **нет идеального выравнивания**
- **по сравнению с текстом – больше вариативности**
(от высоты, скорости речи, настроения, спикера, акцента, фона и т.п.)
- **требование к ресурсам (мобильные устройства, умные колонки и т.п.)**

Задачи со звуком

задача	вход	выход
Automatic speech recognition «speech-to-text»		«Включите свет»
Speaker/language identification Speaker diarization		Егор / русский
SLU Speech understanding		Свет выключается
Speech synthesis:	«Включите свет»	
Speech translation:		

speech enhancement, speech separation
keyword search, dialogue systems,
summarization, language instruction/assessment, voice morphing,
denoising, speaker separation, medical diagnosis...

Признаки в анализе звуков

сырые (Raw signal)

спектrogramma (Spectrogram)

Filter bank (гребёнки фильтров)

fMLLR

MFCC

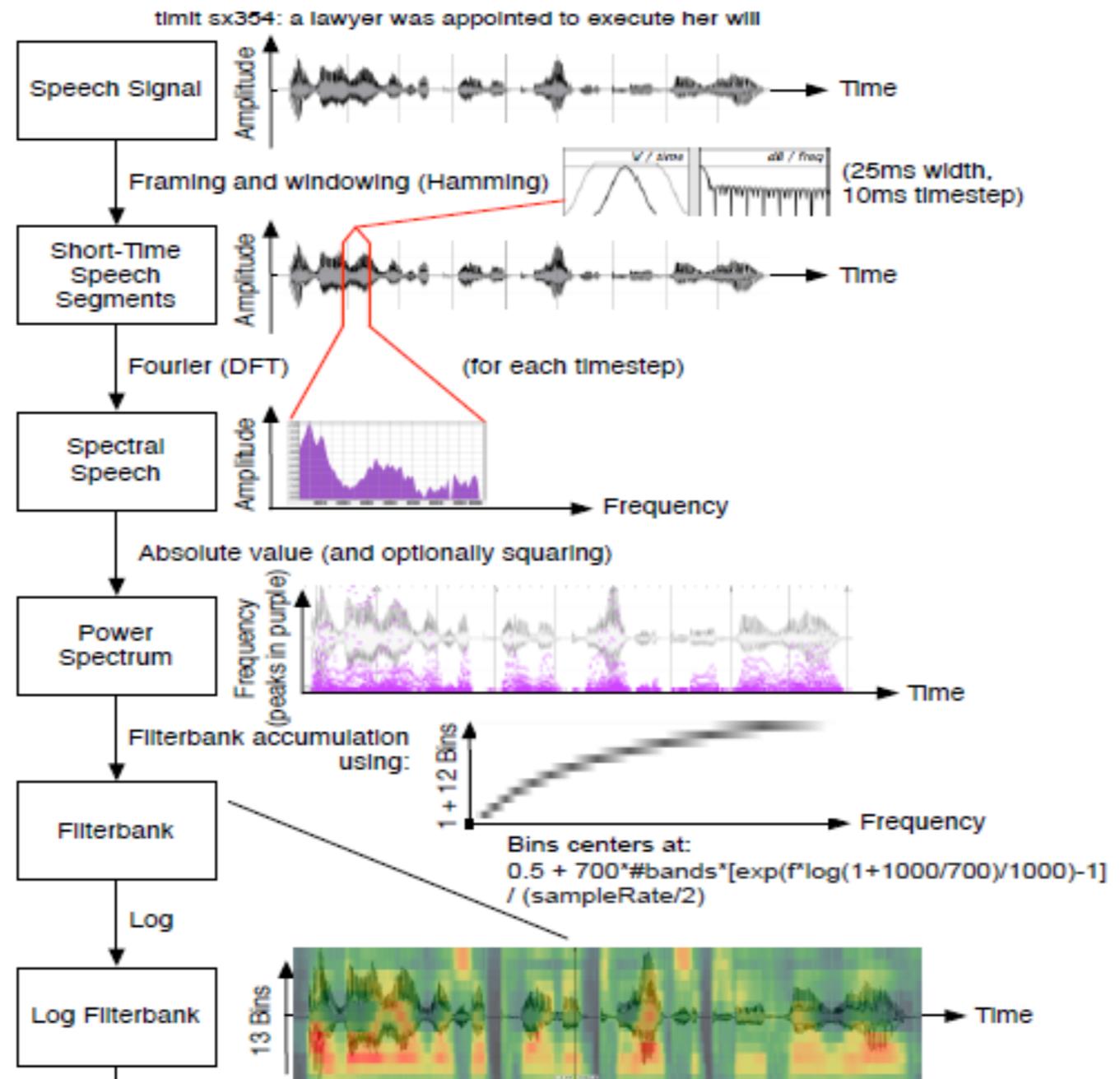
**чем больше предобработка,
тем на меньшем объёме данных работает!**

Извлечение признаков

- **librosa** <http://librosa.github.io/librosa/>
- **speechpy** <https://github.com/astorfi/speechpy>

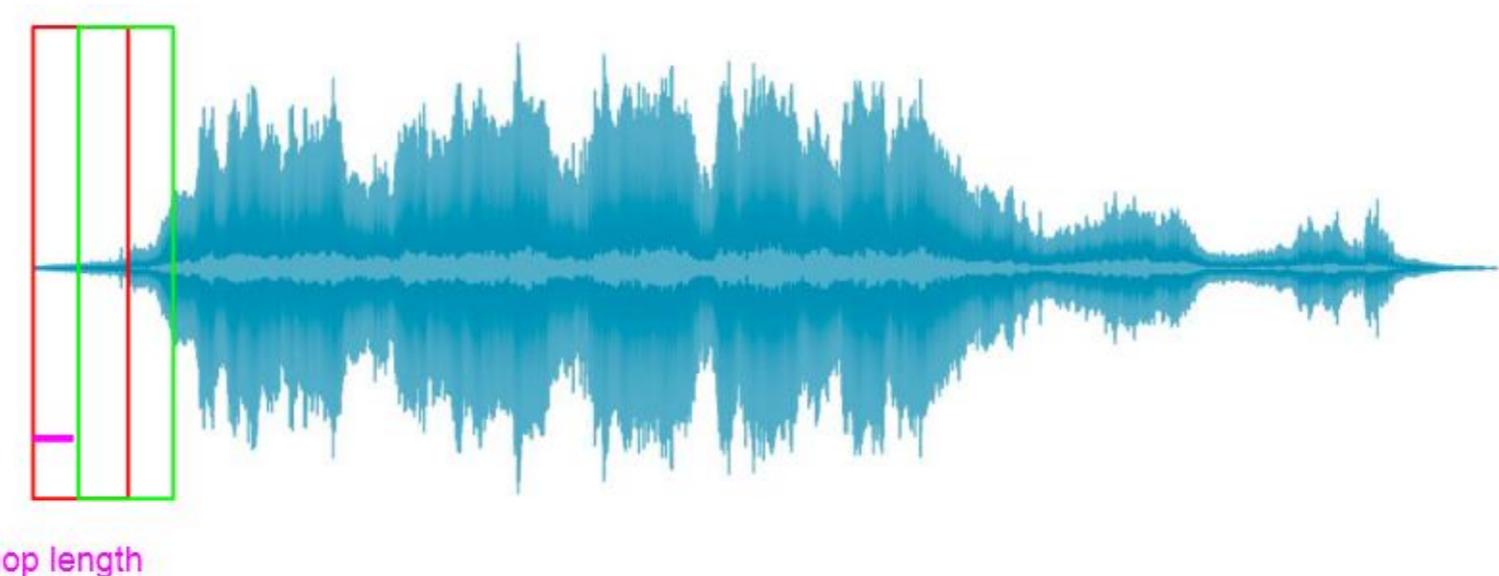
<https://www.youtube.com/watch?v=MhOdbtPhbLU>

Генерация признаков: MFCC



Нарезка на окна (Framing)

Используют перекрывающиеся окна (Overlapping frames)
нет потери информации,
используется функция окна (далше)



Feature vectors обычно вычисляются каждые 10мс, используя перекрывающиеся окна длиной 25мс (20–40). Для 16kHz $0.025 * 16000 = 400$ замеров

<https://github.com/musikalkemist/AudioSignalProcessingForML/>

Windowing

Умножение сигнала (в окне) на функцию окна:

$$w_n = (1 - \alpha) - \alpha \cos\left(\frac{2\pi n}{L-1}\right)$$

$$x'_n = x_n \cdot w_n$$

Hamming $\alpha \approx 0.46$

Hanning $\alpha = 0.5$

L – длина окна

Это ещё и борьба с «Spectral leakage» – появление лишних частот в спектре

Простейшие «старые» признаки

– статистики в фреймах

Amplitude envelope ~ max элементы

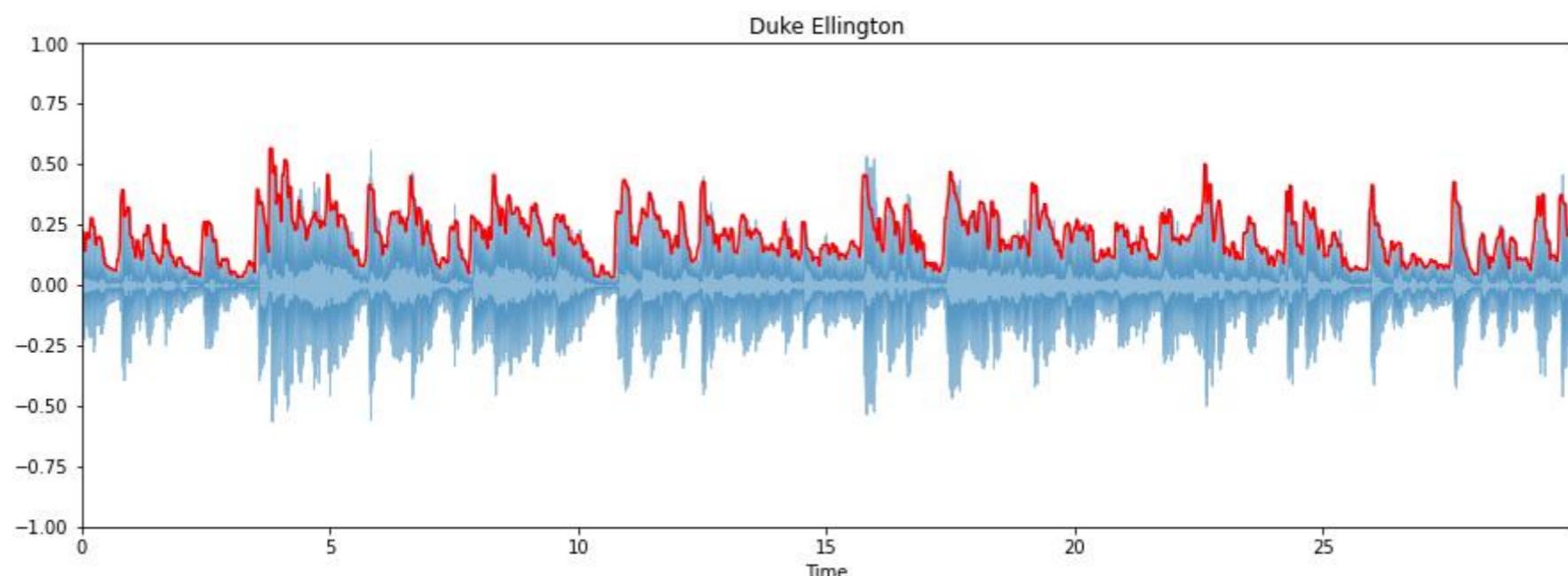
Root-mean-square energy

$$RMS_t = \sqrt{\frac{1}{K} \cdot \sum_{k=t \cdot K}^{(t+1) \cdot K - 1} s(k)^2}$$

Zero crossing rate

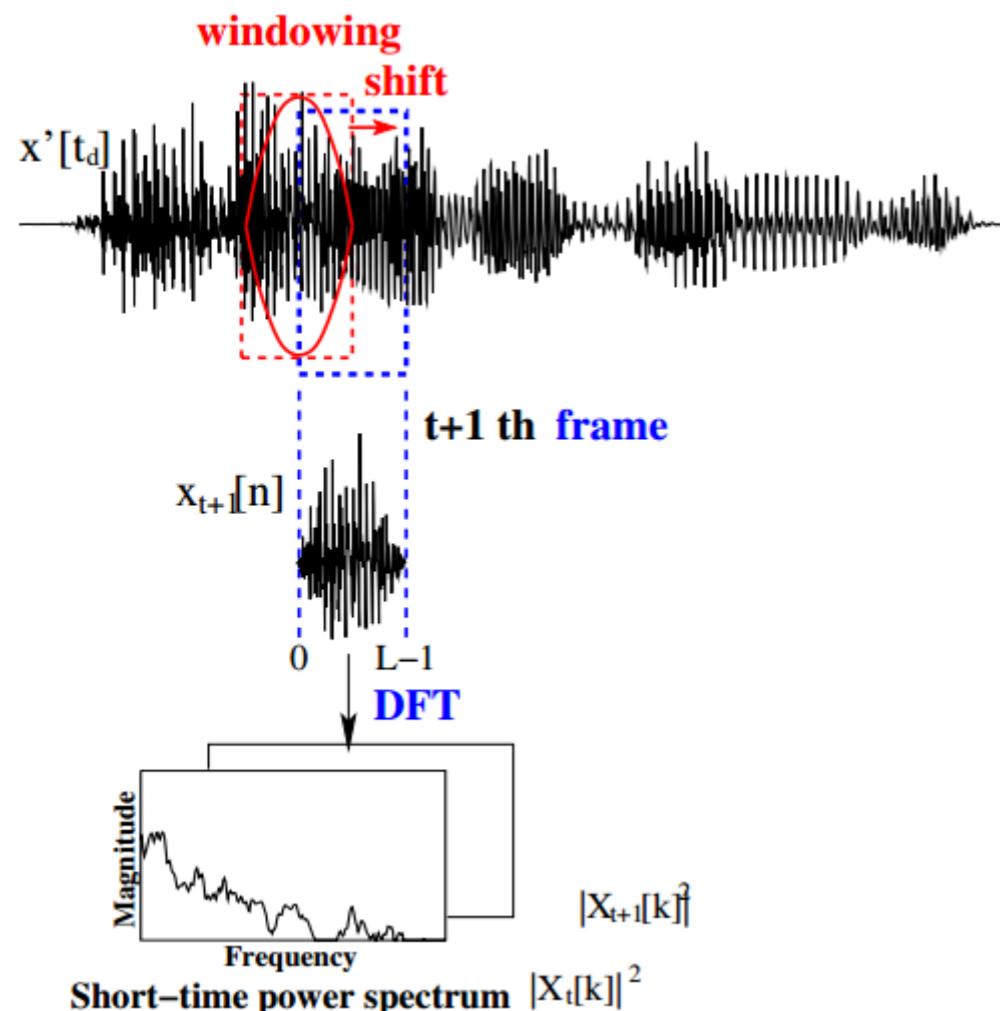
Простейшие «старые» признаки

Amplitude envelope
форма сырого сигнала



<https://github.com/musikalkemist/AudioSignalProcessingForML/blob/master/8-%20Implementing%20the%20amplitude%20envelope/Implementing%20the%20amplitude%20envelope.ipynb>

Локальное преобразование Фурье



После выделения окна и
умножения на функцию окна...

**ДПФ извлекает спектральную информацию из окна –
сколько энергии на какой частоте**

Дискретное преобразование Фурье
Discrete Fourier Transform (DFT)

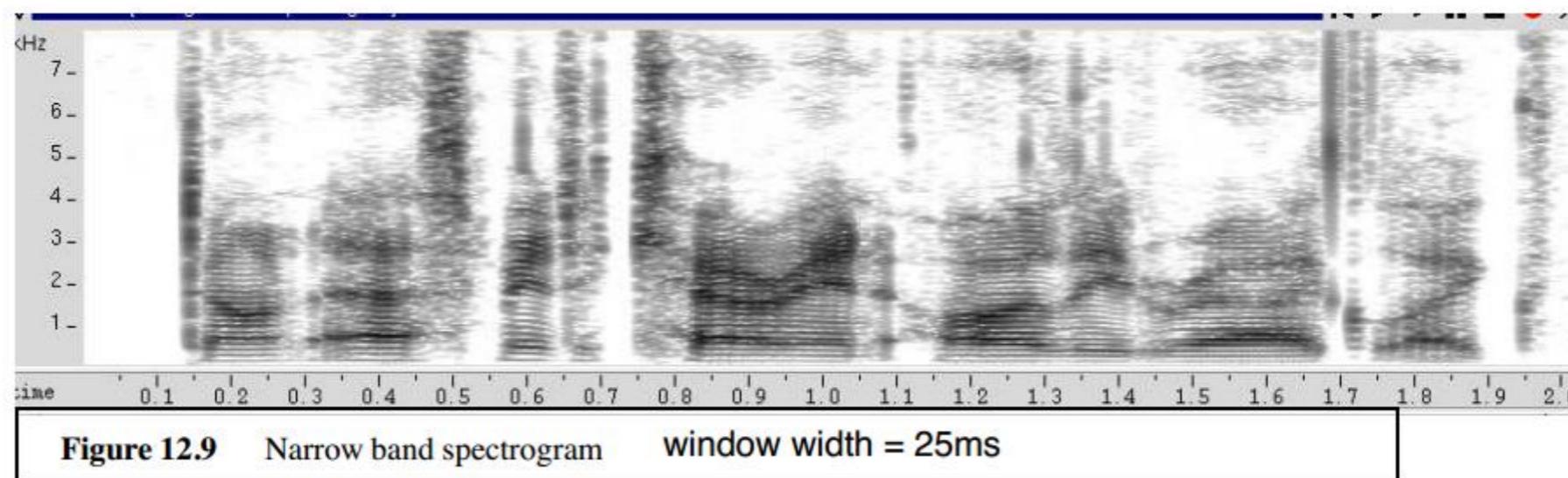
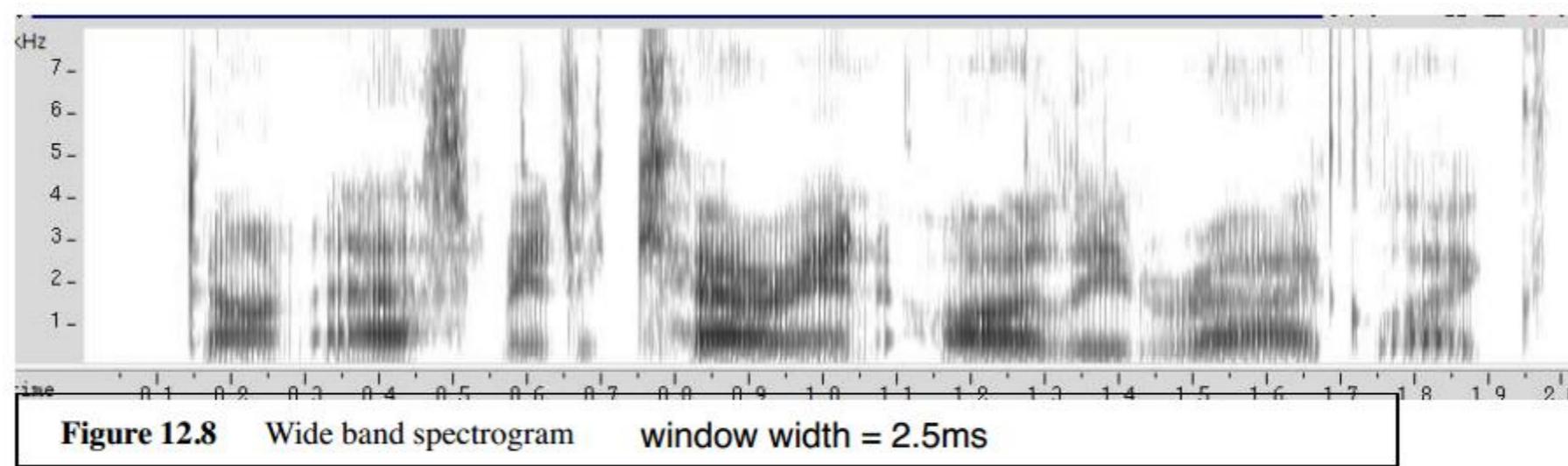
$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(-i \frac{2\pi}{N} kn\right)$$

