

Распознавание речи

Виктор Китов

v.v.kitov@yandex.ru



Содержание

- 1 Введение
- 2 Listen-Attend-Spell
- 3 Connectionist Temporal Classification

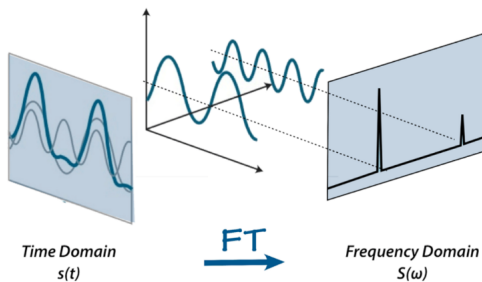
Работа со звуком

- Звук -> класс либо сегментация звуковой дорожки
 - голосовой помощник: команда / фоновый шум
 - определение композиции (shazam)
 - категоризация музыкального стиля (spotify)
- Звук -> сегментация
 - разметка спикеров
- Звук -> текст (automatic speech recognition, ASR)
- Текст -> звук (text to speech, TTS)
- Удаление шумов (denoising)
- Повышение качества аудио
 - ↑ частоты дискретизации (bandwidth expansion)
- Стилизация голоса
- Генерация музыки

Представление звука

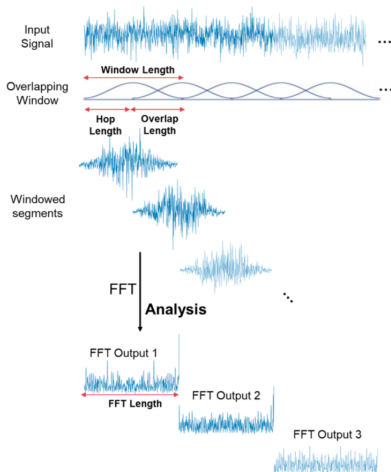
- Звук режем на пересекающиеся окна.
- Для каждого положения окна:
 - спектр (спектрограмма)
 - логарифм спектра (лог-спектрограмма)
 - логарифмически усредненный спектр (мел-спектрограмма)
 - спектр от спектра (MFCC)

Представление звука¹



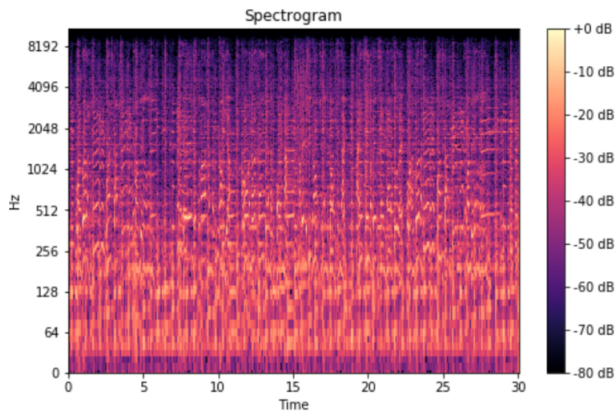
¹Ссылка на иллюстрацию.

Представление звука²



²Ссылка на иллюстрацию.

Представление звука³



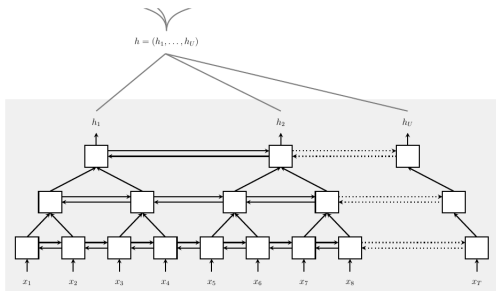
³[Ссылка на иллюстрацию.](#)

Содержание

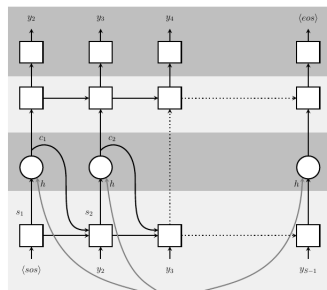
- 1 Введение
- 2 Listen-Attend-Spell
- 3 Connectionist Temporal Classification

Модель Listen-Attend-Spell⁴

- Listen-Attend-Spell - распознавание речи с помощью seq2seq+attention.
- Выход - распределение символов, начиная с $\langle \text{sos} \rangle$ и заканчивая $\langle \text{eos} \rangle$.



