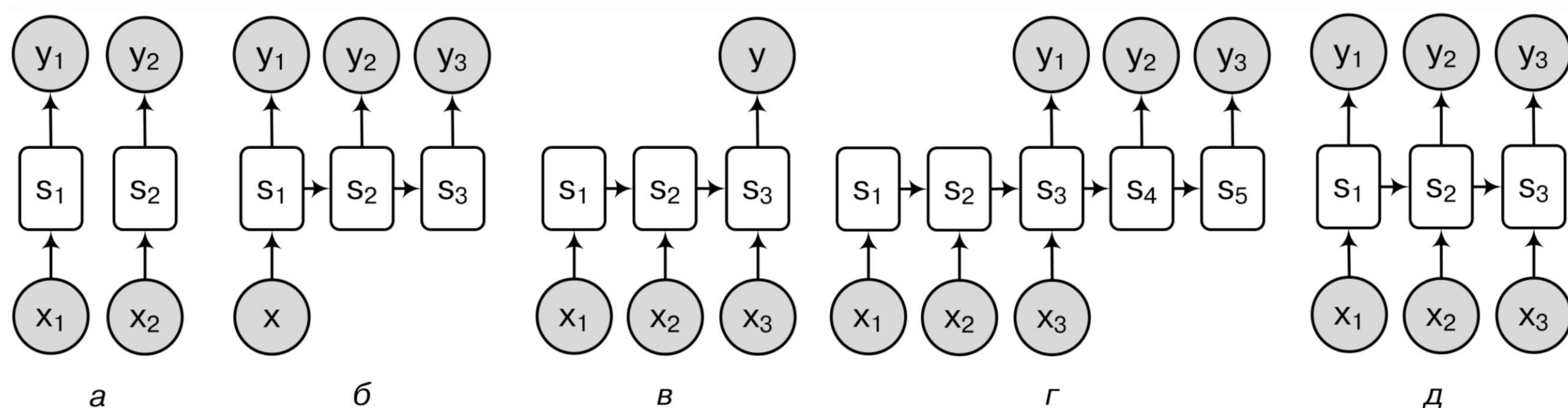


## План на сегодня

1. **Последовательности: задачи и данные**
2. Рекуррентные нейронные сети
3. Ячейки, слои и свойства
4. Практика

