Разбор летучки

Лекция 8

Нейронные сети

Екатерина Тузова

Модель McCulloch-Pitts

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

Модель McCulloch-Pitts

$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

$$a(x, w) = \sigma(\sum_{j=1}^{n} w_j x^j - w_0) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

x^j — признаки объекта, $j = 1, \ldots, n$

$w_j \in R$ – веса признаков

$\sigma(s)$ – функция активации (например, sign)

Модель McCulloch-Pitts

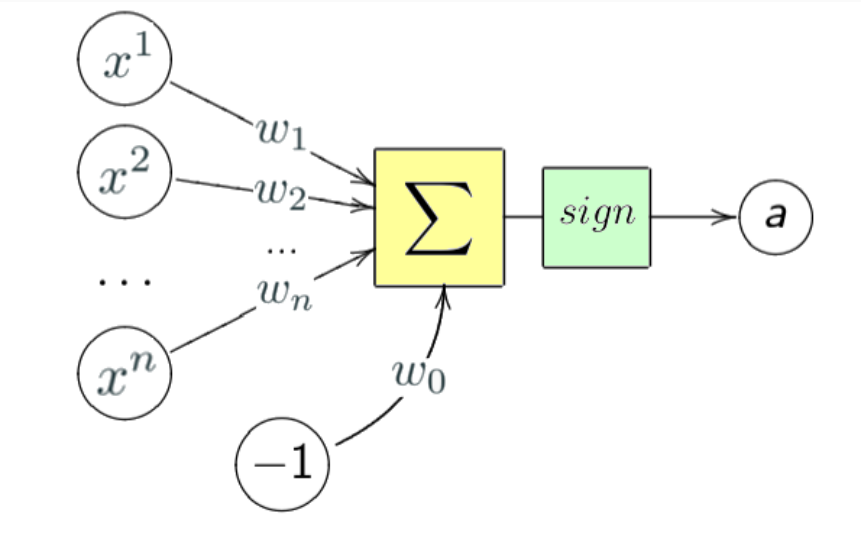
$X = \mathbb{R}^n$, $Y = \{-1, +1\}$

$$a(x, w) = \sigma(\sum_{j=1}^{n} w_j x^j - w_0) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

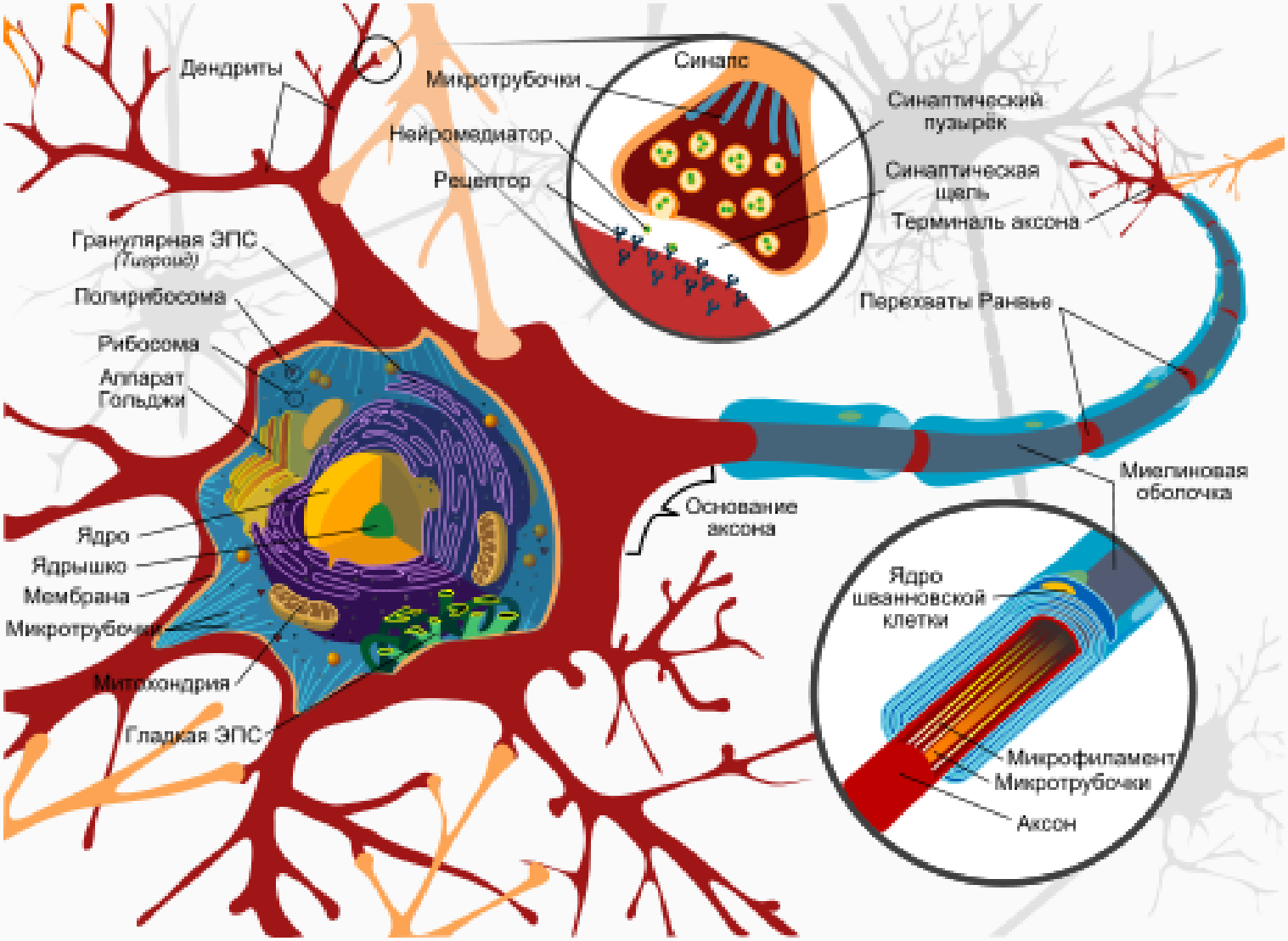
x^j — признаки объекта, $j = 1, \ldots, n$

$w_j \in R$ – веса признаков

$\sigma(s)$ – функция активации (например, sign)



Нейрон


Дендриты
Микротрубочки
Синапс
Нейромедиатор
Рецептор
Синаптический пузырёк
Синаптическая щель
Терминаль аксона
Гранулярная ЭПС
(Тигроид)
Полирибосома
Рибосома
Аппарат Гольджи
Перехваты Ранвье
Миелиновая оболочка
Основание аксона
Ядро
Ядрышко
Мембрана
Микротрубочки
Митохондрия
Гладкая ЭПС
Ядро шванновской клетки
Микрофиламент
Микротрубочки
Аксон

