кафедра Алгоритмов
и Технологий
Программирования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Автоматическая обработка текстов
Нейронные сети

Лекция 4

Емельянов А. А.
login-const@mail.ru

2018

https://github.com/king-menin/nlp-hse-autumn2018

Биологическая модель нейрона

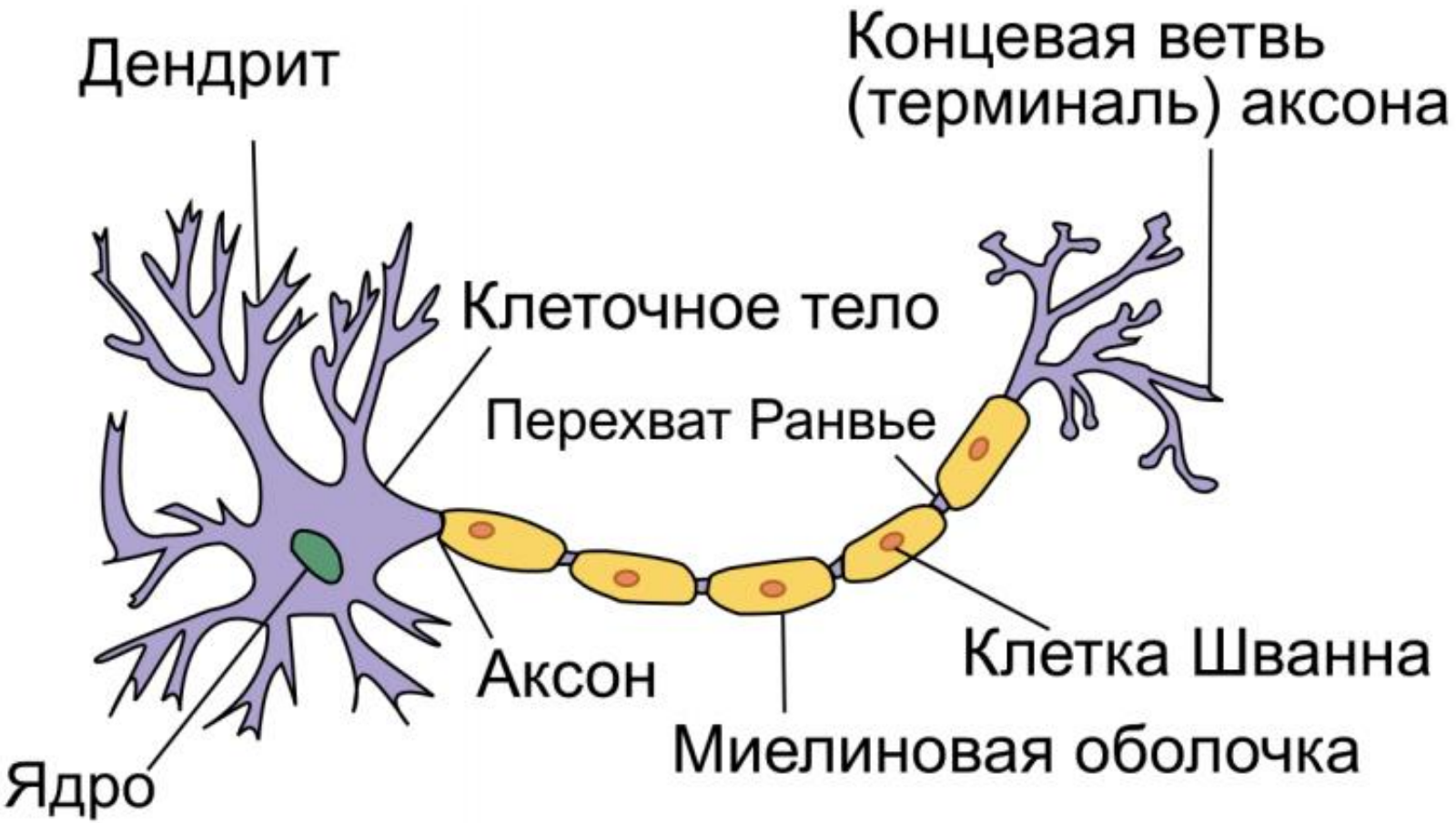


Рис.: Структура нейрона

Биологическая модель нейрона

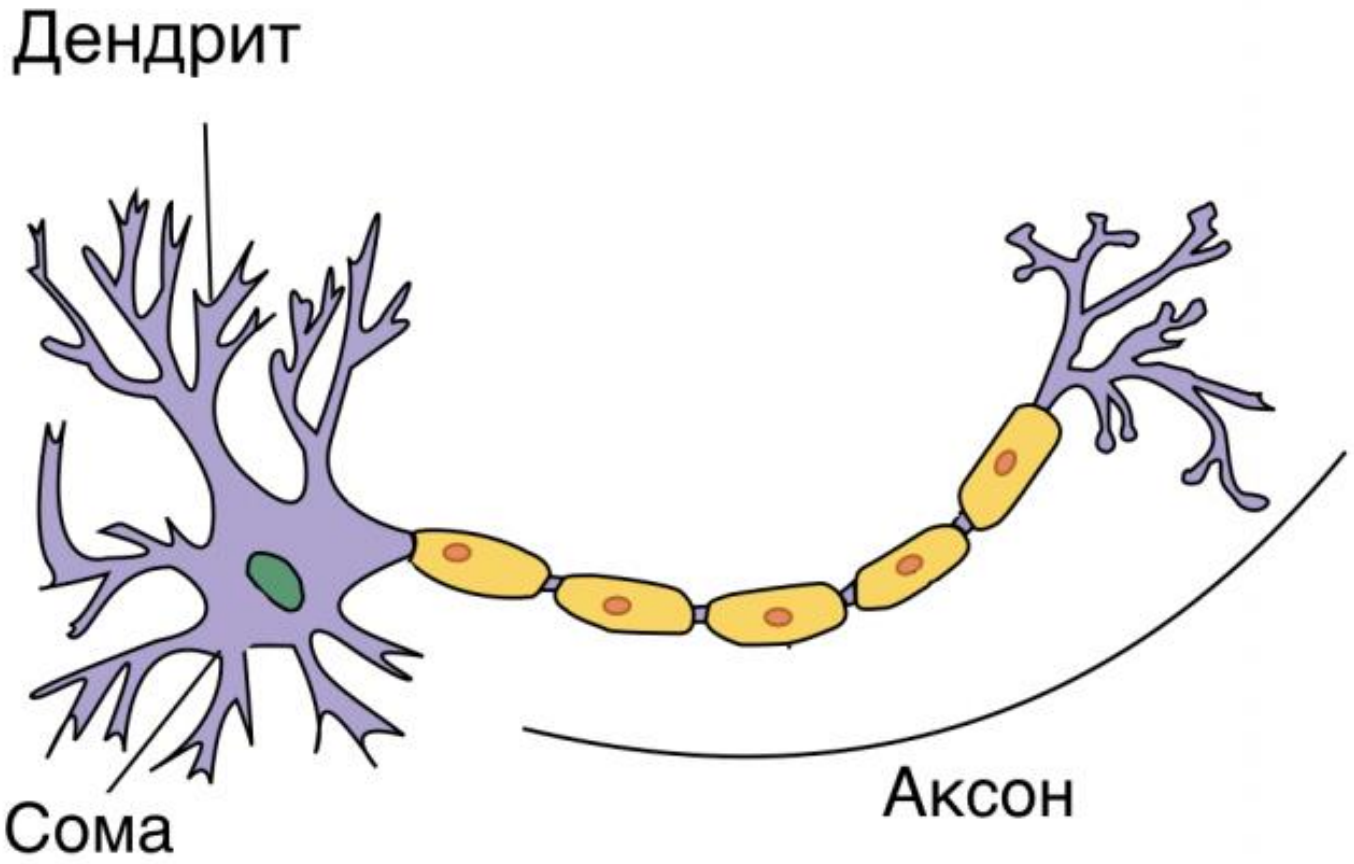


Рис.: Структура нейрона

Биологическая модель нейрона

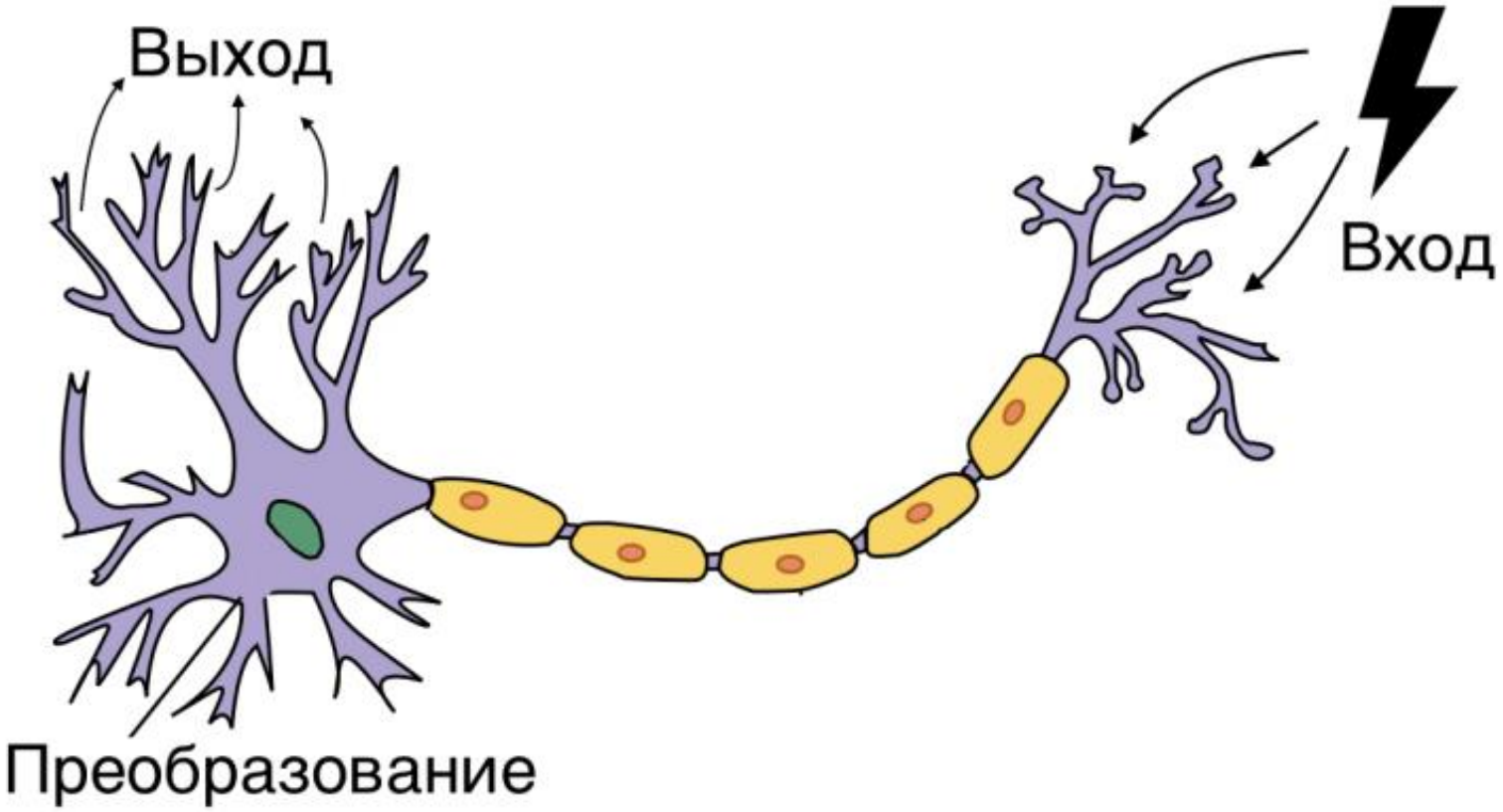


Рис.: Структура нейрона

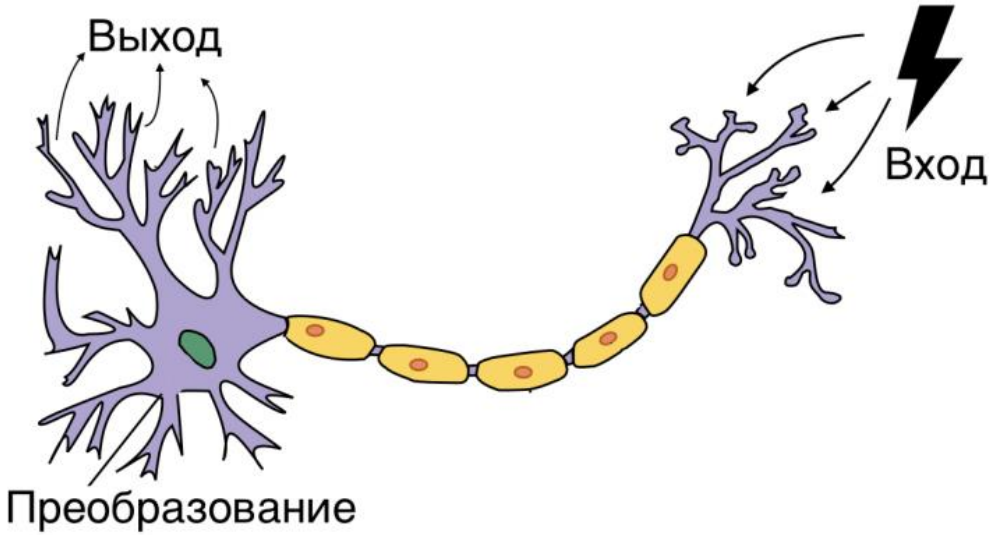
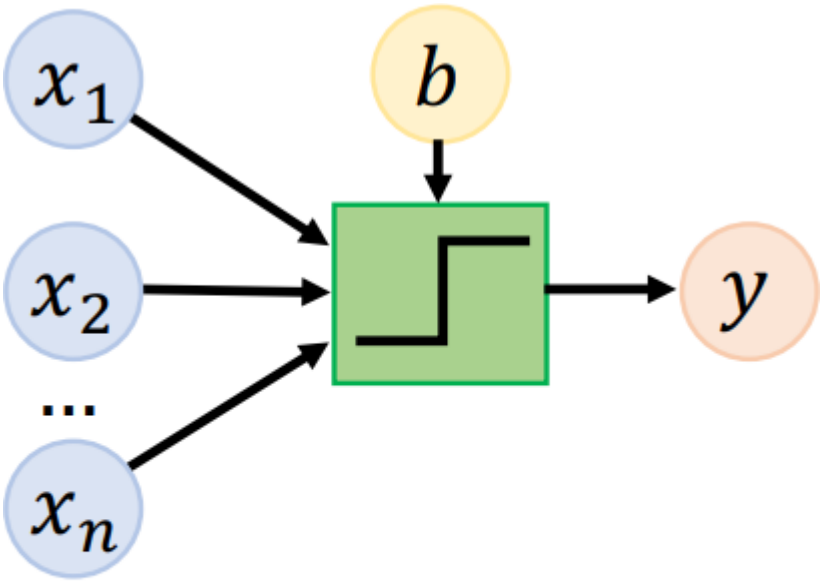


Рис.: Структура нейрона

- Выходной сигнал посылается при достижении определенного уровня входного сигнала. Модель:

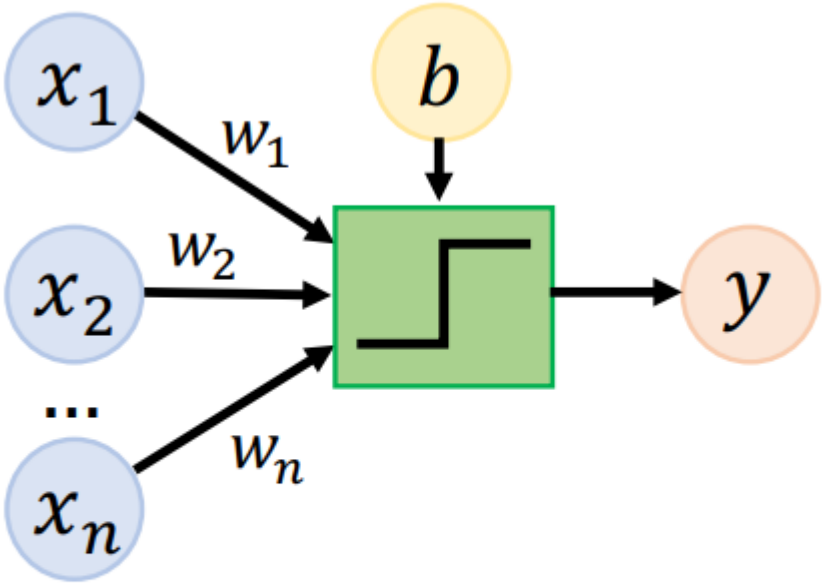
$$y = \begin{cases} 1, & \sum_{i=1}^{N} x_i > b \\ 0, & \text{иначе} \end{cases} = \mathbb{I}[\sum_{i=1}^{N} x_i > b]$$

