

# Глубинное обучение

Лекция 13  
Машинный перевод. Seq2seq.

Михаил Гущин

[mhushchyn@hse.ru](mailto:mhushchyn@hse.ru)

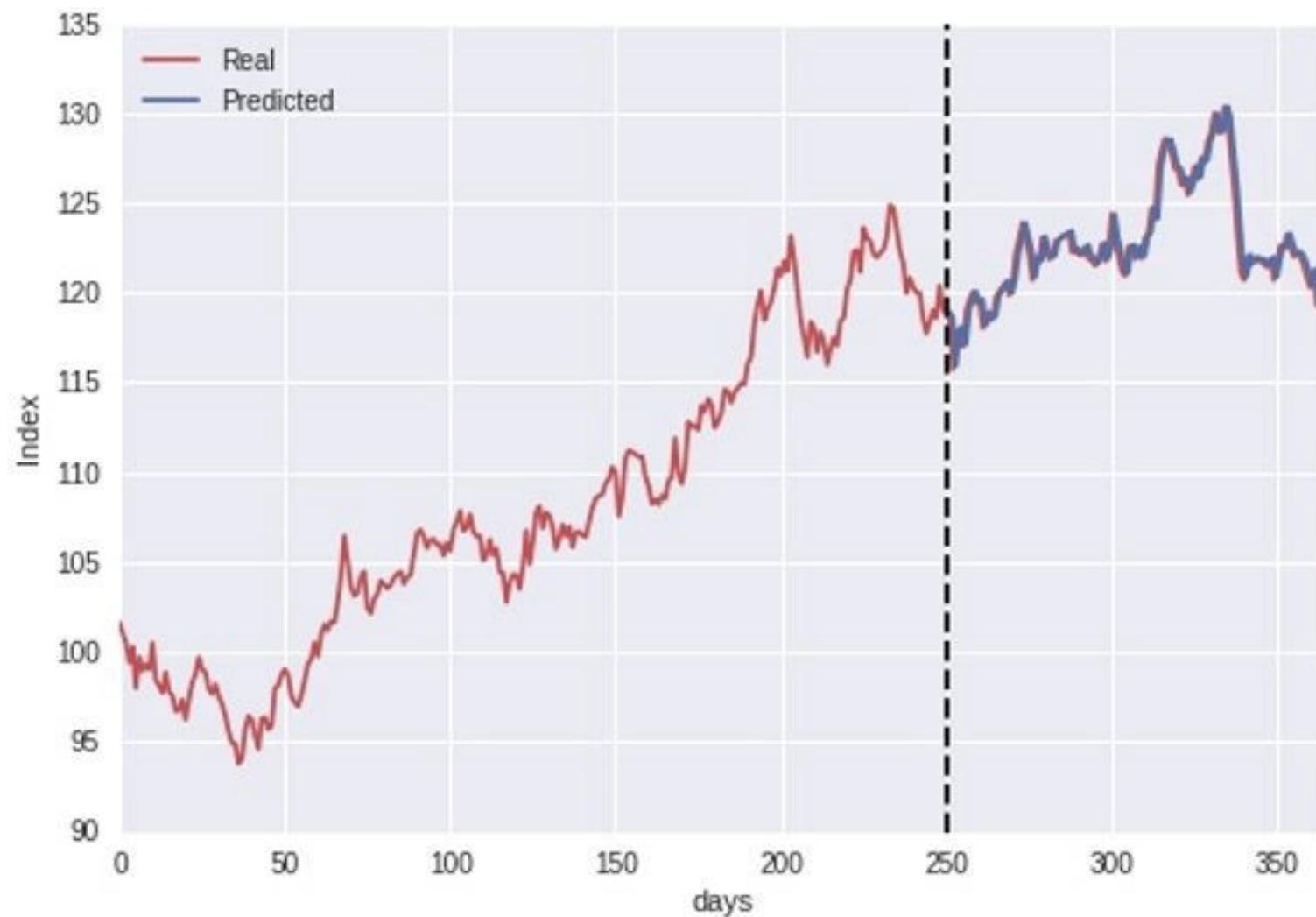
НИУ ВШЭ, 2024



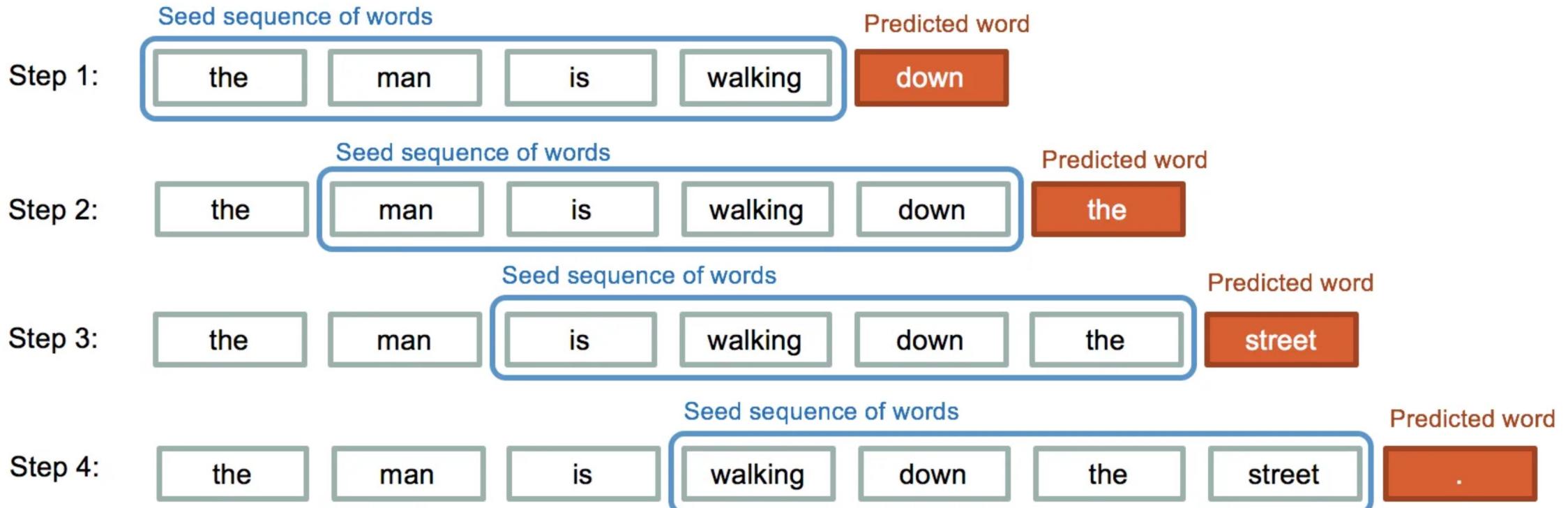
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Анализ последовательностей

# Прогноз временных рядов



# Генерация текста



Link: <https://bansalh944.medium.com/text-generation-using-lstm-b6ced8629b03>

# Задача

- ▶ Пусть дано  $n$  последовательностей из  $T$  наблюдений:

$$x_1^1, x_2^1, \dots, x_t^1, \dots, x_{T-1}^1, x_T^1$$

$$x_1^2, x_2^2, \dots, x_t^2, \dots, x_{T-1}^2, x_T^2$$

...

$$x_1^n, x_2^n, \dots, x_t^n, \dots, x_{T-1}^n, x_T^n$$

где  $x_t^i \in R^d$

- ▶ Хотим делать прогноз на один шаг вперед:

$$\hat{x}_{T+1}^i = f(x_1^i, x_2^i, \dots, x_t^i, \dots, x_{T-1}^i, x_T^i)$$

# Задача

- ▶ Матрица наблюдений  $X_t \in R^{n \times d}$  из  $n$  объектов в момент времени  $t$ , каждый из которых имеет  $d$  входных признаков:

$$X_t = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

- ▶ Число строк – число наблюдений (объектов) в момент времени  $t$
- ▶ Число столбцов – число входных признаков

# Рекуррентный нейронный слой

$$H_t = f(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$$

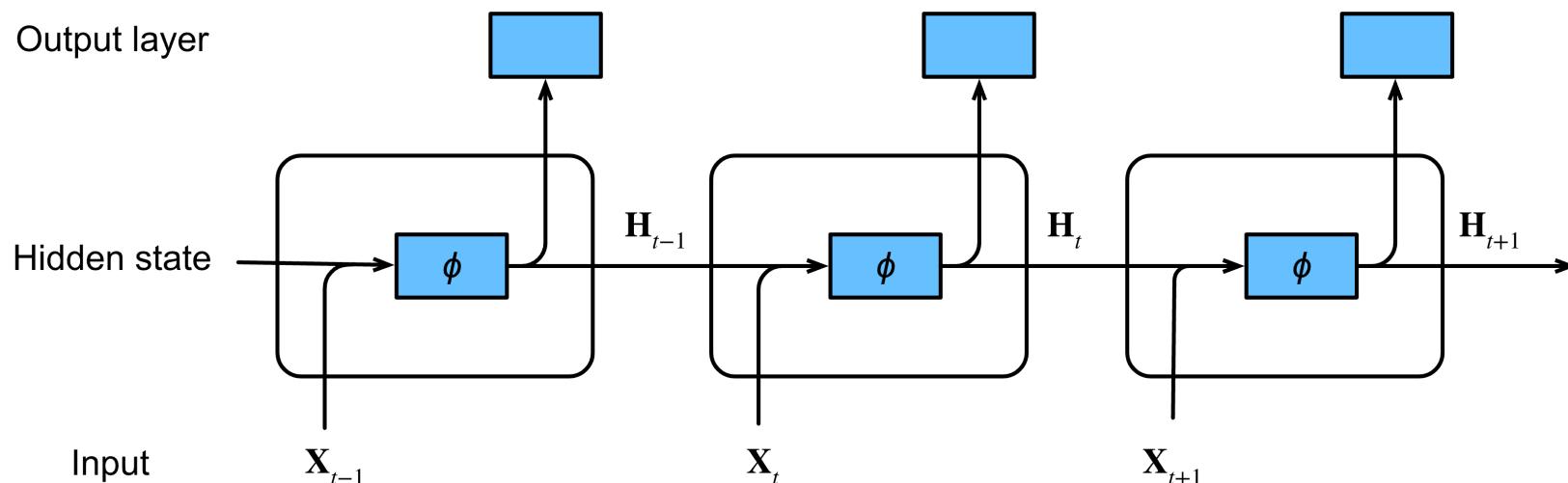
$$O_t = f_o(H_t W_{ho} + b_o)$$

- ▶  $X_t \in R^{n \times d}$  - матрица из  $n$  наблюдений в момент времени  $t$
- ▶  $H_t \in R^{n \times h}$  - матрица из  $n$  скрытых состояний слоя
- ▶  $W_{xh} \in R^{d \times h}, W_{hh} \in R^{h \times h}, W_{ho} \in R^{h \times o}, h$  - число нейронов в слое
- ▶  $b_h \in R^{1 \times h}, b_o \in R^{1 \times o}$