**Быстрое преобразование Фурье (FFT) – эффективный
алгоритм для случай**

$$N = 2^{\lceil \log L \rceil}$$

В результате комплексное число (модуль и фаза)

Локальное преобразование Фурье



Discrete Fourier Transform (DFT)

$$X(n) = \sum_{k=0}^{N-1} x(k) \exp\left(-j \frac{2\pi n k}{N}\right)$$

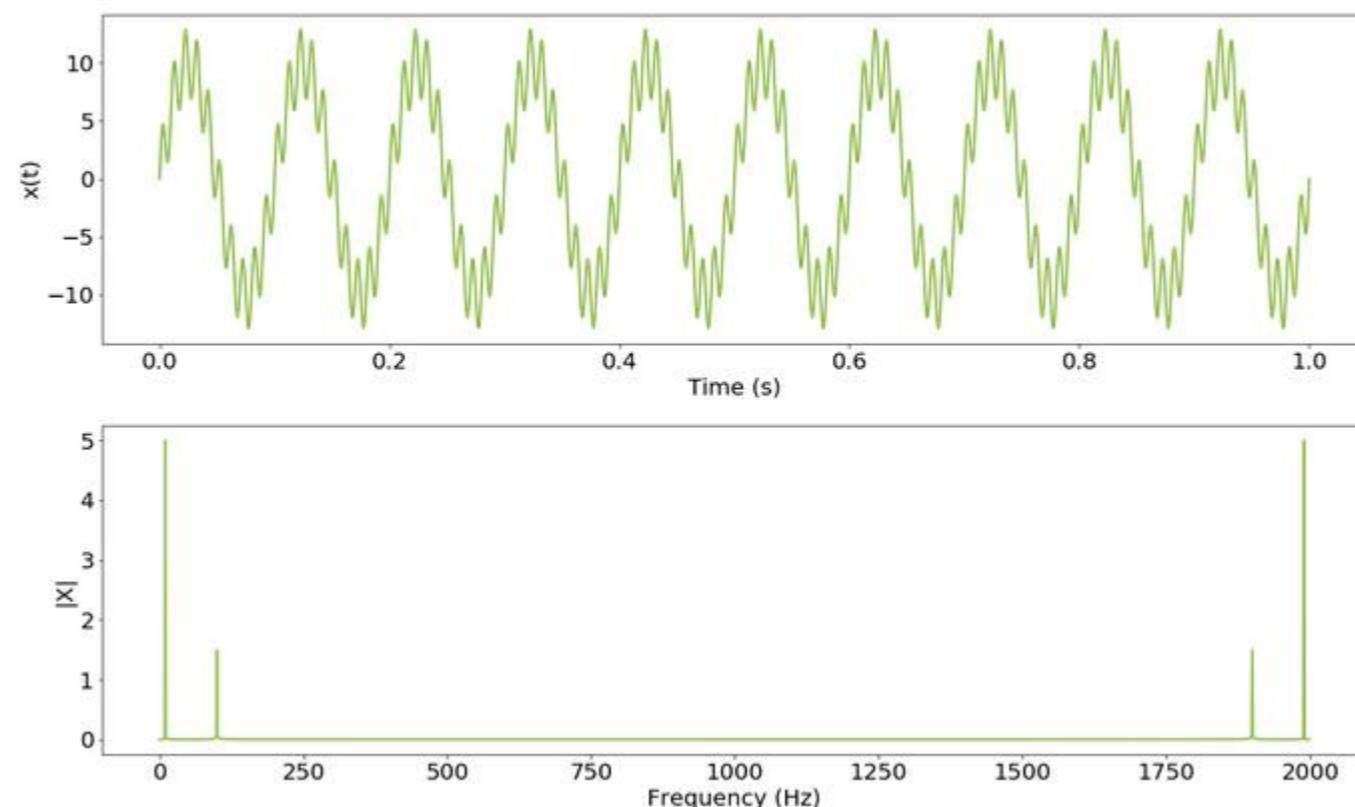
$$X = \mathbf{M}x$$

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-\frac{2\pi i}{N}} & e^{-\frac{4\pi i}{N}} & e^{-\frac{6\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}(N-1)} \\ 1 & e^{-\frac{4\pi i}{N}} & e^{-\frac{8\pi i}{N}} & e^{-\frac{12\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}2(N-1)} \\ 1 & e^{-\frac{6\pi i}{N}} & e^{-\frac{12\pi i}{N}} & e^{-\frac{18\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi i}{N}(N-1)} & e^{-\frac{2\pi i}{N}2(N-1)} & e^{-\frac{2\pi i}{N}3(N-1)} & \dots & e^{-\frac{2\pi i}{N}(N-1)^2} \end{pmatrix}$$

<https://github.com/markovka17/dla/tree/master/week01>

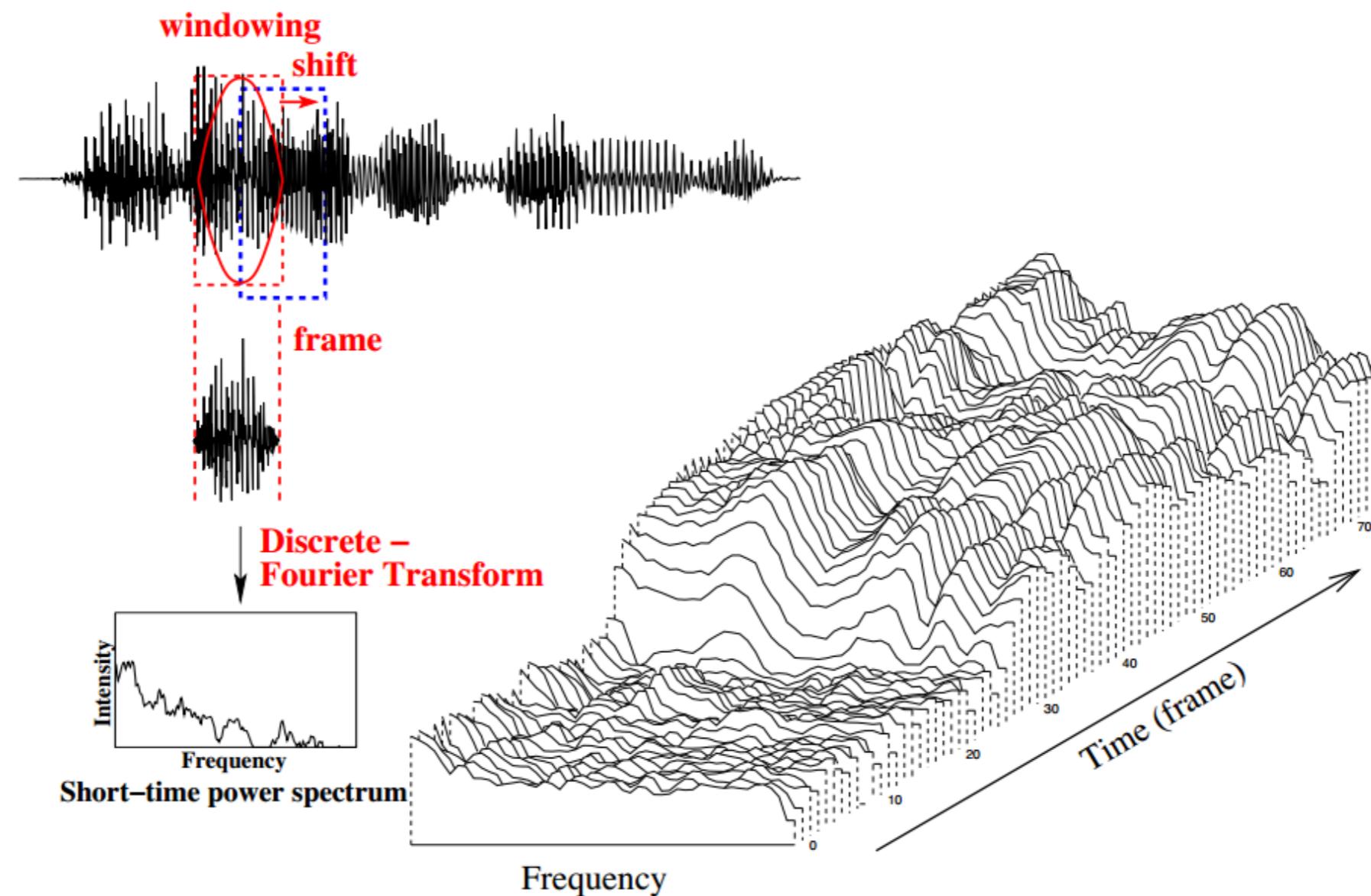
Discrete Fourier Transform (DFT)

$$f(t) = 10 \sin(2\pi 10t) + 3 \sin(2\pi 100t)$$



<https://github.com/markovka17/dla/tree/master/week01>

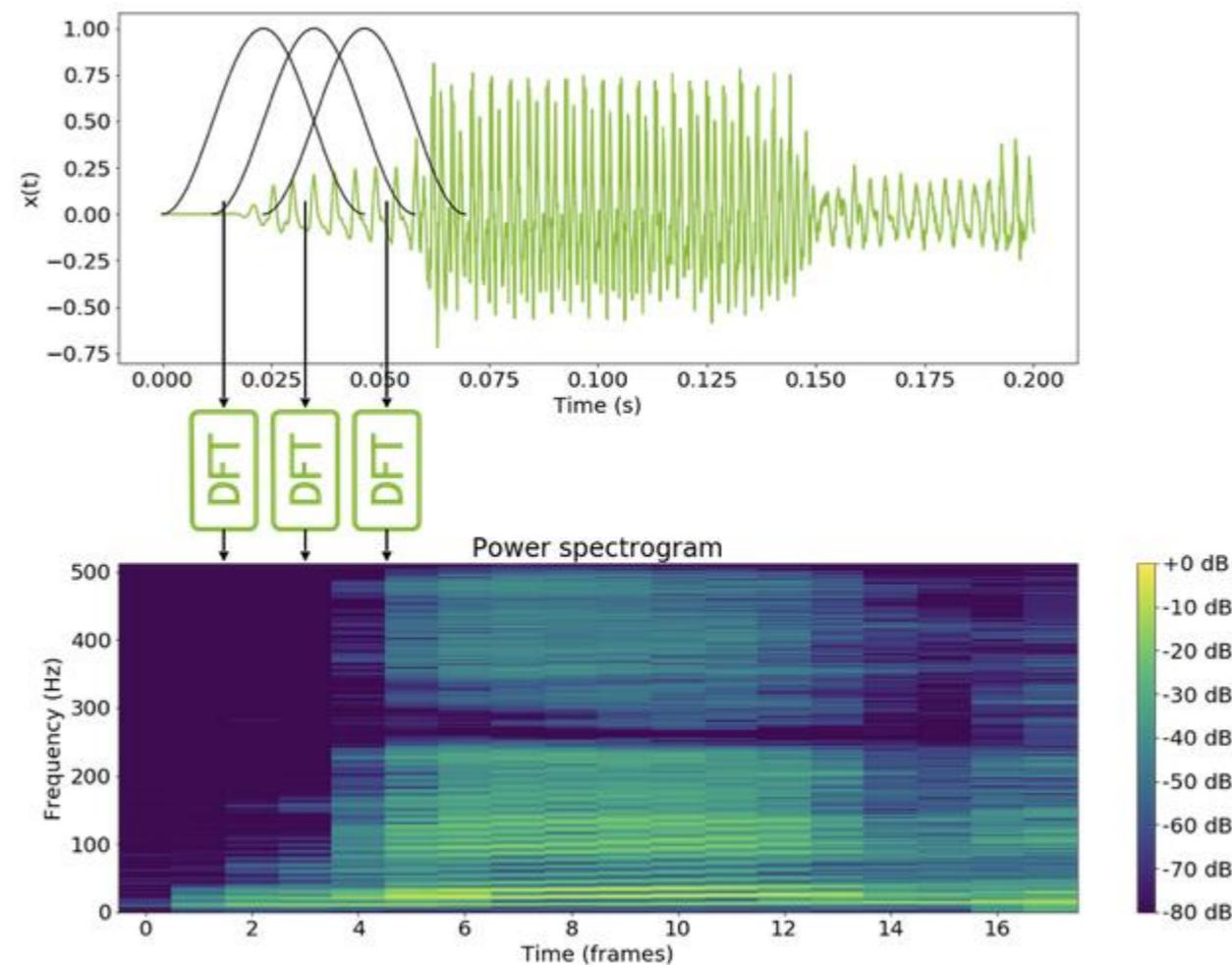
Локальное преобразование Фурье (Short-time Fourier transform)



так делается для каждого окна со сдвигом

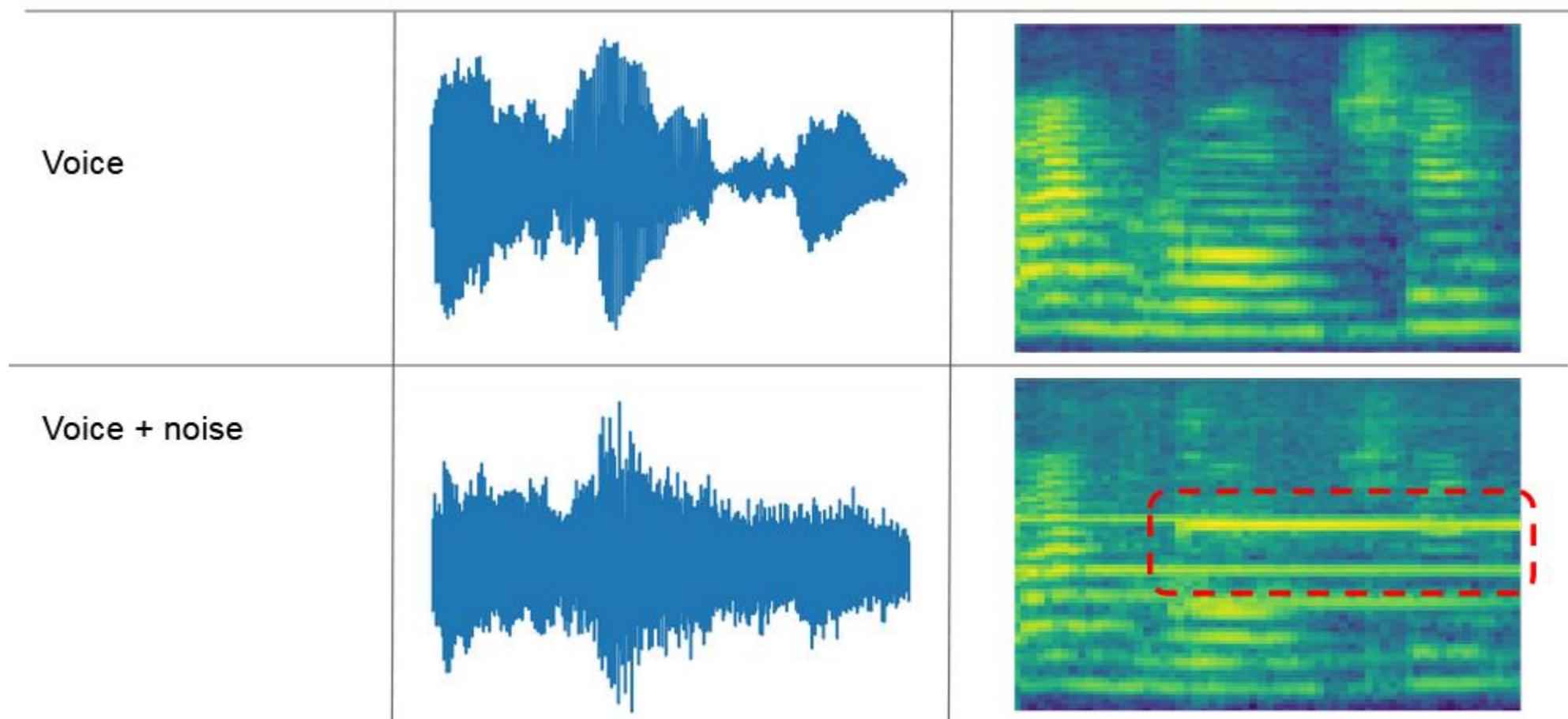
Локальное преобразование Фурье (Short-time Fourier transform)

= FFT + Windowing



<https://github.com/markovka17/dla/tree/master/week01>

Что видно на спектрограммах



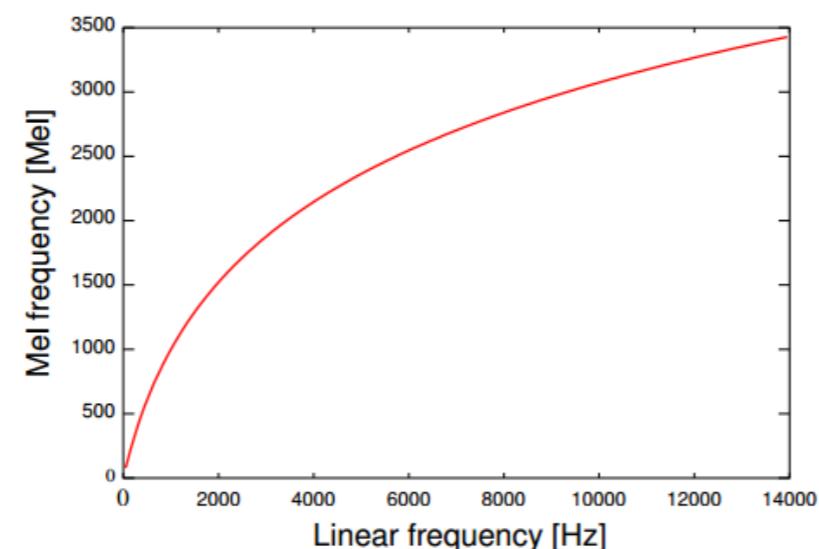
получили картинку ⇒ можно использовать алгоритмы из CV

