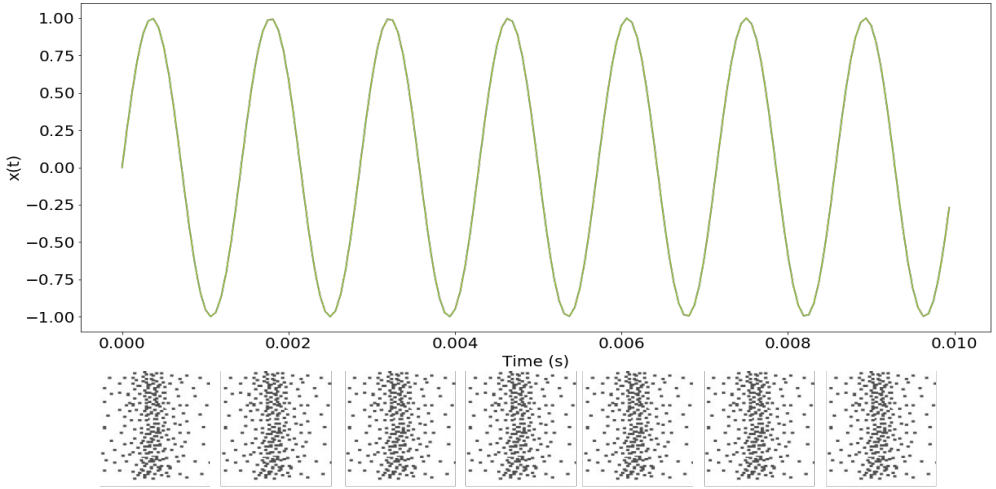
Прикладные задачи анализа данных

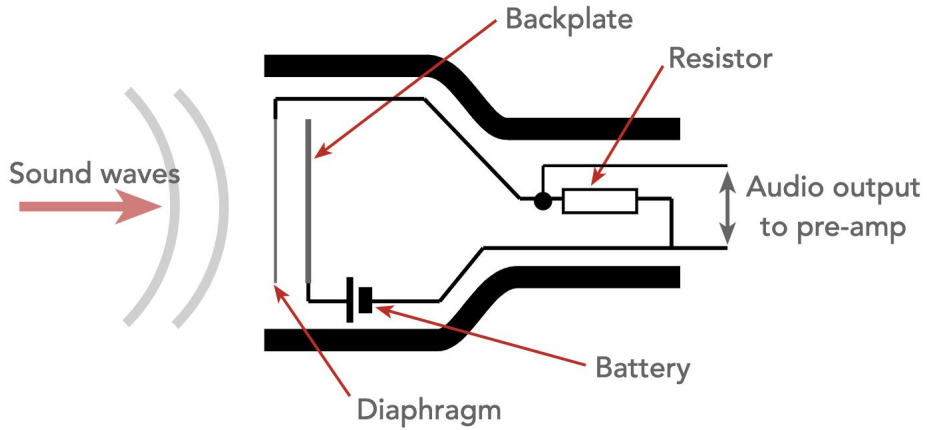
Лекция 6
Распознавание речи

Маркович Александр

Что такое звук?

- **Звук** – это колебания воздуха, то есть серия возрастающих и падающих значений давления воздуха
- **Микрофон** улавливает эти колебания воздуха и преобразовывает их в электрические колебания
- Эти колебания преобразуются в **аналоговый** сигнал, а затем и в **цифровой** сигнал

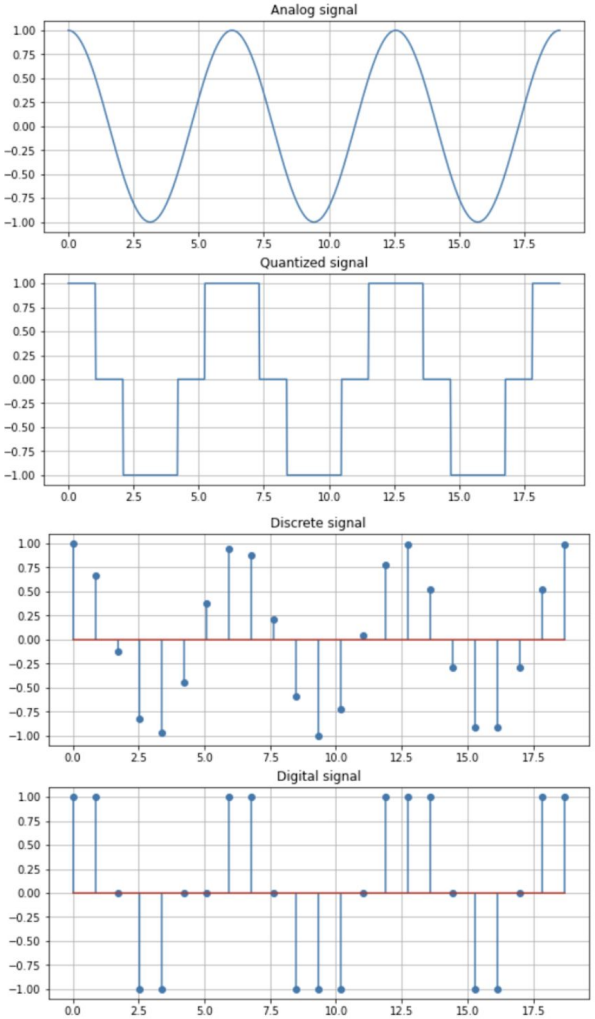




https://pudding.cool/2018/02/waveforms/
https://blog.accusonus.com/science-of-sound/how-microphones-work

Как звук хранится в компьютере?

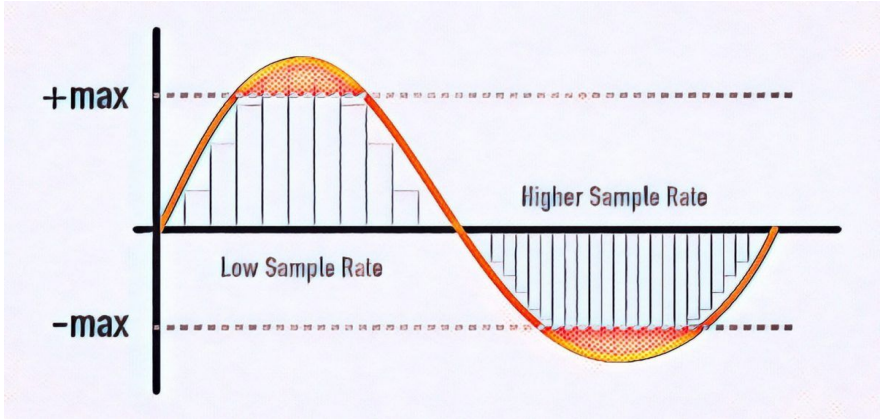
- Аналоговый сигнал подвергается **дискретизации**, **квантованию** и **кодированию**.
- **Дискретизация** аналогового сигнала состоит в том, что сигнал представляется в виде последовательности значений, взятых в дискретные моменты времени **t** с шагом **d**
- **Квантование** сигнала заключается в том, что область значений сигнала разбивается на **N** уровней с шагом **d** и для каждого отсчета выбирается уровень, который ему соответствует

