

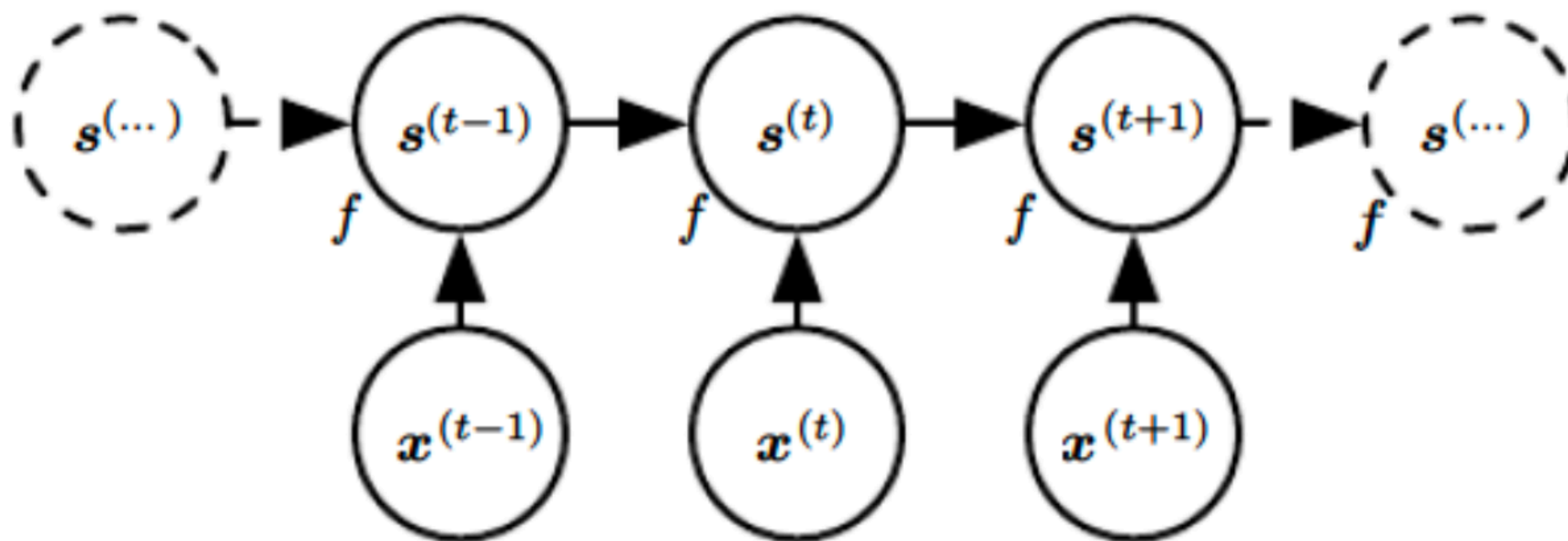
**RNN**

**Буткемп кодИИм**

# Мотивация

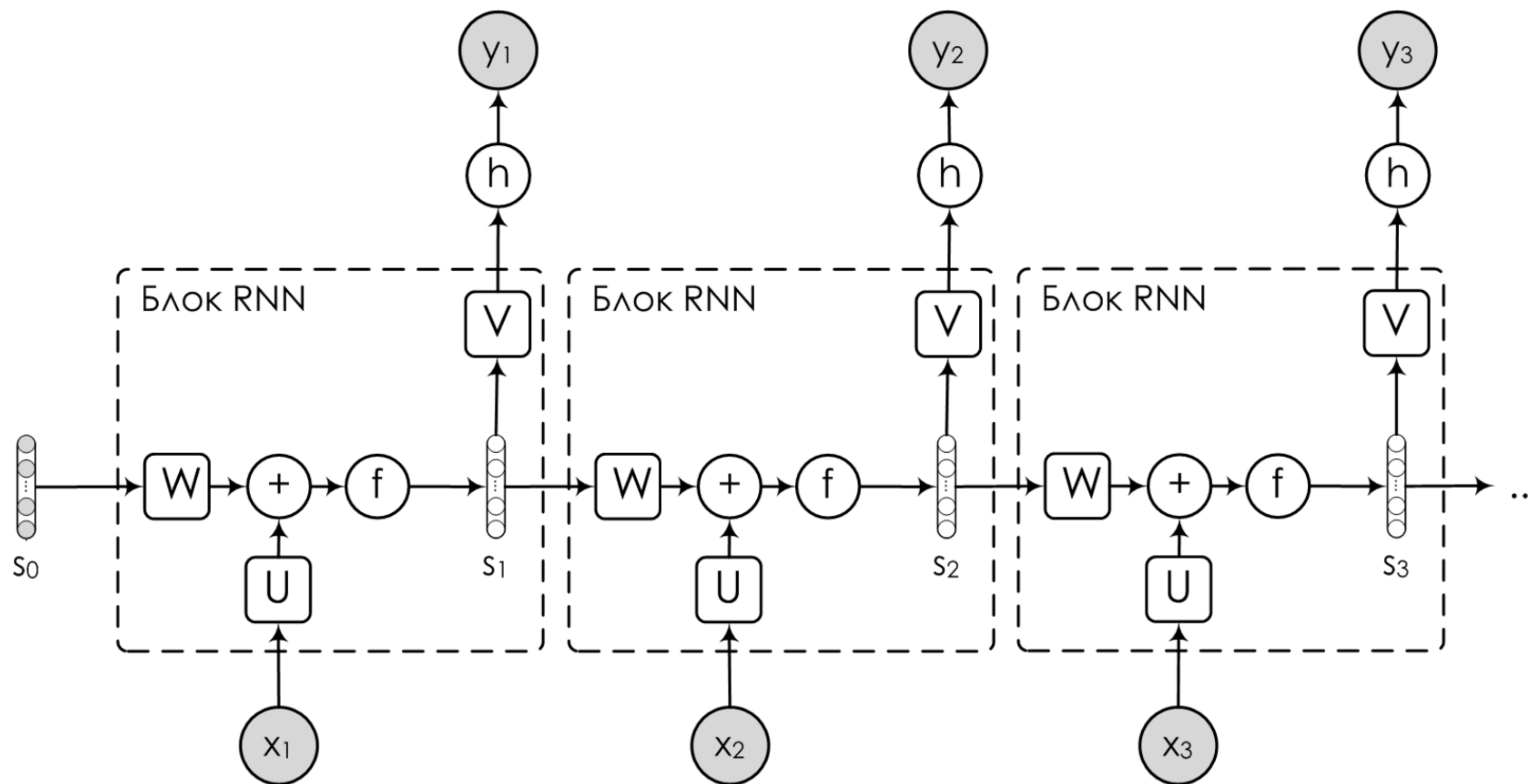
- задачи, где встречаются последовательности
  - временные ряды
  - звуки
  - тексты!

