Распознавание речи
Часть 3: Listen, attend & spell

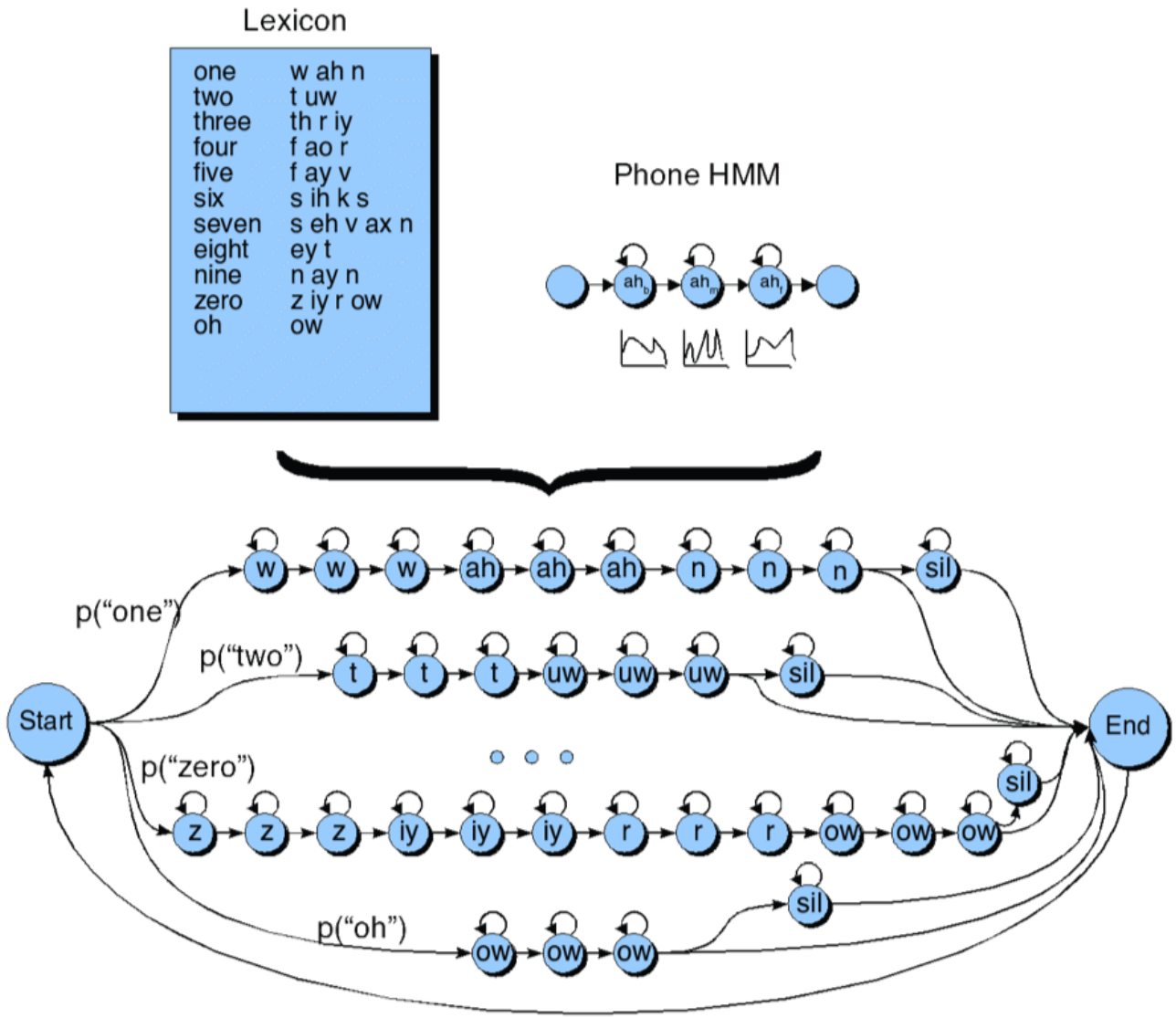
Полыковский Даниил

4 апреля 2018 г.

Краткое содержание предыдущих частей: классический подход

- Выделяем признаки из сигнала: Mel-frequency cepstrum coefficients (MFCC)
- Для каждой фонемы задаем мини-HMM
- Для каждого слова составляем HMM из фонемных мини-HMM (используем фонетический словарь)
- Предсказание: $w = \arg\max p(w|o) = \arg\max p(o|w)p(w)$.

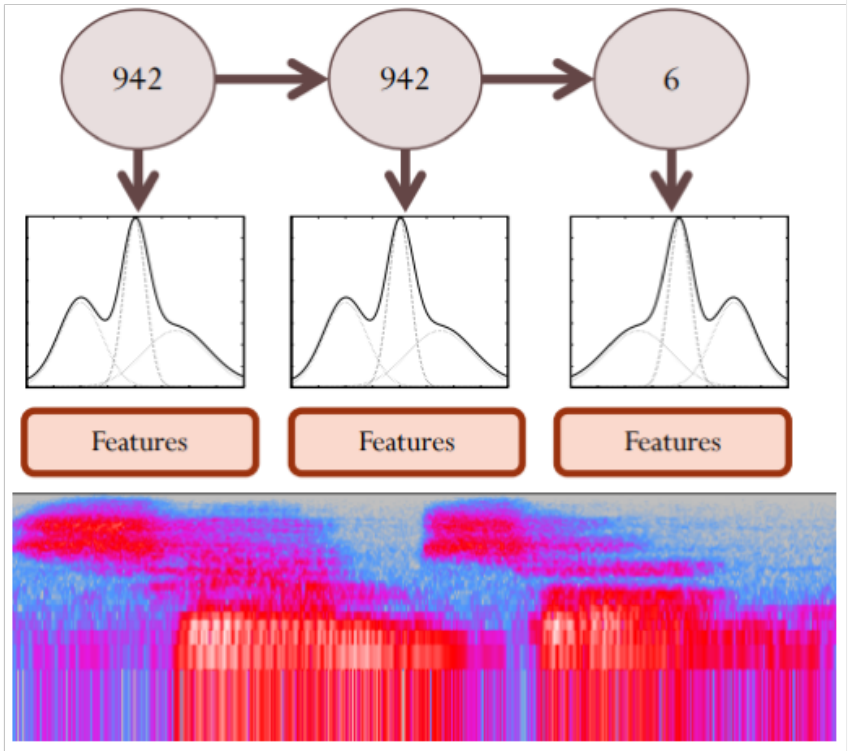
Краткое содержание предыдущих частей: классический подход

