Глубинное обучение для работы со звуком

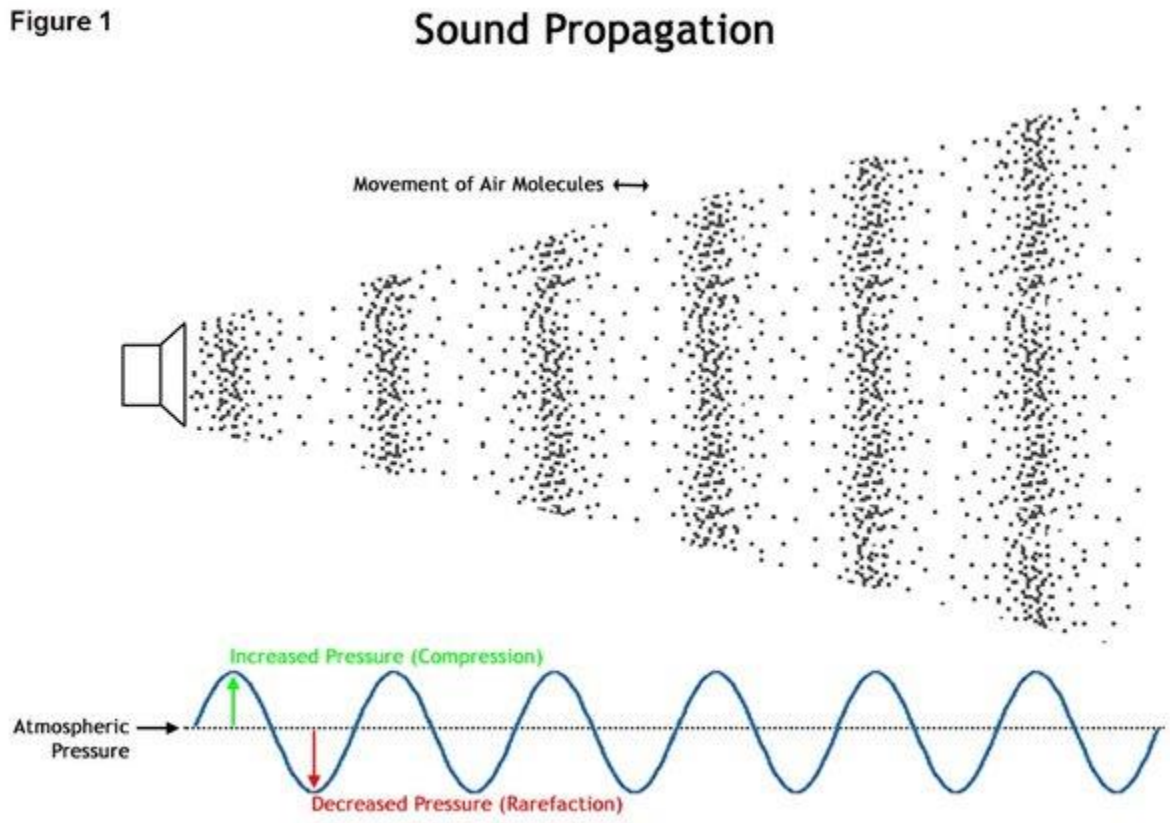
1

Как представляется звук

Даниил Пятько

Звук

Звук получается из-за перепадов давления в воздухе.
Из этого сразу можно получить какое-то представление звука.



Звук – первое представление

- Представление: зависимость звукового давления от времени
- Звуковое давление – избыточное давление, возникающее в среде при прохождении звуковой волны

$$p = p_{total} - p_{atmospheric}$$

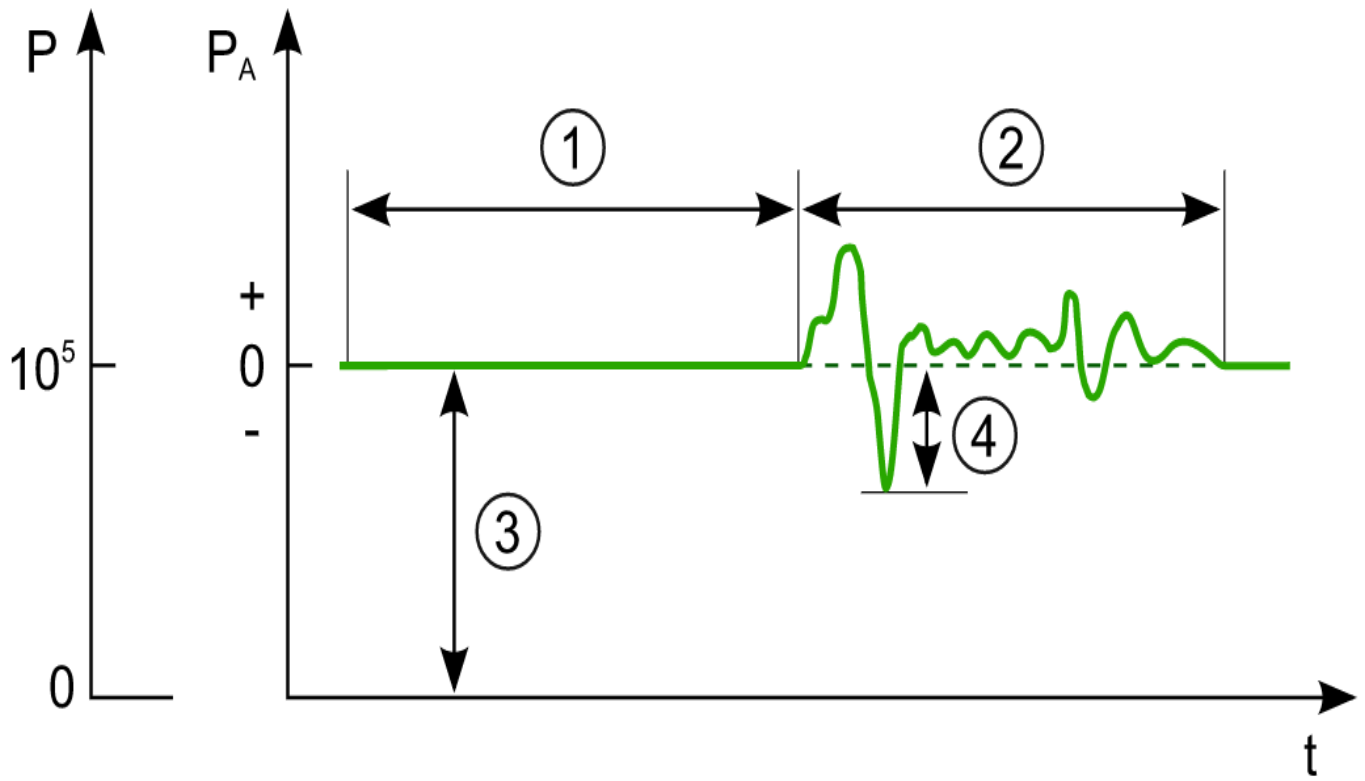
На картинке:

1 – тишина

2 – есть звук

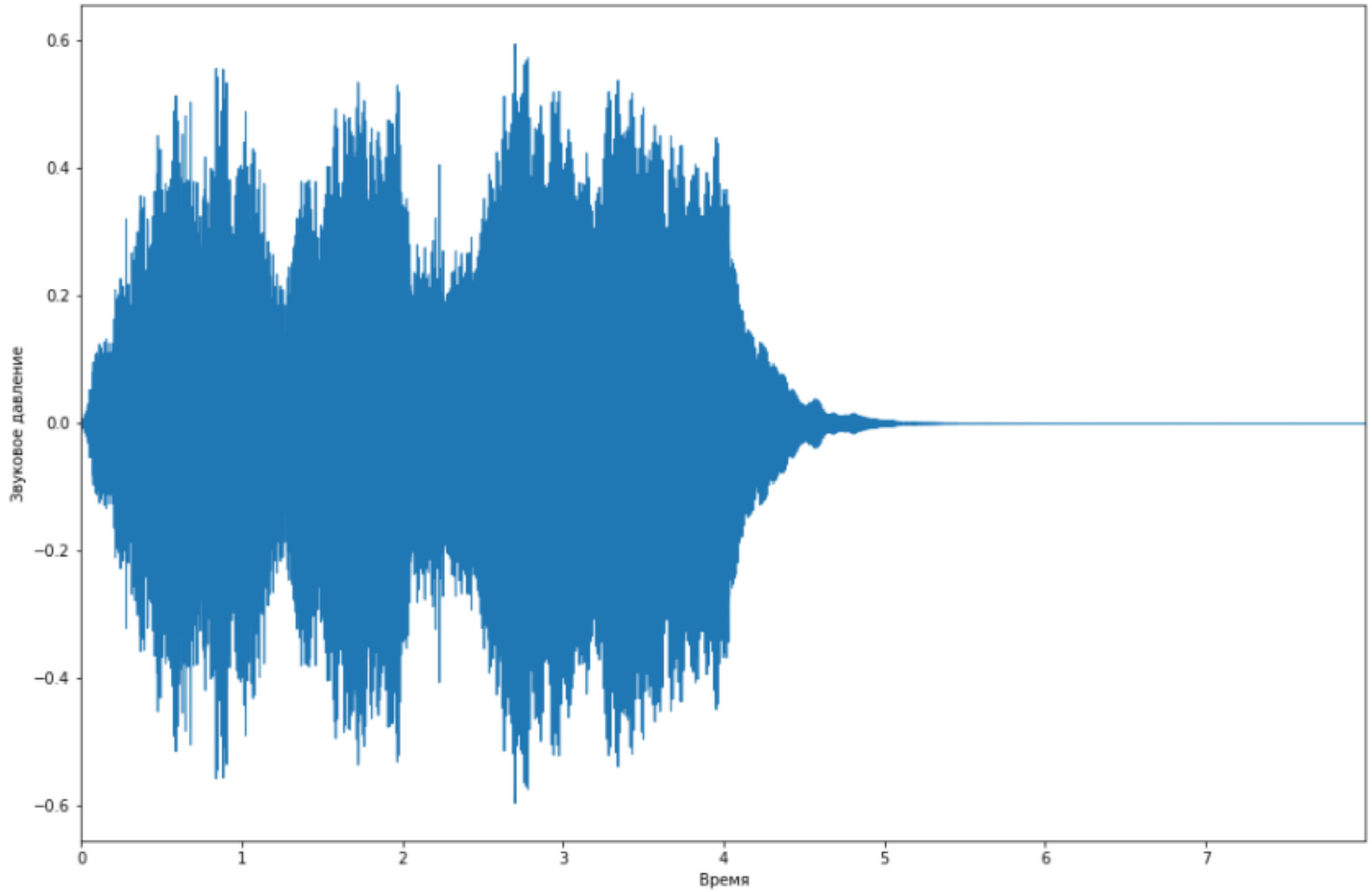
3 – нормальное атмосферное давление

4 – звуковое давление



Звук – первое представление

Чем больше по модулю звуковое давление, тем больше воспринимаемая громкость звука