# Обработка последовательностей

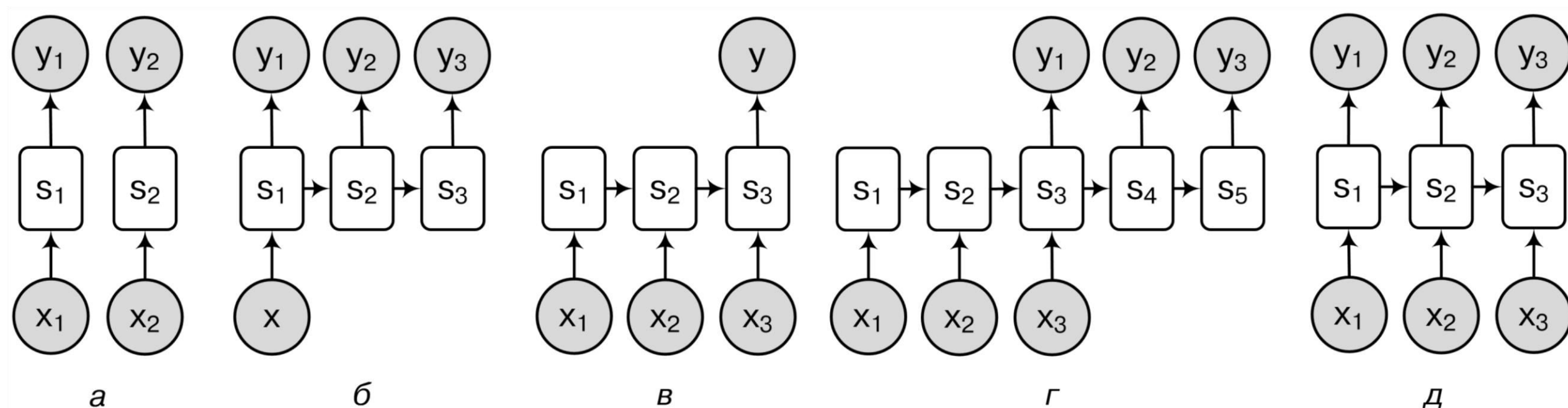


Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

**Любая задача может быть сведена к этой!**

# Обработка последовательностей

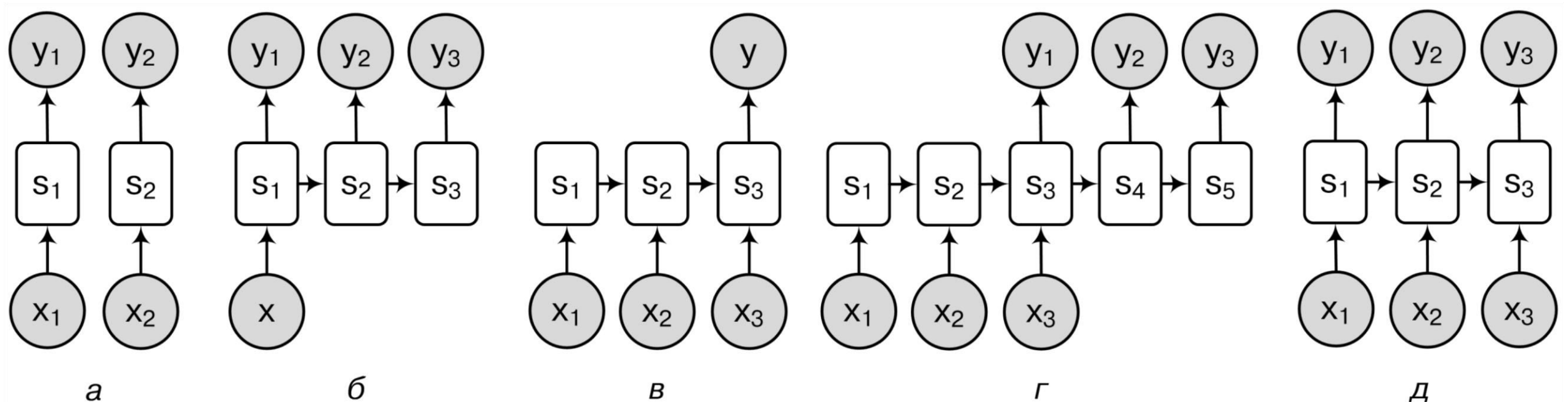


Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример?

# Обработка последовательностей

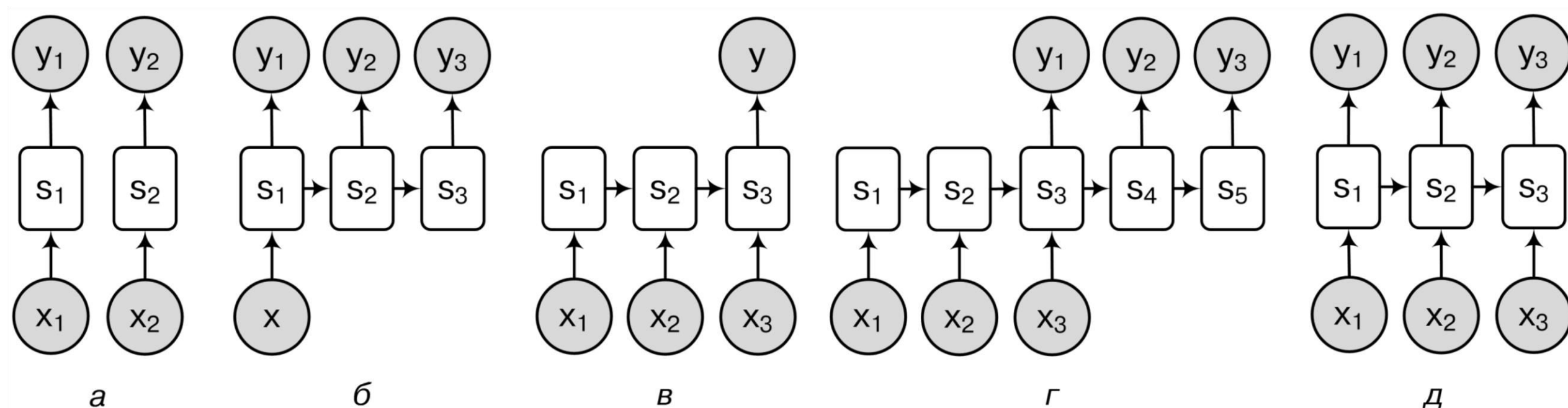


Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример: **аннотация изображения**

# Обработка последовательностей



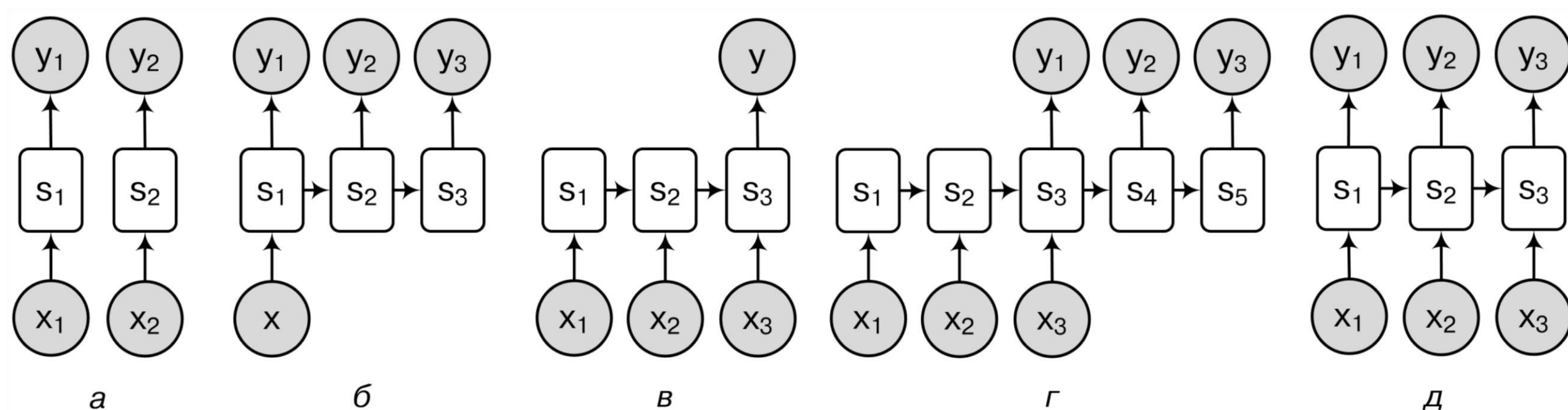
Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример?

