



Deep Learning School

# Машинный перевод, Архитектура Seq2Seq Механизм Attention

# План занятия

- Задача машинного перевода;
- Архитектура Seq2Seq для решения задачи перевода и ее обучение;
- Недостатки RNN-модели для решения задачи перевода
- Механизм внимания (Attention)

# В этом видео

- **Задача машинного перевода;**
- **Архитектура Seq2Seq для решения задачи перевода и ее обучение;**
- **Недостатки RNN-модели для решения задачи перевода**
- **Механизм внимания (Attention)**

# Задача машинного перевода

Предложение на английском и перевод на русский:

<i>The cat is pretty</i>	$source = (x_1, x_2, \dots, x_n)$
<i>Кошечка милая</i>	$target = (y_1, y_2, \dots, y_m)$

Задача машинного перевода — найти наиболее вероятную последовательность токенов перевода на target языке при условии заданной последовательности токенов на source языке:

$$\widehat{target} = \underset{target}{argmax} P(target | source, \theta)$$

# Задача машинного перевода

*The cat is pretty*                       $source = (x_1, x_2, \dots, x_n)$   
*Кошечка милая*                       $target = (y_1, y_2, \dots, y_m)$

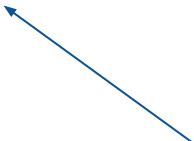
$$P(target \mid source) = P(y_1 \mid source) \cdot P(y_2 \mid y_1, source) \dots \\ \dots P(y_m \mid y_1, \dots, y_{m-1}, source)$$

$$\widehat{target} = \underset{target}{\operatorname{argmax}} \quad P(target \mid source, \theta)$$

# Задача машинного перевода

Задача машинного перевода — это задача  
*Conditional Language Modeling*

$$P(target \mid source) = P(y_1 \mid source) \cdot P(y_2 \mid y_1, source) \dots \\ \dots P(y_m \mid y_1, \dots, y_{m-1}, source)$$

$$\widehat{target} = \underset{target}{argmax} \quad P(target \mid source, \theta)$$


# Задача машинного перевода

Эволюция подходов к решению задачи машинного перевода:

- Правильный МТ (1950-е)
- Phrase-based/статистический МТ (1990-е)
- МТ на основе нейросетей (2010-е)

Хорошая и интересная статья об истории машинного перевода: [link](#)

# RNN для машинного перевода

Давайте построим RNN-модель для решения задачи перевода.

Так как машинный перевод — это задача *Conditional Language Modeling*, то архитектура нейросети будет похожа на архитектуру сети для языкового моделирования



# RNN для машинного перевода

Идея устройства RNN-модели для языкового

моделирования:

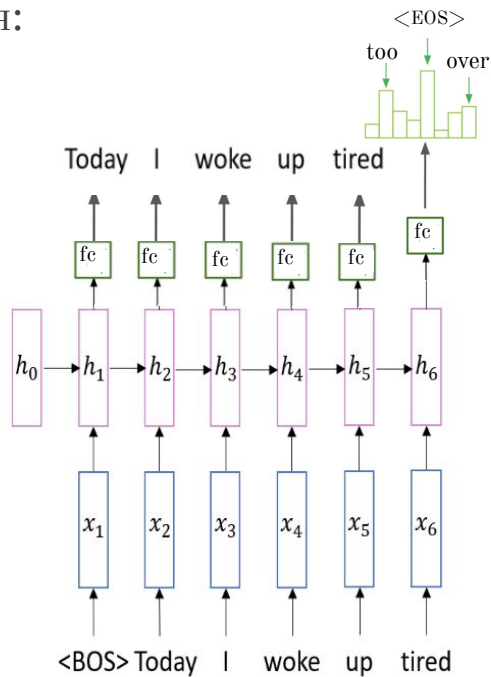
Выбор  
следующего  
токена

Полносвязные  
слои

RNN слои

эмбеддинги

Входящие  
токены

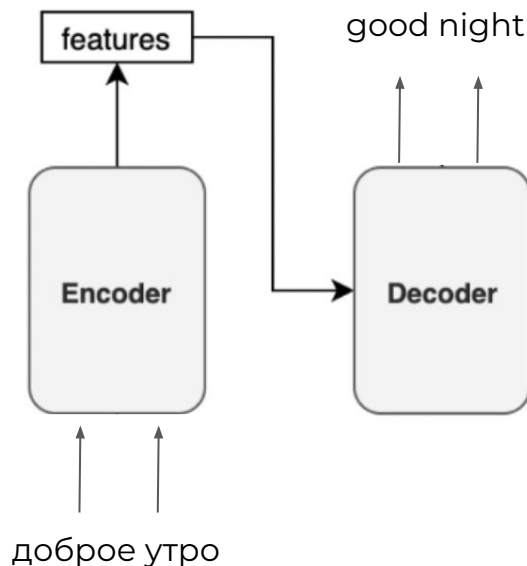


# RNN для машинного перевода

Идея модели:

Нейросеть будет состоять из двух частей: Encoder и Decoder.

- **Encoder** будет принимать на вход предложение на source языке и агрегировать информацию из него;
- **Decoder** будет генерировать предложение-перевод token за tokenом на основе информации, которую передаст ему encoder.

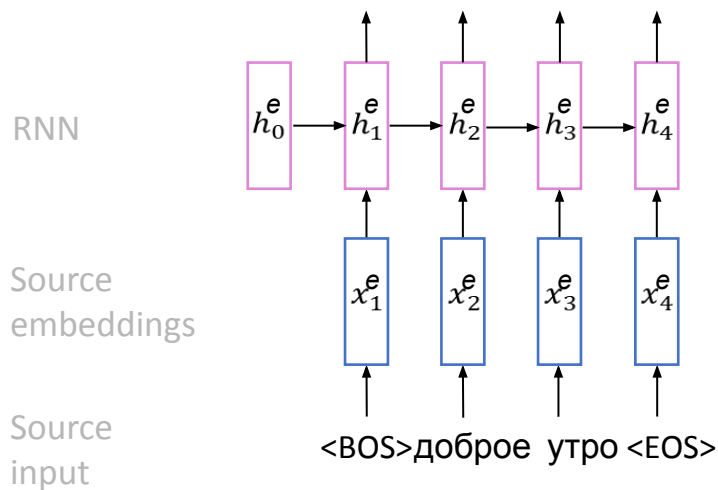


# RNN для машинного перевода

## Encoder

Энкодер состоит из слоя embedding и RNN слоев.