<https://github.com/markovka17/dla/tree/master/week01>

Человеческое восприятие – Nonlinear frequency scaling

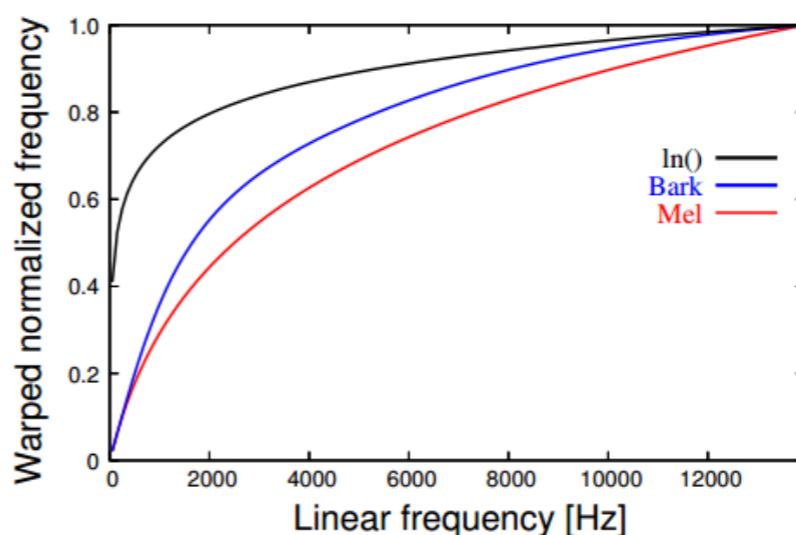
Mel scale

$$M(f) = 1127 \ln(1 + f/700)$$



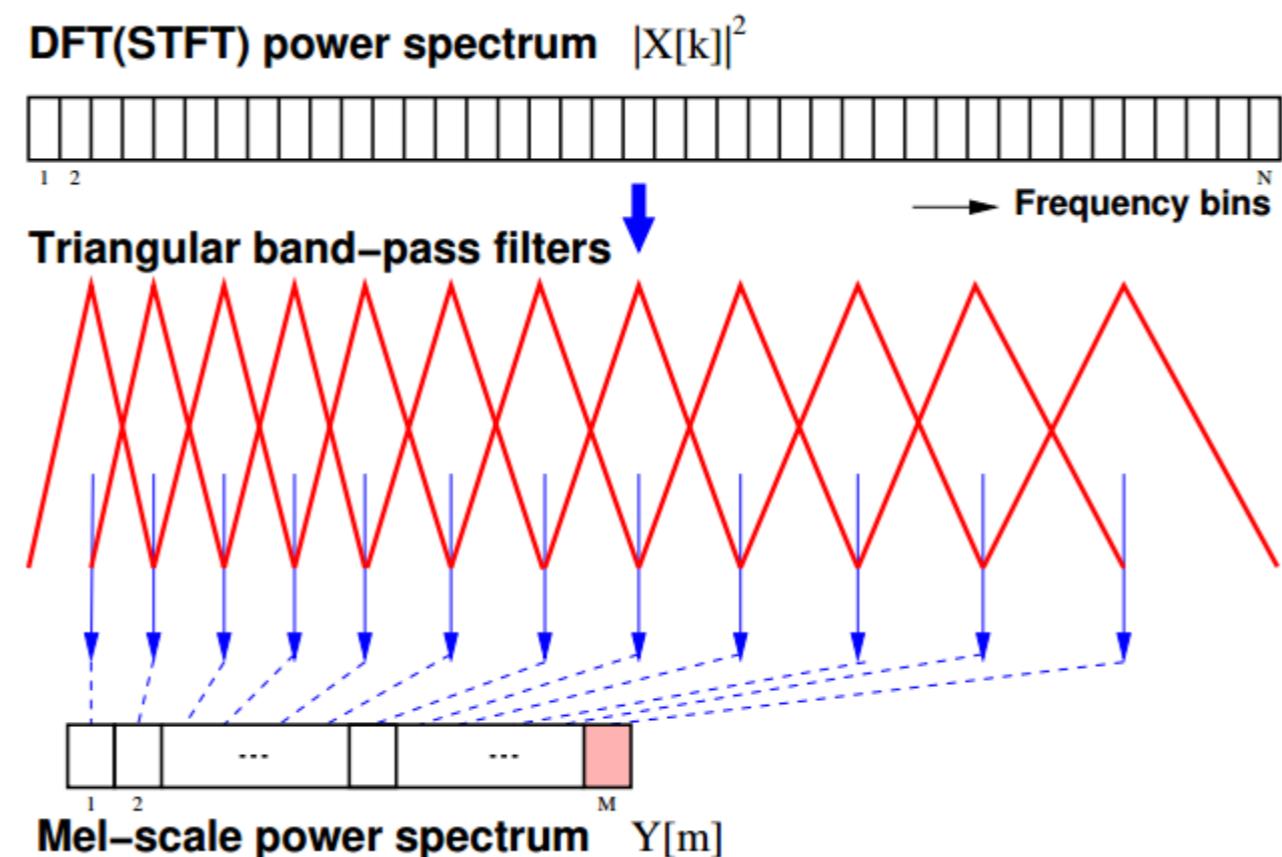
Bark scale

$$b(f) = 13 \arctan(0.00076f) + 3.5 \arctan((f/7500)^2)$$



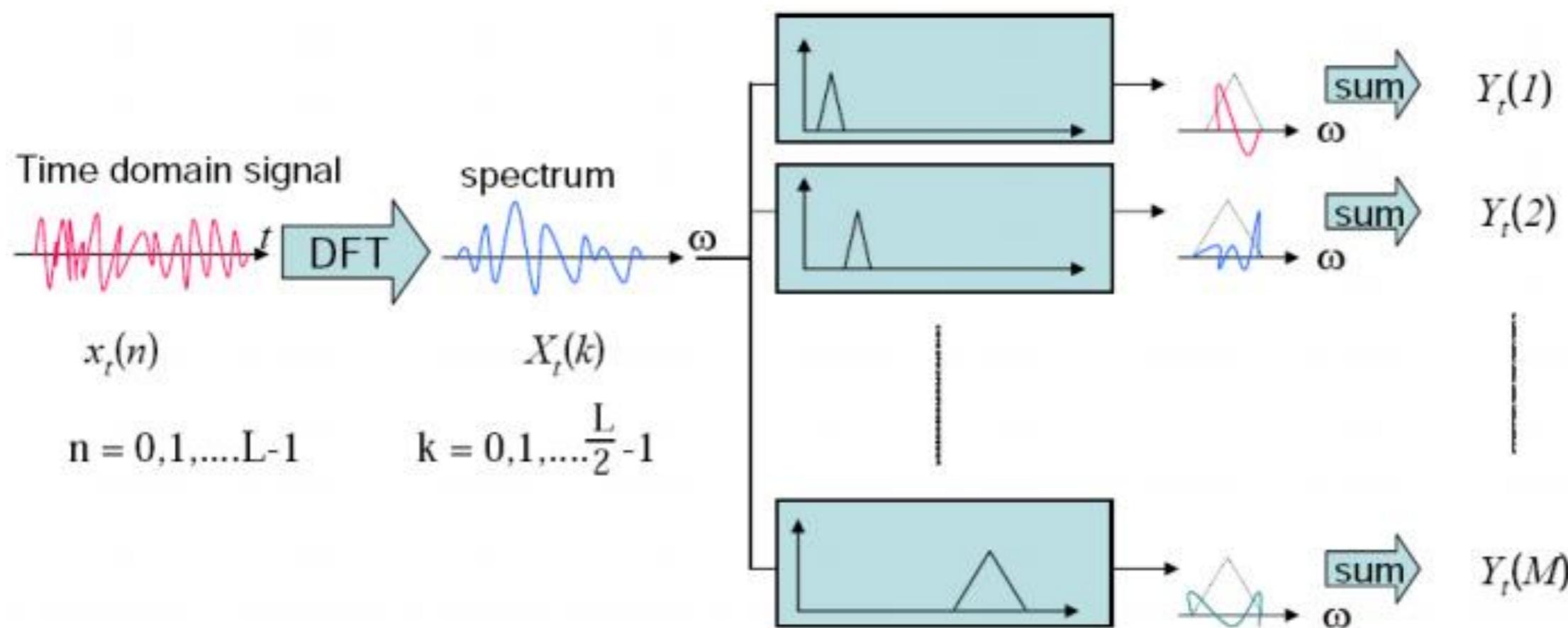
люди по-разному отличают высокие и низкие частоты (эквивалентна формуле выше)

$$f_{\text{mel}} = 2595 \log_{10}(1 + f/700)$$

mel-scale**mel-scale filter bank to DFT power spectrum $|X|^2$** **Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz****стандарт – 26 треугольных фильтров**

Mel filterbank

– применяем мел фильтры к спектру



[рис ?]

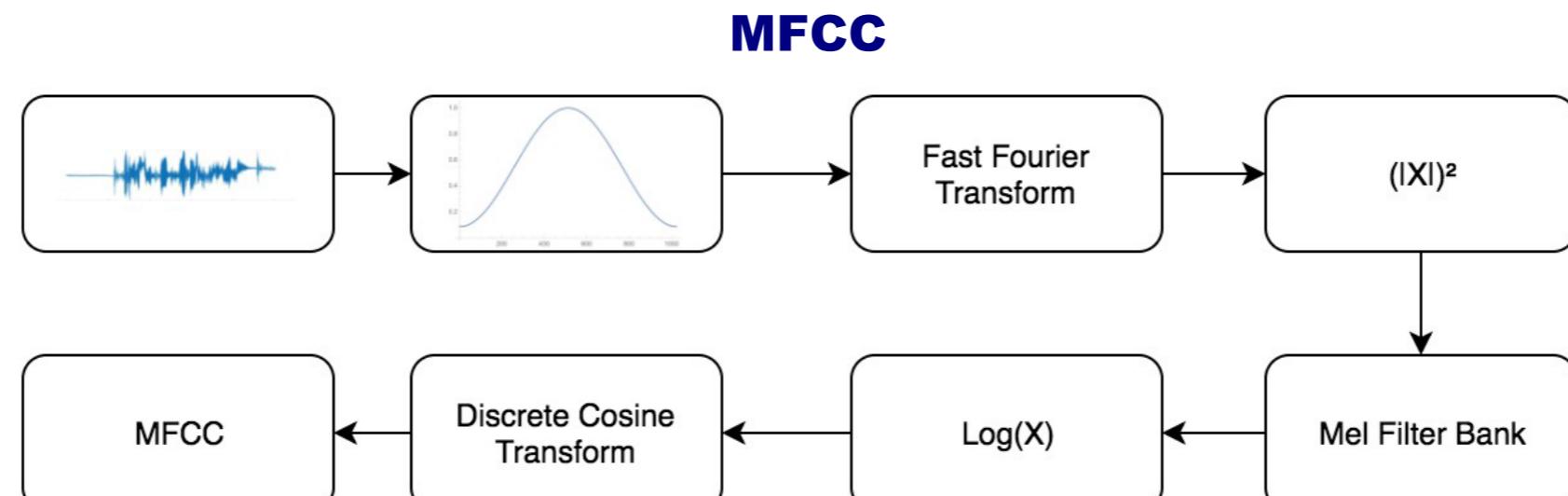
Log Mel Power Spectrum

Log |filter bank|²

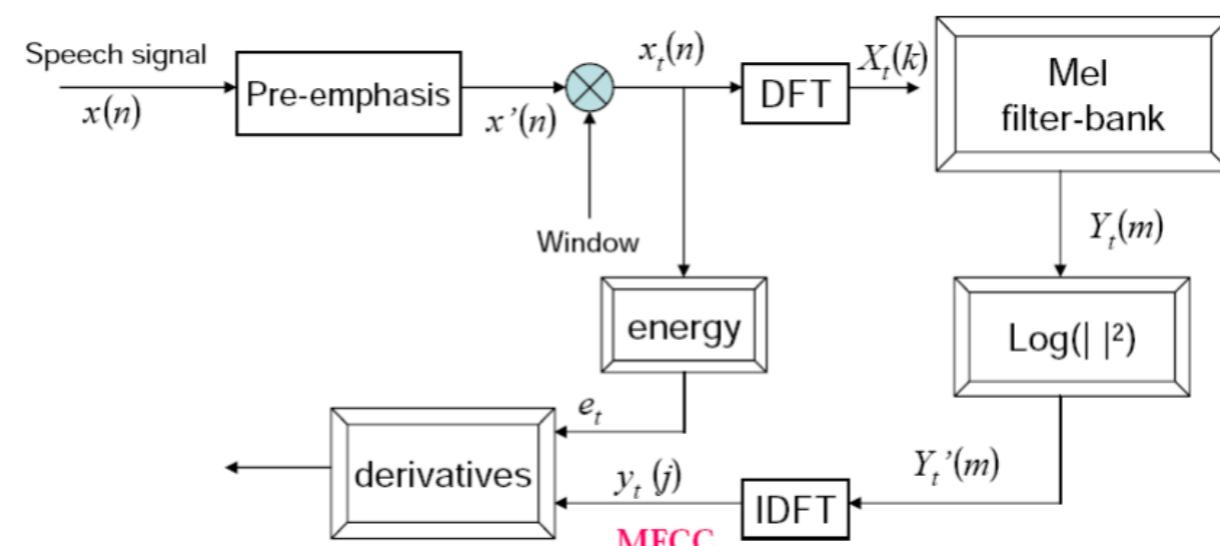
Информация о фазе не важна при распознавании

«Правильное» сжатие

Логарифм, т.к. человек не воспринимает громкость линейно



MFCC (Mel-Frequency Cepstral Coefficient)



MFCCs = Mel-frequency cepstral coefficients

1. Предварительно часто делают «Pre-emphasize» (усиление высоких частот):

$$x'_n = x_n - \alpha x_{n-1}, \alpha \in [0.95, 0.99]$$

иногда + шум (dithering)

2. Нарезаем сигнал на фреймы (перекрывающиеся)

3. Для каждого фрейма DFT

4. Применяем mel filterbank к «power spectra»

5. Log Mel Power Spectrum – логарифм от модуля квадрата

6. Берём DCT (=IDFT) (получается «cepstrum»), и его 2-13 коэффициенты (коэффициент F0 не используется)

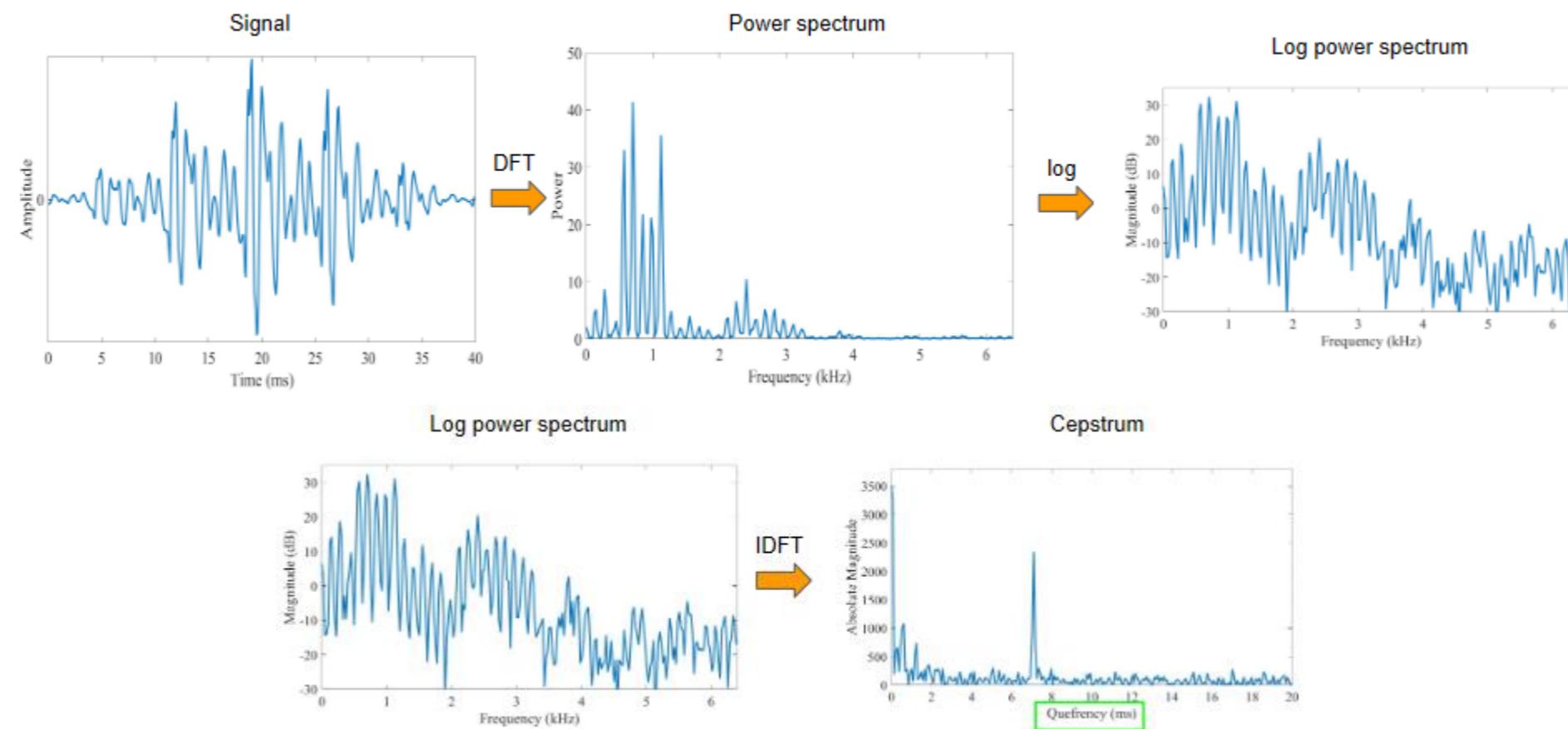
+ не сильно коррелируют

+ компактное представление (12 признаков на 20ms)

+ показали себя как лучшее признаковое описание сигнала

- не устойчивы к шуму, плохи для синтеза

MFCCs = Mel-frequency cepstral coefficients



Почему DFT – упрощённое FT, вещественные коэффициенты

Расширенное MFCCs (39 коэффициентов на фрейм)