# Обработка последовательностей



Задачи обработки последовательностей можно разделить на 5 видов

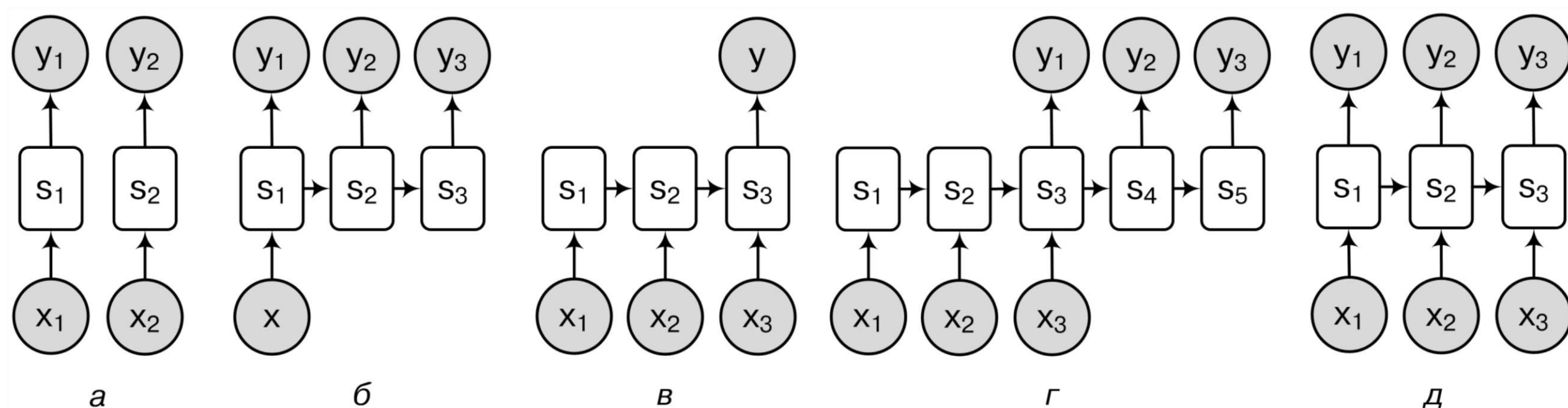
а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример: **классификация**



# Обработка последовательностей



Задачи обработки последовательностей можно разделить на 5 видов

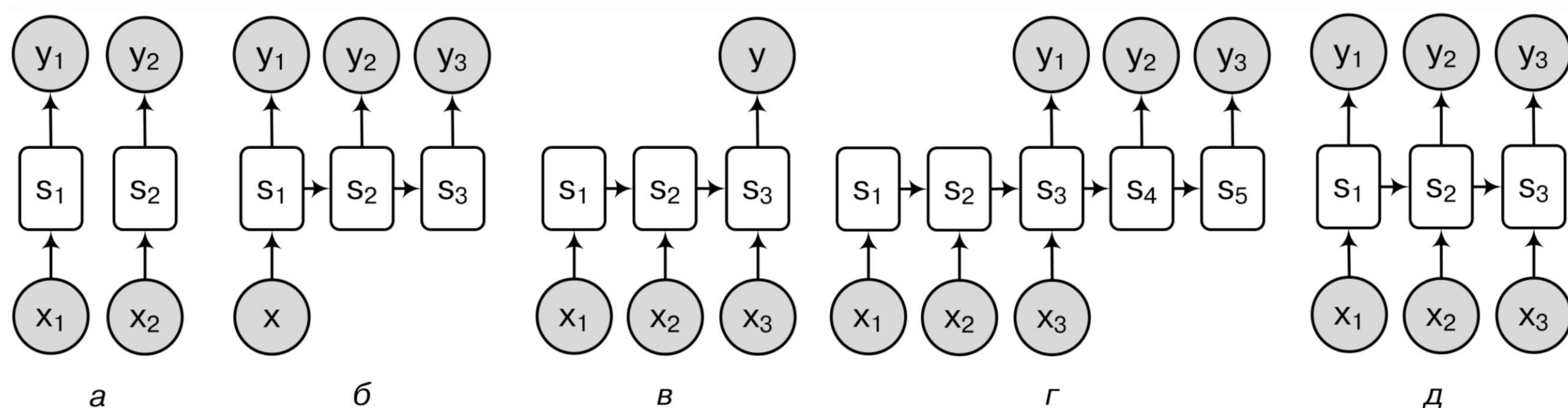
а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример: классификация

г. Последовательность входов — последовательность выходов. Пример?

# Обработка последовательностей



Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

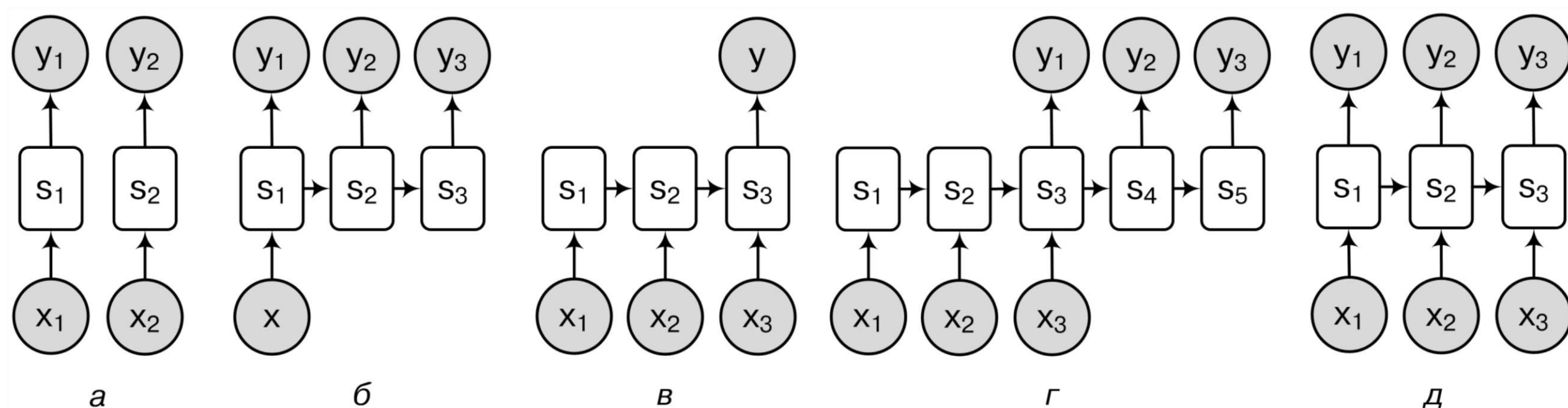
б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример: классификация