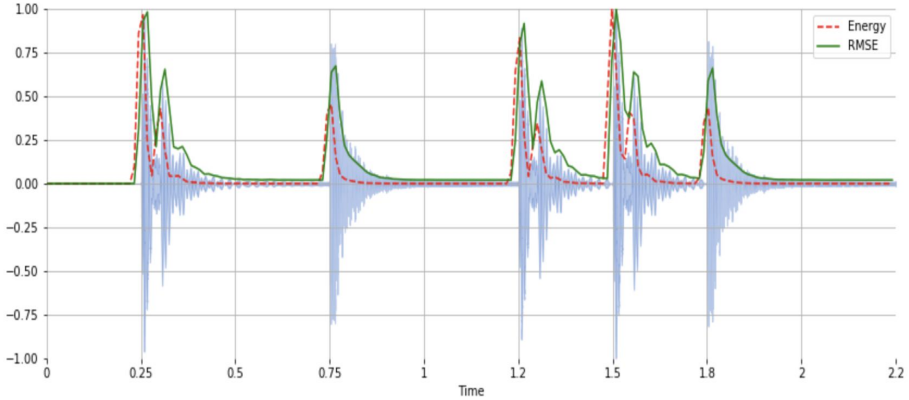


https://github.com/markovka17/dla/blob/master/week01/dsp.ipynb
https://web.sonoma.edu/esee/courses/ee442/archives/sp2019/lectures/lecture09_pcm.pdf

Какие еще есть характеристики?

- **Частота дискретизации** -- количество отсчетов амплитуды в секунду
- **Количество каналов** -- количество сигналов, которые мы записываем одновременно
- **Мощность сигнала** -- $x(n)^2$
- **Энергия сигнала** -- $\sum_n |x(n)|^2$

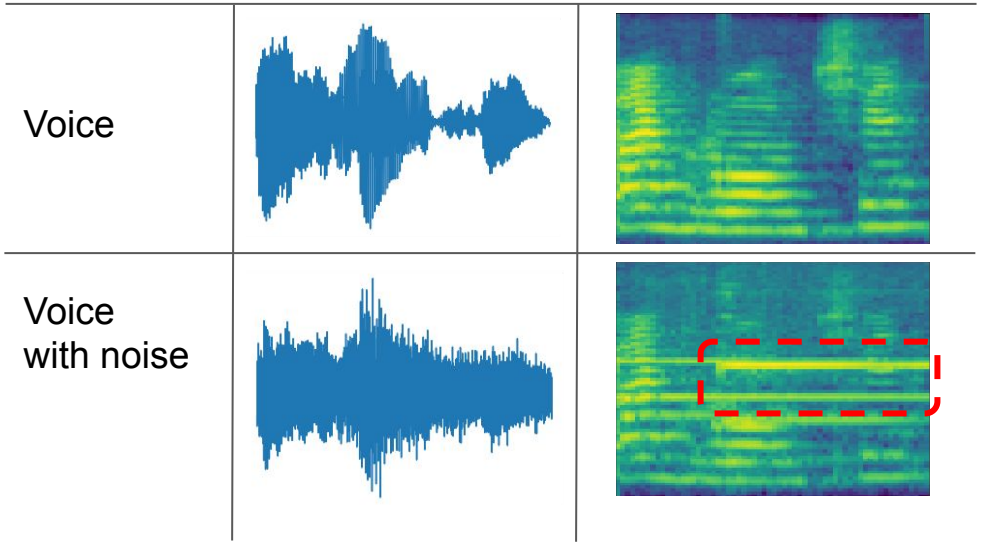




https://musicinformationretrieval.com/energy.html

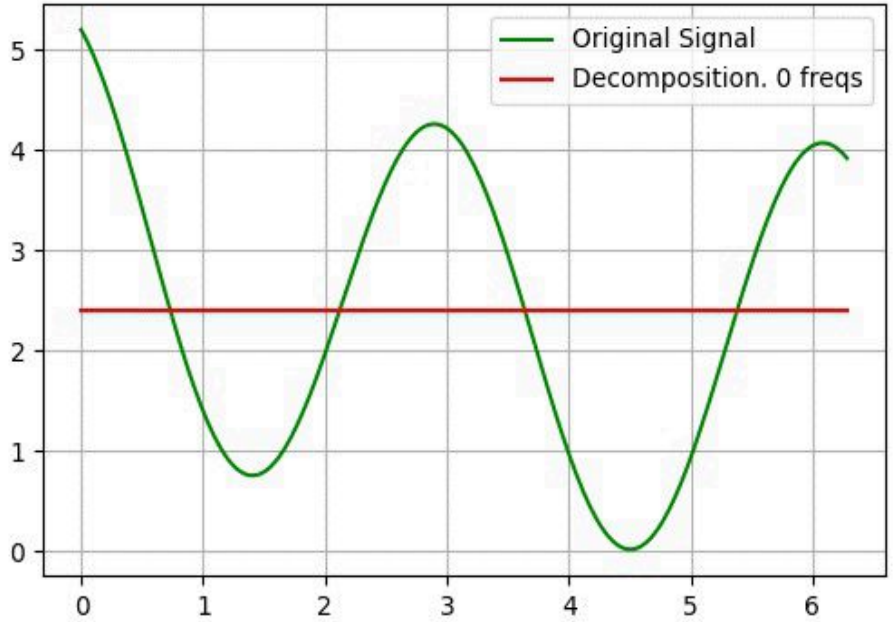
Почему плохо работать со звуком в таком формате?

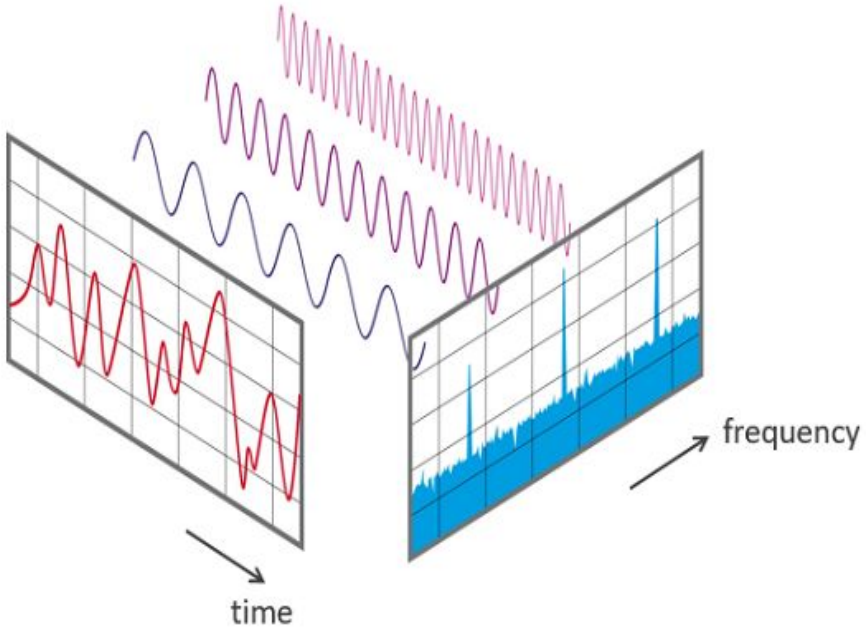
- Одна буква/звук состоит из 2000-4000 амплитуд -- дорого обрабатывать и хранить
- Нет “инварианта” относительно шума и трансформаций



