Рекуррентные нейронные сети

Нейросети для анализа текстов

Анализ текста как последовательности

Анализ текста полносвязной сетью

- Все токены текста поступают на вход каждому нейрону
- Токены анализируются изолированно друг от друга

Проблемные тексты для полносвязной сети:

- Overall, the movie is not bad and has entertainment value.
- Unfortunately, the movie is not so good.
- Ice cream (мороженое, ice – лед, cream – крем, сливки)

Необходимо анализировать текст как последовательность токенов

- Порядок слов/символов/предложений в тексте имеет большой смысл
- Нужны специальные архитектуры нейронных сетей для анализа последовательностей

Анализ текста как последовательности

Анализ текста полносвязной сетью

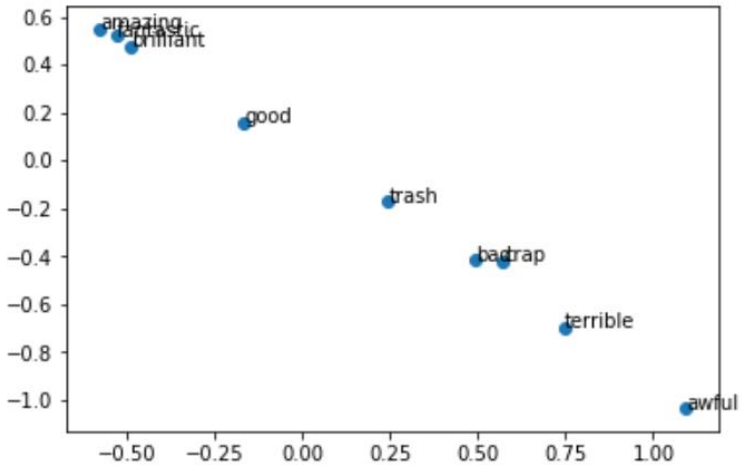
- Все токены текста поступают на вход каждому нейрону
- Токены анализируются изолированно друг от друга

Проблемные тексты для полносвязной сети:

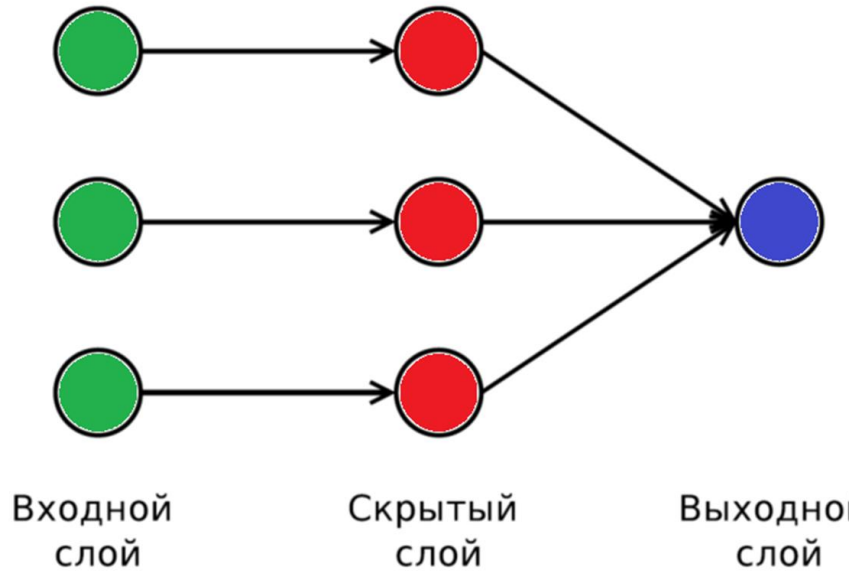
- Overall, the movie is not bad and has entertainment value.
- Unfortunately, the movie is not so good.
- Ice cream (мороженое, ice – лед, cream – крем, сливки)