Модель 1: схема



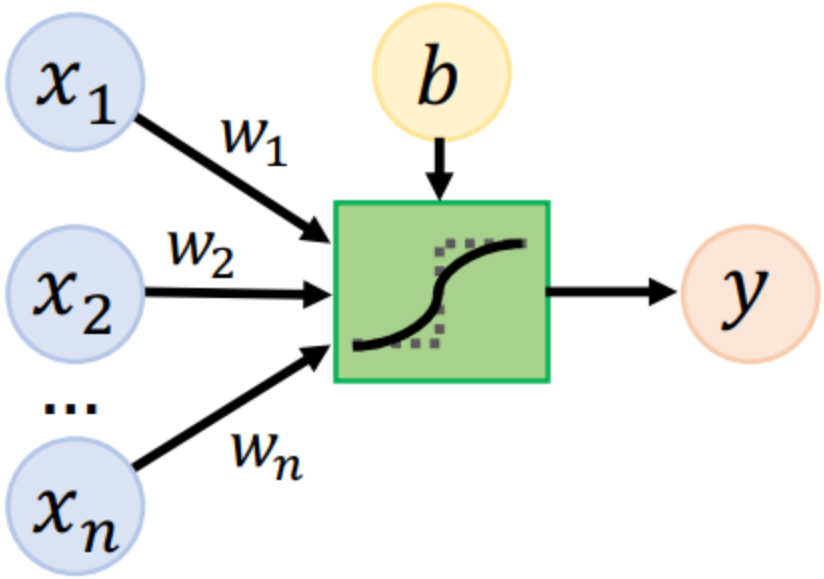
$$y = \mathbb{I}[\sum_{i=1}^{N} x_i > b]$$

Модель 2: Чувствительность нейронов



$$y = \mathbb{I}[\sum_{i=1}^{N} w_i x_i > b] = \mathbb{I}[w^T x > b]$$

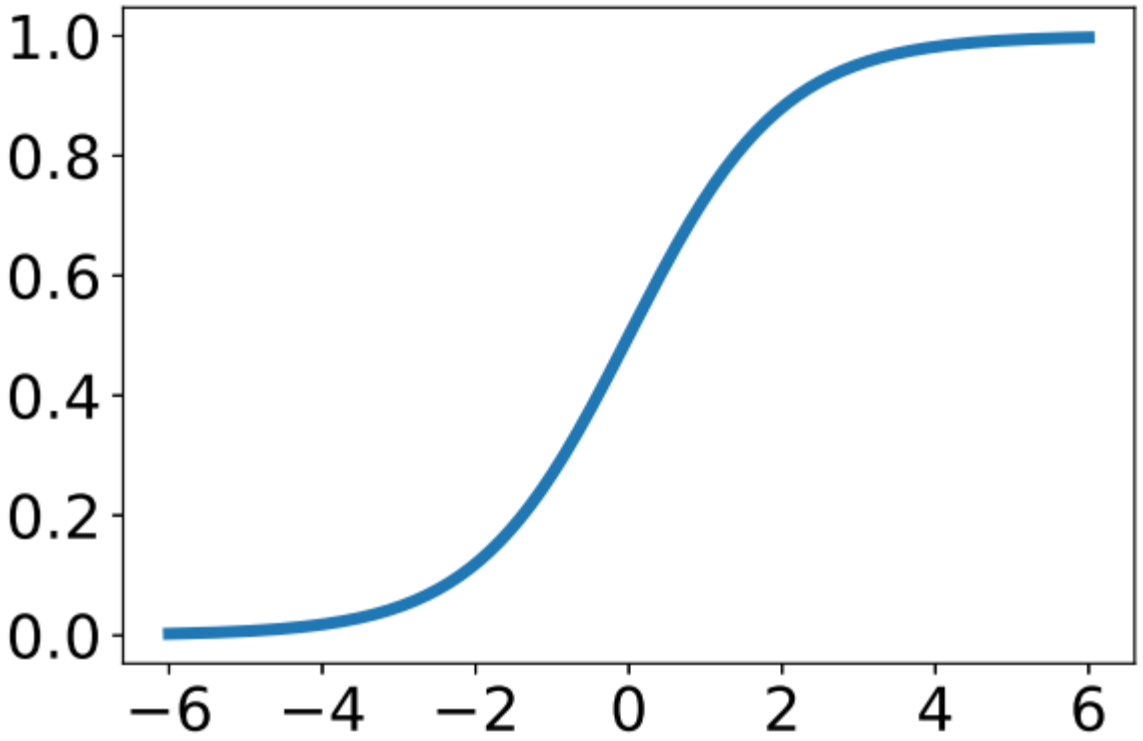
Модель 3: Непрерывная активация



$$y = \sigma[\sum_{i=1}^{N} w_i x_i - b] = \sigma[w^T x - b]$$

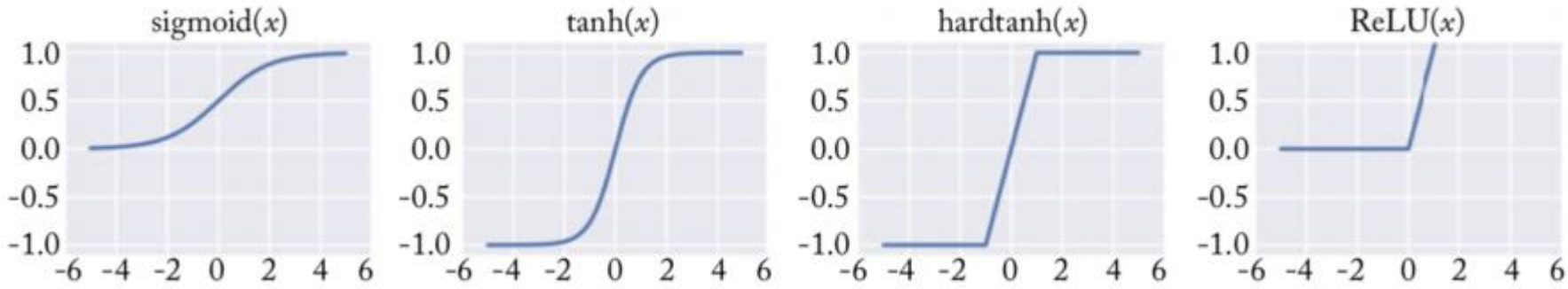
- Параметры нейрона: w — веса, b — смещение.

Функция активации

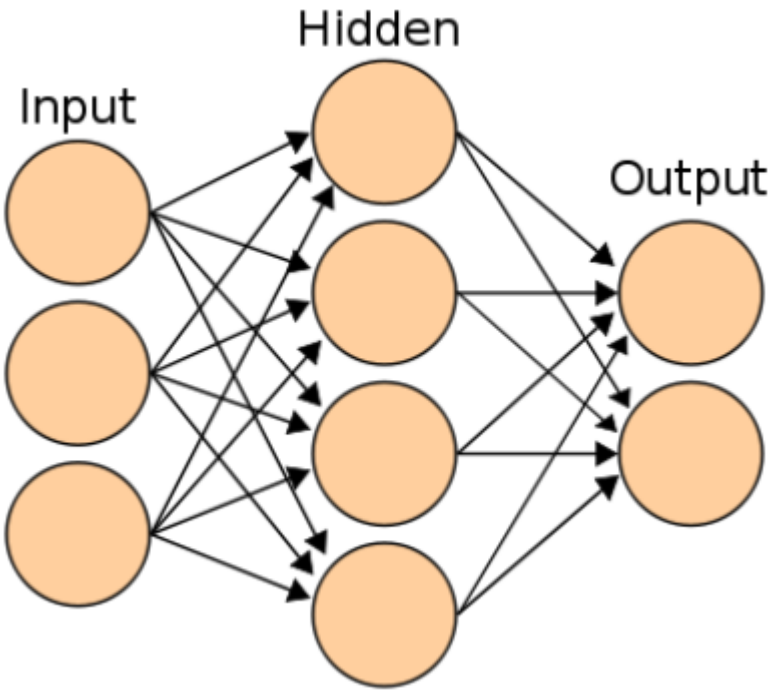


$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Нелинейные функции активации



Сети с одним скрытым слоем



- **Теорема (универсальный аппроксиматор)** Любую непрерывную на компакте функцию можно равномерно приблизить нейронной сетью с одним скрытым слоем.

1. Отличная визуализация: http://neuralnetworksanddeeplearning.com/chap4.html

Как обучить нейронную сеть?

- Обучить нейронную сеть — подобрать значения всех настраиваемых параметров (веса и смещения). Два этапа:

1. Задать функцию потерь L

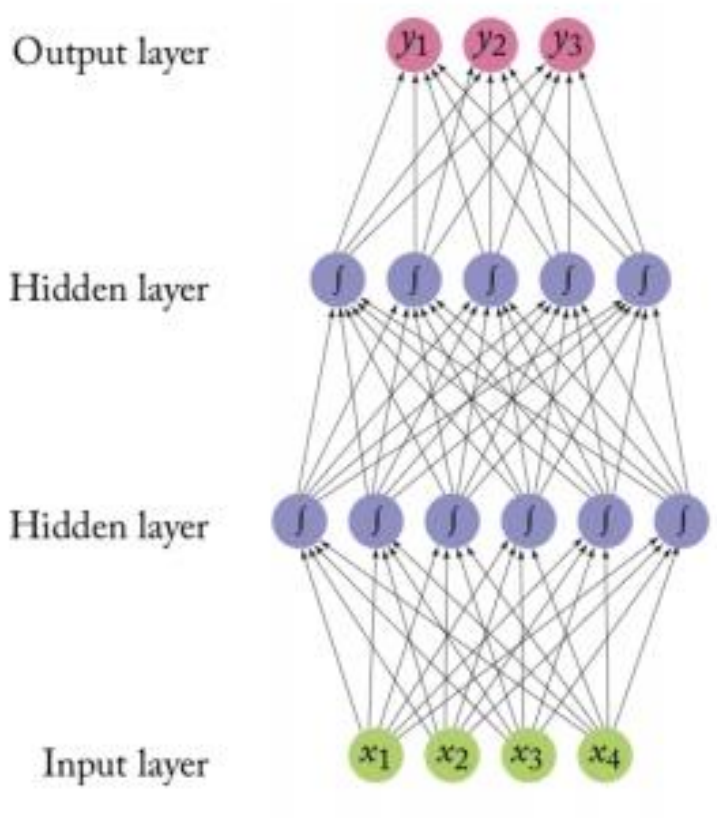
2. Подобрать веса, минимизирующие L.

Многослойные сети прямого распространения

$$\mathrm{NN}_{\mathrm{MLP2}}(x) = y$$

$$h_1 = g^1(xW^1 + b^1)$$
$$h_2 = g^2(h^1 W^2 + b^2)$$
$$y = h^2 W^3$$

$$x \in \mathbb{R}^{d_{in}}, y \in \mathbb{R}^{d_{out}}$$
$$W^1 \in \mathbb{R}^{d_{in} \times d_1}, b^1 \in \mathbb{R}^{d_1}$$
$$W^2 \in \mathbb{R}^{d_1 \times d_2}, b^2 \in \mathbb{R}^{d_2}$$
$$W^3 \in \mathbb{R}^{d_2 \times d_{out}}$$



Векторное представление текста

- Мешок слов [Bag of Words, BoW]
  - $|word \in V| = N$ – словарь
  - $x \in D$ – документ, $|x| = k$
  - $\bar{x}$ – N-мерный вектор, $\bar{x_i} = f(word_i, x_i)$, в котором k ненулевых компонент.
- Распределенное представление слов [Continuous Bag of Words, CBoW])
  - one-hot кодировка: каждое слово word – N-мерный вектор.
  - плотные вектора – эмбеддинги: каждое слово word – d-мерный вектор.
  - $CBOW(x) = \frac{1}{k}\sum_i^k E_i$, матрица эмбеддингов: E.

Другие признаки

- Как учесть часть речи или регистр слова?
- Специальные аффиксы?
- Является ли оно именованной сущностью?

Классификатор на основе сетей прямого распространения

- На последнем слое сети находится функция активации $softmax$ для классификации на K классов:

$$z = W_3 l_3 + b_3 \in \mathbb{R}^K$$

- То есть, на выходном слое находится K нейронов, каждый соответствует своему классу.

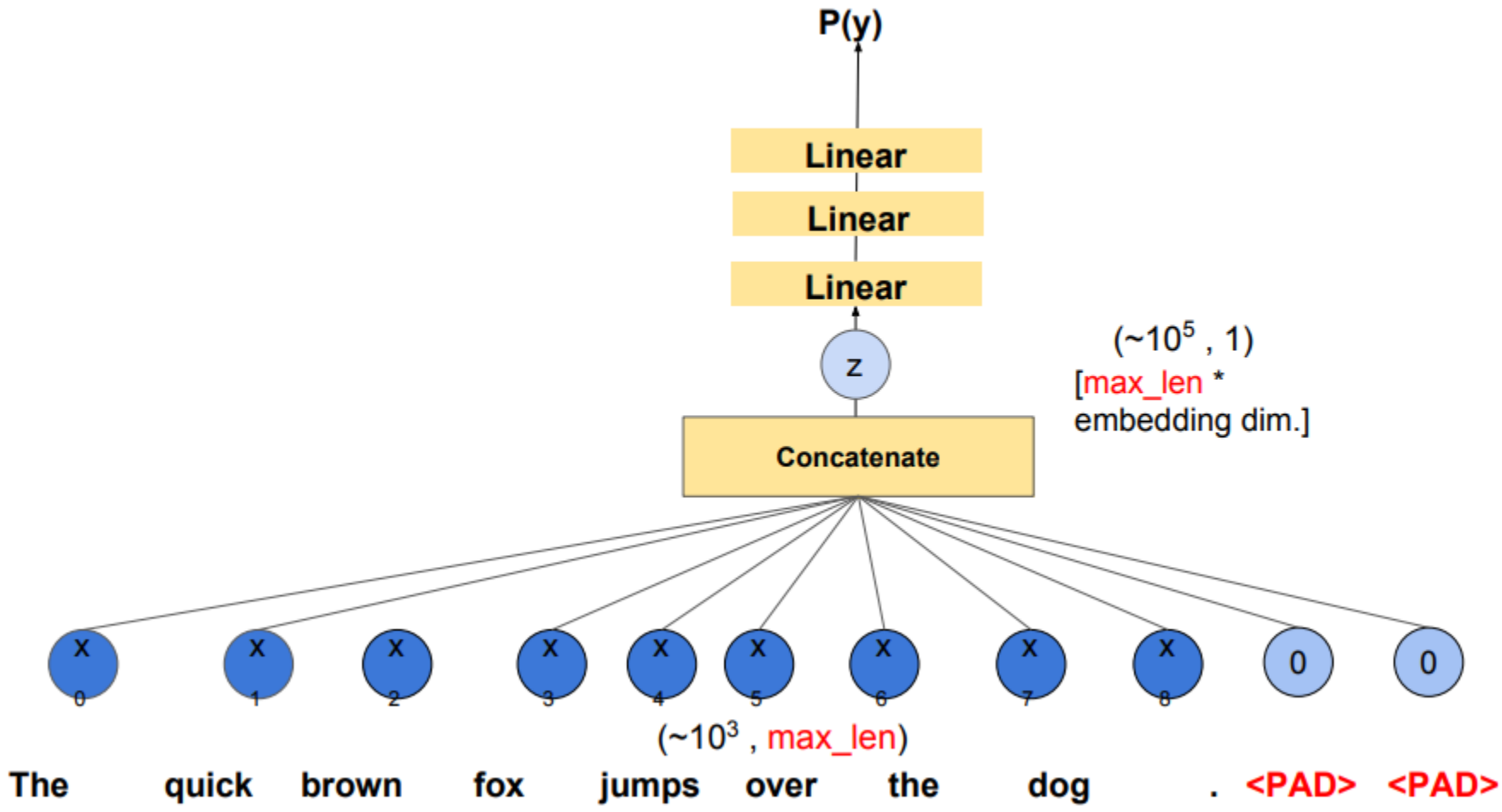
- Для итоговой классификации:

$$y_j = \mathtt{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k \in K} e^{z_k}}$$

- y_j может быть интерпретировано как вероятность класса j

- Как выбрать число скрытых слоев?
  - Например, 2000 → 1000 → 500 → 100

Multilayer perceptron



Проблемы полносвязных нейронных сетей

Проблемы:

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients

- Требуется огромное количество нейронов

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients
- Требуется огромное количество нейронов
- Серьезное переобучение

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients
- Требуется огромное количество нейронов
- Серьезное переобучение
- Отсутствие трансляционной инвариантности (веса специфичны для абсолютной координаты слова).