Преобразование Фурье for dummies

$$f(t) = 5 + 2\sin(2t + 2) - 3\cos(0.2t - 1)$$





https://youtu.be/spUNpyF58BY

Дискретное преобразование Фурье

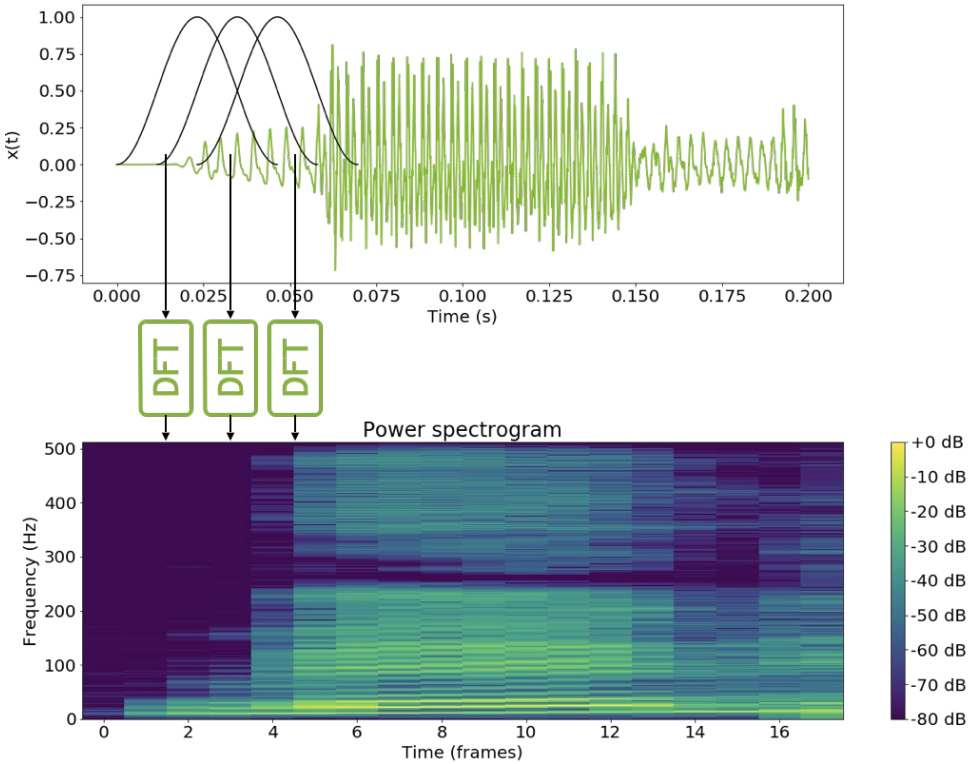
$$X = \mathbf{M}x$$

$$M_{mn} = \exp\left(-2\pi i \frac{(m-1)(n-1)}{N}\right)$$

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

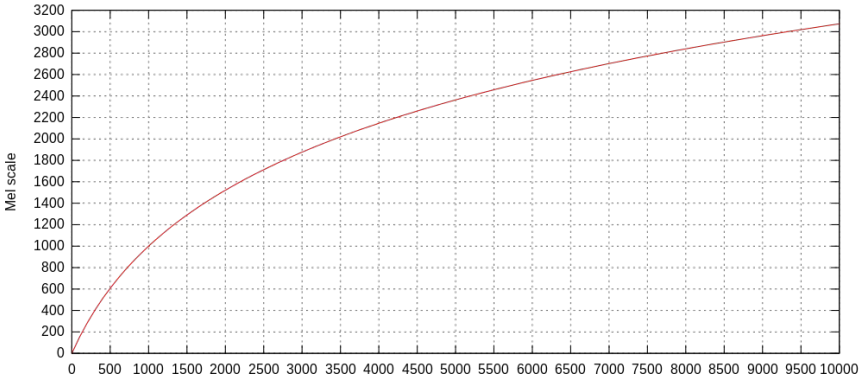
Спектрограмма (та самая)

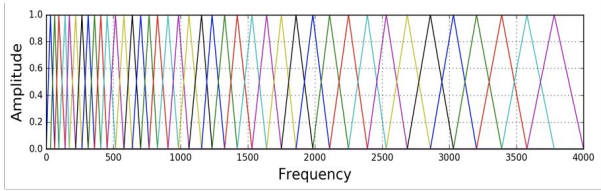
- Нарезаем сигнал на **окна** с **пересечением**
- Применяем **оконную функцию** к вырезанному окну
- Применяем дискретное преобразование **Фурье**
- Считаем **квадрат комплексной нормы**
- Берем **половину вектора** в силу его симметричности

