



Deep Learning School

Лекция 2

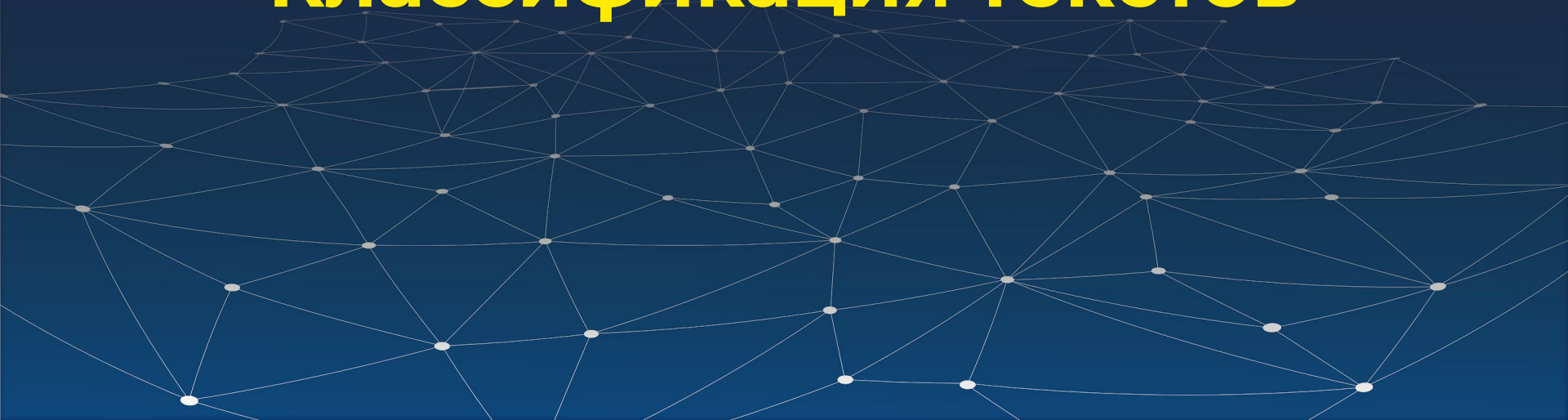
Классификация текста. Рекуррентные
нейронные сети и их применения

План лекции



- Классификация текста
- Рекуррентные нейронные сети
- LSTM, GRU

Классификация текстов



Зачем классифицировать текст?

- Просто чтобы классифицировать тексты

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
 - Спам-фильтр



Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
 - Спам-фильтр
 - Токсичные комментарии



Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
 - Спам-фильтр
 - Токсичные комментарии
 - Фейк-ньюс



Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
 - Спам-фильтр
 - Токсичные комментарии
 - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
 - Оценка тональности текста



Зачем классифицировать текст?

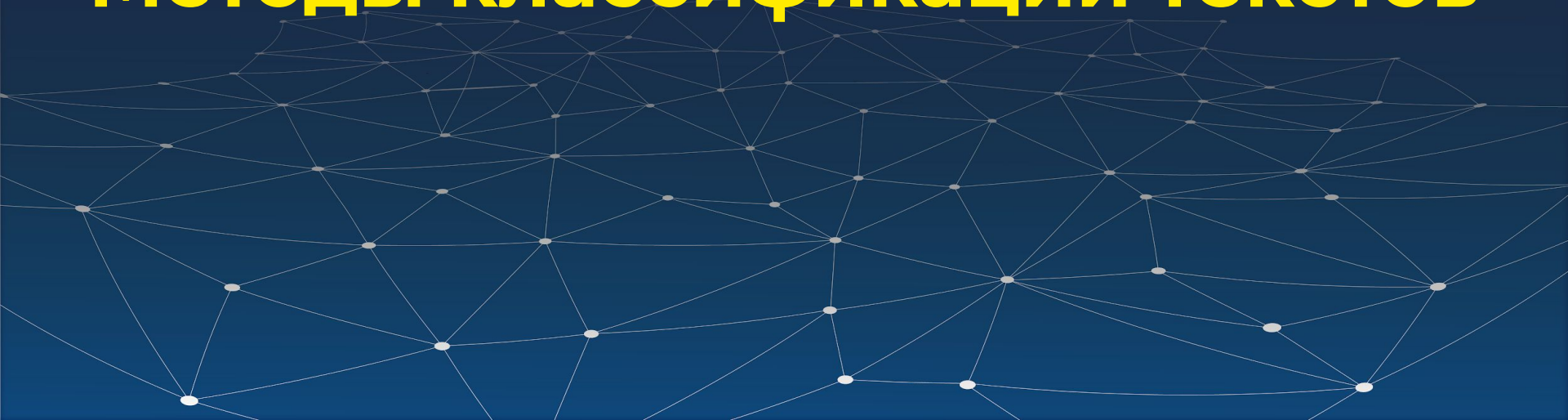
- Просто чтобы классифицировать тексты
 - Спам-фильтр
 - Токсичные комментарии
 - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
 - Оценка тональности текста
- Как часть другой задачи NLP
 - Фильтрация обучающей выборки

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
 - Спам-фильтр
 - Токсичные комментарии
 - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
 - Оценка тональности текста
- Как часть другой задачи NLP
 - Фильтрация обучающей выборки
 - Выбор сценария в диалоговой системе



Методы классификации текстов



Классические методы

- См. предыдущую лекцию
 - Bag of Words
 - tf-idf
 - Классические методы + эмбединги слов

Классические методы

- См. предыдущую лекцию
 - Bag of Words
 - tf-idf
 - Классические методы + эмбединги слов
- Проблемы
 - Большая размерность
 - Не учитывается порядок

CNN для классификации текстов

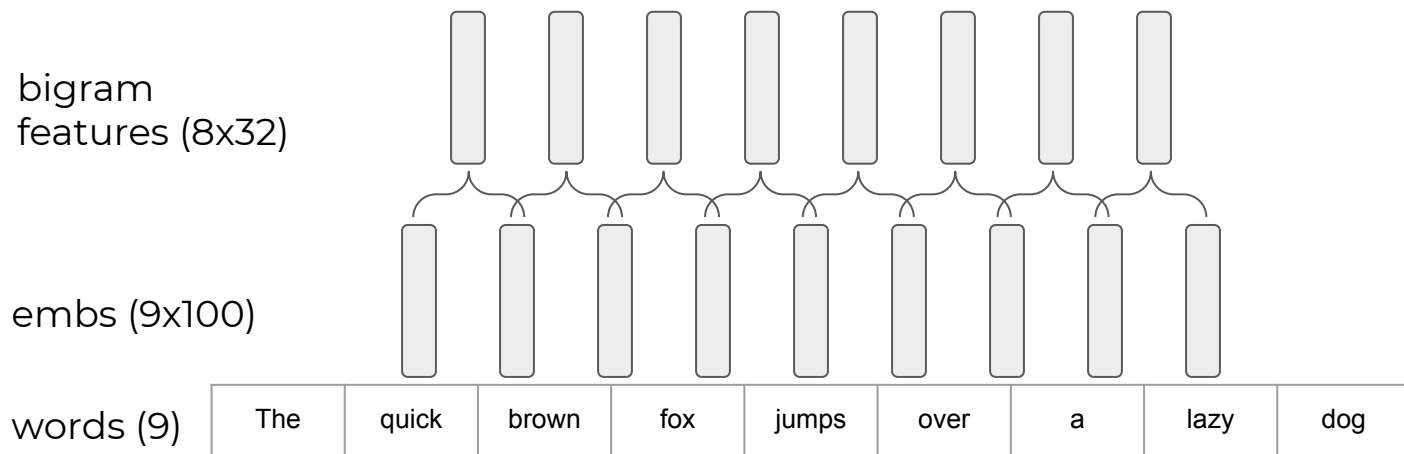
Слова

words (9)

| | | | | | | | | |
|-----|-------|-------|-----|-------|------|---|------|-----|
| The | quick | brown | fox | jumps | over | a | lazy | dog |
|-----|-------|-------|-----|-------|------|---|------|-----|

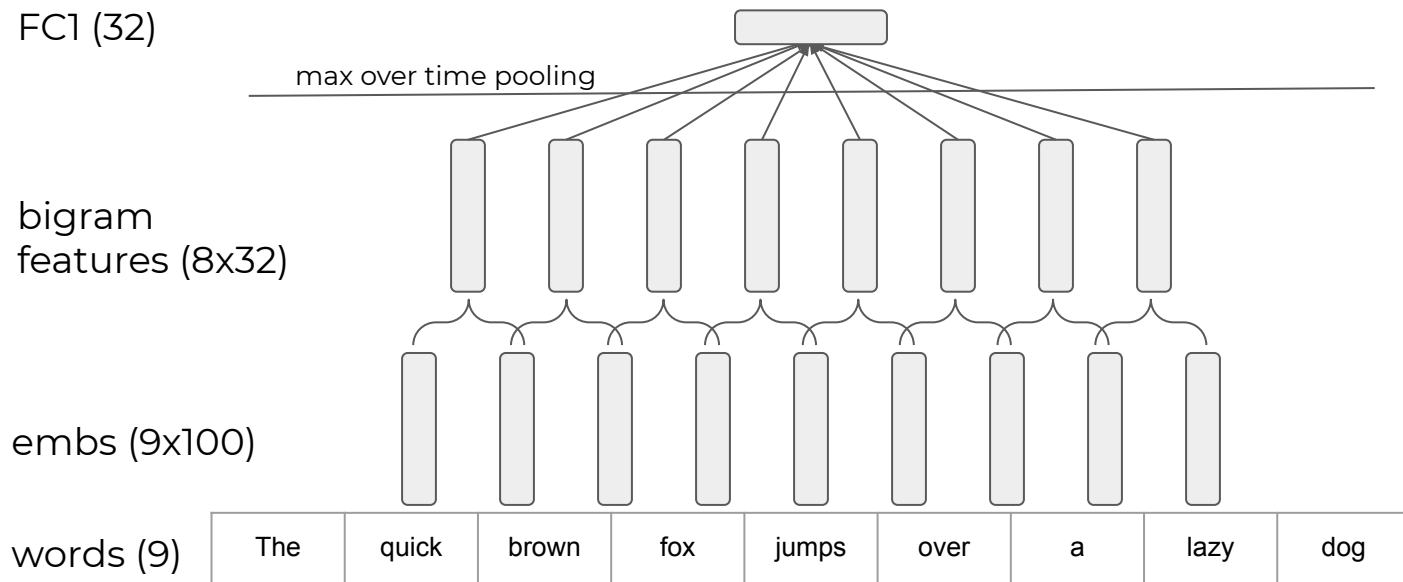
CNN для классификации текстов

Проводим свёртку по оси времени ($v = A_1x_1 + A_2x_2$)