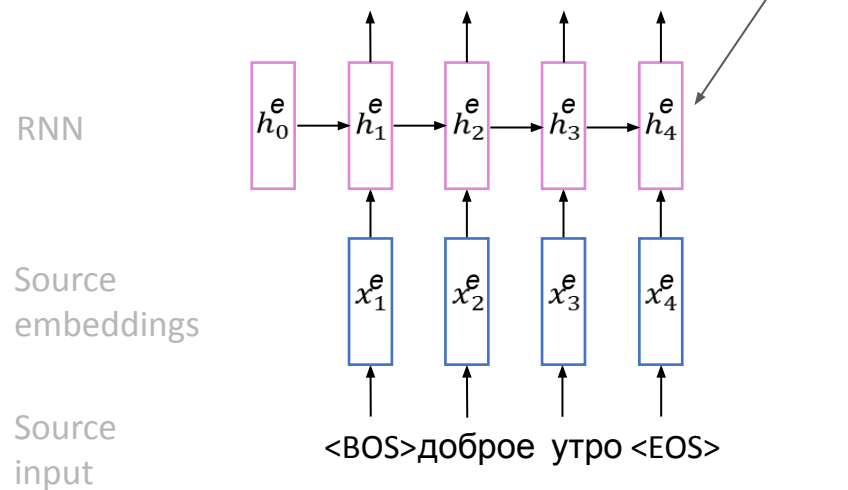
Он “читает” входную последовательность токен за токеном, и обновляет скрытые состояния своих RNN-слоев