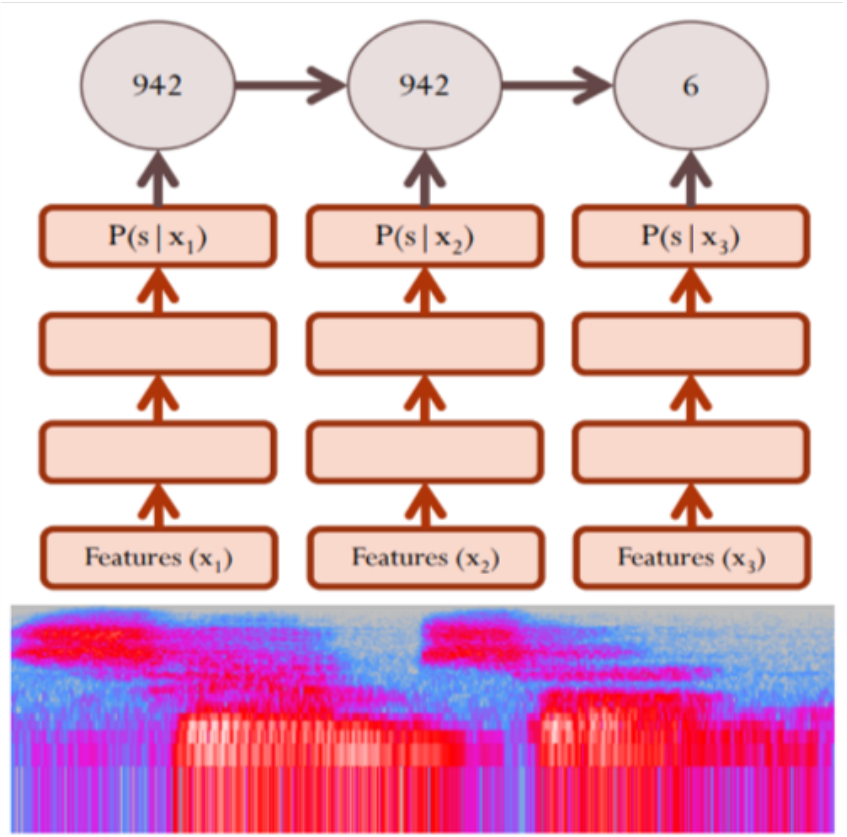


Краткое содержание предыдущих частей: классический подход

- Выделяем признаки из сигнала: Mel-frequency cepstrum coefficients (MFCC)
- Для каждой фонемы задаем мини-HMM
- Для каждого слова составляем HMM из фонемных мини-HMM (используем фонетический словарь)
- Предсказание: $w = \arg\max p(w|o) = \arg\max p(o|w)p(w)$.

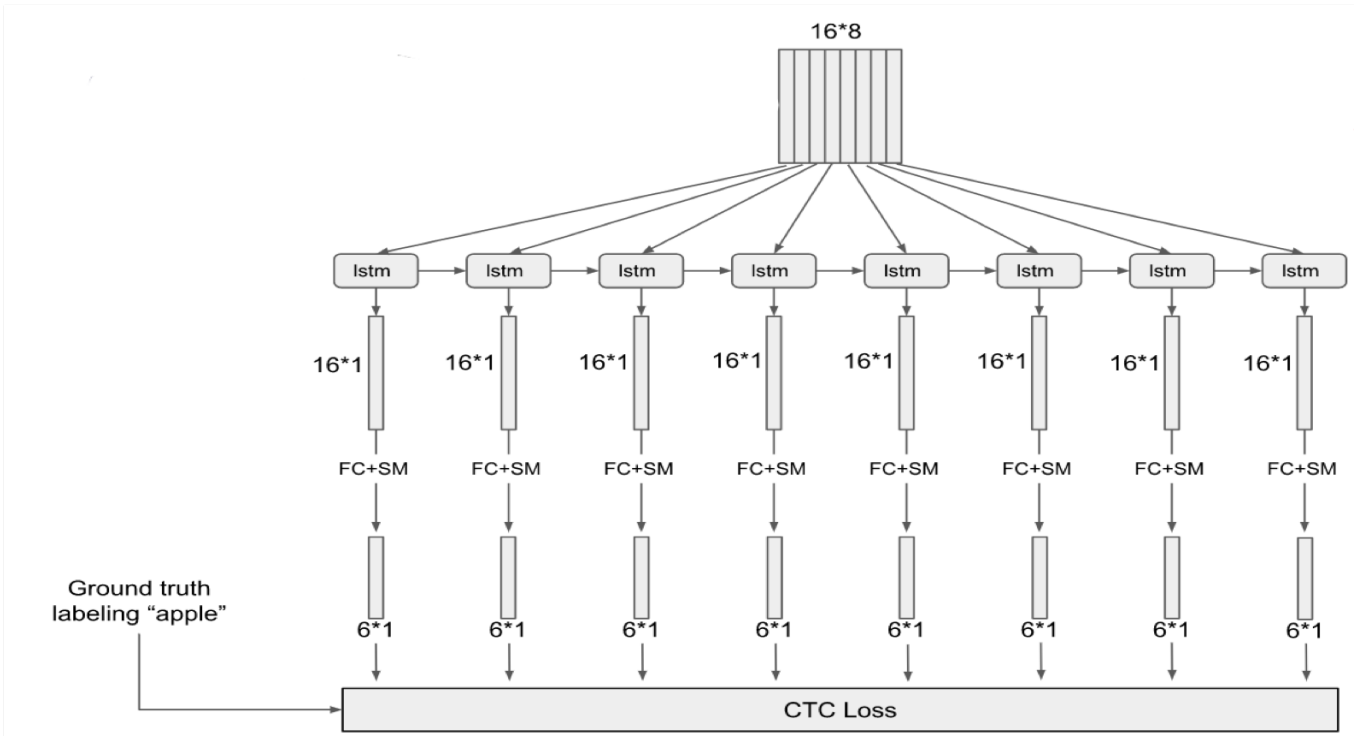
Краткое содержание предыдущих частей: гибридные модели