Возможное **решение**

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients
- Требуется огромное количество нейронов
- Серьезное переобучение
- Отсутствие трансляционной инвариантности (веса специфичны для абсолютной координаты слова).

Возможное **решение** — предобработка данных.

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients
- Требуется огромное количество нейронов
- Серьезное переобучение
- Отсутствие трансляционной инвариантности (веса специфичны для абсолютной координаты слова).

Возможное **решение** — предобработка данных.

Возможное **решение** — введение новых типов слоев:

Проблемы полносвязных нейронных сетей

Проблемы:

- Vanishing/Exploding gradients
- Требуется огромное количество нейронов
- Серьезное переобучение
- Отсутствие трансляционной инвариантности (веса специфичны для абсолютной координаты слова).

Возможное **решение** — предобработка данных.

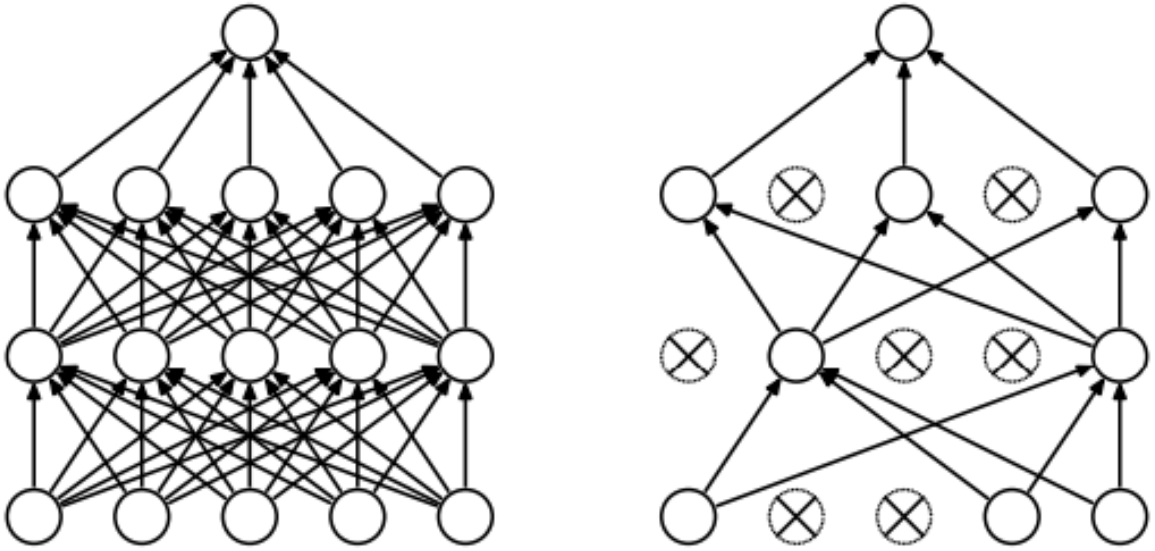
Возможное **решение** — введение новых типов слоев:

- Сверточные слои
- Пулинг
- Dropout
- Нормализация

dropout-регуляризация

$$\mathrm{NN}_{\mathrm{MLP2}}(x) = y$$

$$h_1 = g^1(xW^1 + b^1)$$
$$m^1 \sim \mathrm{Bernouli}(r^1)$$
$$\hat{h^1} = m^1 \odot h^1$$
$$h_2 = g^2(\hat{h^1}W^2 + b^2)$$
$$m^2 \sim \mathrm{Bernouli}(r^2)$$
$$\hat{h^2} = m^2 \odot h^2$$
$$y = \hat{h^2}W^3$$



(a) Standard Neural Net

(b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Сверточные нейронные сети

Сверточные нейронные сети [англ. convolutional neural network]:

- Заимствованы из области компьютерного зрения.

- Пик популярности пришелся на 2014 (до +10% аккуратности в задачах классификации), со временем были вытеснены рекуррентными нейронными сетями.

Помогают справиться с проблемами:

- Часто входы бывают переменной длины (тексты, абзацы, предложения)

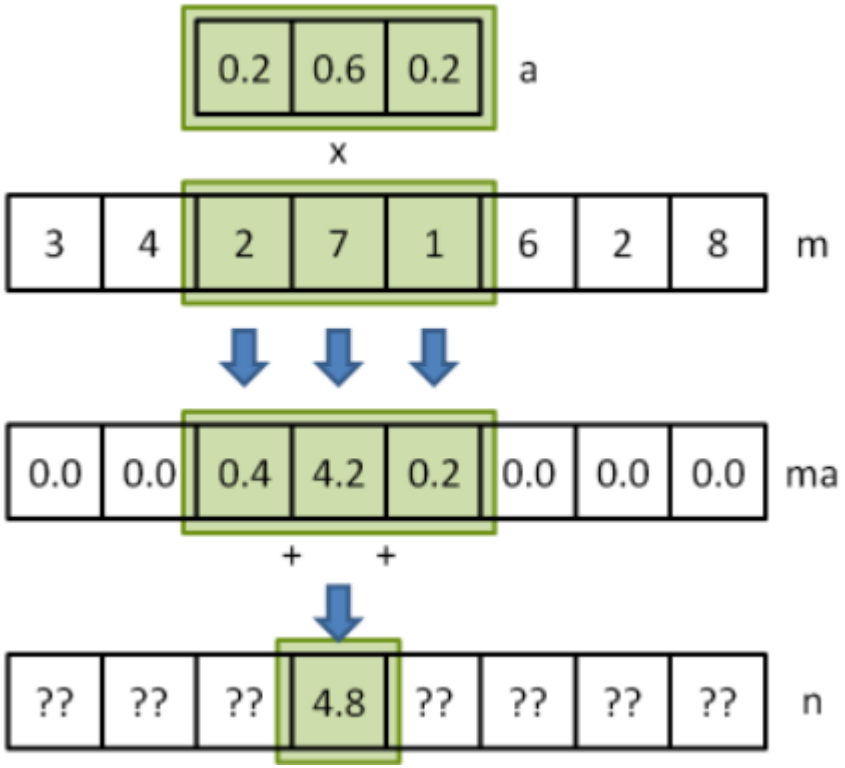
- трансляционной инвариантности

- Если использовать подход, основанный на представлении предложениями окнами, то:
  - число параметров увеличивается,
  - нужно подбирать размер окна.

Одномерная свертка

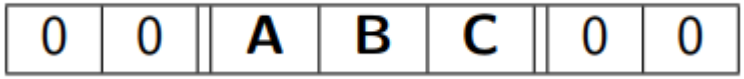
Определение

- Результатом операции свертки массива m с ядром a называется сигнал n: $n[k] = \sum_{i=-w}^{w} m[k+i]a[-i]\,[k + i]$. Обозначение: $n = m * a$.

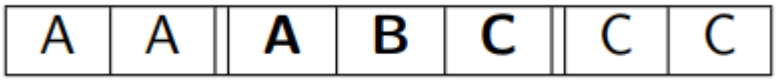


Padding

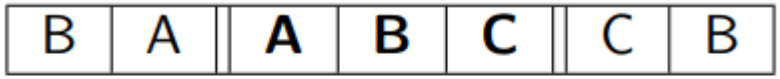
- Нулевой отступ

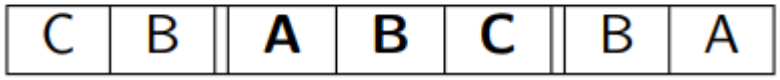


- Продолжение границы

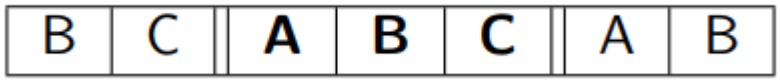


- Зеркальный отступ





- Циклический отступ



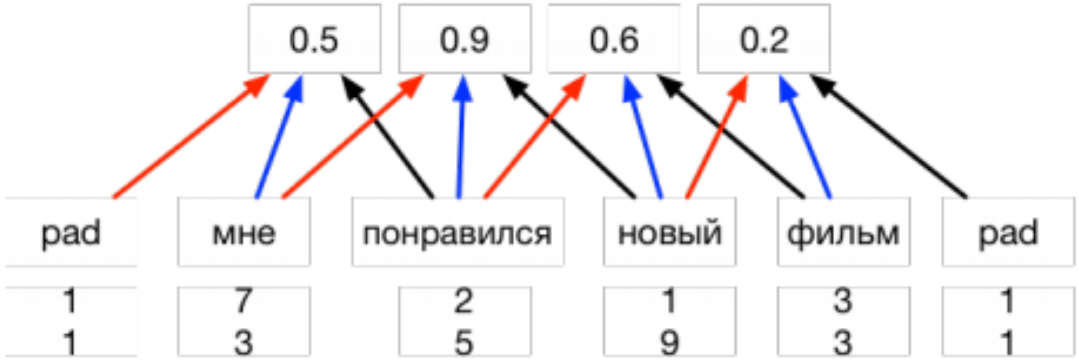
Слой свертки

Фильтр [англ. filter]:

- $w_{1:n}$ – входная последовательность слов,
- E_{w_i} – эмбеддинг слова w_i,

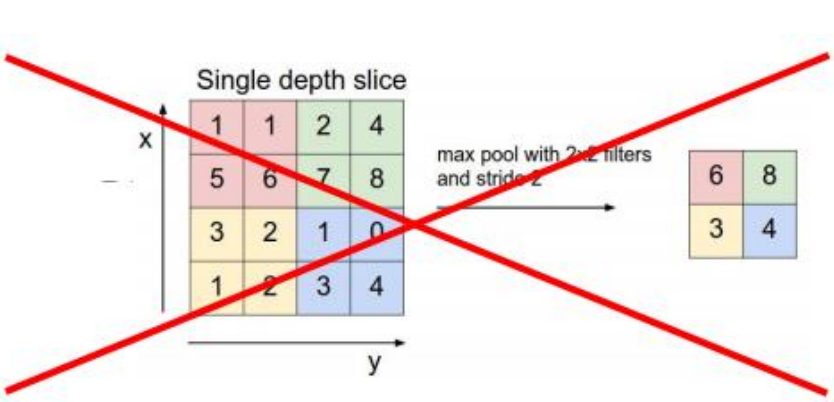
Фильтр: $p_i = g(x_i u)$

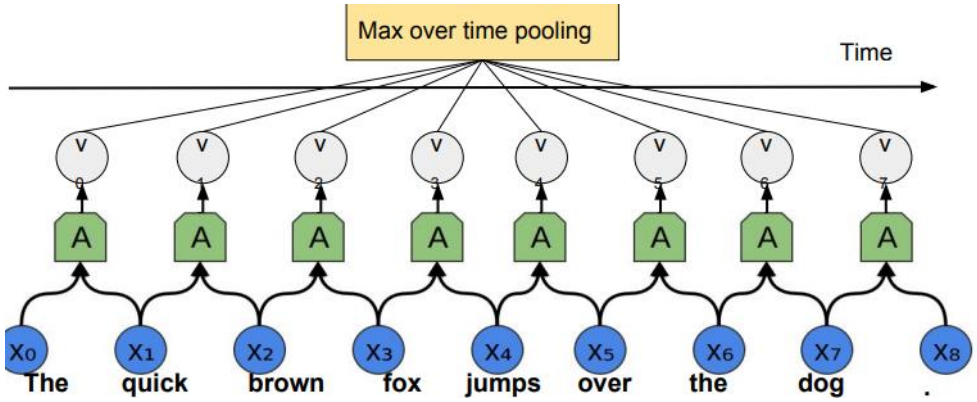
$p_i \in \mathbb{R}, x_i \in \mathbb{R}^{k \cdot d_{emb}}, u \in \mathbb{R}^{k \cdot d_{emb}}$



- Преобразуем каждое входное окно, но пока размерность входа не уменьшается!

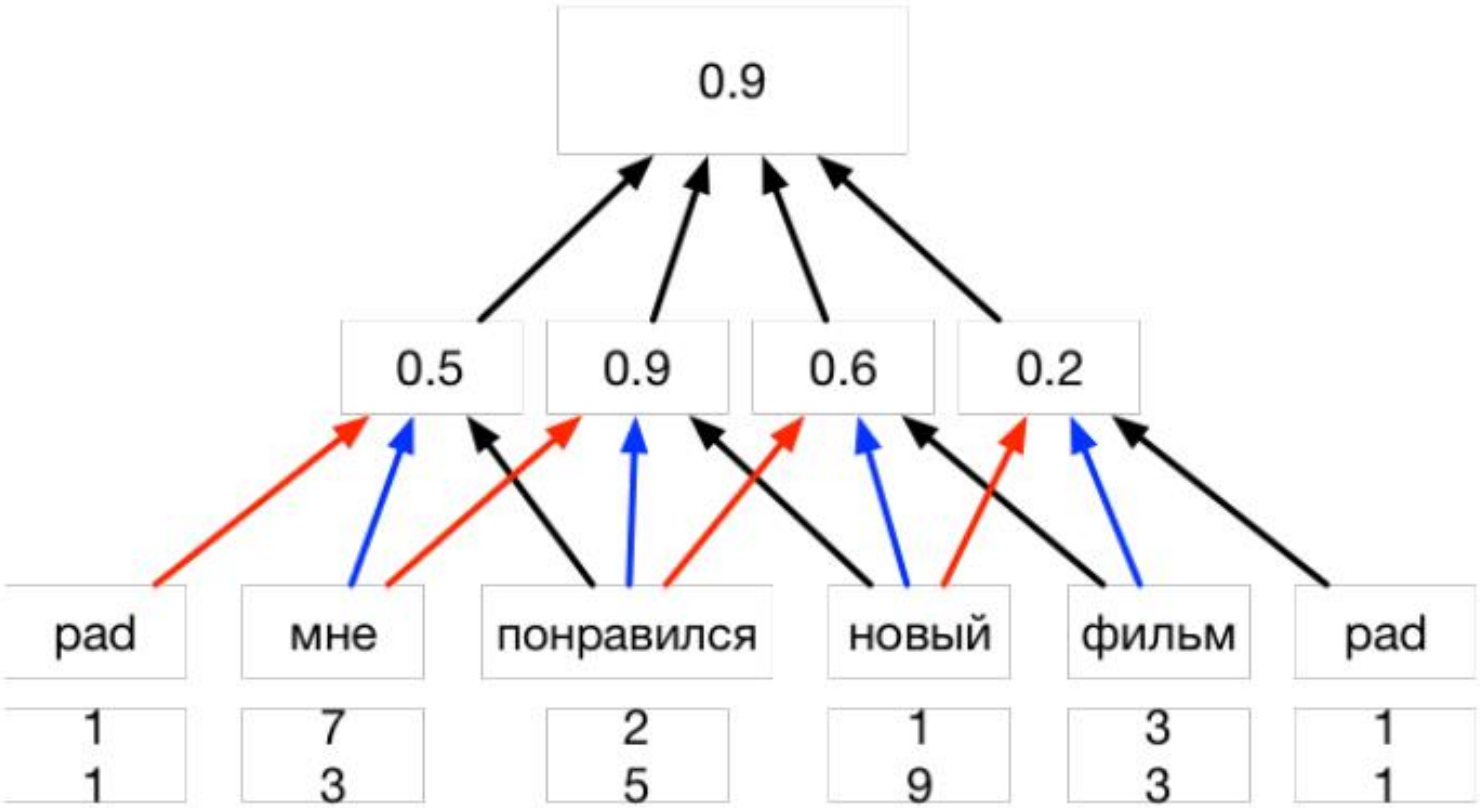
Pooling





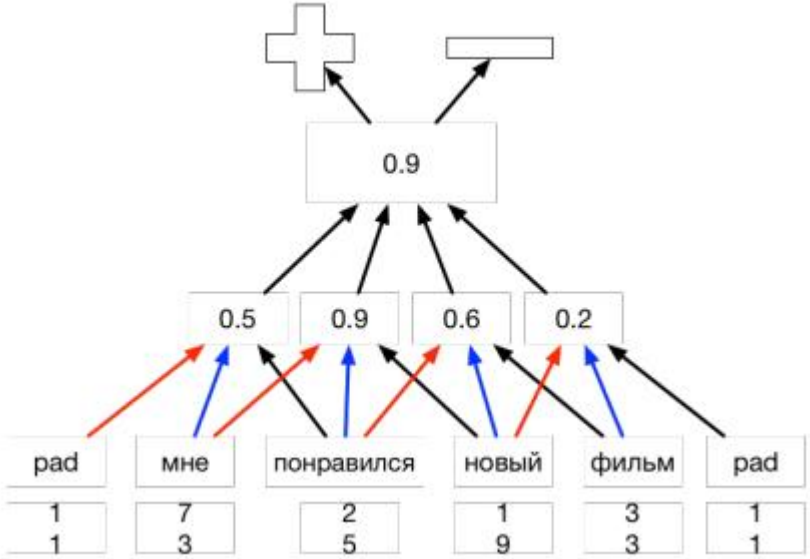
- Голосование: побеждают наиболее активные нейроны.
- Вырабатывается инвариантность к небольшим сдвигам (в рамках окна).
- Уменьшение вычислительных затрат.
- Есть avg pooling, но **max pooling over time** работает лучше в задачах классификации текстов.