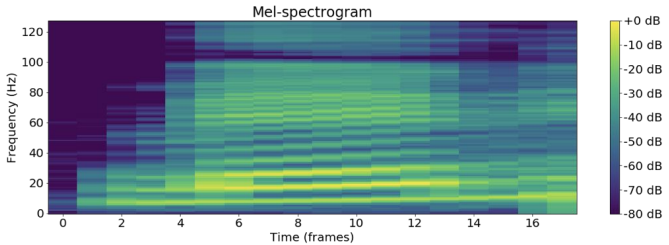


МелСпектрограмма

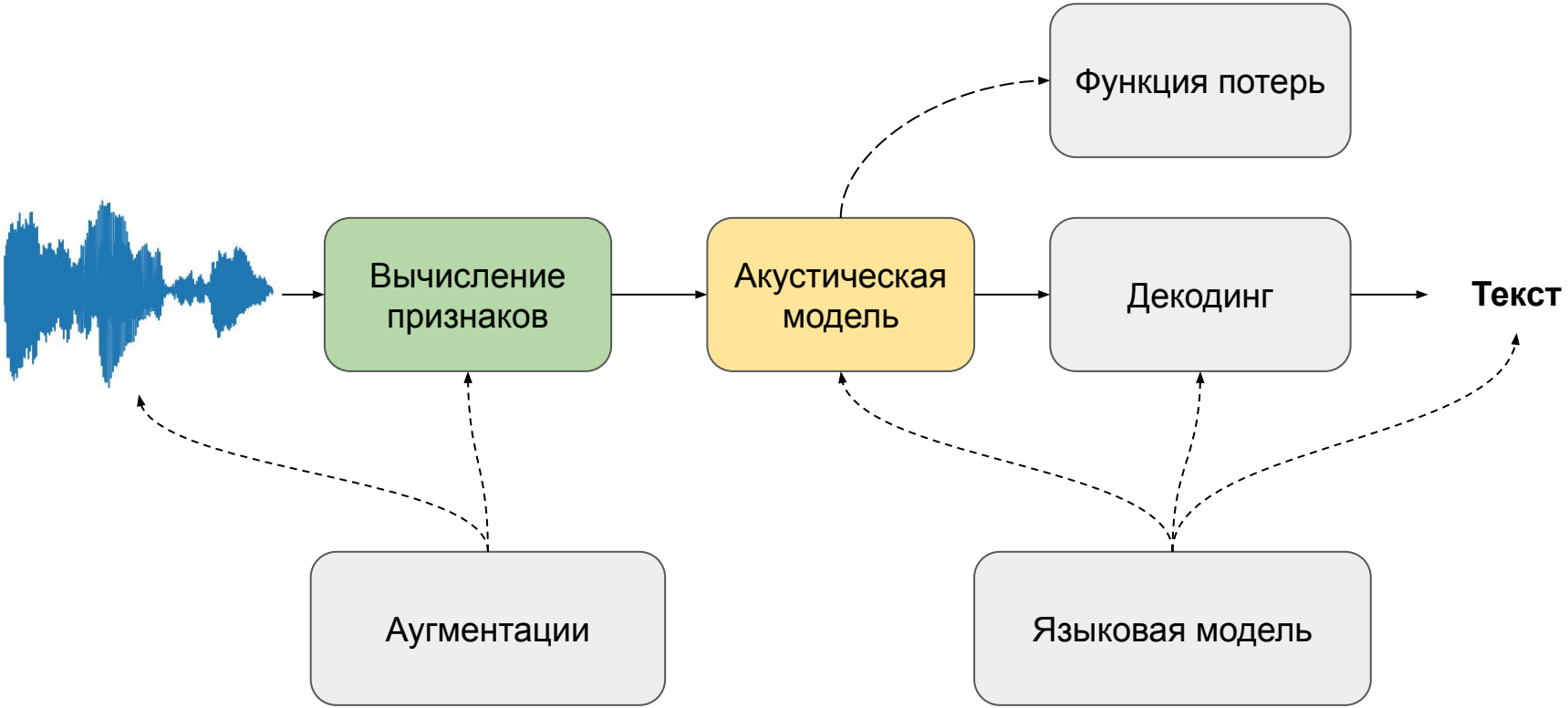
- Человеческое ухо хорошо слышит **низкие** частоты и плохо **высокие**
- Учитываем в спектрограмме **низкие** частоты больше, а **высокие** меньше
- $$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \ln\left(1 + \frac{f}{700}\right)$$
  $$f = 700\left(10^{\frac{m}{2595}} - 1\right) = 700\left(e^{\frac{m}{1127}} - 1\right)$$
- В конце берем **логарифм** от мелспектрограммы







Распознавание голоса в 2к21



Первым делом метрики!

- Word Error Rate (WER) -- доля “неправильных” слов
- $$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$
- **S** – кол-во замен

  D – кол-во удалений

  I – кол-во вставок

  C – кол-во совпадений

  N – S + **D** +**C** - кол-во слов всего
-

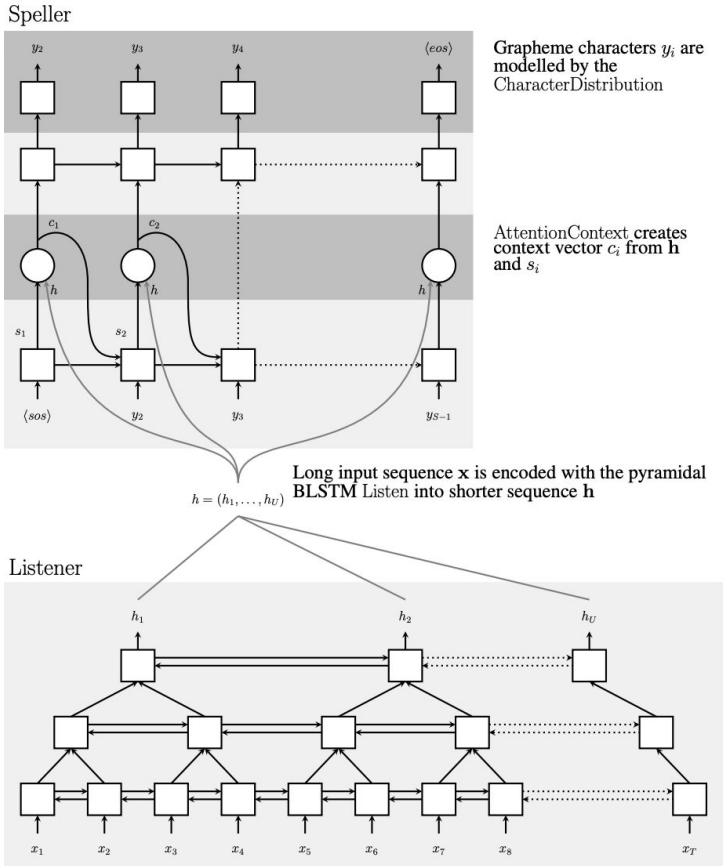
```
True: quick brown     fox jumped over a lazy dog
Pred: quick brow  an  fox jumped over   lazy dog
```

- Character Error Rate (CER) -- то же самое, но для символов (букв, например)
- Проще оптимизировать

https://en.wikipedia.org/wiki/Levenshtein_distance

Listen, Attend & Spell

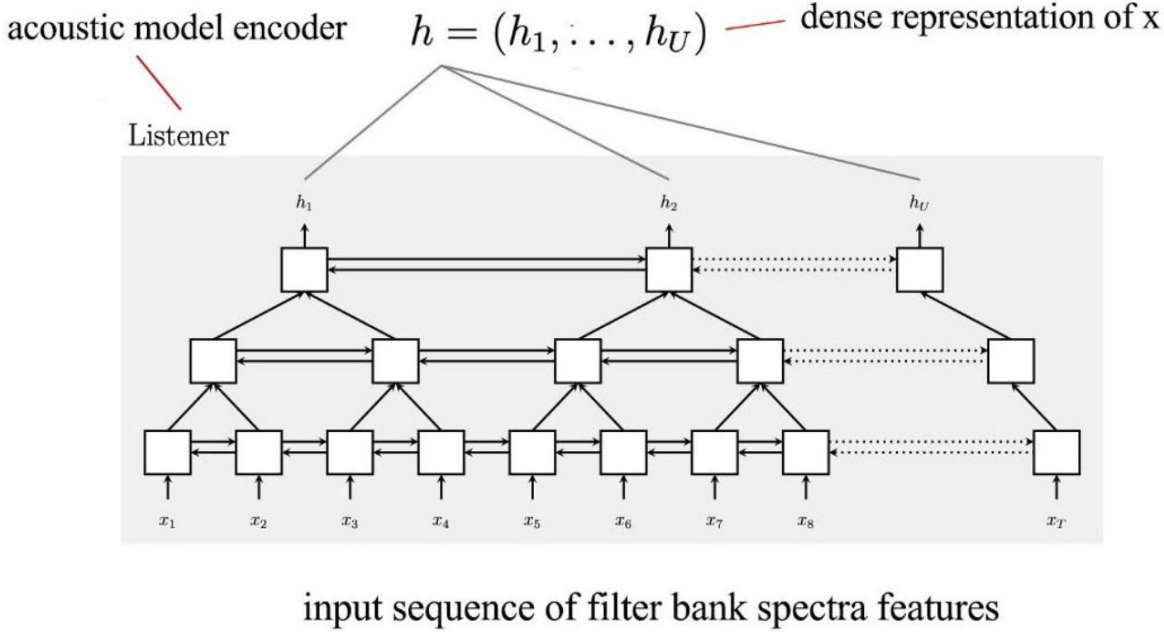
- Базовая старая, но **хорошая** модель. seq2seq с механизмов внимания для выравнивания
- **Listener** - пирамидальный bi-LSTM энкодер
- **Speller** - обычный декодер :)
- **Attend** - внимание со скалярным произведением в качестве функции близости