г. Последовательность входов — последовательность выходов. Пример: **перевод**



# Обработка последовательностей



Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

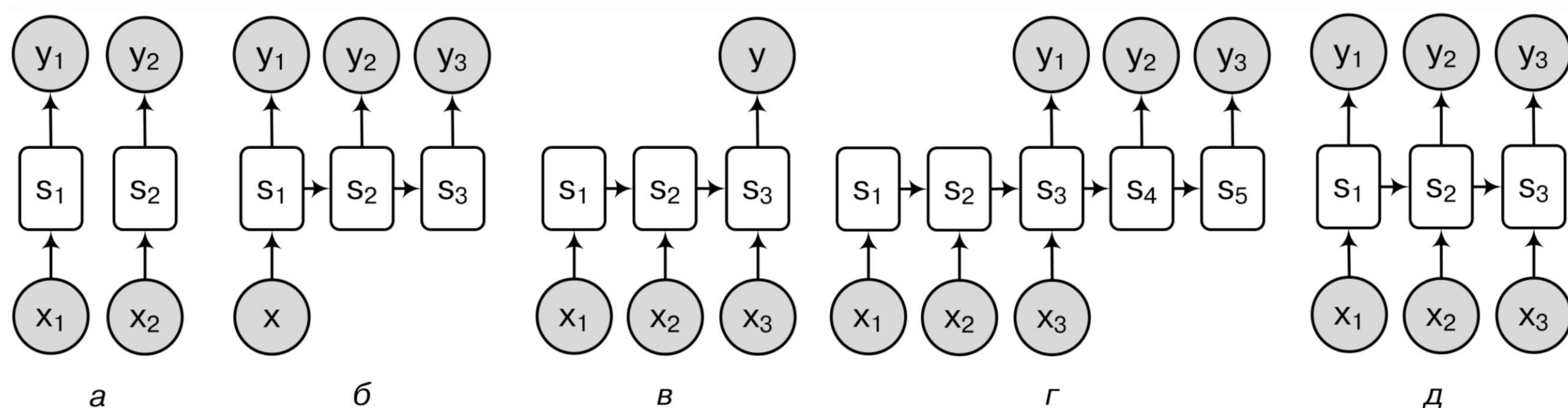
б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример: классификация

г. Последовательность входов — последовательность выходов. Пример: перевод

д. Синхронизированные последовательности. Пример?

# Обработка последовательностей



Задачи обработки последовательностей можно разделить на 5 видов

а. Один вход — один выход. Обычная нейросеть.