# Состояние меняется от входа



$$s^{(t+1)} = f(s^{(t)}, x^{(t+1)}; \theta)$$

# Простая RNN



# Простая RNN

- Применяем **одно и то же** преобразование и общие веса на разных шагах и входах на них.
- На каждом шаге - берем предыдущее состояние  $s^t$ , текущий вход  $x^{t+1}$
- По ним вычисляем новое состояние  $s^{t+1}$  и выход  $y^{t+1}$

# Простая RNN

Что получили?

- Последовательности любой длины!
- Учитываем контекст

# Почему плохо работает?

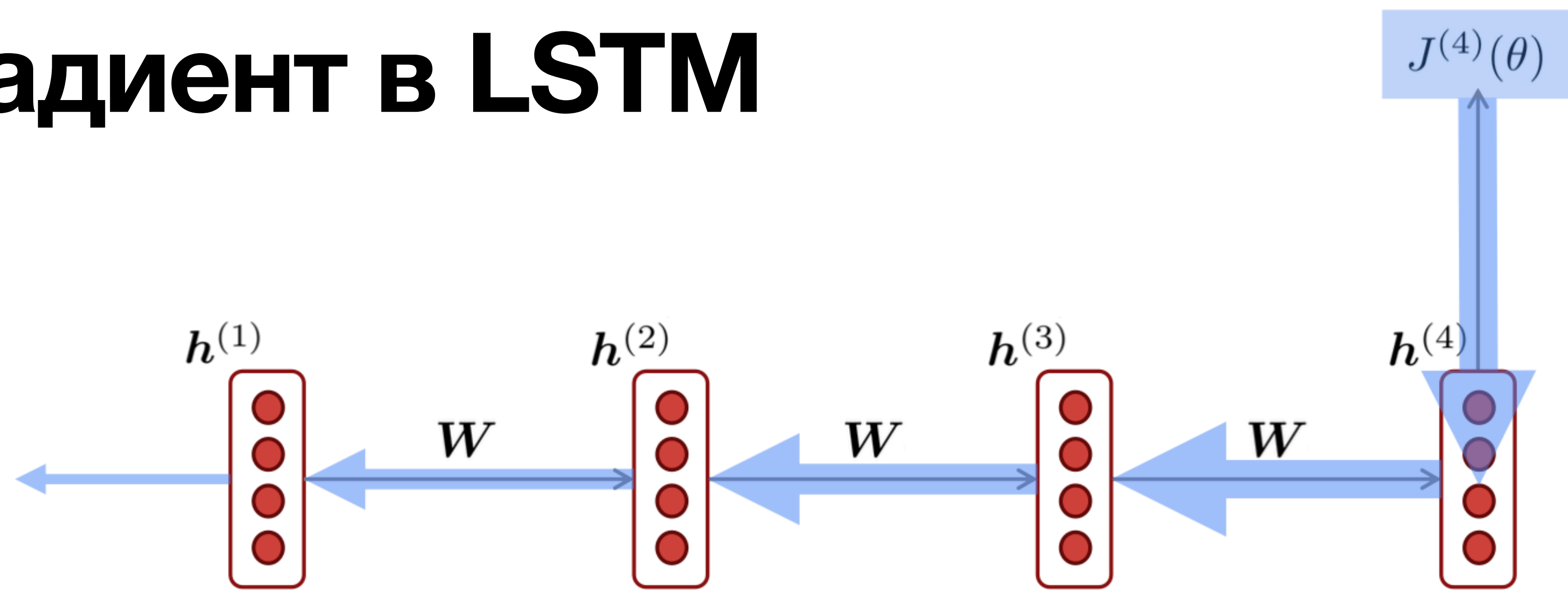
- не учитывает далекие зависимости
- градиенты
  - затухают
  - взрываются