Волны – синусоиды

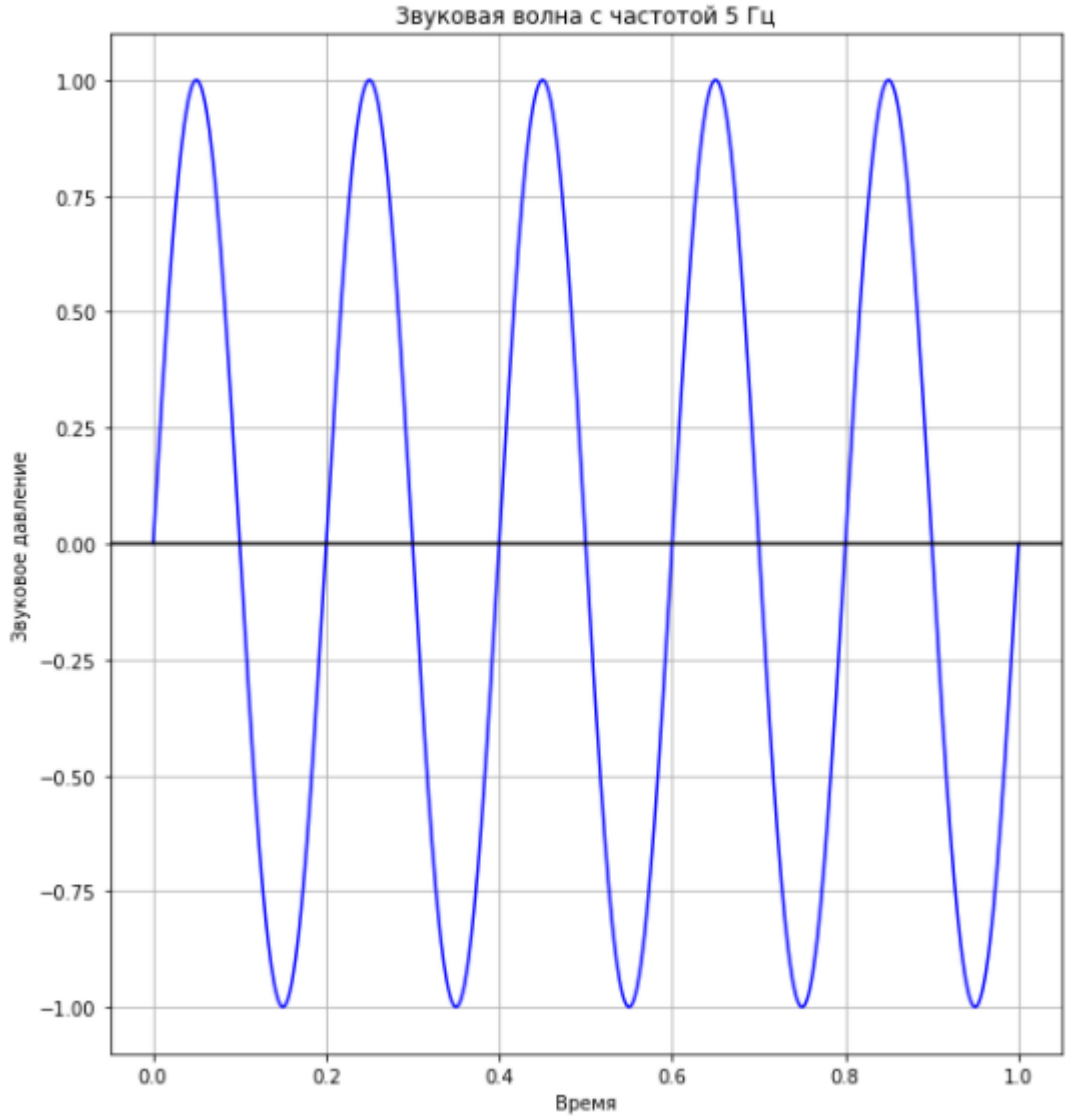
Синусоида – самая простая звуковая волна

$$y(t) = a\sin(2\pi f t)$$
$$y(t) = a\cos(2\pi f t)$$

- Амплитуда (amplitude) синусоидальной волны в Па – максимальное значение звукового давления.

  Соответствует коэффициенту a
- Частота (frequency) синусоидальной волны в Гц – число полных колебаний за секунду.

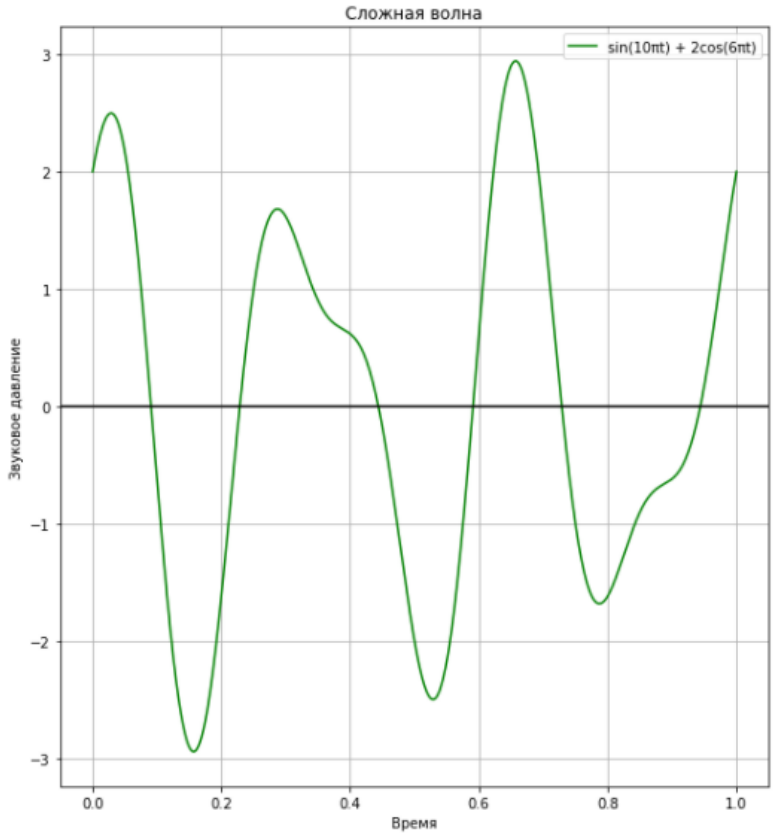
  Соответствует коэффициенту f

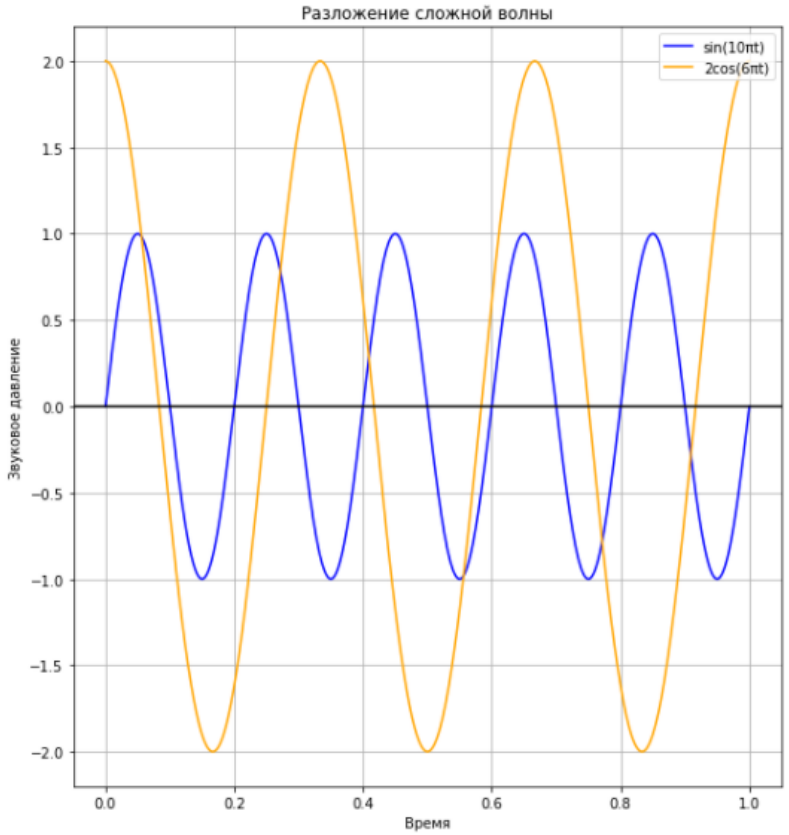


Волны – сложные волны

Волны можно представлять как сумму синусоид с определенными частотами и амплитудами

Сложные волны (complex waves) – волны, которые состоят из хотя бы 2 синусоид





Разложение сложных волн

- Человеческое ухо может распознавать не только уровень звукового давления, но и какие частоты и амплитуды у синусоид, из которых состоит звук. Частоты определяют высоту звука.
- На данный момент мы имеем только зависимость звукового давления от времени.
- Хотим уметь раскладывать сложные волны на синусоиды, чтобы получать частоты.

Discrete Fourier Transform

Пусть есть $X(n)$ — равномерно замерянный сигнал в N моментах времени.
Хотим разложить его в $2N$ дискретных синусоид вида

$$\left\{a(k)\sin\left(2\pi k\frac{n}{N}\right)\right\}_{n=0}^{N-1},\ \left\{b(k)\cos\left(2\pi k\frac{n}{N}\right)\right\}_{n=0}^{N-1},\ k\in\{0,\dots,N-1\}$$

Для фиксированного k: $a(k), b(k)$ — амплитуды, $k\cdot\frac{SR}{N}$Гц частота.
Давайте заменим $a(k), b(k)\in\mathbb{R}$ на одно число $S(k)\in\mathbb{C}$ и запишем в немного другом виде.

$$X(n)=\sum_{k=0}^{N-1}\frac{-\operatorname{Im}(S(k))}{N}\sin\left(\frac{2\pi kn}{N}\right)+\sum_{k=0}^{N-1}\frac{\operatorname{Re}(S(k))}{N}\cos\left(\frac{2\pi kn}{N}\right)$$

DFT – Подсчет коэффициентов

Тогда $S(k)$ можно найти следующим образом:

$$S(k) = \sum_{n=0}^{N-1} X(n) e^{-2\pi i k \frac{n}{N}}$$

$$S(k) = \sum_{n=0}^{N-1} X(n) \cos\left(2\pi k \frac{n}{N}\right) - X(n) i \sin\left(2\pi k \frac{n}{N}\right)$$

$$S(k) = \sum_{n=0}^{N-1} X(n) \cos\left(2\pi k \frac{n}{N}\right) - i \sum_{n=0}^{N-1} X(n) \sin\left(2\pi k \frac{n}{N}\right)$$

$S(k)$ — преобразование Фурье последовательности $X(n)$.

DFT – Интуиция

Пусть есть два сигнала: $X(n), Y(n), n \in \{0, \dots, N-1\}$ со средним 0. Как определить их схожесть? Корреляция:

$$\sum_{n=0}^{N-1} X(n)Y(n)$$

Вернемся к формуле для $S(k)$:

$$S(k) = \underbrace{\sum_{n=0}^{N-1} X(n) \cos\left(2\pi k \frac{n}{N}\right)}_{\text{корреляция с косинусом}} - i \underbrace{\sum_{n=0}^{N-1} X(n) \sin\left(2\pi k \frac{n}{N}\right)}_{\text{корреляция с синусом}}$$