# Рекуррентный нейронный слой

$$H_t = f(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$$

$$O_t = f_o(H_t W_{ho} + b_o)$$



FC layer with  
activation function



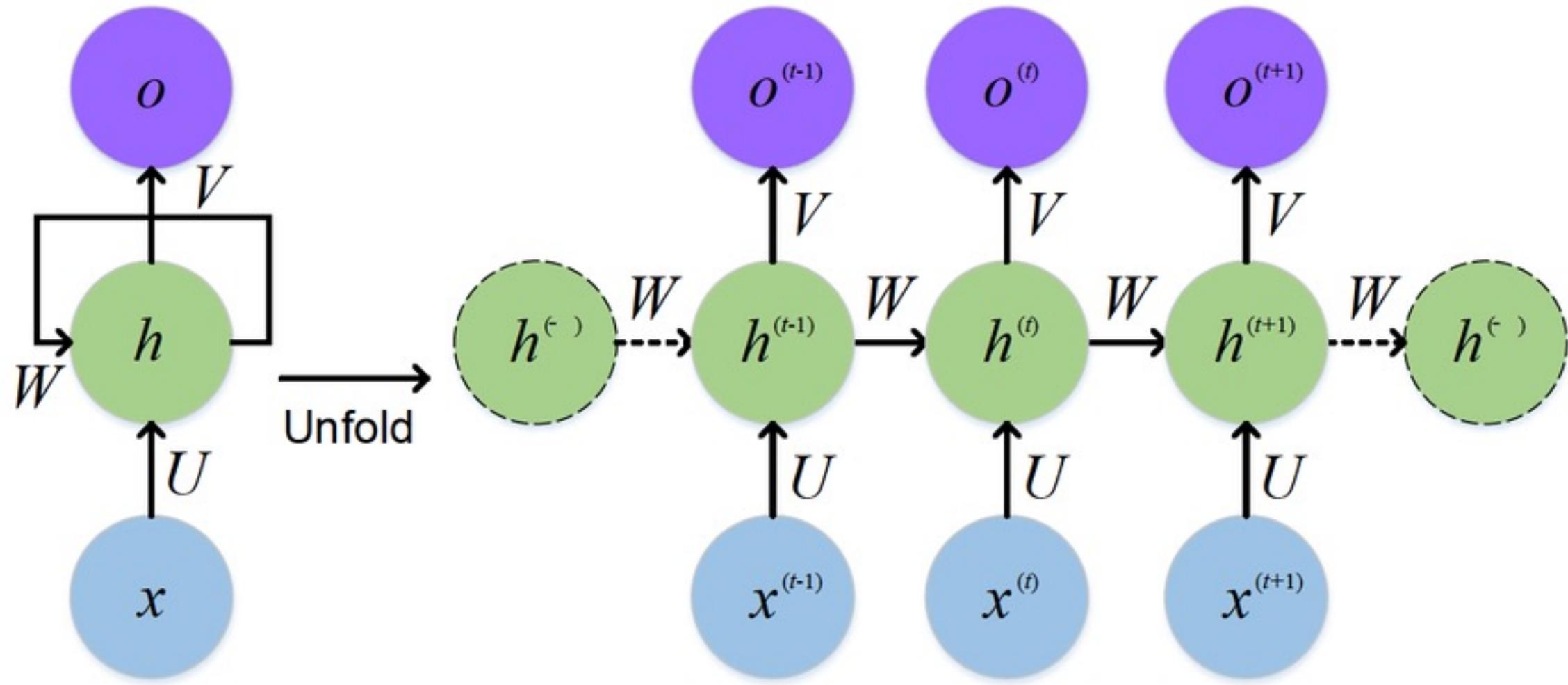
Copy



Concatenate

Link: [https://d2l.ai/chapter\\_recurrent-modern/lstm.html](https://d2l.ai/chapter_recurrent-modern/lstm.html)

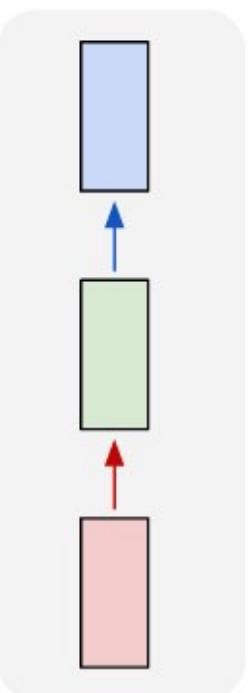
# Рекуррентный нейронный слой



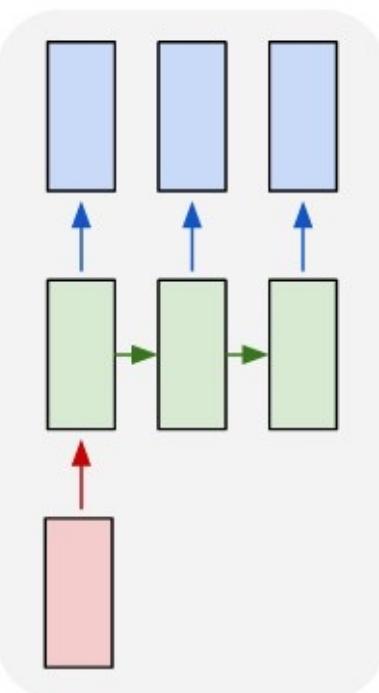
# Типы задач с последовательностями

# Типы задач

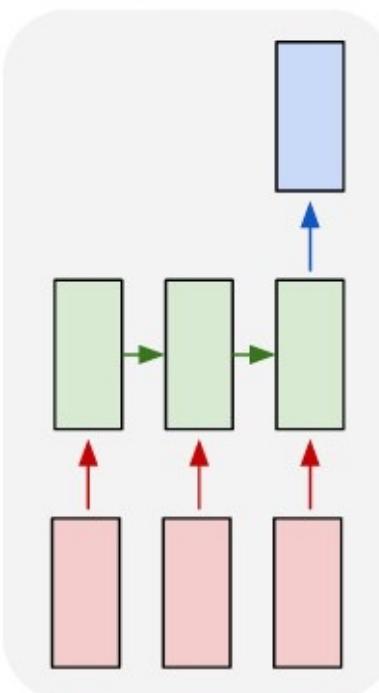
one to one



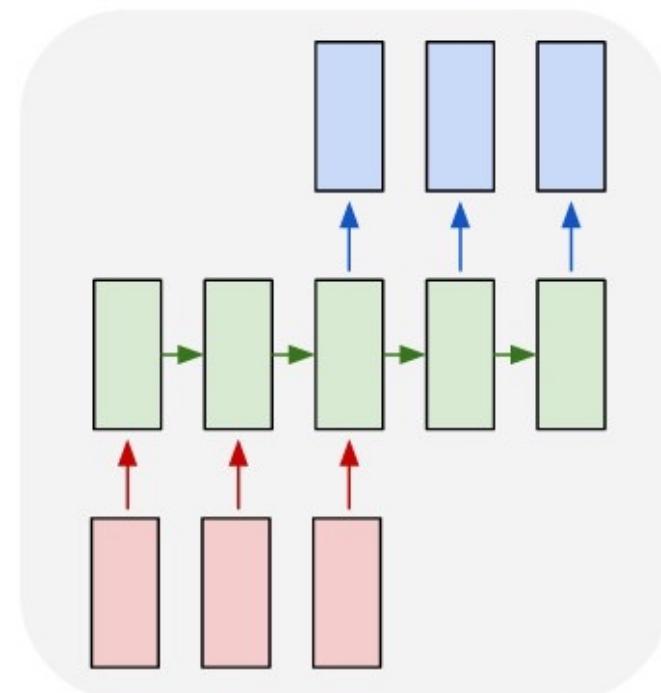
one to many



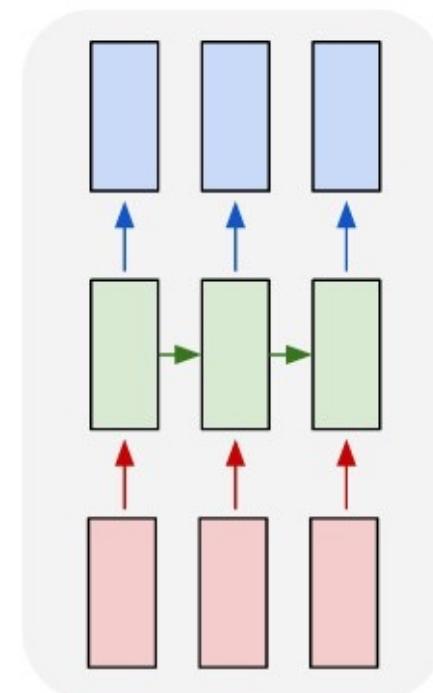
many to one



many to many



many to many

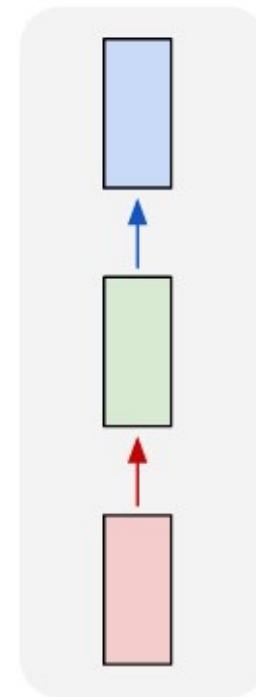


Источник: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

# One2one

- ▶ Фиксированный размер входа и выхода
- ▶ Нет временной зависимости
- ▶ Полносвязные нейронные сети
- ▶ Задачи классификации в табличных данных и изображений (image captioning)