# Почему плохо работает?

- не учитывает далекие зависимости
- градиенты
  - затухают
  - взрываются - **обрежем**



# Градиент в LSTM



$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \boxed{\frac{\partial h^{(2)}}{\partial h^{(1)}}} \times \boxed{\frac{\partial h^{(3)}}{\partial h^{(2)}}} \times \boxed{\frac{\partial h^{(4)}}{\partial h^{(3)}}} \times \frac{\partial J^{(4)}}{\partial h^{(4)}}$$

What happens if these are small?

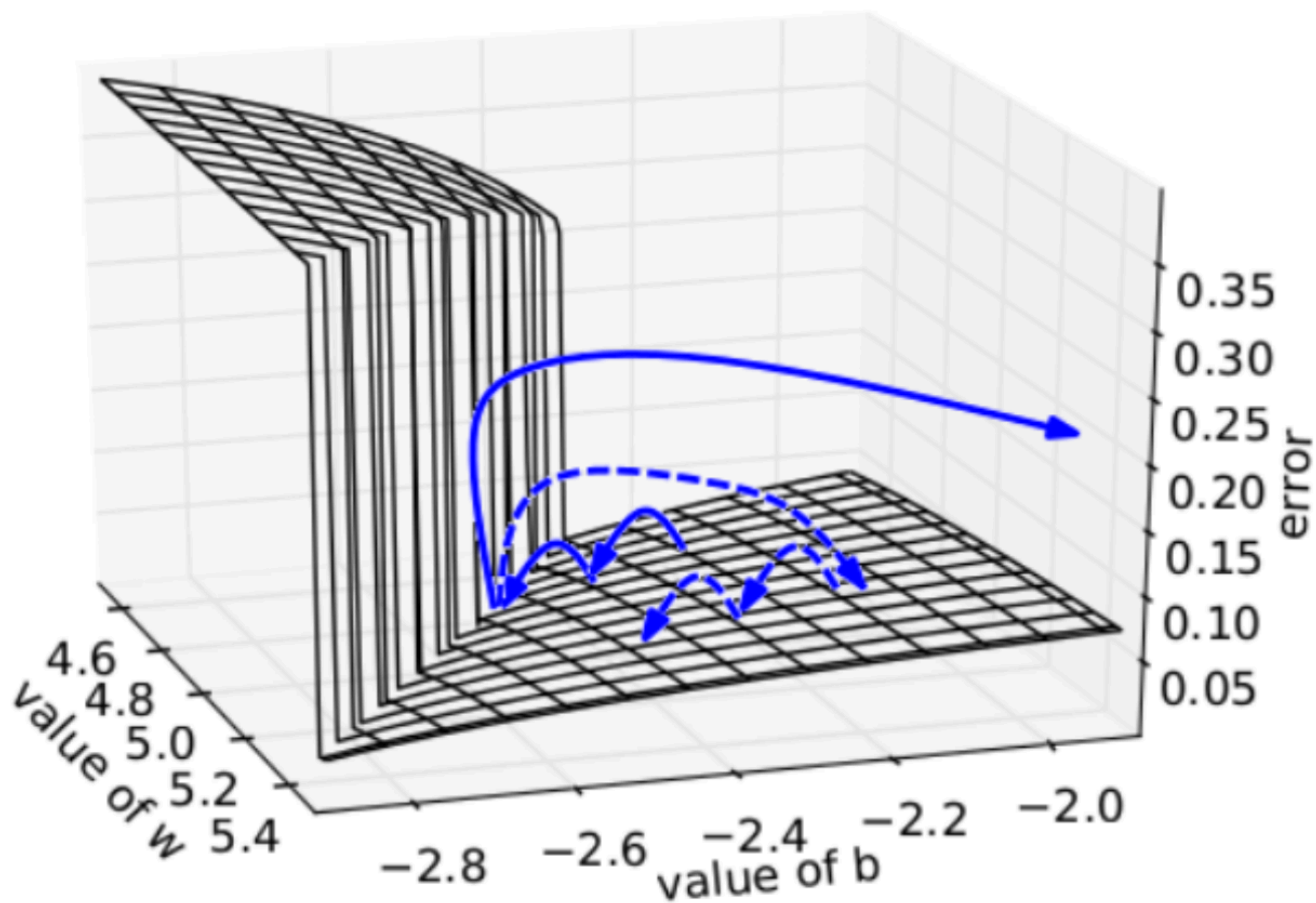
**Vanishing gradient problem:**  
When these are small, the gradient signal gets smaller and smaller as it backpropagates further