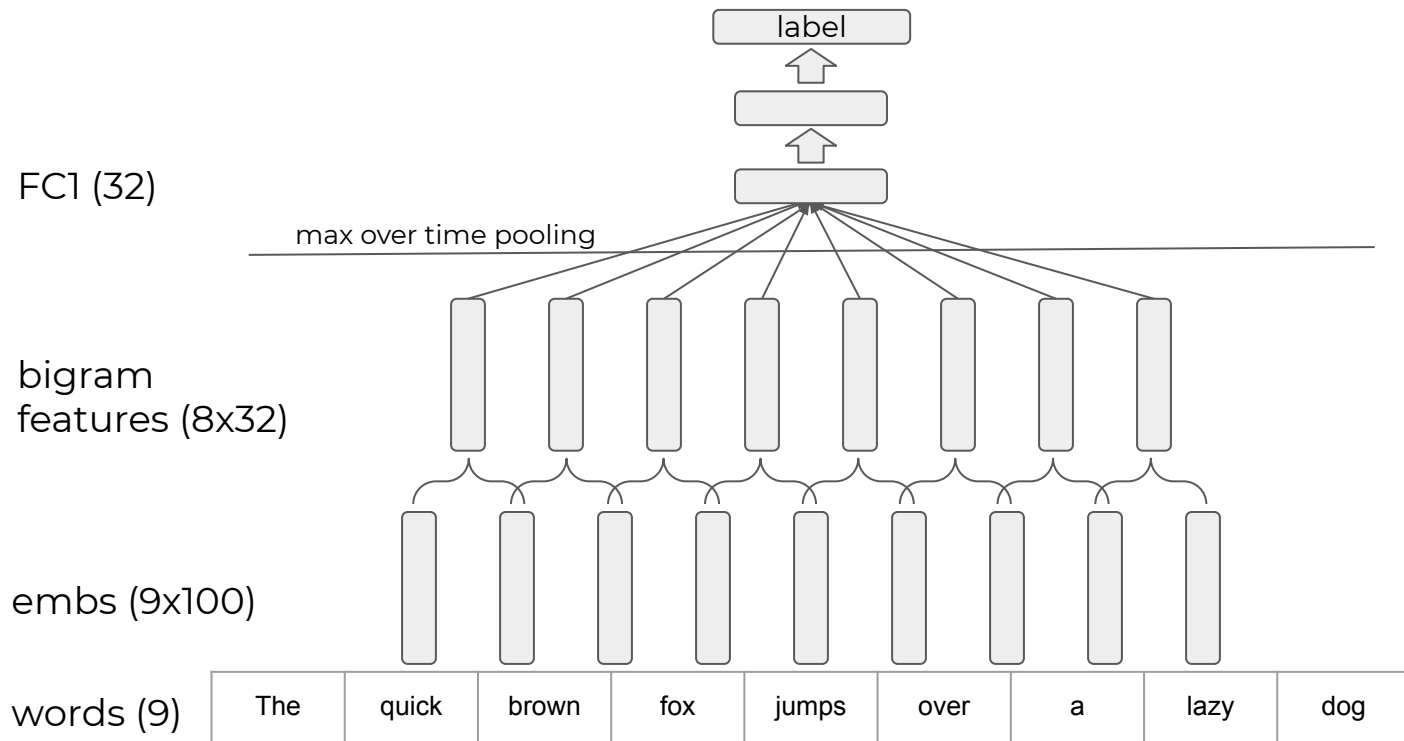
CNN для классификации текстов

“Max over time pooling”

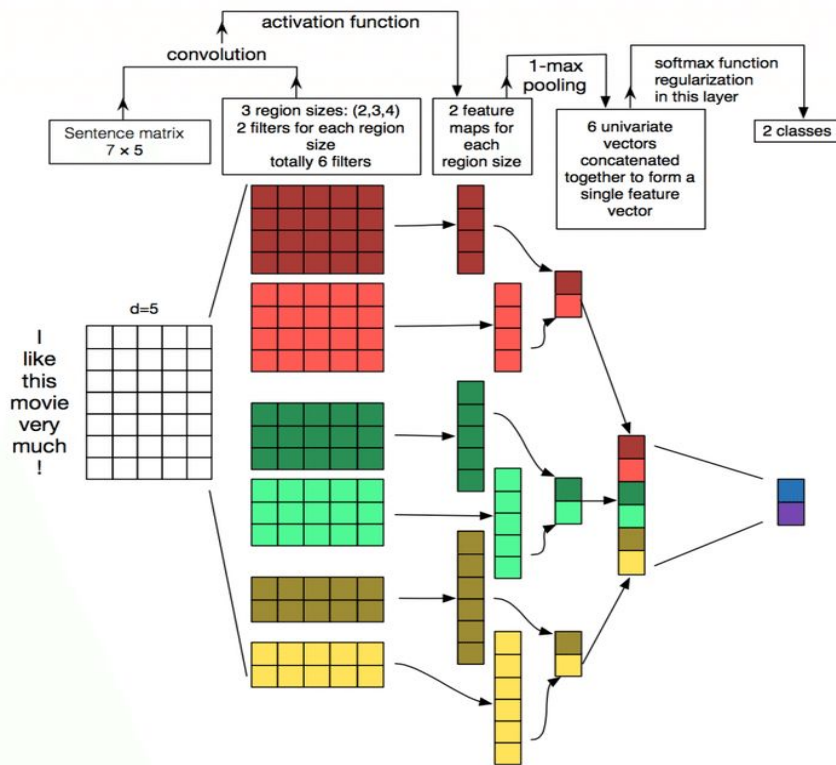


CNN для классификации текстов

Fully-connected NN



Улучшение CNN для классификации текстов



Классификация текстов с помощью RNN

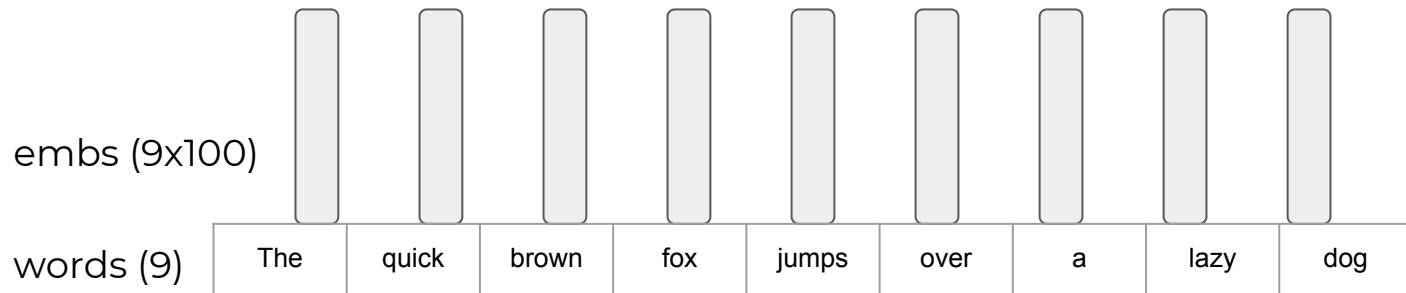


Недостаток предыдущих подходов

Мы по-прежнему не можем учитывать информацию с неограниченно длинного предложения

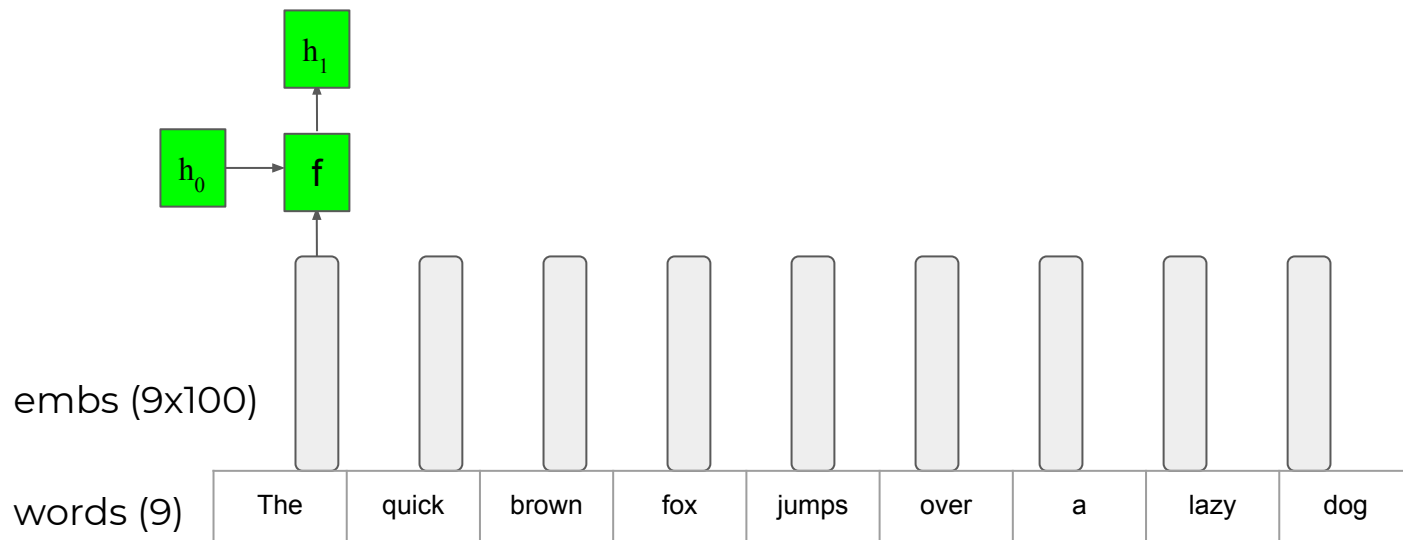
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$