б. Один вход — последовательность выходов. Пример: аннотация изображения

в. Последовательность входов — один выход. Пример: классификация

г. Последовательность входов — последовательность выходов. Пример: перевод

д. Синхронизированные последовательности. Пример: **Аннотация видео**

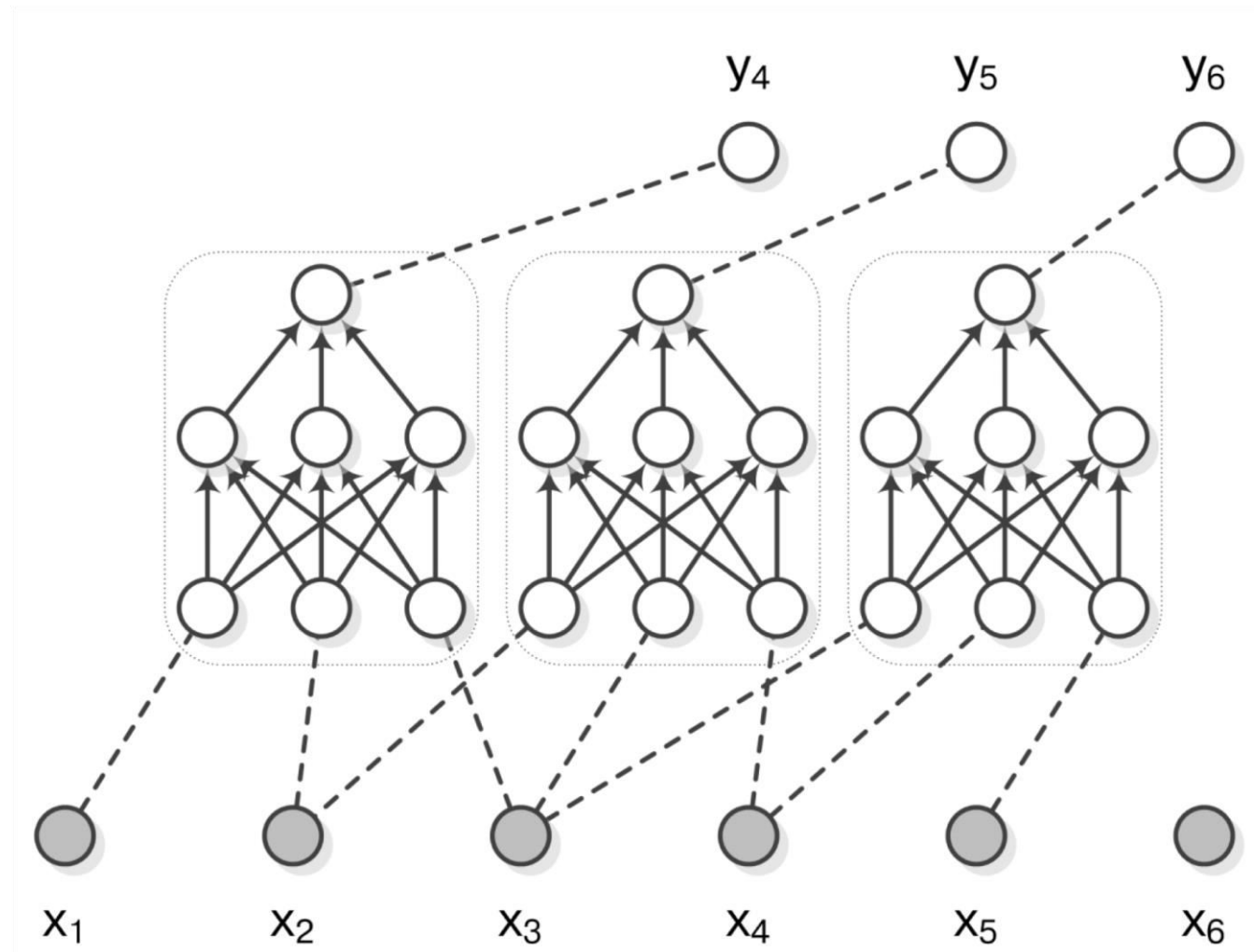
# Простое решение

Зафиксировав длину последовательности мы можем предсказывать ее свойства или очередной элемент с помощью многослойной полносвязной нейросети.

**Какие проблемы?**

# Простое решение

Зафиксировав длину последовательности мы можем предсказывать ее свойства или очередной элемент с помощью многослойной полносвязной нейросети. Можно делать свертки по времени!



# Представление данных

Если корпус текстов содержит ограниченное количество слов, то каждое слово может быть представлено **one-hot** вектором размерности равной количеству различных слов в корпусе. Для больших словарей проще хранить **id** слова. Для этого в PyTorch есть слой `nn.Embeddings`

**Проблемы?**

# Представление данных

Если корпус текстов содержит ограниченное количество слов, то каждое слово может быть представлено **one-hot** вектором размерности равной количеству различных слов в корпусе. Для больших словарей проще хранить **id** слова.

1. Слов очень много  $\sim 10^5$
2. Словоформ очень много(особенно в русском языке)
3. Редкие слова — очень редкие.



# Представление данных

Если корпус текстов содержит ограниченное количество слов, то каждое слово может быть представлено **one-hot** вектором размерности равной количеству различных слов в корпусе. Для больших словарей проще хранить **id** слова.

Слово может быть представлено списком **n-грамм**:

#слово# —> [#сл, сло, лов, ово, во#]

**Преимущества?**

# Представление данных

Если корпус текстов содержит ограниченное количество слов, то каждое слово может быть представлено **one-hot** вектором размерности равной количеству различных слов в корпусе. Для больших словарей проще хранить **id** слова.

Слово может быть представлено списком **n-грамм**:

#слово# —> [#сл, сло, лов, ово, во#]

1. Словарь **n-грамм** почти всегда сильно меньше обычного. ~10\$
2. Словоформы похожи!
3. Редкие слова сразу! имеют смысл

**Проблемы?**

# Представление данных

Если корпус текстов содержит ограниченное количество слов, то каждое слово может быть представлено **one-hot** вектором размерности равной количеству различных слов в корпусе. Для больших словарей проще хранить **id** слова.

Слово может быть представлено списком **n-грамм**:

#слово# —> [#сл, сло, лов, ово, во#]

1. Словарь **n-грамм** почти всегда сильно меньше обычного. ~10\$

2. Словоформы похожи!

3. Редкие слова сразу! имеют смысл

Проблемы:

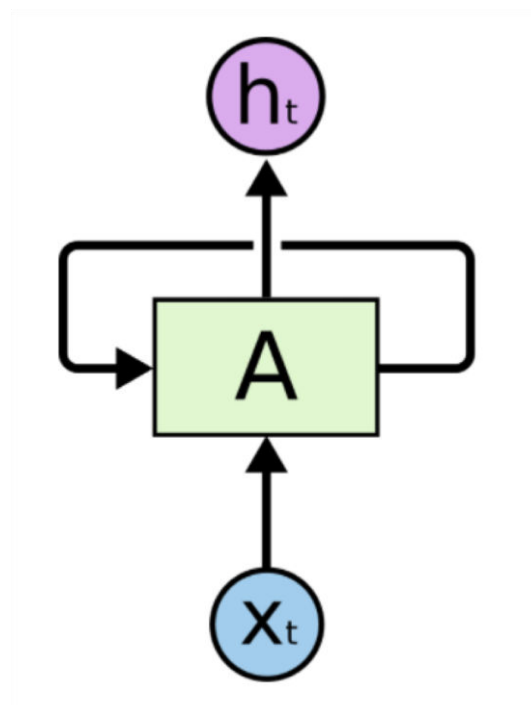
4. Разное количество токенов в слове —> разный уровень активации в первом слое

5. Коллизии. (на самом деле их очень мало)