Линейные алгоритмы классификации и регрессии

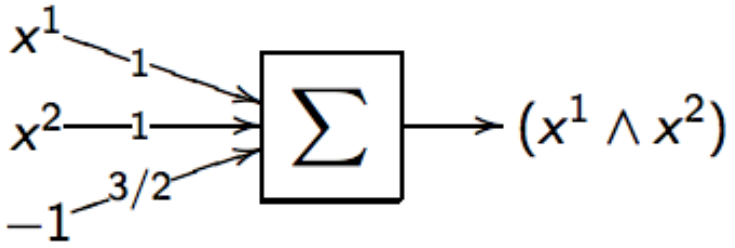
Задача классификации:

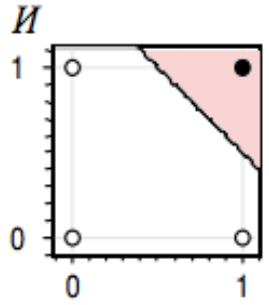
$$a(\mathbf{x}, \mathbf{w}) = \text{sign}\langle \mathbf{w}, \mathbf{x} \rangle$$

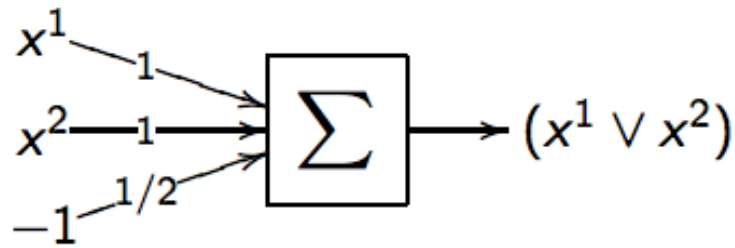
$$Q(w; X^l) = \sum_{i=1}^{l} \mathcal{L}(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i) \to \min_w$$

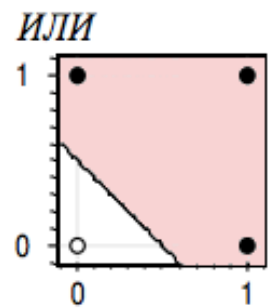
Какой класс функций можно реализовать нейроном?

Нейронная реализация логических функций









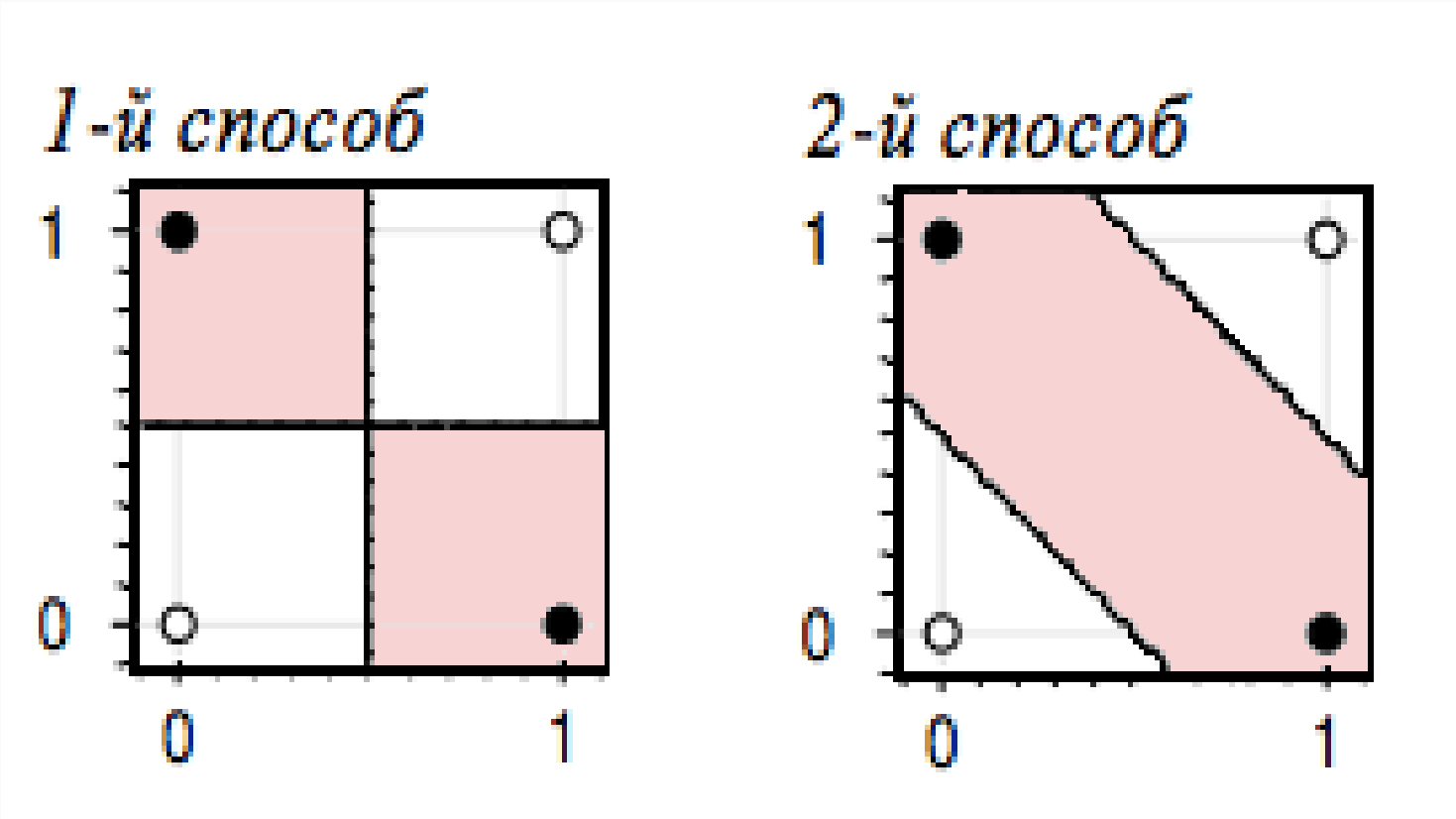
Нейронная реализация логических функций

Функции И, ИЛИ от бинарных переменных x_1 и x_2:

$$x_1 \wedge x_2 = x_1 + x_2 - \frac{3}{2} > 0$$

$$x_1 \vee x_2 = x_1 + x_2 - \frac{1}{2} > 0$$

Логическая функция XOR



Логическая функция XOR

Функция $x_1 \oplus x_2 = [x_1 \neq x_2]$ не реализуема одним нейроном.