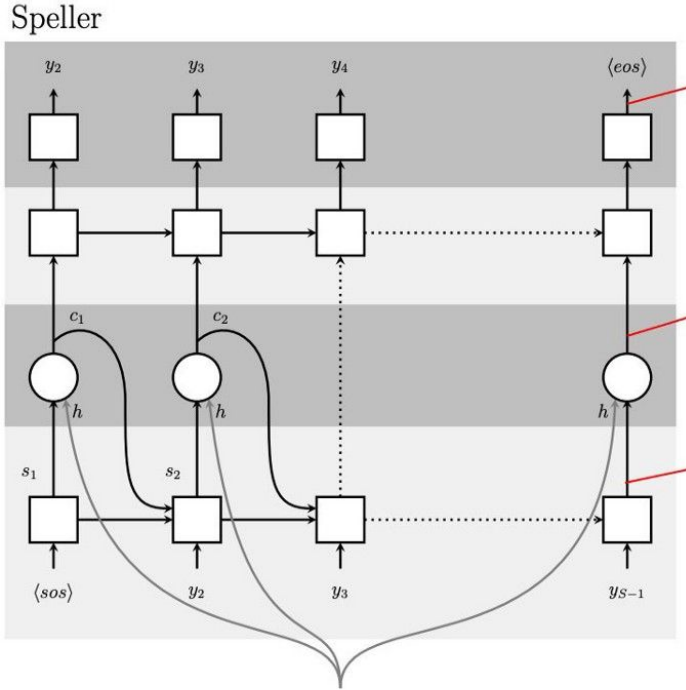
https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html#attention_idea
https://arxiv.org/pdf/1508.01211.pdf

Listener

- На вход батч из **МелСпектрограм**
- Конкатенируем **соседнием активации** в одну и передаем выше
- Уменьшаем временную размерность в **4-8 раз**