# План на сегодня

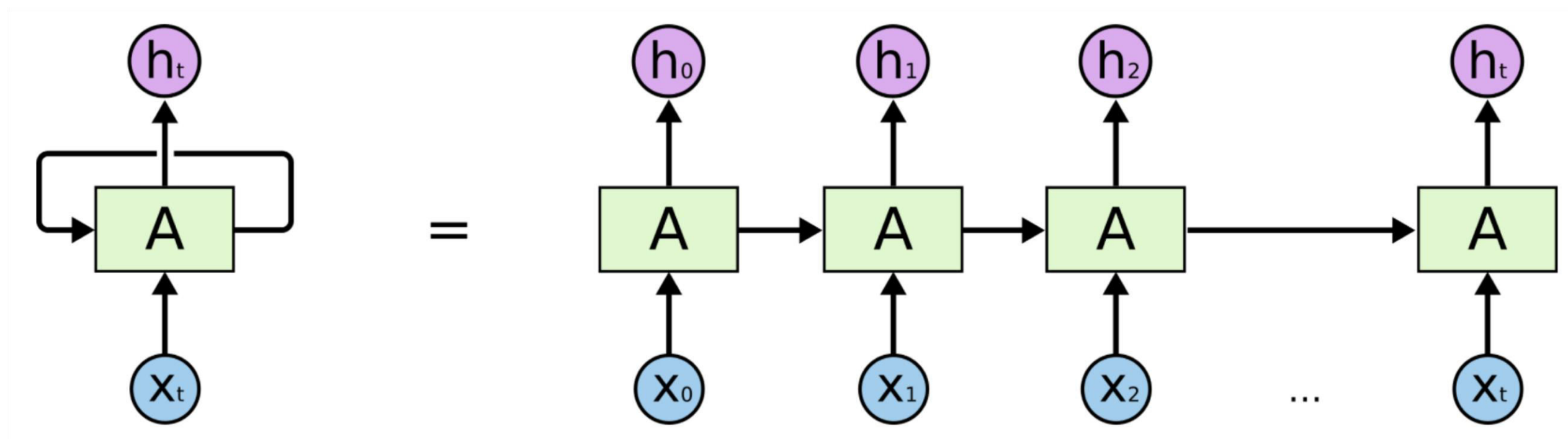
1. Последовательности: задачи и данные
- 2. Рекуррентные нейронные сети**
3. Ячейки, слои и свойства
4. Практика

# Рекуррентная нейронная сеть



А — ячейка рекуррентной нейронной сети, она получает на вход очередной элемент последовательности вместе с выходом с предыдущего шага и возвращает какой-то результат.

# Рекуррентная нейронная сеть



А — ячейка рекуррентной нейронной сети, она получает на вход очередной элемент последовательности вместе с выходом с предыдущего шага и возвращает какой-то результат.

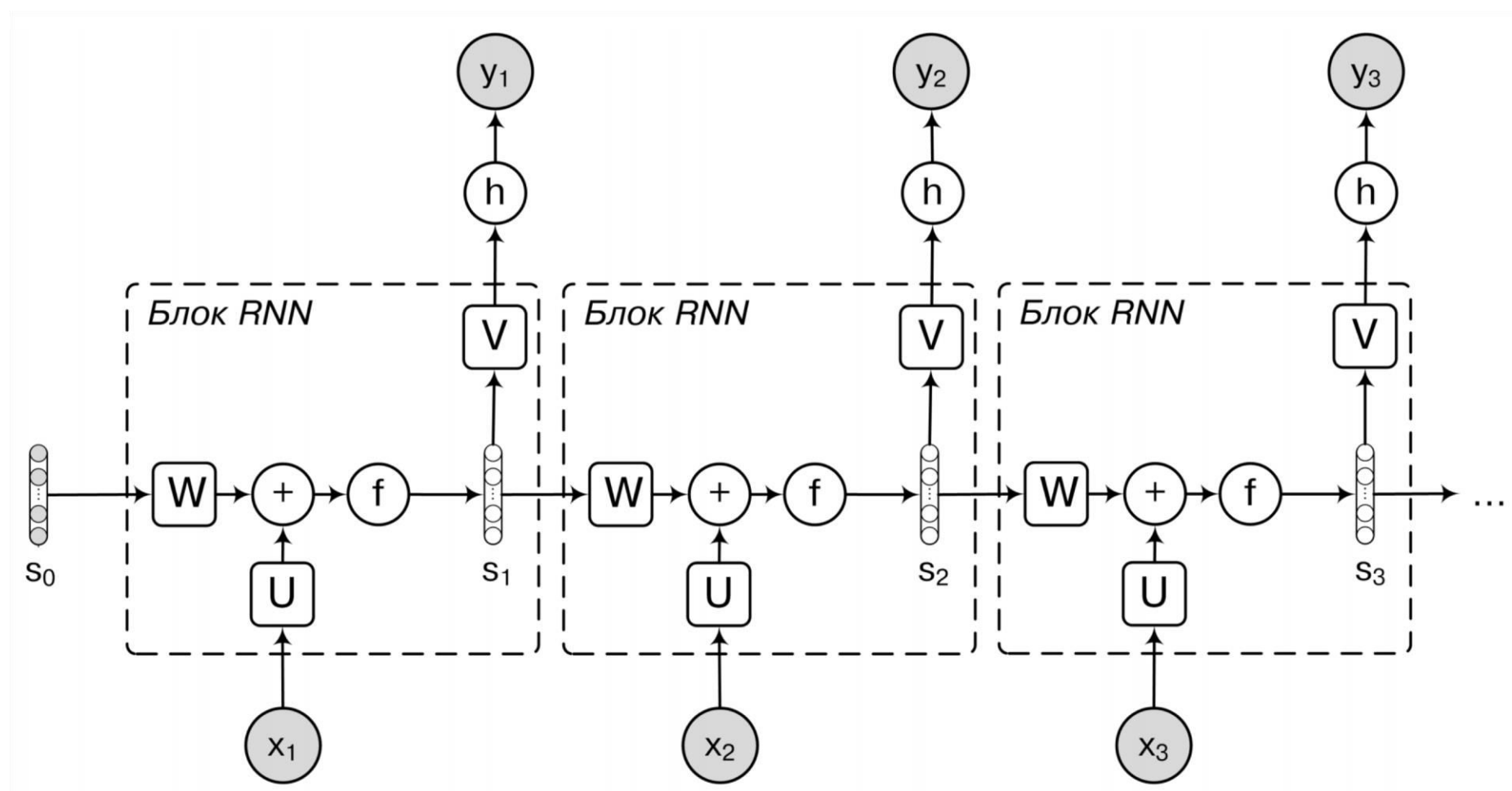
Рекуррентная нейронная сеть подразумевает циклы. Но на самом деле, когда мы говорим про обработку последовательностей, мы можем развернуть цикл в последовательные вычисления и применять стандартные методы обучения.