Pooling


0.9
0.5
0.9
0.6
0.2
pad
мне
понравился
новый
фильм
pad
1
1
7
3
2
5
1
9
3
3
1
1

Классификатор на основе сверточной сети

- $y \in [0, 1]$ - истинные значения
- $\hat{y} = c$ - предсказанные значения.



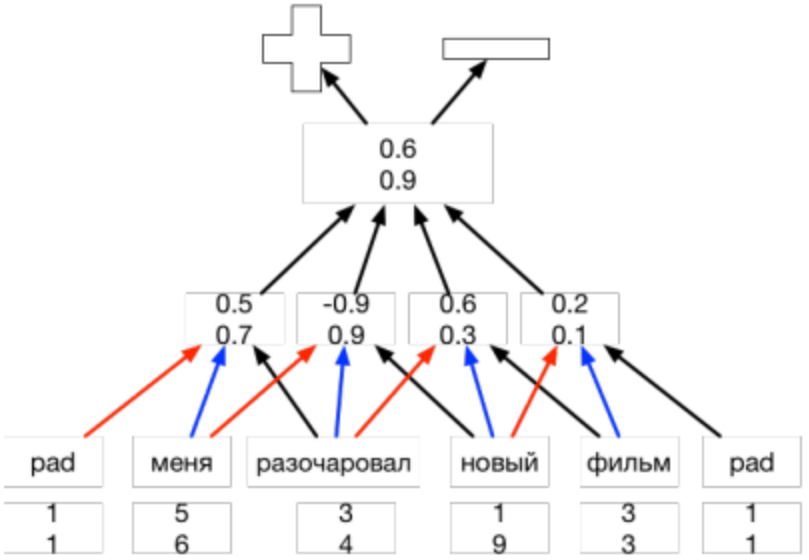
- Для обучения сверточной сети можно использовать обычный алгоритм распространения ошибки.
- Одномерные фильтры – это сильное ограничение. Что делать, если c = 0.5?

Многомерные фильтры

Применяем фильтр l раз:

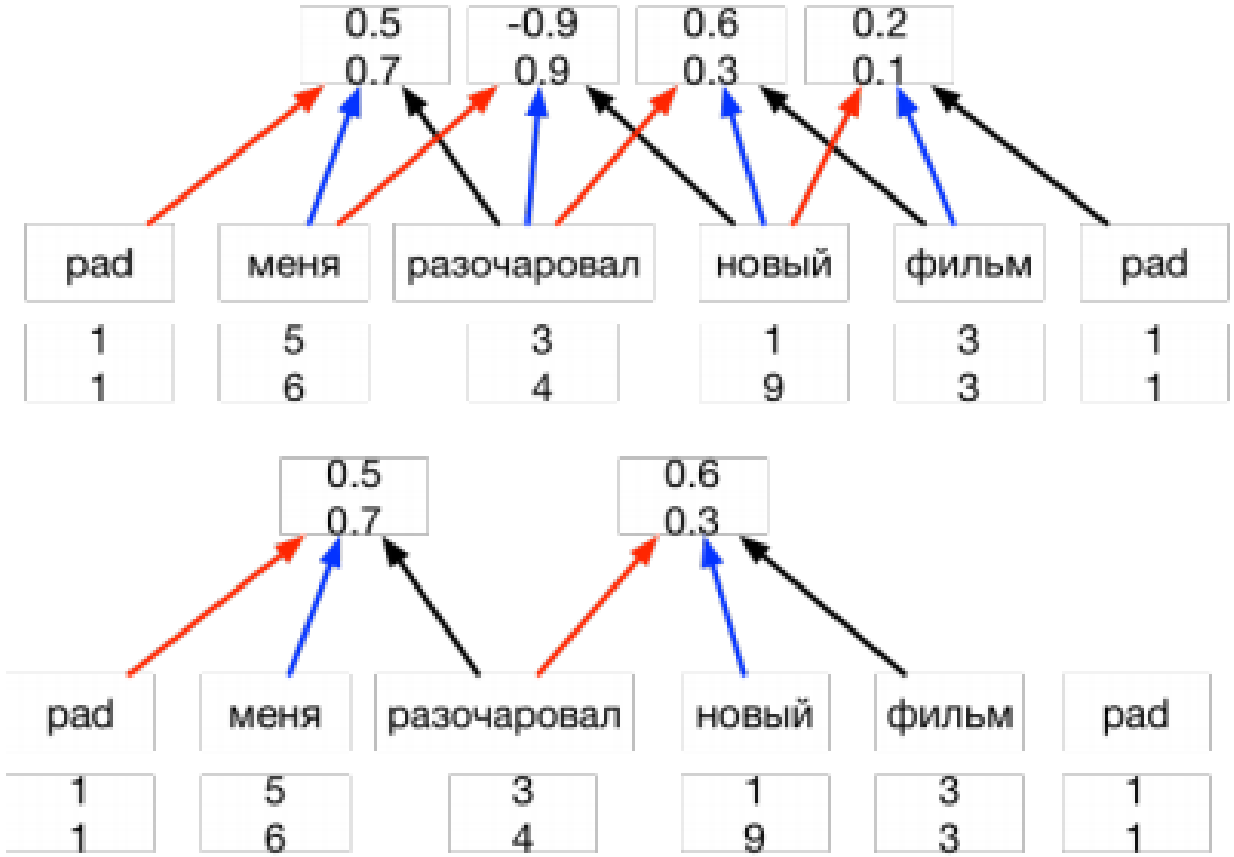
$$p_i = g(x_i \cdot U + b)$$

$$p_i \in \mathbb{R}^l, x_i \in \mathbb{R}^{k \cdot d_{emb}}$$
$$U \in \mathbb{R}^{k \cdot d_{emb} \times l}, b \in \mathbb{R}^l$$



Шаг окна

- Можно использовать непересекающиеся окна, чтобы уменьшить объем вычисления



Как выбирать вектора слов?

- Случайная инициализация (если нет обученных моделей word2vec, GloVe).
- word2vec, GloVe без обновления.
- word2vec, GloVe с обновлением на каждой эпохе (увеличивается количество параметров!)
- Несколько каналов: копируем два входа и
  - на один подаем word2vec и не обновляем эти входы во время обучения, на второй подаем word2vec и обновляем эти входы во время обучения
  - на один вход подаем word2vec, на второй – GloVe.

Как использовать padding?

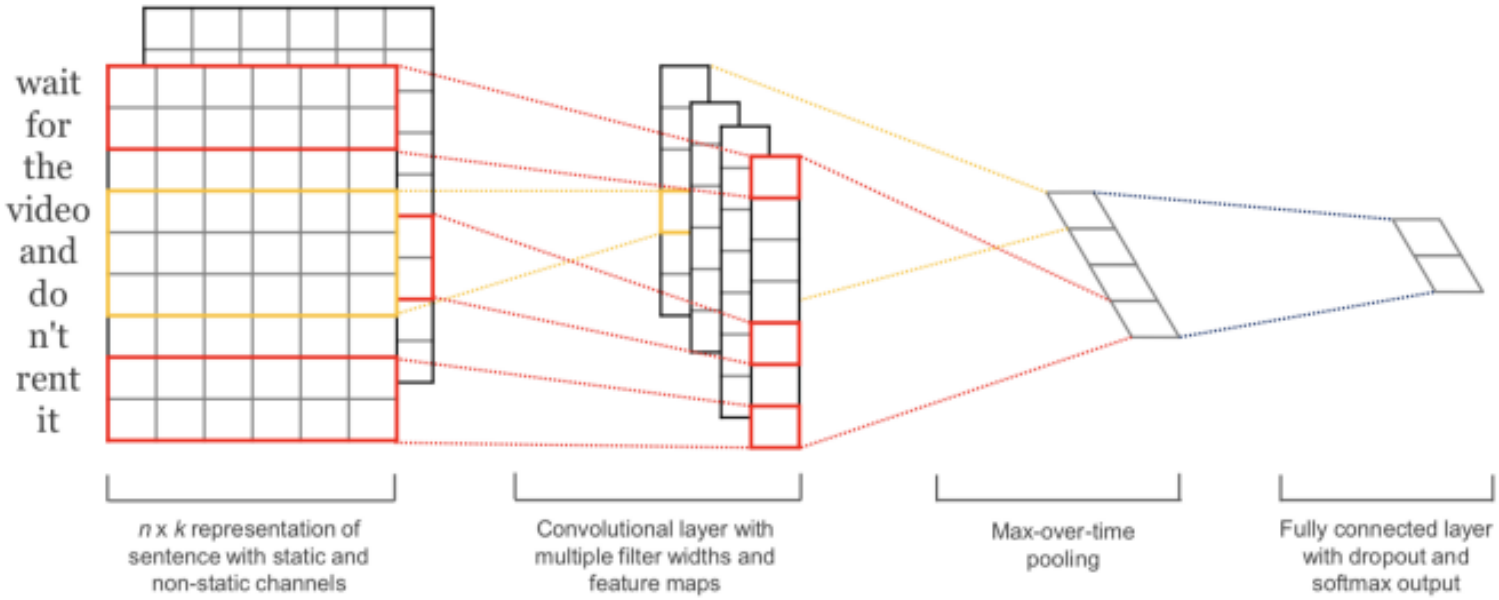
- [[мое первое короткое предложение], [второе очень длинное предложение, которое никогда не заканчивается], [третье предложение]]

- Окружить все предложения баластными символами pad и сделать их одной длины

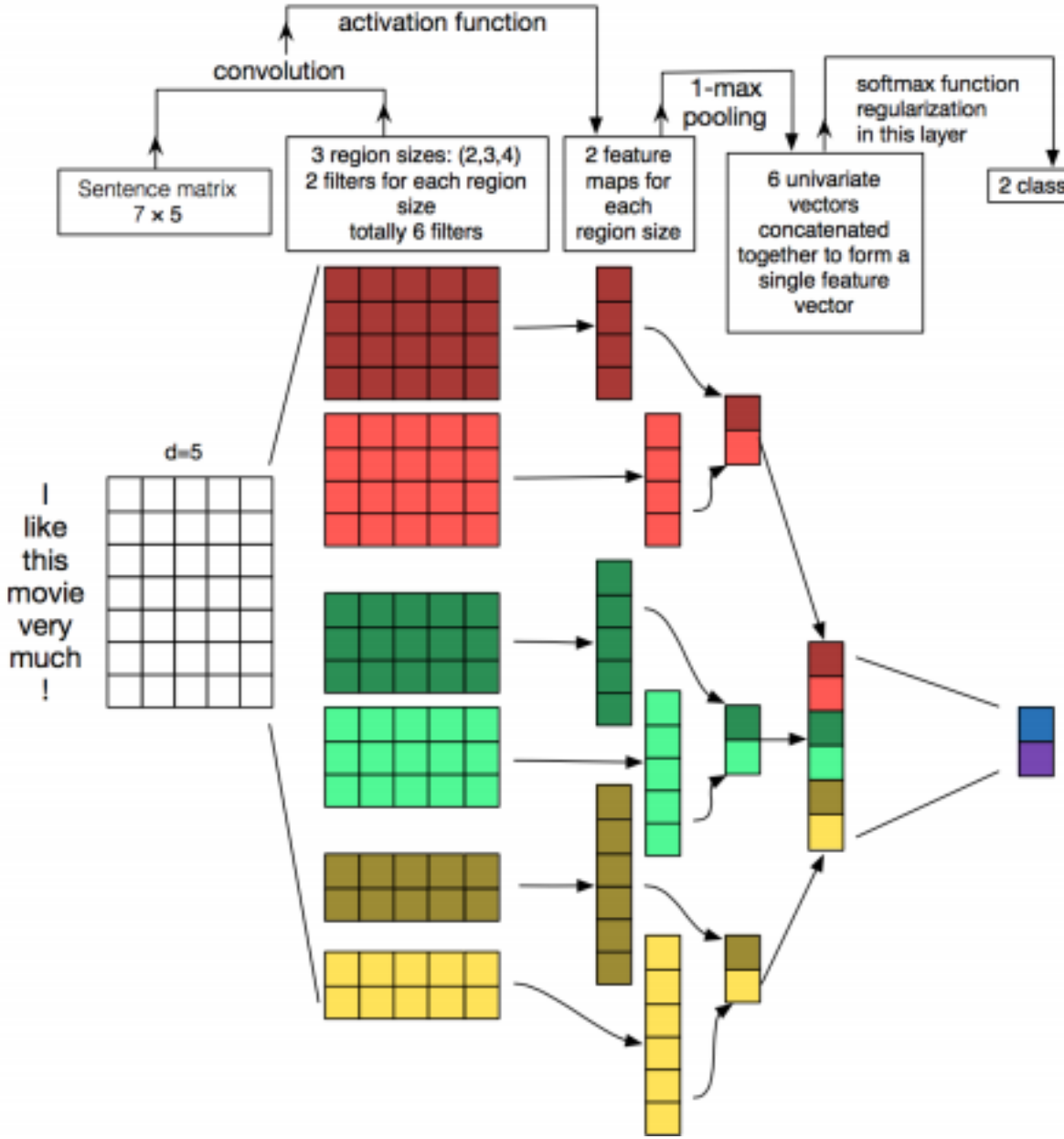
  – Надо убедиться, что max-пулинг не выберет значения, соответствующие pad.