Необходимо анализировать текст как последовательность токенов

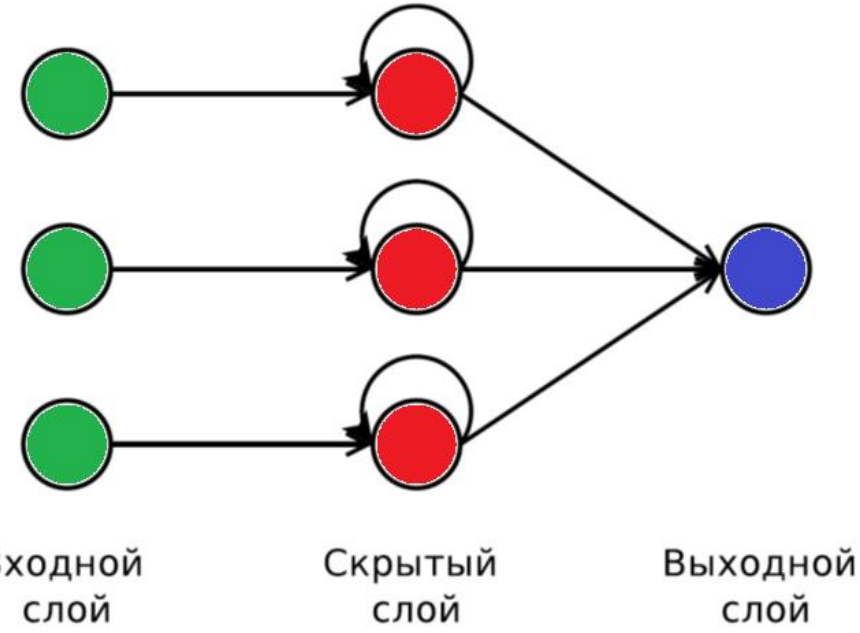
- Порядок слов/символов/предложений в тексте имеет большой смысл
- Нужны специальные архитектуры нейронных сетей для анализа последовательностей