Логическая функция XOR

Функция $x_1 \oplus x_2 = [x_1 \neq x_2]$ не реализуема одним нейроном.

Два способа реализации:

1. Добавлением нелинейного признака:

$$x_1 \oplus x_2 = [x_1 + x_2 + 2x_1x_2 - \frac{1}{2} > 0]$$

Логическая функция XOR

Функция $x_1 \oplus x_2 = [x_1 \neq x_2]$ не реализуема одним нейроном.

Два способа реализации:

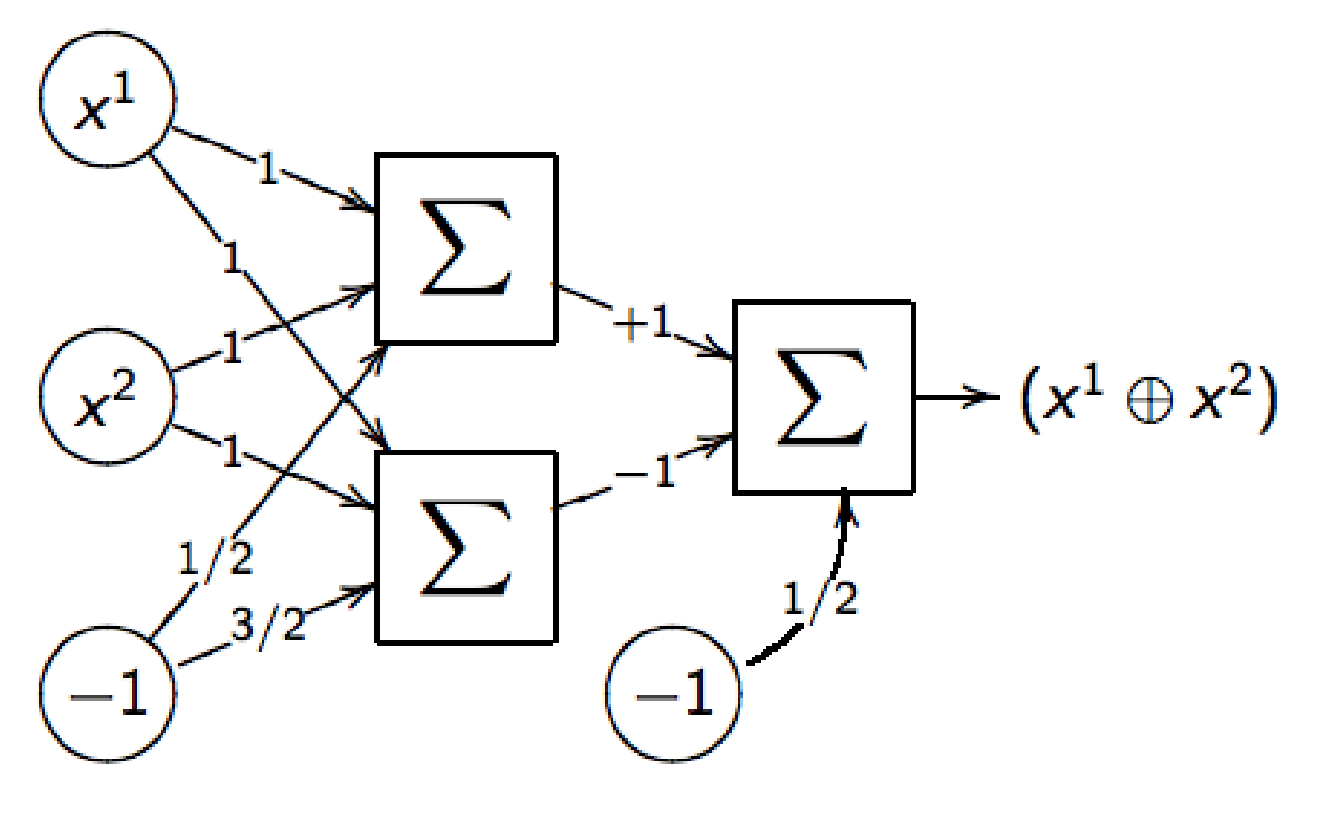
1. Добавлением нелинейного признака:

$$x_1 \oplus x_2 = [x_1 + x_2 + 2x_1x_2 - \frac{1}{2} > 0]$$

2. Сетью (двухслойной суперпозицией) функций И, ИЛИ:

$$x_1 \oplus x_2 = [(x_1 \vee x_2) - (x_1 \wedge x_2) - \frac{1}{2} > 0]$$

Логическая функция XOR



Любую ли функцию можно представить нейросетью?

- Двухслойная сеть в $\{0,1\}^n$ позволяет реализовать произвольную булеву функцию.

Любую ли функцию можно представить нейросетью?

- Двухслойная сеть в $\{0,1\}^n$ позволяет реализовать произвольную булеву функцию.
- Двухслойная сеть в $\mathbb{R}^n$ позволяет отделить произвольный выпуклый многогранник.

Любую ли функцию можно представить нейросетью?