  – Надо убрать выбросы, то есть, супер-длинные предложения, возникшие, например, из-за ошибок сегментатора

CNN для классификации предложений [Kim14]

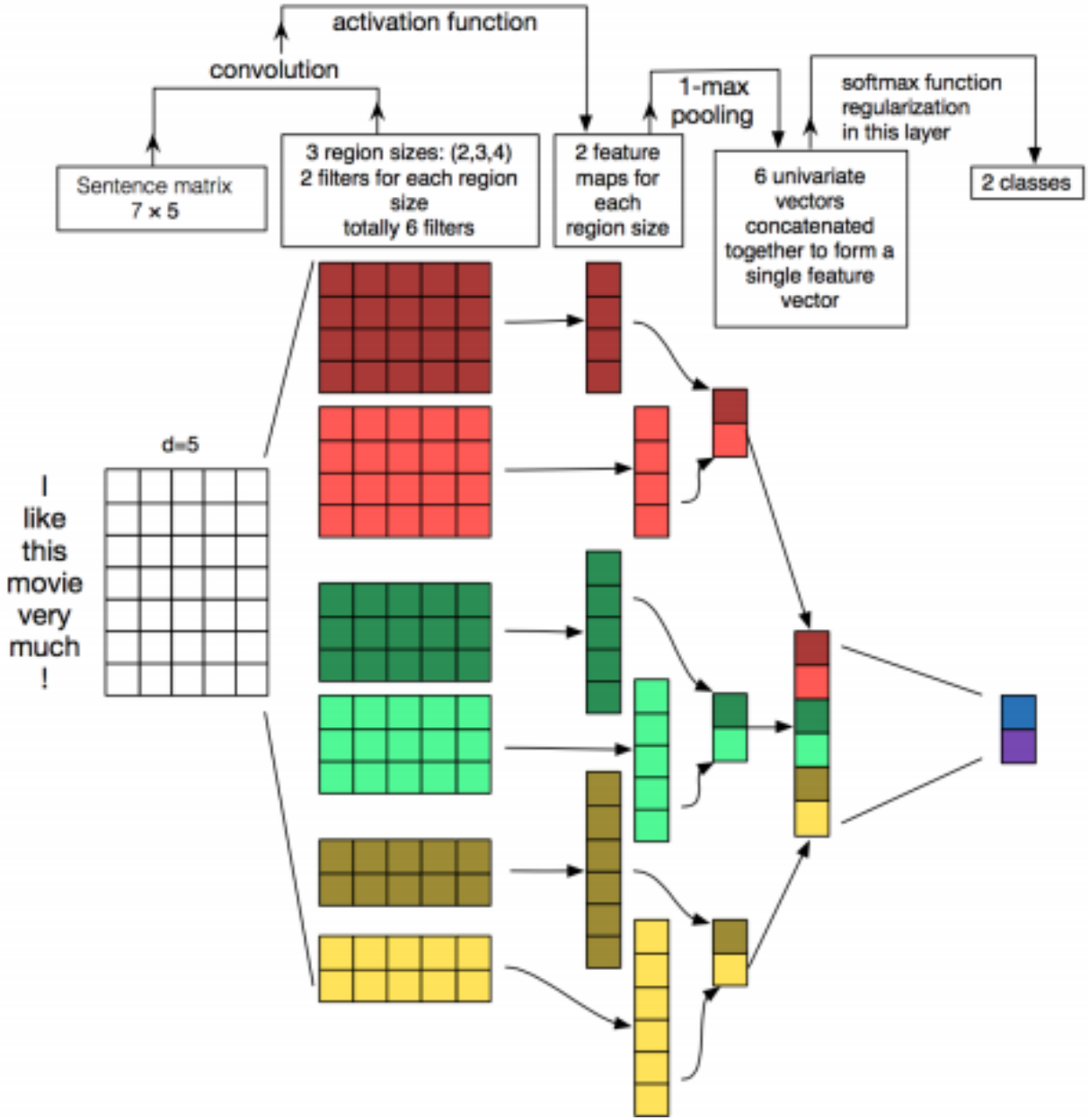


CNN для классификации предложений [ZW15]



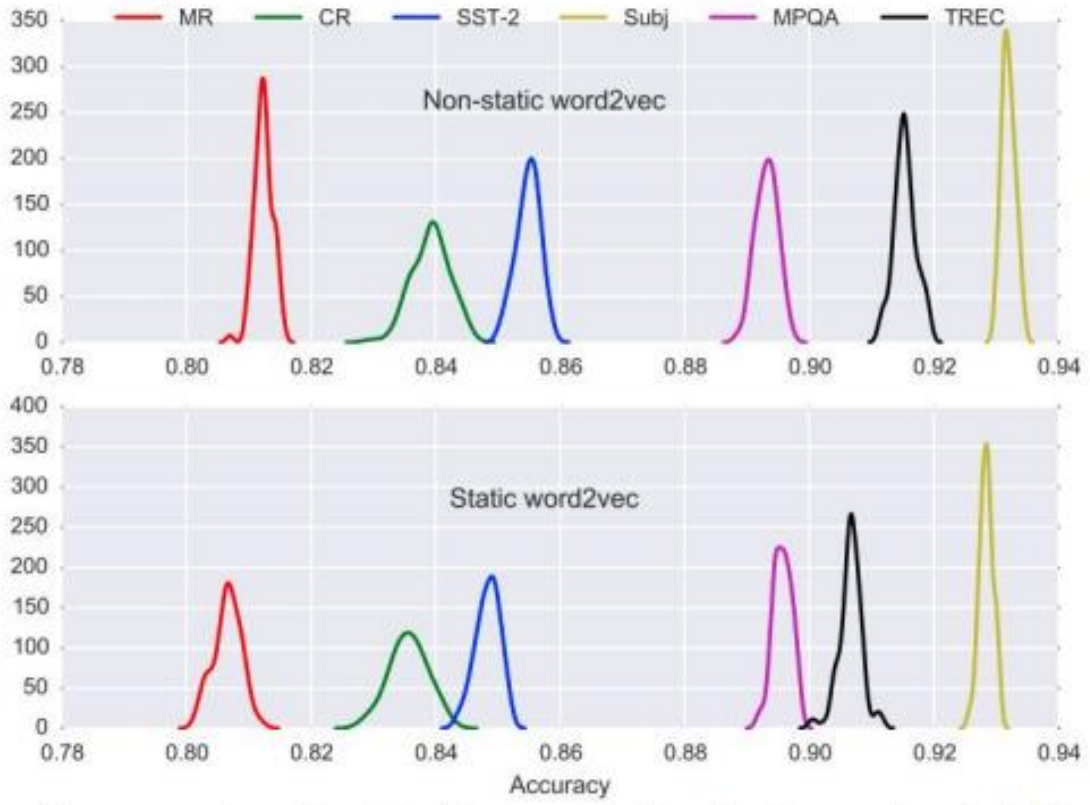
- Используем свертки различного размера.

CNN для классификации предложений [ZW15]



- Используем свертки различного размера.
- K-max pooling – берем не одну наибольшую активацию, а K в оригинальном порядке.

CNN для классификации предложений [ZW15]



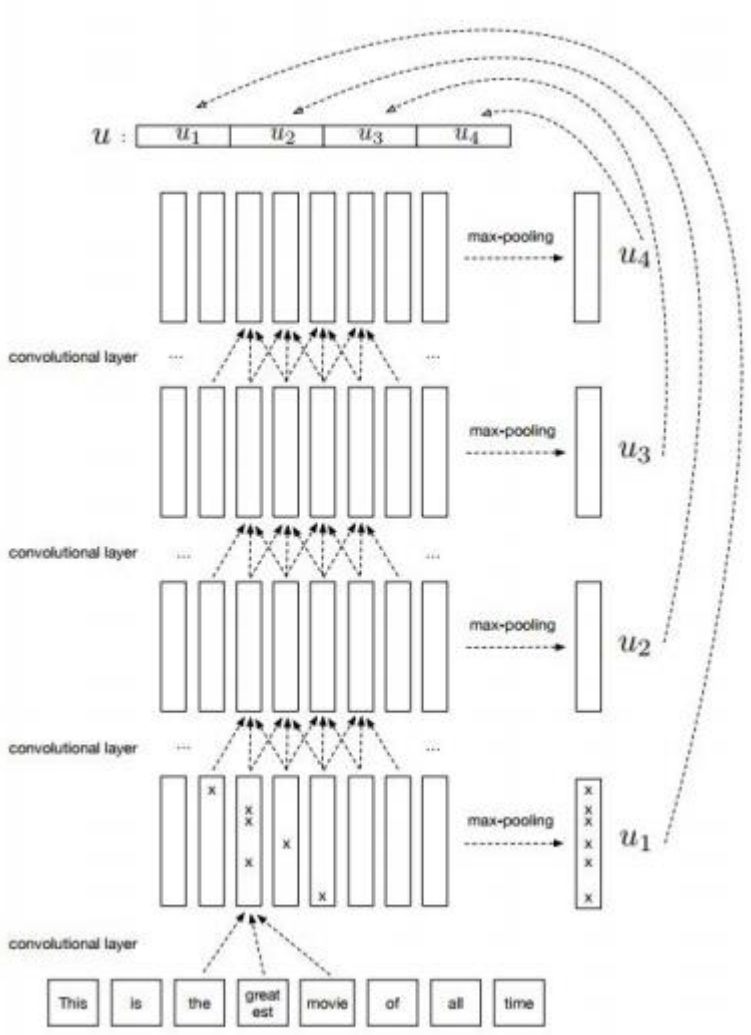
Accuracy density plots for non-static w2v (upper) and static w2v (lower) [for 10-fold CV over the 100 replications]

- Используем свертки различного размера.
- K-max pooling – берем не одну наибольшую активацию, а K в оригинальном порядке.
- Используем предобученные эмбеддинги только для инициализации, а дальше дообучаем вместе с моделью.

Надо ли соединять много слоев?

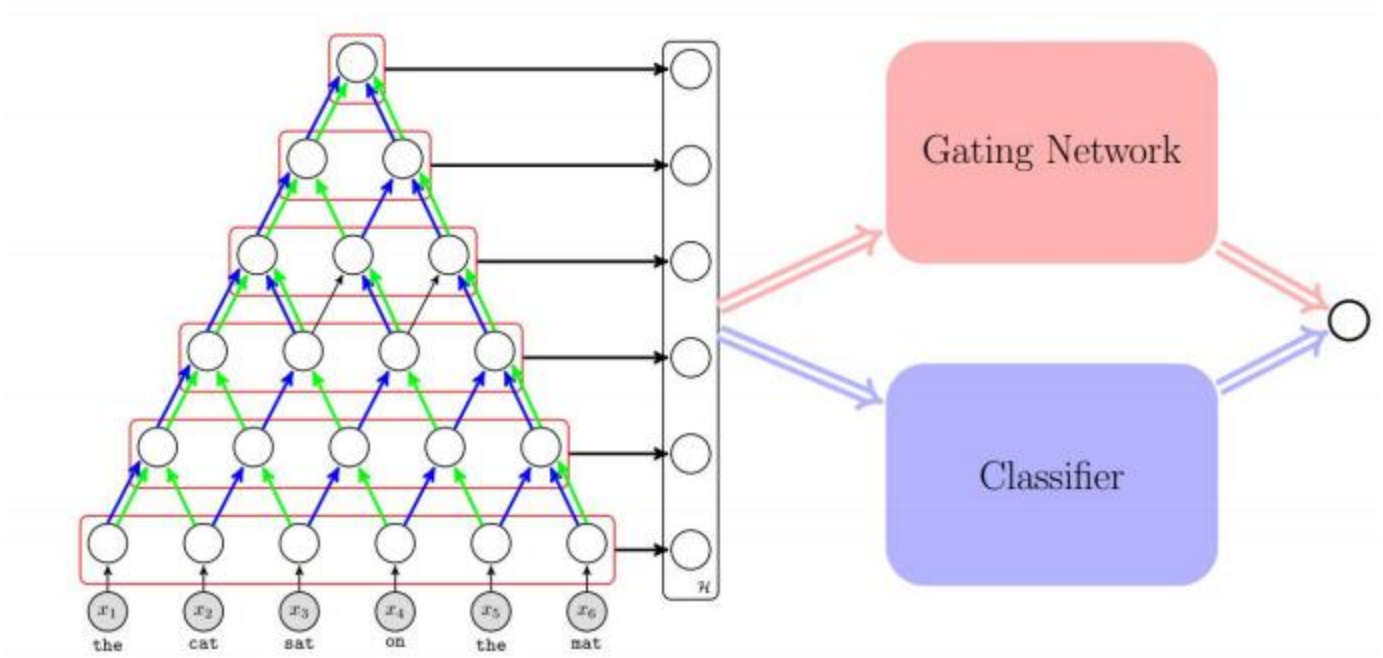
- Hierachical ConvNet [Conneau et al. 2017].

- На каждом слое представления вычисляются с помощью операции max pooling over time. Окончательное представление $u = [u_1, u_2, u_3, u_4]$ объединяет представления с разных уровней входного предложения.



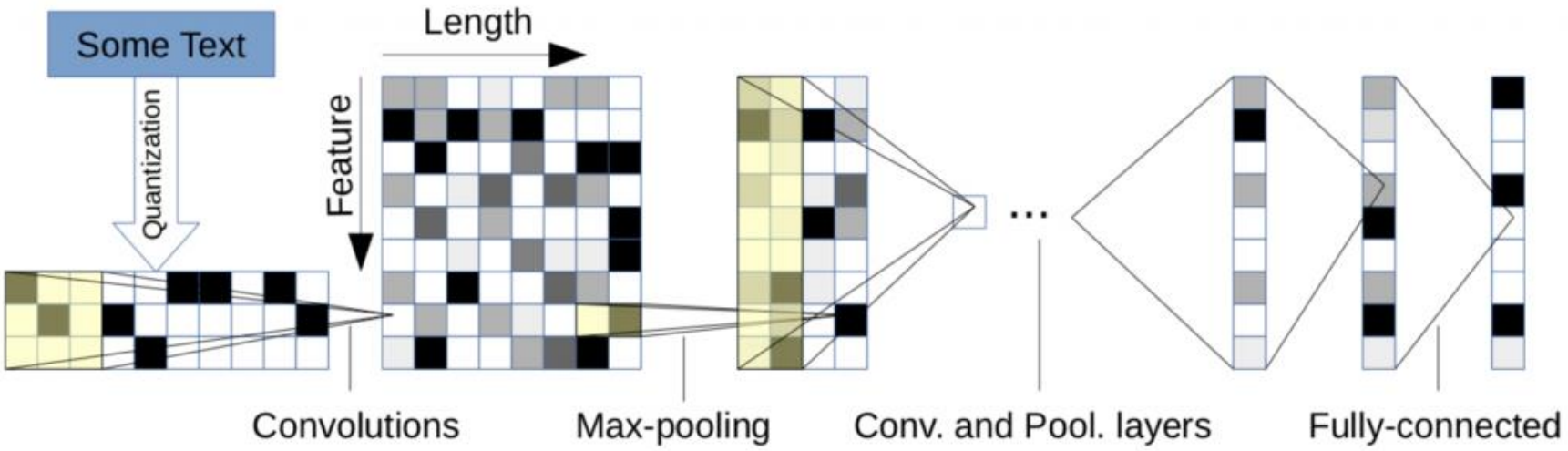
Надо ли соединять много слоев?

- AdaSent [Zhao et al. 2015].



Символьные CNN для классификации предложений [ZZL15]

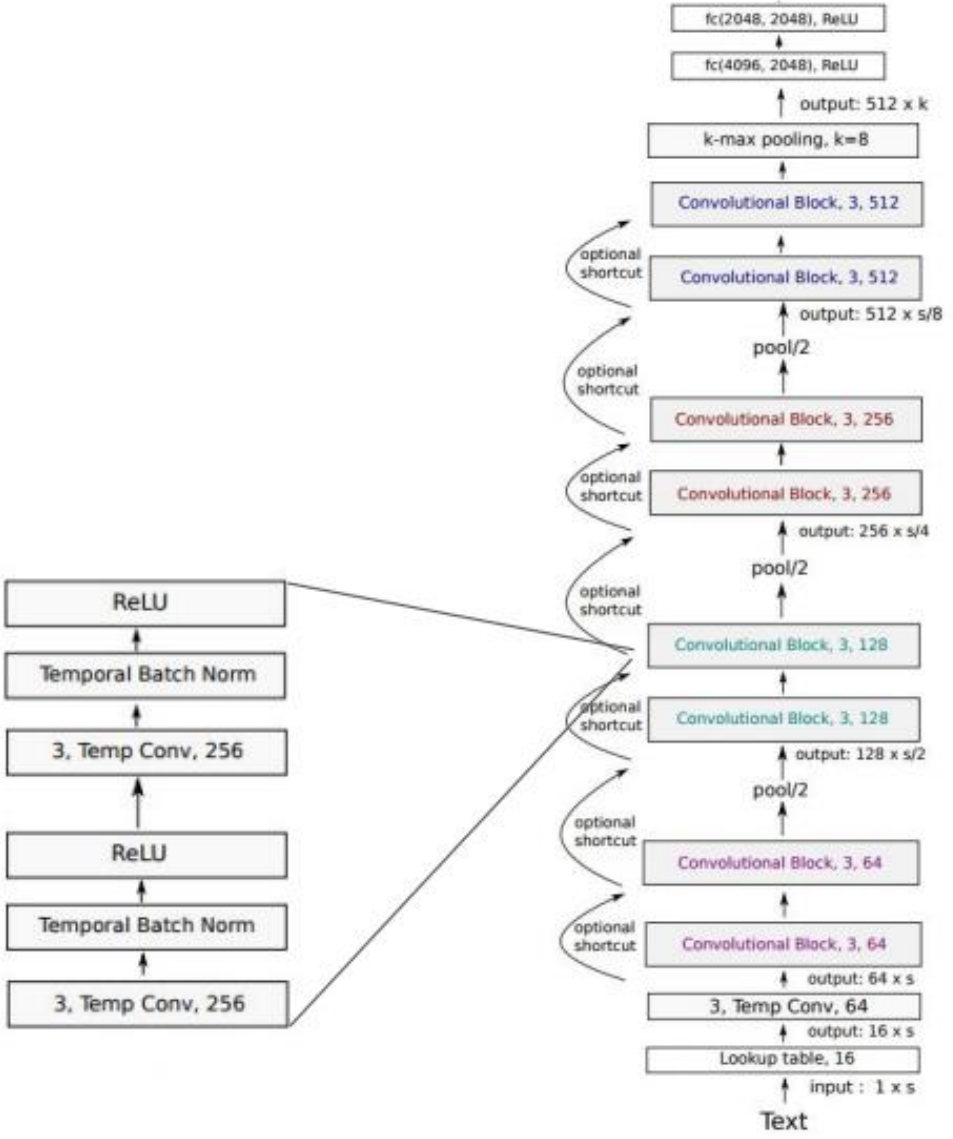
- Представление текста: one-hot вектора для 70 алфавитных и неалфавитных символов



Глубокие сверточные нейронные сети

- **Вопрос**:

Можем ли мы получить несколько процентов качества, просто объединяя гораздо больше слоев?