12 MFCCs

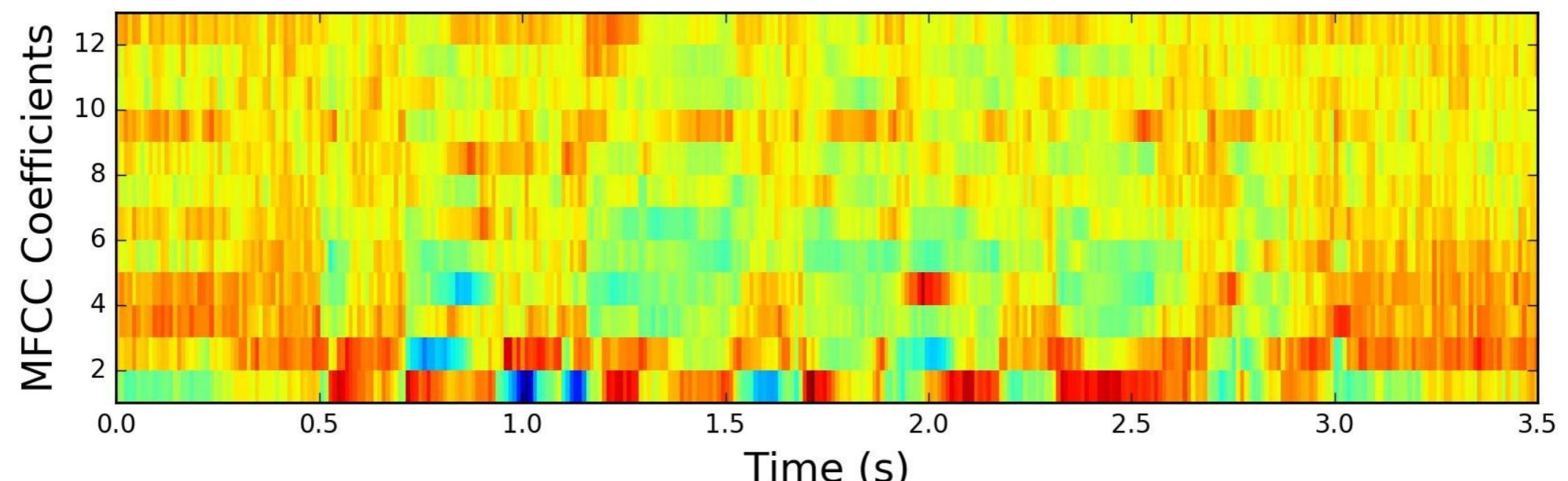
1 energy

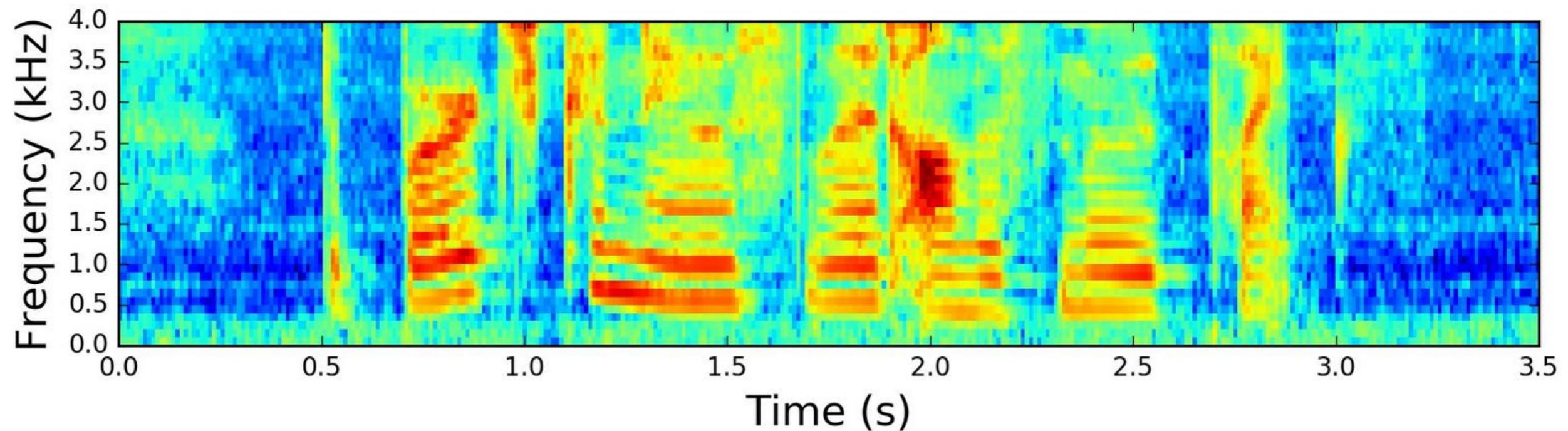
12 Δ MFCCs (как бы производные по времени)

12 изменений Δ (Δ Δ MFCCs)

Δ energy

Изменение Δ energy

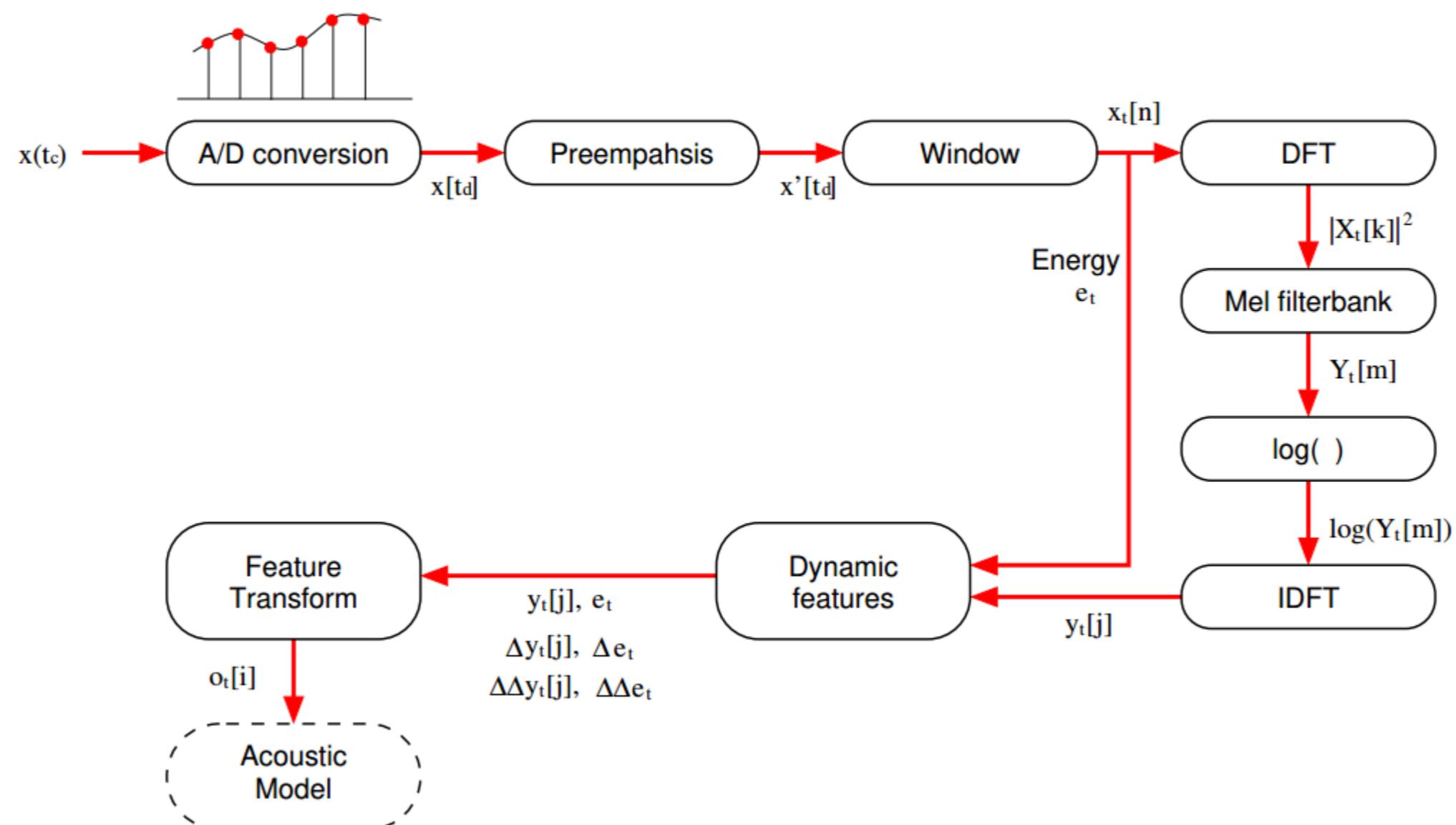


MFCC после вычитания среднего

```
filter_banks -= (numpy.mean(filter_banks, axis=0) + 1e-8)
```

<https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>

Расширенное MFCCs



Что используют разные методы...

DS — 80 dimensional linearly spaced log filterbanks + energy term with a 20ms window strided by 10ms

DS2 — raw data

DS3 — PCEN

Wav2letter++ — mfcc, mfsc, raw audio, linearly scaled power spectrum

LAS (filterbanks, 2016) — 80 dimensional filterbanks computed with a 25ms window every 10ms with delta and deltadelta acceleration normalized with per speaker mean and variance

LAS (filterbanks, 2018) — 80 dimensional log-mel features computed with a 25ms window every 10ms + stacked with 3 frames left and downsampled to a 30ms frame rate.

Показатель качества: WER = Word Error Rate

$$\text{WER} = 100 \cdot \frac{S + D + I}{N} \%$$
$$\text{Accuracy} = 100 - \text{WER}\%$$

N – число слов в транскрипции**S = substitutions****D = deletions****I = insertions**

- Word Error Rate =

$$\frac{100 (\text{Insertions} + \text{Substitutions} + \text{Deletions})}{\text{Total Word in Correct Transcript}}$$

REF: portable **** PHONE UPSTAIRS last night so

HYP: portable FORM OF STORES last night so

Eval I S S

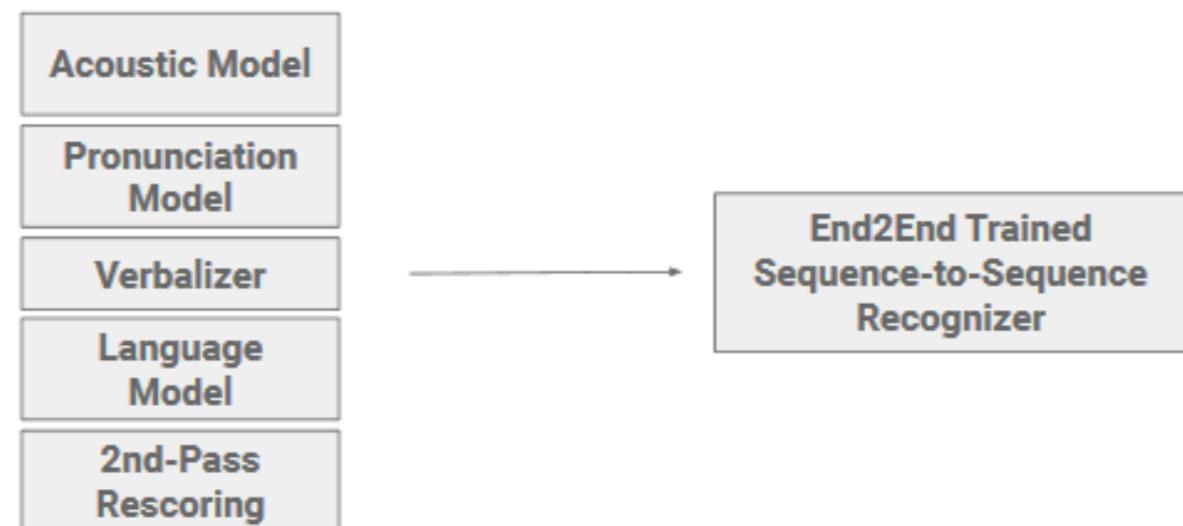
- WER = $100 (1+2+0)/6 = 50\%$

Jurafsky and Martin (2008). Speech and Language Processing

Что такое end2end-подход в ASR?

система, которая напрямую переводит последовательность акустических признаков в последовательность слов

обучается оптимизируя нужный критерий – WER

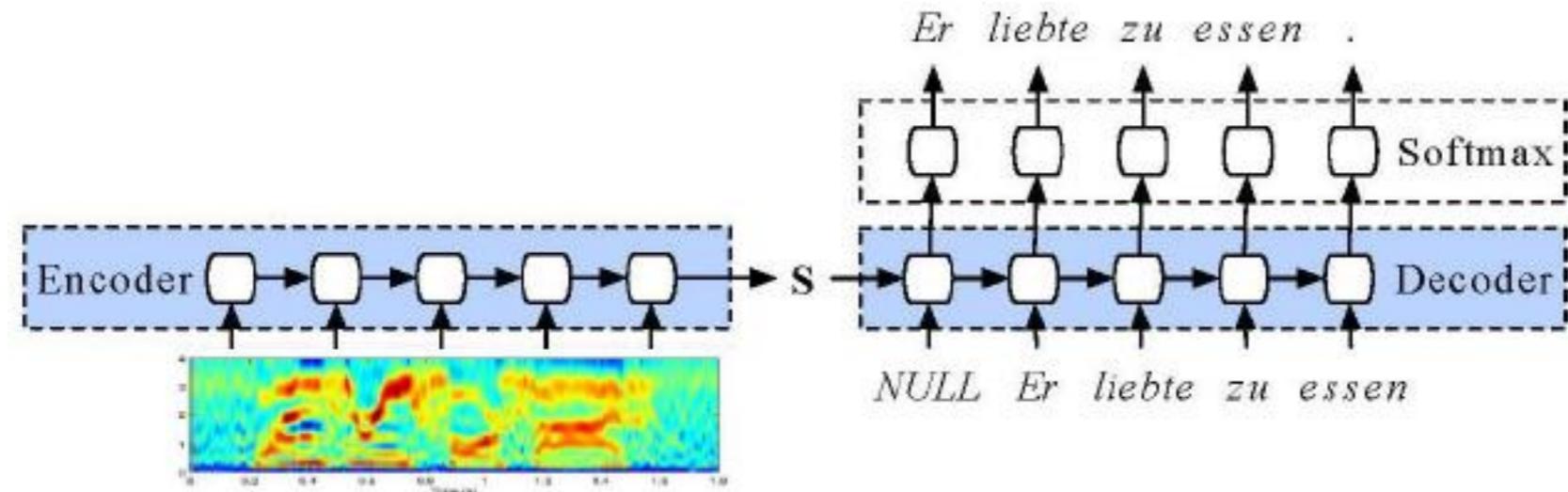


что ещё желательно продумать...

Online – для реальных приложений

Production – взаимодействие с другими компонентами

Напрашивающиеся нейросетевые подходы seq2seq (как в МТ)



можно учить напрямую, без выравнивания

вход: acoustic frames

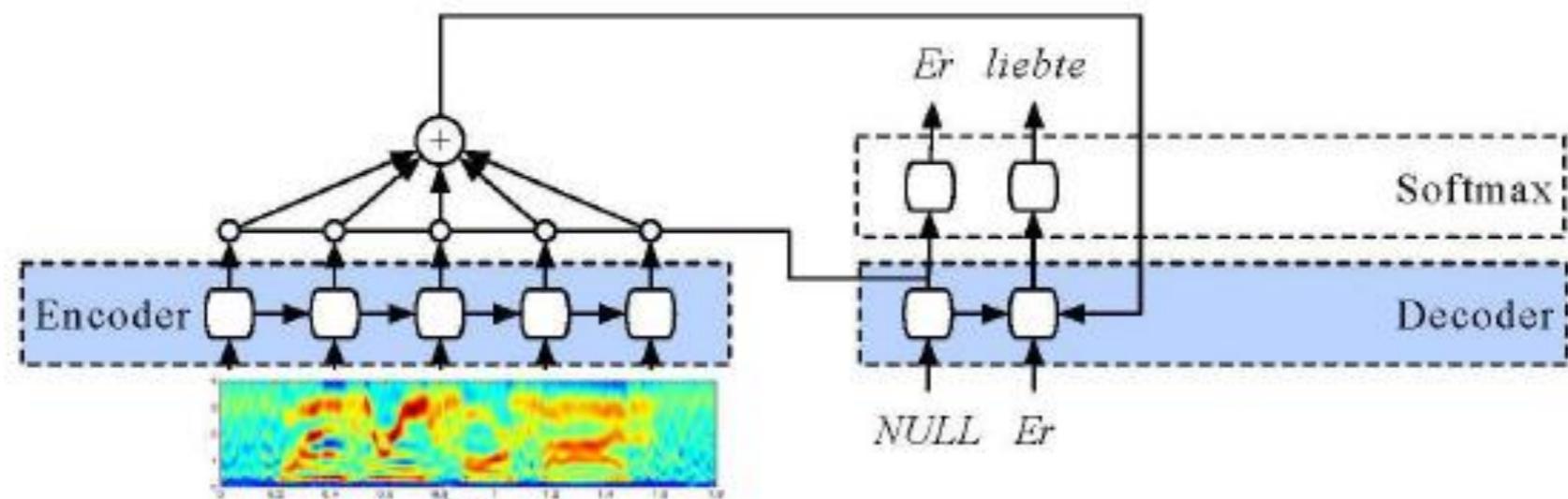
выход: characters/words

cross-entropy (log loss)

идею можно продолжить:
разные виды рекурентностей
дву направленность
внимание

Напрашающиеся нейросетевые подходы

Attention models



как всегда, *Scheduled sampling*
вычислительно трудоёмко... выход вся посл-ть
монотонность в выравнивании (вроде как должна быть)

СТС (Connectionist Temporal Classification)

метод для разметки несегментированных последовательностей

Дифференцируемая целевая функция

Позволяет не размечать каждую букву в аудиосигнале

обучение без предварительного выравнивания по фреймам!

Входной и выходной сигналы могут быть разной длины

Сначала на выходе были фонемы, поэтому не end2end...

<ftp://ftp.idsia.ch/pub/juergen/icml2006.pdf>

СТС (Connectionist Temporal Classification)

такая ошибка возникает не только в ASR

the quick brown fox



The quick brown fox

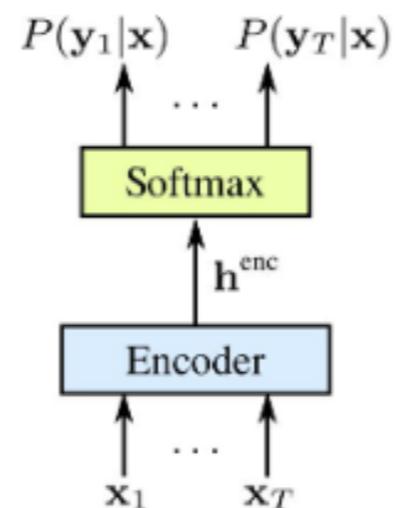
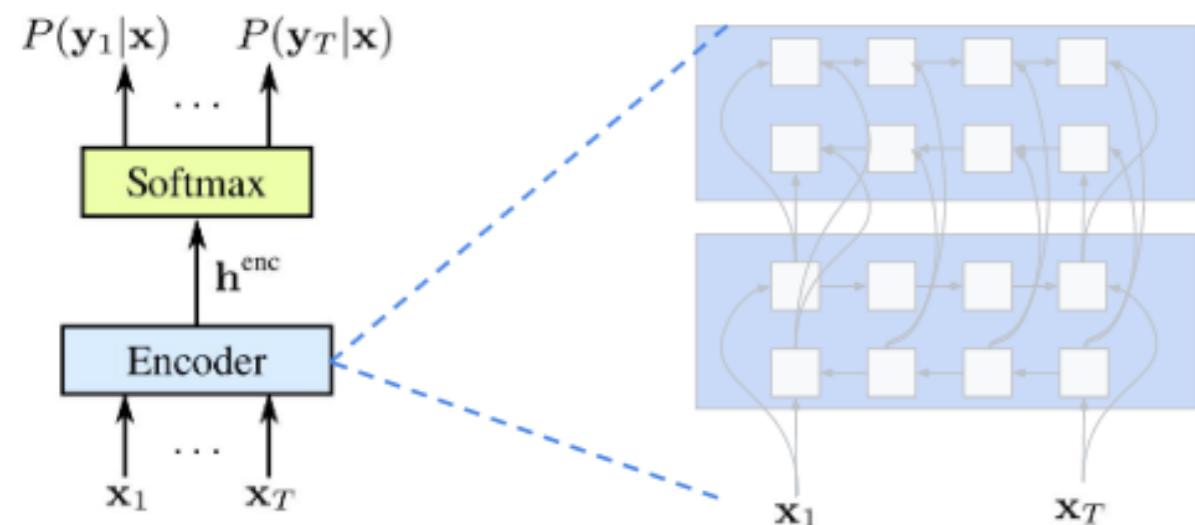
Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.