Listener



Speller

$$h = (h_1, \dots, h_T)$$

⁴<https://arxiv.org/abs/1508.01211>

Модель Listen-Attend-Spell

- Декодер выдаёт распределение на символах
 - $\{a, b, c, \dots, z, 0, \dots, 9, \langle \text{space} \rangle, \langle \text{comma} \rangle, \langle \text{period} \rangle, \langle \text{apostrophe} \rangle, \langle \text{unk} \rangle$ (для прочих символов).
- На вход декодера (после $\langle \text{sos} \rangle$)
 - с $p = 0.9$: реальный символ (teacher forcing)
 - с $p = 0.1$: ранее сгенерированный (free run)
 - учим модель исправлять ошибки
- Энкодер - 3х уровневый bidirectional LSTM (BLSTM)
- Перерасчет состояния на уровне l :

$$h_t^l = BLSTM \left(h_{t-1}^l, \left[h_{2t}^{l-1}, h_{2t+1}^{l-1} \right] \right)$$

- уменьшение в 2 раза длины последовательности
- 3 уровня, длина выходных состояний в 2^3 раз короче длины входа.

Внимание в декодировщике

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

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

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

- Расчёт контекста c_i :

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

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

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

- $\phi(\cdot), \psi(\cdot)$ - многослойные перцептроны.

Генерация выходов

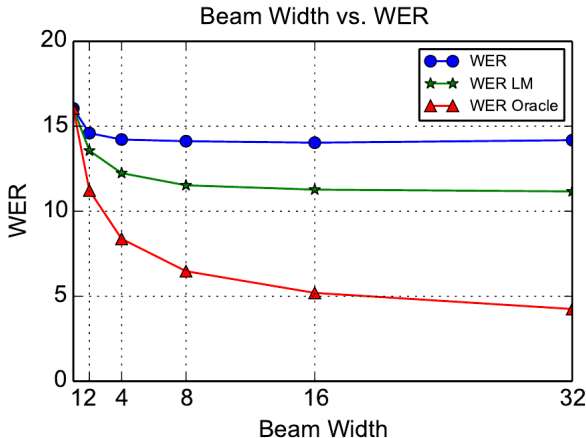
- Генерация выходной последовательности - через BeamSearch (32 лучших гипотезы)
- Считался score:

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

- $\log P < 0$, поэтому нормировка на #символов выхода $|y|_c$
↑ score
 - чтобы поощрить модель выдавать более длинные последовательности
- $P_{LM}(y)$ - вероятность y по языковой модели.
 - много текстов для обучения
 - существенно ↑ качество
- Аугментация при обучении:
 - добавление эхо (reverberations)
 - добавление внешних шумов (из видео YouTube)

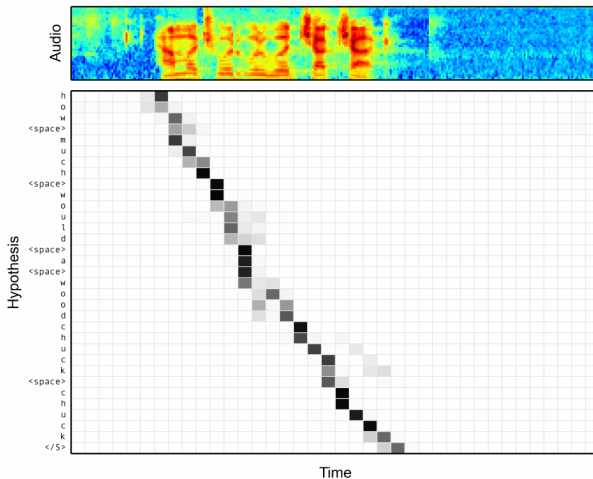
Word-error-rate от #гипотез лучевого поиска

- С 16 гипотез качество почти не улучшается.
- Включение языковой модели \uparrow качество



Визуализация внимания

Alignment between the Characters and Audio



Содержание

- 1 Введение
- 2 Listen-Attend-Spell
- 3 Connectionist Temporal Classification

CTC^{5,6}

- Требуется построить $x_1x_2...x_T \rightarrow y_1y_2...y_U$,
 - $T \neq U$, объекты посл-тей монотонно связаны во времени
- Примеры:
 - рукописный текст->текст
 - звук->текст
 - видео->разметка событий на фреймах

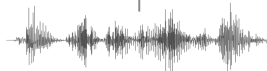
the quick brown fox



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.