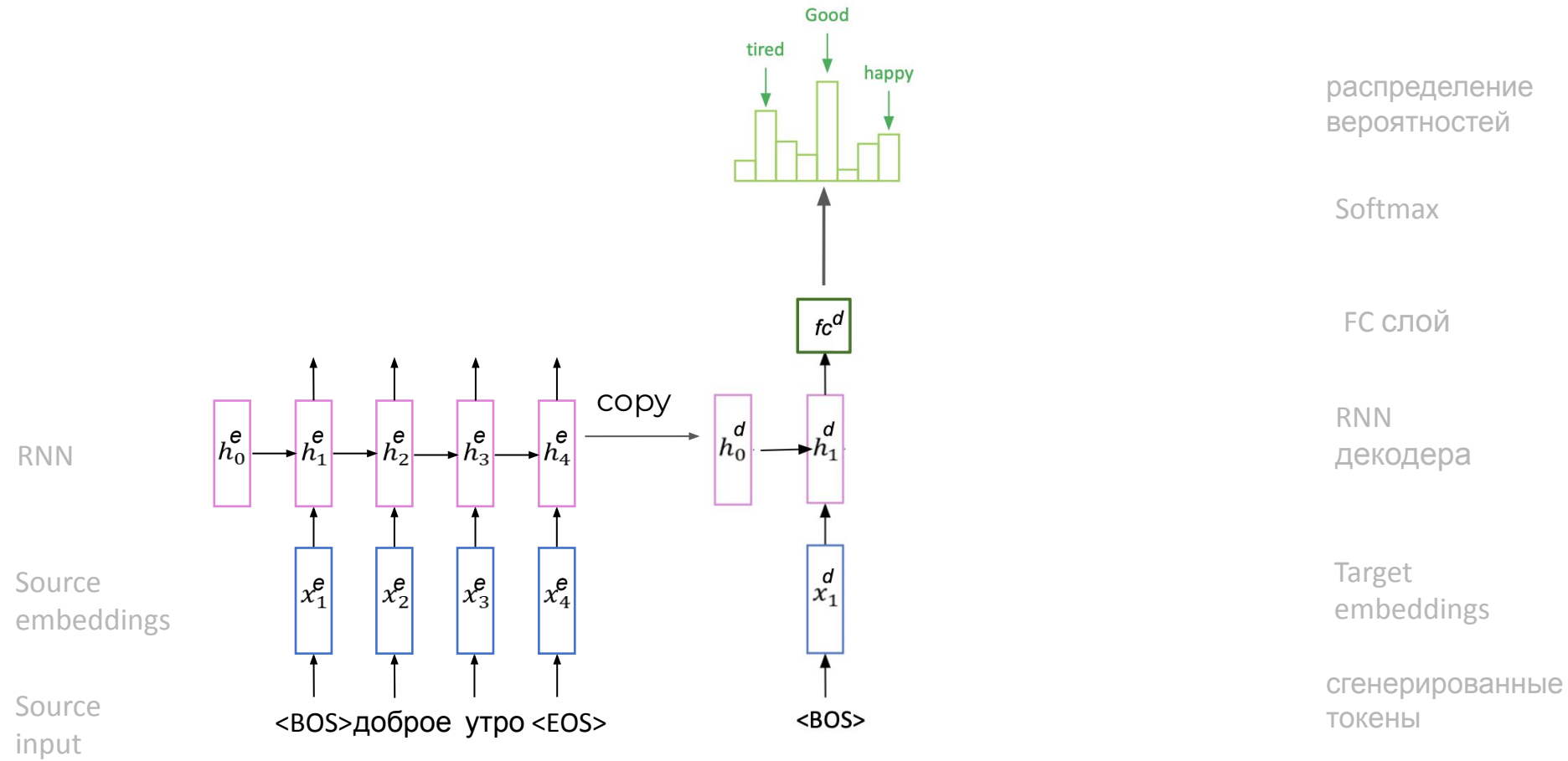


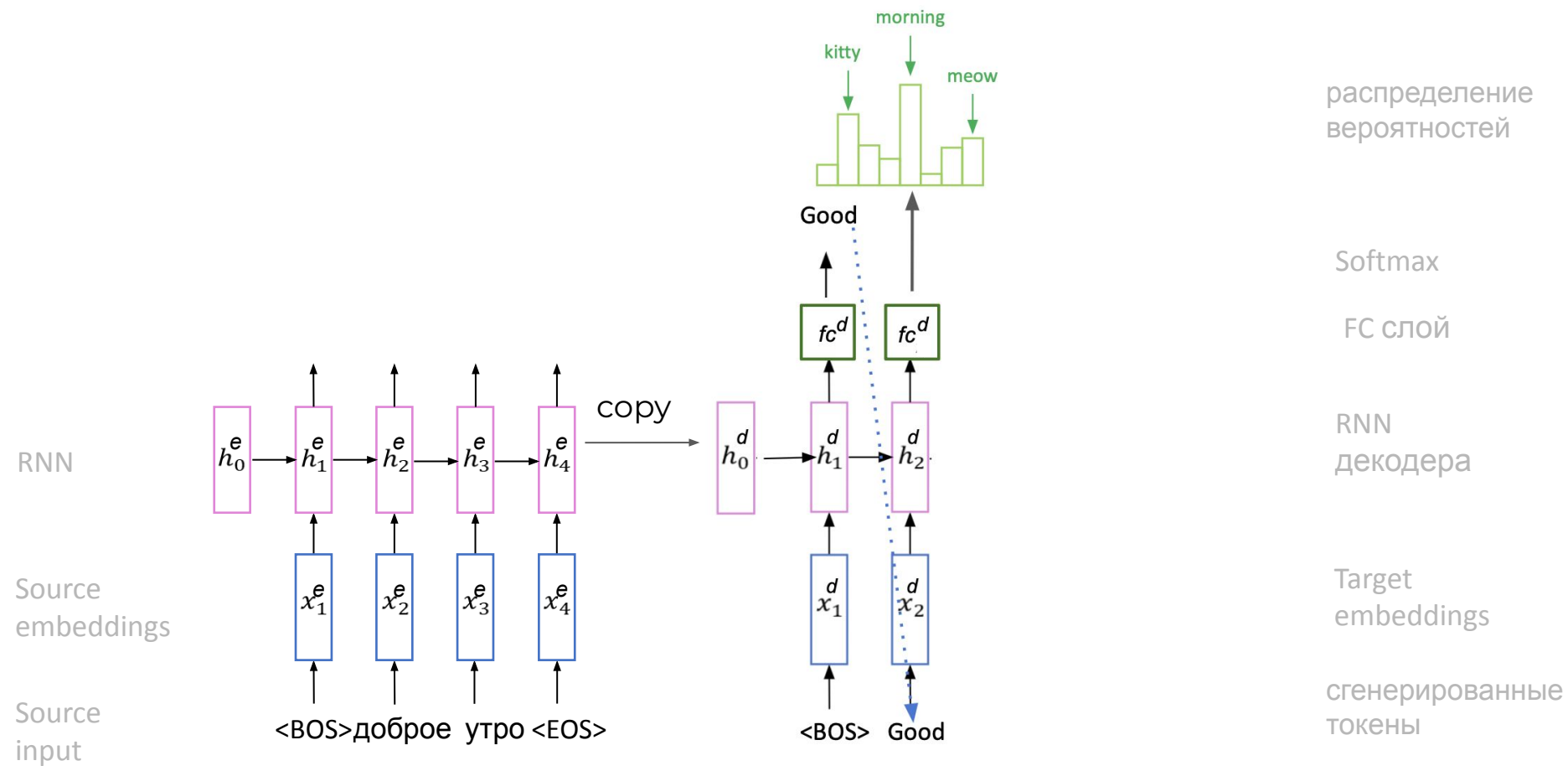
# RNN для машинного перевода

## Encoder

Скрытое состояние RNN в конце обработки предложения будет содержать агрегированную информацию о предложении