Краткое содержание предыдущих частей: end-to-end

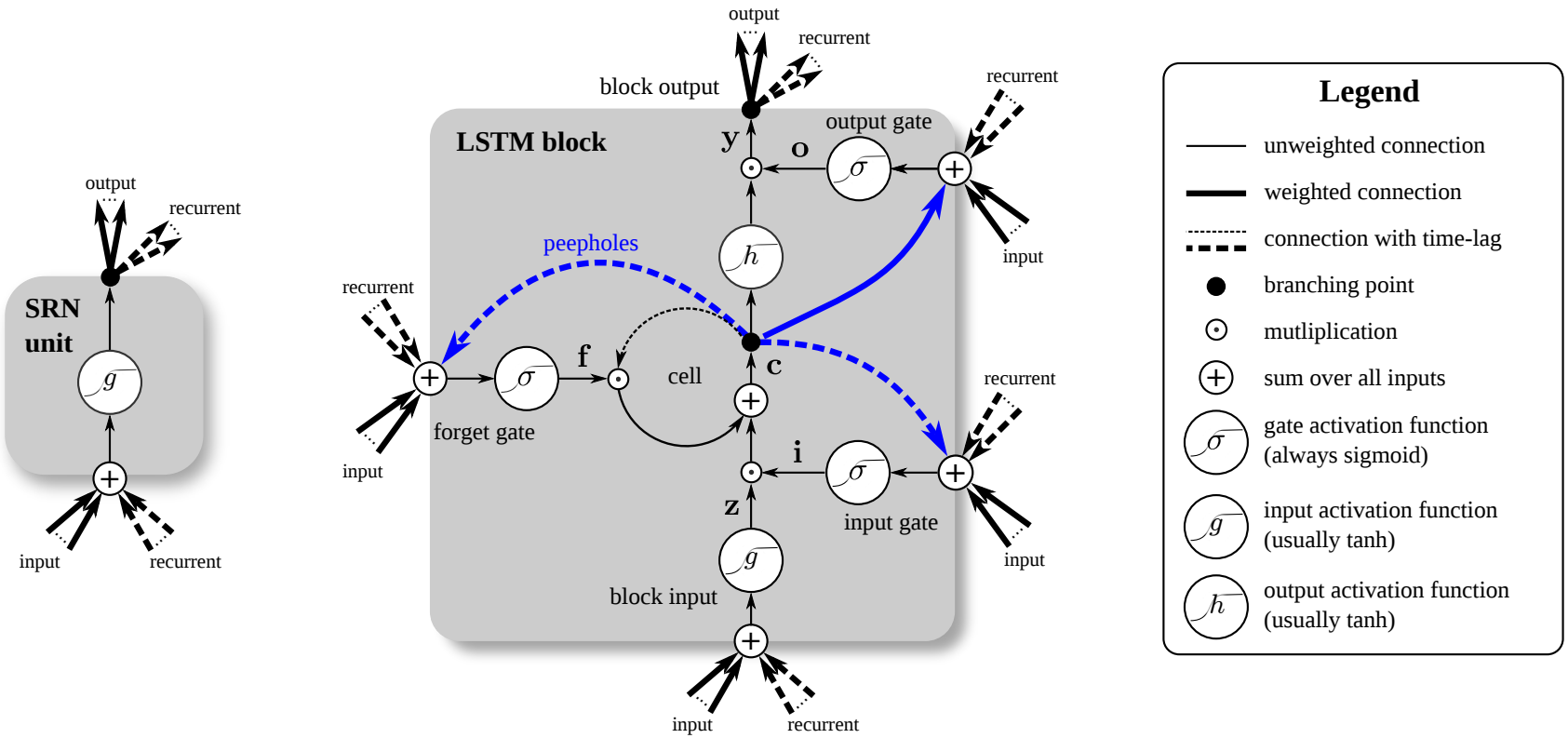
- Connectionist Temporal Classification (CTC)



Сегодняшняя лекция

- Обучение в парадигме encoder-decoder
- Механизмы внимания
- Соберем все в месте в «Listen, attend & spell»
- + бонус: посмотрим код на PyTorch

LSTM сети


output
recurrent
SRN
unit
input
recurrent
output
recurrent
block output
LSTM block
output gate
recurrent
y
o
input
peepholes
recurrent
f
cell
c
recurrent
forget gate
input
i
input gate
z
input
block input
input
recurrent
Legend
unweighted connection
weighted connection
connection with time-lag
branching point
mutliplication
sum over all inputs
gate activation function (always sigmoid)
input activation function (usually tanh)
output activation function (usually tanh)

Вероятностная постановка

В обучающей выборке — аудиофайлы (MFCC признаки)
$\mathbf{O} = (o_1, o_2, \ldots, o_n)$ и соответствующие транскрипции
$\mathbf{W} = (w_1, w_2, \ldots, w_m)$.

Задача: оценить распределение
$P(w_1, w_2, \ldots, w_m | o_1, o_2, \ldots, o_n)$.

- Chain rule $P(w_1, w_2, \ldots, w_m | \mathbf{O}) = \prod_{i=1}^{n} P(w_i | w_1, w_2, \ldots, w_{i-1}, \mathbf{O})$

Вероятностная постановка

В обучающей выборке — аудиофайлы (MFCC признаки) $\mathbf{O} = (o_1, o_2, \ldots, o_n)$ и соответствующие транскрипции $\mathbf{W} = (w_1, w_2, \ldots, w_m)$.

Задача: оценить распределение $P(w_1, w_2, \ldots, w_m | o_1, o_2, \ldots, o_n)$.

- Chain rule $P(w_1, w_2, \ldots, w_m|\mathbf{O}) = \prod_{i=1}^{n} P(w_i|w_1, w_2, \ldots, w_{i-1}, \mathbf{O})$
- Encoder-decoder