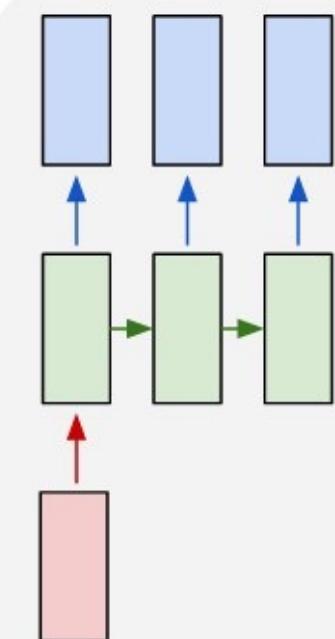
one to one



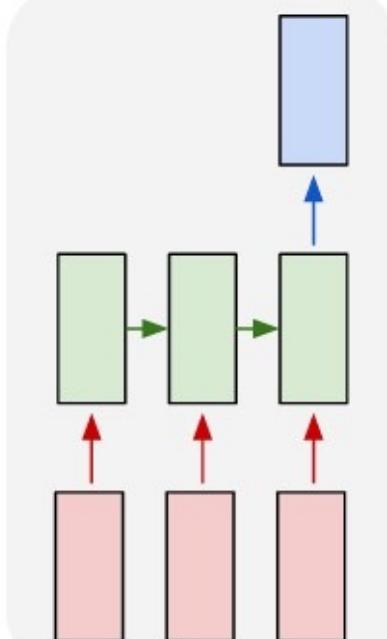
# One2seq и seq2one

- ▶ Учитываются зависимости в последовательности
- ▶ Используются RNN
- ▶ One2seq:
  - Вход – картинка, выход – текстовое описание картинки
- ▶ Seq2one:
  - Вход – текст, выход – класс текста (sentiment analysis)

one to many

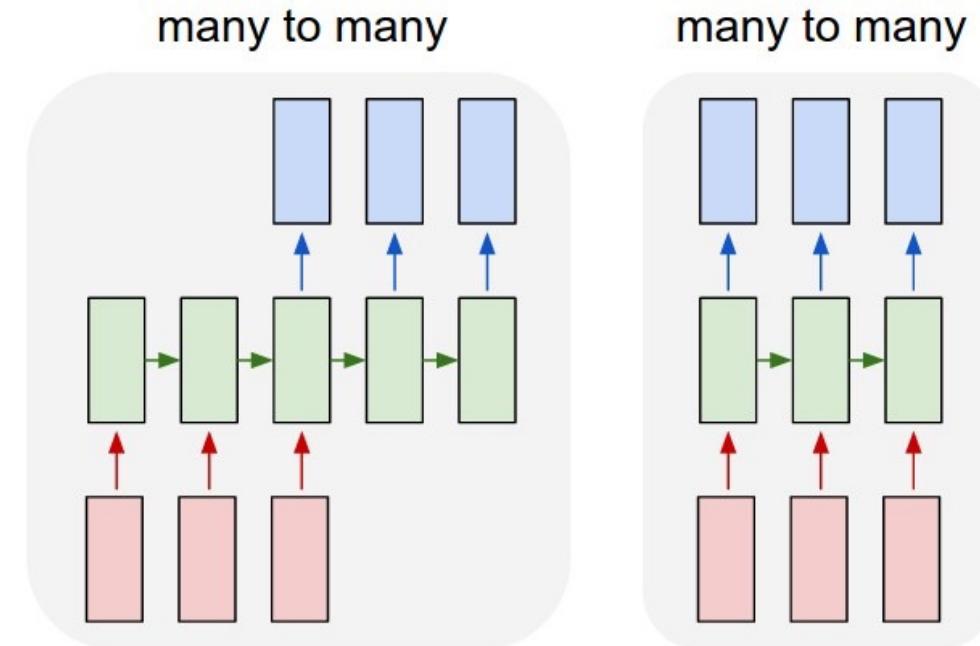


many to one



# Seq2seq

- ▶ Seq2seq (1):
  - Вход – текст на русском, выход – текст на английском
  - Вход – начало текста, выход – продолжение текста (генерация текста)
- ▶ Seq2seq (2):
  - Вход – видео, выход – класс каждого кадра видео



# Генерация текста

# Задача

- ▶ Пусть мы хотим генерировать текст буква-за-буквой
- ▶ Для простоты рассмотрим словарь только из четырех букв [h, e, l, o]
- ▶ Для решения будем использовать RNN:

$$\begin{aligned} H_t &= f(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \\ O_t &= f_o(H_t W_{ho} + b_o) \end{aligned}$$

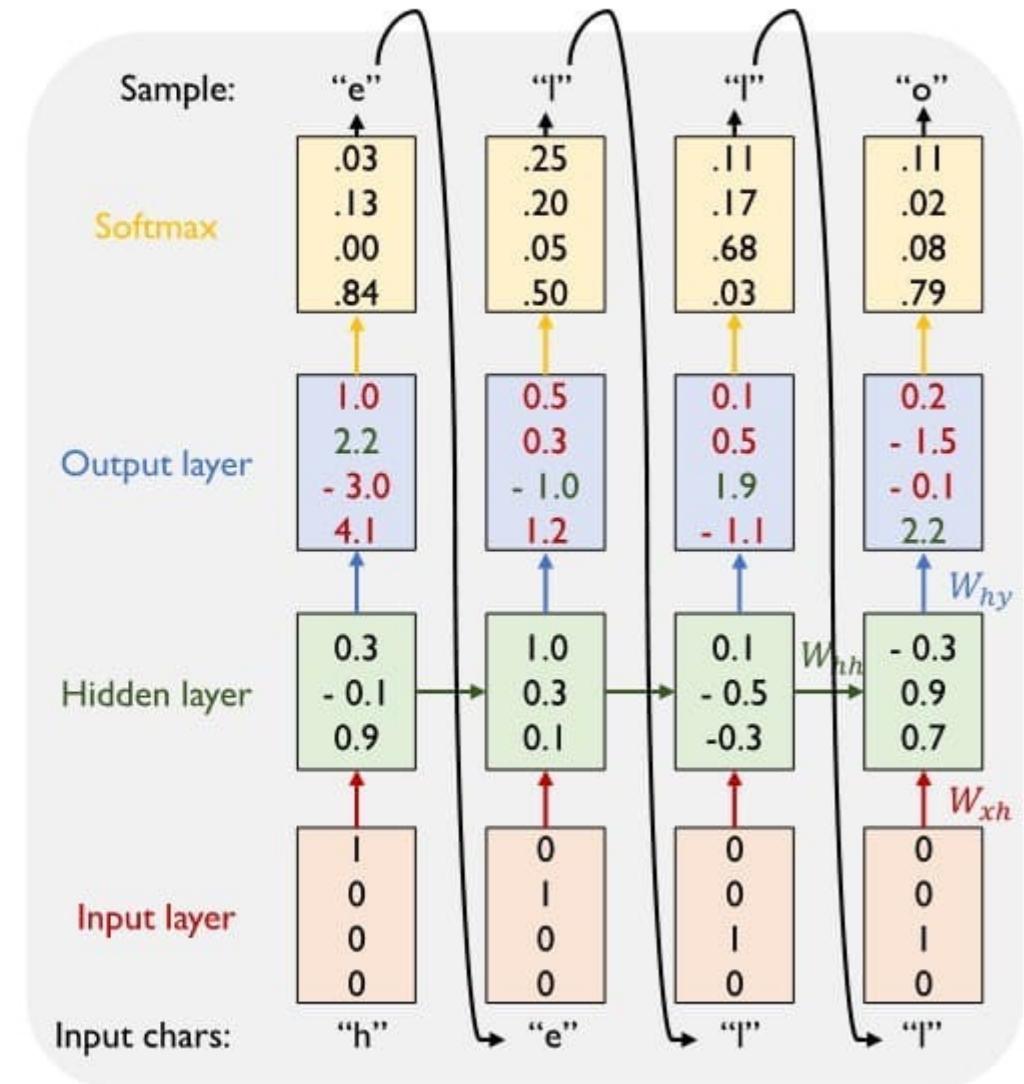
# Пример решения задачи

- ▶ Напоминалка по RNN:

$$H_t = f(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$$

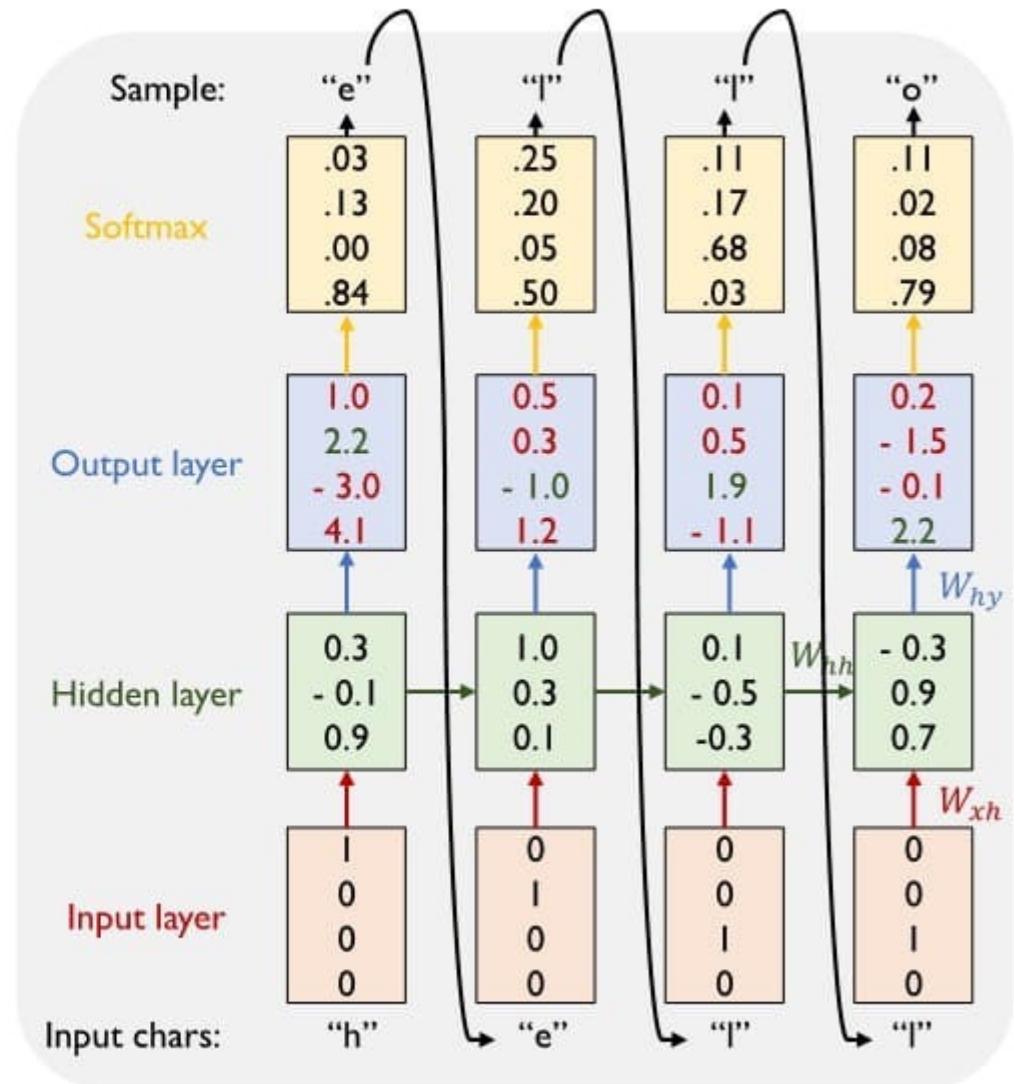
$$O_t = f_o(H_t W_{ho} + b_o)$$

- ▶ На выходе - вероятность следующей буквы в тексте (для всех букв в словаре)
- ▶ Как выбирать следующую букву из прогноза (несколько вариантов)?

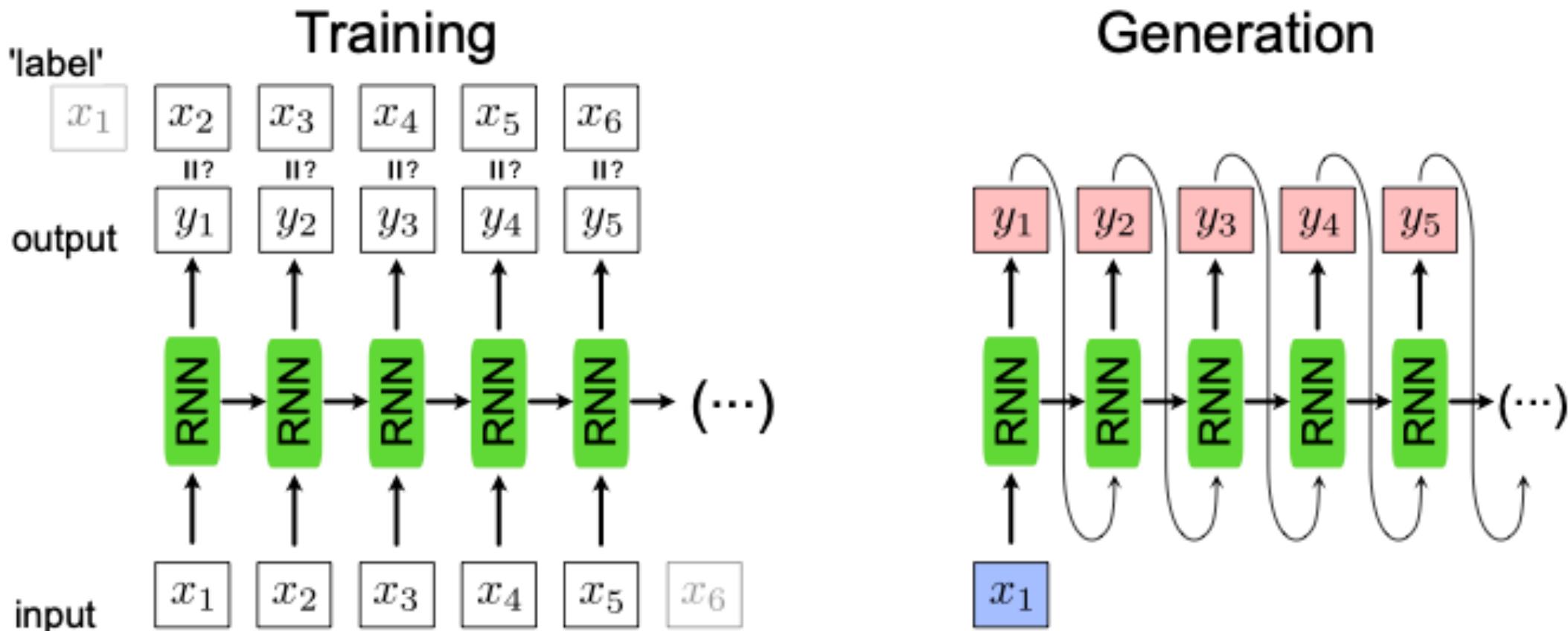


# Пример решения задачи

- ▶ Способы выбора следующей буквы из прогноза (хорошие и не очень):
  - Выбирать наиболее вероятную (не очень)
  - Выбирать случайную из топ-к наиболее вероятных (лучше)
  - Beam search (отличный)