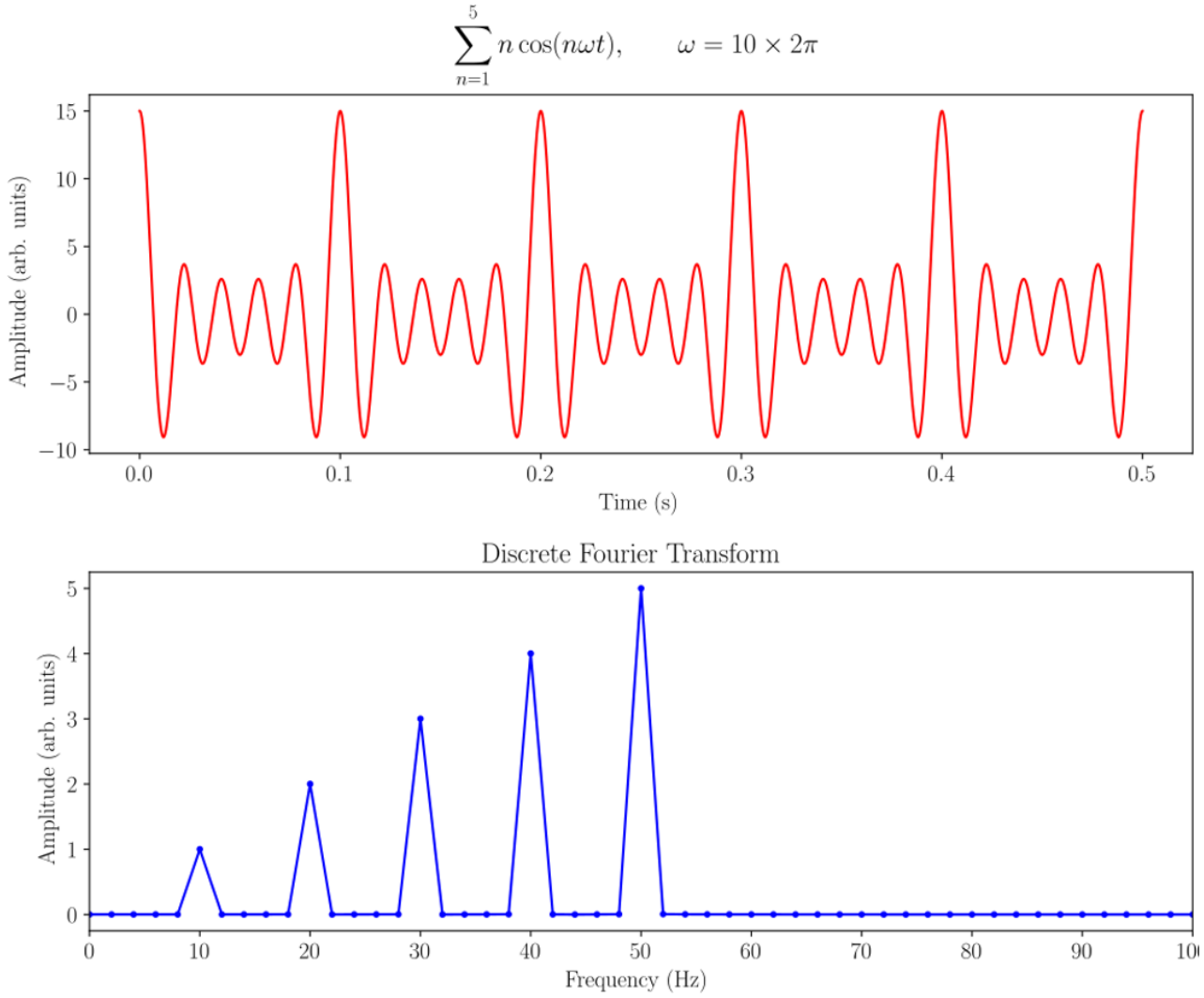
Вспомним как представлялся X:

$$X(n) = \sum_{k=0}^{N-1} \frac{-\operatorname{Im}(S(k))}{N} \sin\left(\frac{2\pi k n}{N}\right) + \sum_{k=0}^{N-1} \frac{\operatorname{Re}(S(k))}{N} \cos\left(\frac{2\pi k n}{N}\right)$$

Итого:
$\frac{|S(k)|^2}{N}$ — насколько сигнал коррелирует с косинусом и синусом определенной частоты.

DFT – Новое представление звука

- Давайте построим график $\frac{|S(k)|^2}{N}$ для сложной волны

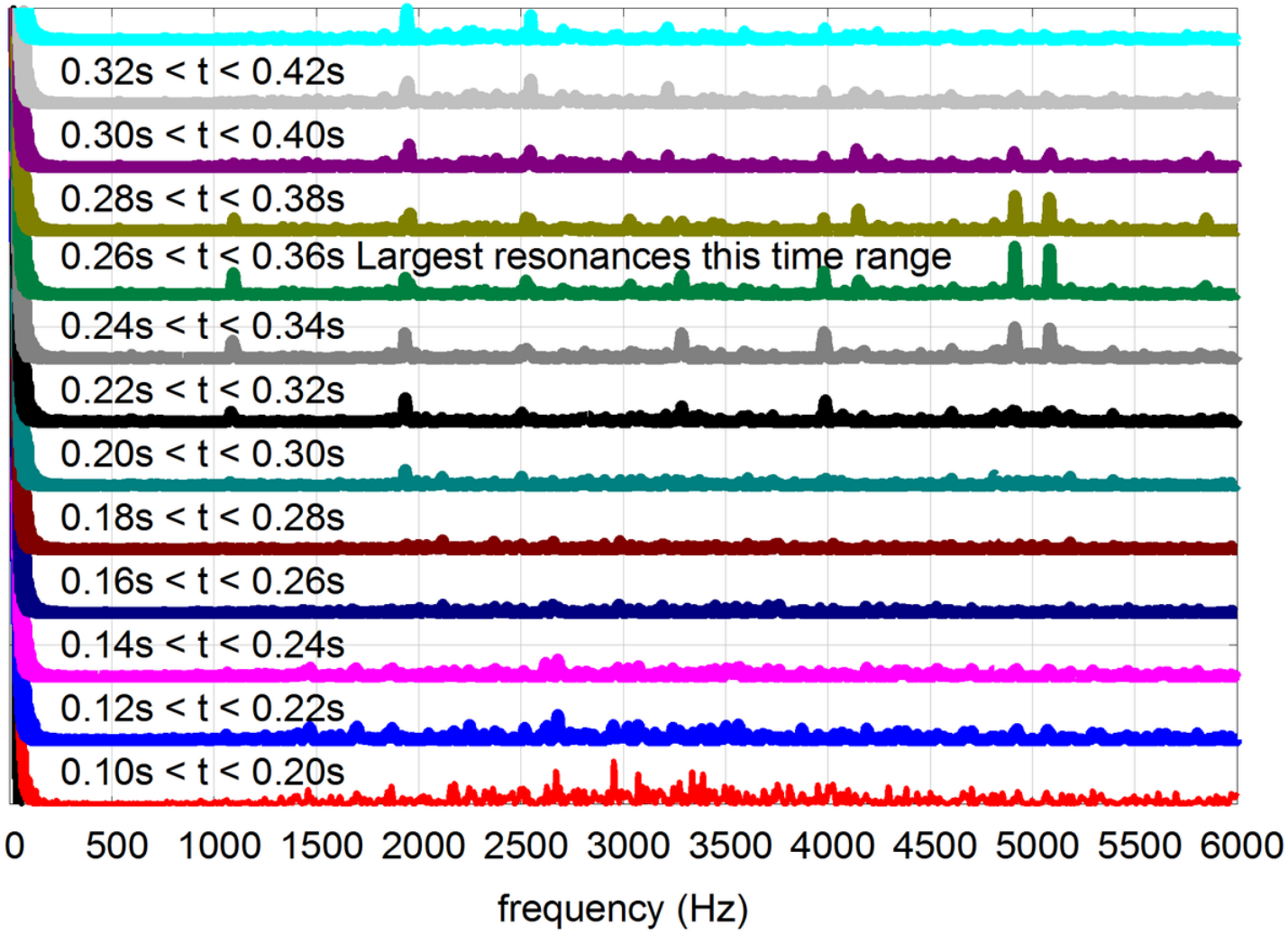


DFT – проблемы наивного подхода

- Подсчет за $O(N^2)$. Решение: FFT за $O(N \log N)$
- На самом деле, мы можем корректно применять DFT только если каждая синусоида проходит целое число периодов. В жизни это далеко не всегда так, поэтому происходят “утечки спектра”, когда добавляются лишние частоты. Решение: window functions.
- DFT раскладывает сигнал в предположении, что он состоит из синусоид, которые не меняются, в реальности это совсем не так. Решение: Short Time Fourier Transform.

Short Time Fourier Transform

- Разбиваем временной промежуток на отрезки фиксированной длины, на каждом из них строим DFT.



Уровень звукового давления

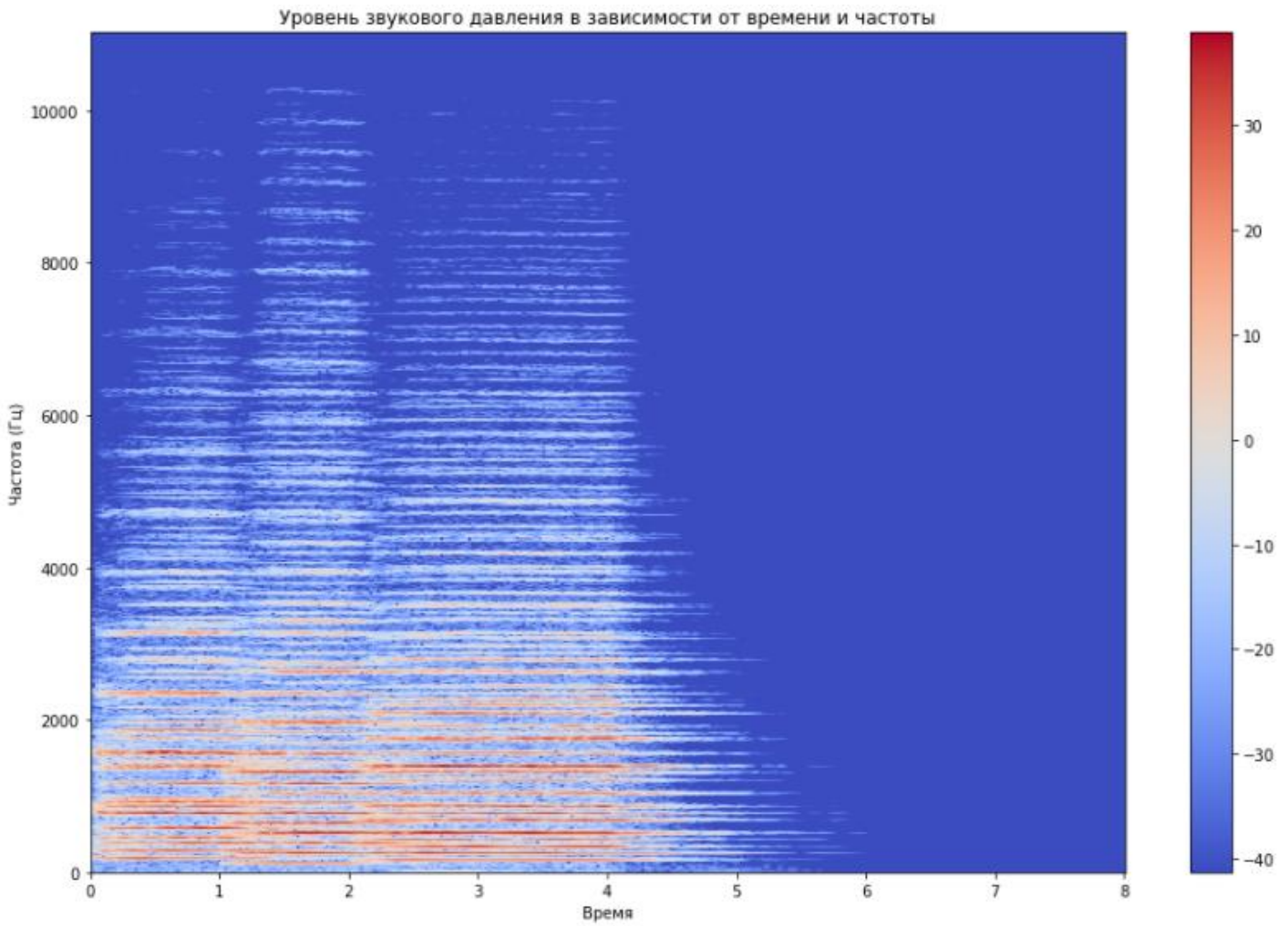
Паскали в децибелы:

$$L(p) = 20 \log_{10} \left(\frac{p}{p_0} \right)$$

p_0 — уровень, относительно которого измеряем. Обычно едва различимый звук (писк комара на расстоянии 3 метров).