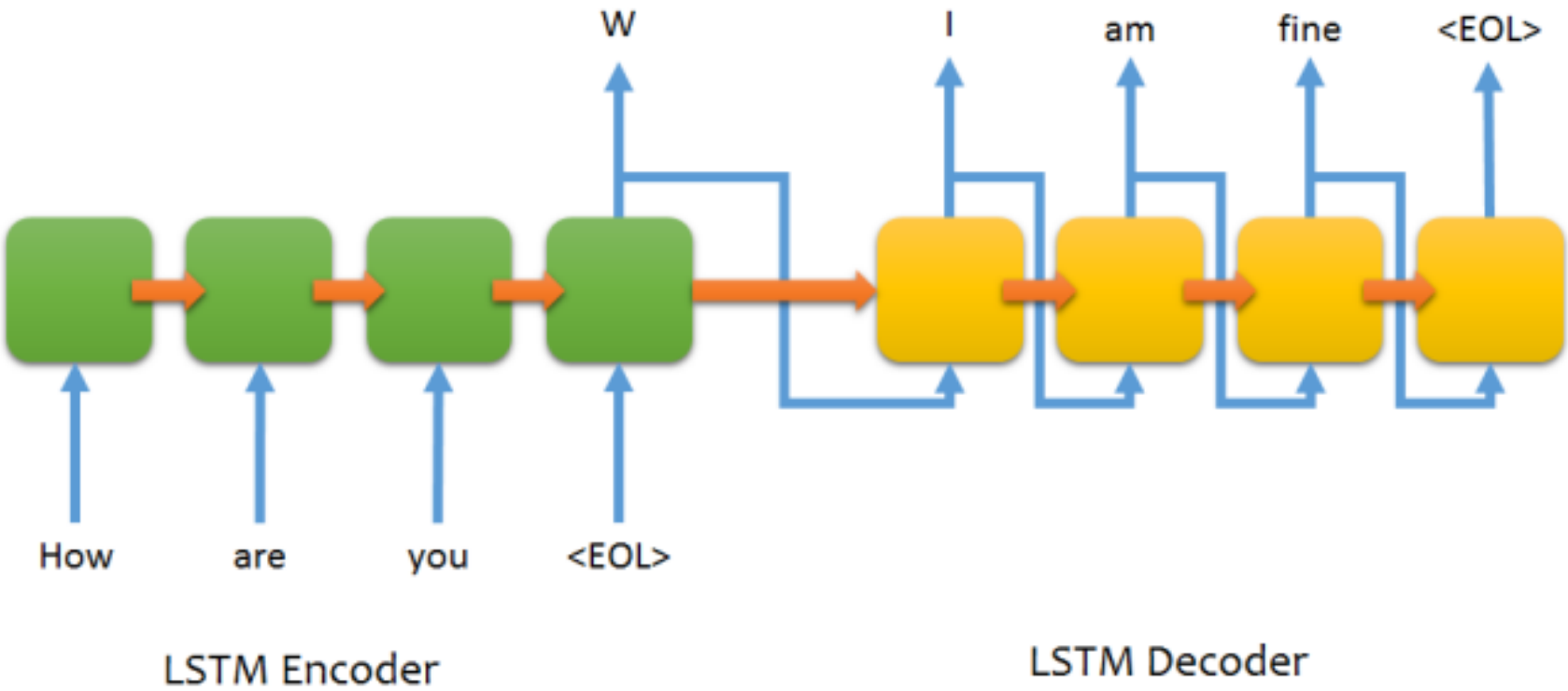
$$h_0 = E(\mathbf{O})$$

$$h_i = H(w_{i-1}, h_{i-1})$$

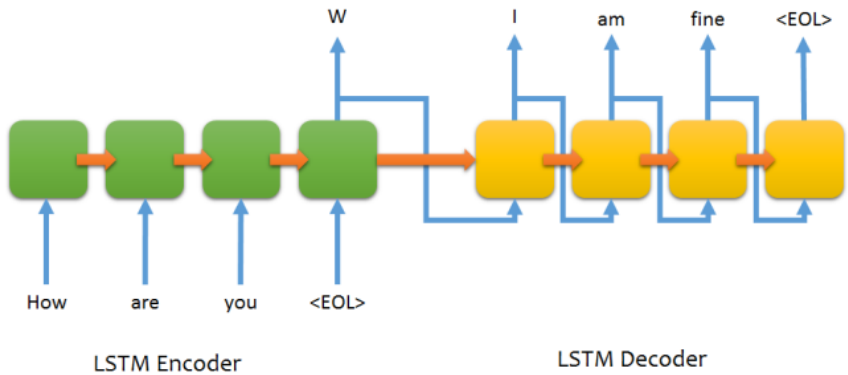
$$P(w_i|w_1, w_2, \ldots, w_{i-1}, \mathbf{O}) \approx D(w_i|w_{i-1}, h_{i-1})$$

Encoder-Decoder модель (Sequence to sequence)

$$\sum_{i=1}^{m} \log P(w_i | w_1, w_2, \ldots, w_{i-1}, \mathbf{O}) \to \max_{E,D}$$

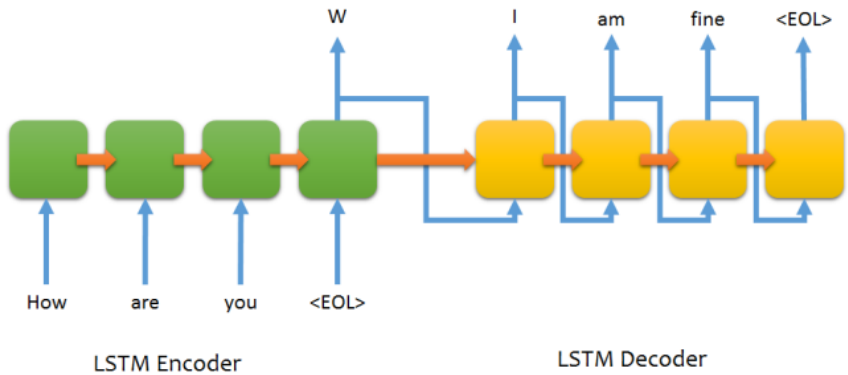


Teacher forcing



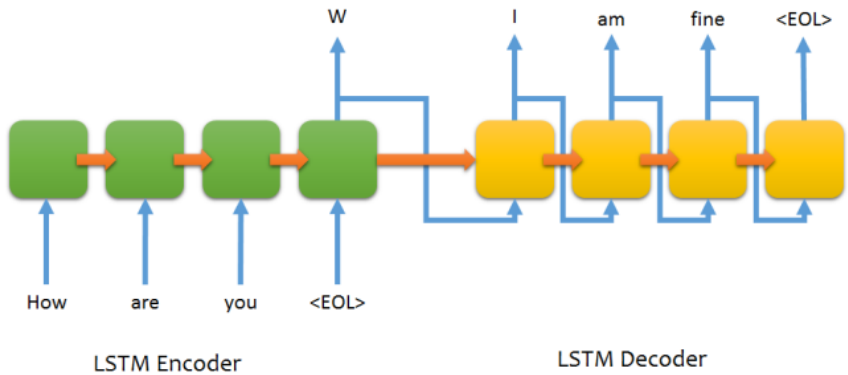
- На каждой итерации decoder возвращает распределение вероятностей над очередным символом.
- Надо как-то сообщать сети о том, какой символ был выбран

Teacher forcing



- На каждой итерации decoder возвращает распределение вероятностей над очередным символом.
- Надо как-то сообщать сети о том, какой символ был выбран
- Free run: семплируем символ из полученного распределения и подаем его на следующей итерации

Teacher forcing



- На каждой итерации decoder возвращает распределение вероятностей над очередным символом.
- Надо как-то сообщать сети о том, какой символ был выбран
- Free run: семплируем символ из полученного распределения и подаем его на следующей итерации
- Teacher forcing: подаем “правильный” символ (w_i)

Teacher forcing

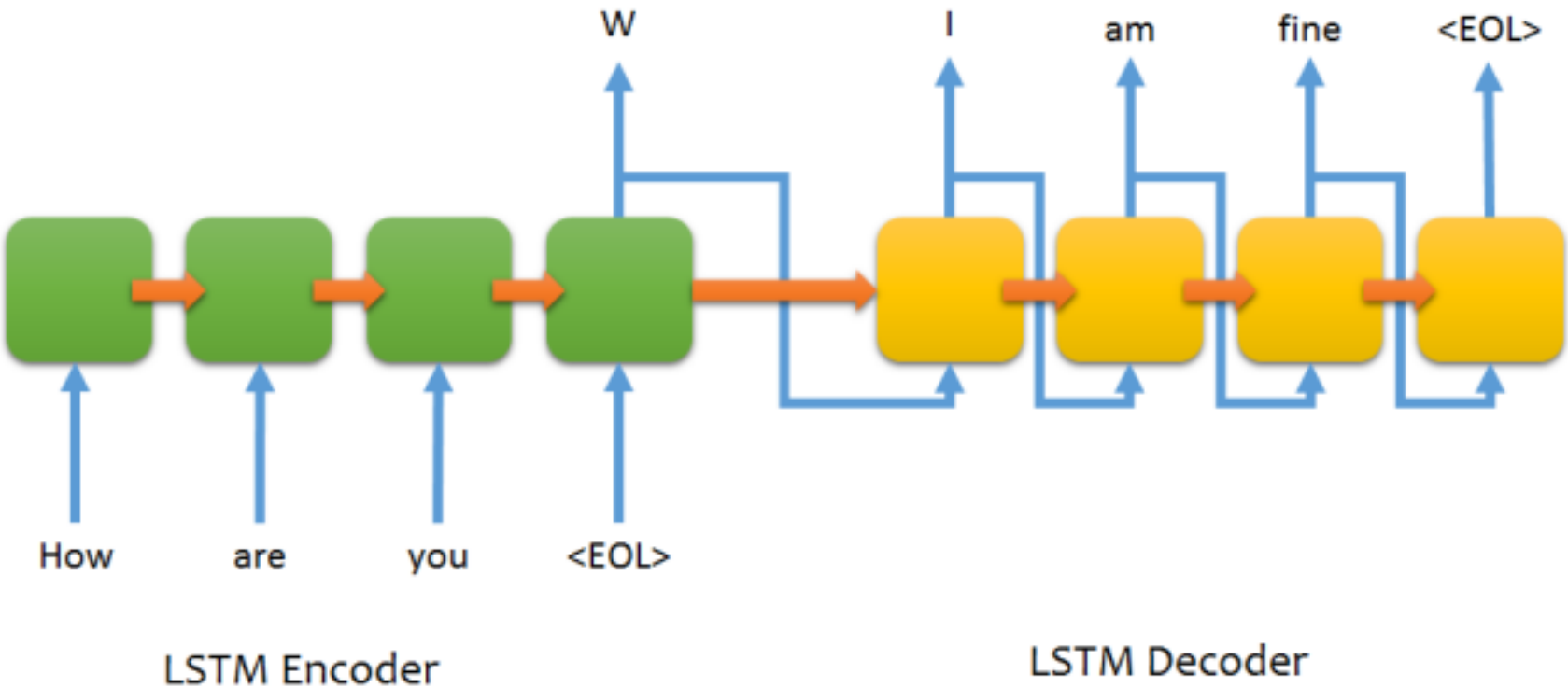
- Free run: получаем неправильный градиент (пропускать градиент через семплирование?), на первых эпохах редко будем получать неправильные семплы — долгое обучение
- Teacher forcing: при предсказании будем работать в ином режиме — непредсказуемое поведение