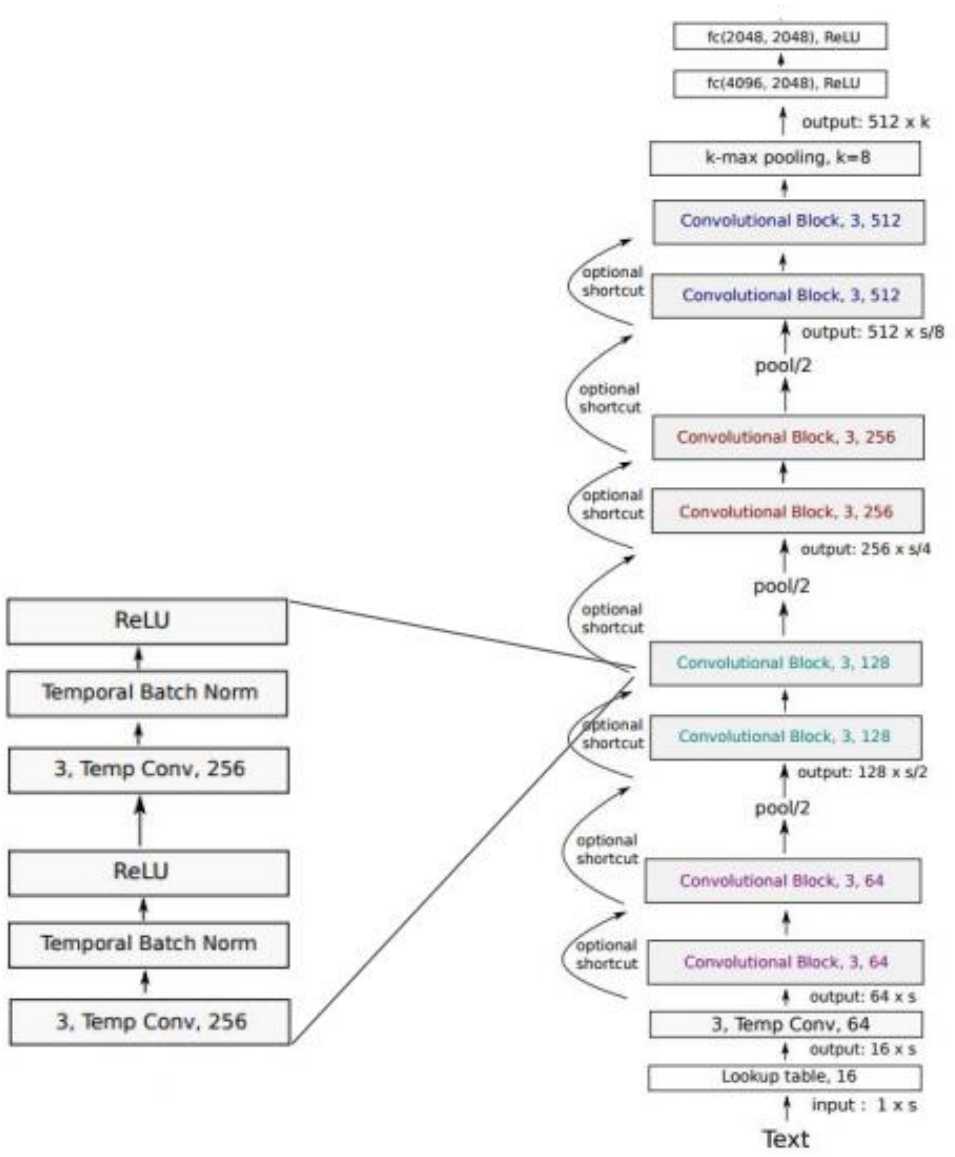
Глубокие сверточные нейронные сети

- **Вопрос**:

Можем ли мы получить несколько процентов качества, просто объединяя гораздо больше слоев?

- **Ответ**:

Это имеет смысл, когда работаем на **уровне символов**.



Глубокие сверточные нейронные сети

- **Вопрос**:

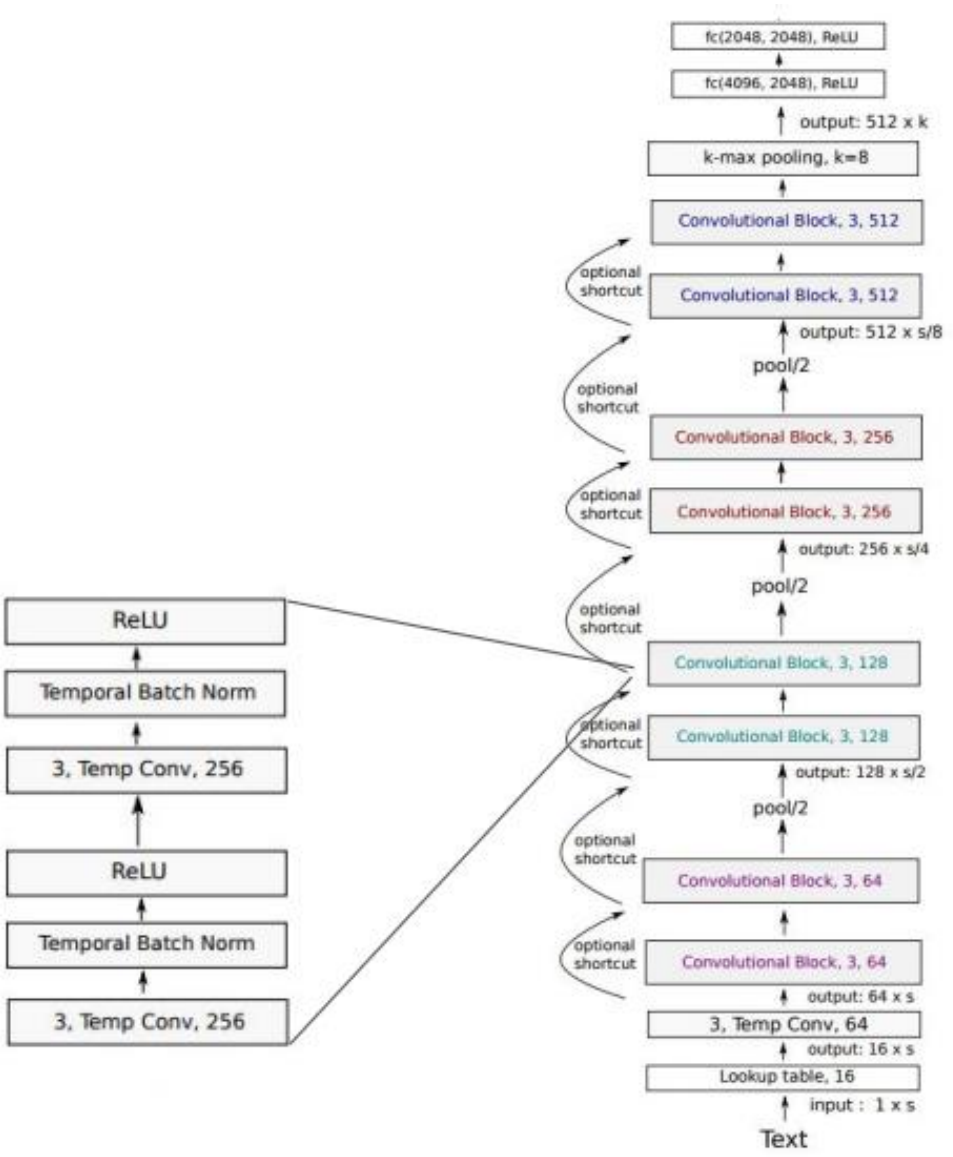
Можем ли мы получить несколько процентов качества, просто объединяя гораздо больше слоев?

- **Ответ**:

Это имеет смысл, когда работаем на **уровне символов**.

- VDCNN [Conneau et al. 2015]

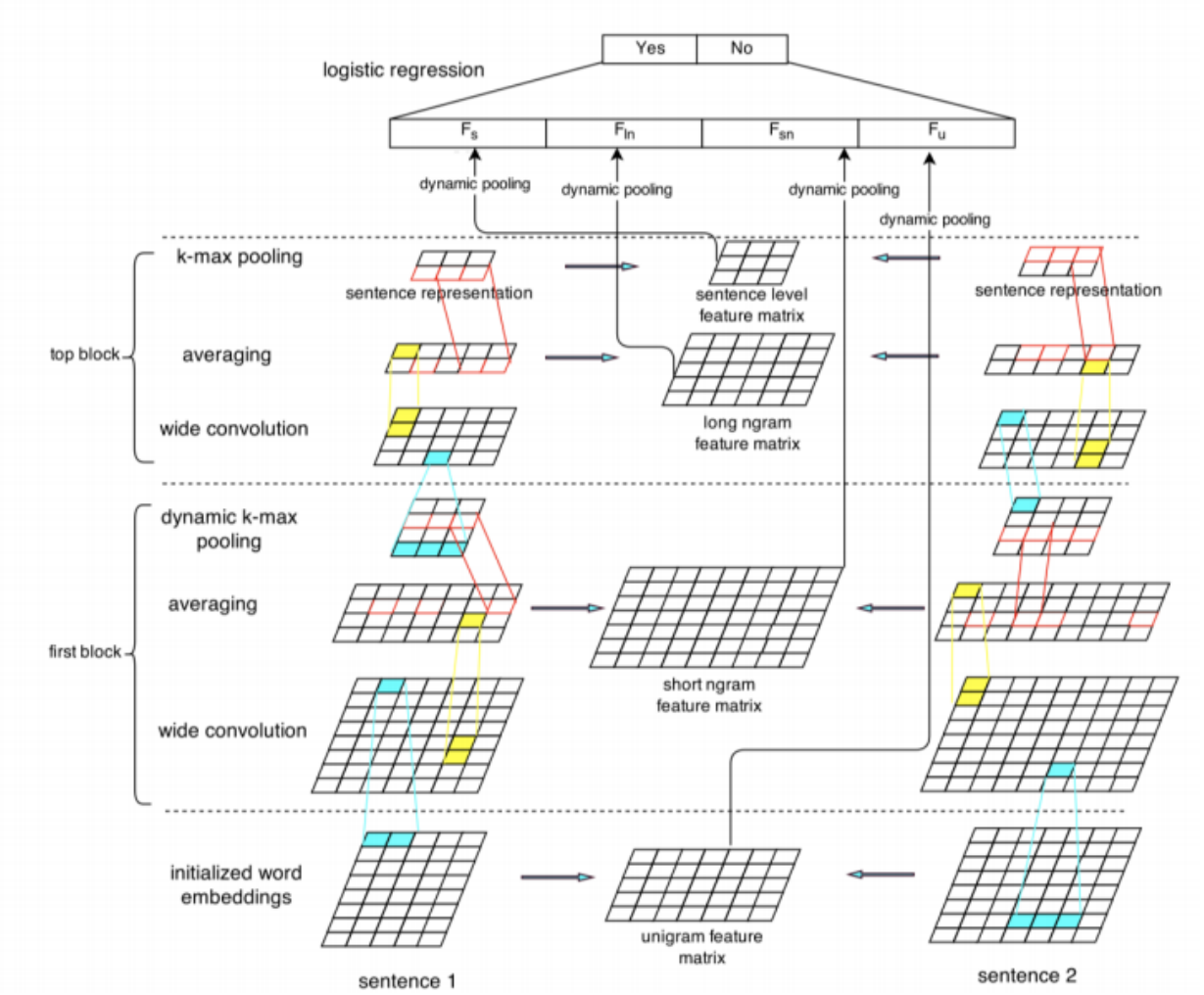
ResNet-like network with
29 conv. layers



Bi-CNN-MI для определения парафраза [paraphrase detection] [YS15]

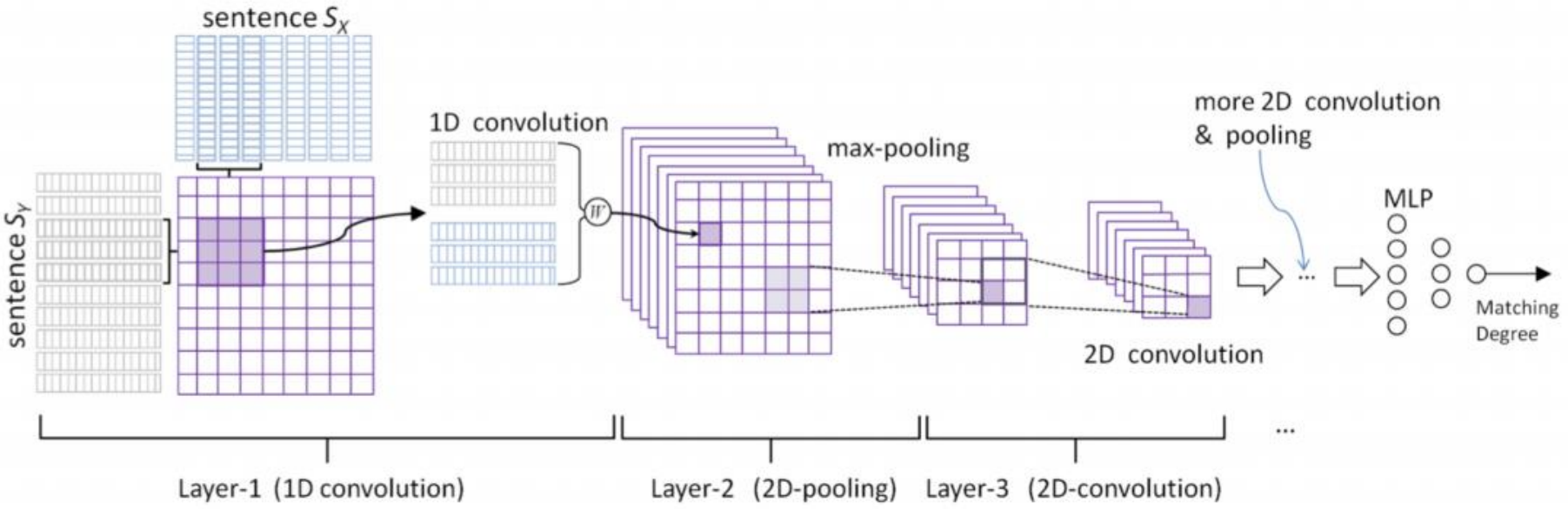
A1: Detroit manufacturers have raised vehicle prices by ten percent
A2: GM, Ford and Chrysler have raised car prices by five percent

B1: Mary gave birth to a son in 2000
B2: He is 18 years old and his mother is Mary



CNN для определения близости между предложениями[HLLC14]

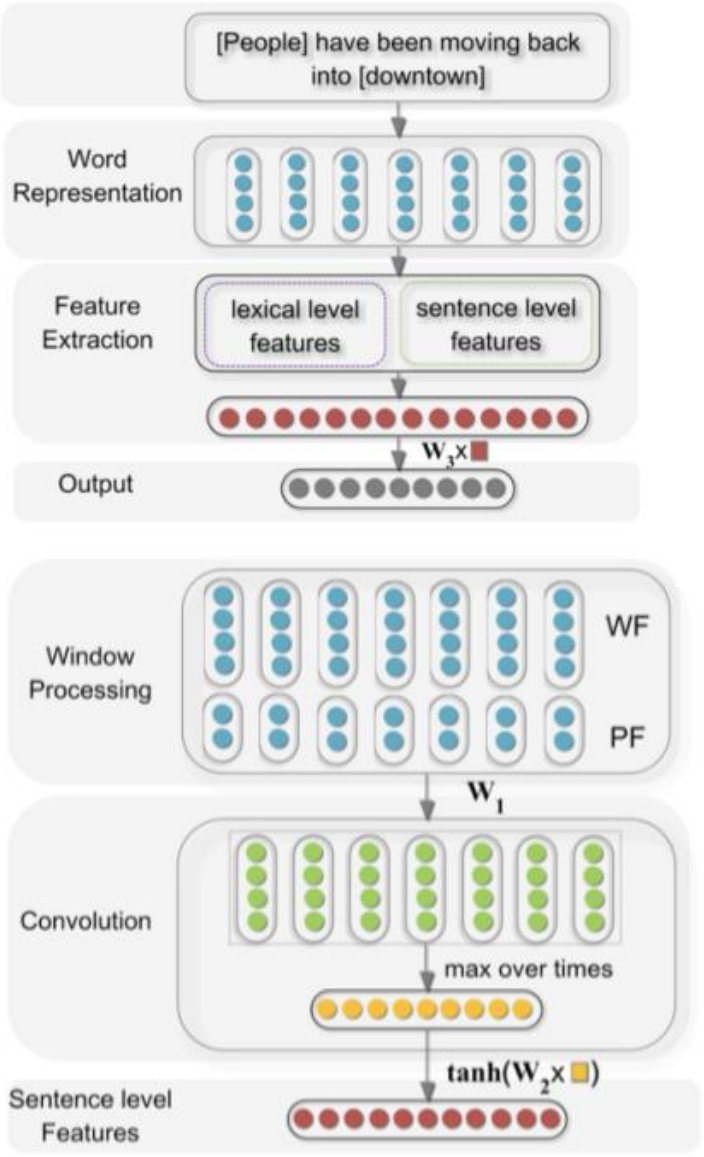
- Дополнение предложения [sentence completion]
- Определение ответа на вопрос [matching a response to a tweet]
- Определения парафраза [paraphrase detection]