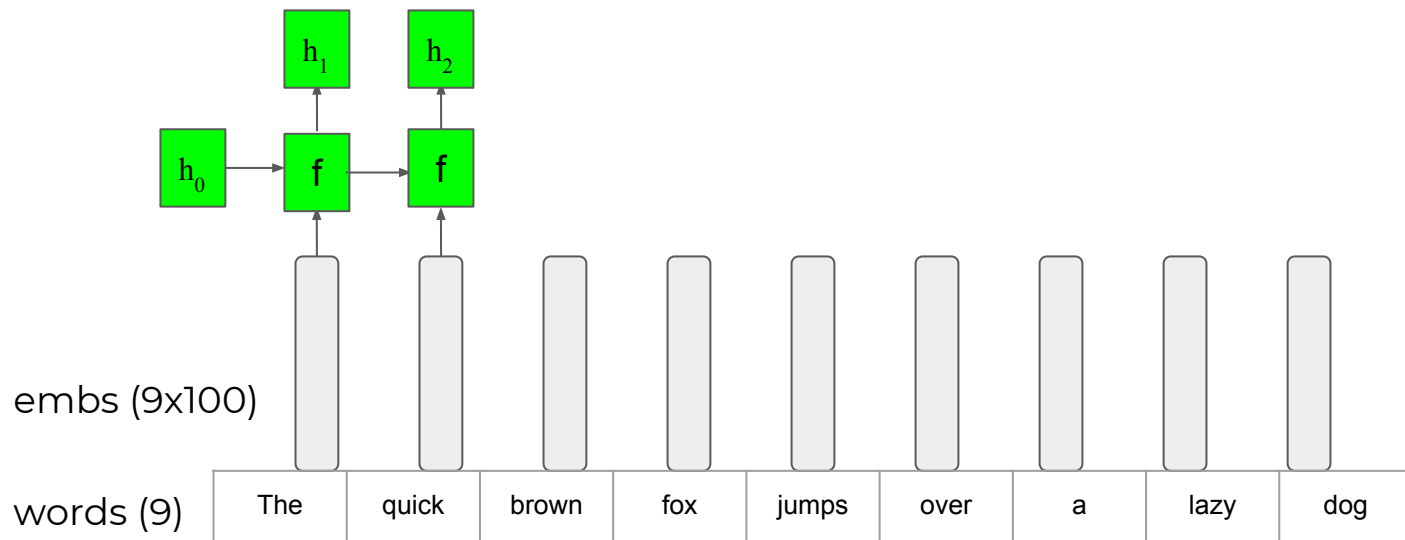
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



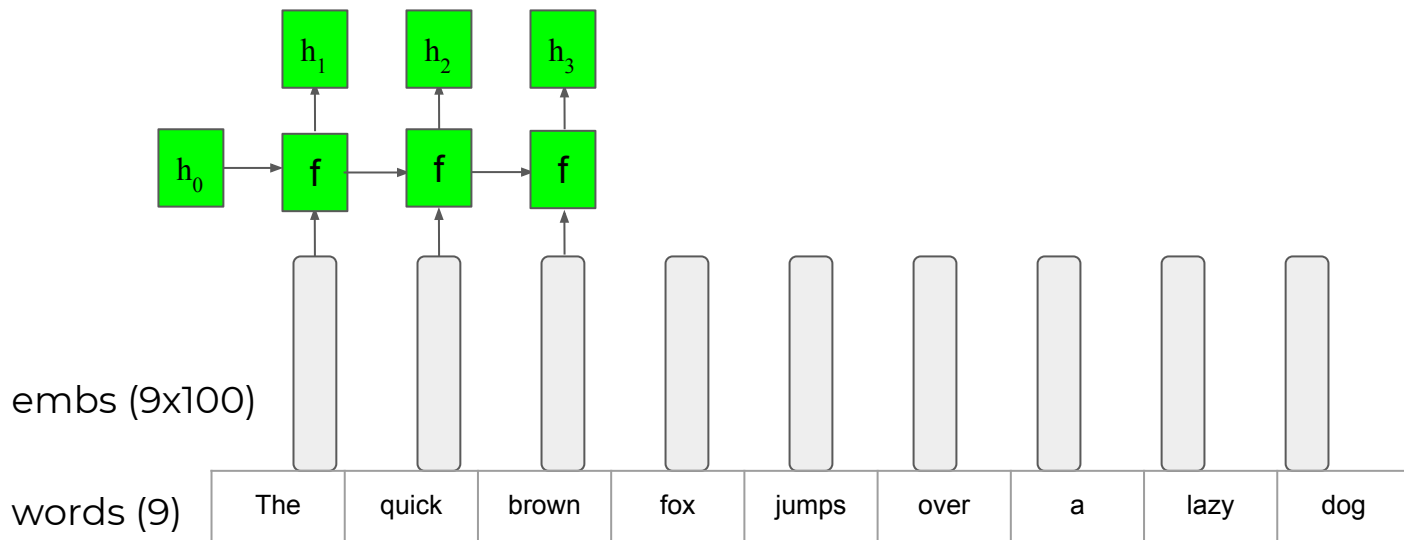
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



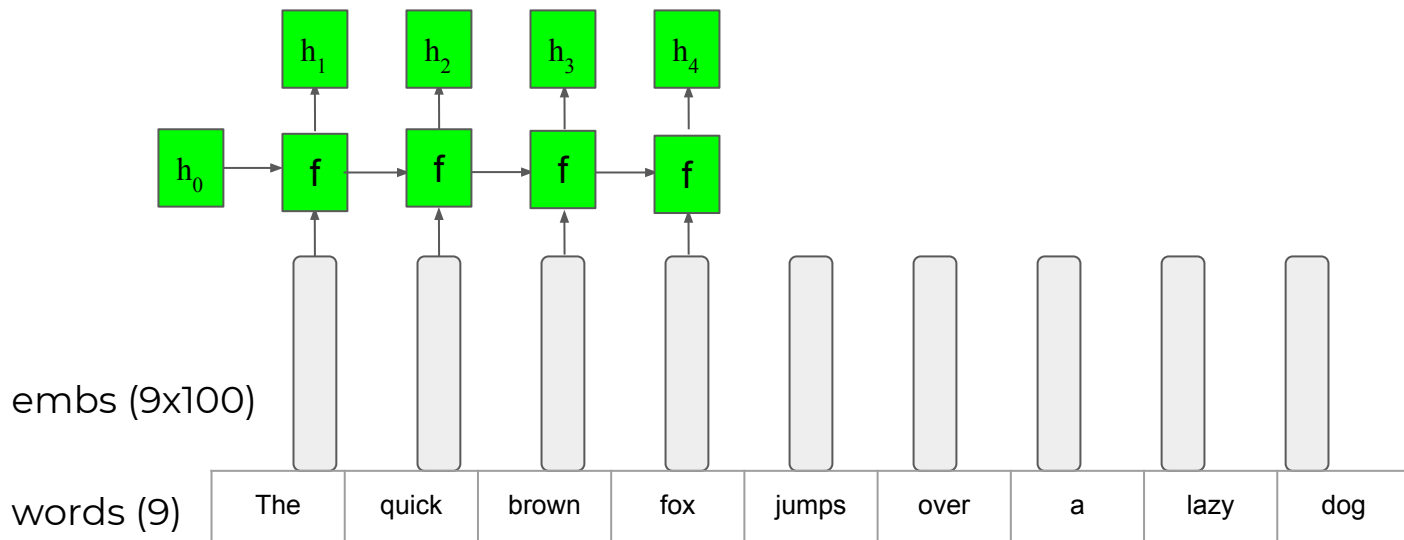
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