Teacher forcing

- Free run: получаем неправильный градиент (пропускать градиент через семплирование?), на первых эпохах редко будем получать неправильные семплы — долгое обучение
- Teacher forcing: при предсказании будем работать в ином режиме — непредсказуемое поведение
- Комбинированный режим: с вероятностью 90% применяем teacher forcing, с вероятностью 10% — free run

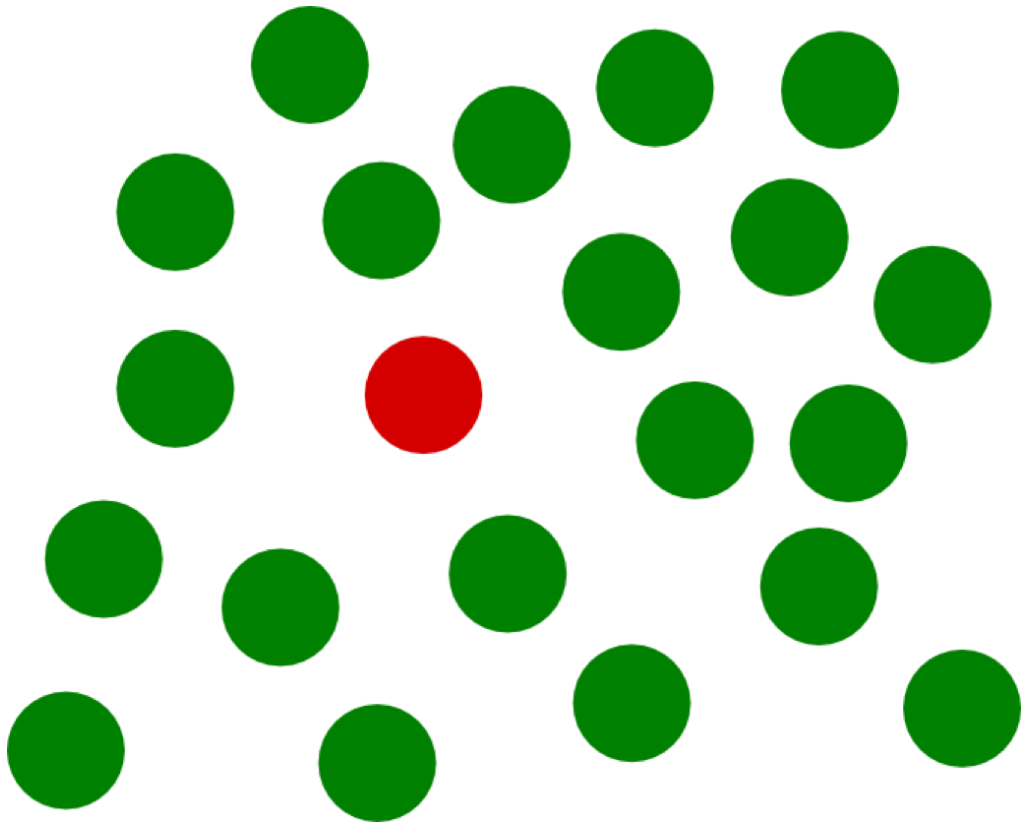
Проблемы Seq2Seq

Приходится сжимать всю информацию об аудиозаписи в один вектор. В результате теряем часть важной для распознавания информации.