CNN для извлечения отношений [relation extraction] [ZLL+14]

The $[\text{fire}]_{e1}$ inside WTC was caused by exploding $[\text{fuel}]_{e2}$

The $[\text{company}]_{e1}$ fabricates $[\text{plastic chairs}]_{e2}$



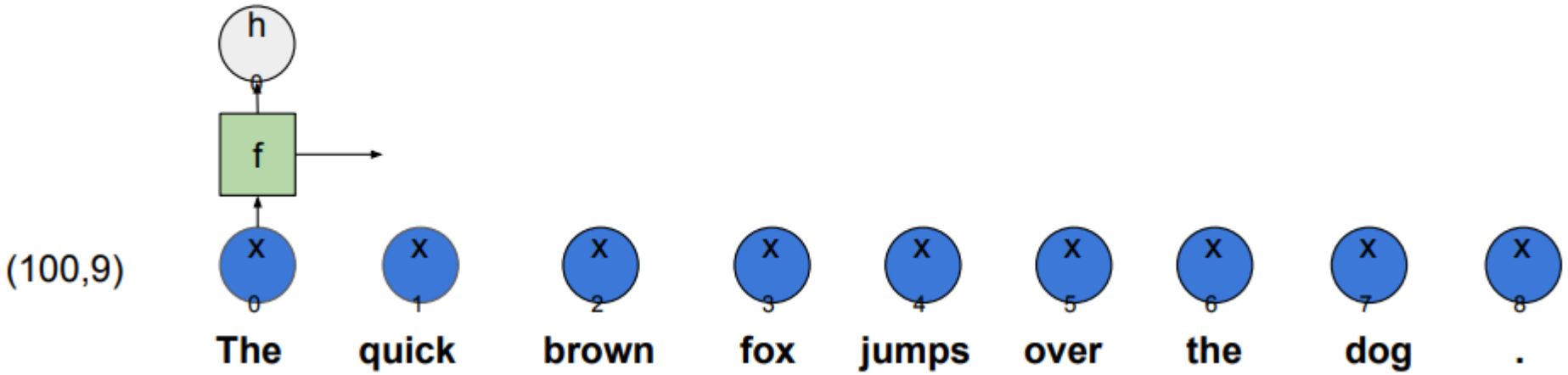
Рекуррентные нейронные сети

- RNN позволяют уйти от Марковских допущений и позволяют учитывать предысторию произвольной длины.

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$

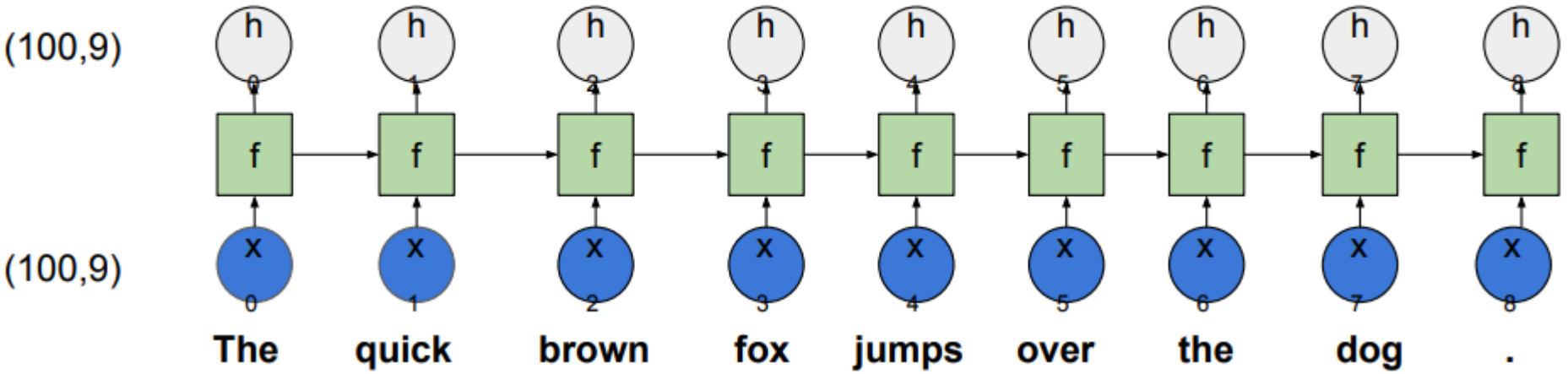
Рекуррентные нейронные сети

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Рекуррентные нейронные сети

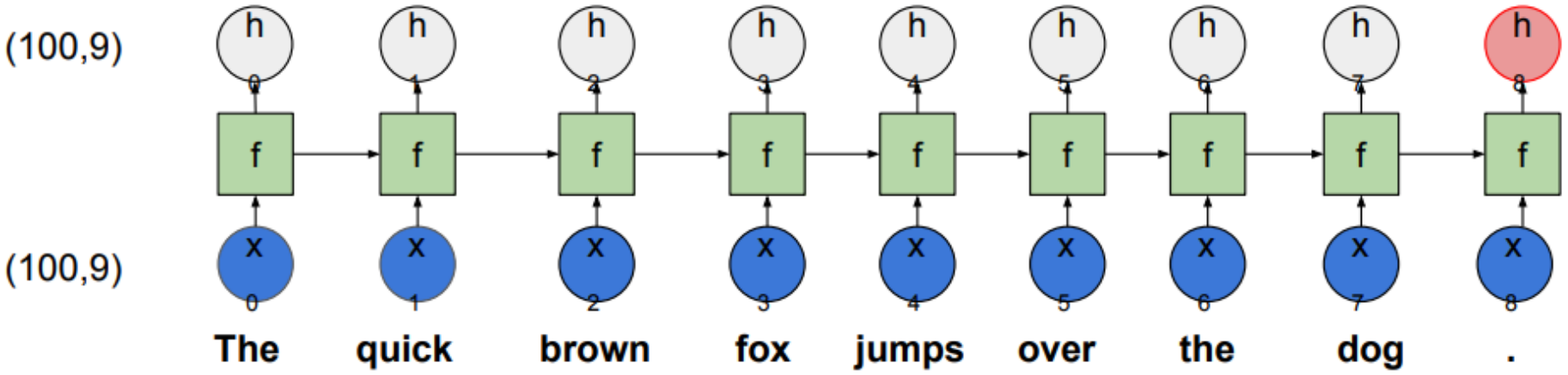
$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Рекуррентные нейронные сети

$$\mathbf{h}_8 = f(f(f(...(f(\mathbf{0}, \mathbf{x}_0)), \mathbf{x}_6), \mathbf{x}_7), \mathbf{x}_8)$$

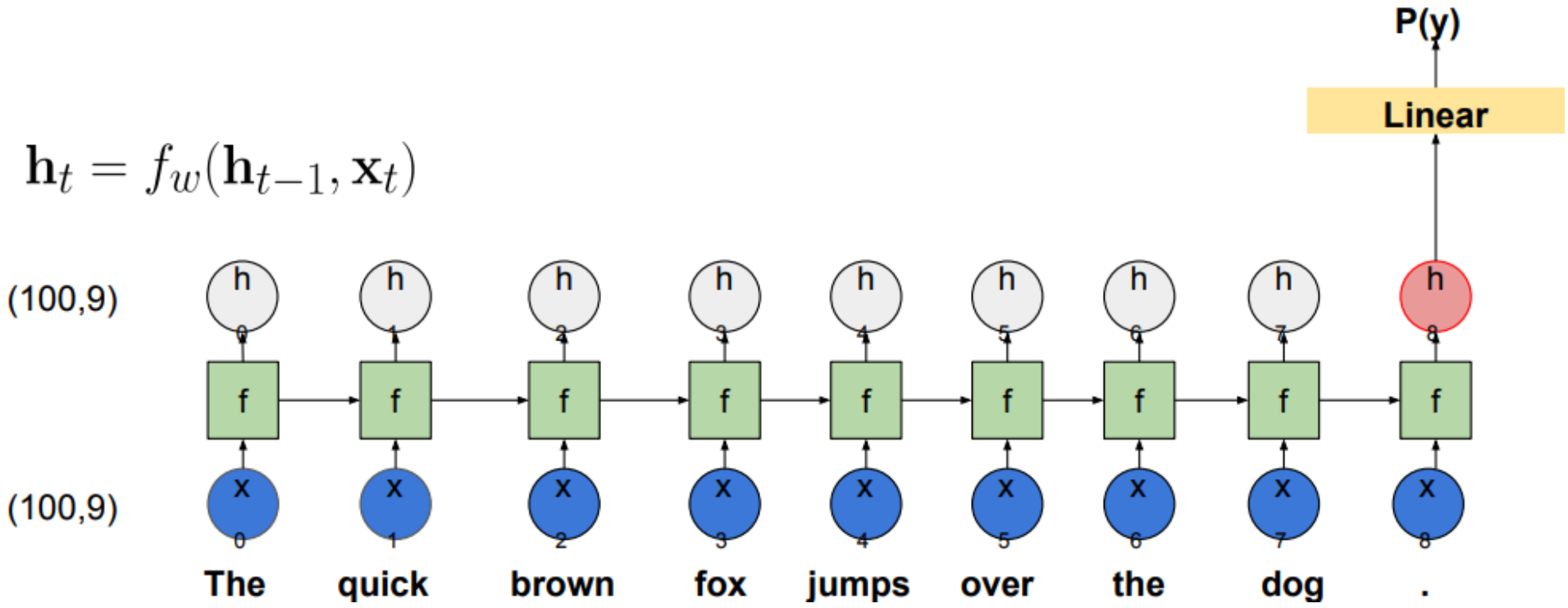
$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