# Gradient clipping

- Чтобы градиент не взрывался нужно всего лишь...
  - его ограничить)
  - по значению
    - `sgd = optimizers.SGD(lr=0.01, clipvalue=0.05)`
  - или по всей норме
    - `sgd = optimizers.SGD(lr=0.01, clipnorm=1.)`

# Gradient clipping

Вот что будет происходить

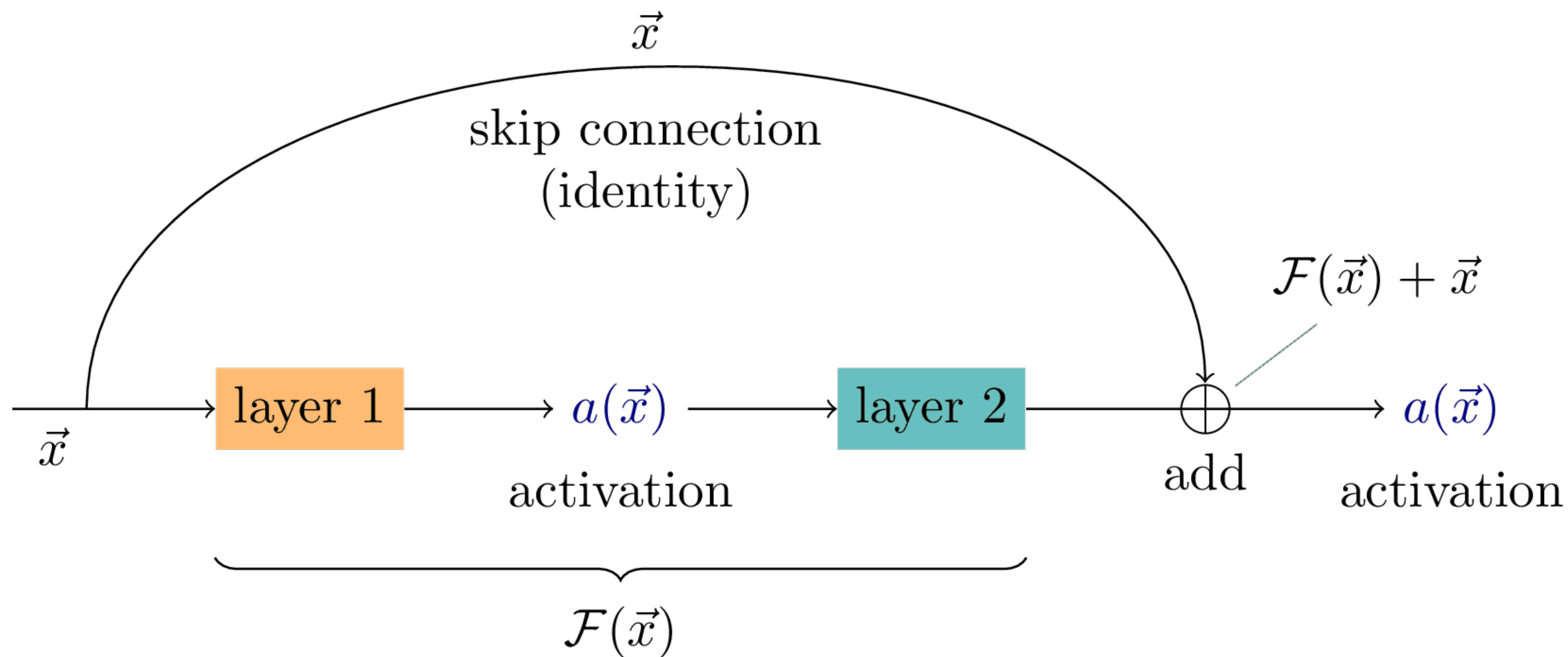


# Почему плохо работает?

- не учитывает далекие зависимости
- градиенты
  - затухают - добавим skip-connections
  - взрываются - обрежем

# Skip-connections

Что это и как работает?