Типы нейронных сетей

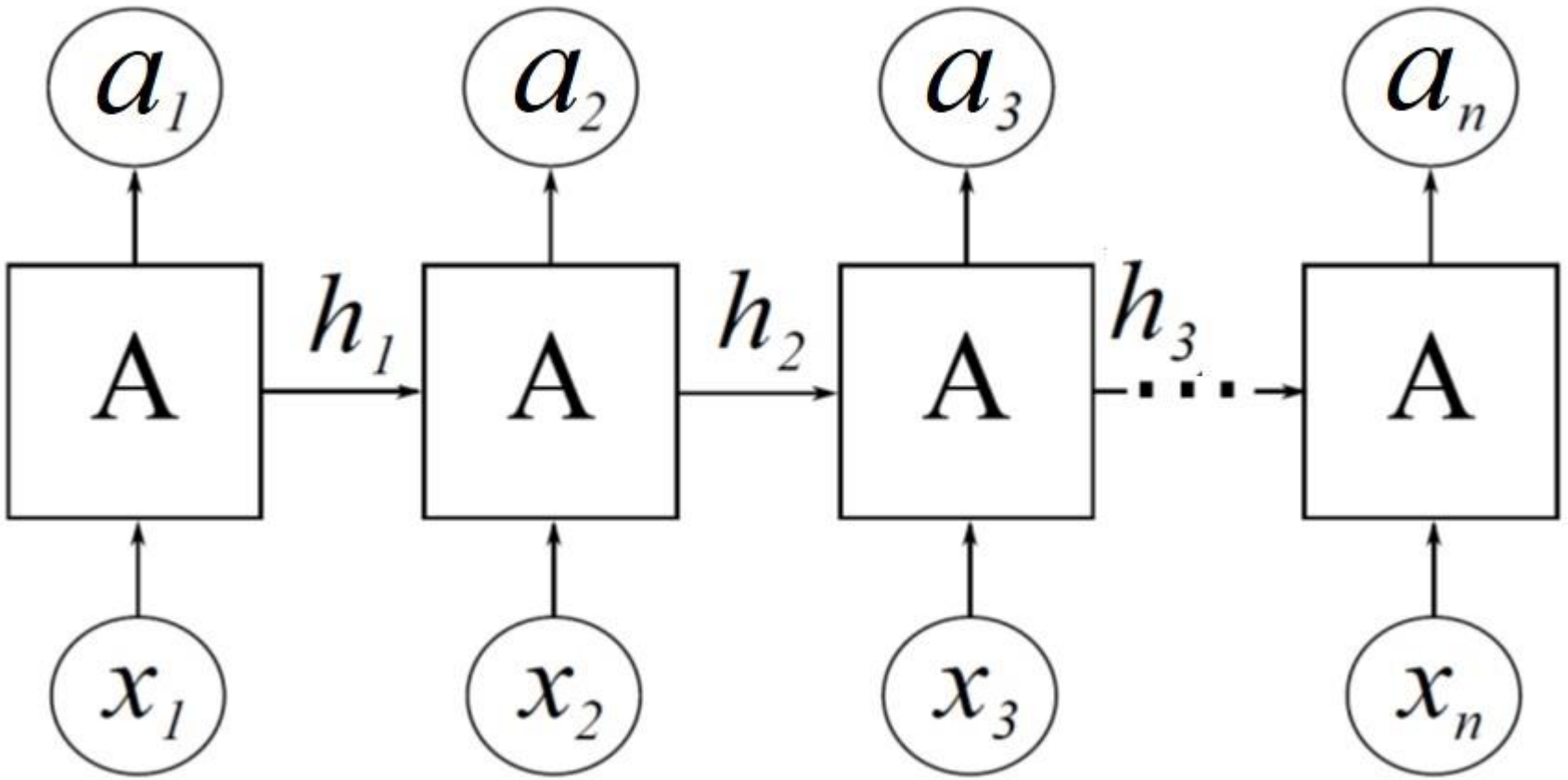


Сети с прямым распространением сигнала
(feedforward networks)



Рекуррентные сети
(recurrent networks)