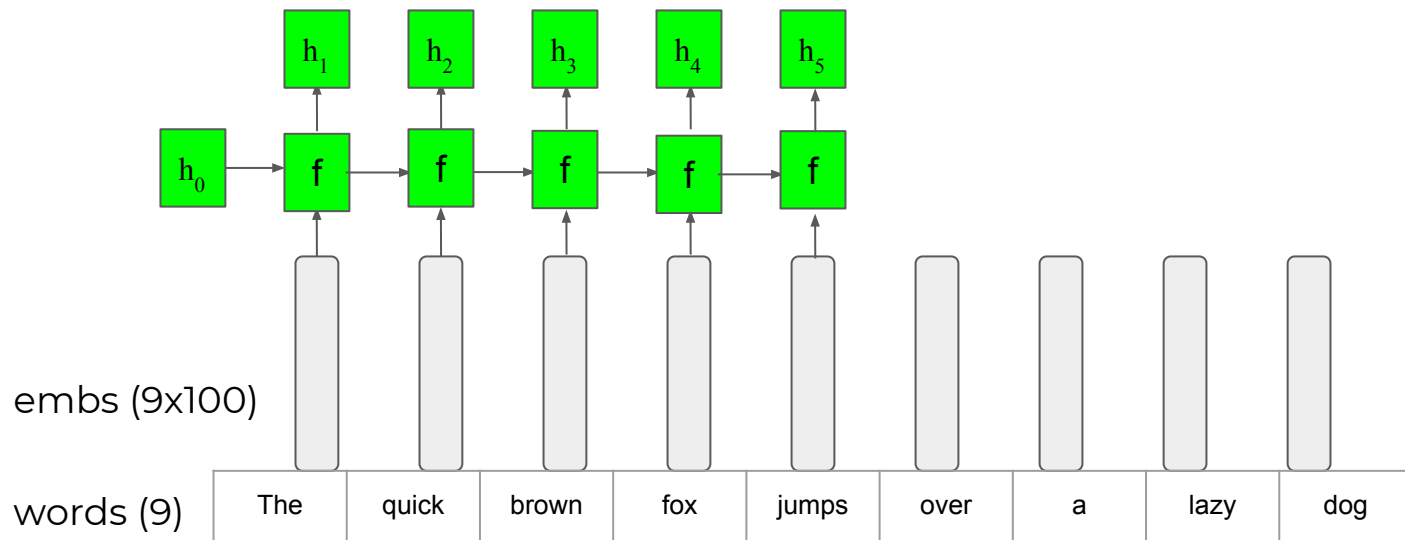
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



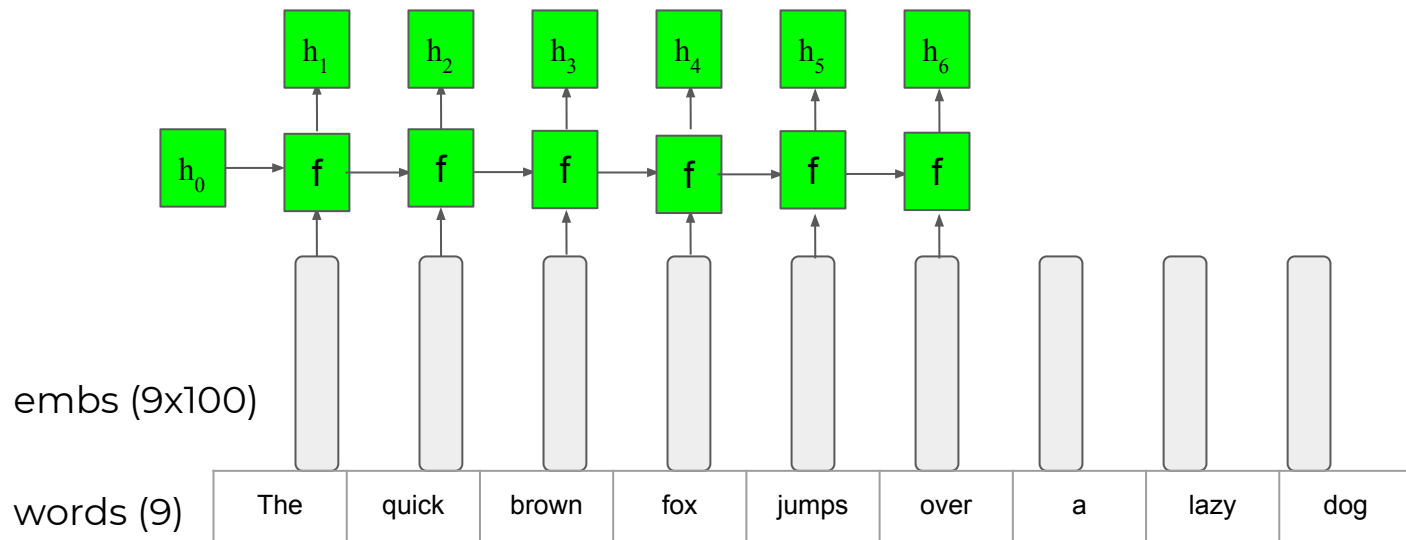
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



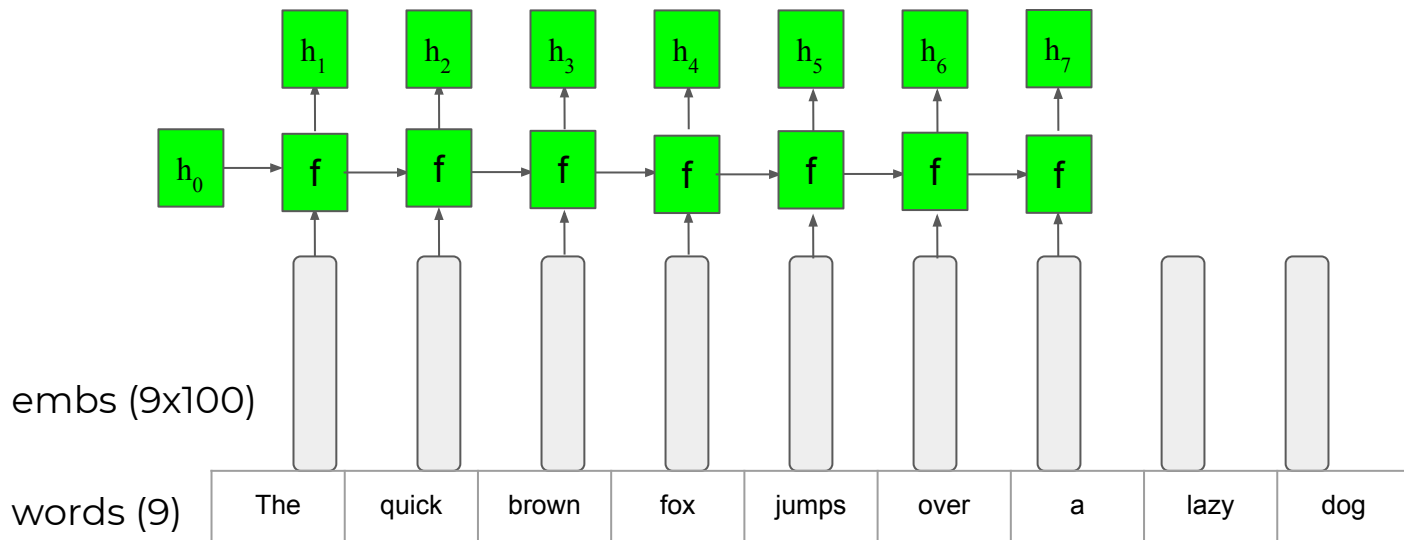
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



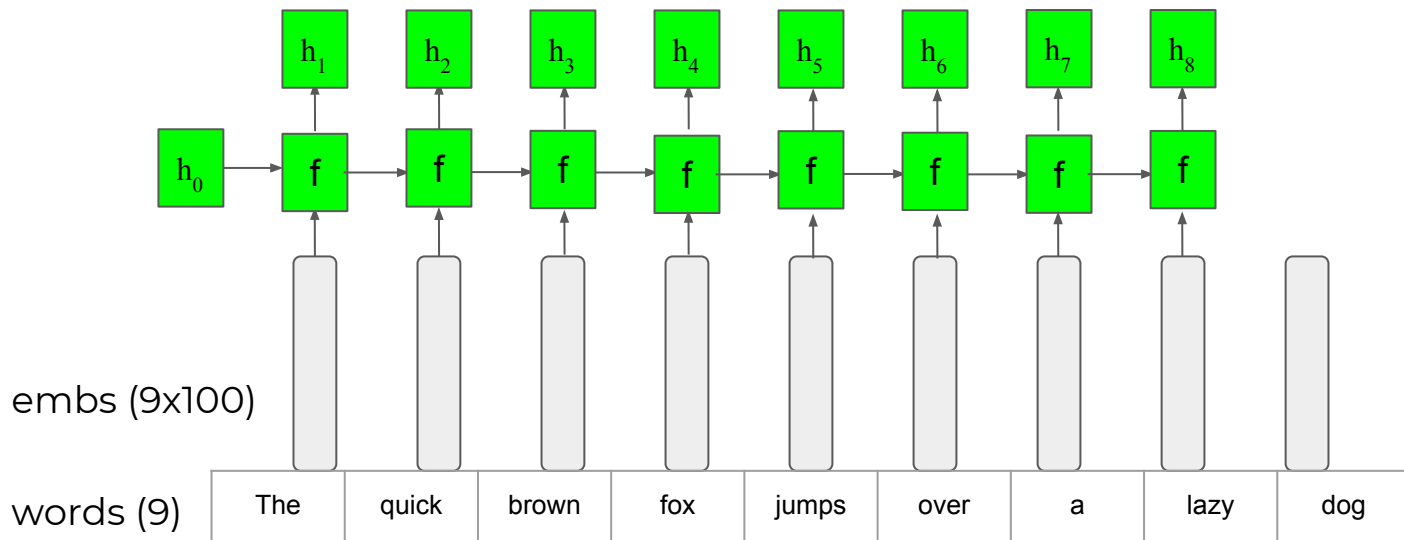
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