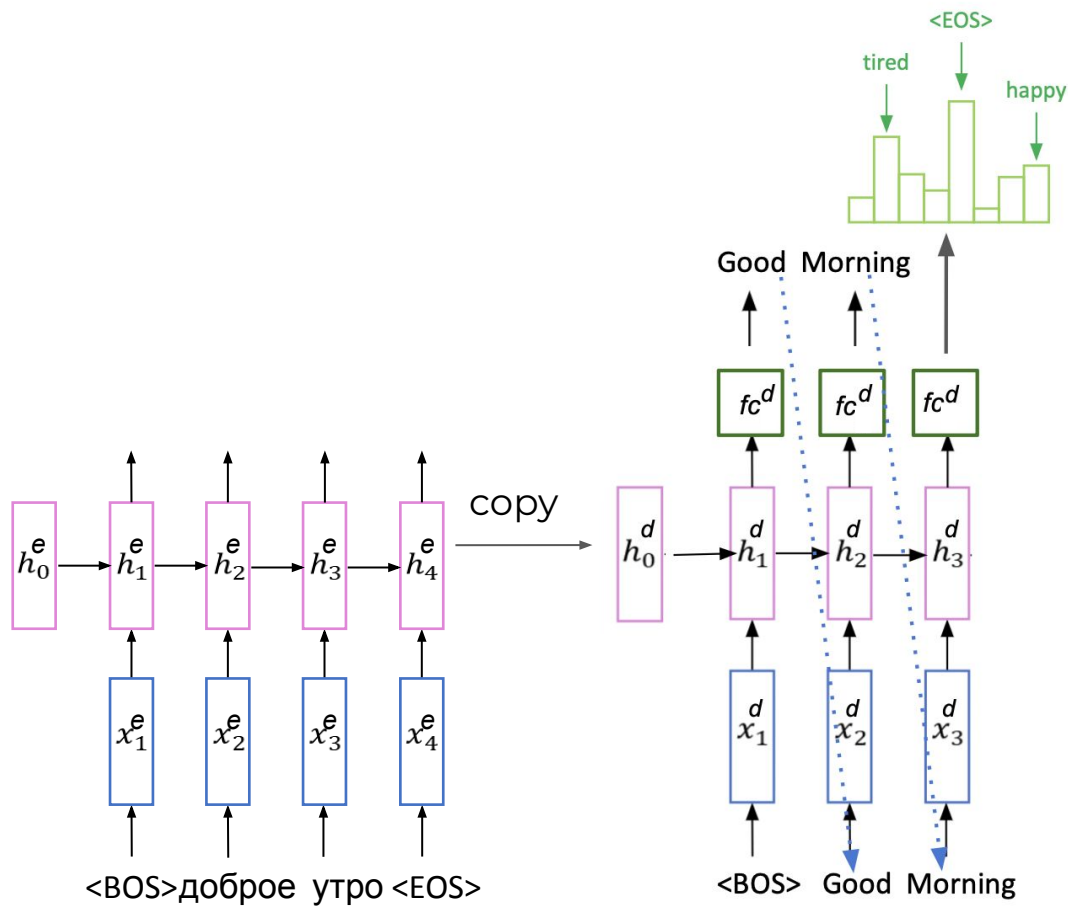




RNN

Source  
embeddings

Source  
input



распределение  
вероятностей

Softmax

FC слой

RNN  
декодера

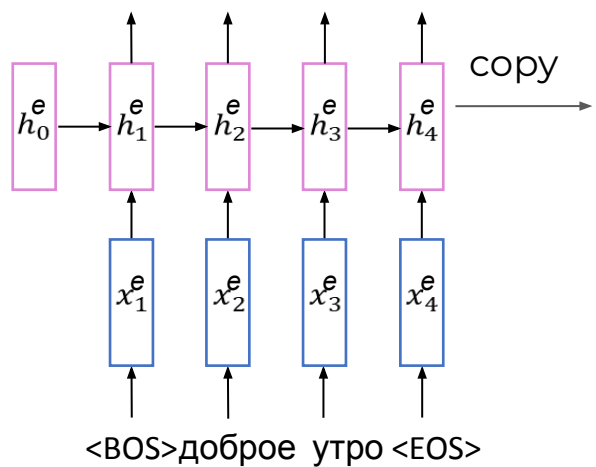
Target  
embeddings

сгенерированные  
токены

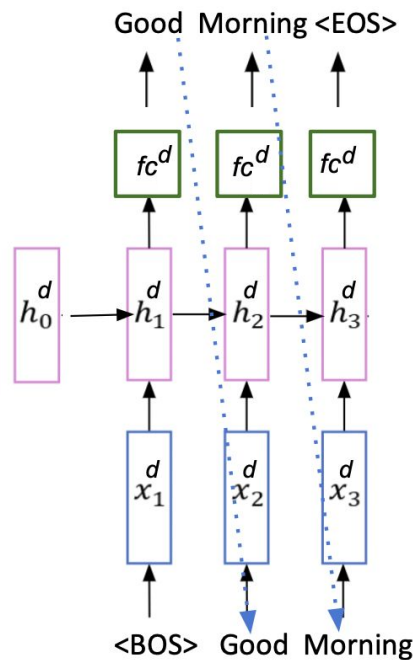
RNN

Source  
embeddings

Source  
input



copy



сгенерированные  
токены

Softmax

FC слой

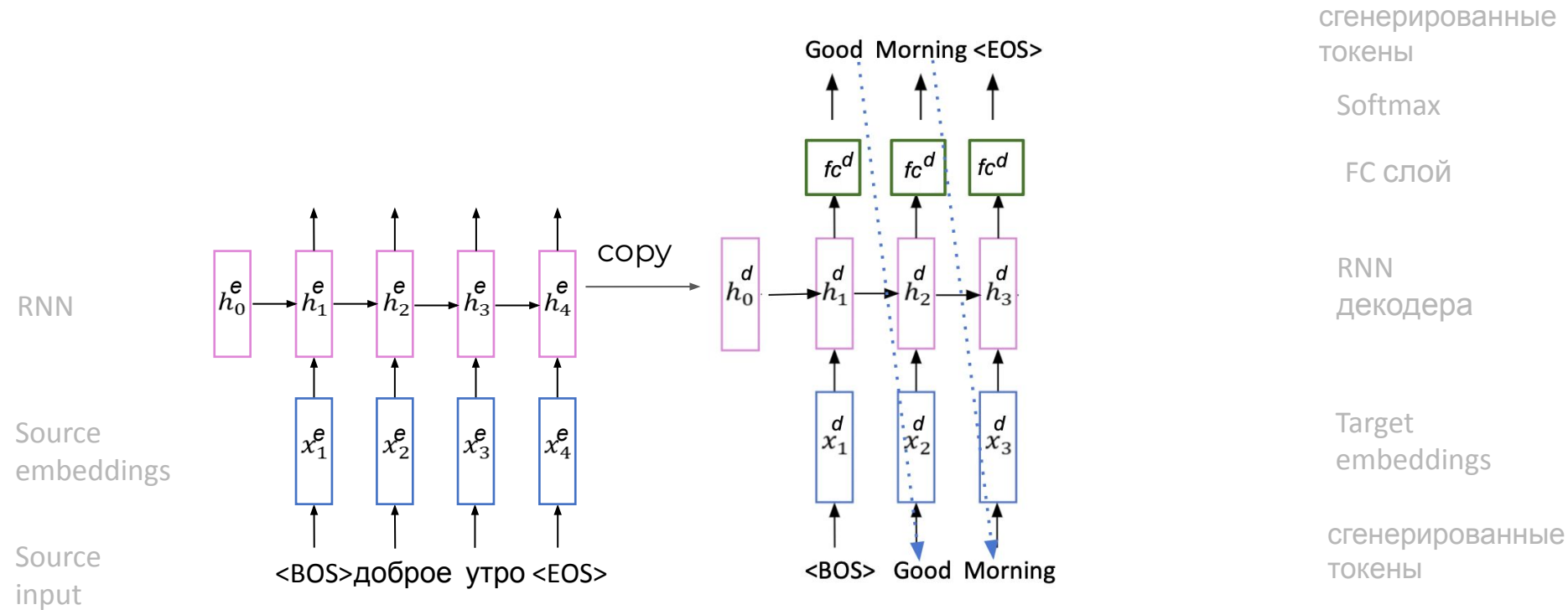
RNN  
декодера

Target  
embeddings

сгенерированные  
токены



Такая архитектура модели  
называется **Seq2Seq**  
(sequence-to-sequence)



# Обучение модели перевода

Чтобы обучить нашу модель, нам нужны данные.