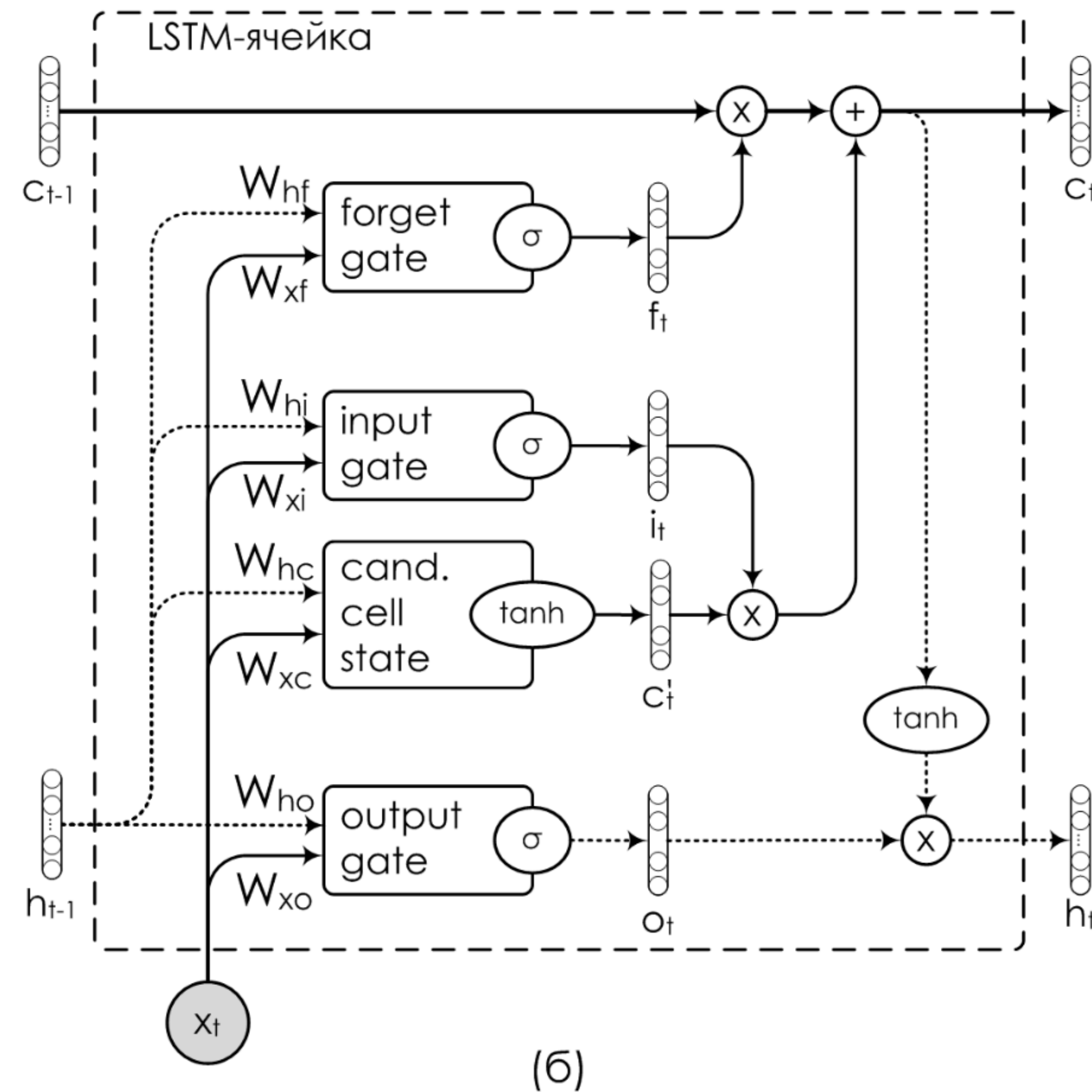
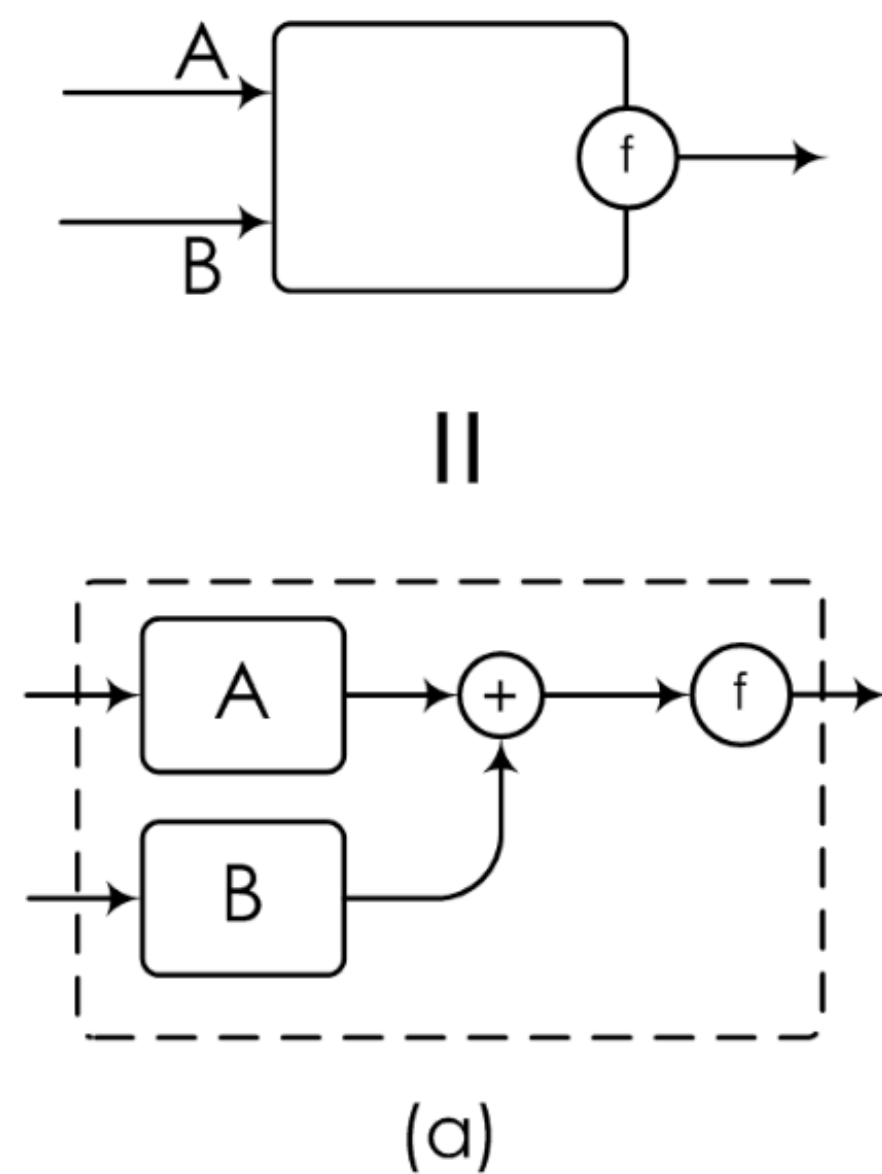