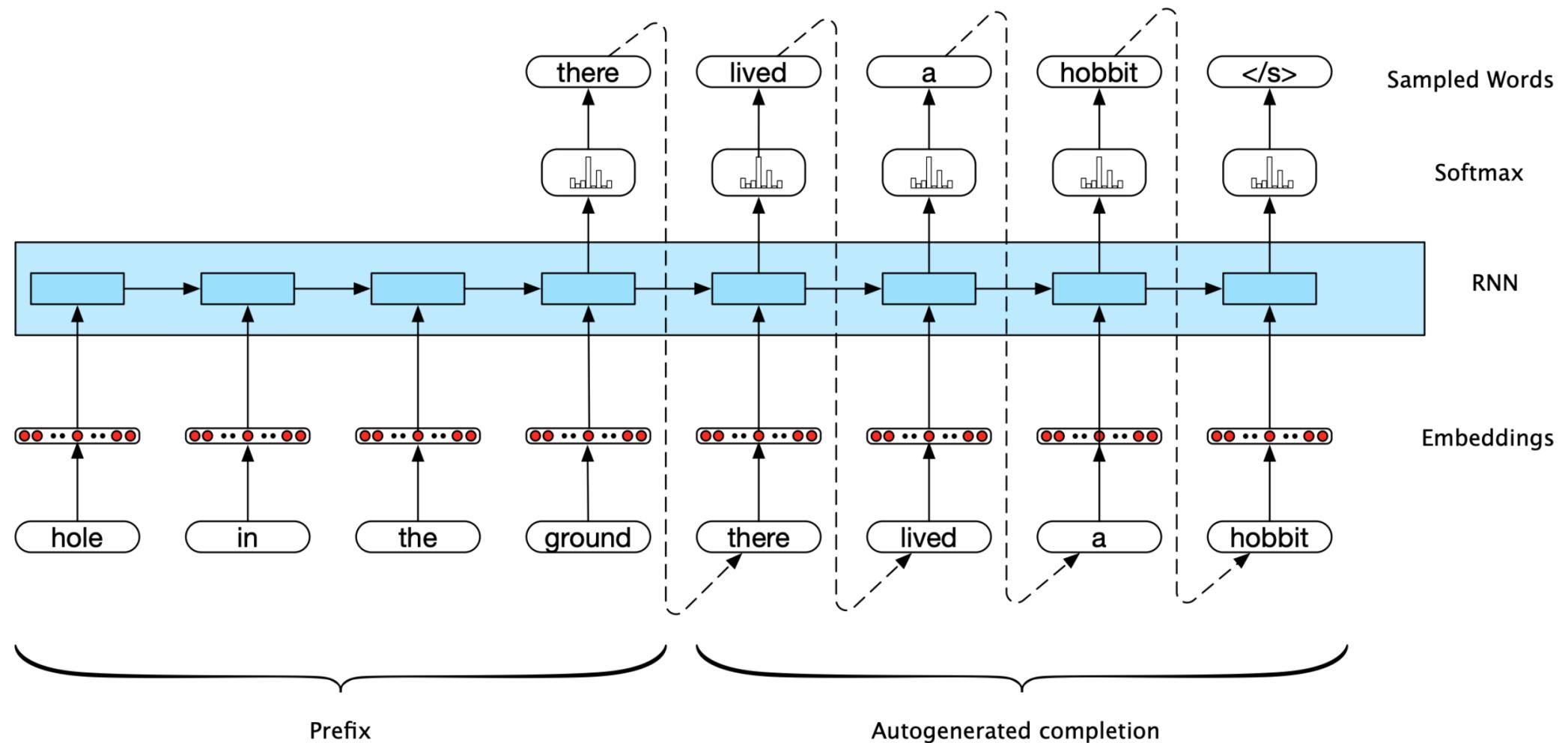


# Обучение и генерация



Источник: [https://ml-lectures.org/docs/unsupervised\\_learning/ml\\_unsupervised-2.html](https://ml-lectures.org/docs/unsupervised_learning/ml_unsupervised-2.html)

# Модификация



Источник: <https://courses.grainger illinois.edu/cs447/sp2023/Slides/Lecture13.pdf>

# Модификация

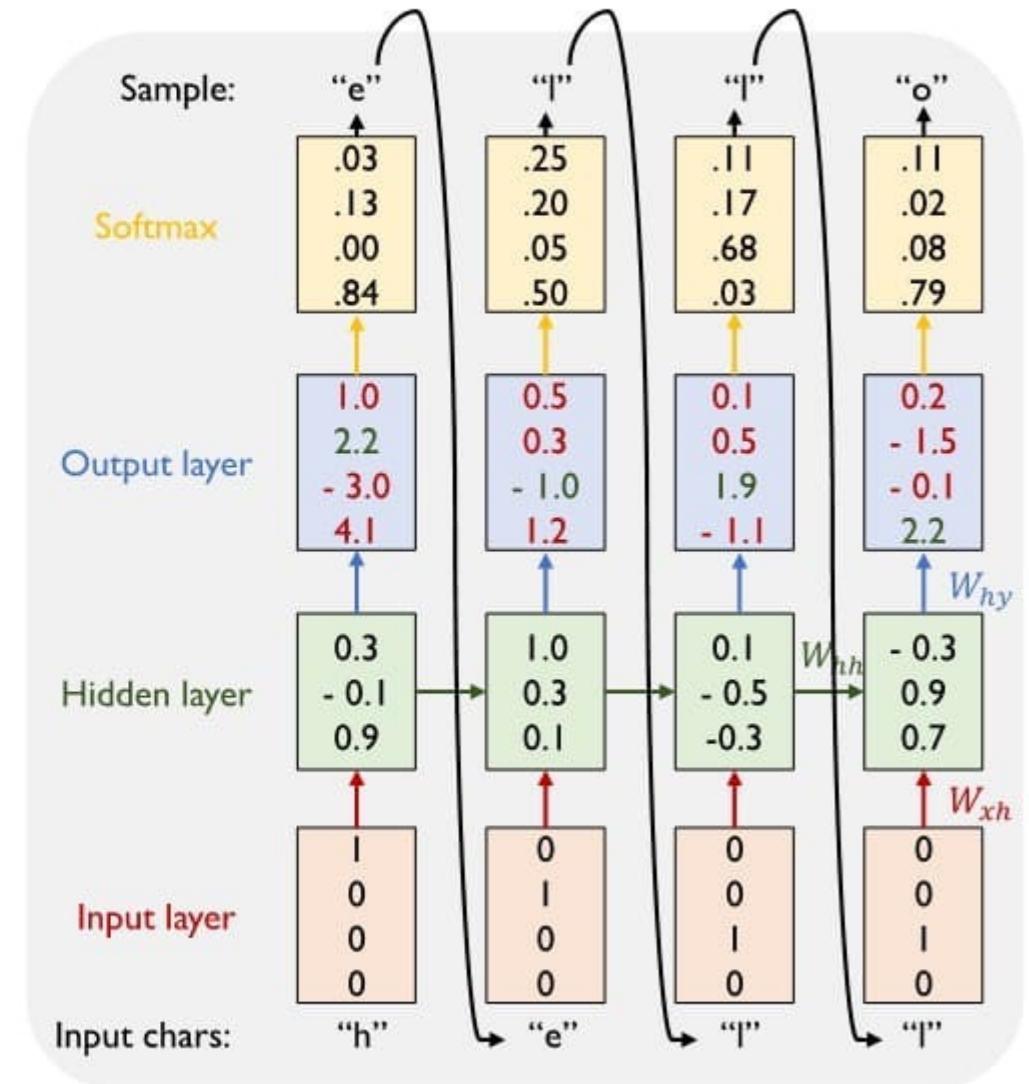
- ▶ Генерировать текста можно не только буква-за-буквой, но и слово-за-словом
- ▶ Начинать генерацию можно с одного токена (буква, слово), так и с нескольких токенов (промт)
- ▶ Промт задает контекст на генерируемый текст
- ▶ Примеры (промт - прогноз):
  - Вопрос – ответ
  - Тема текста – текст



Beam search

# Задача

- ▶ На выходе - вероятность следующей буквы в тексте (для всех букв в словаре)
- ▶ Как выбирать следующую букву из прогноза?



# Жадный алгоритм

- ▶ Greedy decoding: на каждом шаге берем токен (букву или слово) с самой большой вероятностью
- ▶ Итоговой прогноз – последовательность токенов. Жадный способ на каждом шаге выбирает локальный оптимум

# Неоптимальность жадного алгоритма

Прогноз с использованием жадного алгоритма

Time step	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<eos>	0.1	0.2	0.2	0.6

Вероятность последовательности:  
 $0.5 \times 0.4 \times 0.4 \times 0.6 = 0.048$

Прогноз со случайным выбором 2-го самого популярного токена на 2-м шаге

Time step	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<eos>	0.1	0.2	0.1	0.6

Вероятность последовательности:  
 $0.5 \times 0.3 \times 0.6 \times 0.6 = \mathbf{0.054}$

Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

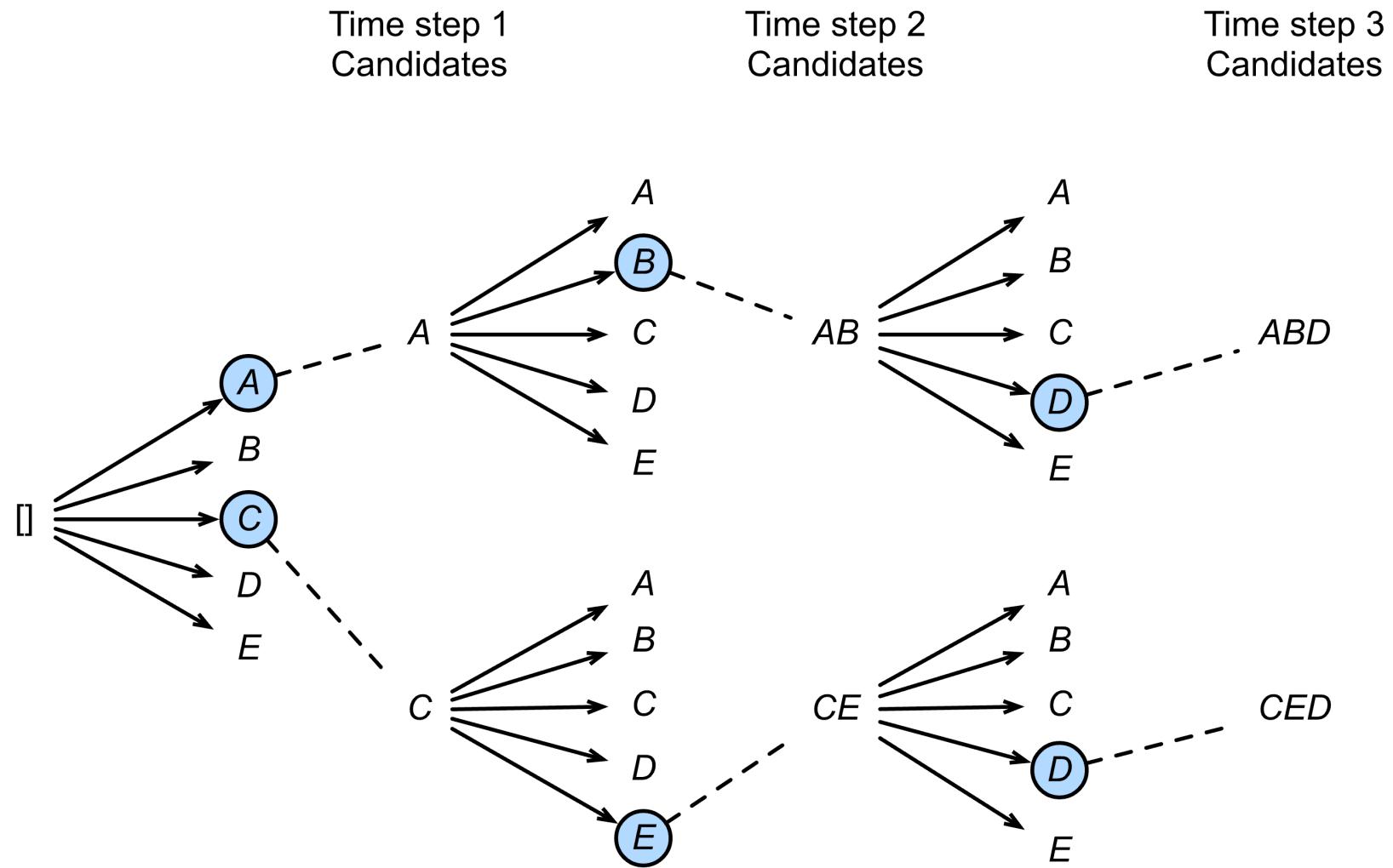
# Beam search

- ▶ Давайте поддерживать несколько самых вероятных траекторий (**beam size**)
- ▶ Такая стратегия называется Beam Search
- ▶ Число траекторий, которое мы помним будет гиперпараметром (больше 10 брать не стоит)