STFT - спектрограмма

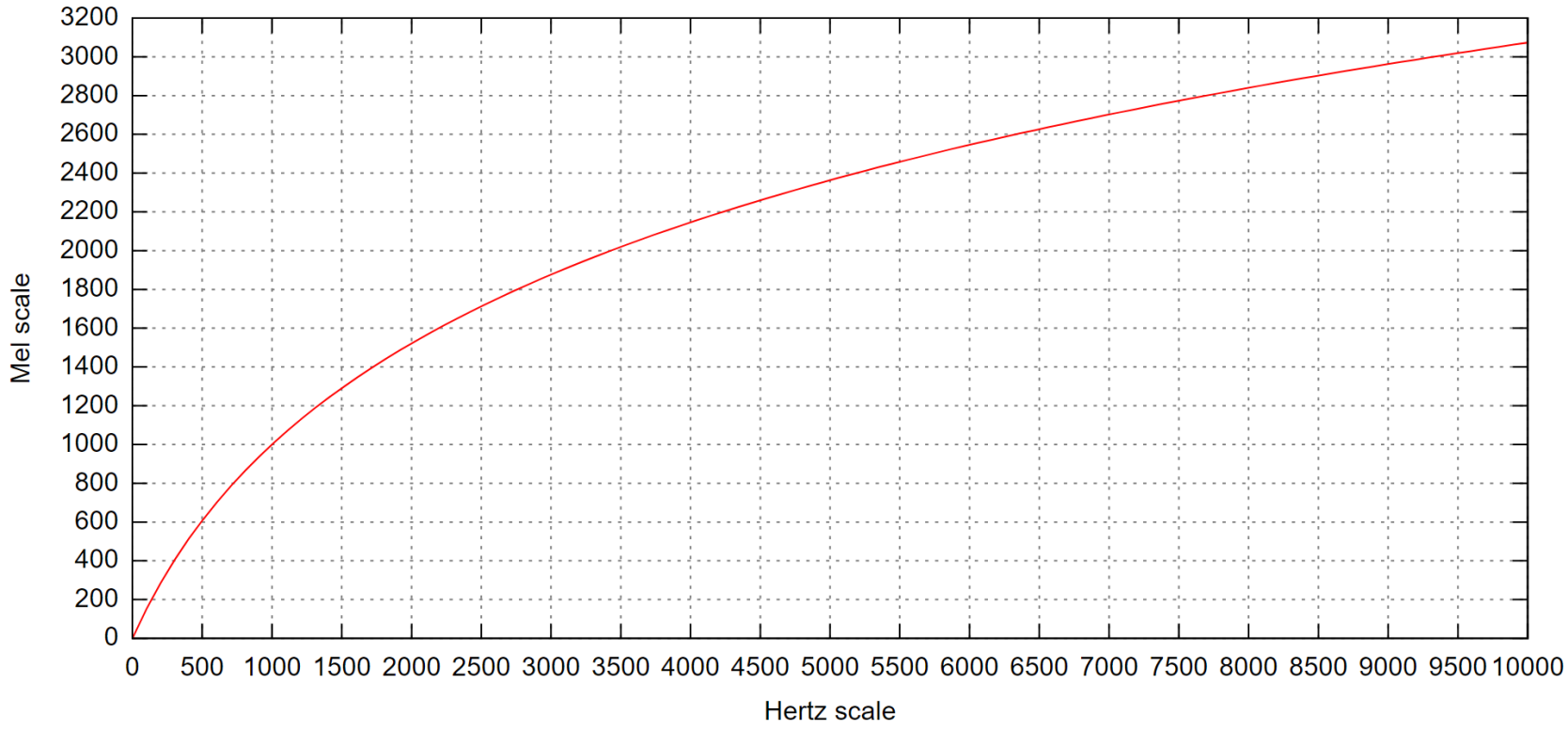
- Для фиксированного промежутка времени и частоты изображаем уровень звукового давления в децибелах



Mel Scale

Перевод из герц в мелы:

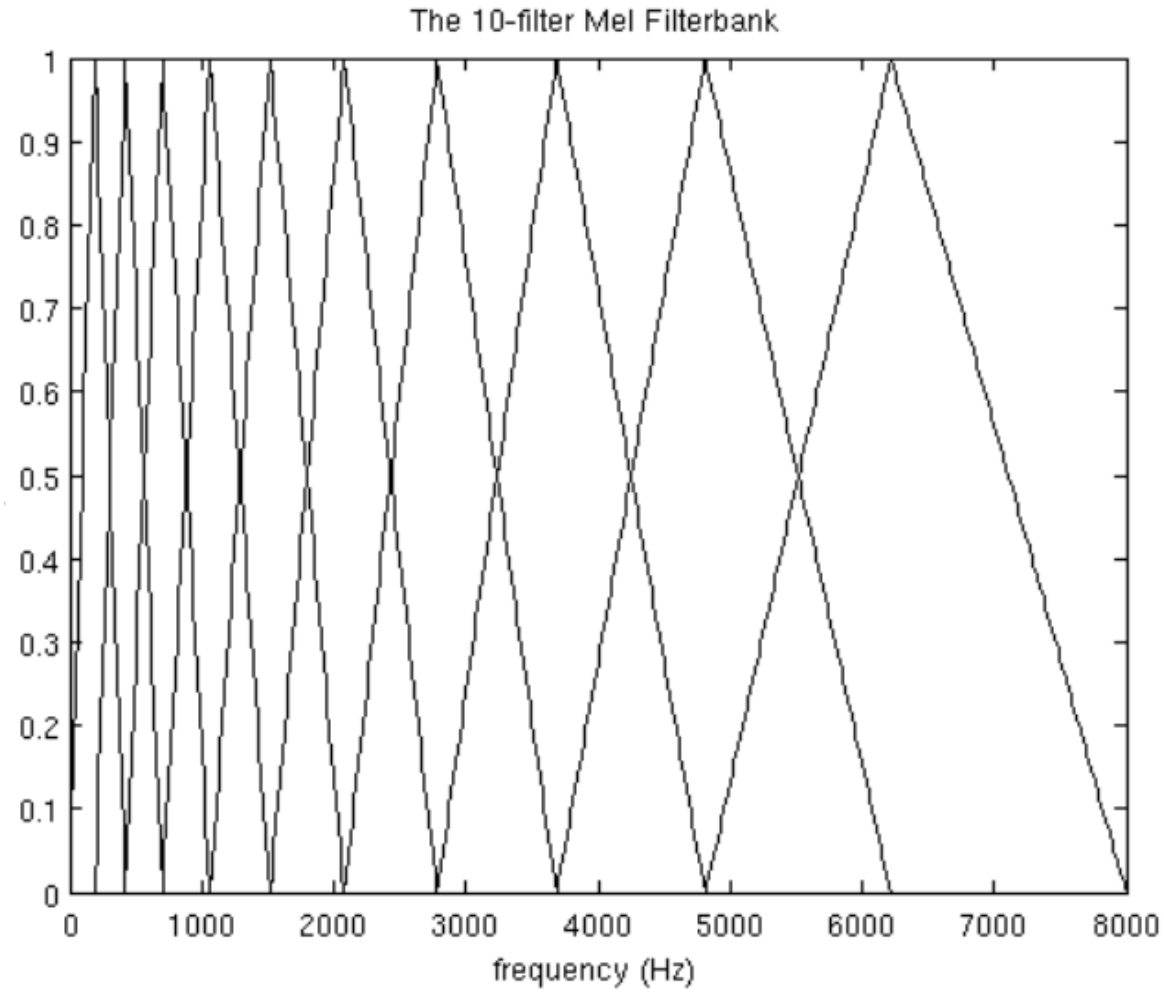
$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right)$$



Mel Filterbank

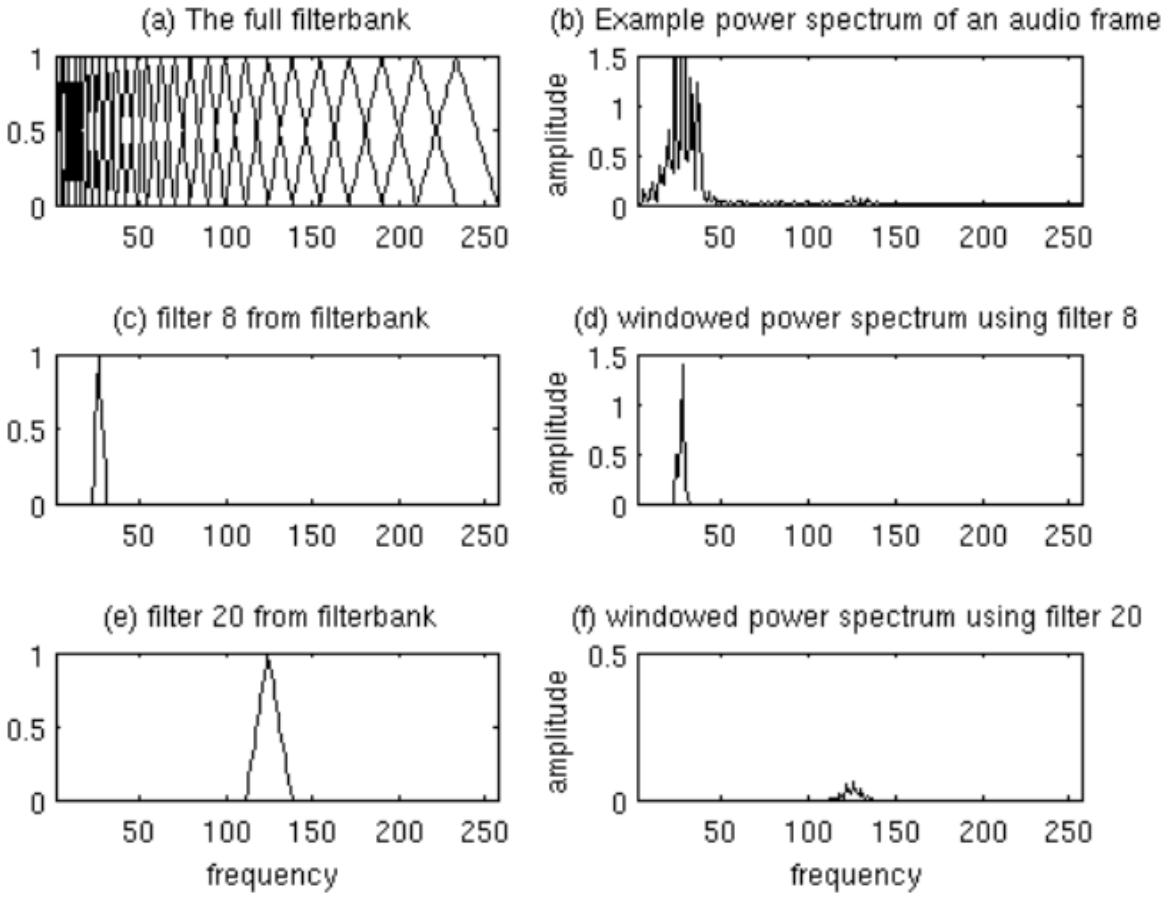
Хотим сжать частотную информацию

На mel scale равномерно выбираем точки и строим фильтры



Mel Filterbank

Для каждого фильтра из получаем одно число – энергию в нем



Источники

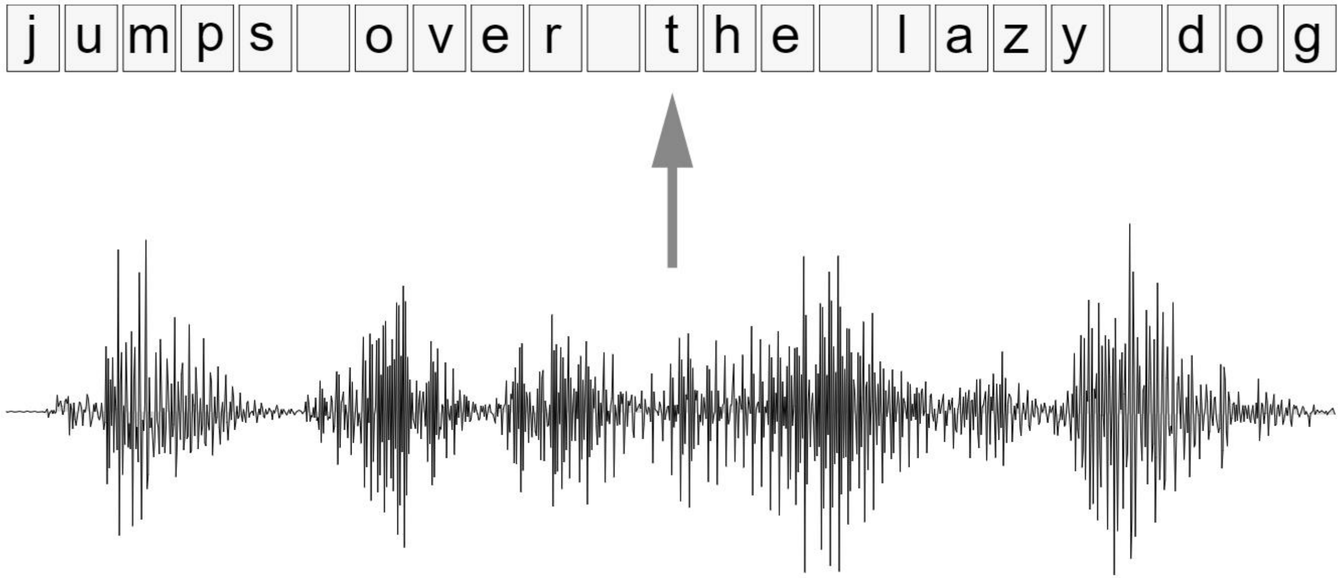
- http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
- http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/
- https://en.wikipedia.org/wiki/Sound_pressure
- https://en.wikipedia.org/wiki/Discrete_Fourier_transform
- https://en.wikipedia.org/wiki/Short-time_Fourier_transform