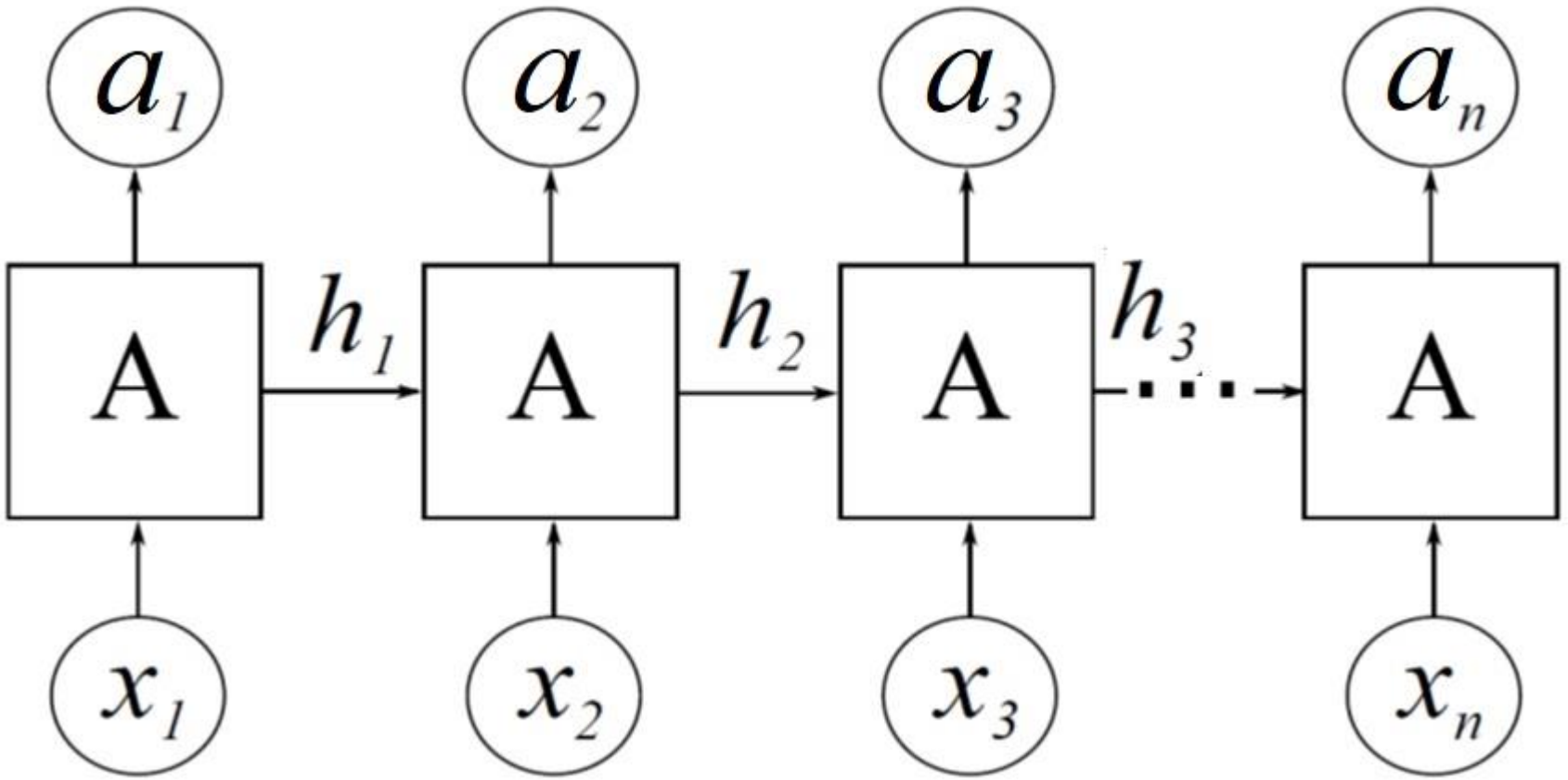
Разворачивание рекуррентной нейронной сети во времени