jumps over the lazy dog



Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

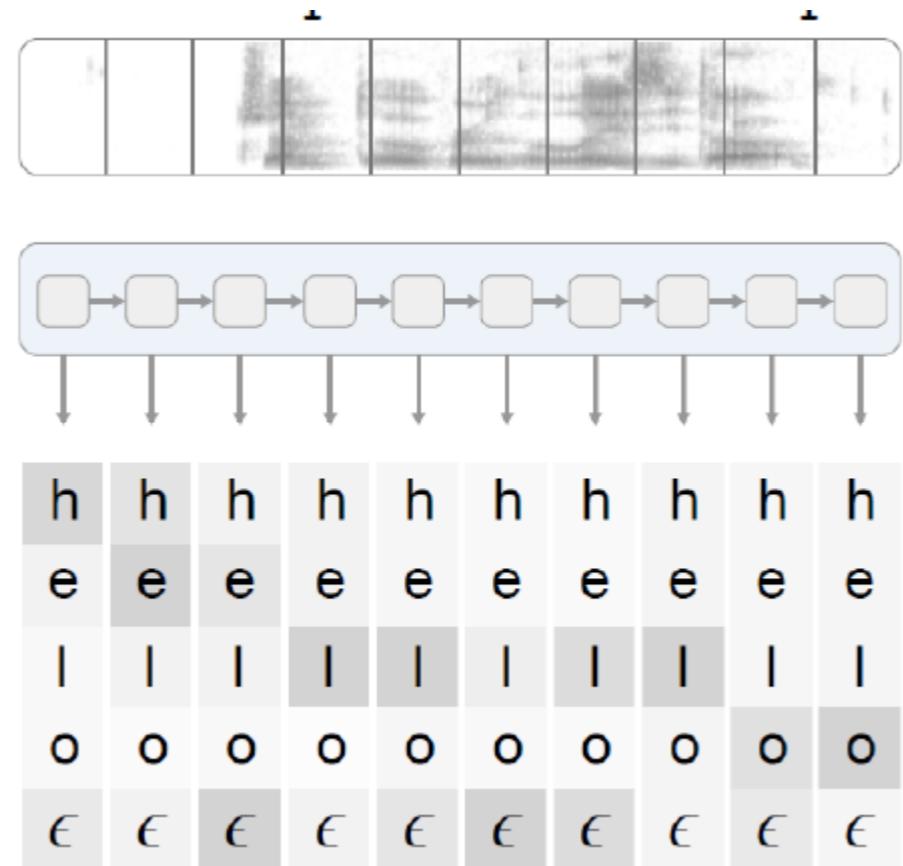
СТС (Connectionist Temporal Classification)



B	B	c	B	B	a	a	B	B	t
B	c	c	B	a	B	B	B	B	t
...									
B	c	B	B	a	B	B	t	t	B

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in \mathcal{B}(\mathbf{y}, \mathbf{x})} \prod_{t=1}^T P(\hat{y}_t|\mathbf{x})$$

СТС (Connectionist Temporal Classification)



**RNN с softmax-выходом выдаёт вероятности символов + ε
(на каждом фрейме)**

CTC (Connectionist Temporal Classification)

h h e ε ε l l l ε l l o

h e ϵ | ϵ | o

h e | | l o

h e l l o

**что выдали надо
схлопнуть повторения
удалить спецсимвол**

СТС (Connectionist Temporal Classification)

**Есть разные варианты «выравнивания»
(даже для конкретного правильного ответа):**

Valid Alignments

€ | c | c | € | a | t

c | c | a | a | t | t

c | a | € | € | € | t

Invalid Alignments

c | € | c | € | a | t

c | c | a | a | t | _

c | € | € | € | t | t

corresponds to
 $Y = [c, c, a, t]$

has length 5

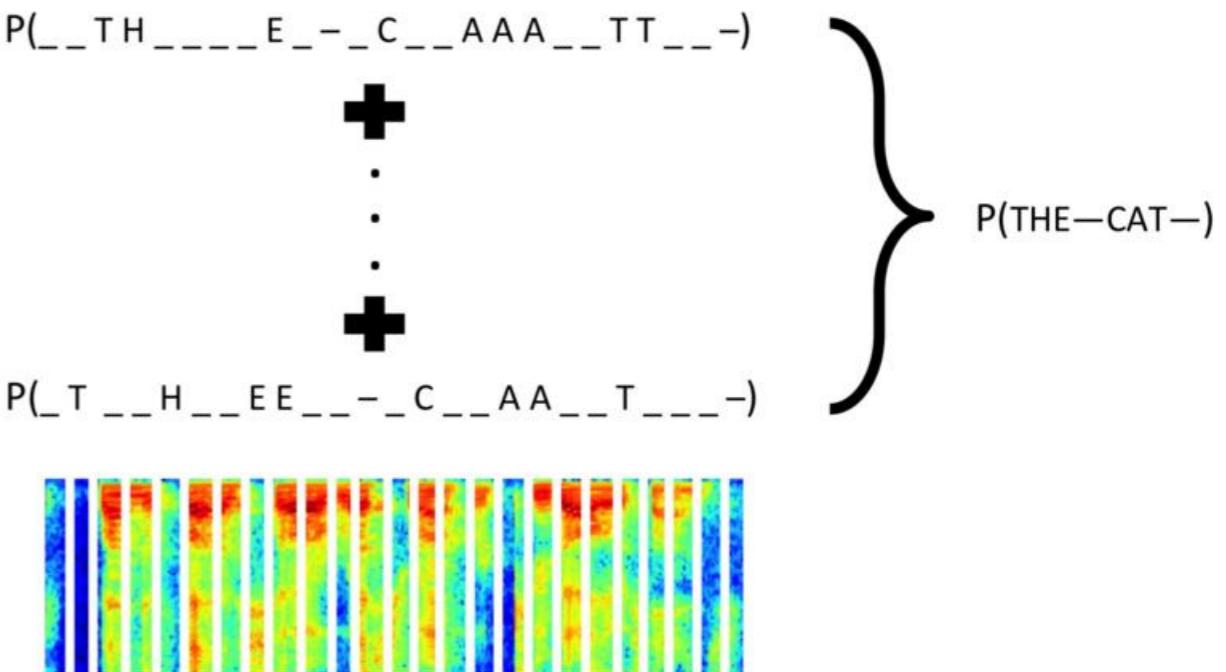
missing the 'a'

[Hannun 2017]

СТС (Connectionist Temporal Classification)

**если $A_{X,Y}$ – множество всех валидных посл-ей,
тогда marginal log loss:**

$$-\log p(Y | X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^T (a_t | X)$$

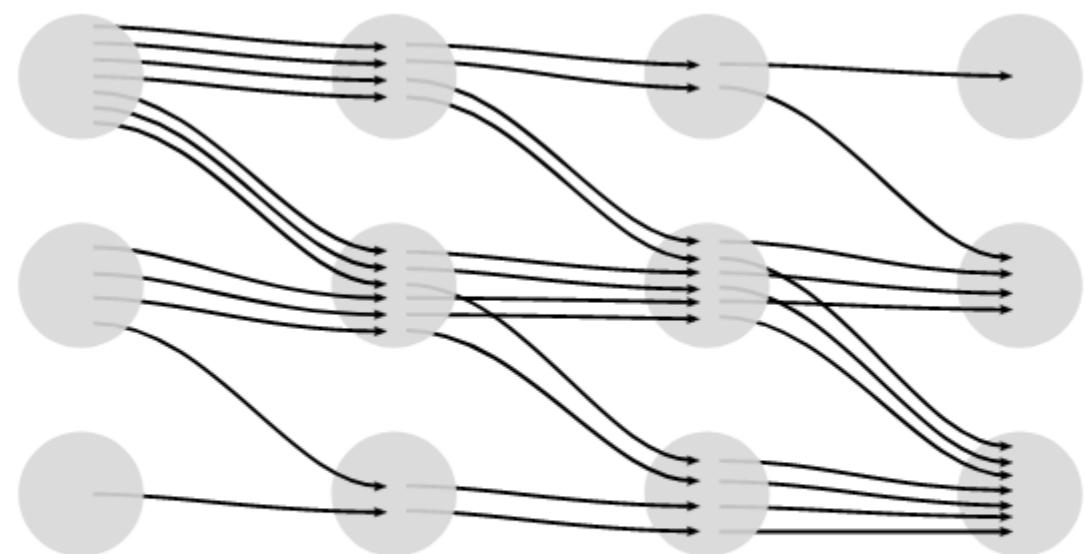


<https://github.com/baidu-research/warp-ctc>

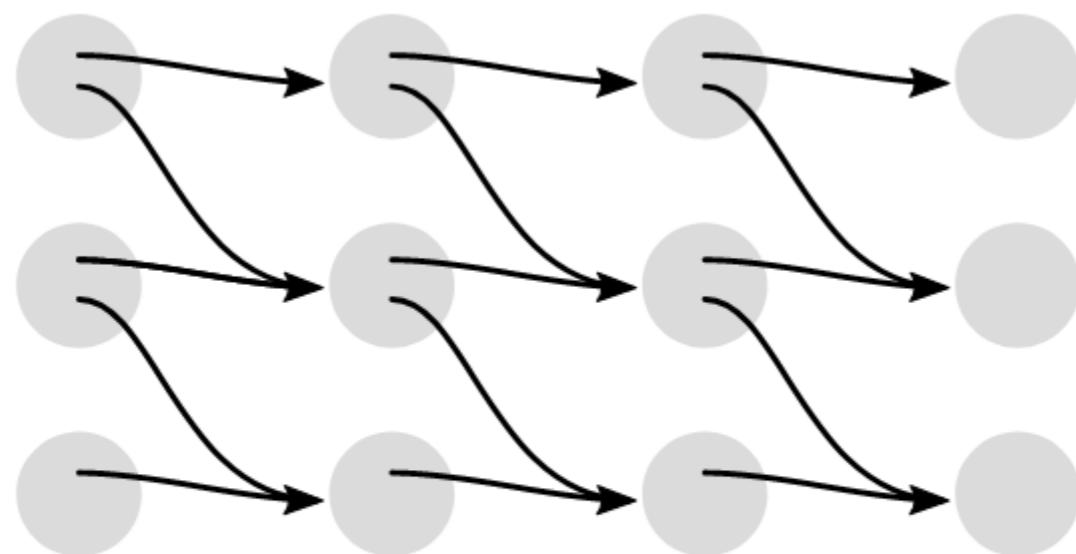
на вид сложно делать обратное распространение...
но это похоже на forward-backward-HMM-algorithm!

**СТС-loss ~ softmax-слой, кол-во выходов = число символов + 1 (blank)
Получаем вероятности символов**

СТС: есть приёмы быстрого вычисления



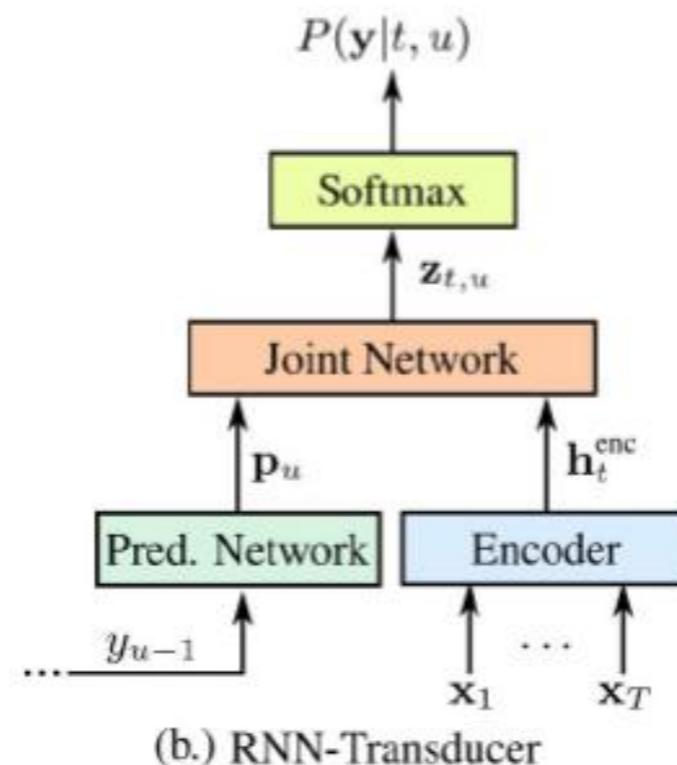
Summing over all alignments can be very expensive.



Dynamic programming merges alignments, so it's much faster.

см. <https://distill.pub/2017/ctc/>

Recurrent Neural Network Transducer (RNN-T)



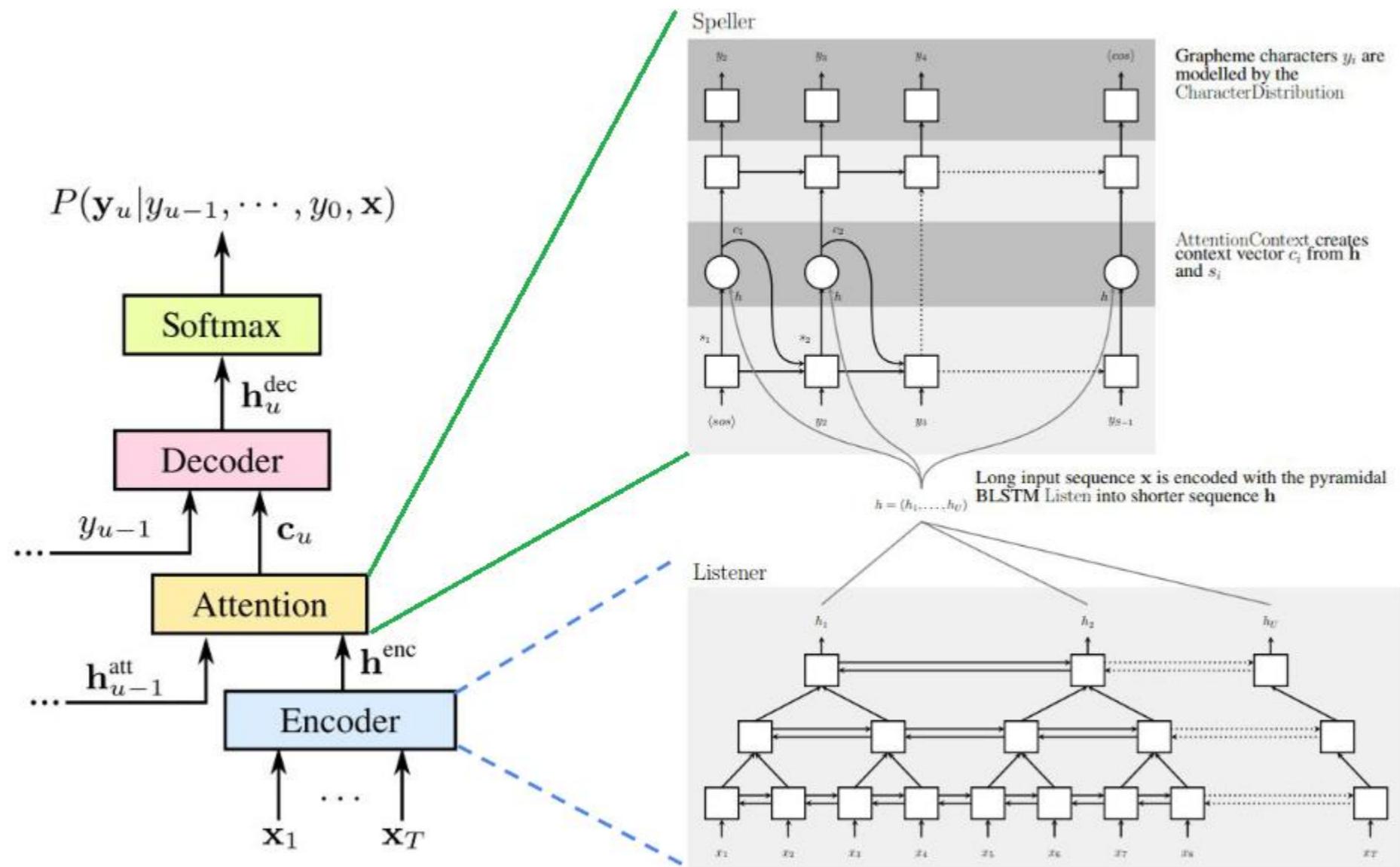
Listen, Attend and Spell (LAS)

Sequence-to-sequence + attention модель для генерации
Sampling trick при обучении (комбинирование teacher forcing и
free run)

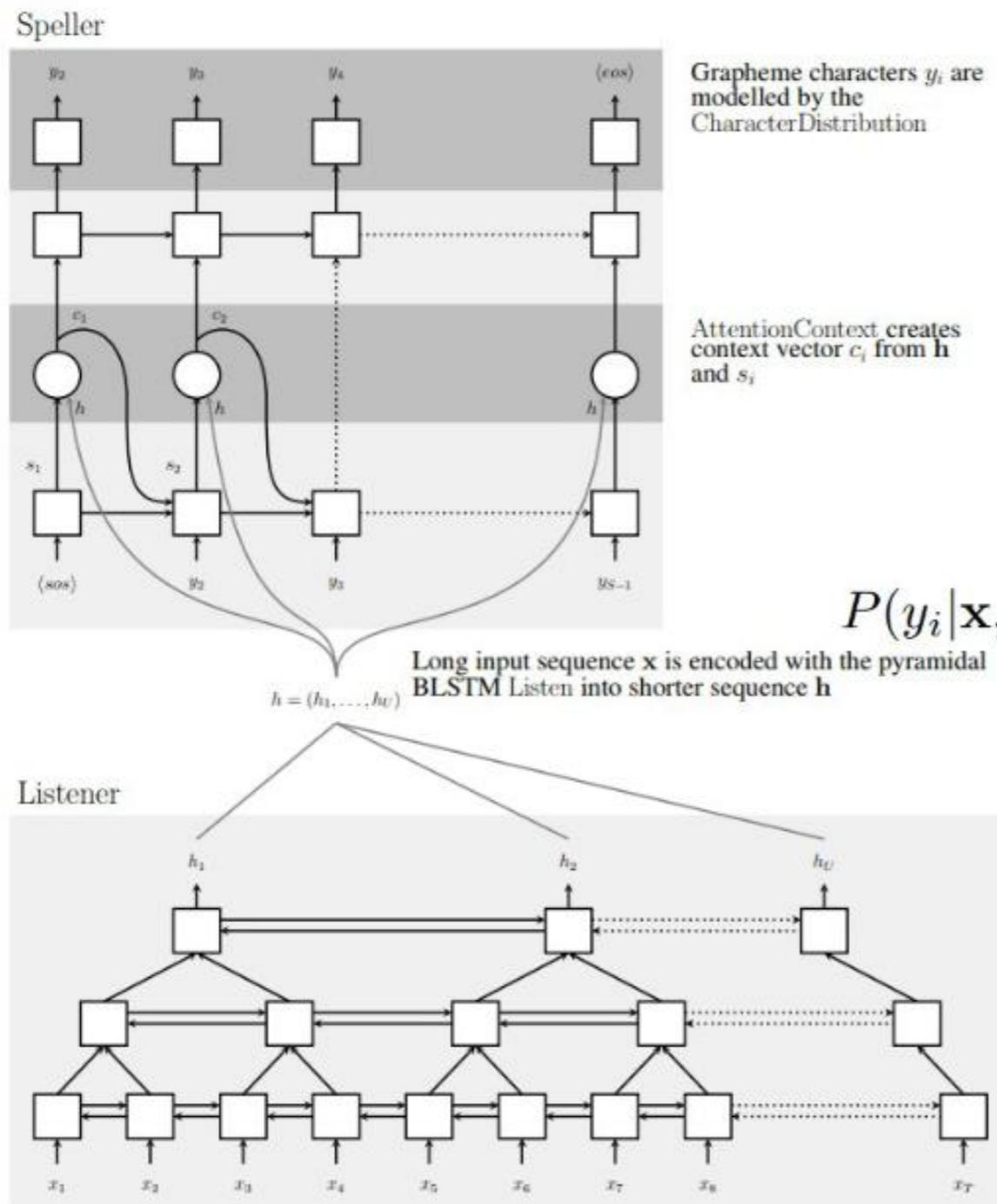
- **Listener — pyramidal bi-LSTM encoder**
уменьшает время обучения («кодирует» аудио)
- **Speller — attention-based RNN-decoder**
(декодирование в транскрипцию)
Декодирование – beam search
- **На входе — 40-dimensional log-mel filter bank features (10 ms)**
- **На выходе — символы (нет фонем)**

William Chan, Navdeep Jaitly, Quoc V. Le, Oriol Vinyals «Listen, Attend and Spell»
<https://arxiv.org/pdf/1508.01211.pdf>

Listen, Attend and Spell (LAS)



окончание – вывод «EOS»


 $\mathbf{h} = \text{Listen}(\mathbf{x})$

$$P(\mathbf{y}|\mathbf{x}) = \text{AttendAndSpell}(\mathbf{h}, \mathbf{y})$$

Cost Function:

$$s(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|_c} + \lambda \log P_{\text{LM}}(\mathbf{y})$$

Attend and spell:

$$c_i = \text{AttentionContext}(s_i, \mathbf{h})$$

$$s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1})$$

$$P(y_i|\mathbf{x}, y_{<i}) = \text{CharacterDistribution}(s_i, c_i)$$

Deep Bidirectional LSTM:

$$h_i^j = \text{BLSTM}(h_{i-1}^j, h_i^{j-1})$$

Pyramidal Bidirectional LSTM:

$$h_i^j = \text{pBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}])$$

Listen, Attend and Spell (LAS)

- **Dataset – Google voice search (~2000 часов)**
 - **Hold-out validation (~10 часов)**
- **Тестовая выборка — 22К высказываний (~16 часов)**
 - **Аугментация (добавляли шум) — увеличение выборки ~в 20 раз**
- **Listener — 3 слоя 512 pBLSTM (256 на направление), уменьшение в 8 раз**
 - **Speller — 2 слоя LSTM по 512**
 - **ASGD**

LAS учитывает контекст, поэтому не онлайн!