# Почему плохо работает?

- не учитывает далекие зависимости — придумаем «память»
- градиенты
  - затухают - добавим skip-connections
  - взрываются - обрежем

# Long short-term memory

В карточках



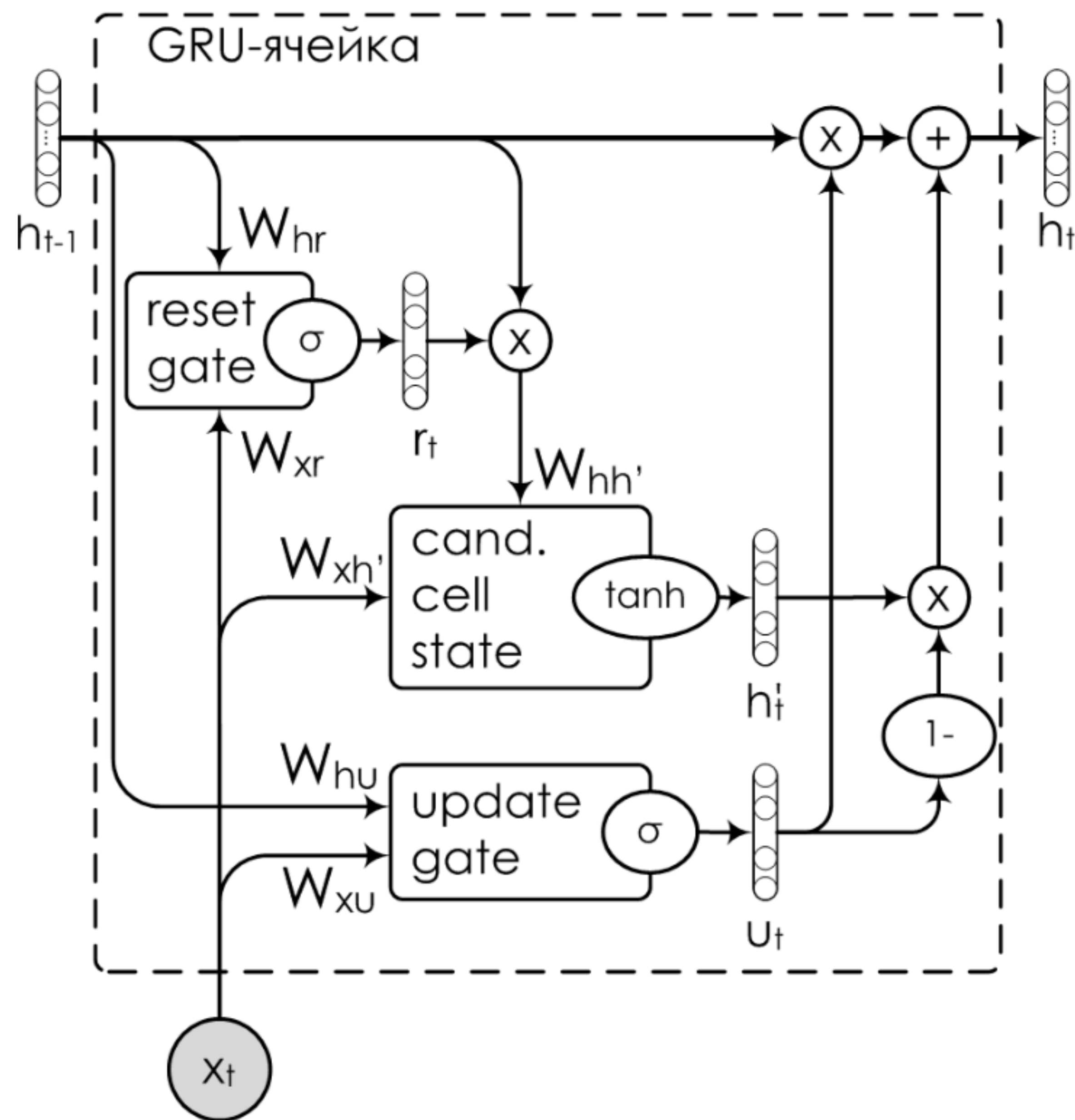
# LSTM

- На самом деле, тут огромное пространство для вариантов, как расположить гейты, какие активации использовать и тд..