# Beam search

Beam size = 2

Максимальная  
длина  
последовательности = 3

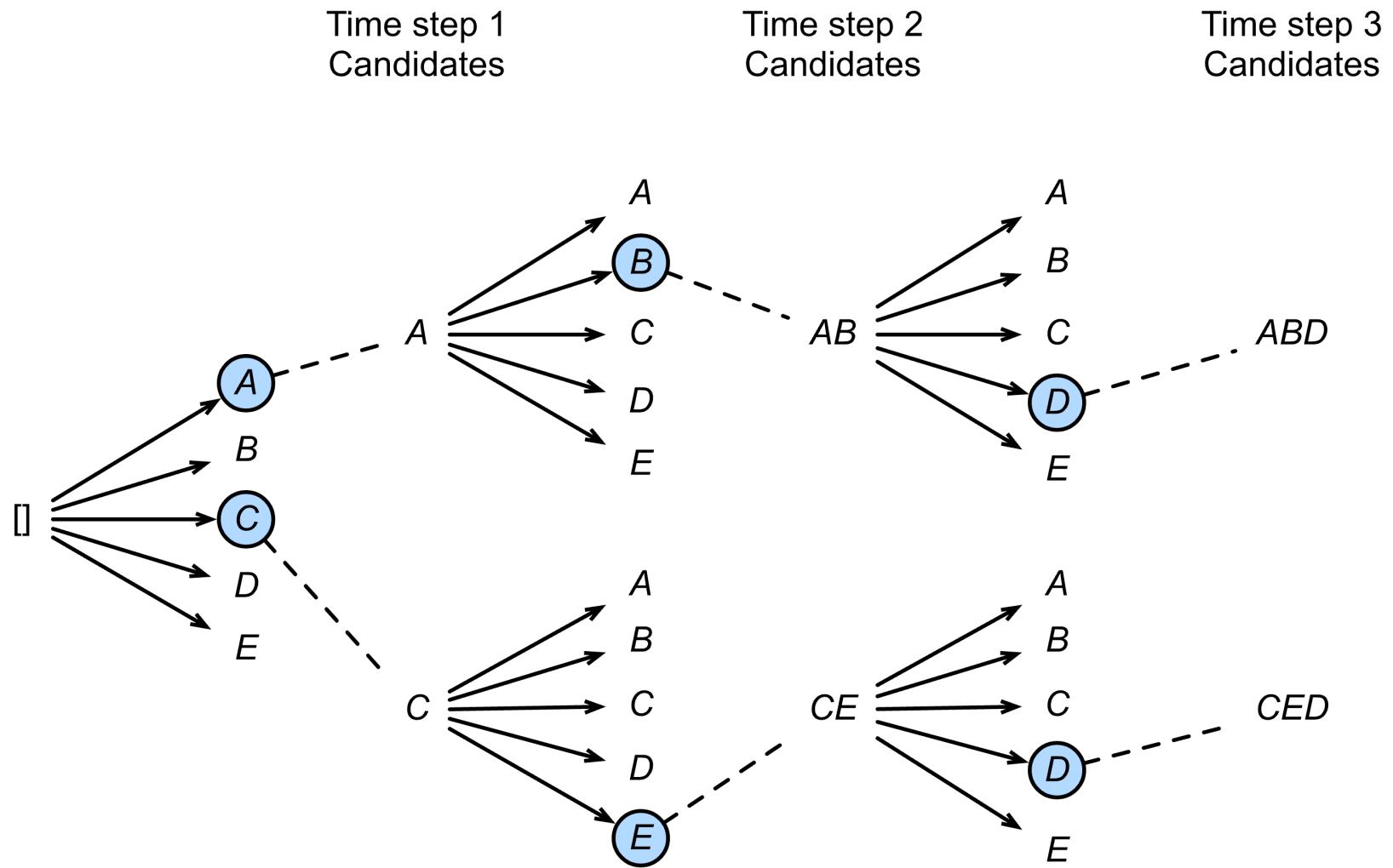


Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

# Beam search

Шаг 1:

Самые вероятные: A, C



Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

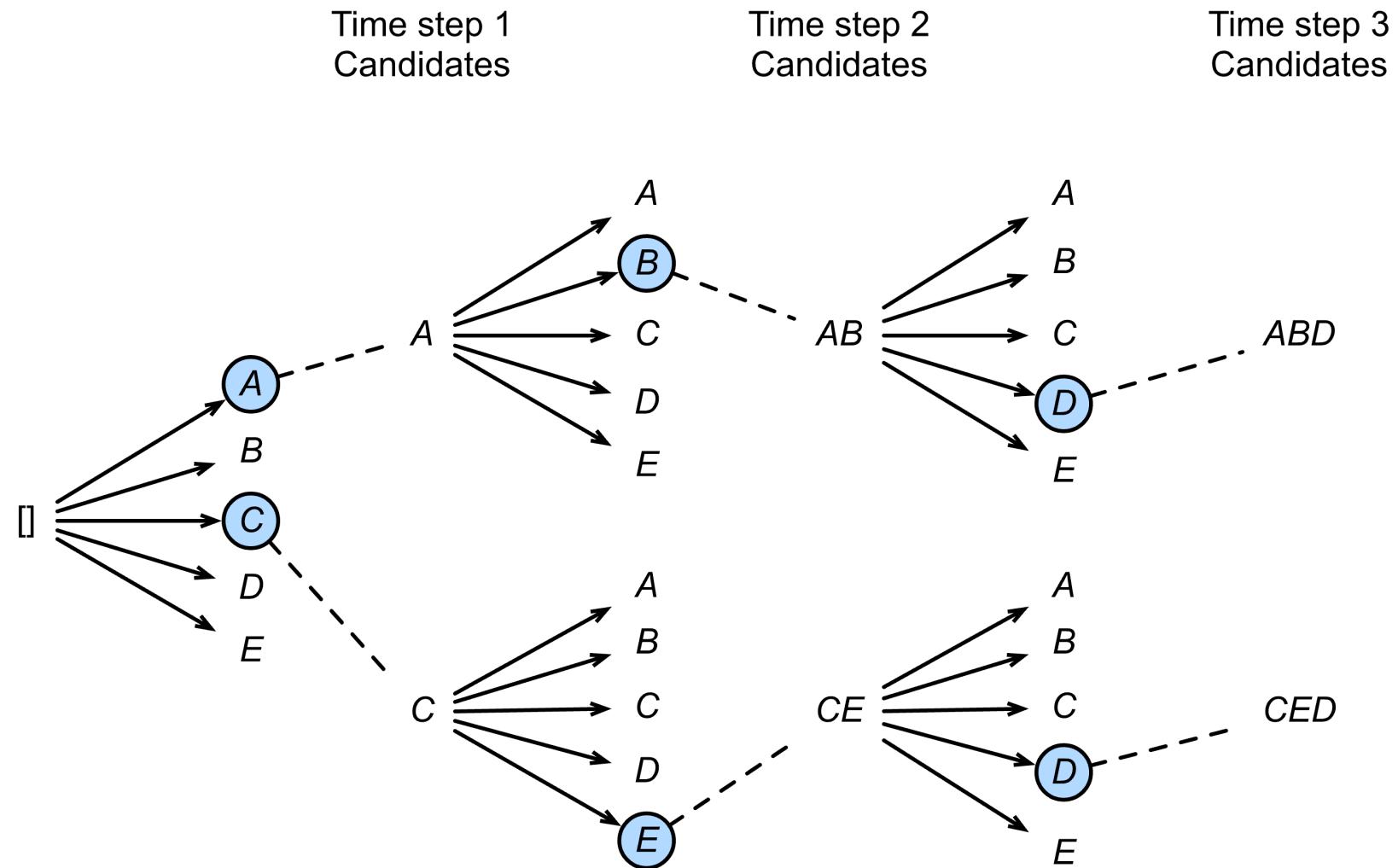
# Beam search

Шаг 2:

Перебираем пары токенов.