2

Распознавание речи

Даниил Волгин

Распознавание речи End-to-End



Хотим найти отображение входных последовательностей вида $X = [x_1, x_2, \dots, x_T]$ (зависящие от времени характеристики, полученные по аудиозаписи) в соответствующие им выходные последовательности вида $Y = [y_1, y_2, \dots, y_U]$ (расшифровка произнесённого в аудиозаписи текста).

Распознавание речи End-to-End

Проблемы:

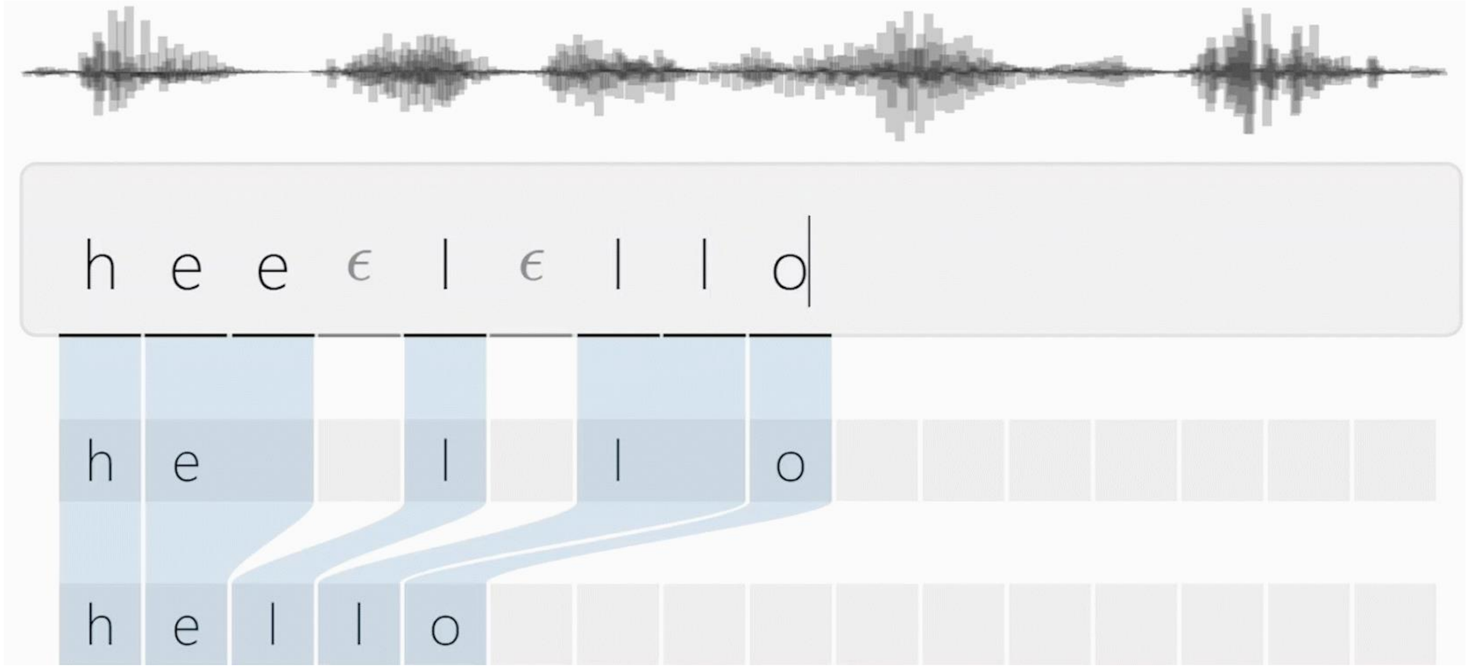
- Длина последовательности X может отличаться в разных объектах выборки (аналогично для Y)
- Соотношение длин X и Y может отличаться в разных объектах выборки
- Нет хорошего соответствия между элементами X и Y

Решает Connectionist Temporal Classification (CTC)

Alignment

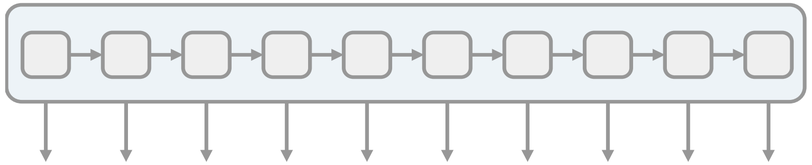
Alignment не нужен как фактор, но чтобы по входу получить распределение, CTC суммирует вероятности всех возможных alignments для каждого выхода

- Дополнительный символ ϵ (blank)
- Рассматриваем alignments такой же длины, как и вход



Отображение из alignment в выходной текст:

1. Сжимаем подряд идущие символы в один
2. Удаляем ϵ-символы



h	h	h	h	h	h	h	h	h	h
e	e	e	e	e	e	e	e	e	e
l	l	l	l	l	l	l	l	l	l
o	o	o	o	o	o	o	o	o	o
ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ

h	e	ϵ	l	l	ϵ	l	l	o	o
h	h	e	l	l	ϵ	ϵ	l	ϵ	o
ϵ	e	ϵ	l	l	ϵ	ϵ	l	o	o

h	e	l	l	o
e	l	l	o	
h	e	l	o	

Оптимизируемая метрика

Дан объект (X, Y) обучающей выборки D

1. Входная последовательность X
2. Используя, например, RNN, получаем $p_t(a \mid X)$, распределение по выходам $\{h, e, l, o, \epsilon\}$ для каждой временной метки t.
3. С помощью распределений по дополненному алфавиту считаем вероятности для различных alignments
4. Суммируя по alignments, получаем вероятности последовательностей-текстов.

Хотим максимизировать правдоподобие последовательности Y при условии входа X:

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

The CTC conditional **probability**

marginalizes over the set of valid alignments

computing the **probability** for a single alignment step-by-step.

Эффективный подсчет вероятности

Будем использовать динамическое программирование

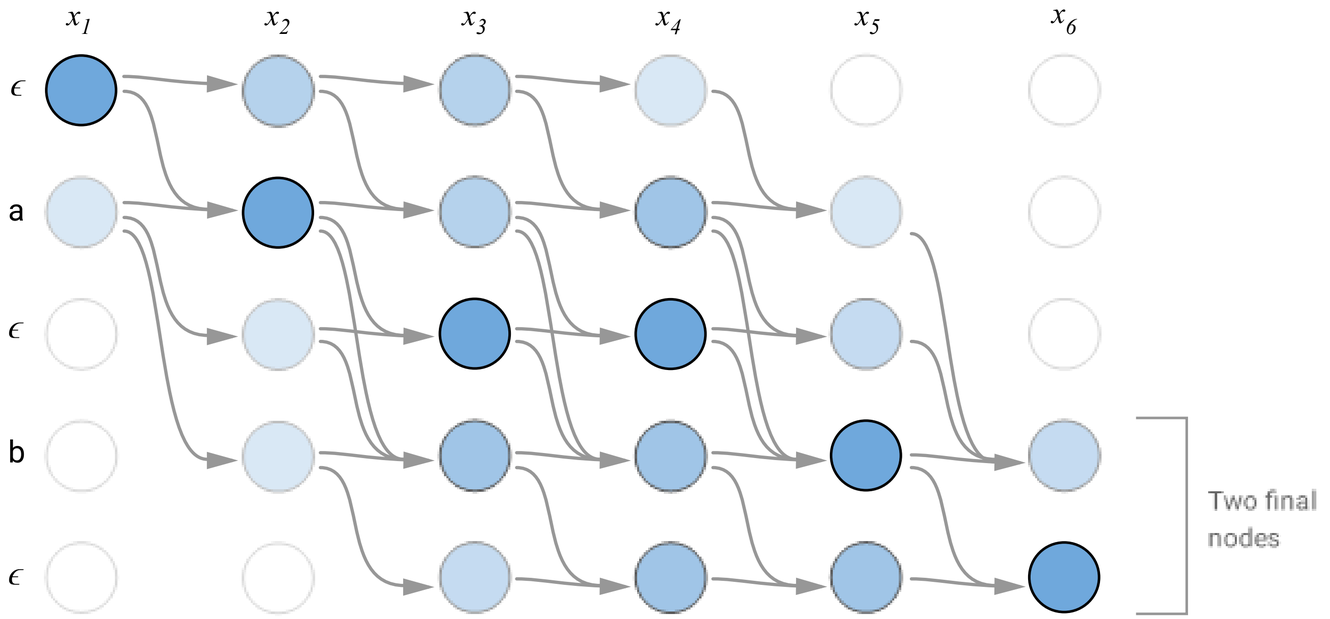
Введем

- $Z = [\epsilon, y_1, \epsilon, y_2, \dots, \epsilon, y_U, \epsilon]$
- $\alpha_{s,t}$ – вероятность получить $Z_{1:s}$ из $x_{1:t}$

Инициализируем $\alpha_{1,1}$ и $\alpha_{2,1}$ вероятностями соответствующих символов на первом ($t = 1$) шаге. Остальные $\alpha_{s,1}$ заполняем нулями

Научимся пересчитывать $\alpha_{s,t}$, зная $\alpha_{s,t-1}$

$$P(Y|X) = \alpha_{2U,T} + \alpha_{2U+1,T}$$



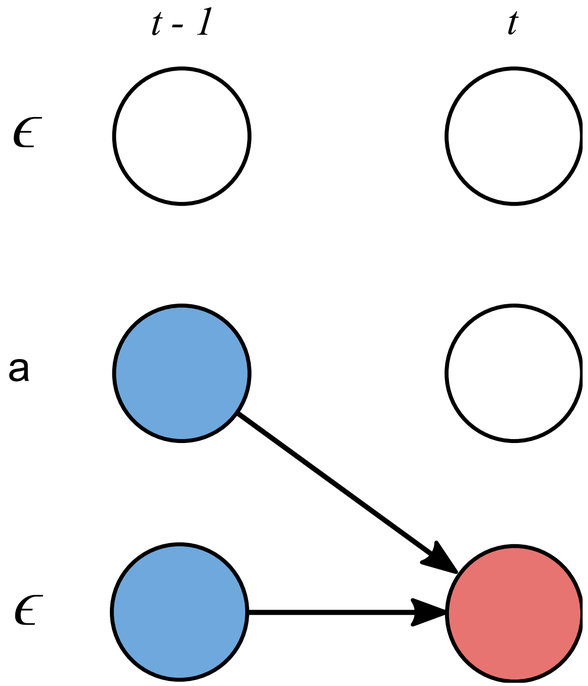
Эффективный подсчет вероятности

$$(Z_s = \epsilon) \vee (Z_s = Z_{s-2})$$

$$\alpha_{s,t} = (\alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot p_t(z_s \mid X)$$

$(\alpha_{s-1,t-1} + \alpha_{s,t-1})$: The CTC probability of the two valid subsequences after $t-1$ input steps.