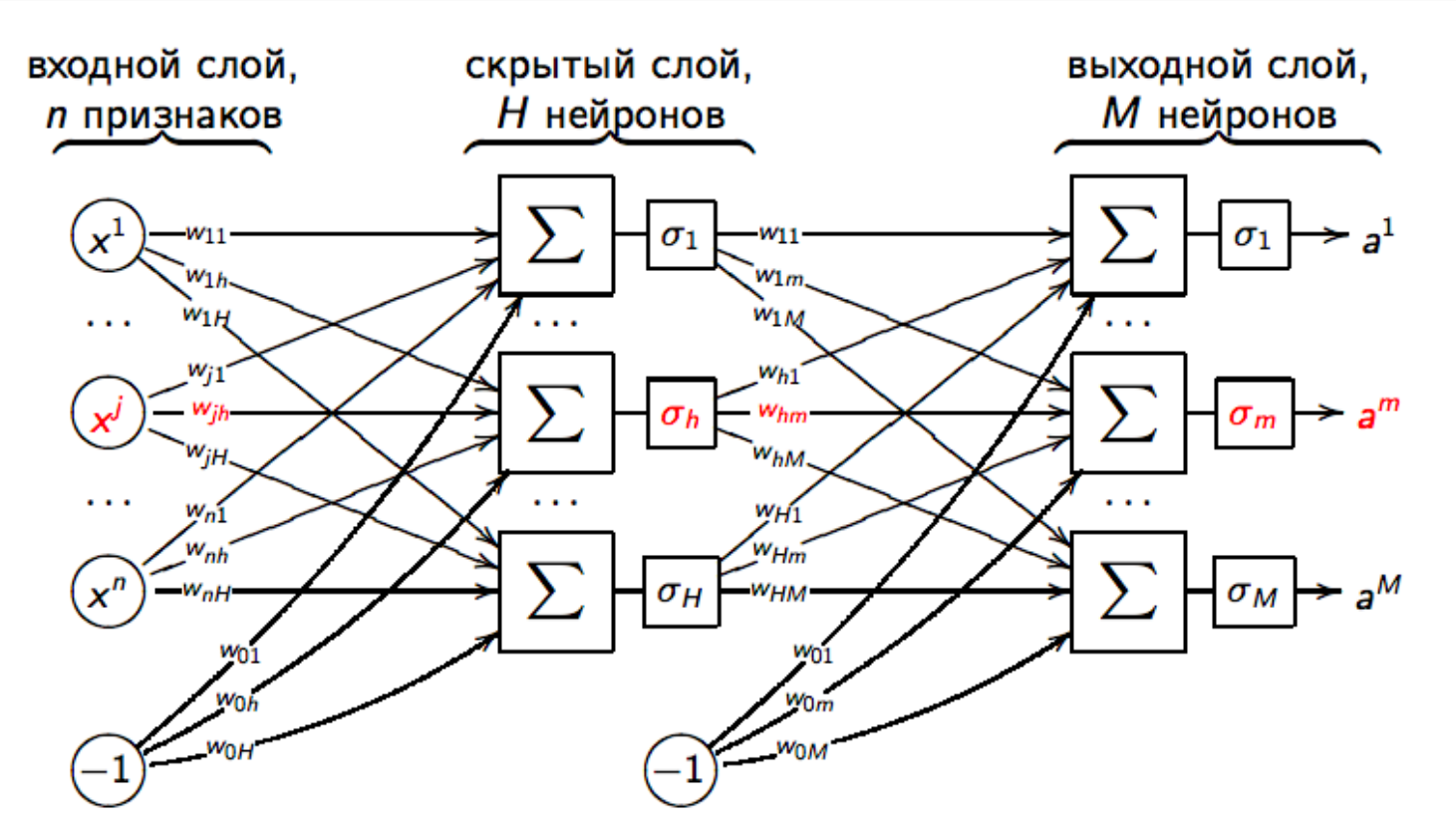
- Двухслойная сеть в $\{0,1\}^n$ позволяет реализовать произвольную булеву функцию.
- Двухслойная сеть в $\mathbb{R}^n$ позволяет отделить произвольный выпуклый многогранник.
- Трёхслойная сеть $\mathbb{R}^n$ позволяет отделить произвольную многогранную область, не обязательно выпуклую.

Любую ли функцию можно представить нейросетью?

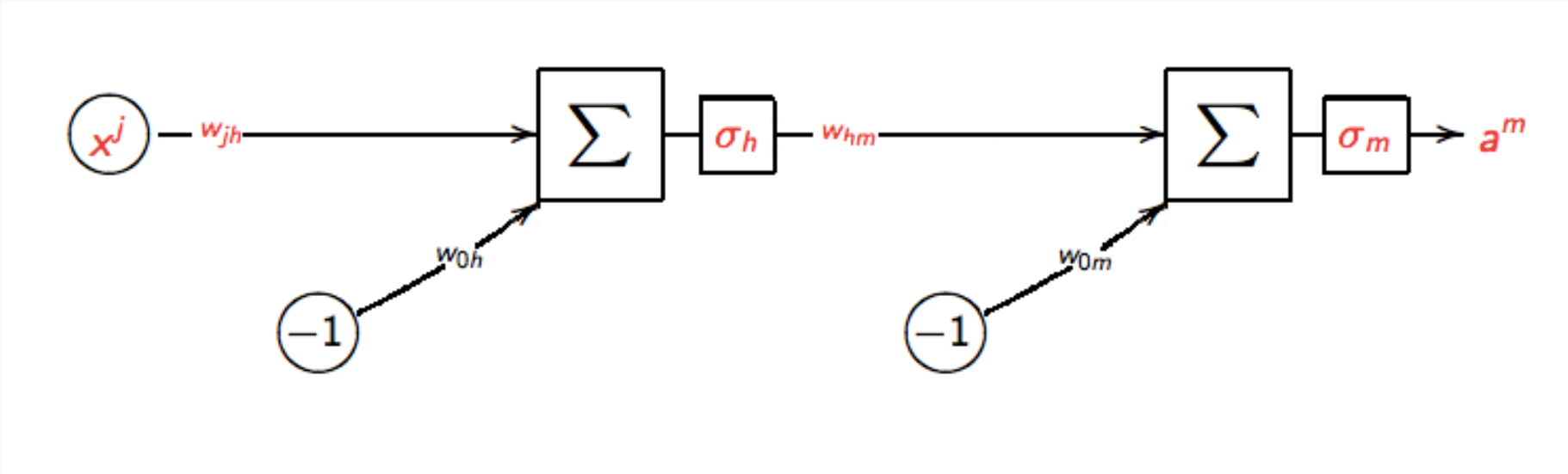
- Двухслойная сеть в $\{0,1\}^n$ позволяет реализовать произвольную булеву функцию.
- Двухслойная сеть в $\mathbb{R}^n$ позволяет отделить произвольный выпуклый многогранник.
- Трёхслойная сеть $\mathbb{R}^n$ позволяет отделить произвольную многогранную область, не обязательно выпуклую.
- С помощью линейных операций и одной нелинейной функции активации σ можно приблизить любую непрерывную функцию с любой желаемой точностью

Многослойная нейронная сеть

Пусть $Y = \mathbb{R}^M$, два слоя в сети.



Многослойная нейронная сеть



Стохастический градиент

$$Q(\mathbf{w}) = \sum_{i=1}^{l} \mathcal{L}(w, x_i, y_i) \to \min_w$$

1 **function** STOCHASTIC_GRADIENT(X^l, α, η)
2 Перемешать данные в X^l
3 Инициализировать $\mathbf{w}$
4 $Q(\mathbf{w}) = \sum_{i=1}^{l} \mathcal{L}(\langle \mathbf{w}, \mathbf{x_i} \rangle y_i)$
5 **repeat**[пока Q и/или w не стабилизируются]
6 Взять x_i из X^l
7 Потеря: $\varepsilon_i = \mathcal{L}(w, x_i, y_i)$
8 Градиентный шаг: $w = w - \alpha \nabla \mathcal{L}(w, x_i, y_i)$
9 Оценить $Q = (1 - \eta)Q + \eta\varepsilon_i$

Сколько операций потребуется для вычисления градиента в точке?