⁵https://www.cs.toronto.edu/~graves/icml_2006.pdf

⁶Тutorial.

Проблема выравнивания

- Для оценки модели $X \rightarrow Y$ необходимо оценивать $p(Y|X)$

x_1	x_2	x_3	x_4	x_5	x_6	input (X)
c	c	a	a	a	t	alignment
c		a		t		output (Y)

- Важно построить соответствие между символами.
 - линейная связь во времени? -> нет
 - вручную -> дорого
- СТС агрегирует по всем возможным выравниваниям:

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

The CTC conditional probability marginalizes over the set of valid alignments computing the **probability** for a single alignment step-by-step.

- Распределение $p(a_t|X)$ выдает модель
 - сначала CNN (conv1d или широкая свёртка), потом RNN.

Выравнивание и кодировка

- Введем пустой символ ϵ в кодировке a_t .
- Преобразование $A \rightarrow Y$:
 - 1 объединить повторяющиеся символы в один
 - 2 убрать ϵ (за счёт этого шага Y может повторять символы)

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

First, merge repeat characters.

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

Then, remove any ϵ tokens.

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

The remaining characters are the output.

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

Выравнивание и кодировка

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'

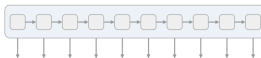
- Свойства выравнивания:

- монотонность соответствий символов $A_t \rightarrow Y_s$
- отображение $A_t \rightarrow Y_s$ many-to-one, $|A| \geq |Y|$

Последовательность



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



The input is fed into an RNN, for example.

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
€	€	€	€	€	€	€	€	€	€

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

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

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

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

By marginalizing over alignments, we get a distribution over outputs.

$$p(Y | X) =$$

The CTC conditional probability

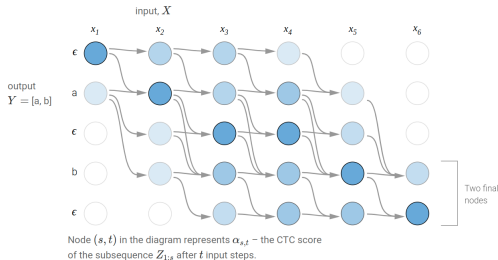
$$\sum_{A \in \mathcal{A}_{X,Y}}$$

marginalizes over the set of valid alignments

$$\prod_{t=1}^T p_t(a_t | X)$$

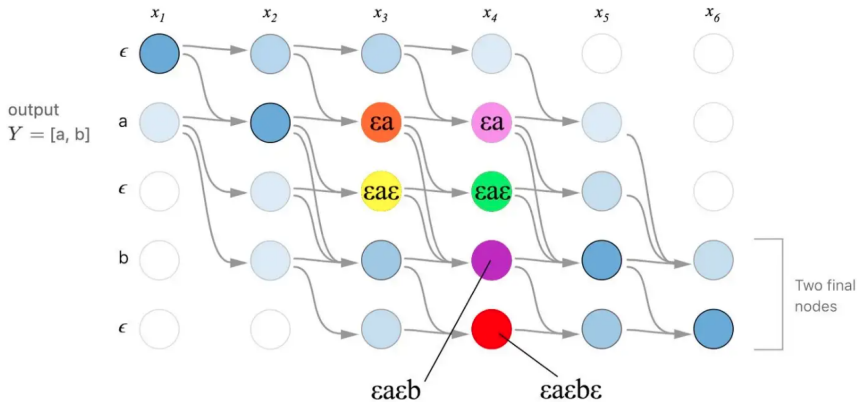
computing the probability for a single alignment step-by-step.

Агрегация по выравниваниям



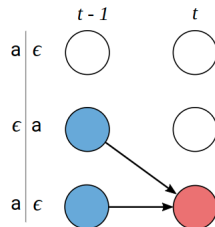
- По строкам $Z = [\epsilon, y_1, \epsilon, y_2, \epsilon, \dots, \epsilon, y_N, \epsilon]$
 - $Y = \text{'hello'}$ $\rightarrow Z = \epsilon \mathbf{h} \epsilon \epsilon \epsilon \epsilon \epsilon \mathbf{o} \epsilon$
- По столбцам - входы (фреймы аудио)
- $\alpha_{s,t} = p(Z_{1:s} | X_{1:t})$