$p_t(z_s \mid X)$: The probability of the current character at input step t.

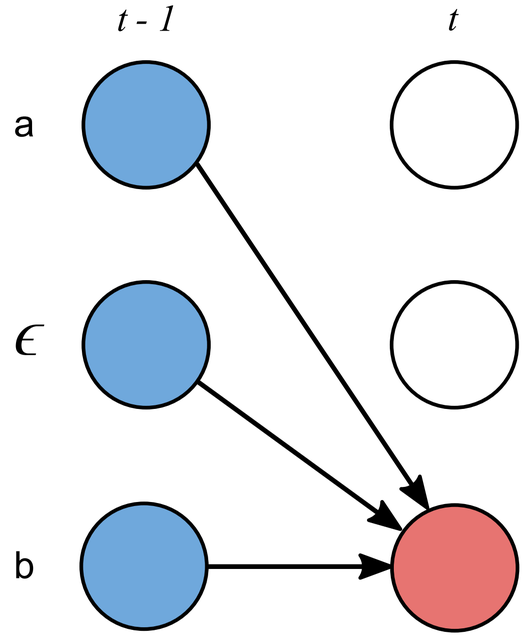


$$(Z_s \neq \epsilon) \wedge (Z_s \neq Z_{s-2})$$

$$\alpha_{s,t} = (\alpha_{s-2,t-1} + \alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot p_t(z_s \mid X)$$

$(\alpha_{s-2,t-1} + \alpha_{s-1,t-1} + \alpha_{s,t-1})$: The CTC probability of the three valid subsequences after $t-1$ input steps.

$p_t(z_s \mid X)$: The probability of the current character at input step t.



Inference

Хотим найти наиболее вероятную строку Y*, используя построенное в процессе обучения условное распределение

$$Y^* = \underset{Y}{\operatorname{argmax}}\ p(Y \mid X)$$

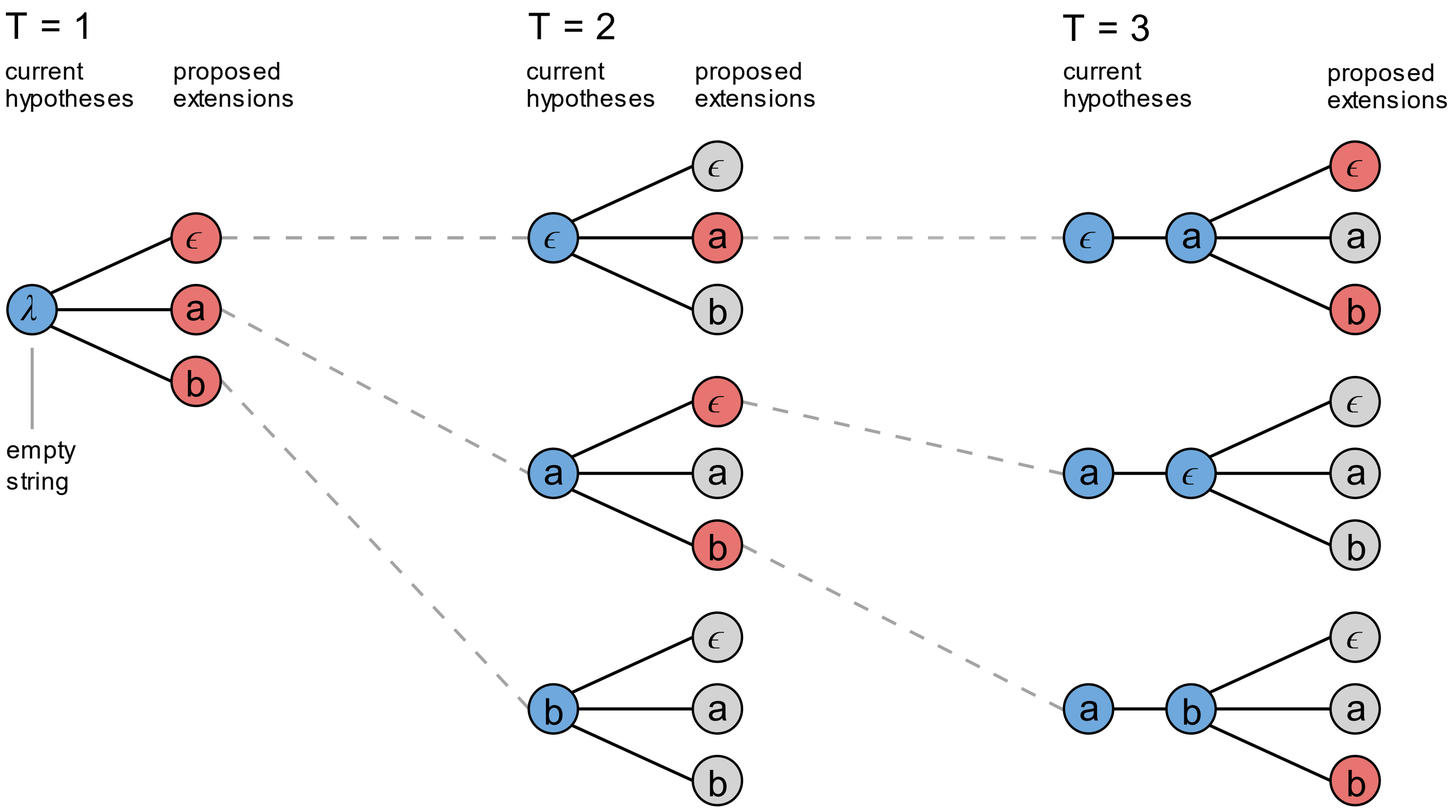
Перебирать все alignments длины T и считать их вероятности долго

«Наивная» эвристика – брать наиболее вероятный символ для каждой временной метки:

$$A^* = \underset{A}{\operatorname{argmax}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

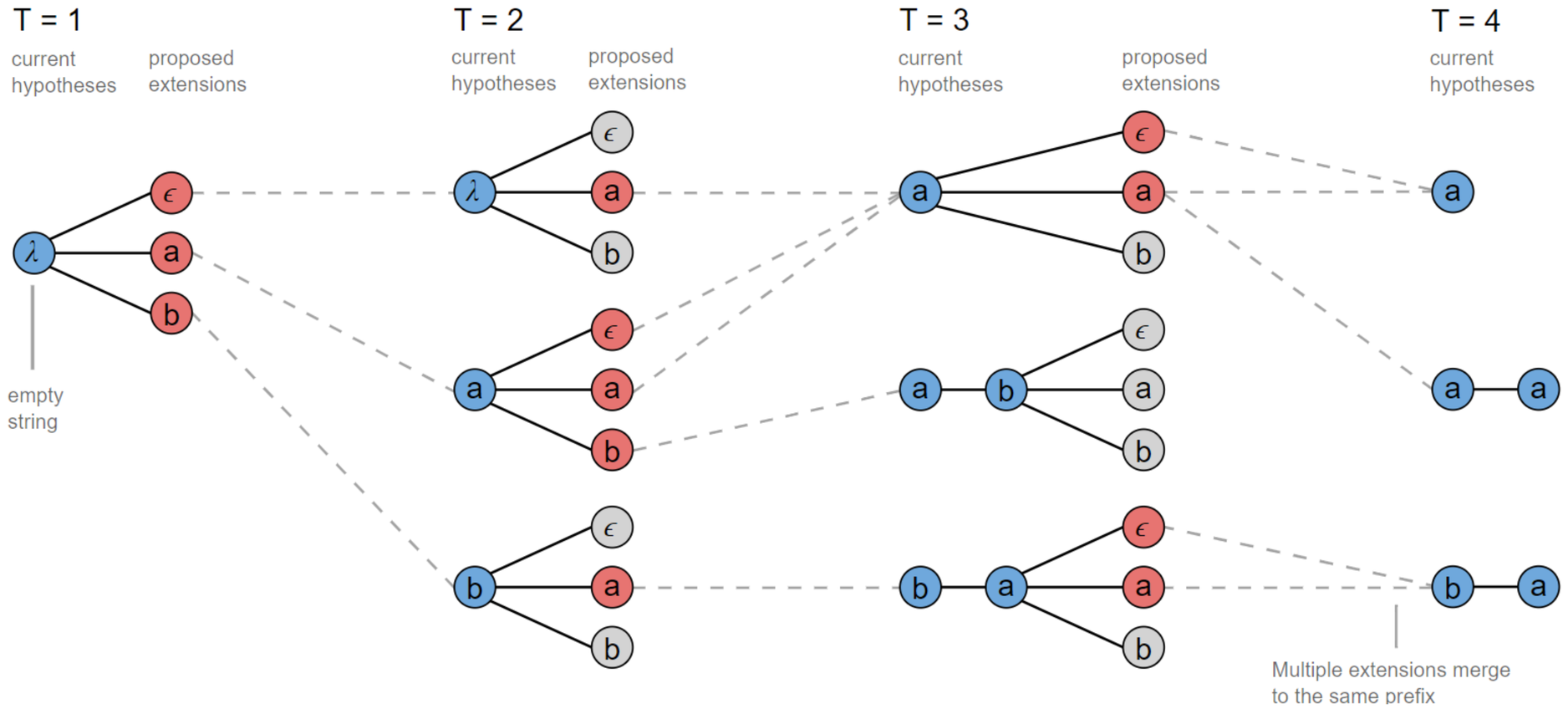
Работает плохо: вероятности $aa\epsilon$ и aaa по отдельности меньше вероятности bbb, но их сумма – больше.

Beam Search



Полный перебор в порядке BFS, но поддерживаем только топ-K вариантов

Beam Search



Модификация: сразу сжимаем повторы и ϵ-символы

Language model

Acoustic Model. До этого мы строили распределение по пространству текстов ограниченной длины, зная последовательность звуковых характеристик, т.е. условное распределение $P(TextSequence \mid AudioSequence)$

Проблема: различающиеся тексты, звучащие одинаково

Language Model. Возьмем корпус текстов языка и обучим модель, которая безусловно будет предсказывать $P(TextSequence)$, основываясь лишь на языковом распределении.

$$Y^* = \underset{Y}{\operatorname{argmax}} \quad p(Y \mid X) \cdot \quad p(Y)^{\alpha} \cdot \quad L(Y)^{\beta}$$

The CTC conditional probability.

The language model probability.

The “word” insertion bonus.