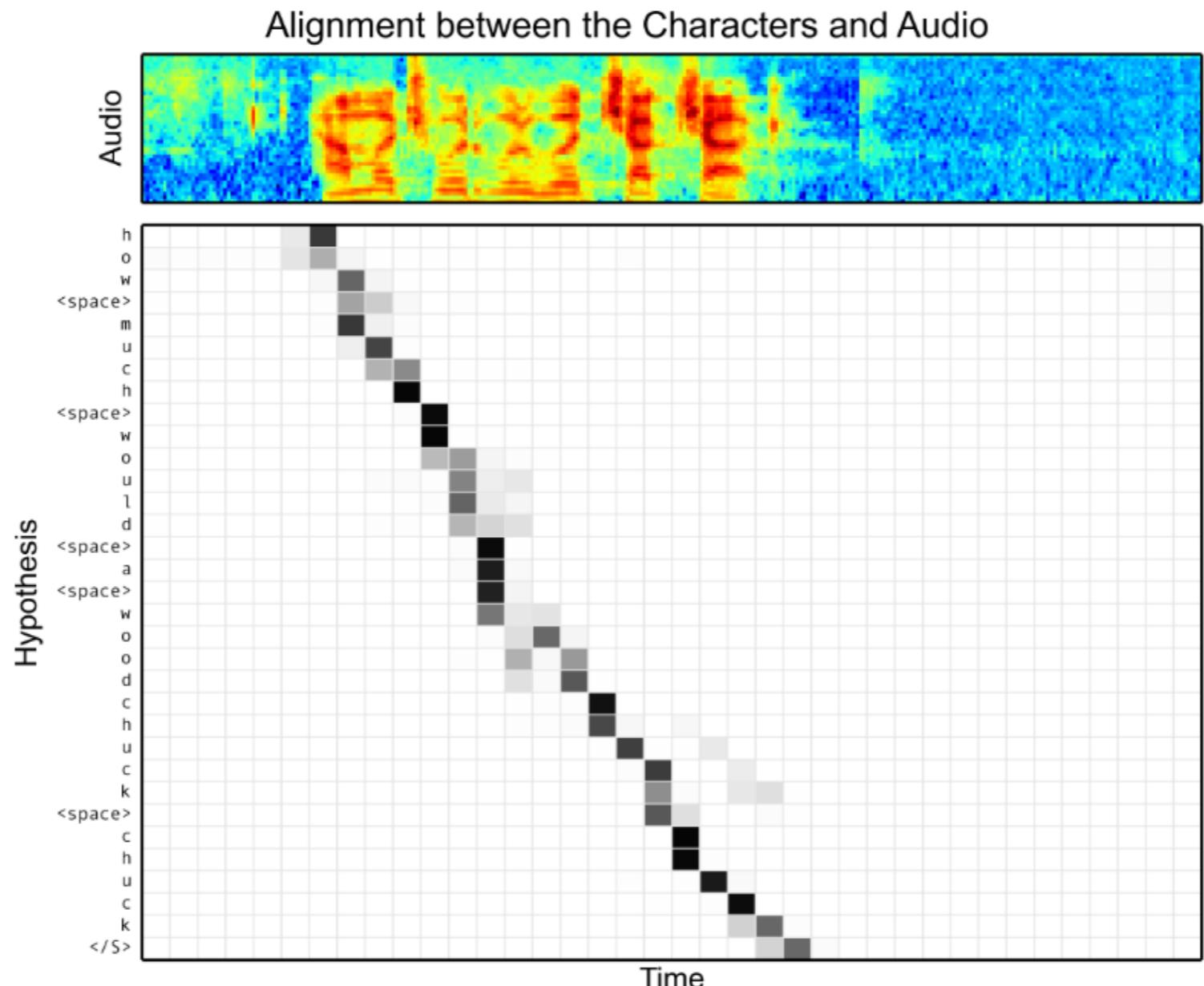
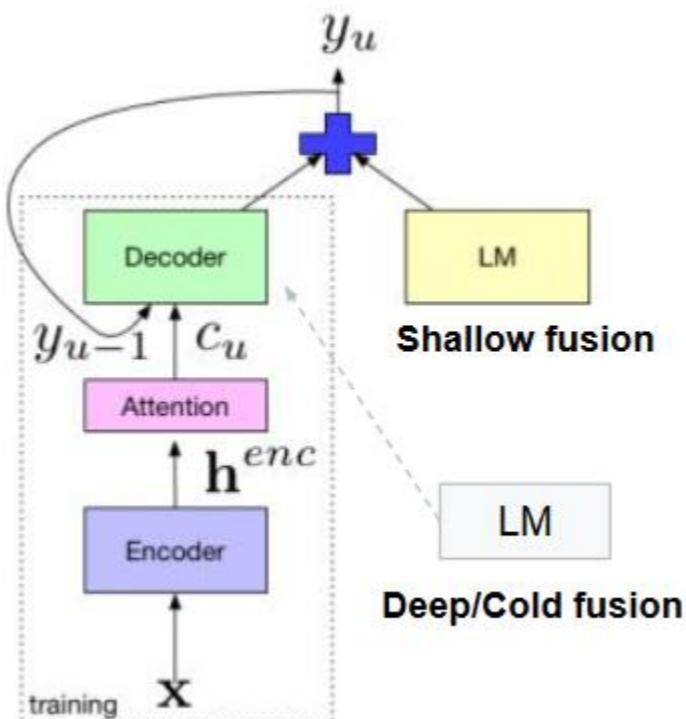


Figure 2: Alignments between character outputs and audio signal produced by the Listen, Attend and Spell (LAS) model for the utterance “how much would a woodchuck chuck”. The content based attention mechanism was able to identify the start position in the audio sequence for the first character correctly. The alignment produced is generally monotonic without a need for any location based priors.

Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art system, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring was applied to our beams, and a sampling trick was applied to bridge the gap between training and inference.

Model	Clean WER	Noisy WER
CLDNN-HMM [20]	8.0	8.9
LAS	16.2	19.0
LAS + LM Rescoring	12.6	14.7
LAS + Sampling	14.1	16.5
LAS + Sampling + LM Rescoring	10.3	12.0

Использование языковой модели: External Language Model



Shallow fusion [Kannan et al., 2018] – LM на выходе

Deep fusion [Gulcehre et al., 2015] – LM фиксирована

Cold fusion [Sriram et al., 2018] – простой интерфейс между LN и кодировщиком, можно переключаться в задаче-ориентированной манере

Использование языковой модели: Shallow / Deep / Cold Fusion

Table 2: Word error rates (%) on Google voice search (VS14K) and dictation data sets (D15K) for the baseline model and fusion approaches.

Model	VS14K	D15K
LAS	5.6	4.0
Shallow Fusion	5.3	3.7
Deep Fusion	5.5	4.1
Cold Fusion	5.3	3.9

После добавления языковых моделей проблемы с редкими словами и именами собственными остаются по-прежнему

Предположение в том, что языковая модель не учитывает тех ошибок, которые совершает end-to-end модель

S. Toshniwal, A. Kannan, C. C. Chiu, and et al, «A comparison of techniques for language model integration in encoderdecoder speech recognition», in to appear in Proc. SLT, 2018.

Spelling correction модель

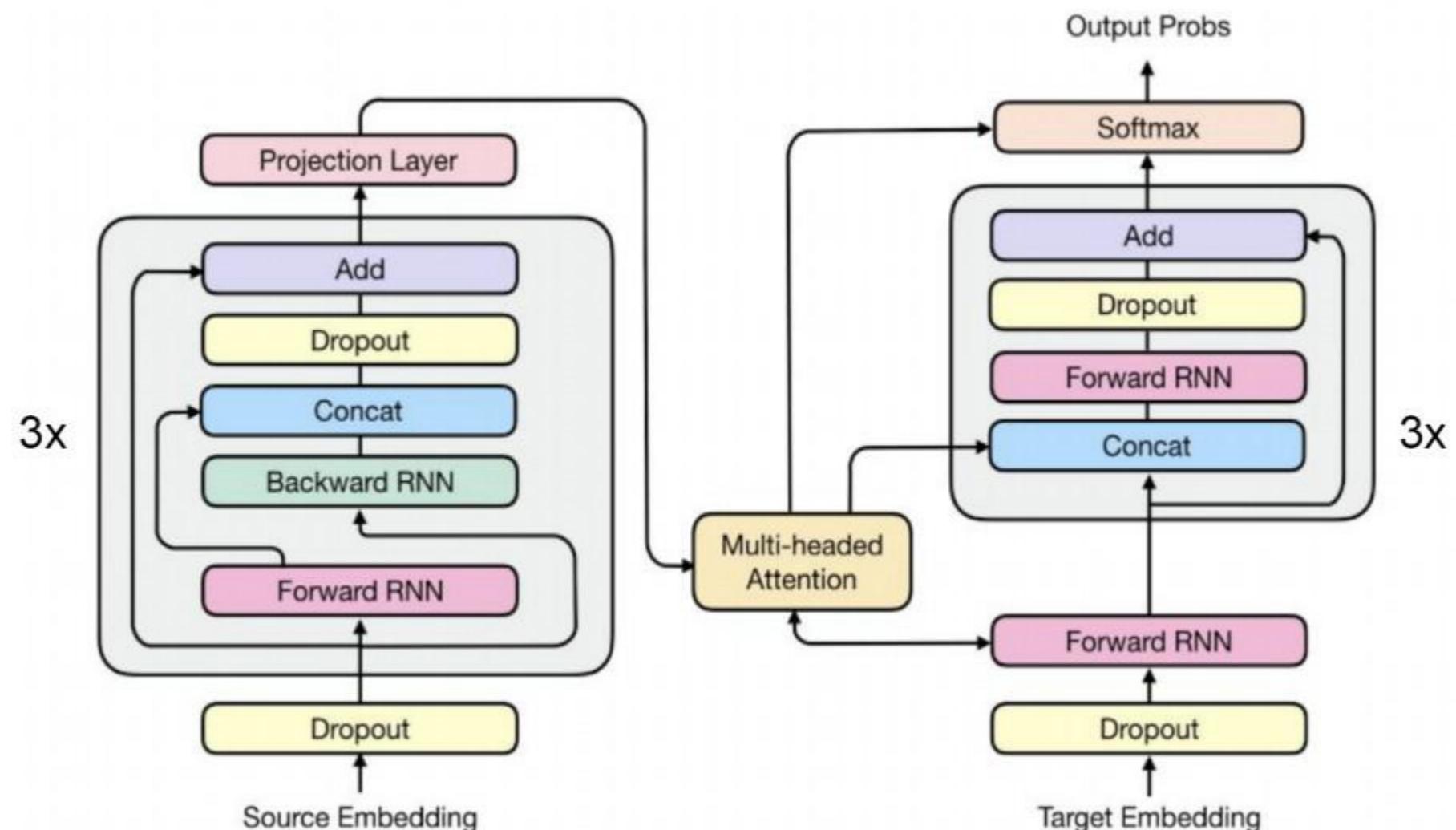


Fig. 1. Spelling Correction model architecture.

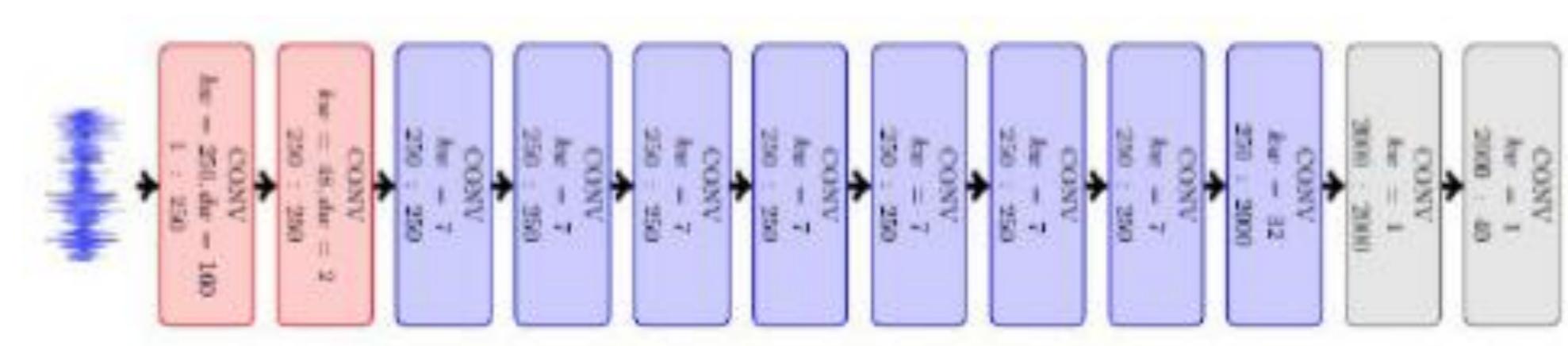
Spelling correction модель

LAS + LM rescore	LAS + SC + LM rescore
ready to hand over to trevellion	ready to hand over to trevelyan
has countenance the belief the hope the wish that the epeanites or at least the nazarines	has countenanced the belief the hope the wish that the ebionites or at least the nazarenes
a wandering tribe of the blamis or nubians	a wandering tribe of the blemmyes or nubians

Table 4. LAS + SC + LM rescore Wins. LAS + LM rescore (in bold)

J. Guo, Tara N. Sainath, Ron J. Weiss, «A spelling correction model for end-to-end speech recognition», ICASSP, 2019

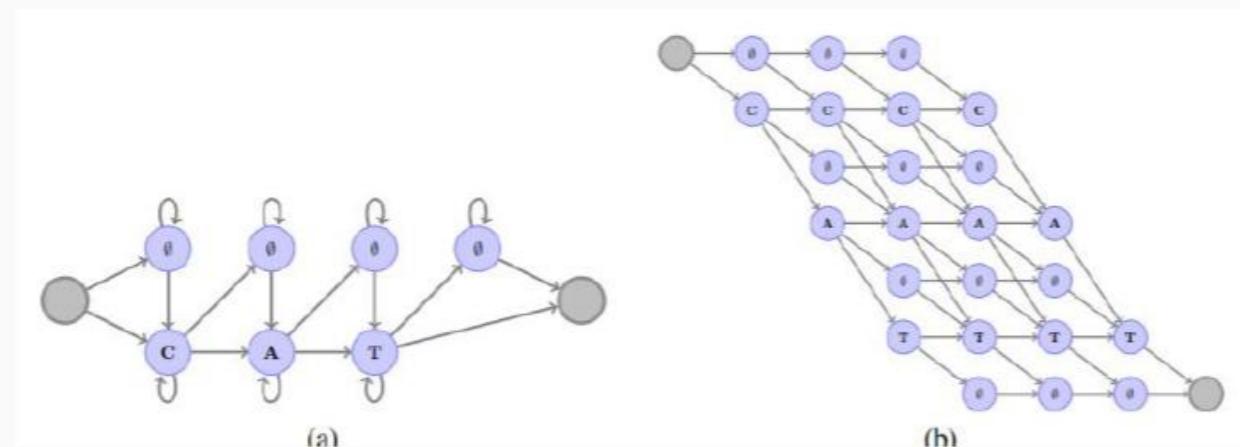
Wav2Letter



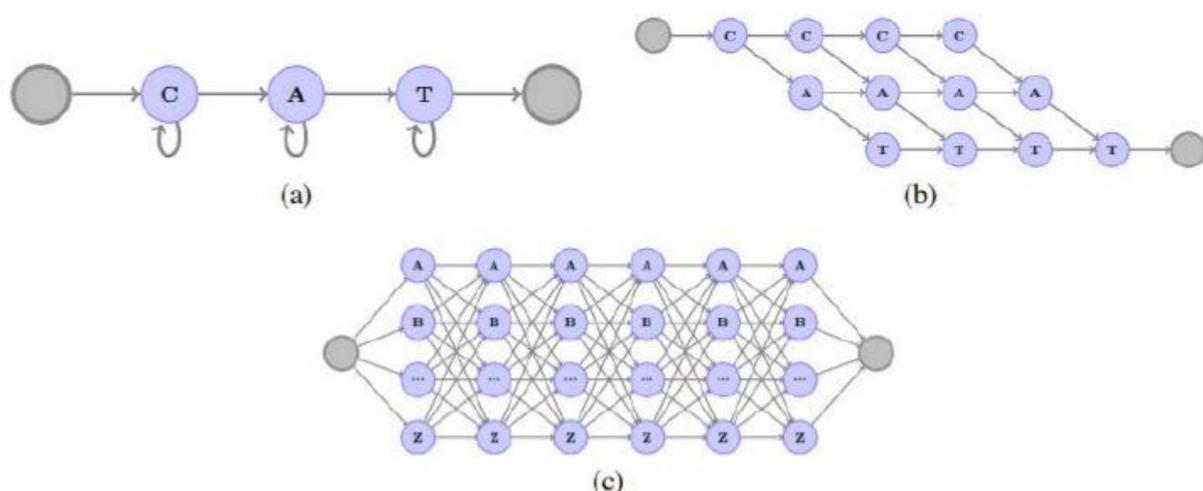
Полностью свёрточная архитектура от FaceBook