Данные для задачи машинного перевода —

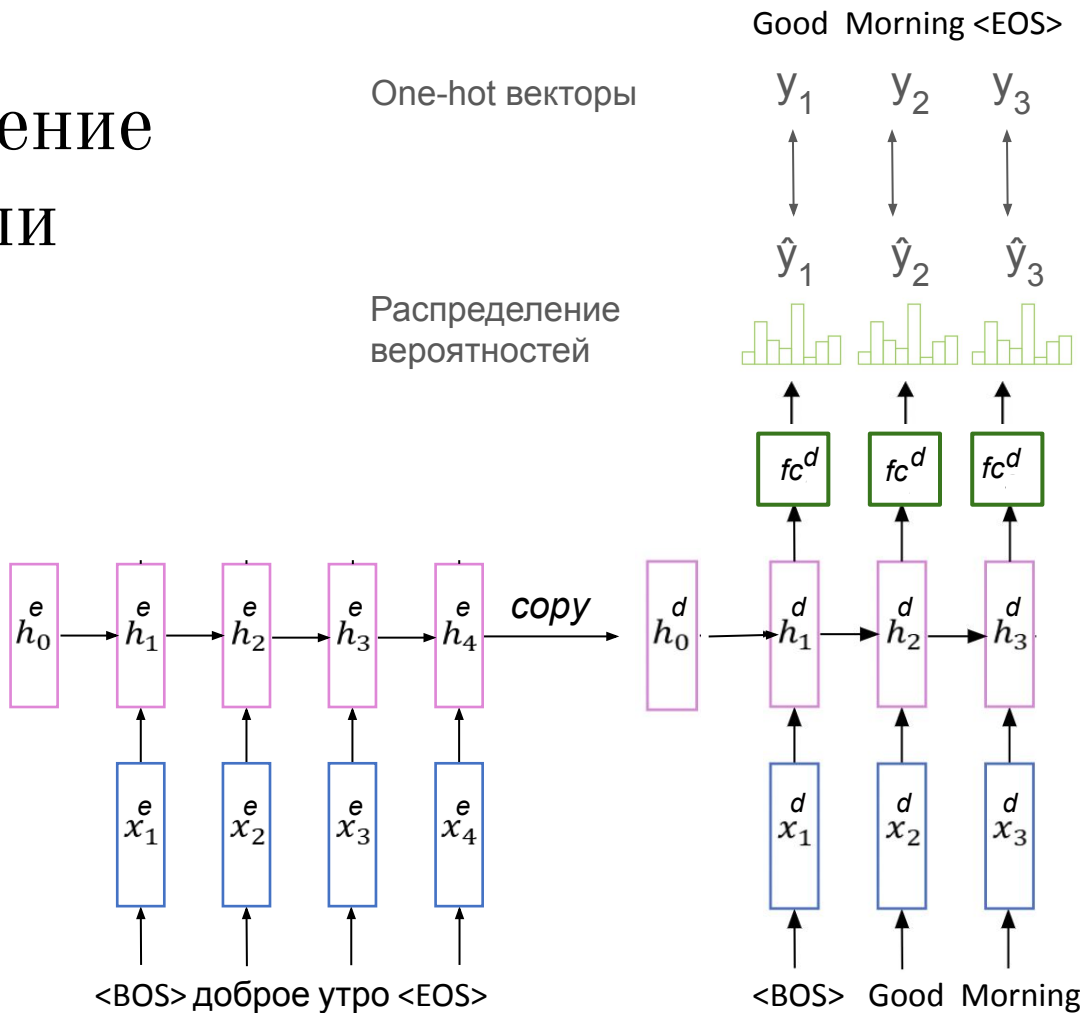
это корпус параллельных текстов, т.е. набор пар вида:

(source\_sentence, target\_sentence)

(доброе утро, good morning)