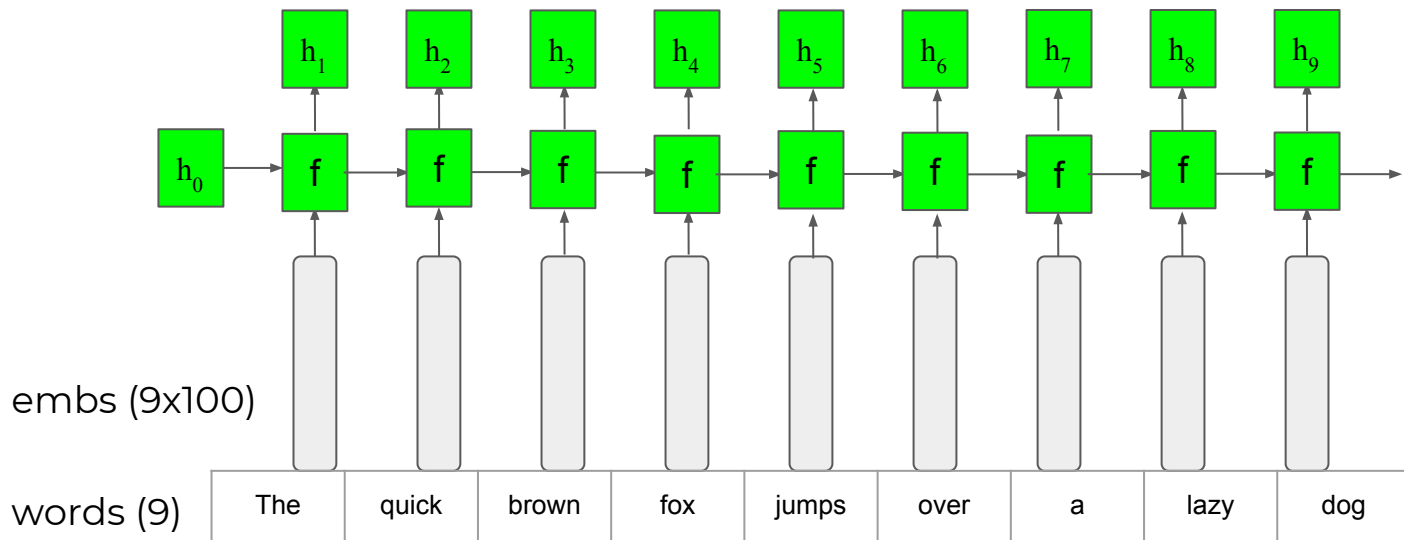
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}



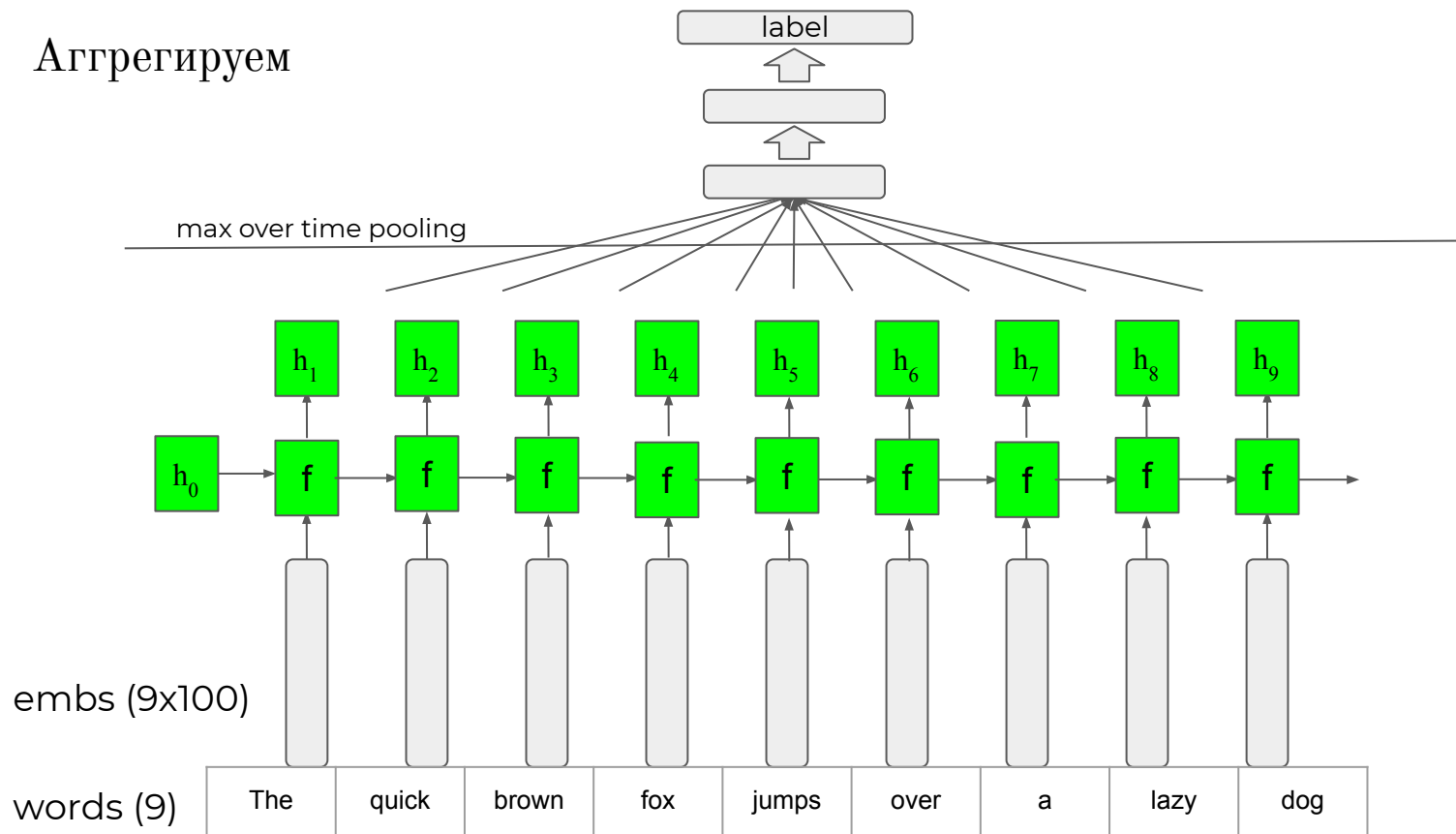
CNN для классификации текстов

- Заводим функцию $f_w(h_{t-1}, x_t)$
- Применяем её рекурсивно к скрытым состояниям h_{t-1}
- Получили h_1, h_{10}