СТС & ASG

СТС



ASG



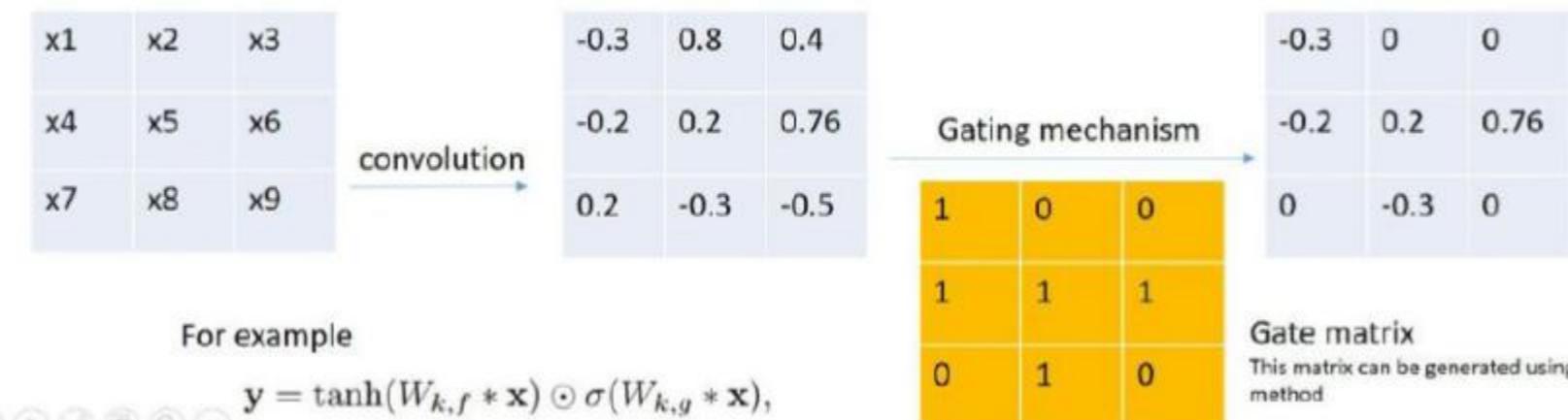
	ASG	CTC
dev-clean	10.4	10.7
test-clean	10.1	10.5

batch size	CTC		ASG CPU
	CPU	GPU	
1	1.9	5.9	2.5
4	2.0	6.0	2.8
8	2.0	6.1	2.8

batch size	CTC		ASG CPU
	CPU	GPU	
1	40.9	97.9	16.0
4	41.6	99.6	17.7
8	41.7	100.3	19.2

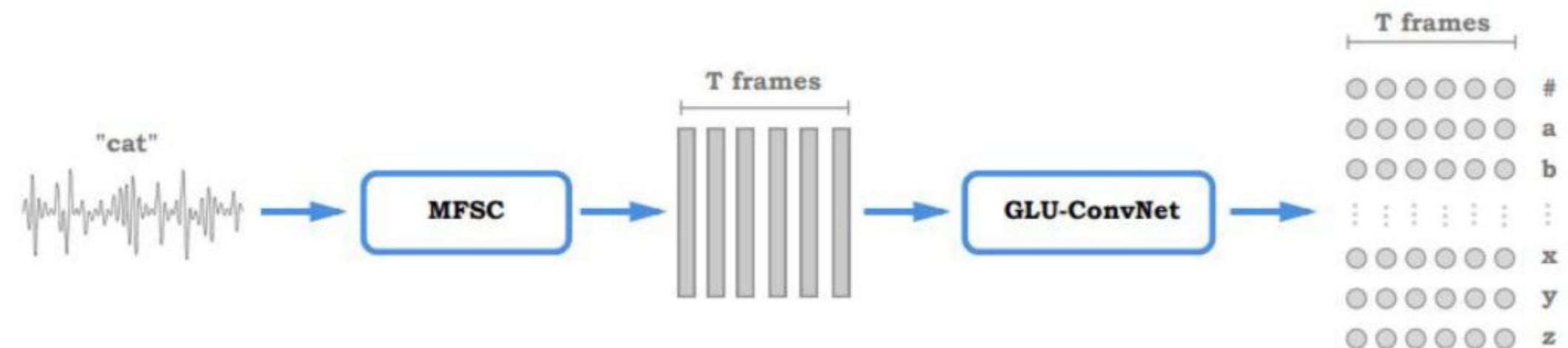
Gated ConvNets

Gated CNN

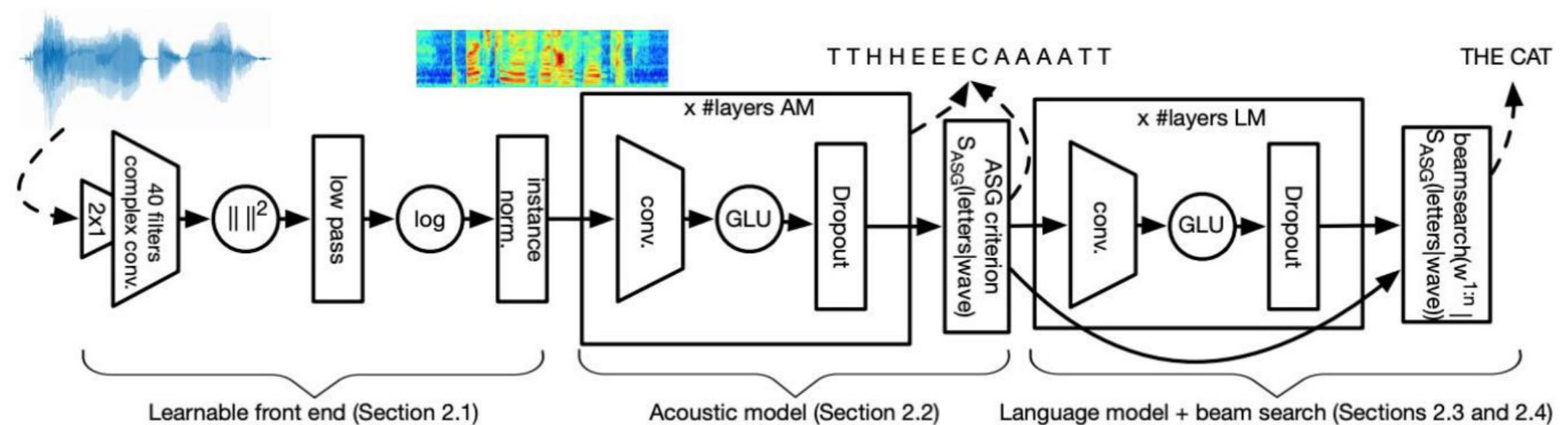


For example

$$y = \tanh(W_{k,f} * x) \odot \sigma(W_{k,g} * x),$$



Wav2Letter++



Dataset	Architecture	#conv.	dropout first/last layer	#hu first/last layer	kw first/last layer	#hu full connect
WSJ	Low Dropout	17	0.25/0.25	100/375	3/21	1000
LibriSpeech	Low Dropout	17	0.25/0.25	200/750	13/27	1500
	High Dropout	19	0.20/0.60	200/1000	13/29	2000

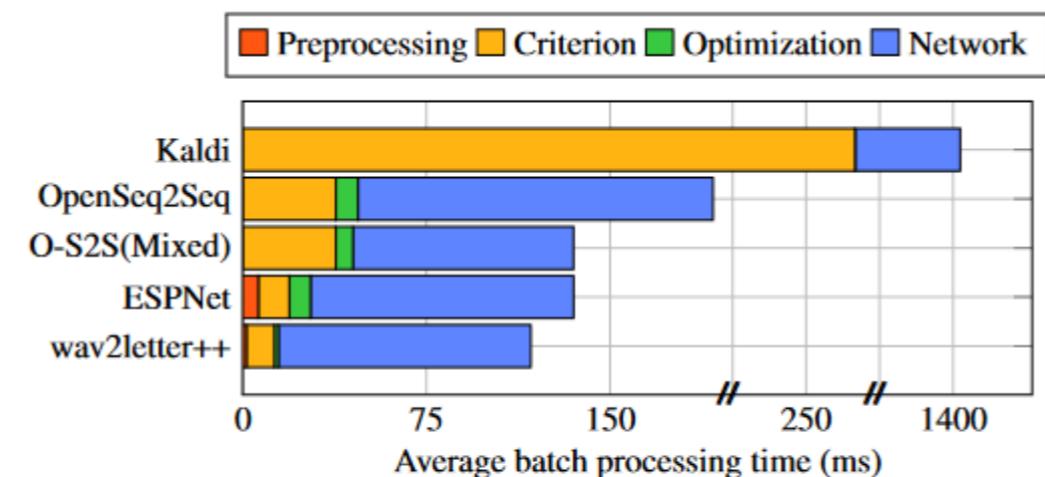


Fig. 3. Time in milliseconds for the major steps in the training loop. The times are averaged for each batch over a full epoch.

Vineel Pratap et. al. «WAV2LETTER++: THE FASTEST OPEN-SOURCE SPEECH RECOGNITION SYSTEM // <https://arxiv.org/pdf/1812.07625.pdf>

ASR Baidu: DeepSpeech

«Baidu Research»

Не моделируются

- **шумы**
- **смена спикера**
 - **эхо**

– сразу робастное обучение

Нет словаря фонем

И даже понятия «фонема»!

**RNN + новый подход к синтезу данных
+ модель языка**

Awni Hannun «Deep Speech: Scaling up end-to-end speech recognition», 2014 // <https://arxiv.org/abs/1412.5567>

ASR Baidu: DeepSpeech

Обучающая выборка:
(произнесение в спектрограмме, метка-текст)

Dataset	Type	Hours	Speakers
WSJ	read	80	280
Switchboard	conversational	300	4000
Fisher	conversational	2000	23000
Baidu	read	5000	9600

Table 2: A summary of the datasets used to train Deep Speech. The Wall Street Journal, Switchboard and Fisher [3] corpora are all published by the Linguistic Data Consortium.

«Lombard Effect» – говорящий меняет громкость/тембр, чтобы подавить шум (редко бывает в датасетах, записанных в спокойной обстановке)

ASR Baidu: DeepSpeech

Архитектура: очень простая, нет LSTM

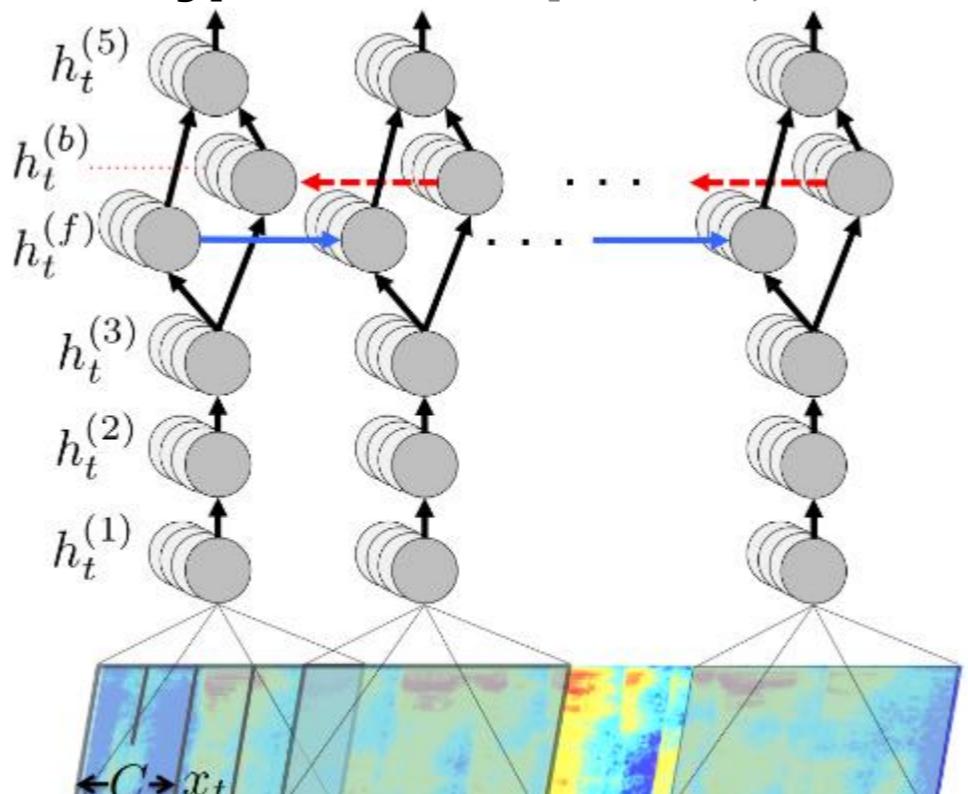


Figure 1: Structure of our RNN model and notation.

**RNN 5 слоёв = 3 обычных (clipped ReLu) + 1 двусторонний рекуррентный + 1 обычный (на вход сумма выходов 4го)
выход = softmax для алфавита**

ASR Baidu: DeepSpeech

обучение

- Nesterov's Accelerated gradient method
- dropout 0.05 – 0.1 (не в рекуррентности)
- небольшие смещения сигнала по времени

(потом усреднение вероятностей по всем смещениям)

~ аналог ансамбля (и при обучении и при teste)

функция ошибки: CTC (Connectionist Temporal Classification)

$$\mathcal{L}(x, y; \theta) = -\log \sum_{\ell \in \text{Align}(x, y)} \prod_t^T p_{\text{ctc}}(\ell_t | x; \theta).$$

where $\text{Align}(x, y)$ is the set of all possible alignments of the characters of the transcription y to frames of input x under the CTC operator.

<https://distill.pub/2017/ctc/>

ASR Baidu: DeepSpeech

модель языка (language model)

– для исправления неверно распознанных символов

RNN output	Decoded Transcription
what is the weather like in bostin right now	what is the weather like in boston right now
prime miniter nerenern modi	prime minister narendra modi
arther n tickets for the game	are there any tickets for the game

Table 1: Examples of transcriptions directly from the RNN (left) with errors that are fixed by addition of a language model (right).

N-граммная модель на 220 млн. фразах, 495000 слов

$$Q(c) = \log(\mathbb{P}(c|x)) + \alpha \log(\mathbb{P}_{lm}(c)) + \beta \text{word_count}(c)$$

лучевой поиск для максимизации

ASR Baidu: DeepSpeech

Оптимизация

очень много трюков, чтобы всё считалось быстрее...

1. Параллелизация в обработке данных

вместо одного набора признаков в момент t обрабатываем несколько в момент t

$$Wh \rightarrow W[h_1, \dots, h_k]$$

выборку отсортировали по продолжительности

батч = сигналы с примерно равной длиной
(дополнили чуть тишиной)

2. Параллелизация в модели

Например, разные направления RNN считать параллельно

3. Strides в RNN

1й слой – temporal convolution + stride parameter

ASR Baidu: DeepSpeech

System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	6.56	19.06	11.85

Table 4: Results (%WER) for 5 systems evaluated on the original audio. Scores are reported *only* for utterances with predictions given by all systems. The number in parentheses next to each dataset, e.g. Clean (94), is the number of utterances scored.

ASR Baidu: DeepSpeech2

«Baidu Research», 34 соавтора

по сравнению с DeepSpeech скорость 7x

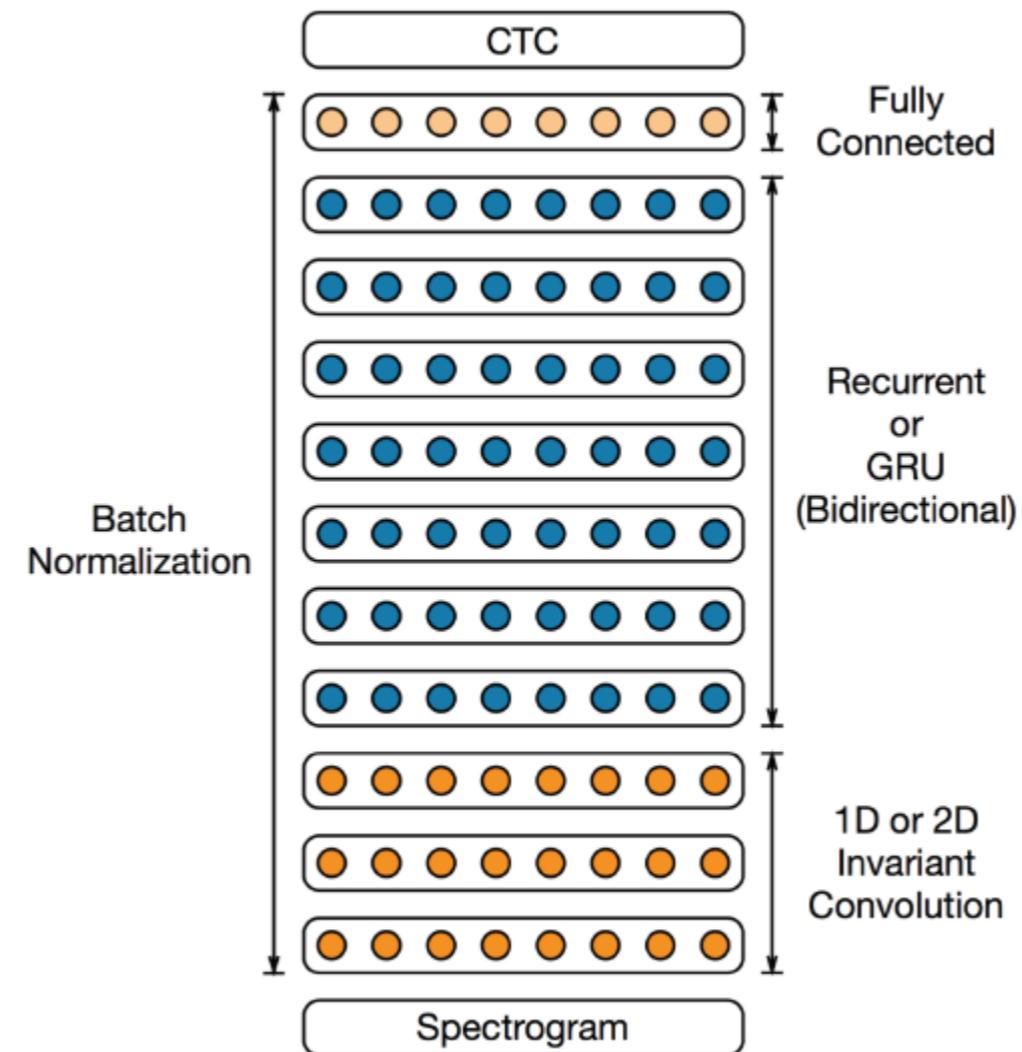
11 слоёв, много рекуррентных и свёрточных
BN для RNN
«SortaGrad»
long strides

интересно: иногда лучше односторонние RNN

Dario Amodei и др. «Deep Speech 2: End-to-End Speech Recognition in English and Mandarin», 2015 //
<https://arxiv.org/abs/1512.02595v1>

ASR Baidu: DeepSpeech2

Архитектура только «разрослась», суть DS сохранилась...



ASR Baidu: DeepSpeech2

Эксперименты

Architecture	Hidden Units	Train		Dev	
		Baseline	BatchNorm	Baseline	BatchNorm
1 RNN, 5 total	2400	10.55	11.99	13.55	14.40
3 RNN, 5 total	1880	9.55	8.29	11.61	10.56
5 RNN, 7 total	1510	8.59	7.61	10.77	9.78
7 RNN, 9 total	1280	8.76	7.68	10.83	9.52

Table 1: Comparison of WER on a training and development set for various depths of RNN, with and without BatchNorm. The number of parameters is kept constant as the depth increases, thus the number of hidden units per layer decreases. All networks have 38 million parameters. The architecture “M RNN, N total” implies 1 layer of 1D convolution at the input, M consecutive bidirectional RNN layers, and the rest as fully-connected layers with N total layers in the network.

Лучше BN делать так:

$$\vec{h}_t^l = f(\mathcal{B}(W^l h_t^{l-1}) + \vec{U}^l \vec{h}_{t-1}^l)$$

(до нелинейности и ещё на одном из слагаемых)

ASR Baidu: DeepSpeech2

специальная техника оптимизации «SortaGrad»

**первая эпоха – минибатчи в порядке возрастания длины самой длинной записи в
минибатче**

это почему-то помогает...