# Обучение модели

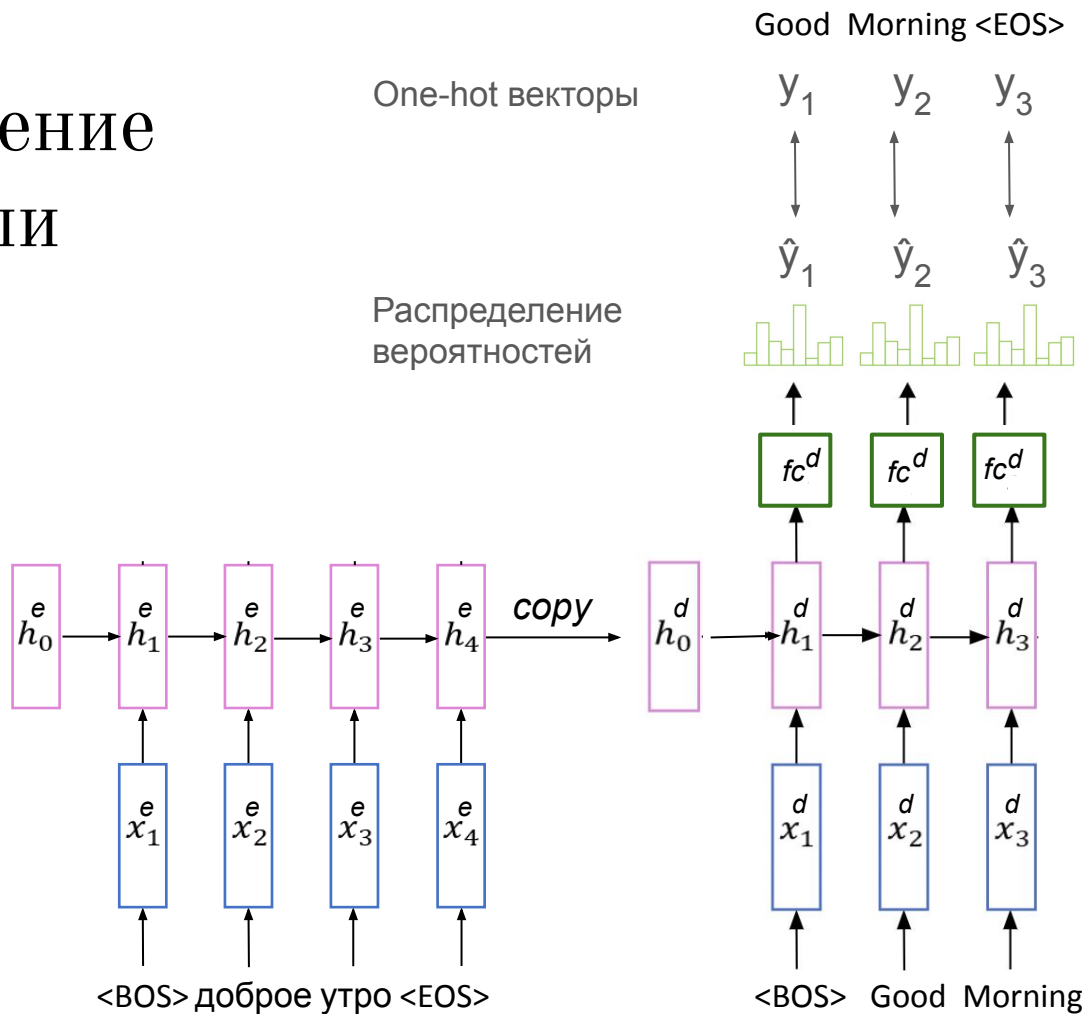
Source  
sentence



$$loss = \sum_{i=1}^n CE(\hat{y}_i, y_i)$$

# Обучение модели

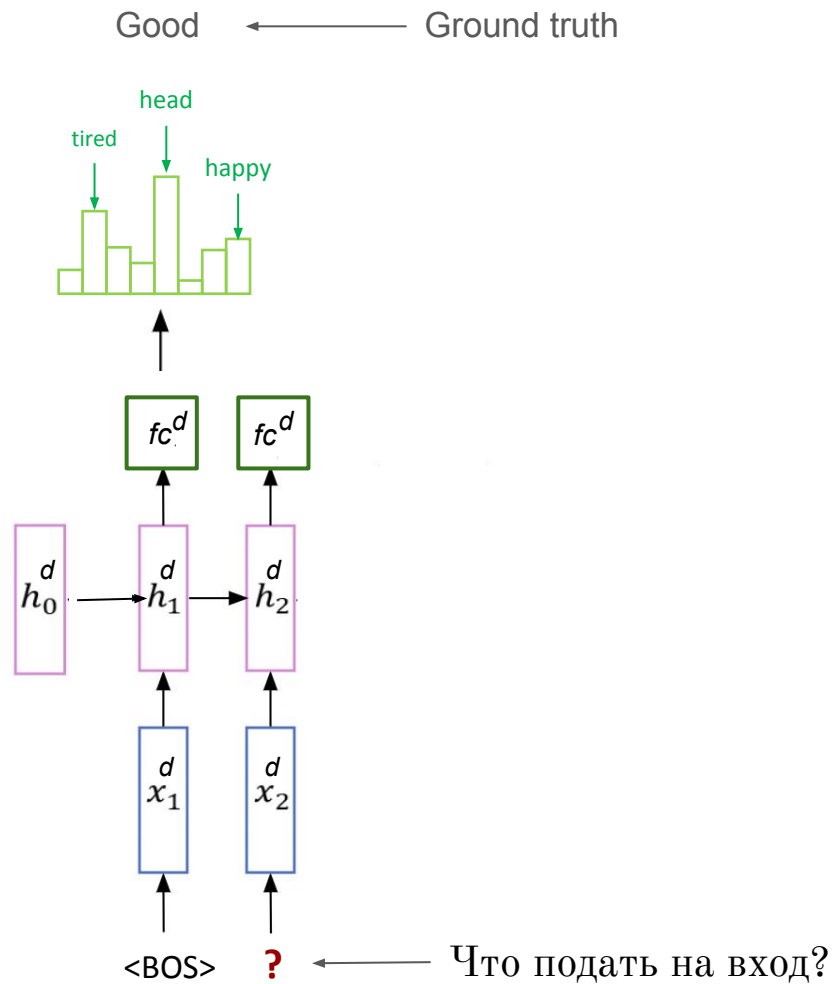
Source  
sentence



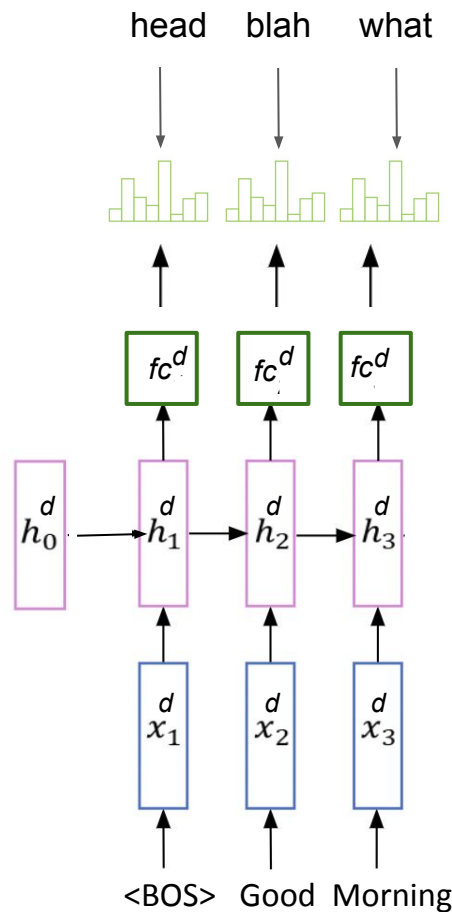
$$loss = \sum_{i=1}^n CE(\hat{y}_i, y_i)$$