Идея

Идея: Сохранять некоторые промежуточные результаты в узлах сети.

Задача дифференцирования суперпозиции функций

Выходные значения $a^m(x_i)$, $m = 1, \ldots, M$ на объекте x_i:

$$a^m(x_i) = \sigma_m\Big(\sum_{h=0}^{H} w_{hm} u^h(x_i)\Big), \qquad u^h(x_i) = \sigma_h\Big(\sum_{j=0}^{n} w_{jh} x_i^j\Big)$$

Задача дифференцирования суперпозиции функций

Выходные значения $a^m(x_i)$, $m = 1, \ldots, M$ на объекте x_i:

$$a^m(x_i) = \sigma_m\Big(\sum_{h=0}^{H} w_{hm} u^h(x_i)\Big), \qquad u^h(x_i) = \sigma_h\Big(\sum_{j=0}^{n} w_{jh} x_i^j\Big)$$

Возьмем $\mathcal{L}_i(w)$ – средний квадрат ошибки:

$$\mathcal{L}_i(w) = \frac{1}{2}\sum_{m=1}^{M}(a^m(x_i) - y_i^m)^2$$

Задача дифференцирования суперпозиции функций

Выходные значения $a^m(x_i)$, $m = 1, \ldots, M$ на объекте x_i:

$$a^m(x_i) = \sigma_m\Big(\sum_{h=0}^{H} w_{hm} u^h(x_i)\Big), \qquad u^h(x_i) = \sigma_h\Big(\sum_{j=0}^{n} w_{jh} x_i^j\Big)$$

Возьмем $\mathcal{L}_i(w)$ – средний квадрат ошибки:

$$\mathcal{L}_i(w) = \frac{1}{2}\sum_{m=1}^{M}(a^m(x_i) - y_i^m)^2$$

Промежуточная задача: найти частные производные

$$\frac{\partial \mathcal{L}_i(w)}{\partial a^m} \qquad \frac{\partial \mathcal{L}_i(w)}{\partial u^h}$$

Быстрое вычисление градиента

Ошибка на выходном слое:

$$\frac{\partial \mathcal{L}_i(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m$$

Быстрое вычисление градиента

Ошибка на выходном слое:

$$\frac{\partial \mathcal{L}_i(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m$$

Ошибка на скрытом слое:

$$\frac{\partial \mathcal{L}_i(w)}{\partial u^h} = \sum_{m=1}^{M} (a^m(x_i) - y_i^m)\sigma'_m w_{hm} = \sum_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm} = \varepsilon_i^h$$

Быстрое вычисление градиента

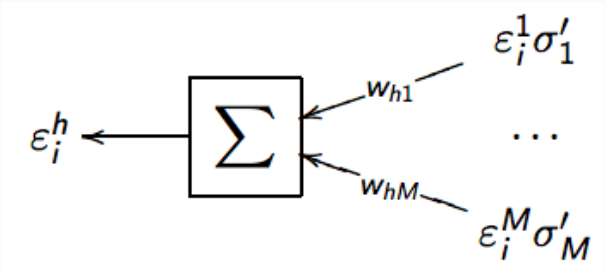
Ошибка на выходном слое:

$$\frac{\partial \mathcal{L}_i(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m$$

Ошибка на скрытом слое:

$$\frac{\partial \mathcal{L}_i(w)}{\partial u^h} = \sum_{m=1}^{M} (a^m(x_i) - y_i^m)\sigma'_m w_{hm} = \sum_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm} = \varepsilon_i^h$$

ε_i^h вычисляется по ε_i^m, если запустить сеть «задом наперёд»:



Быстрое вычисление градиента

Теперь, имея частные производные $\mathcal{L}_i(w)$ по a^m и u^h, легко выписать градиент $\mathcal{L}_i(w)$ по весам w.

Быстрое вычисление градиента

Теперь, имея частные производные $\mathcal{L}_i(w)$ по a^m и u^h, легко выписать градиент $\mathcal{L}_i(w)$ по весам w.

Вопрос: как?

Быстрое вычисление градиента

$$\frac{\partial \mathcal{L}_i(w)}{\partial w_{hm}} = \frac{\partial \mathcal{L}_i(w)}{\partial a^m} \frac{\partial a^m}{\partial w_{hm}} = \varepsilon_i^m \sigma'_m u^h(x_i),$$

$$m = 1, \ldots, M, \quad h = 0, \ldots, H$$

Быстрое вычисление градиента

$$\frac{\partial \mathcal{L}_i(w)}{\partial w_{hm}} = \frac{\partial \mathcal{L}_i(w)}{\partial a^m} \frac{\partial a^m}{\partial w_{hm}} = \varepsilon_i^m \sigma'_m u^h(x_i),$$

$$m = 1, \ldots, M, \quad h = 0, \ldots, H$$

$$\frac{\partial \mathcal{L}_i(w)}{\partial w_{jh}} = \frac{\partial \mathcal{L}_i(w)}{\partial u^h} \frac{\partial u^h}{\partial w_{jh}} = \varepsilon_i^h \sigma'_h x_i^j,$$

$$h = 1, \ldots, H, \quad j = 0, \ldots, n$$

Алгоритм обратного распространения ошибки

```
1 function BACKPROPAGATION(X^l, H, α, η)
2     ...
3     repeat[пока Q не стабилизируются]
4         Взять x_i из X^l
```

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

$$\begin{cases} u_i^h = \sigma_h(\sum\limits_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \ldots, H \\ a_i^m = \sigma_m(\sum\limits_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \ldots, M \\ \mathcal{L}_i = \frac{1}{2} \sum\limits_{m=1}^{M} (\varepsilon_i^m)^2 \end{cases}$$

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

$$\begin{cases} u_i^h = \sigma_h(\sum_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \ldots, H \\ a_i^m = \sigma_m(\sum_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \ldots, M \\ \mathcal{L}_i = \frac{1}{2} \sum_{m=1}^{M} (\varepsilon_i^m)^2 \end{cases}$$

6

$$\begin{cases} \varepsilon_i^h = \sum_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \ldots, H \end{cases}$$