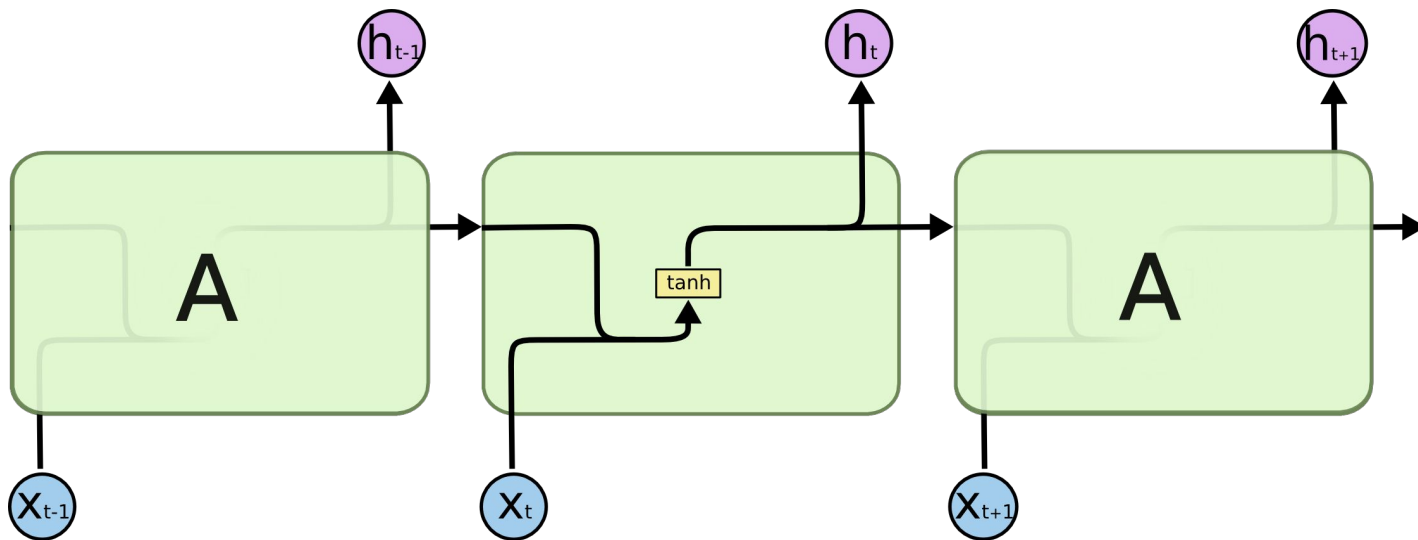
CNN для классификации текстов

- Агрегируем



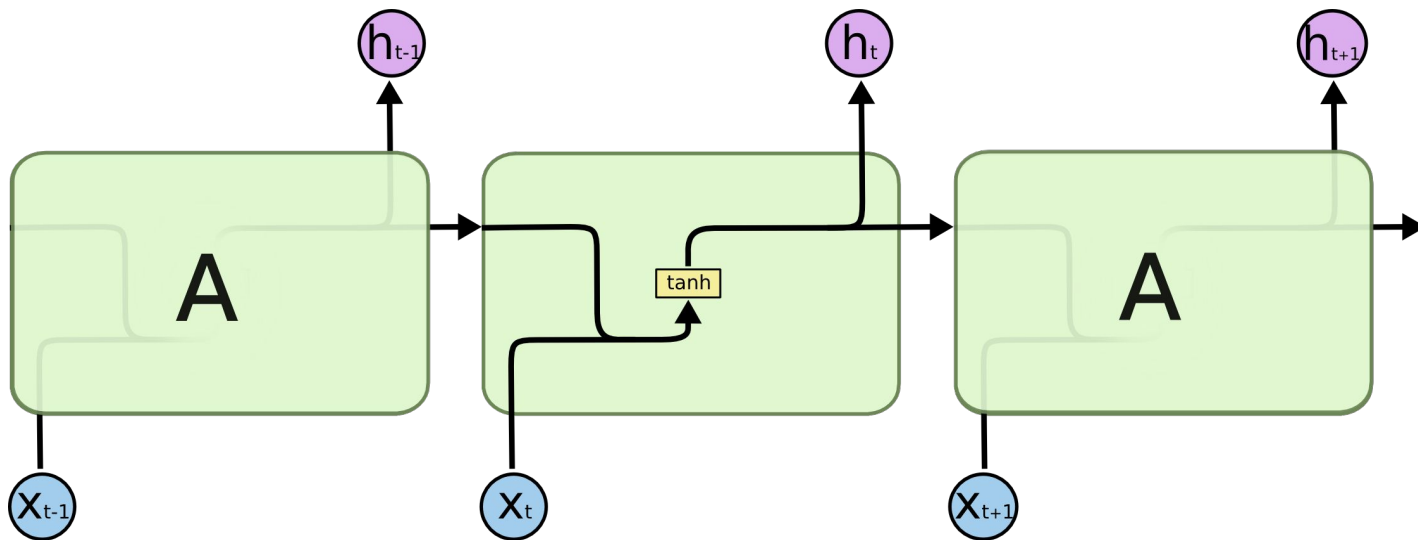
Vanilla RNN

- $f_w(h, x) = \tanh(A_h h + A_x x)$

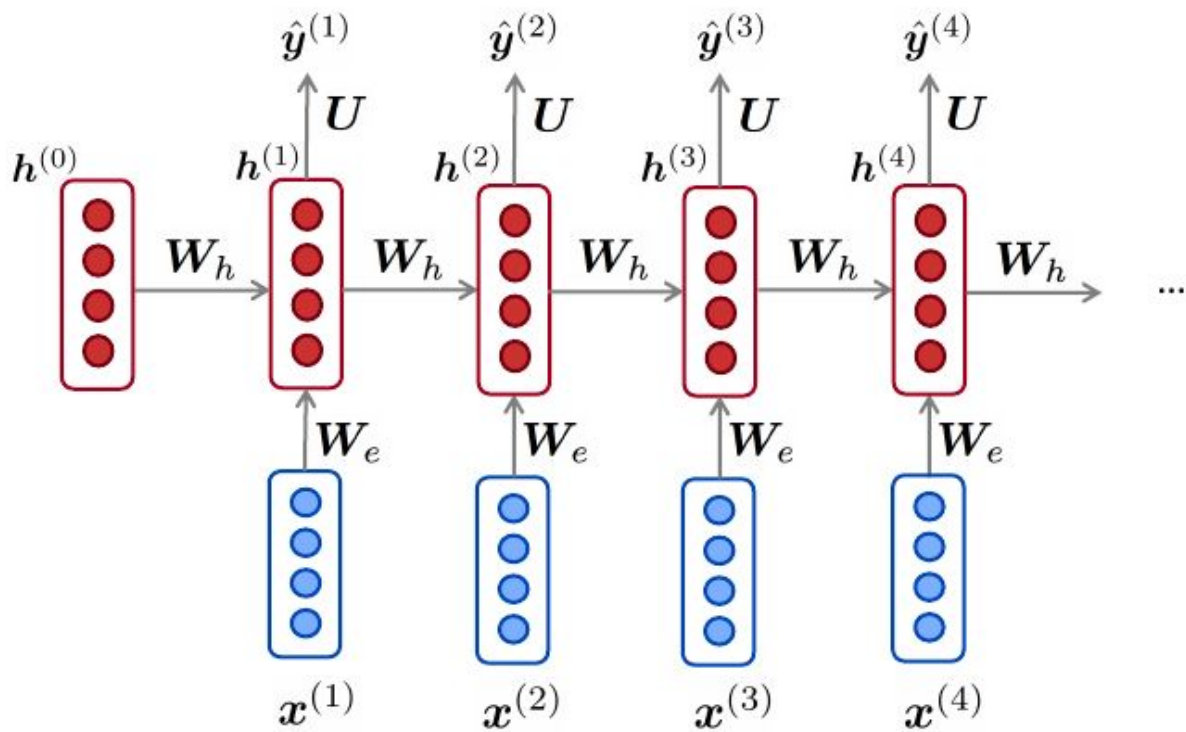


Обучение RNN

Алгоритм “Back Propagation Through Time”



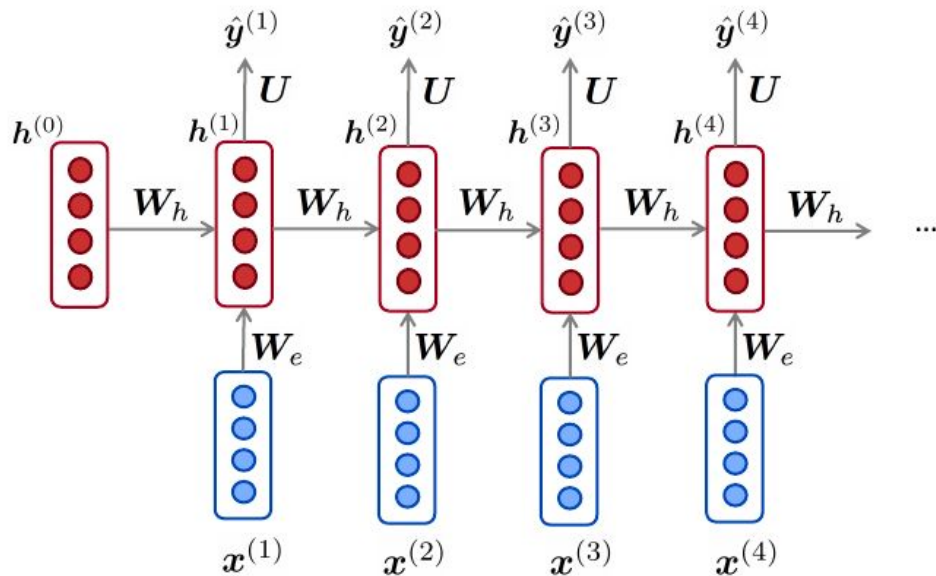
Обучение RNN



Обучение RNN

$$h_t = W f(h_{t-1}) + W^{(hx)} x_{[t]}$$

$$\hat{y}_t = W^{(S)} f(h_t)$$



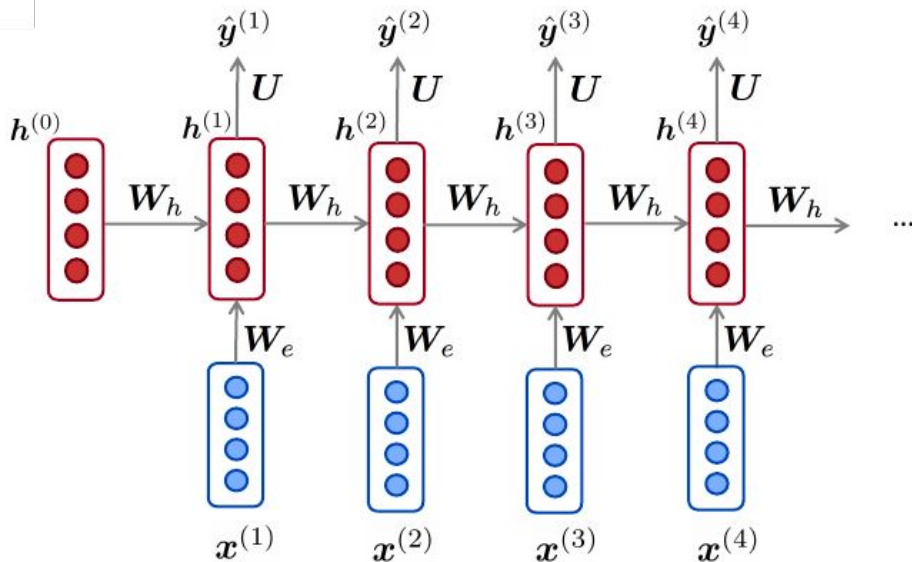
Вычисление градиента функции потерь

$$h_t = W f(h_{t-1}) + W^{(hx)} x_{[t]}$$

$$\hat{y}_t = W^{(S)} f(h_t)$$

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W} \quad \text{E = error}$$

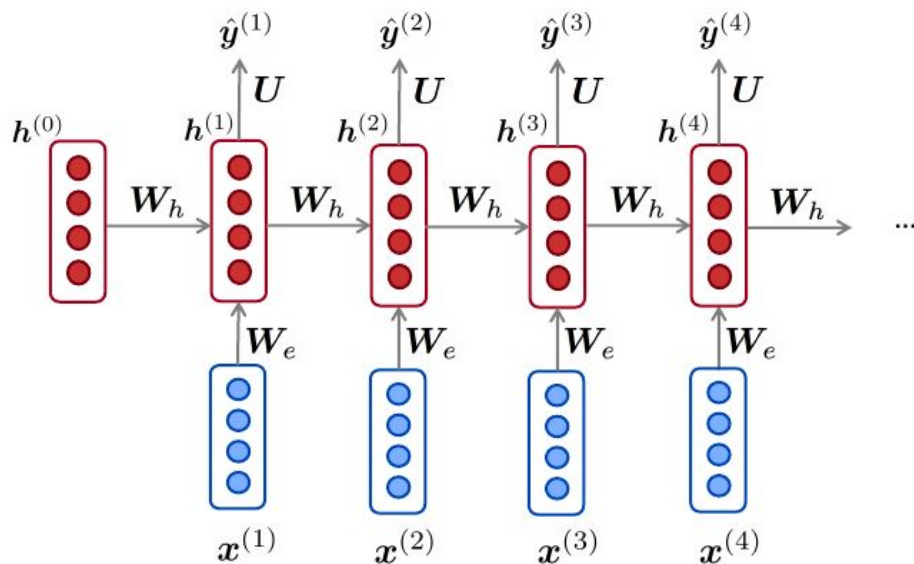
$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$