Вся сеть целиком  
обучается с помощью  
обратного  
распространения ошибки

# Обучение модели



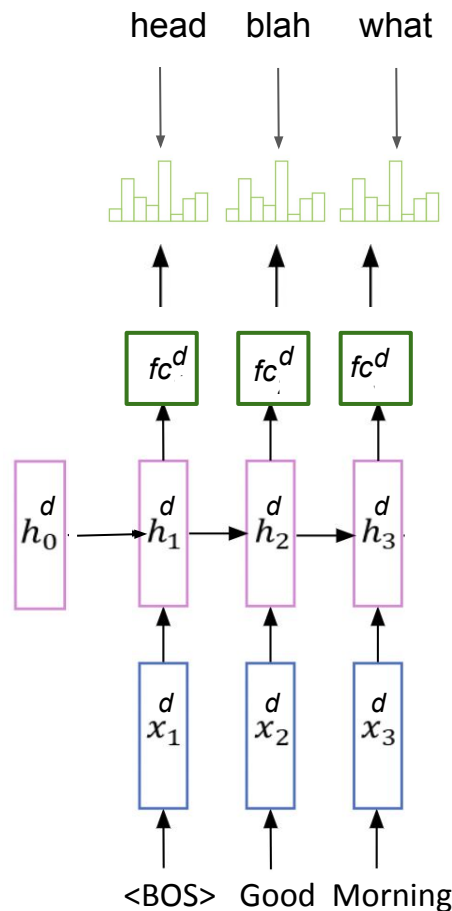
# Обучение модели



## Teacher forcing:

Подаем на вход декодеру токены, которые модель должна сгенерировать в идеале

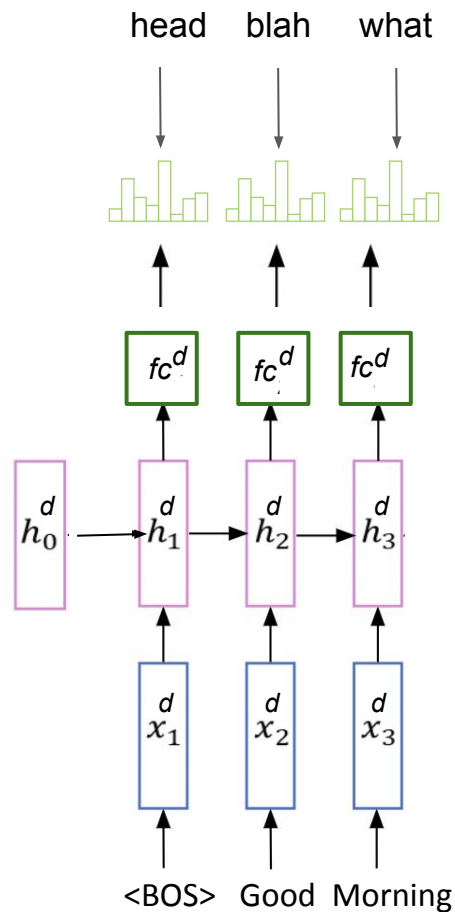
# Обучение модели



## Teacher forcing:

Можно использовать teacher forcing с вероятностью  $\rho$  на каждой итерации обучения, начиная с  $\rho=1$  в начале обучения и постепенно уменьшая  $\rho$

# Обучение модели



## Teacher forcing:

Во время инференса  
использовать teacher  
forcing не получится



# Обучение модели

Нюансы обучения:

- У Encoder и Decoder разные словари  
(т.к языки разные);
- Для обоих словарей можно  
использовать subword tokenization;
- При генерации можно использовать  
beam search

# Метрики качества для МТ

Как оценивать качество перевода?

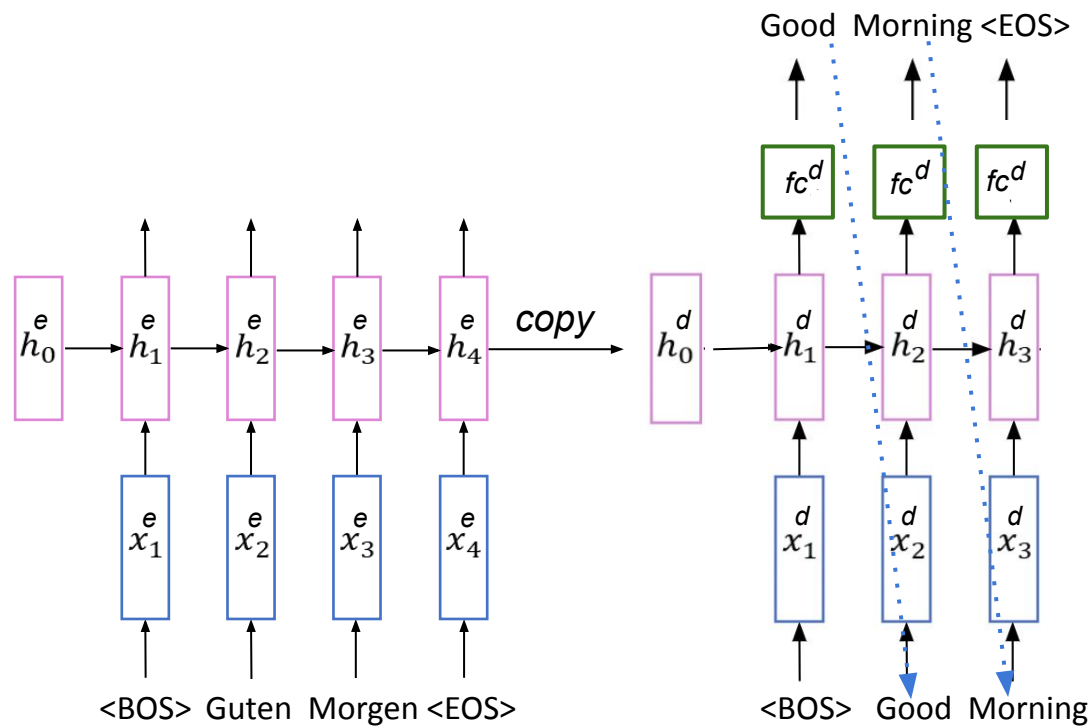
При обучении мы используем кросс-энтропию, но она слабо коррелирует с реальным качеством перевода.