RNNs vs GRUs

Architecture	Simple RNN	GRU
5 layers, 1 Recurrent	14.40	10.53
5 layers, 3 Recurrent	10.56	8.00
7 layers, 5 Recurrent	9.78	7.79
9 layers, 7 Recurrent	9.52	8.19

Table 3: Comparison of development set WER for networks with either simple RNN or GRU, for various depths. All models have batch normalization, one layer of 1D-invariant convolution, and approximately 38 million parameters.

ASR Baidu: Свёртки

Эксперименты со свёртками по времени и времени-частоте (2D, в спектрограммах)

Architecture	Channels	Filter dimension	Stride	Regular Dev	Noisy Dev
1-layer 1D	1280	11	2	9.52	19.36
2-layer 1D	640, 640	5, 5	1, 2	9.67	19.21
3-layer 1D	512, 512, 512	5, 5, 5	1, 1, 2	9.20	20.22
1-layer 2D	32	41x11	2x2	8.94	16.22
2-layer 2D	32, 32	41x11, 21x11	2x2, 2x1	9.06	15.71
3-layer 2D	32, 32, 96	41x11, 21x11, 21x11	2x2, 2x1, 2x1	8.61	14.74

Table 4: Comparison of WER for various arrangements of convolutional layers. In all cases, the convolutions are followed by 7 recurrent layers and 1 fully connected layer. For 2D-invariant convolutions the first dimension is frequency and the second dimension is time. All models have BatchNorm, SortaGrad, and 35 million parameters.

ASR Baidu: приём для использования односторонних RNN

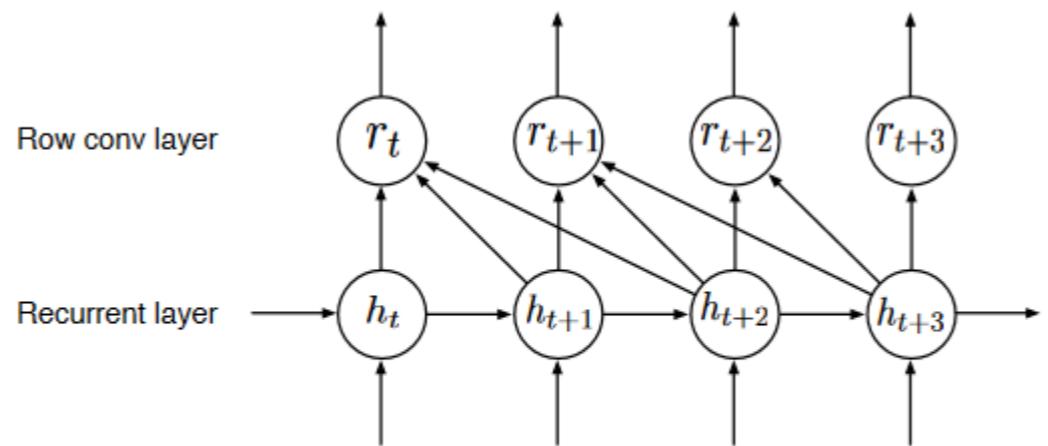


Figure 3: Row convolution architecture with future context size of 2

RNN → свёртки

Из будущего надо немножко информации...

хорошая ссылка: <http://commoncrawl.org/>

ASR Baidu: + модель языка

Language	Architecture	Dev no LM	Dev LM
English	5-layer, 1 RNN	27.79	14.39
English	9-layer, 7 RNN	14.93	9.52
Mandarin	5-layer, 1 RNN	9.80	7.13
Mandarin	9-layer, 7 RNN	7.55	5.81

Table 6: Comparison of WER for English and CER for Mandarin with and without a language model. These are simple RNN models with only one layer of 1D invariant convolution.

Оптимизация

- **deep learning library C++**
- **high-performance linear algebra library CUDA&C++**
- **8 Titan X GPUs per node (минибатч 612)**
- **synchronous SGD**
- **GPU implementation of the CTC loss function**

ASR Baidu: Датасет

Dataset	Speech Type	Hours
WSJ	read	80
Switchboard	conversational	300
Fisher	conversational	2000
LibriSpeech	read	960
Baidu	read	5000
Baidu	mixed	3600
Total		11940

Table 9: Summary of the datasets used to train DS2 in English. The Wall Street Journal (WSJ), Switchboard and Fisher [13] corpora are all published by the Linguistic Data Consortium. The LibriSpeech dataset [46] is available free on-line. The other datasets are internal Baidu corpora.

ASR Baidu: Результаты

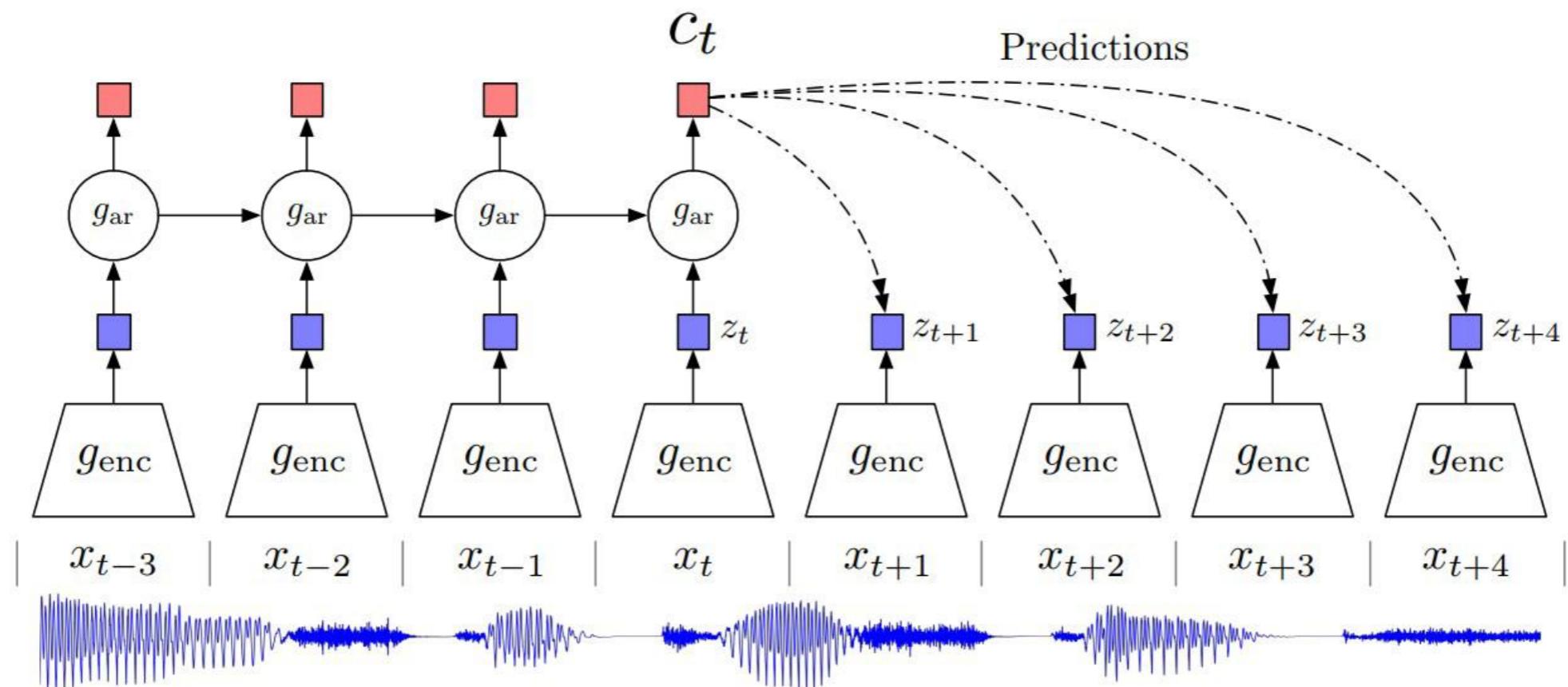
Model size	Model type	Regular Dev	Noisy Dev
18×10^6	GRU	10.59	21.38
38×10^6	GRU	9.06	17.07
70×10^6	GRU	8.54	15.98
70×10^6	RNN	8.44	15.09
100×10^6	GRU	7.78	14.17
100×10^6	RNN	7.73	13.06

Table 11: Comparing the effect of model size on the WER of the English speech system on both the regular and noisy development sets. We vary the number of hidden units in all but the convolutional layers. The GRU model has 3 layers of bidirectional GRUs with 1 layer of 2D-invariant convolution. The RNN model has 7 layers of bidirectional simple recurrence with 3 layers of 2D-invariant convolution. Both models output bigrams with a temporal stride of 3. All models contain approximately 35 million parameters and are trained with BatchNorm and SortaGrad.

Test set	DS1	DS2
Baidu Test	24.01	13.59

Table 12: Comparison of DS1 and DS2 WER on an internal test set of 3,300 examples. The test set contains a wide variety of speech including accents, low signal-to-noise speech, spontaneous and conversational speech.

wav2vec: Unsupervised Pre-training for Speech Recognition



Идея: Contrastive Prediction Coding

$$\mathcal{L}_k = - \sum_i \left(\log \sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda \mathbb{E}_{\tilde{\mathbf{z}} \sim p_n} [\log \sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))] \right)$$

<https://arxiv.org/abs/1904.05862v1>

Аугментация

Добавление случайного шума

Добавление фонового шума, фоновой речи

Масштабирование по частотам, по времени

SpecAugment

Три типа деформаций:

- деформация по времени;**
- маскирование блоков по частотам;**
- маскирование блоков по временным шагам.**

SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition

<https://arxiv.org/pdf/1904.08779.pdf>

Аугментация

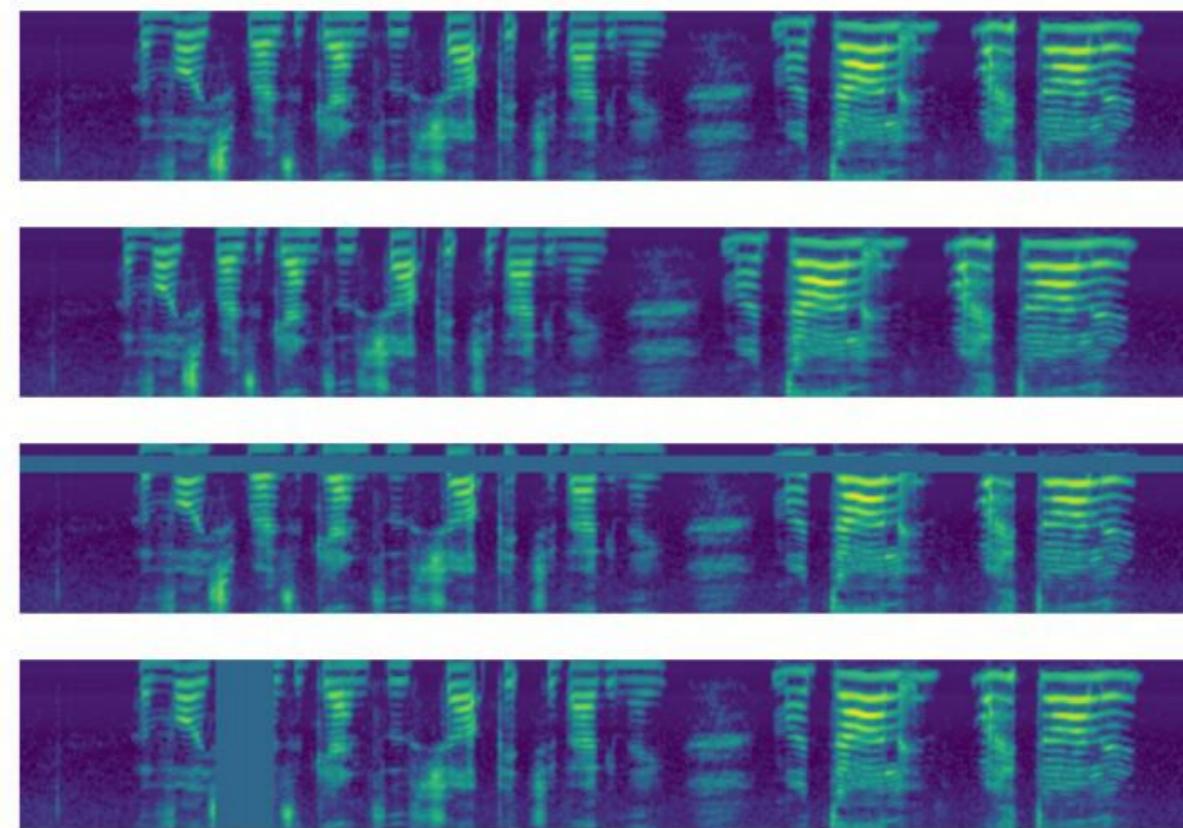


Figure 1: *Augmentations applied to the base input, given at the top. From top to bottom, the figures depict the log mel spectrogram of the base input with no augmentation, time warp, frequency masking and time masking applied.*

Аугментация

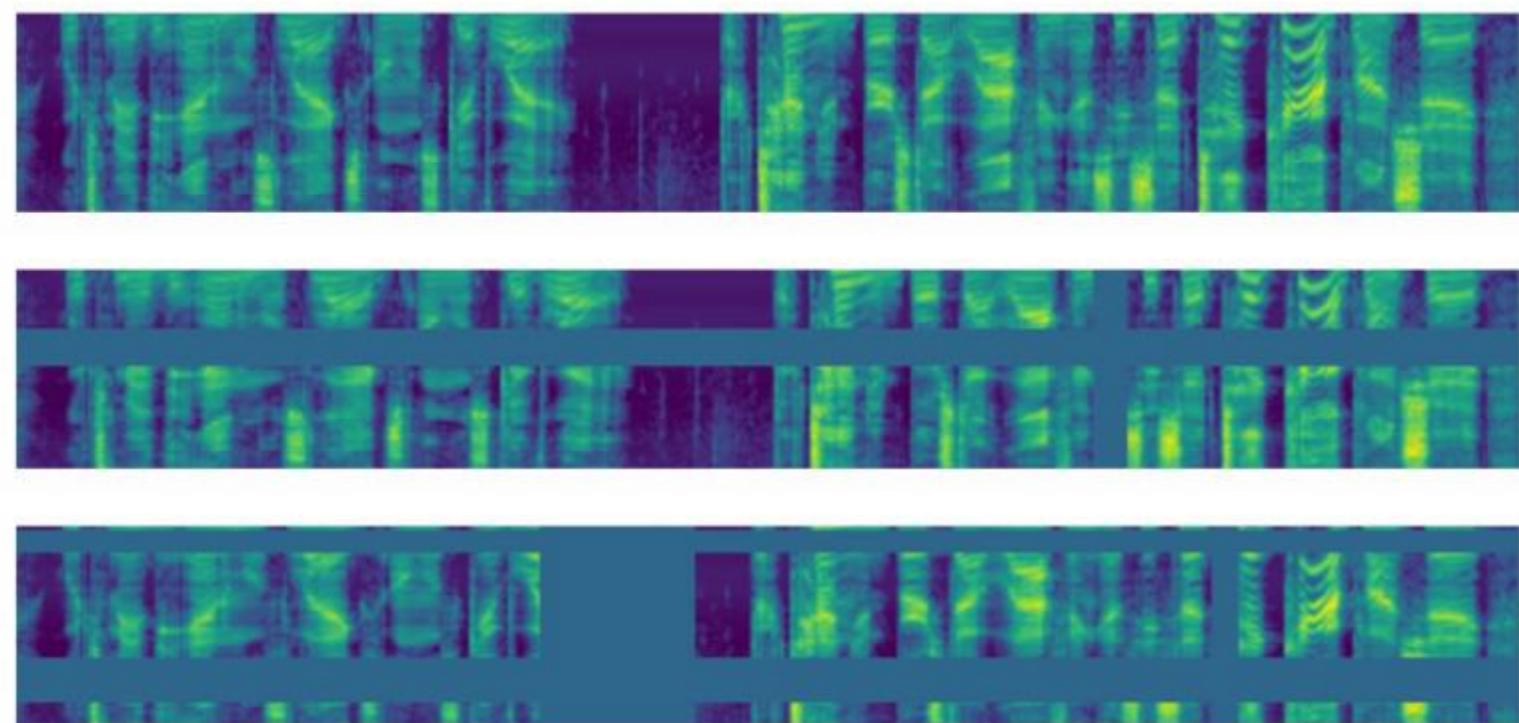


Figure 2: Augmentation policies applied to the base input. From top to bottom, the figures depict the log mel spectrogram of the base input with policies None, LB and LD applied.

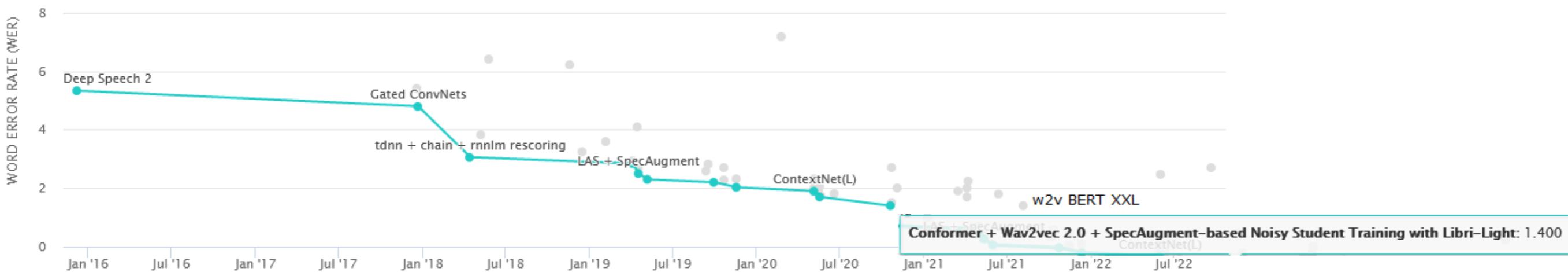
Аугментация

Table 5: *Switchboard 300h WERs (%)*.

Method	No LM		With LM	
	SWBD	CH	SWBD	CH
HMM				
Veselý et al., (2013) [40]			12.9	24.5
Povey et al., (2016) [29]			9.6	19.3
Hadian et al., (2018) [41]			9.3	18.9
Zeyer et al., (2018) [23]			8.3	17.3
CTC				
Zweig et al., (2017) [42]	24.7	37.1	14.0	25.3
Audhkhasi et al., (2018) [43]	20.8	30.4		
Audhkhasi et al., (2018) [44]	14.6	23.6		
LAS				
Lu et al., (2016) [45]	26.8	48.2	25.8	46.0
Toshniwal et al., (2017) [46]	23.1	40.8		
Zeyer et al., (2018) [23]	13.1	26.1	11.8	25.7
Weng et al., (2018) [47]	12.2	23.3		
Zeyer et al., (2018) [37]	11.9	23.7	11.0	23.1
Our Work				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	7.2	14.6	6.8	14.1
LAS + SpecAugment (SS)	7.3	14.4	7.1	14.0

Speech Recognition on LibriSpeech test-clean

≡



Ссылки

по слайдам Антон Бахтин (Facebook Research)

[www.machinelearning.ru/wiki/index.php?title=Методы_анализа_текстов_\(семинар%2C_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Методы_анализа_текстов_(семинар%2C_К.В.Воронцов))

Karen Livescu «Speech Processing» // https://github.com/mlss-2019/slides/tree/master/speech_processing

Kaldi:

<http://mi.eng.cam.ac.uk/~sjy/papers/gayo07.pdf>