Агрегация по выравниваниям



Случай 1

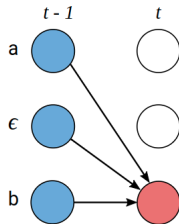
In this case, we can't jump over z_{s-1} , the previous token in Z . The first reason is that the previous token can be an element of Y , and we can't skip elements of Y . Since every element of Y in Z is followed by an ϵ , we can identify this when $z_s = \epsilon$. The second reason is that we must have an ϵ between repeat characters in Y . We can identify this when $z_s = z_{s-2}$.



$$\alpha_{s,t} = (\alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot \underbrace{p(z_s|X)}_{\text{из модели для выхода } t}$$

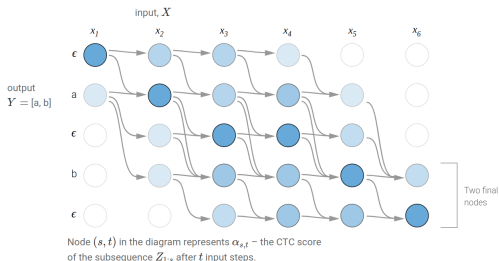
Случай 2

In the second case, we're allowed to skip the previous token in Z . We have this case whenever z_{s-1} is an ϵ between unique characters. As a result there are three positions we could have come from at the previous step.



$$\alpha_{s,t} = (\alpha_{s-2,t-1} + \alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot \underbrace{p(z_s|X)}_{\text{из модели для выхода } t}$$

Пример выравнивания



- Нужно агрегировать по 2м узлам на старте (ϵ, a) и 2м узлам на финише (b, ϵ).
- Можем эффективно вычислить $p(Y|X)$ и настраивать модель $f_\theta : X \rightarrow p(a|X)$ из

$$\sum_{(X,Y) \in \text{TrainSet}} \log p_\theta(Y|X) \rightarrow \max_{\theta}$$

Построение прогноза (наивный подход)

Построение прогноза

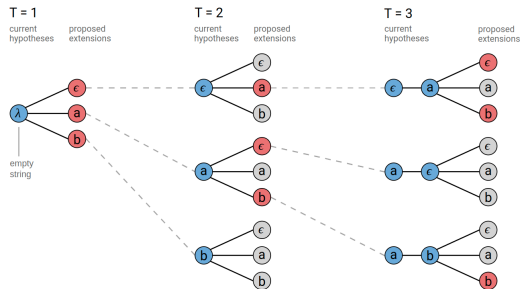
$$\hat{Y} = \arg \max_Y p(Y|X)$$

Простой прогноз:

$$\hat{A} = \arg \max_A \prod_{t=1}^T p(a_t|X)$$

$\hat{A} \rightarrow \hat{Y}$ через CTC преобразование

Построение прогноза (наивный подход)



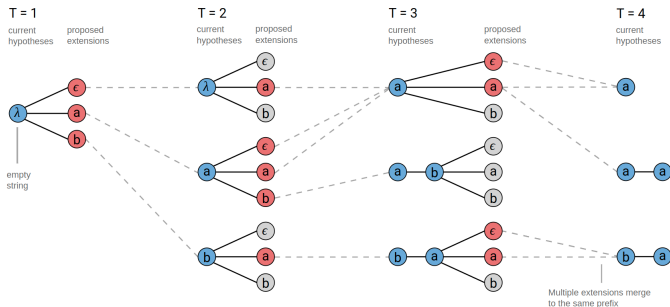
A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

Недостаток простого подхода: разные выравнивания могут соответствовать одному выходу

- например $p(bbb) > p(aa\epsilon)$ и $p(bbb) > p(aaa)$, но $p(aa\epsilon) + p(aaa) > p(bbb)$ и нужно выдавать $\hat{Y} = a$.

Построение прогноза с учетом выходного Y

Поэтому правильнее в лучевом поиске ранжировать не лучшие $A_{:t}$ гипотезы, а лучшие соответствующие $Y_{:s}$ гипотезы



The CTC beam search algorithm with an output alphabet $\{\epsilon, a, b\}$ and a beam size of three.