- Некоторые более простые архитектуры не сильно проигрывают LSTM (Greff et al., 2015). И это приводит нас к...



# GRU

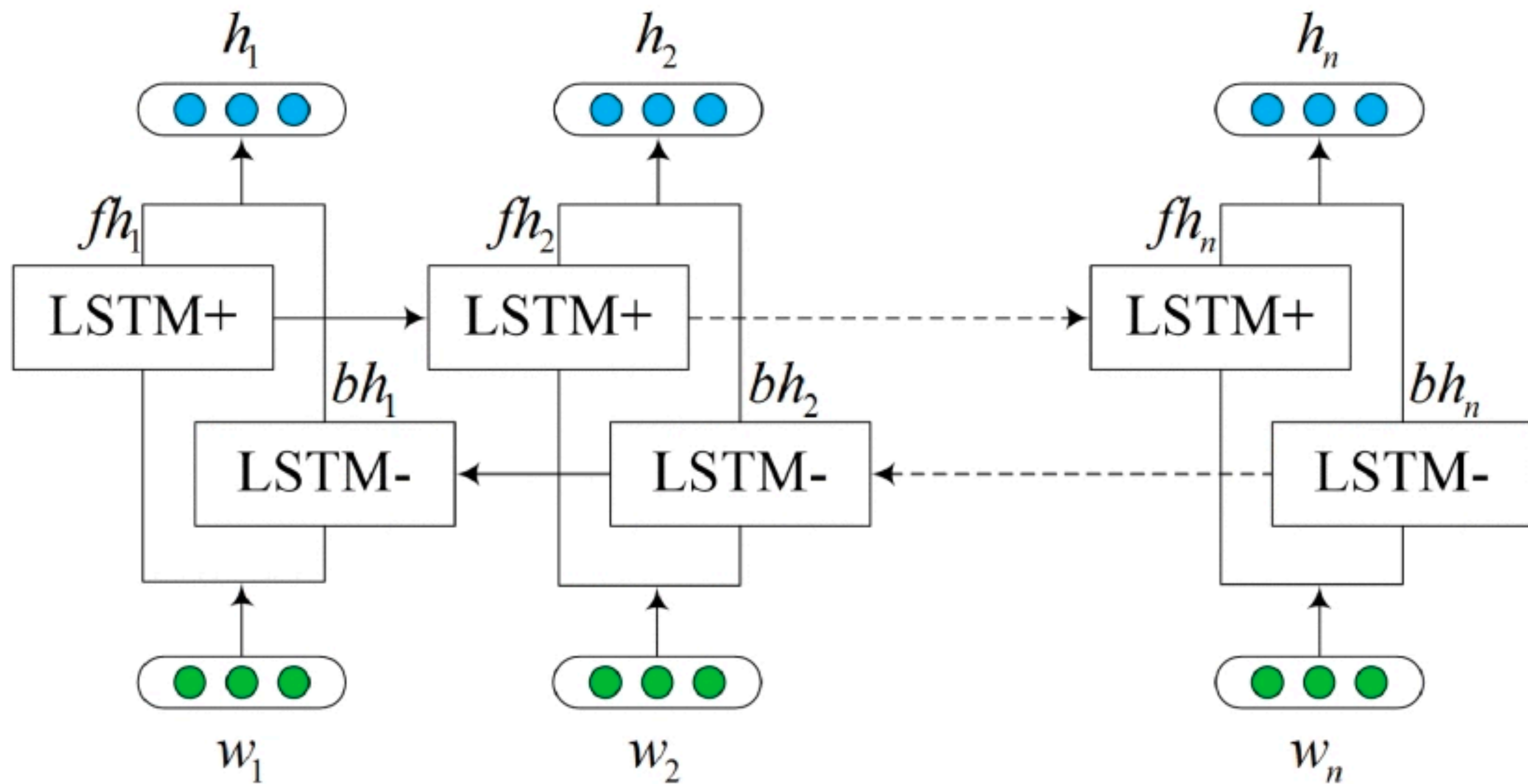
- Меньше, быстрее
- По качеству не сильно хуже LSTM
- Немного хуже работает с длинными последовательностями