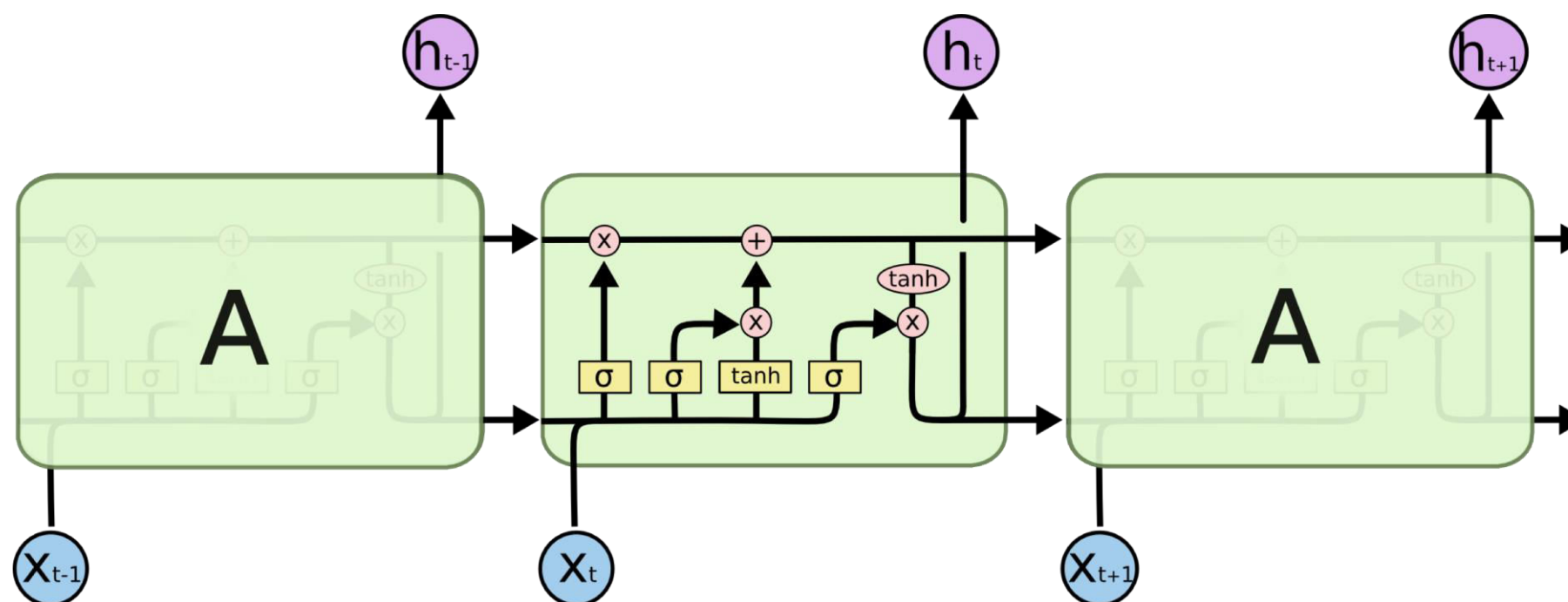
## План на сегодня

1. Последовательности: задачи и данные
2. Рекуррентные нейронные сети
- 3. Ячейки, слои и свойства**
4. Практика