Вычисление градиента функции потерь

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^t \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^t \frac{\partial h_j}{\partial h_{j-1}}$$



Преимущества и недостатки RNN

- Преимущества

- обрабатывает вход произвольной длины
- (теоретически) может использовать информацию с любого момента времени
- можно обрабатывать все префиксы

- Недостатки

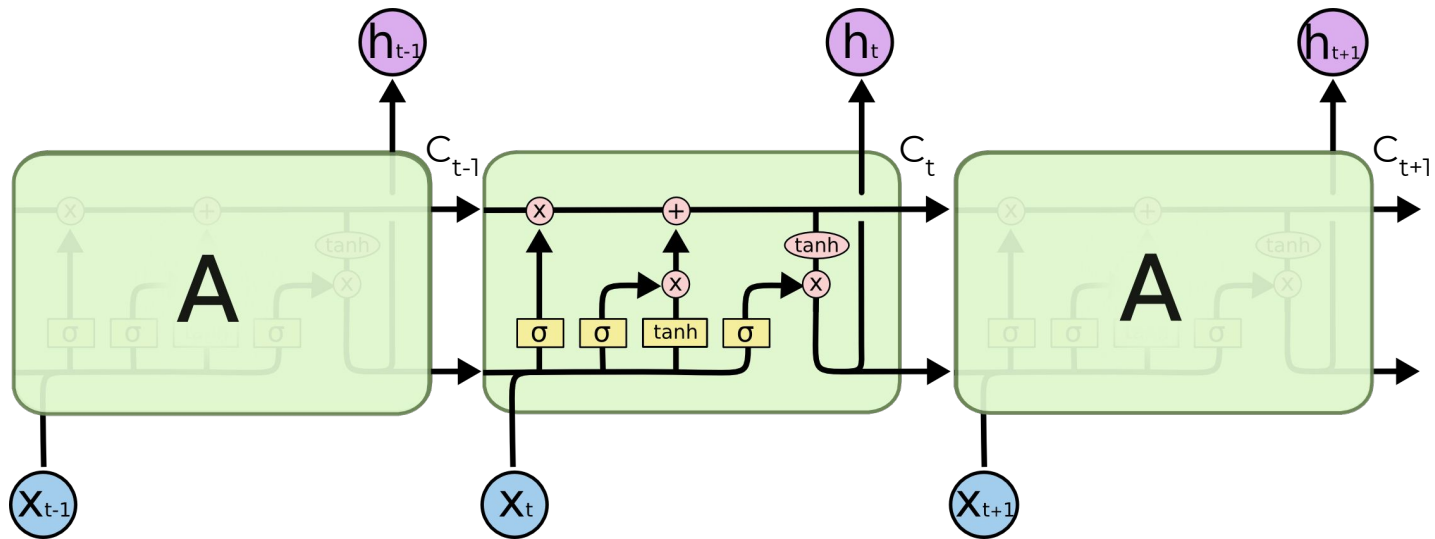
- проблема взрывающегося градиента
- вся информация закодирована одним вектором (достаточно ли?)

Архитектуры RNN



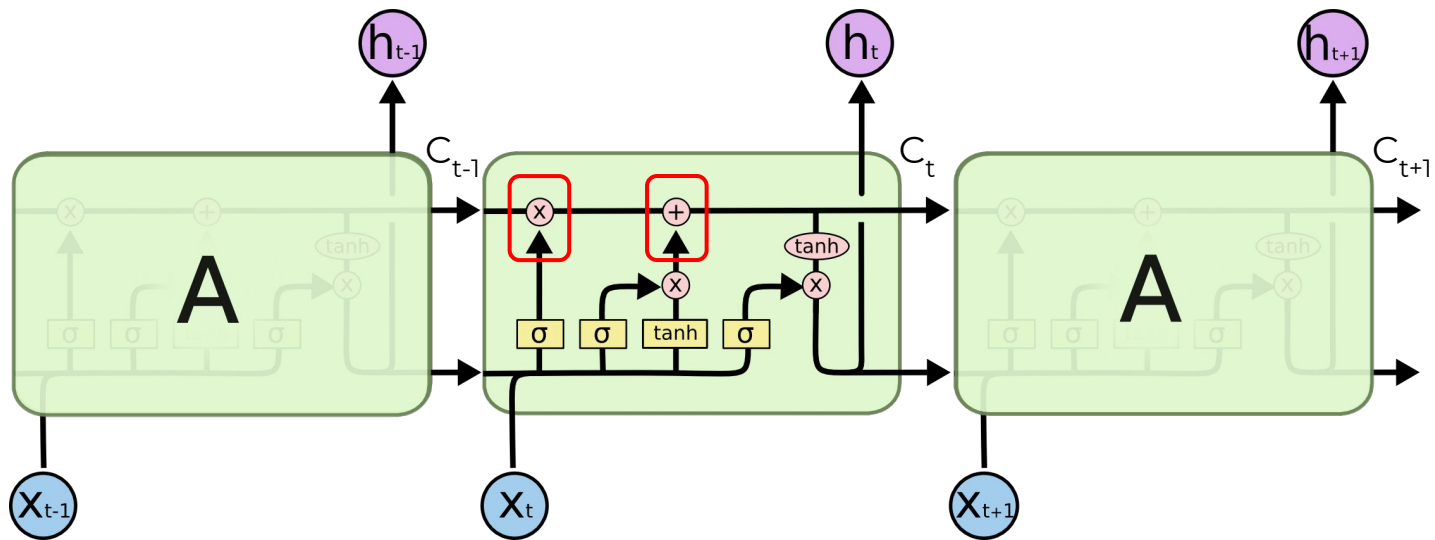
Long Short Term Memory Cell (LSTM)

- Дополнительный путь течения информации (состояние сети, C_t)



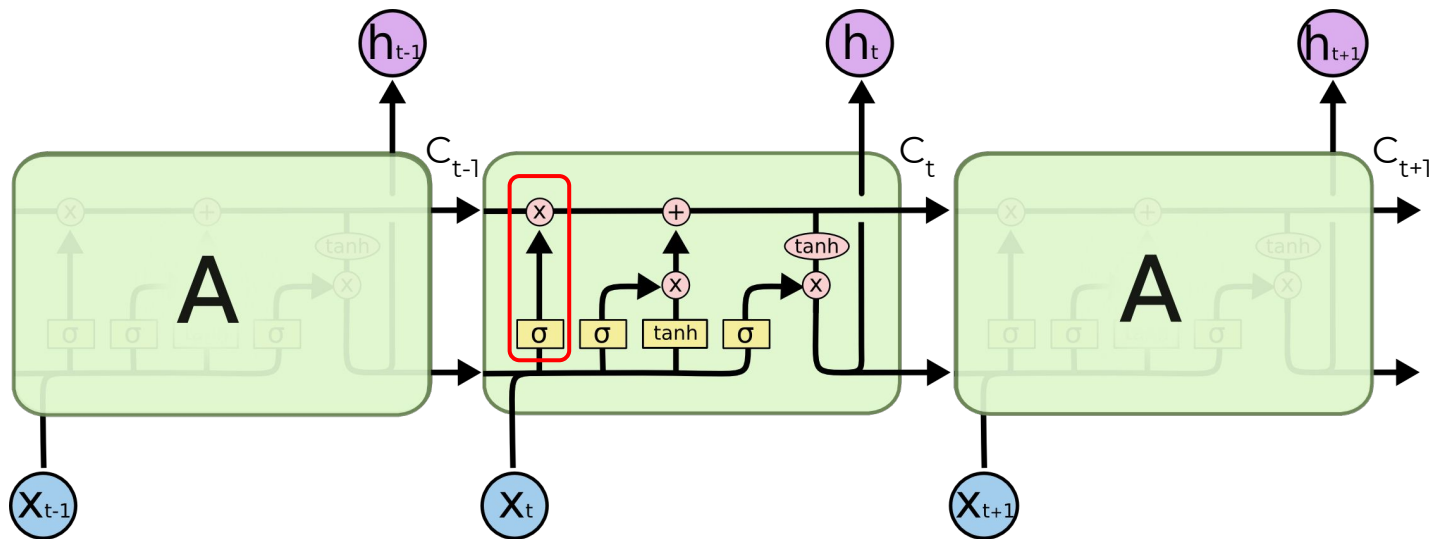
Long Short Term Memory Cell (LSTM)

- Состояние сети обновляется в каждой клетке (очень аккуратно, с возможностью не обновляться вообще)