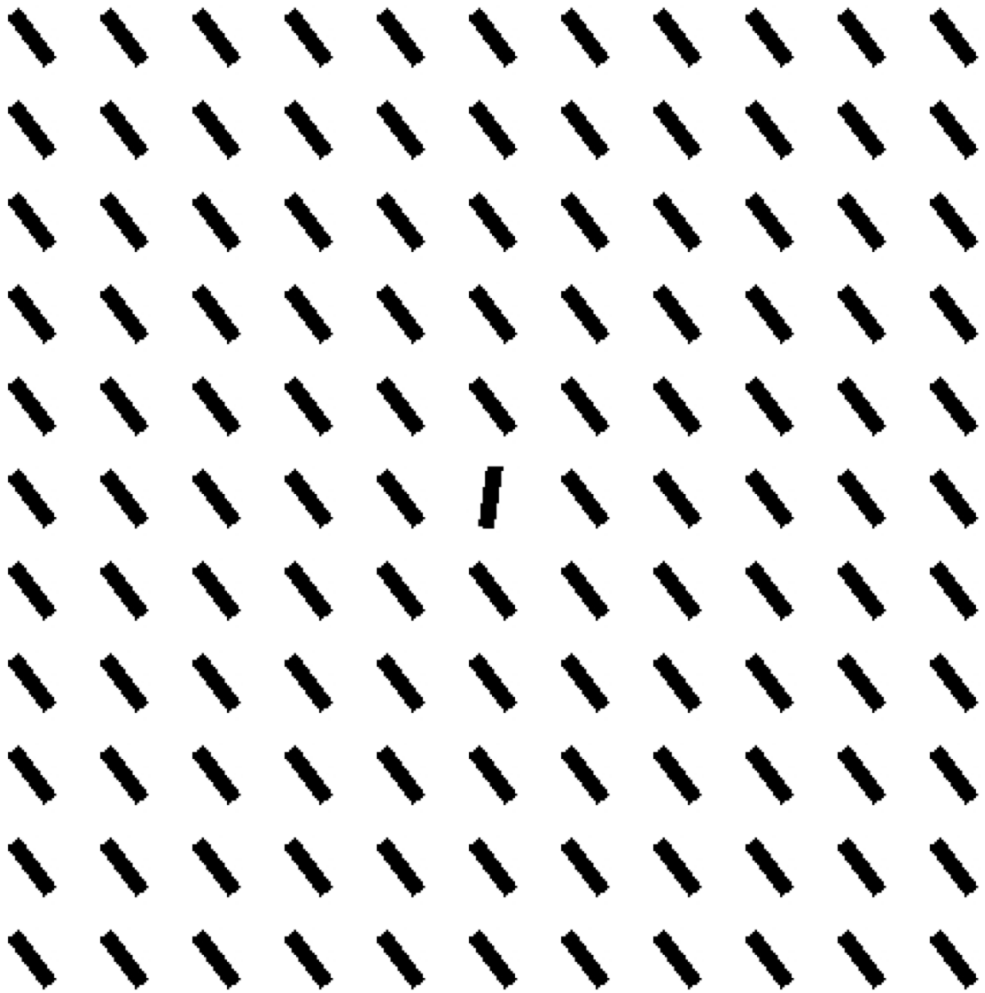


Механизмы внимания

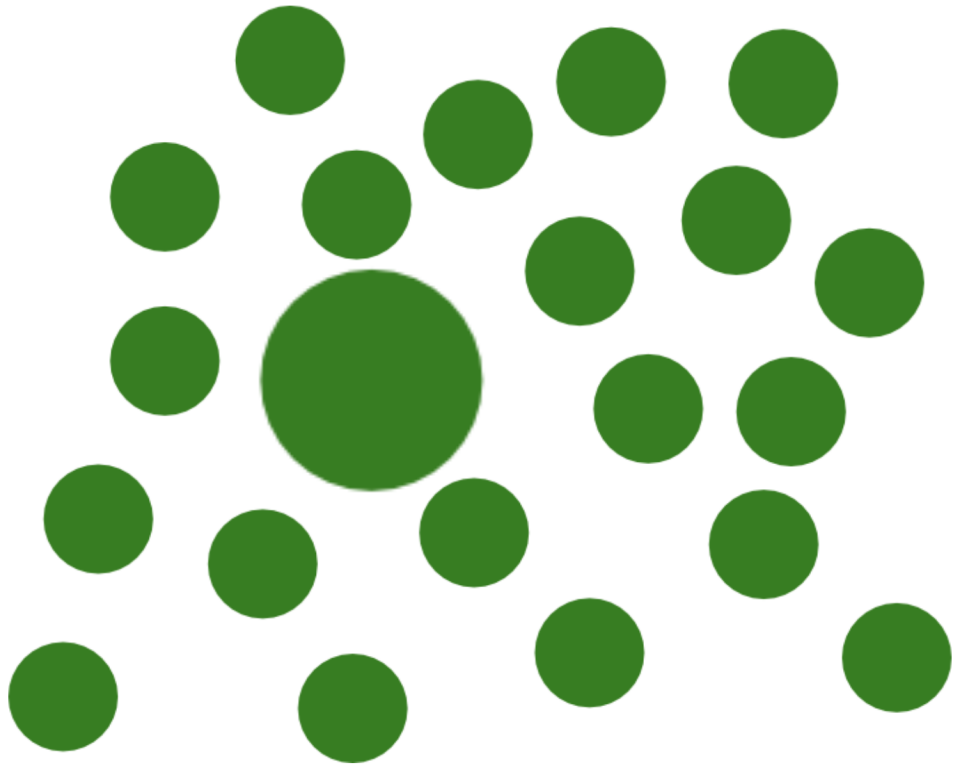
Выделяющиеся области



Выделяющиеся области



Выделяющиеся области



Основная идея

- Вспомогательная сеть генерирует распределение над объектами внимания:
- $e_i = f_{att}(x)_i, \quad a_i = \frac{e^{e_i}}{\sum_j e^{e_j}}$
- Q = “The red boat sank.”
  A = “What color is the boat?”
  - The, p=0.033
  - red, p=0.9
  - boat, p=0.033
  - sank, p=0.034

Основная идея

- Вспомогательная сеть генерирует распределение над объектами внимания:
- $e_i = f_{att}(x)_i, \quad a_i = \frac{e^{e_i}}{\sum_j e^{e_j}}$
- Q = “The red boat sank.”
  A = “What color is the boat?”
  - The, p=0.033
  - red, p=0.9
  - boat, p=0.033
  - sank, p=0.034
- Выбираем объект согласно вероятности
- Например, выбрали “red” (p=0.9): $y = f(\text{red})$
- Как считать градиент?

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x \sim p(x)} f_\theta(x)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x \sim p(x)} f_\theta(x) \simeq \frac{1}{N} \sum_{x_1, \ldots, x_N \sim p(x)} f_\theta(x_i)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N} \sum_{x_1,\ldots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N} \sum_{x_1,\ldots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Случай 2: $p(x)$ зависит от θ

- $\mathbb{E}_{x\sim p_\theta(x)} f(x)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Случай 2: $p(x)$ зависит от θ

- $\mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p_\theta(x)} f(x_i)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\dots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\dots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Случай 2: $p(x)$ зависит от θ

- $\mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\dots,x_N\sim p_\theta(x)} f(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)} f(x) = \nabla_\theta \int p_\theta(x) f(x) dx = \int \nabla_\theta p_\theta(x) f(x) dx$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Случай 2: $p(x)$ зависит от θ

- $\mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p_\theta(x)} f(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)} f(x) = \nabla_\theta \int p_\theta(x) f(x) dx = \int \nabla_\theta p_\theta(x) f(x) dx$
- $\nabla_\theta p_\theta(x) = p_\theta(x) \nabla_\theta \log p_\theta(x)$
- $\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p_\theta(x)} f(x_i) \nabla_\theta \log p_\theta(x_i)$

Вероятностные интегралы

Случай 1: $p(x)$ не зависит от θ

- $\mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} f_\theta(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p(x)} f_\theta(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p(x)} \nabla_\theta f_\theta(x_i)$

Случай 2: $p(x)$ зависит от θ

- $\mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p_\theta(x)} f(x_i)$
- $\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)} f(x) = \nabla_\theta \int p_\theta(x) f(x) dx = \int \nabla_\theta p_\theta(x) f(x) dx$
- $\nabla_\theta p_\theta(x) = p_\theta(x) \nabla_\theta \log p_\theta(x)$
- $\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)} f(x) \simeq \frac{1}{N}\sum_{x_1,\ldots,x_N\sim p_\theta(x)} f(x_i) \nabla_\theta \log p_\theta(x_i)$

Проблема: большая дисперсия градиентов (в среднем првильно, но каждый раз сильно ошибаемся). Есть более “умные” способы оценки градиента (нпр., gumbel trick).