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

$$\begin{cases} u_i^h = \sigma_h(\sum\limits_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \ldots, H \\ a_i^m = \sigma_m(\sum\limits_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \ldots, M \\ \mathcal{L}_i = \frac{1}{2} \sum\limits_{m=1}^{M} (\varepsilon_i^m)^2 \end{cases}$$

6

$$\begin{cases} \varepsilon_i^h = \sum\limits_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \ldots, H \end{cases}$$

7

$$\begin{cases} w_{hm} = w_{hm} - \alpha \varepsilon_i^m \sigma'_m u_i^h, \quad h = 0, \ldots, H, \quad m = 1, \ldots, M \\ w_{jh} = w_{jh} - \alpha \varepsilon_i^h \sigma'_h x_i^j, \quad j = 0, \ldots, n, \quad h = 1, \ldots, H \\ Q = (1 - \eta) Q + \eta \mathcal{L}_i \end{cases}$$

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i
+ Возможность распараллеливания

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i
+ Возможность распараллеливания

– Возможна медленная сходимость

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i
+ Возможность распараллеливания

– Возможна медленная сходимость
– Застревание в локальных минимумах

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathcal{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i
+ Возможность распараллеливания

– Возможна медленная сходимость
– Застревание в локальных минимумах
– Проблема переобучения

Особенности

+ Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
+ Легко обобщается на любые σ, $\mathscr{L}$
+ Возможно динамическое (потоковое) обучение
+ На сверхбольших выборках не обязательно брать все x_i
+ Возможность распараллеливания

– Возможна медленная сходимость
– Застревание в локальных минимумах
– Проблема переобучения
– Подбор комплекса эвристик

Стандартные эвристики для метода SG

Применимы все те же эвристики, что и в обычном SG.

Стандартные эвристики для метода SG

Применимы все те же эвристики, что и в обычном SG.

Напомните.

Стандартные эвристики для метода SG

- Инициализация весов

Стандартные эвристики для метода SG

- Инициализация весов
- Порядок предъявления объектов

Стандартные эвристики для метода SG

- Инициализация весов
- Порядок предъявления объектов
- Оптимизация величины градиентного шага

Стандартные эвристики для метода SG

- Инициализация весов
- Порядок предъявления объектов
- Оптимизация величины градиентного шага
- Регуляризация (сокращение весов)

Новые проблемы

- Выбор функций активации в каждом нейроне

Новые проблемы

- Выбор функций активации в каждом нейроне
- Выбор числа слоёв и числа нейронов

Примеры функций активации

- Сигмоида

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

Примеры функций активации

- Сигмоида

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

- Гиперболический тангенс

$$\tanh(x) = 2\sigma(2x) - 1$$

Примеры функций активации

- Сигмоида

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

- Гиперболический тангенс

$$\tanh(x) = 2\sigma(2x) - 1$$

- Rectified Linear Unit

$$f(x) = \max(0, x)$$

Практическое применение

Специальные нейронные сети

Почему их снова стали использовать?

Специальные нейронные сети

Почему их снова стали использовать?

- Большие вычислительные возможности (в том числе распределенные вычисления)
- Большие объемы данных
- Новые методы предобучения, новые архитектуры

Свёрточные сети

Свёрточные сети

1. Пиксель изображения сильнее связан с соседними пикселями (локальная корреляция)
2. Объект может встретиться в любой части изображения (инвариантность к перемещению)

http://www.cs.nyu.edu/ yann/talks/lecun-ranzato-icml2013.pdf

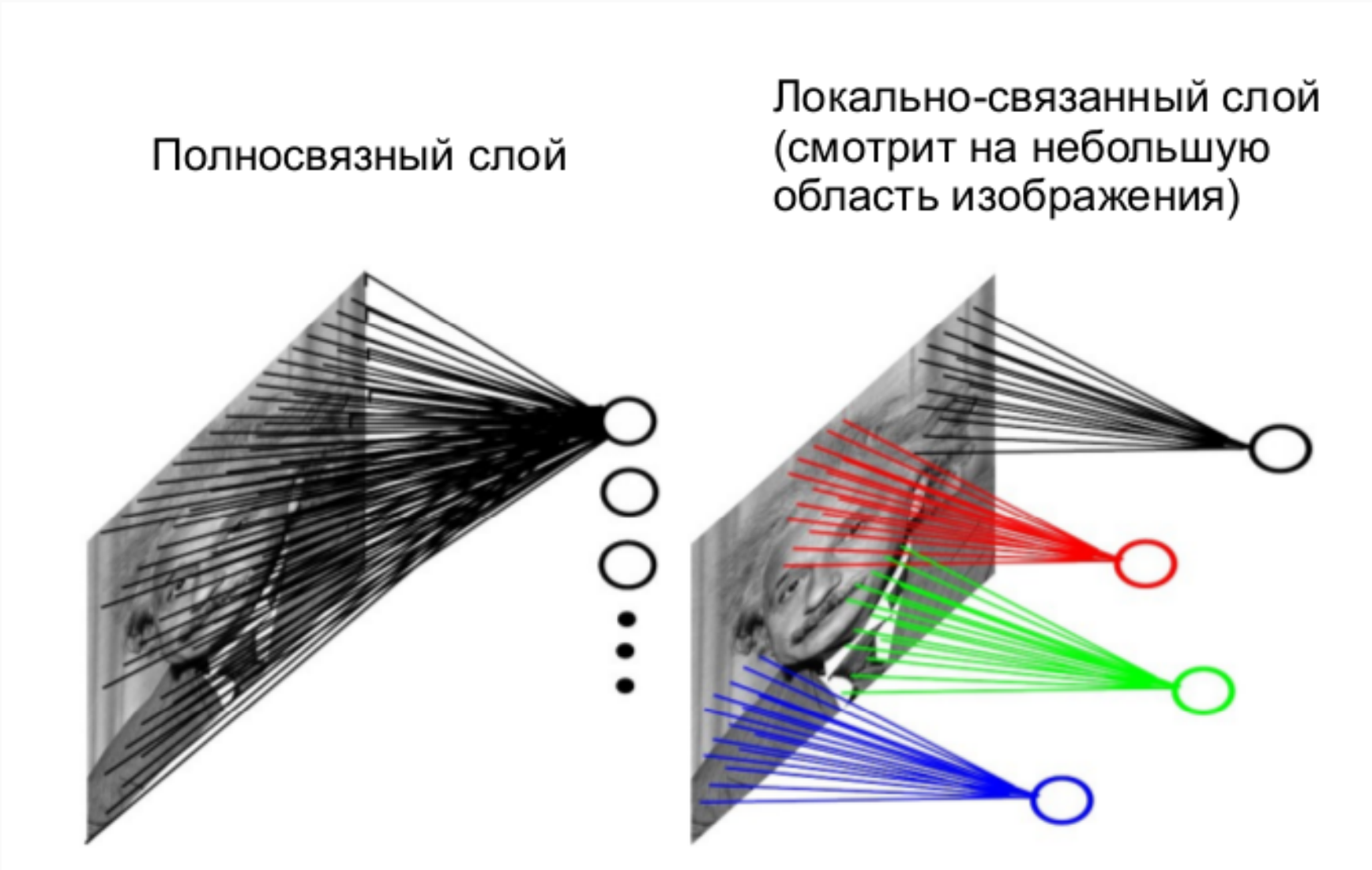
Свёрточные сети

1. Пиксель изображения сильнее связан с соседними пикселями (локальная корреляция)
2. Объект может встретиться в любой части изображения (инвариантность к перемещению)