$$P(AB) = P(A \mid \text{шаг 1}) P(B \mid \text{шаг 2})$$

Топ-2 самых вероятных: AB, CE



Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

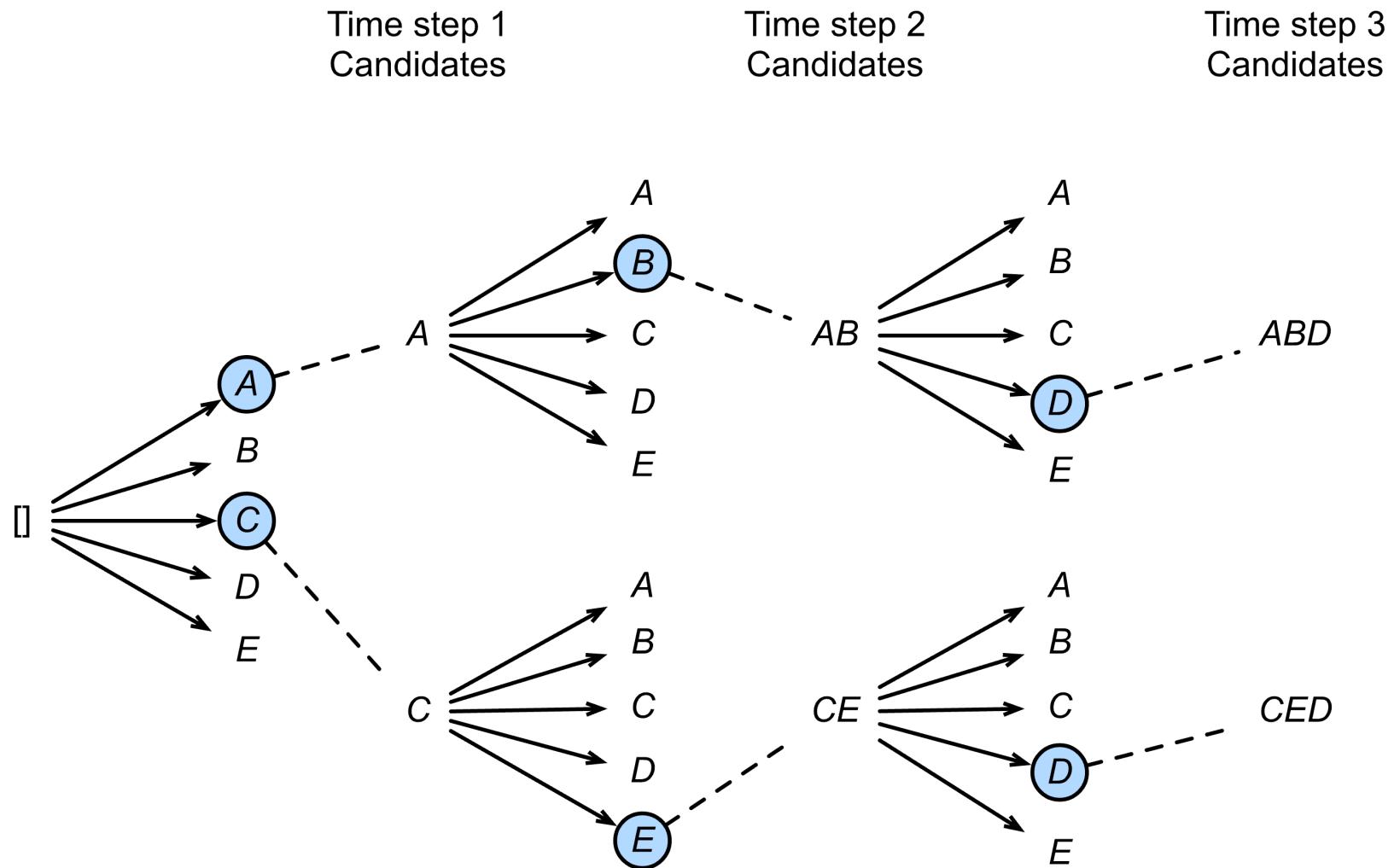
# Beam search

Шаг 3:

Перебираем пары токенов.

$$\begin{aligned} P(ABD) &= P(A \mid \text{шаг 1}) P(B \mid \text{шаг 2}) \\ P(D \mid \text{шаг 3}) \end{aligned}$$

Топ-2 самых вероятных: ABD,  
CED



Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

# Beam search

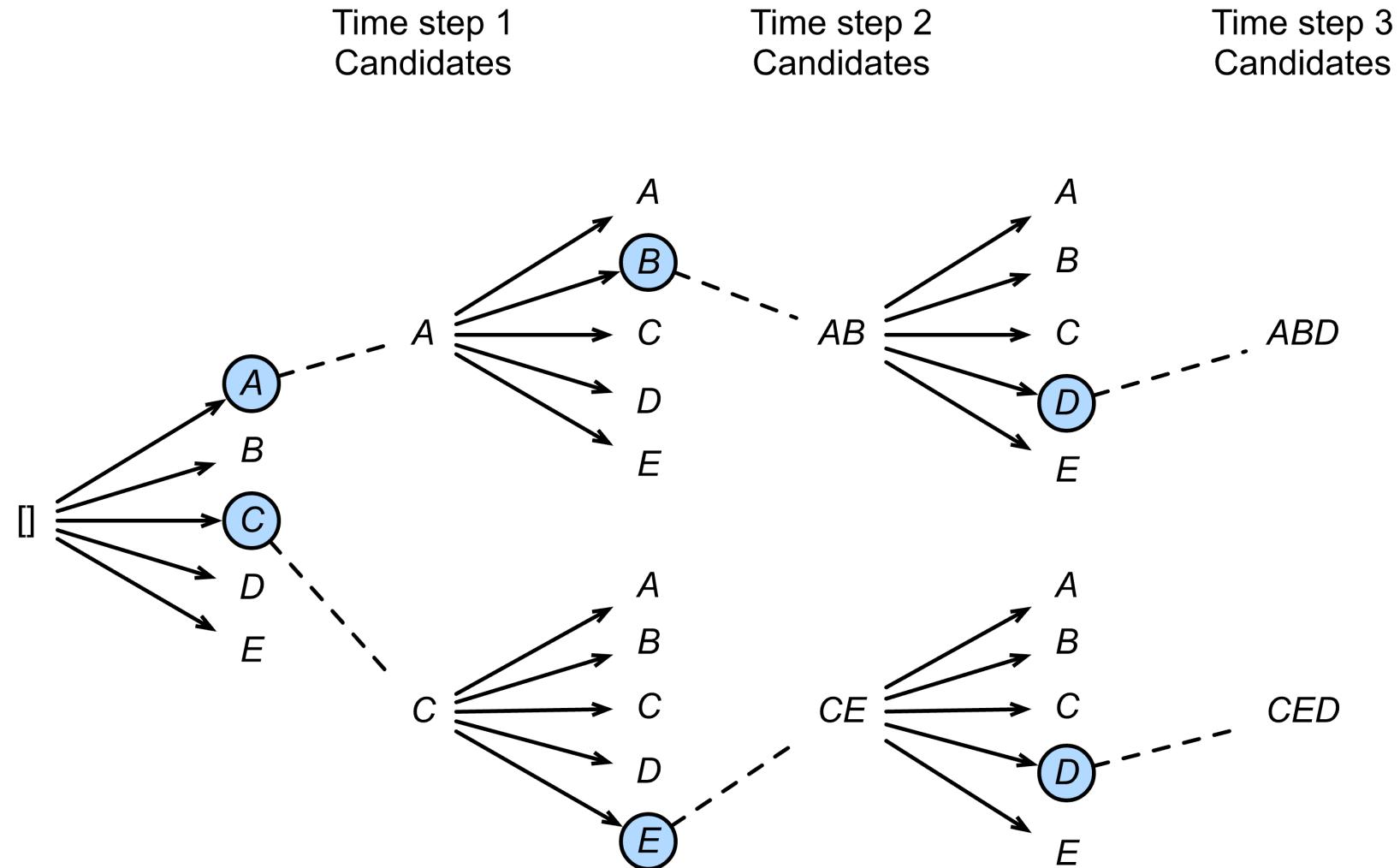
Итоговый прогноз:

Вариант 1:  
выбираем наиболее вероятную  
траекторию. Например, ABD

Вариант 2:  
Можем также учитывать  
промежуточные траектории. Тогда  
выбираем ту, где больше:

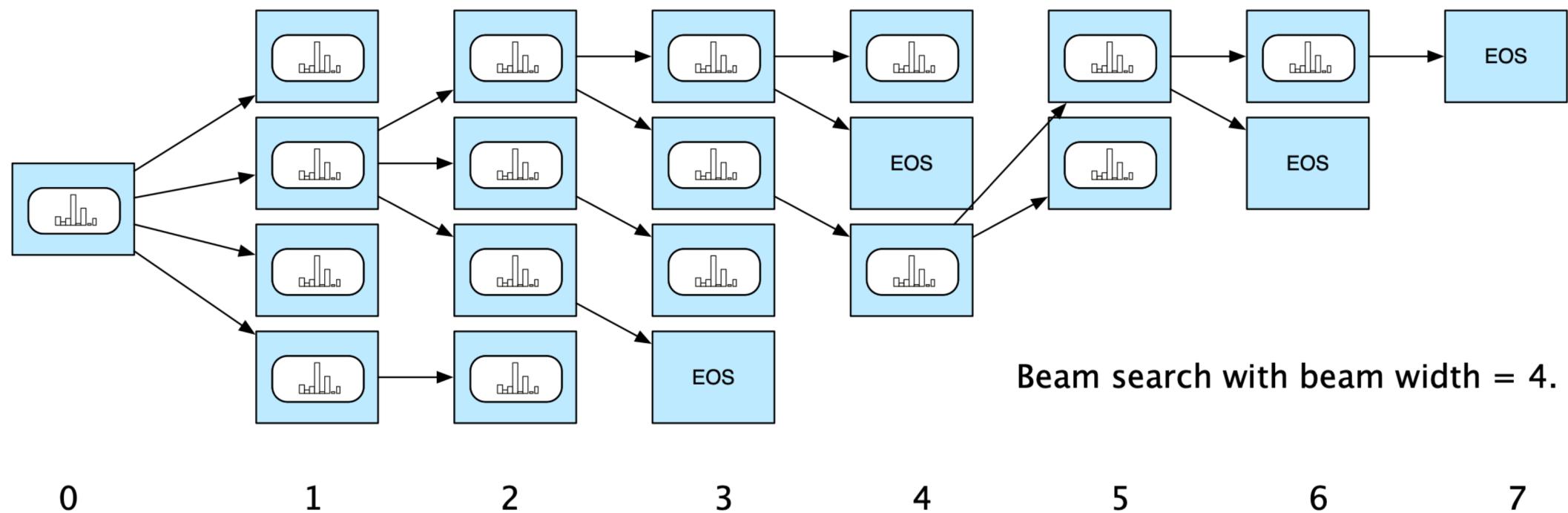
$$\frac{1}{L} \log P(\text{траектория})$$