Во время подбора оптимального текста добавление ϵ-символа не будет изменять вклад языковой модели, что заставит поиск поощрять более короткие ответы. Добавим бонус за количество токенов

Deep Speech

$$h_{t,k}^{(6)} = \hat{y}_{t,k} \equiv \mathbb{P}(c_t = k|x) = \frac{\exp(W_k^{(6)} h_t^{(5)} + b_k^{(6)})}{\sum_j \exp(W_j^{(6)} h_t^{(5)} + b_j^{(6)})}$$

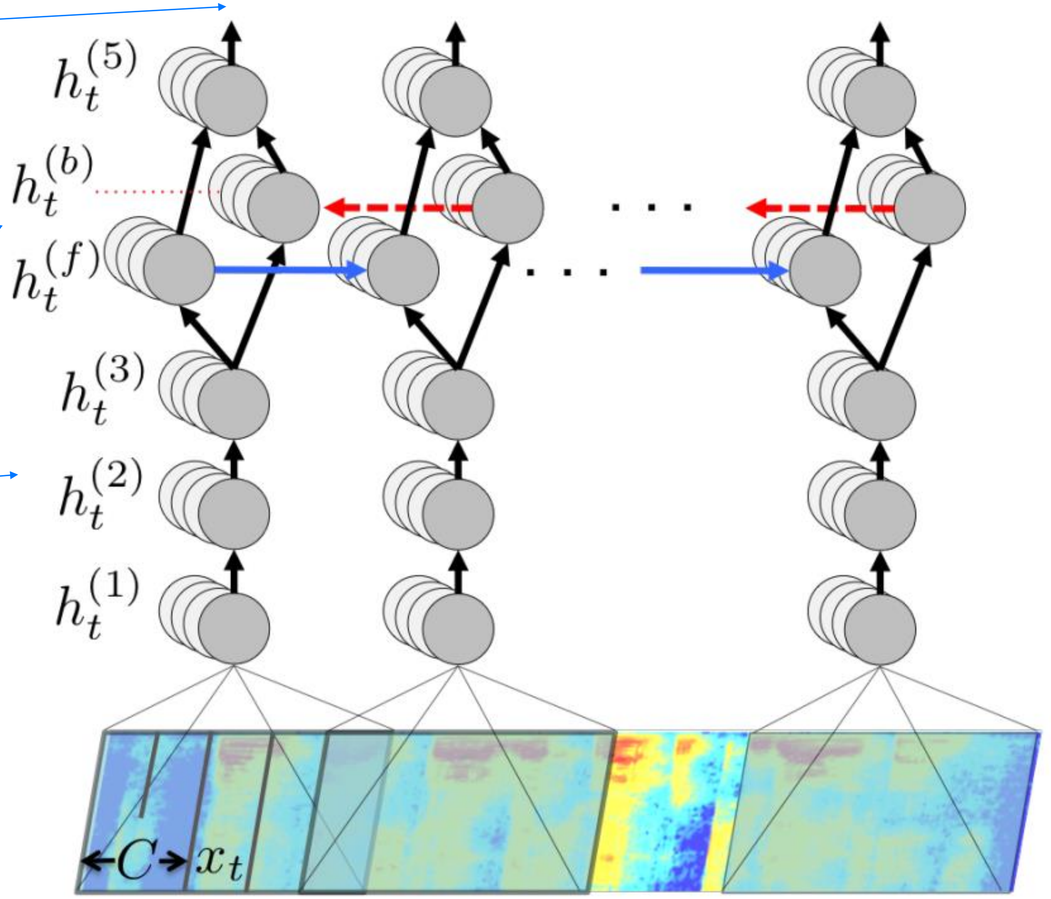
$$h_t^{(4)} = h_t^{(f)} + h_t^{(b)}$$

$$h_t^{(f)} = g(W^{(4)} h_t^{(3)} + W_r^{(f)} h_{t-1}^{(f)} + b^{(4)})$$

$$h_t^{(b)} = g(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t+1}^{(b)} + b^{(4)})$$

$$h_t^{(l)} = g(W^{(l)} h_t^{(l-1)} + b^{(l)})$$

$$g(z) = \min\{\max\{0, z\}, 20\}$$



1,2,3 Полносвязный слой

4 Двунаправленный рекуррентный

5 Полносвязный

Источники

- https://www.cs.toronto.edu/~graves/icml_2006.pdf
- https://distill.pub/2017/ctc/
- https://arxiv.org/pdf/1412.5567.pdf
- http://www.machinelearning.ru

3

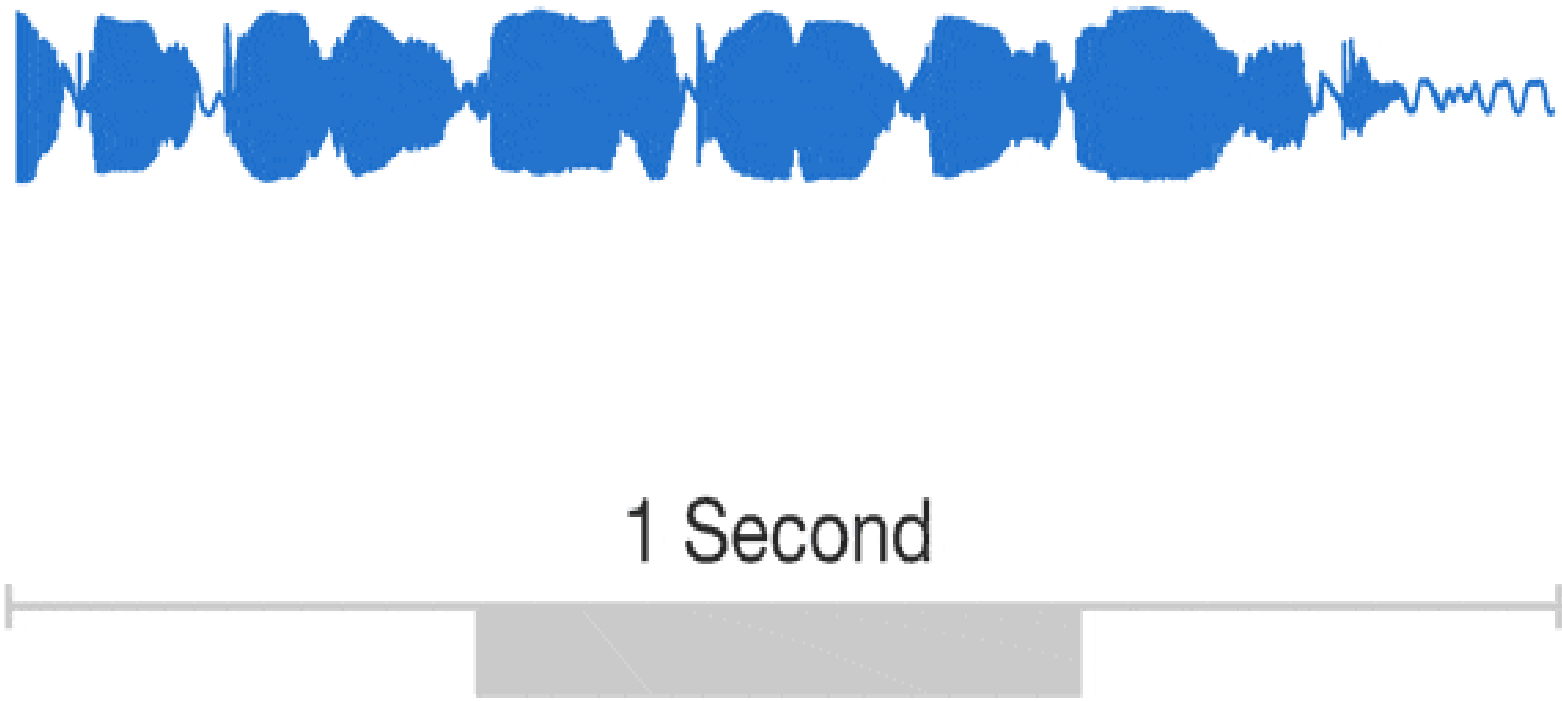
Генерация звука

Иван Фридман

WaveNet

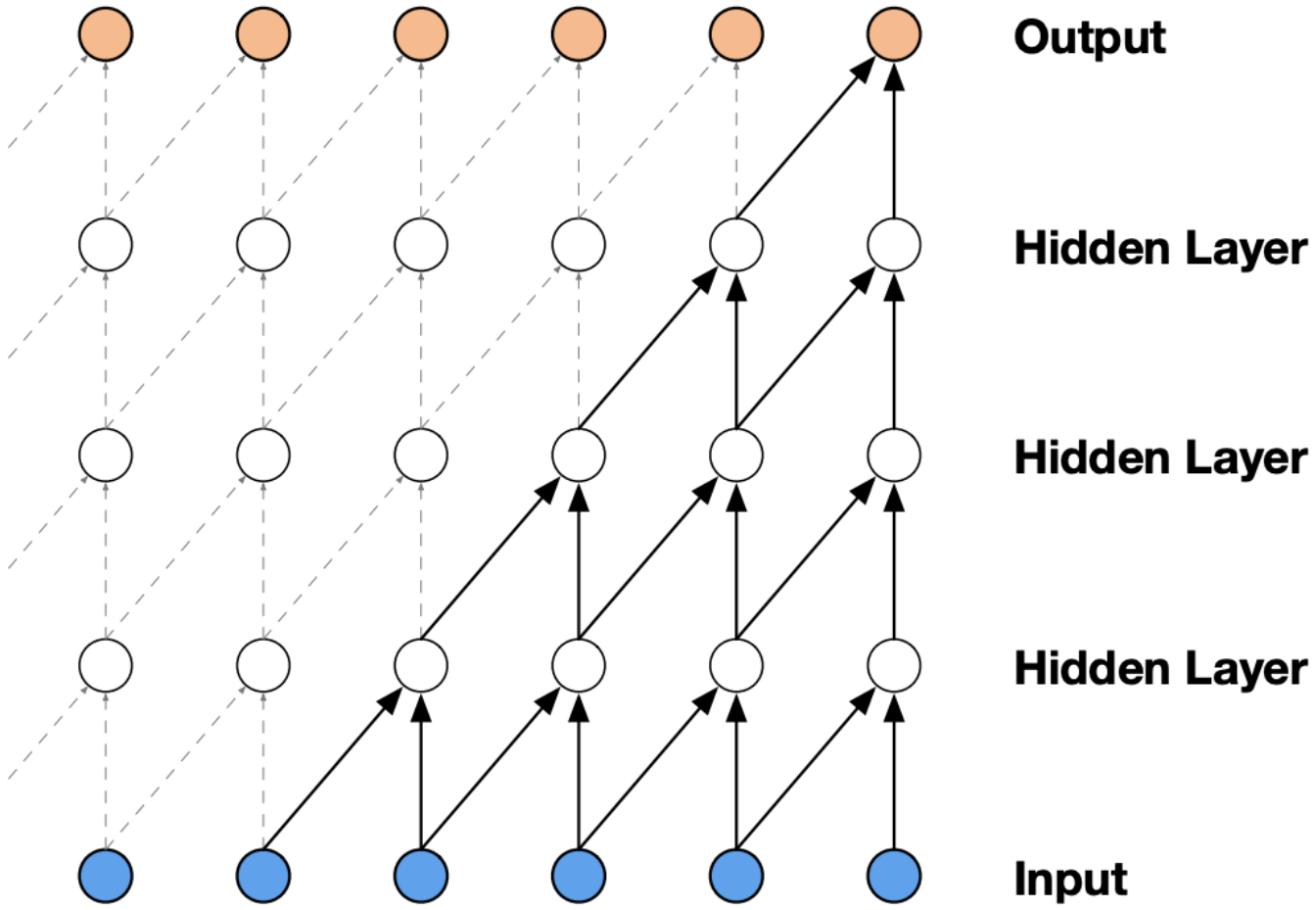
WaveNet – модель глубокого обучения для генерации звука в формате raw audio waveform

WaveNet отдает на выход звук в сыром формате - зависимость звукового давления от времени. Благодаря этому можно использовать WaveNet не только для генерации речи, но и других звуков, например музыки



Causal convolution

Основной минус обычных Causal Convolution заключается в том, что receptive field линейно зависит от числа слоев