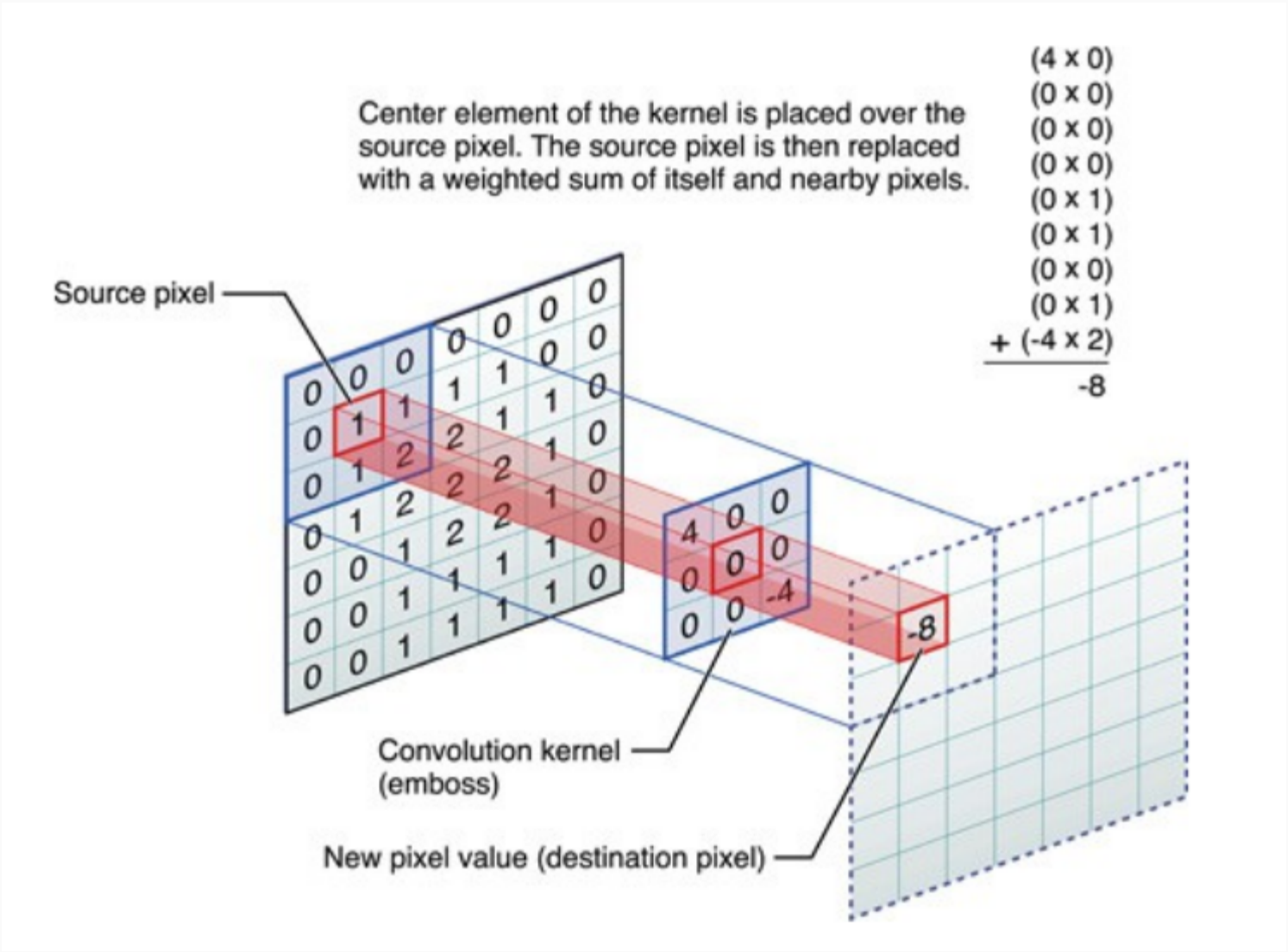
1. Каждый из нейронов подсоединен только к небольшой окрестности изображения
2. Нейроны обладают одними и теми же весами

http://www.cs.nyu.edu/ yann/talks/lecun-ranzato-icml2013.pdf

Локально-связанный слой



Операция свертки



Простейшая свертка

Original

0	0	0
0	1	0
0	0	0

?

Простейшая свертка

Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Простейшая свертка

Original

0	0	0
0	0	1
0	0	0

?