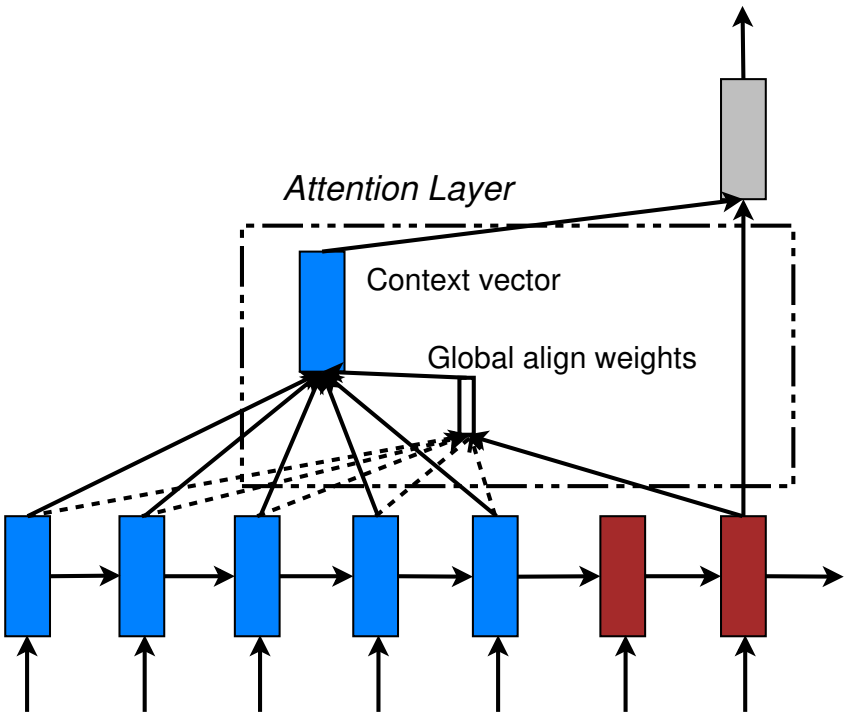
“Мягкое” внимание

- Вспомогательная сеть генерирует распределение над объектами внимания:
- $e_i = f_{att}(x)_i, \quad a_i = \frac{e^{e_i}}{\sum_j e^{e_j}}$
- Пример: The red boat sank.
  - The, p=0.033
  - red, p=0.9
  - boat, p=0.033
  - sank, p=0.034
- Взвешиваем все объекты согласно вероятностям
- $y = \sum a_i \cdot f_i$
- Как считать градиент?

“Мягкое” внимание против “жесткого”

- Жесткое внимание: когда тяжело посчитать результат обращения к объекту (нпр., результат игры)
- Мягкое внимание: когда просто считать объекты

Внимание для Seq2Seq



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

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

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

Listen, attend & spell

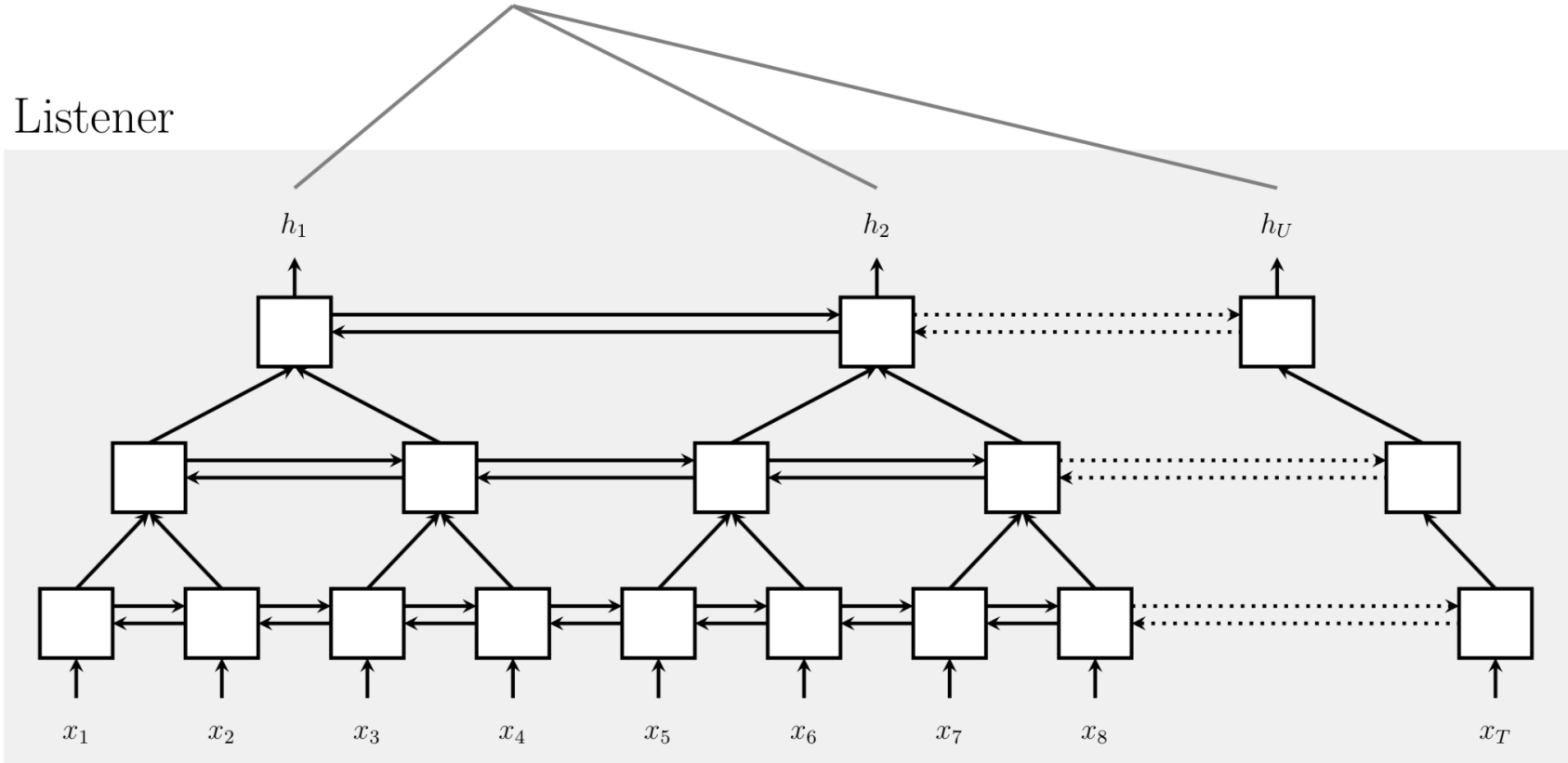
Listen, Attend & Spell (LAS)

Собираем все вместе, чтобы построить модель распознавания речи:

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

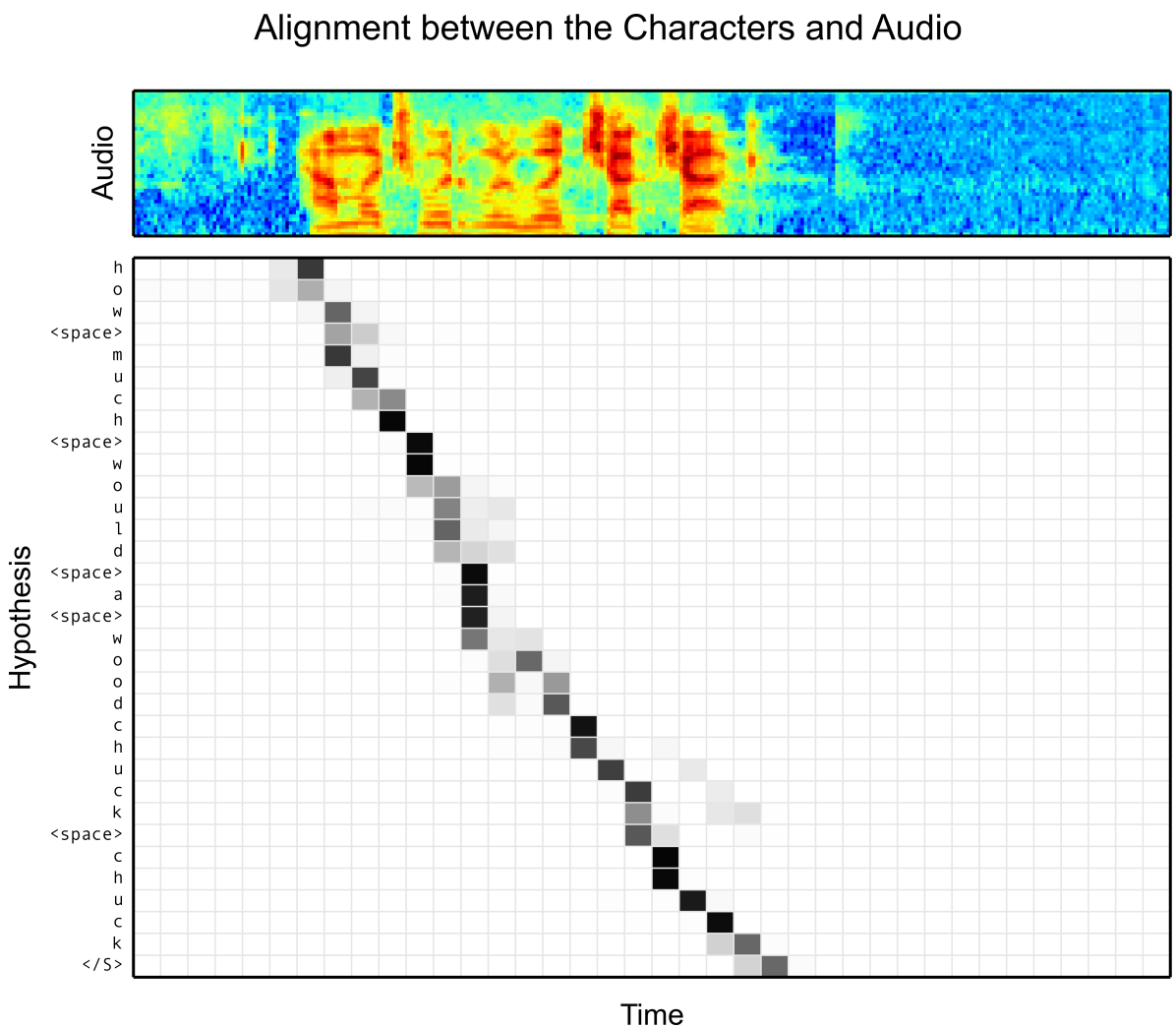
Pyramid BRNN (pBRNN)

Проблема: сложность soft attention — $\mathcal{O}(nm)$. Хотим ускорить в константное число раз.



Строим пирамиду. На каждом уровне пропускаем данные через двунаправленную LSTM и аггрегируем пары соседних состояний. Спустя 3 слоя уменьшаем число состояний в 8 раз.

Выравнивание



Декодирование: beam search

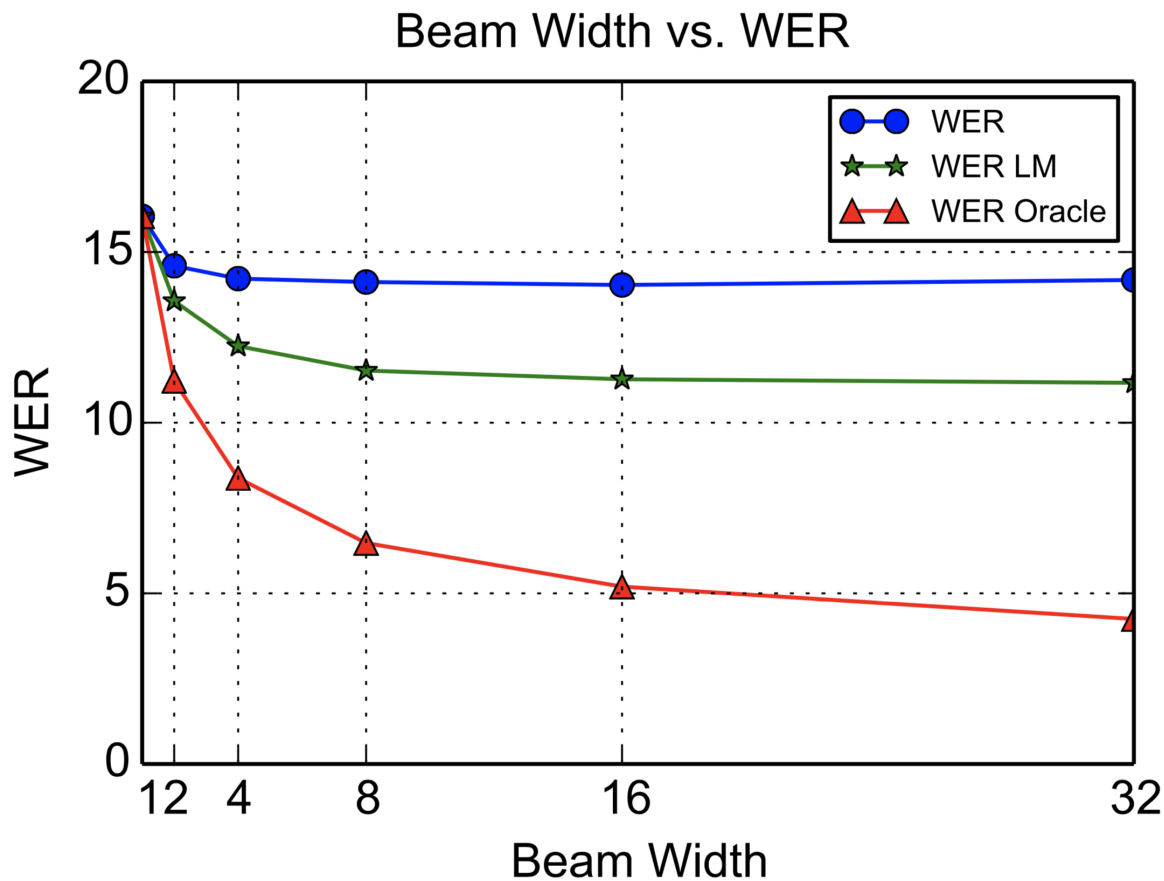
Как выбрать итоговый текст?

- Семплирование: $\mathbf{W} \sim p(\mathbf{W}|\mathbf{O})$, $w_i \sim p(w_i|w_1, \ldots, w_{i-1})$
- Наиболее вероятное слово: $\mathbf{W} = \arg\max p(\mathbf{W}|\mathbf{O})$.

Во втором случае не так тривиально:
$w_i = \arg\max p(w_i|w_1, \ldots, w_{i-1})$ не даст глобальный $\arg\max$.

Декодирование: beam search

1. Храним β лучших префиксов
2. На очередной итерации пробуем дописать по одному символу для каждого префикса
3. Из новых строк оставляем только β лучших



Декодирование: beam search

Beam	Text	Log Probability	WER
Truth	call aaa roadside assistance	-	-
1	call aaa roadside assistance	-0.5740	0.00
2	call triple a roadside assistance	-1.5399	50.00
3	call trip way roadside assistance	-3.5012	50.00
4	call xxx roadside assistance	-4.4375	25.00

150 строк на PyTorch

https://github.com/XenderLiu/Listen-Attend-and-Spell-Pytorch/blob/master/model/las_model.py