Рекуррентные нейронные сети

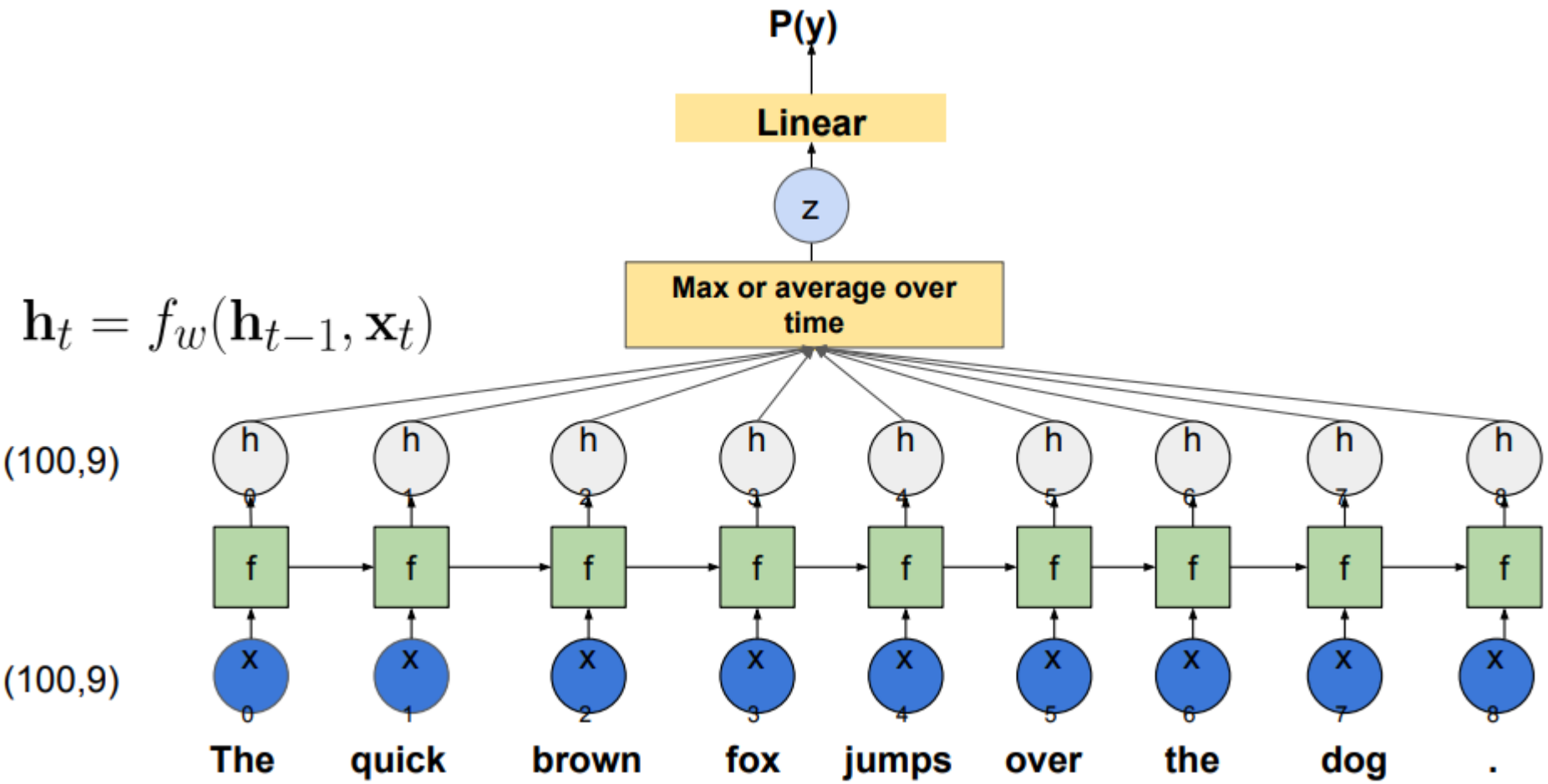
- Рекуррентная сеть для классификации

$$\mathbf{h}_t = f_w(\mathbf{h}_{t-1}, \mathbf{x}_t)$$



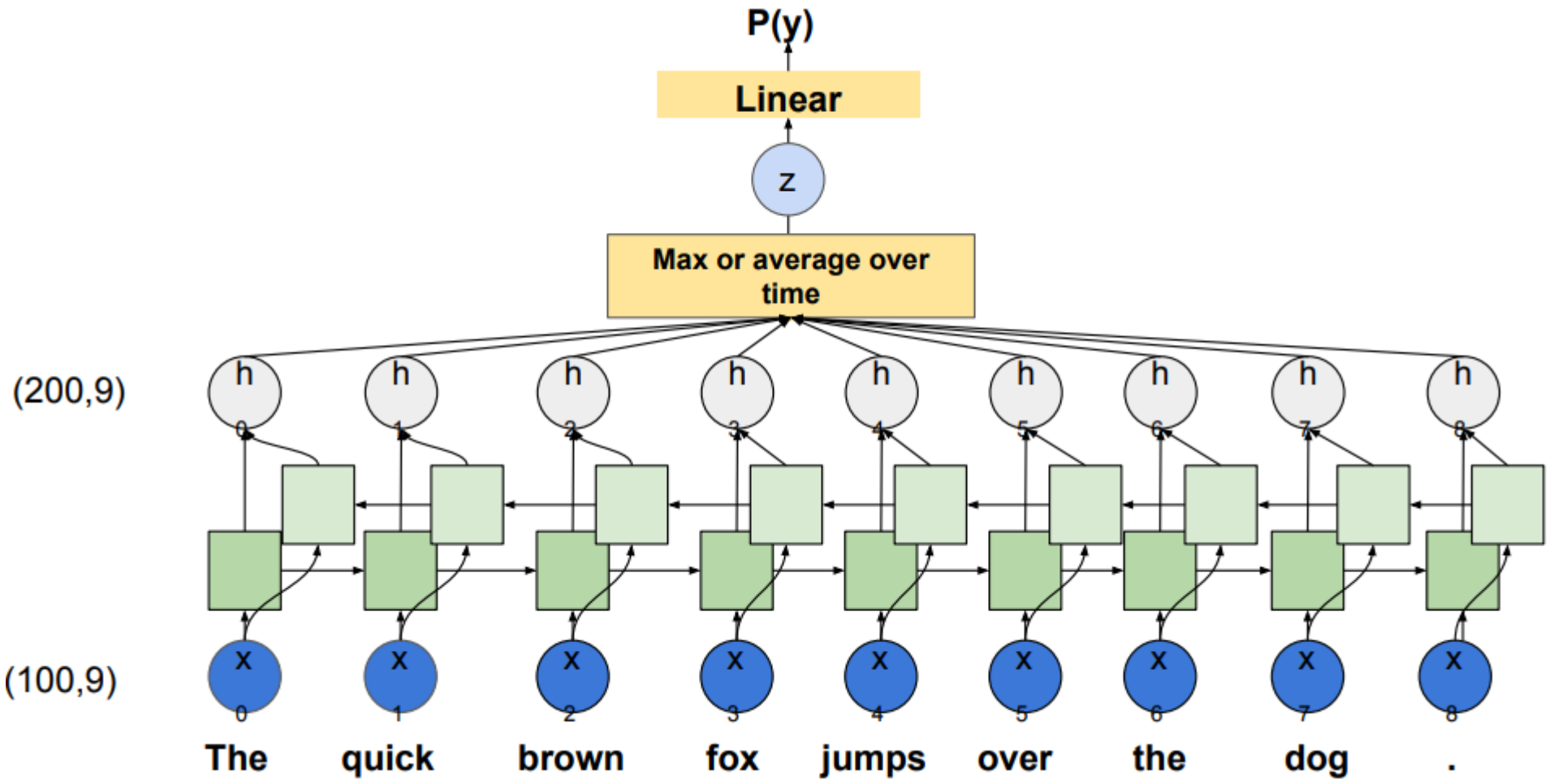
Рекуррентные нейронные сети

- Рекуррентная сеть для классификации



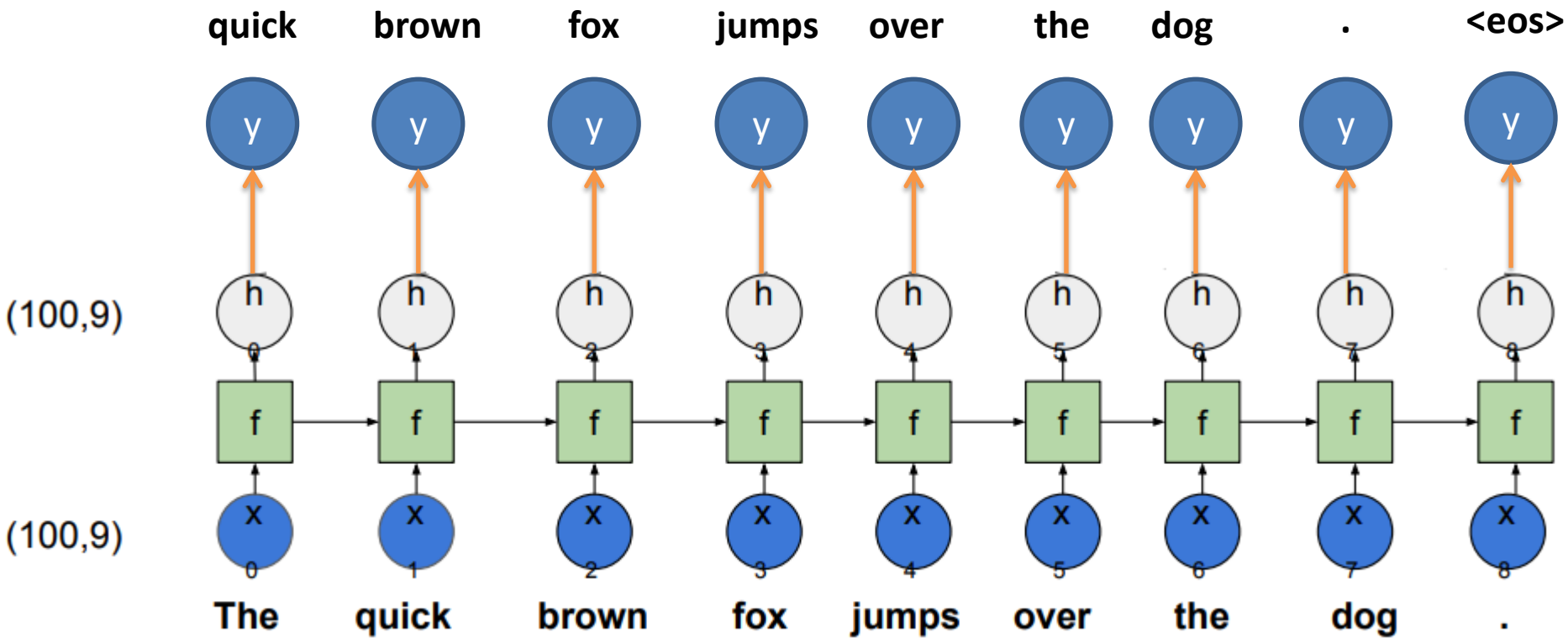
Рекуррентные нейронные сети

- Двунаправленная рекуррентная сеть для классификации



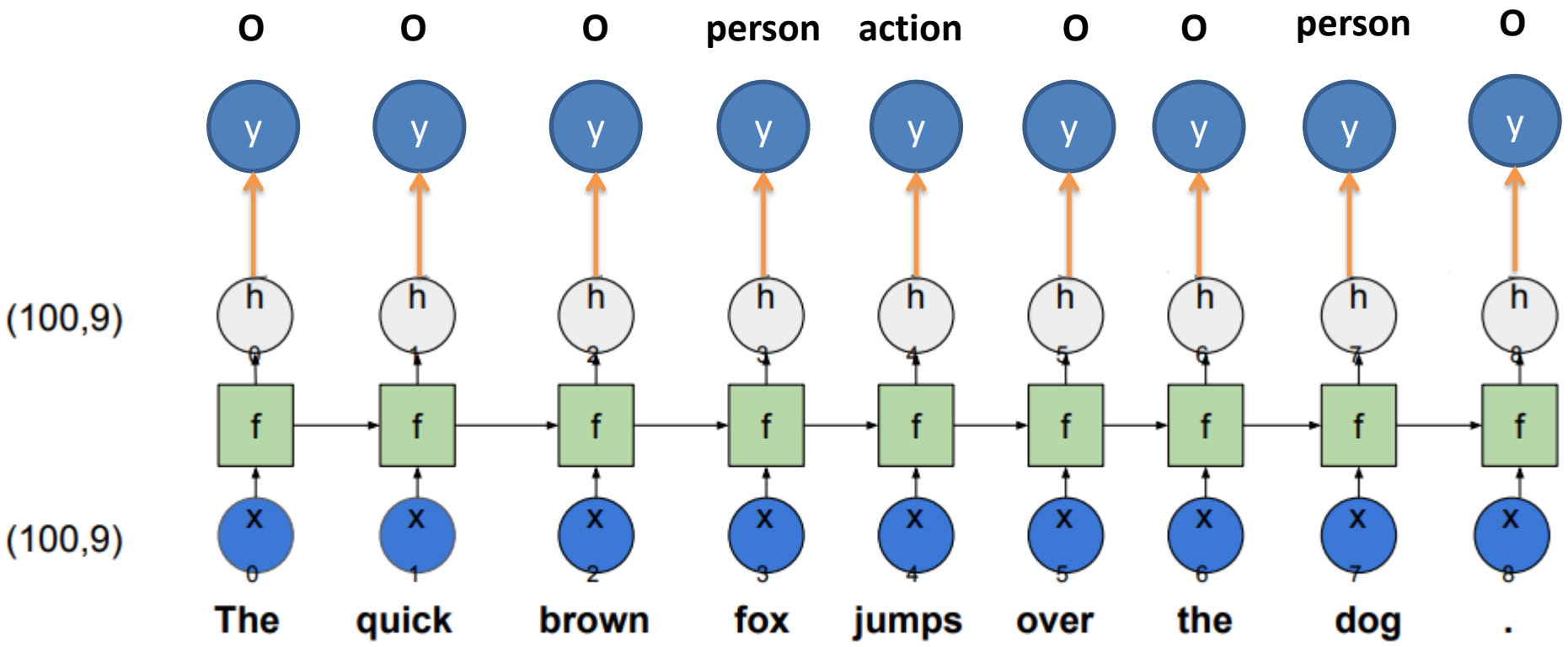
Рекуррентные нейронные сети

- Рекуррентная нейронная сеть для языковой модели.



Рекуррентные нейронные сети

- Рекуррентная нейронная сеть для языковой модели (или NER).



Управляемые архитектуры

- **RNN** трудно обучать:
  - проблема исчезающего градиента
  - проблема быстрого забывания
- Решение: управляемые нейроны специального вида: **LSTM** и **GRU**.
- Другие модификации: peephole lstm, QRNN, AWD LSTM.

CNNs vs. RNNs

- With a lot of reservations RNNs demonstrates slightly better results on the benchmark classification tasks.
- CNNs work well on the tasks that can be reduced to keyword search. Keyword mean NEs, angry terms and so on.
- Also, RNNs have slower inference than CNNs. CNNs are easier to train.
- For RNN you need more data.

1. Comparative Study of CNN and RNN for Natural Language Processing [Yin et al 2017] - https://arxiv.org/abs/1702.01923.

CNNs vs. RNNs

- With a lot of reservations RNNs demonstrates slightly better results on the benchmark classification tasks.
- CNNs work well on the tasks that can be reduced to keyword search. Keyword mean NEs, angry terms and so on.
- Also, RNNs have slower inference than CNNs. CNNs are easier to train.
- For RNN you need more data.

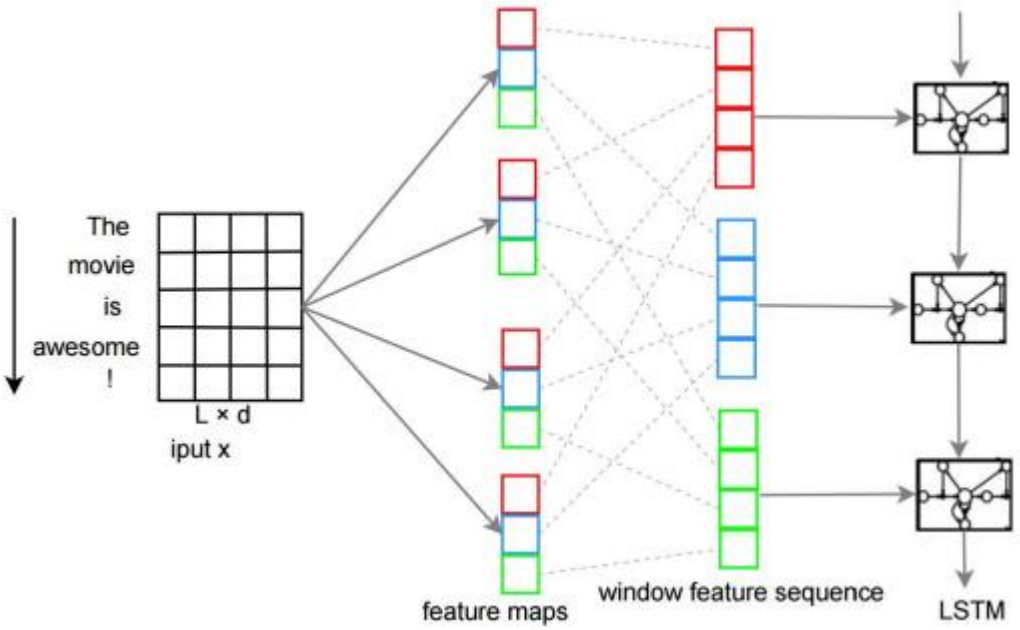
It's seems to be very task-dependent thing.
So you should try both options.

1. Comparative Study of CNN and RNN for Natural Language Processing [Yin et al 2017] - https://arxiv.org/abs/1702.01923.

Объединяем CNN и RNN

- C-LSTM [Zhou et al. 2015]

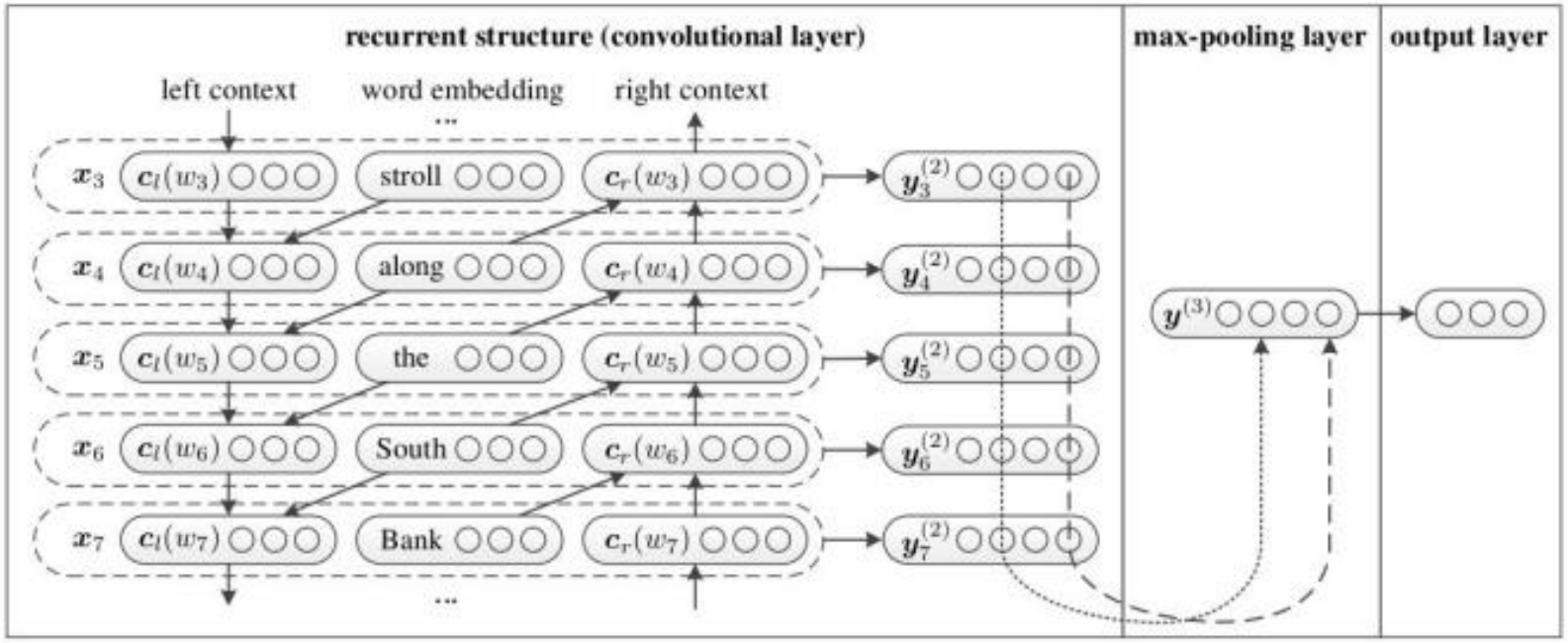
[conv.]->[LSTM]
C-LSTM использует CNN для извлечения последовательности представлений фразы более высокого уровня и подается в LSTM, чтобы получить представление предложения.



Объединяем CNN и RNN

- RCNN [Lai et al. 2015]

[Bi-RNN]->[conv.]



СПАСИБО ЗА ВНИМАНИЕ

https://github.com/king-menin/nlp-hse-autumn2018

Литература

- NLP курс в яндексе https://github.com/yandexdataschool/nlp_course
- Курс в Stanford CS224N http://cs224n.stanford.edu/
- Понимаем lstm: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Литература

- Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen, Convolutional neural network architectures for matching natural language sentences, Advances in neural information processing systems, 2014, pp. 2042–2050.

- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv:1404.2188 (2014).

- Yoon Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).

- Wenpeng Yin and Hinrich Schutze, ¨ Convolutional neural network for paraphrase identification, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 901–911.

Литература

- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao, Relation classification via convolutional deep neural network, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 2335–2344.

- Ye Zhang and Byron Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, arXiv preprint arXiv:1510.03820 (2015).

- Xiang Zhang, Junbo Zhao, and Yann LeCun, Character-level convolutional networks for text classification, Advances in neural information processing systems, 2015, pp. 649–657.