L – длина траектории

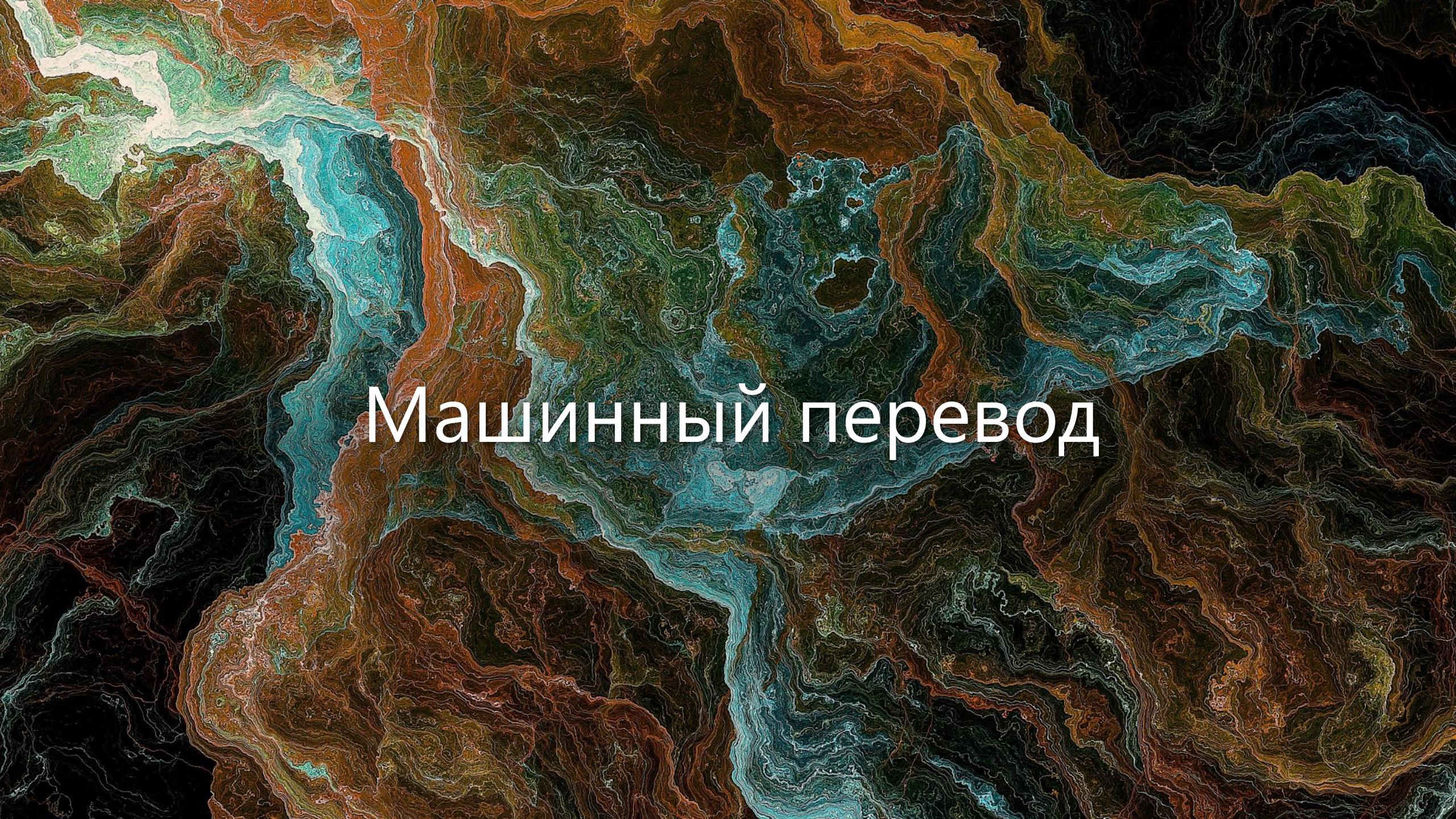


Источник: [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

# Специальные символы

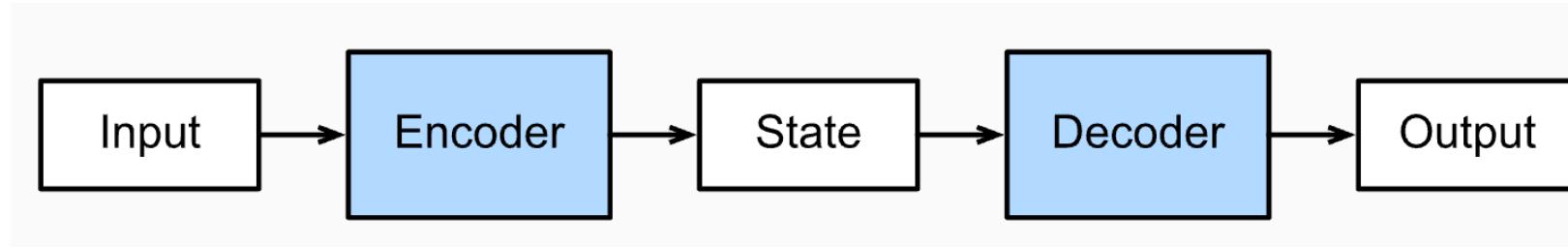


EOS – специальный символ конца последовательности

The background of the image is a complex, abstract pattern resembling a topographic map or a microscopic view of organic tissue. It features a dense network of fine lines in various colors, primarily shades of orange, green, blue, and white, which create a sense of depth and movement. The overall effect is organic and fluid.

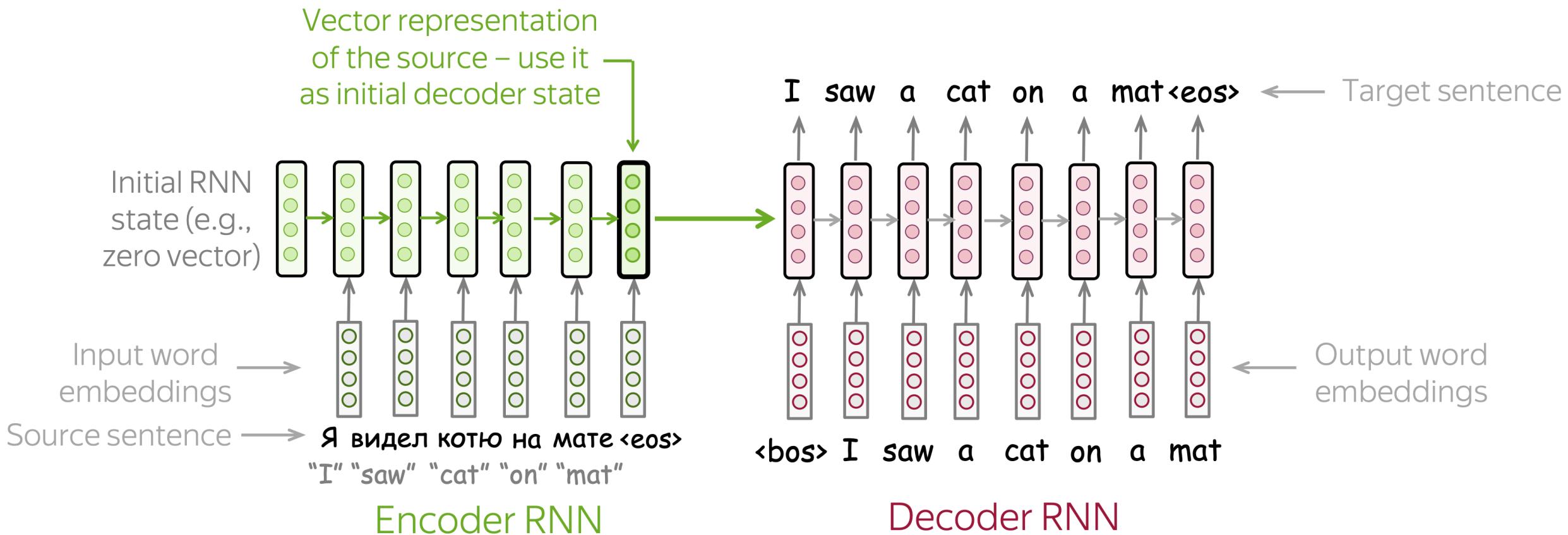
Машинный перевод

# Encoder-Decoder



- ▶ На вход подаем текст на одном языке
- ▶ Encoder получает векторное представление «смысла» текста
- ▶ Полученный эмбеддинг подаем на вход Decoder
- ▶ Decoder генерирует перевод с учетом «смысла»

# Архитектура RNN для перевода



Источник: [https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Архитектура RNN для перевода

Особенности:

- ▶ Encoder и Decoder – многослойные LSTM, GRU
- ▶ Размерность скрытых векторов порядка 1000
- ▶ Beam search для генерации

Недостатки:

- ▶ Нужно сжать весь текст в один вектор
- ▶ Теряется информация о первых словах
- ▶ Декодер тоже может терять информацию при генерации