Самая распространенная метрика оценки качества перевода — BLEU

# Обучение модели

У построенной нами архитектуры есть  
недостатки

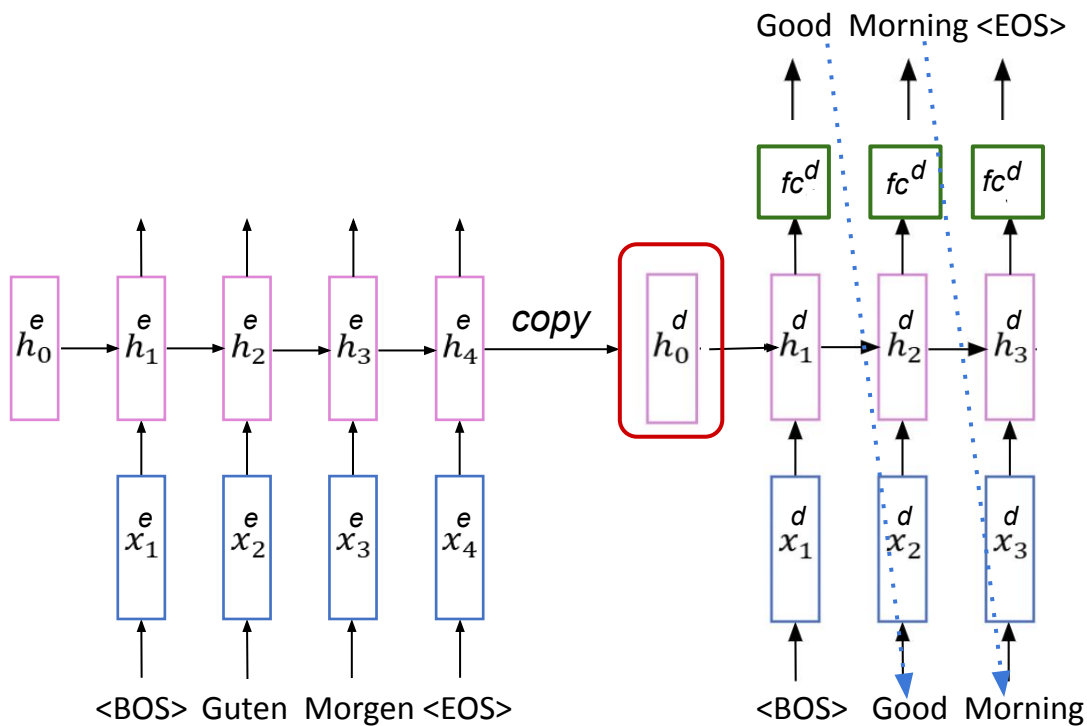
# RNN for MT



Недостатки:

- Encoder RNN может забывать информацию из начала предложения;

# RNN for MT



Недостатки:

- Encoder RNN может забывать информацию из начала предложения;
- Вся информация о входном предложении содержится в одном векторе  $h_d^0$

# RNN for MT

Как можно бороться с этими проблемами:

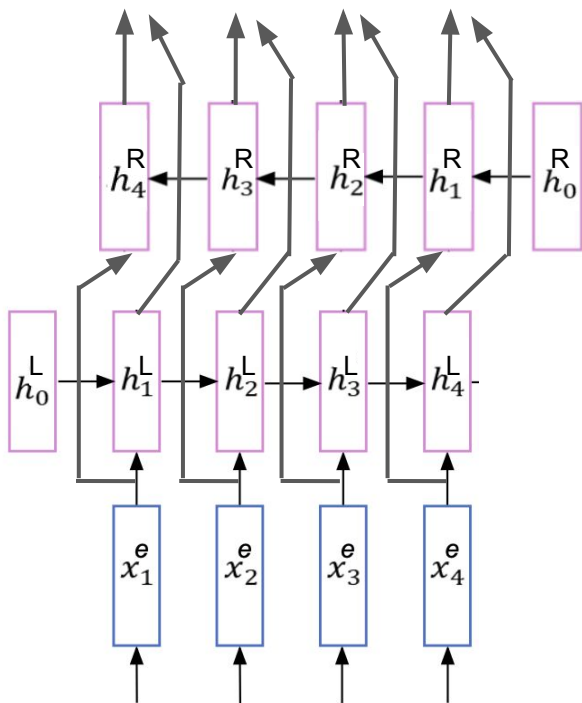
- Использовать GRU/LSTM;
- Использовать bidirectional RNN

Но ничего из этого не поможет решить  
проблему достаточно хорошо

# Bidirectional RNN

Один bidirectional слой имеет два вектора скрытого состояния.

Один “читает” последовательность слева направо, второй — справа налево



# Итоги видео

В этом видео мы разобрали:

- Задачу машинного перевода;
- RNN-архитектуру Seq2Seq для решения задачи перевода;
- Принцип и нюансы обучения модели Seq2Seq для перевода;
- Недостатки модели Seq2Seq, основанной на RNN.