- Для этого агрегируем

$$\epsilon a + a \epsilon + a a \rightarrow a, \epsilon a a + \epsilon a \epsilon \rightarrow a, b a a + b a \epsilon \rightarrow b a, \dots$$

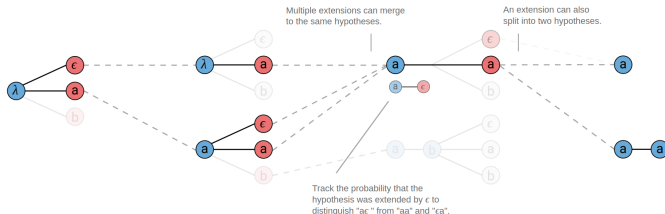
Построение прогноза с учетом выходного Y

- $\epsilon a + a\epsilon + aa \rightarrow a$ но внутри нужно разделять 2 случая (и запоминать их вероятности отдельно) $\epsilon a + aa \rightarrow a$ и $a\epsilon \rightarrow a$ т.к. при последующей склейке с a результат различный:

$$\epsilon a + a \rightarrow a, \text{ но } a\epsilon a \rightarrow aa$$

- При склейке же с b результат одинаковый:

$$\epsilon a + b \rightarrow ab \text{ и } a\epsilon b \rightarrow ab$$



Повышение качества прогнозов

Улучшенный рейтинг для \uparrow качества выходов:

$$p(Y|X) \cdot p(Y)^\alpha \cdot |Y|^\beta$$

- $p(Y)$ - языковая модель (можем оценить по большим корпусам текстов)
- $|Y|$ - длина последовательности (иначе поощряются более короткие из-за $p \times p \times p \dots$)

Проверка корректной реализации СТС:

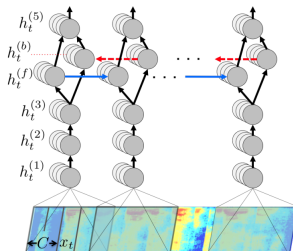
- 1 считаем $p_{true}(Y|X)$ по всем разбиениям
- 2 считаем $p_{СТС,\beta}(Y|X)$ с помощью СТС и лучевого поиска ширины β

Должно получиться:

$$p_{СТС,\beta}(Y|X) \leq p_{true}(Y|X)$$
$$p_{СТС,\beta}(Y|X) \rightarrow p_{true}(Y|X) \text{ при } \uparrow \beta$$

DeepSpeech⁷

- Модель DeepSpeech реализует CTC.
- X_t - окно спектрограммы ($\pm 3,5,7$ фреймов).
- Для прогнозирования $p(a_t|X_t)$ используются
 - 3 FC, left-to-right и right-to-left RNN, 1 FC (от суммы состояний RNN), SoftMax:



⁷<https://arxiv.org/pdf/1412.5567.pdf>

DeepSpeech - особенности

- FC слои можно воспринимать как 1D свёртку
 - т.к. веса FC слоев не зависят от t
- Использовался Dropout на всех FC слоях
 - $p_{drop} = 0.05, 0.1$.
 - но не к пересчету состояний RNN
- Во время обучения и теста использовался ансамбль:

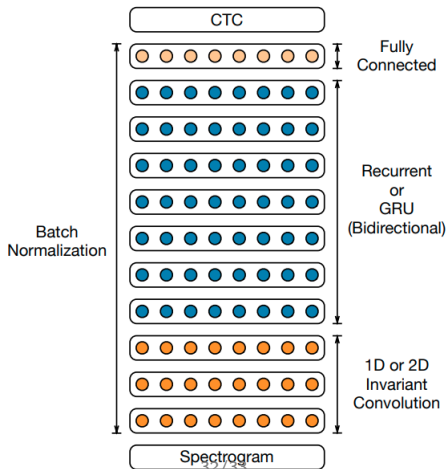
$$\frac{1}{3} (F(x_{-5ms}) + F(x) + F(x_{+5ms}))$$

- x_{Kms} - сигнал x , сдвинутый на K миллисекунд.
- Нелинейность - Clipped ReLU:

$$\text{ReLU}_{clip}(x) = \min \{ \max \{x, 0\}, 20 \}$$

DeepSpeech 2⁸

- DeepSpeech 2 также реализует CTC.
- Для прогнозирования $p(a_t|X_t)$ используются



Заключение

- Распознавание речи основано на
 - seq2seq+attention: Listen-Attend-Spell (LAS)
 - потери CTC: DeepSpeech, DeepSpeech 2.
- качество \uparrow , если
 - используем языковую модель
 - поощряем длительность y
 - используем аугментацию
 - добавляется эхо
 - учимся на x чуть смещенных по t
 - добавляем посторонний шум