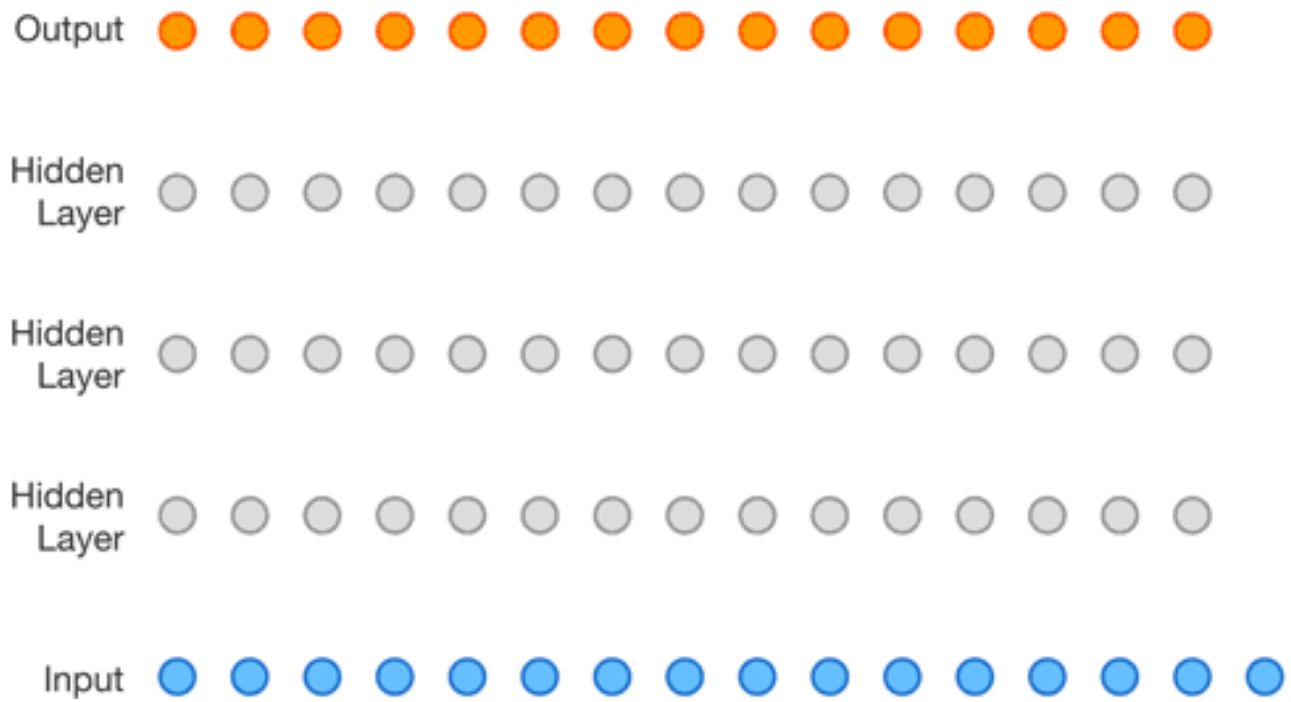


Dilated Causal Convolution

В Dilated Causal Convolution receptive field зависит экспоненциально от числа слоев

В WaveNet’e dilation сначала экспоненциально растет до какого-то предела, а потом повторяется, например:

[1, 2, …, 512, 1, 2, …, 512, …, 1, 2, …, 512]



Функции активации

Обычная: $$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right)$$

Условная: $$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^{T}\mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^{T}\mathbf{h}\right)$$

Здесь W – свертка, V – 1 полносвязный слой, $\odot$ - поэлементное умножение

В h хранится вся дополнительная информация, необходимая для генерации – номер спикера, текст, жанр музыки, etc

μ-law companding transformation

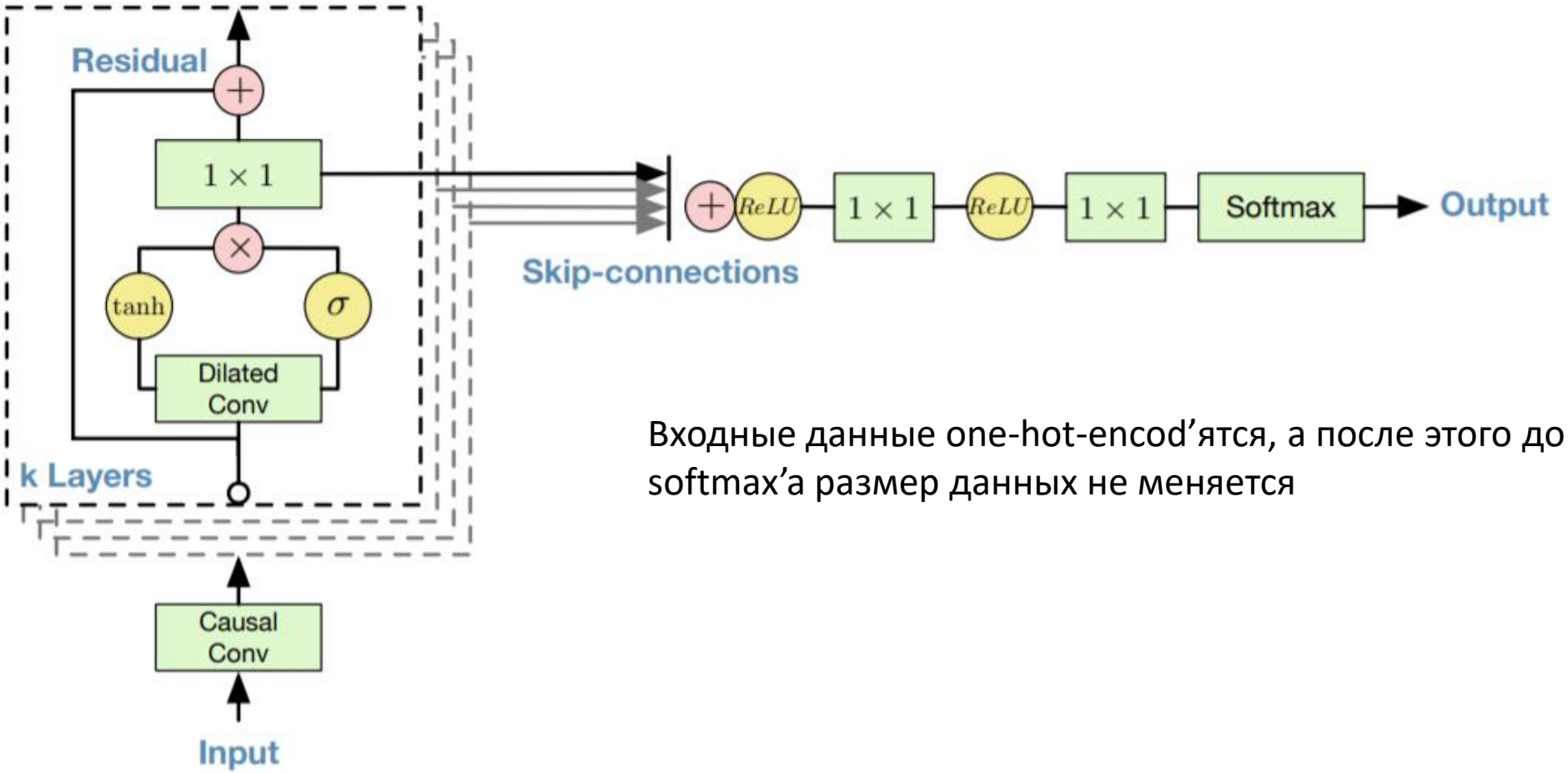
$$f(x_t) = \text{sign}(x_t)\frac{\ln(1+\mu|x_t|)}{\ln(1+\mu)}$$

μ = 255

Raw audio обычной кодируется последовательностью 16-битных чисел – по одному числу для каждого момента времени. Мы хотим для каждого момента времени решать задачу классификации, в этом случае нам пришлось бы для каждого таймстемпа генерировать 65536 вероятностей.

С помощью применения функции f, округления и применения обратной функции, мы уменьшаем количество необходимых вероятностей для Softmax’a с 65536 до 256

Архитектура WaveNet’a



Входные данные one-hot-encod’ятся, а после этого до softmax’а размер данных не меняется

Примеры результатов работы

Без текста:

Музыка:

С текстом:

Parametric

Concatenative

WaveNet

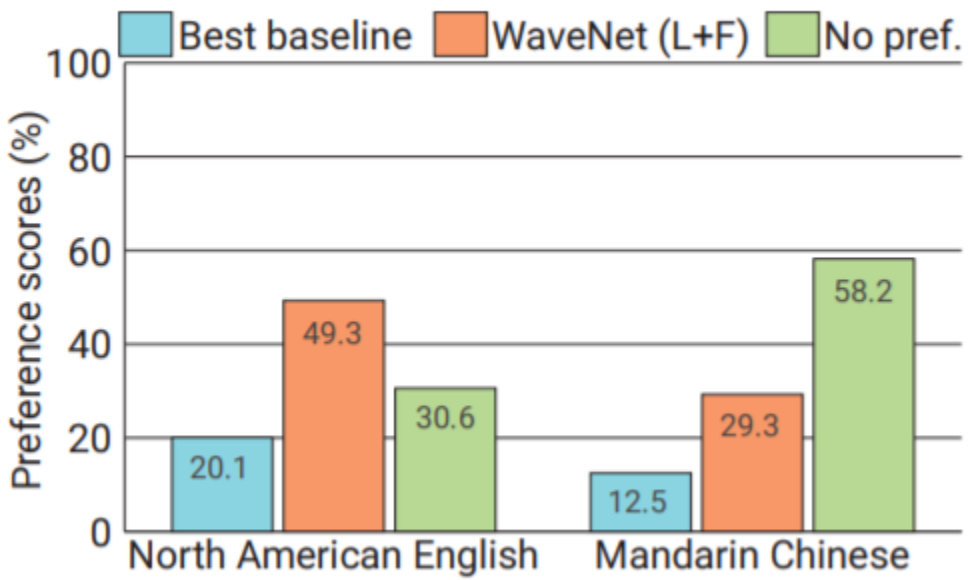
Оценка результатов работы

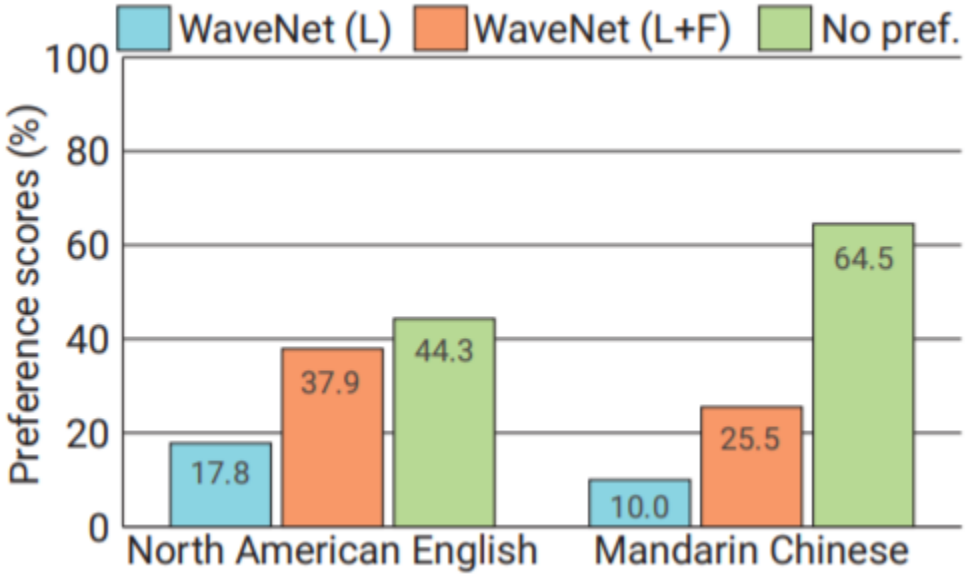
Speech samples	Subjective 5-scale MOS in naturalness	
	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082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

MOS – каждый респондент оценивает качество работы модели от 1 до 5

1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent

Оценка результатов работы





Оценка субъективного предпочтения – каждого респондента просят сравнить два результата работы моделей. Респондент может отдать предпочтение одной из моделей или сказать, что они примерно равны

Итог

Сильные стороны WaveNet’a:

- Очень высокое качество генерируемых аудиозаписей
- Большое разнообразие применений благодаря формату генерируемых аудиозаписей
- Широкий простор для кастомизации передаваемых параметров h: можно передавать что угодно от номера спикера и текста до жанра музыки и основной частоты аудиозаписи
- В задаче TTS WaveNet имитирует не только речь автора, но и такие сигналы как вздохи, чавканье, движения губ

Слабые стороны:

- Главный минус – очень длительная генерация из-за того что модель принимает на вход свой предыдущий сгенерированный фрагмент
- Невозможность параллельной генерации отдельных фрагментов аудиозаписи, так как нам нужно знать предыдущий сгенерированный фрагмент, чтобы начать генерировать следующий (но параллельное обучение возможно)