# Bidirectional RNN

- Иногда результат зависит от последовательности **целиком**
- Например в распознавании речи - чтобы понять слово мы опираемся и на сказанные после этого слова
- Чтобы обрабатывать последовательности с обеих сторон и были предложены BiRNN

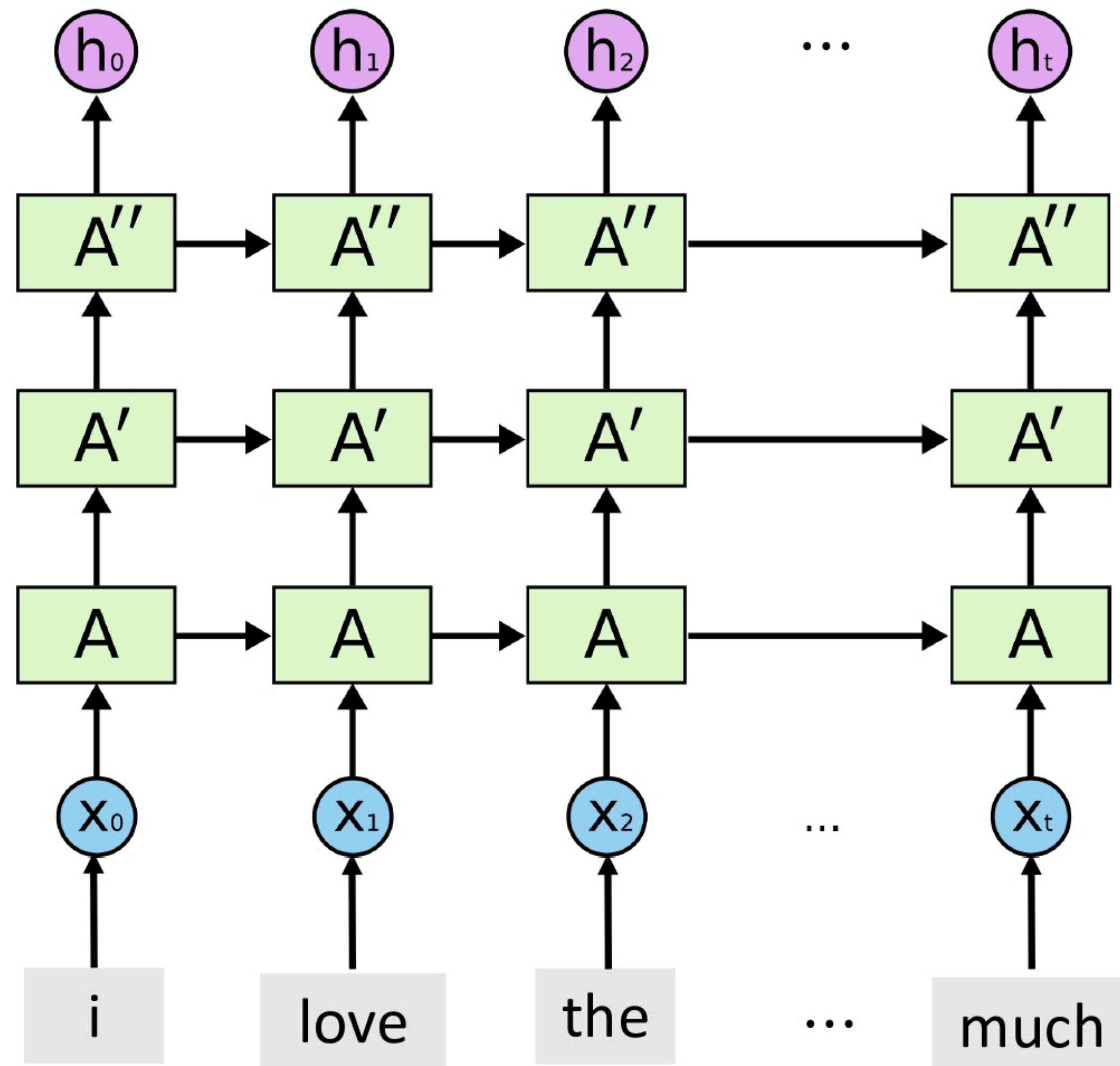
# Bidirectional RNN



# Stacked RNN

- А как будет выглядеть двухслойная RNN? А многослойная?
- Мы же смогли добавить еще одну RNN
- Давайте добавим еще, только теперь **выходы** первой сети — будут **входами** второй

# Stacked RNN



# Визуализируем RNN

<https://distill.pub/2019/memorization-in-rnns/>

context the formal study of grammar is an important part of  
education

Nested  
LSTM

context the formal study of grammar is an important part of  
education

LSTM

context the formal study of grammar is an important part of  
education

GRU

**Спасибо!**