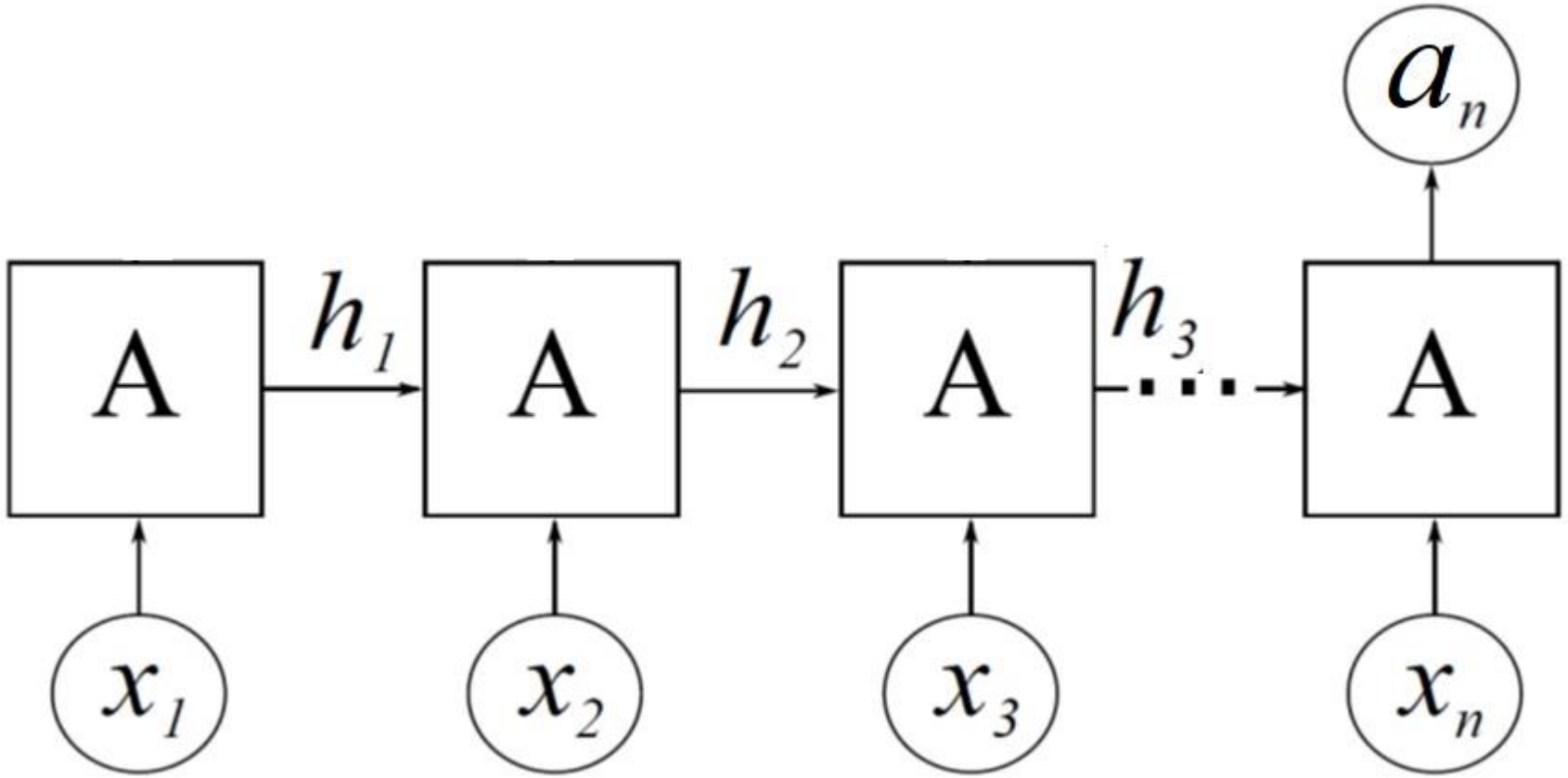
Рекуррентные нейросети в Keras

```
model = Sequential()
model.add(Embedding(input_dim=max_words,
                    output_dim=50,
                    input_length=maxlen))
model.add(SimpleRNN(8))
model.add(Dense(1, activation='sigmoid'))
```

Режим sequence to sequence



Режим sequence to vector



Режимы работы рекуррентных нейросетей в Keras

```
model = Sequential()
model.add(Embedding(input_dim=max_words,
                    output_dim=50,
                    input_length=maxlen))
model.add(SimpleRNN(16,
                    return_sequences=True))
model.add(SimpleRNN(8))
model.add(Dense(1, activation='sigmoid'))
```

Обучение рекуррентных нейронных сетей

Для обучения рекуррентных нейросетей используется разворачивание во времени и обратное распространение ошибки

- Количество слоев в развернутой сети зависит от длины последовательности входных данных

Проблема исчезающего градиента

- При передаче от слоя к слою сигнал об изменении весов уменьшается
- Сеть с большим количеством слоев сложно обучить

Более сложные архитектуры рекуррентных нейросетей

- LSTM (Long-Short Term Memory)
- GRU (Gated Recurrent Unit)

Итоги

Рекуррентные нейронные сети

- Сети с циклами

Анализ последовательностей

- Тексты, речь, временные ряды

Проблемы рекуррентных нейросетей

- Обучение требует длительного времени
- Проблема исчезающего градиента
- Ограниченная «длительность» запоминания предыдущей информации

Пути решения проблем

- Более совершенные архитектуры рекуррентных сетей LSTM и GRU
- Одномерные сверточные нейронные сети
- Механизм внимания (attention)