Attend & Spell



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

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

dot product

$$e_{i,u} = \langle \phi(s_i), \psi(h_u) \rangle$$

$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_u \exp(e_{i,u})}$$

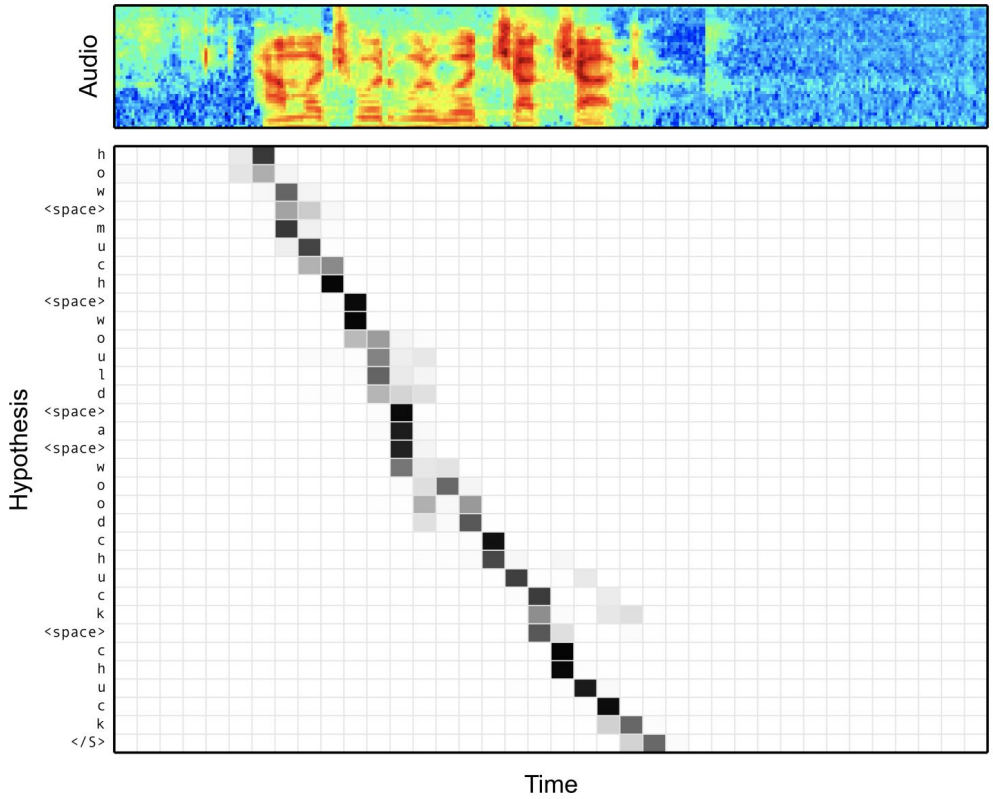
$$c_i = \sum_u \alpha_{i,u} h_u$$

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

decoder state

$$h = (h_1, \ldots, h_U)$$

Attention Visualization

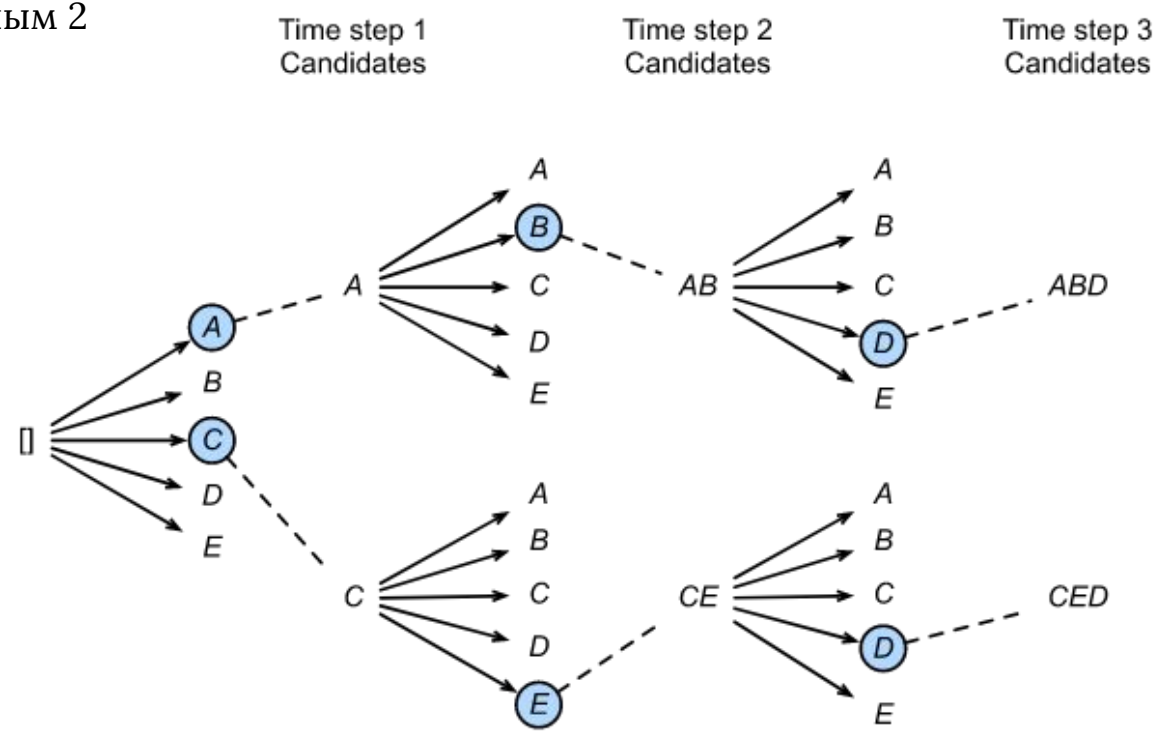


Несколько деталей о LAS

- Минимизируем кросс-энтропию между предсказаниями (**категориальное распределение**) и правильными буквами (**one-hot**)
- Учись с помощью **teacher-forcing**-а
- Декодируем с помощью **beam-searching**-а
- Добавляем **LM** для улучшения качества

Что такое beam-searching?

Для простоты фиксируем размер словаря равным 5 и размер beam-а равным 2

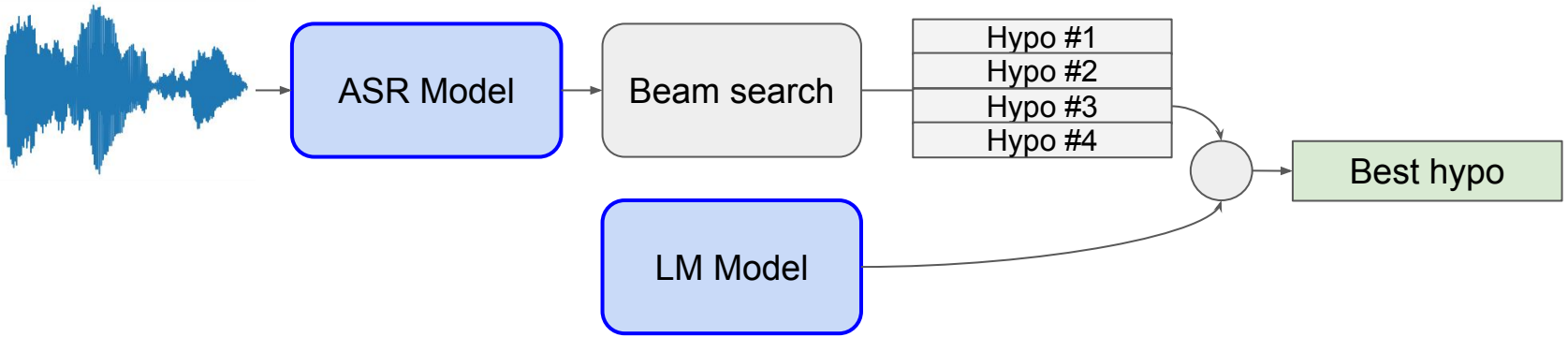


Что такое LM fusing?

Second-pass rescoring

Обучаем ASR и LM отдельно. Пересчитываем вероятности гипотез с помощью LM

$$P_{\text{final}} = P_{ASR}(Y \mid X) + \alpha \cdot P_{LM}(Y) + \beta \cdot \text{len}(Y)$$

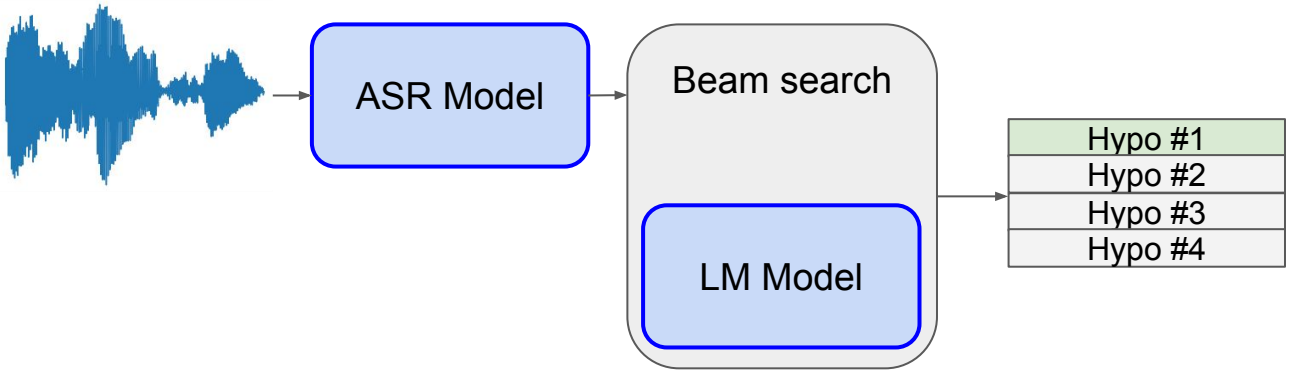


Что такое LM fusing?

Shallow fusing

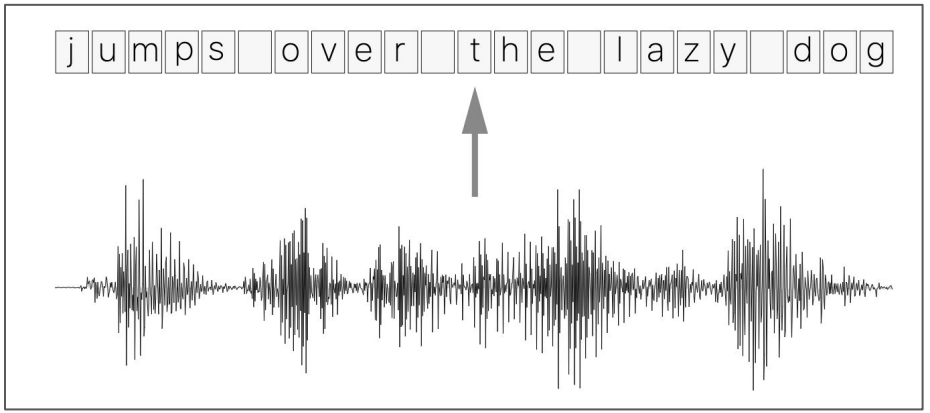
Обучаем ASR и LM отдельно. Добавляем LM на каждый шаг BM

$$P_{\text{final}} = P_{ASR}(Y \mid X) + \alpha \cdot P_{LM}(Y) + \beta \cdot \text{len}(Y)$$