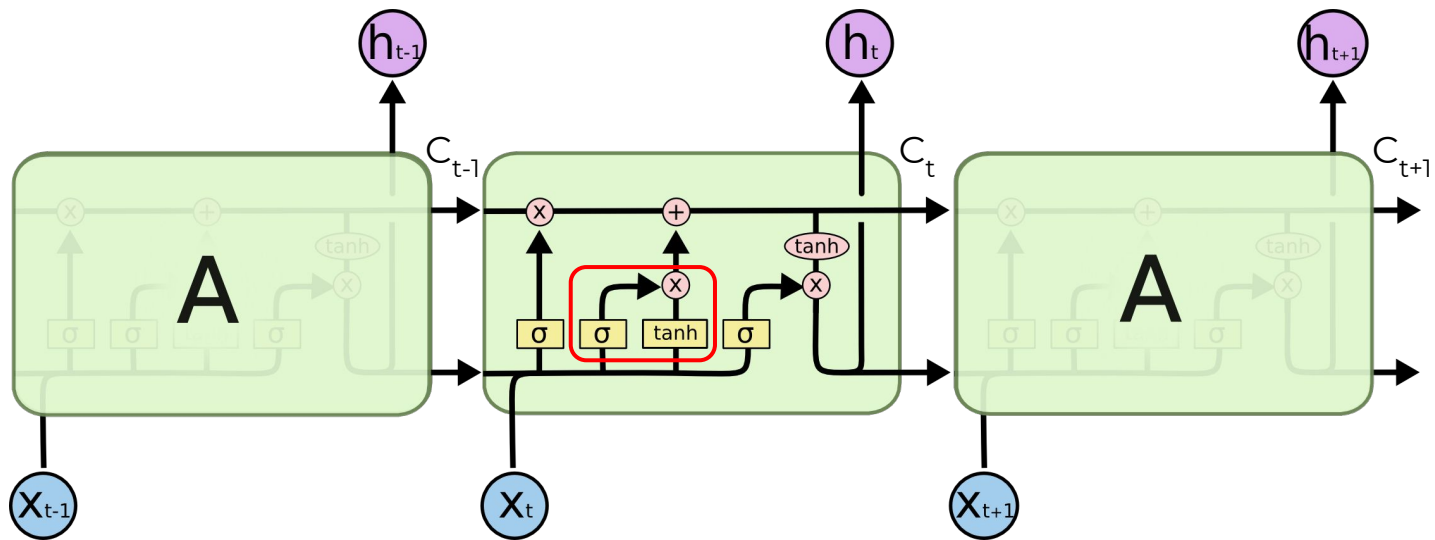
Long Short Term Memory Cell (LSTM)

- Forget gate: часть информации забывается (домножение на результаты сигмоиды)



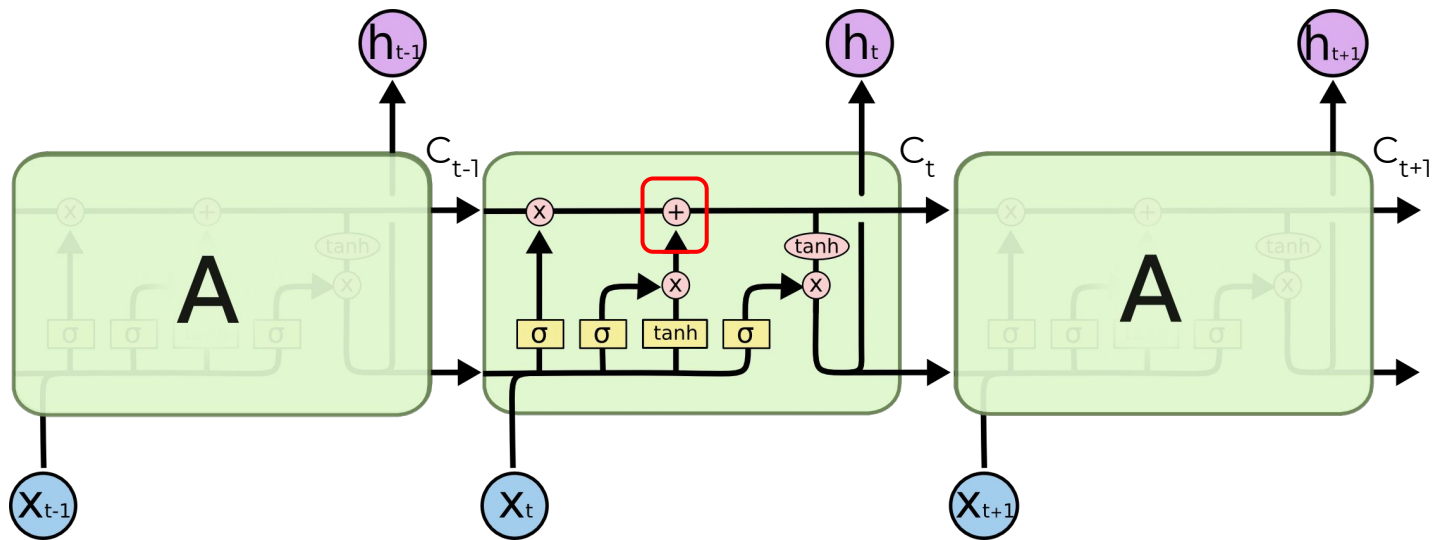
Long Short Term Memory Cell (LSTM)

- Input gate: новая информация формируется



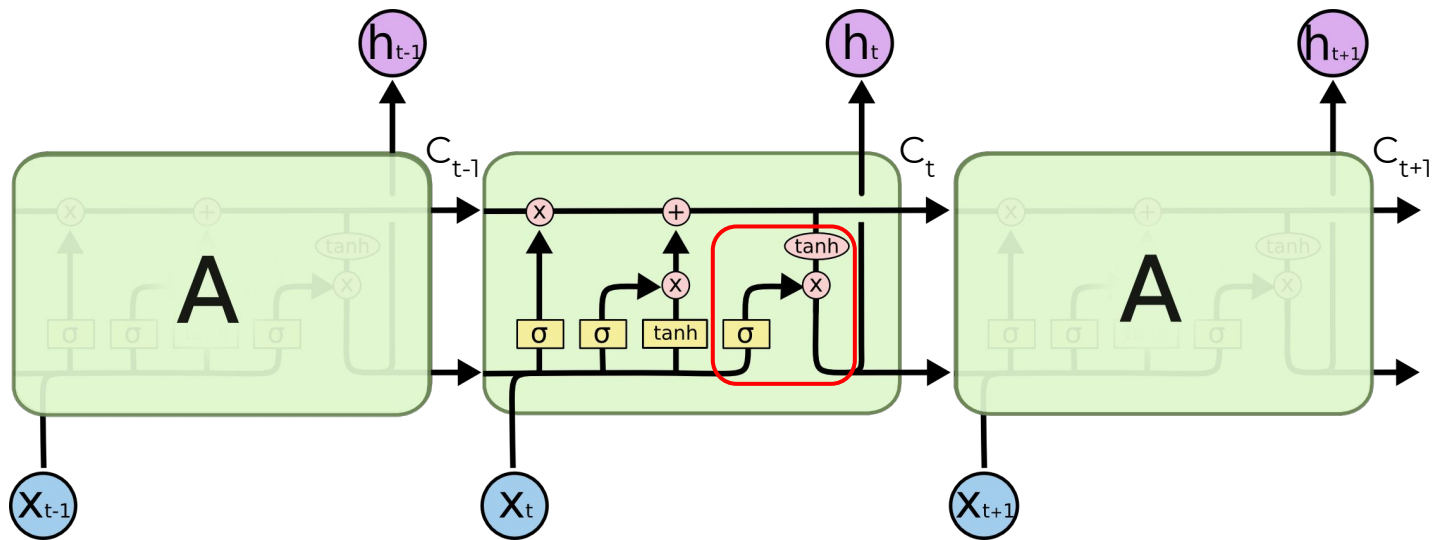
Long Short Term Memory Cell (LSTM)

- Input gate: новая информация прибавляется к состоянию сети

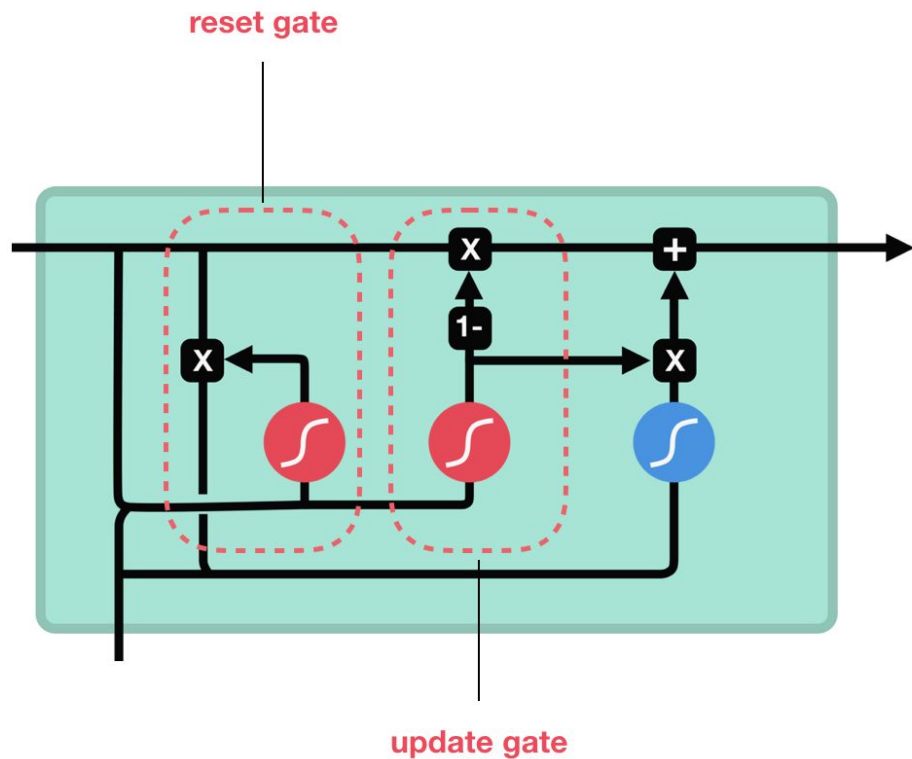


Long Short Term Memory Cell (LSTM)

- Output: новое состояние сети скрещивается с h_{t-1} и формирует h_t



Gated Recurrent Units (GRU)



The End