# Простейшая рекуррентная ячейка



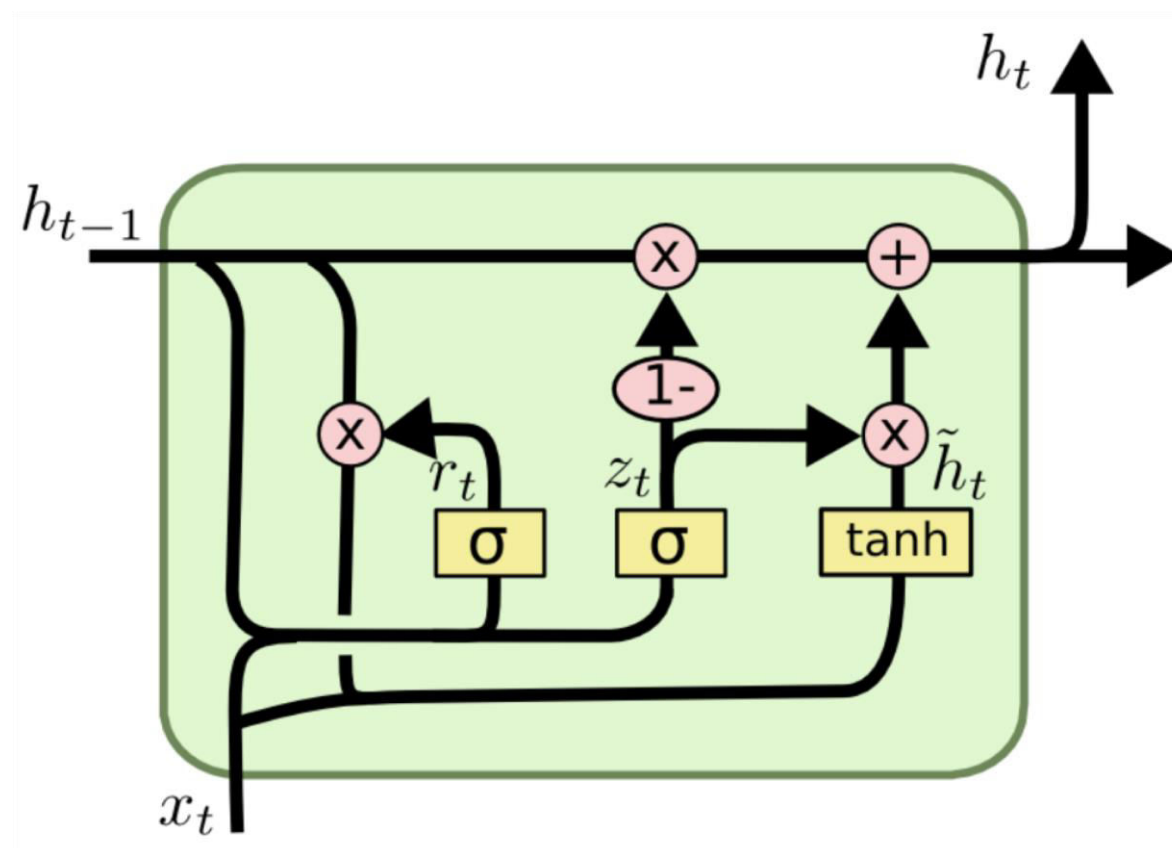
# Long Short-Term Memory



**LSTM**-ячейка устроена намного сложнее. Она содержит 3 основных вида гейтов:

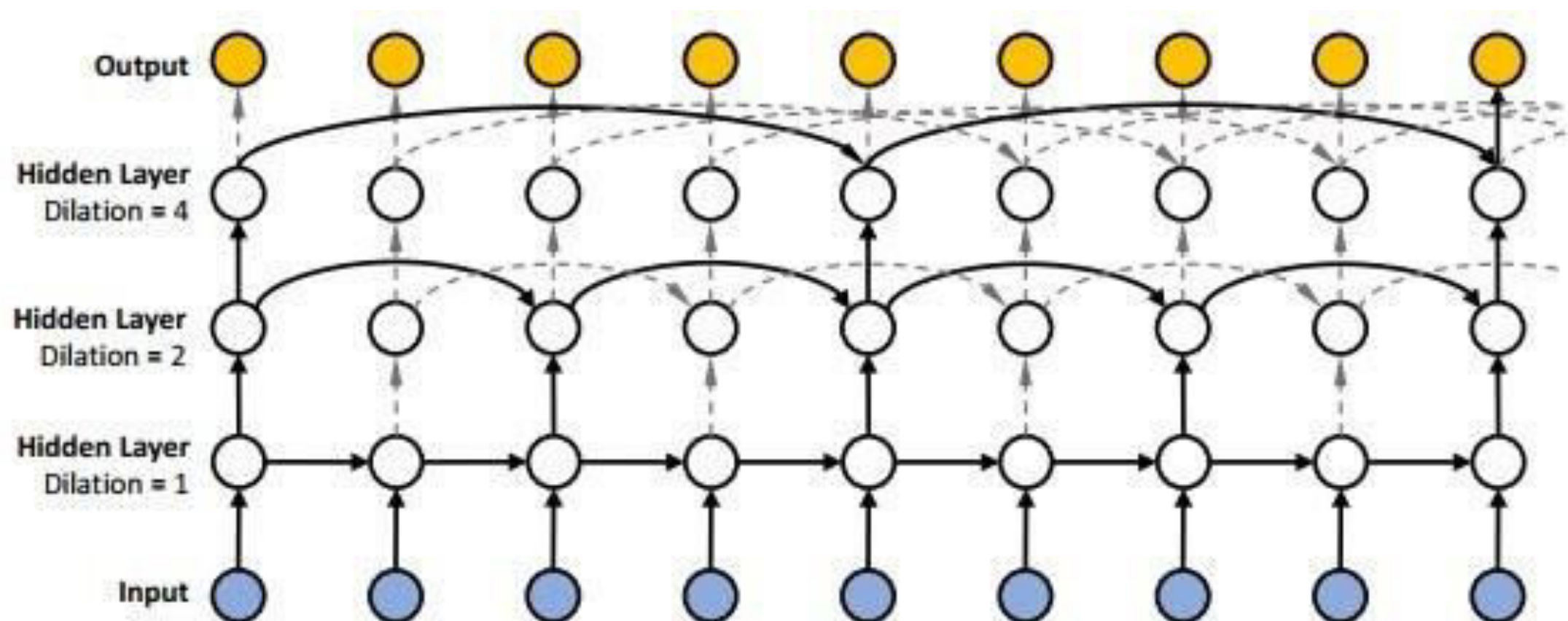
1. **Забывающий гейт** «чистит» память.
2. **Входной гейт** выбирает какую новую информацию сохранить в ячейке.
3. **Выходной гейт** выбирает что ячейка выдаст в качестве выходов.

# Gated Recurrent Unit



Несмотря на визуально более сложную структуру ячейки **GRU** требует меньшего количества параметров. Кроме того, **GRU** объединяет состояние ячейки и память в один вектор.

# Параметры



Так же как к сверточным сетям, к рекуррентным сетям применимы растяжение и шаг.





Спасибо  
за внимание!