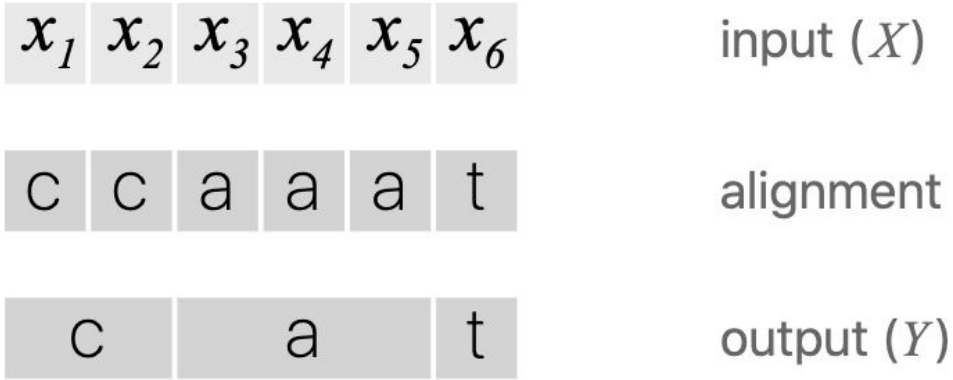
Connectionist Temporal Classification

- Вход и выход разной длины и они не выровнены
- Хочется уметь считать P(Y | X) для любой длины |X| и |Y|
- argmax P(Y | X)

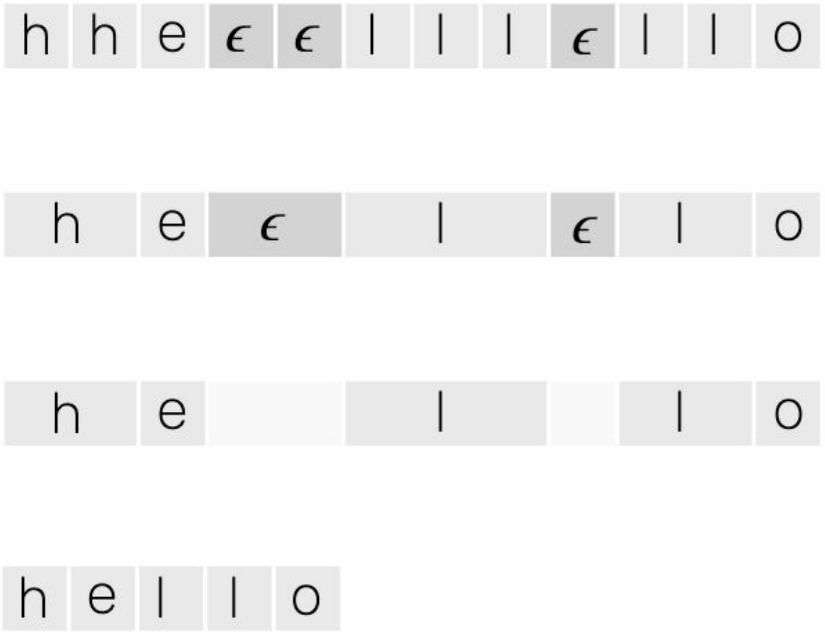


Connectionist Temporal Classification

- Для каждого фрейма делаем предсказание над словарем (например, букв)
- Склеиваем соседние одинаковые предсказания
- Возникают проблемы с тишиной между словами и повторяющимися символами



Connectionist Temporal Classification

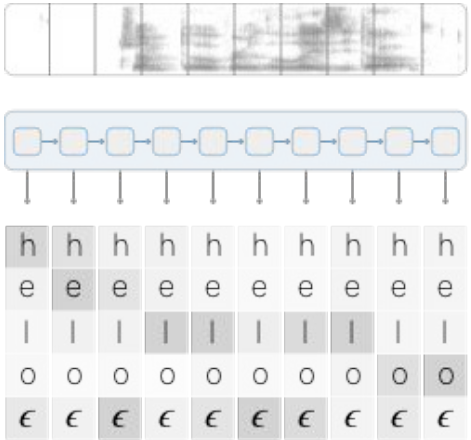


First, merge repeat characters.

Then, remove any ϵ tokens.

The remaining characters are the output.

Connectionist Temporal Classification



We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs

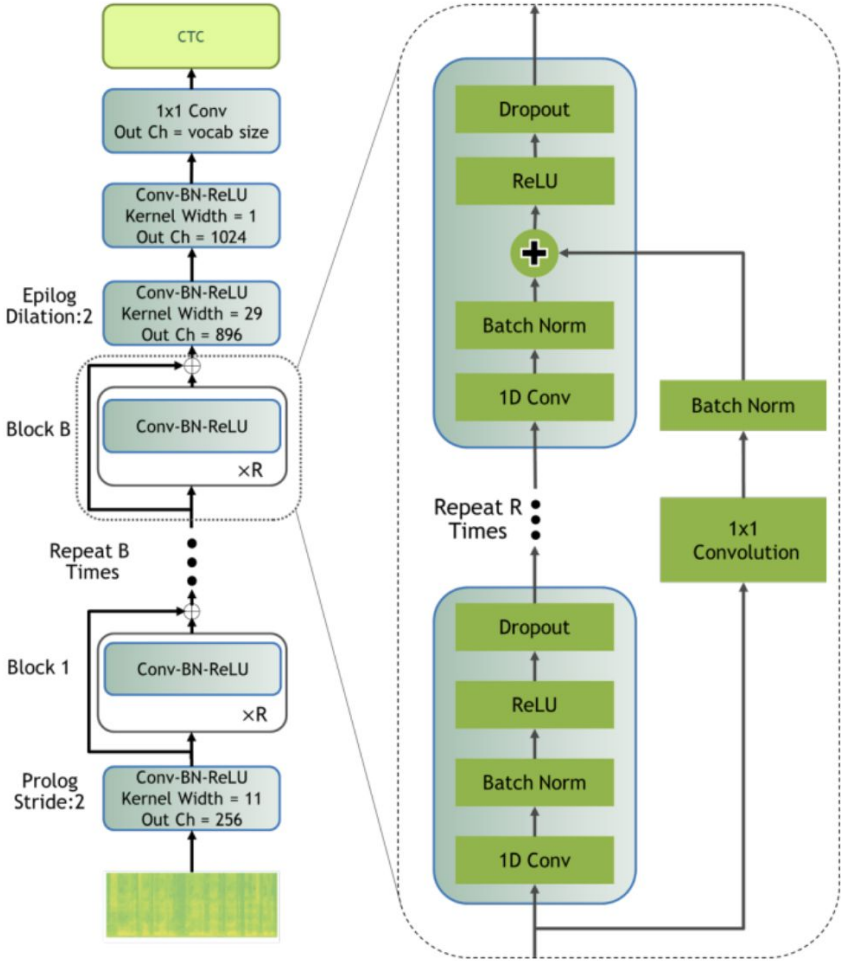
Но как считать и пробрасывать градиенты через CTC?

Connectionist Temporal Classification

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

Jasper

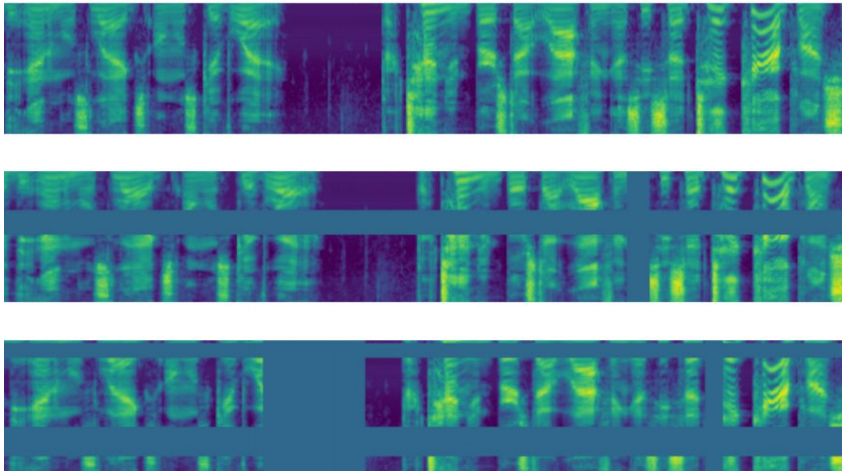
- 1D convolution, BatchNorm, ReLU, and Dropout
- Residual connections
- Функция потерь CTC
- Уменьшаем временную размерность в 4 раза
- Очень быстрый и дешевый инференс



https://arxiv.org/pdf/1904.03288.pdf

Какие бывают аугментации?

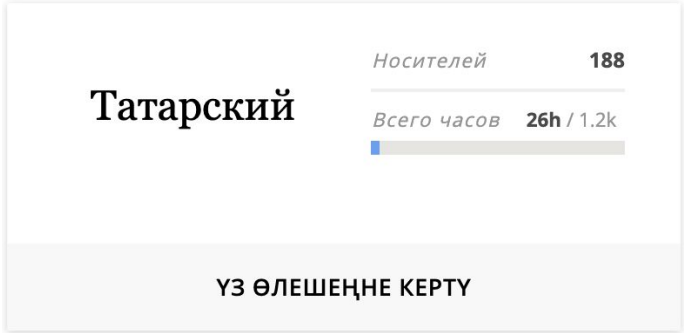
- Аугментации наше все
- Помогают с нехваткой данных и улучшением робастности
- Можно применять либо к звуковому сигналу, либо с уже МелСпектрограмме
- Добавляем шум, меняем громкость, скорость, частоту, акустику...
- Маскируем частоты или время в МелСпектрограмме



https://github.com/markovka17/dla/blob/master/week02/seminar.ipynb
https://arxiv.org/pdf/1904.08779.pdf

Что на счет открытых данных?

- **LibriSpeech**.
  ~1000 часов английской речи
- **CommonVoice**
  Десятки языков и тысячи часов
- **VoxCeleb**
  Тысячи голосов
- **И еще много чего**
  https://tinyurl.com/swker6w6

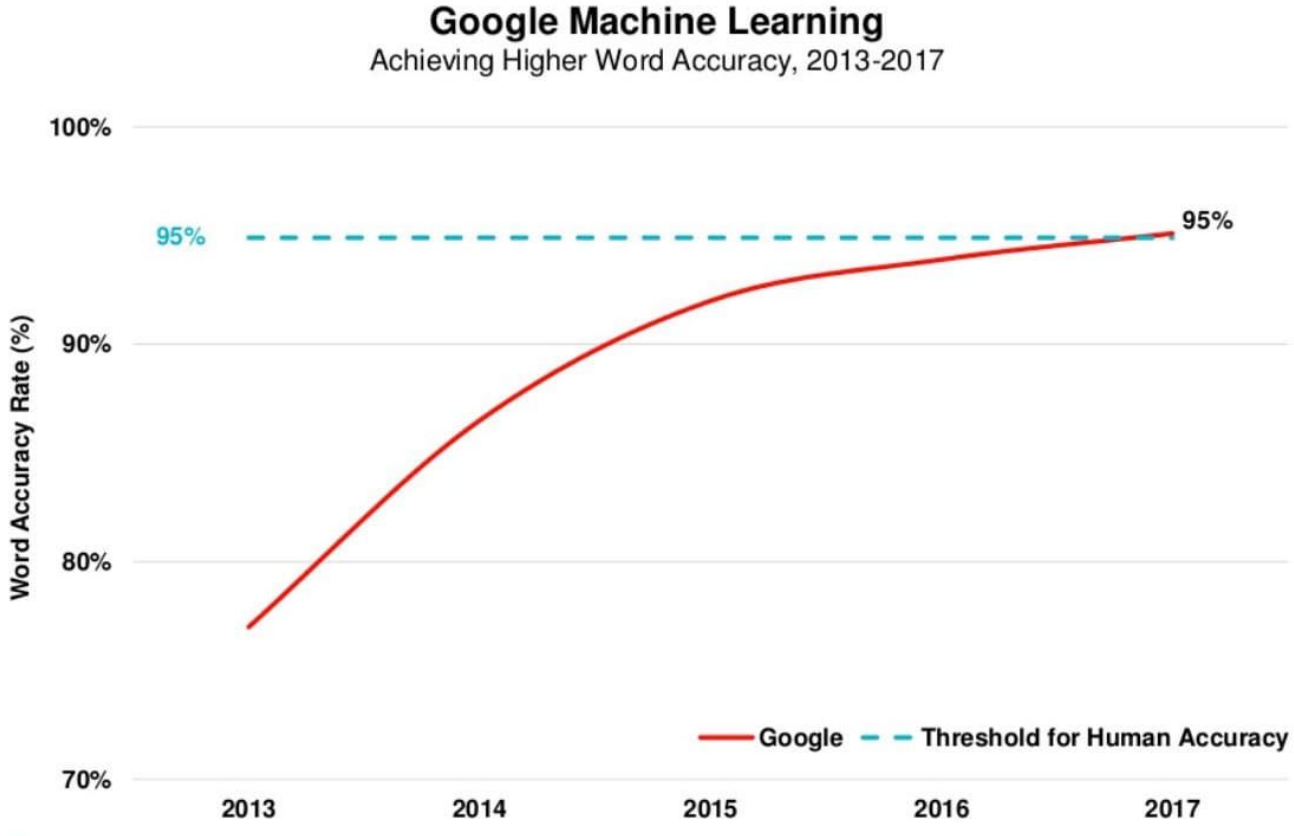


https://www.openslr.org/12
https://commonvoice.mozilla.org/

А как размечать данные?

https://labelstud.io/

В целом это решенная задача :)



Вопросы/набросы?