Простейшая свертка

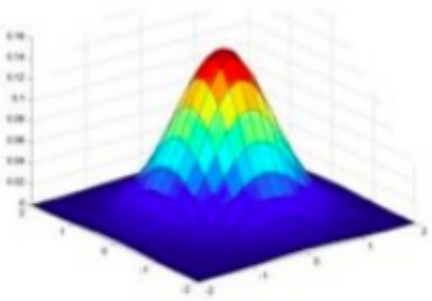
Original

0	0	0
0	0	1
0	0	0

Shifted left
By 1 pixel

Фильтр Гаусса

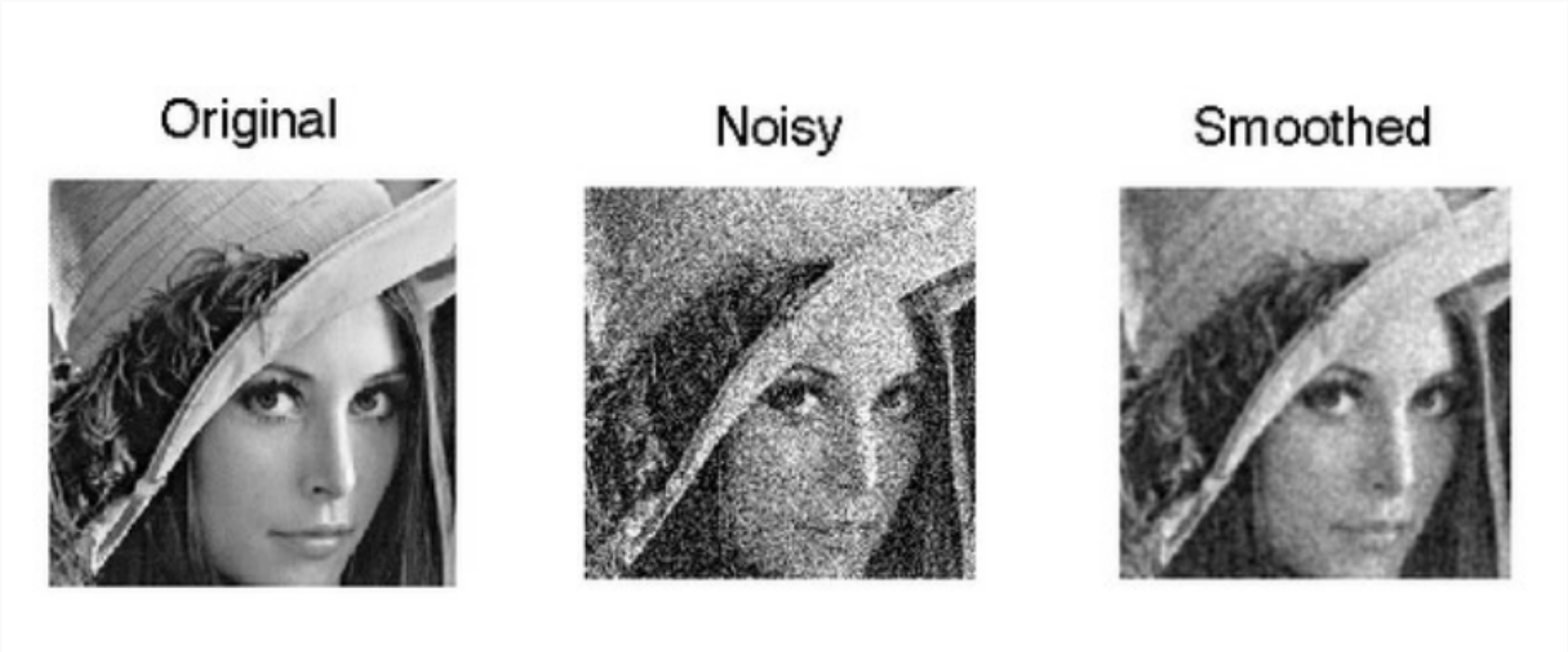
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

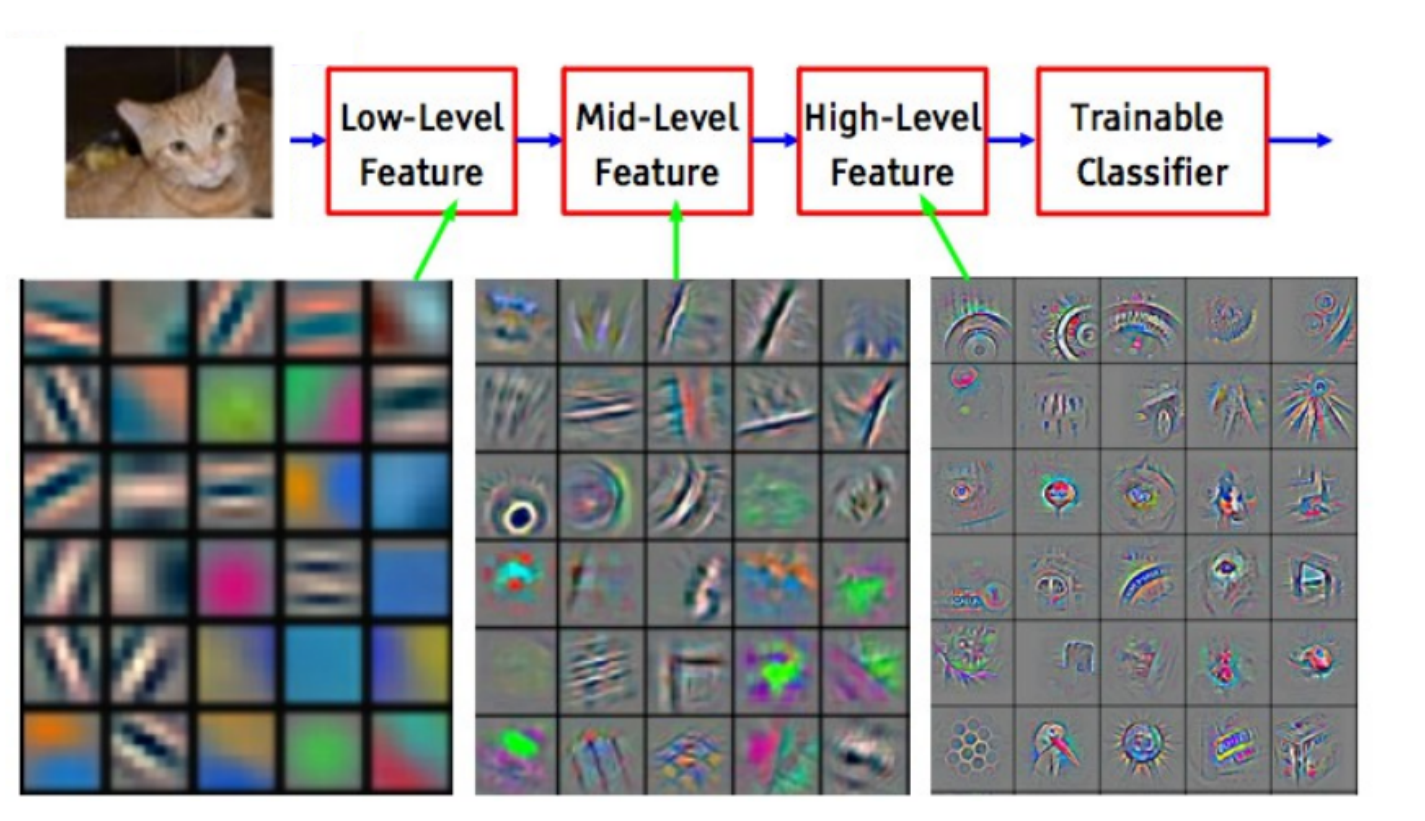
5 x 5, $\sigma = 1$

Подавление шумов

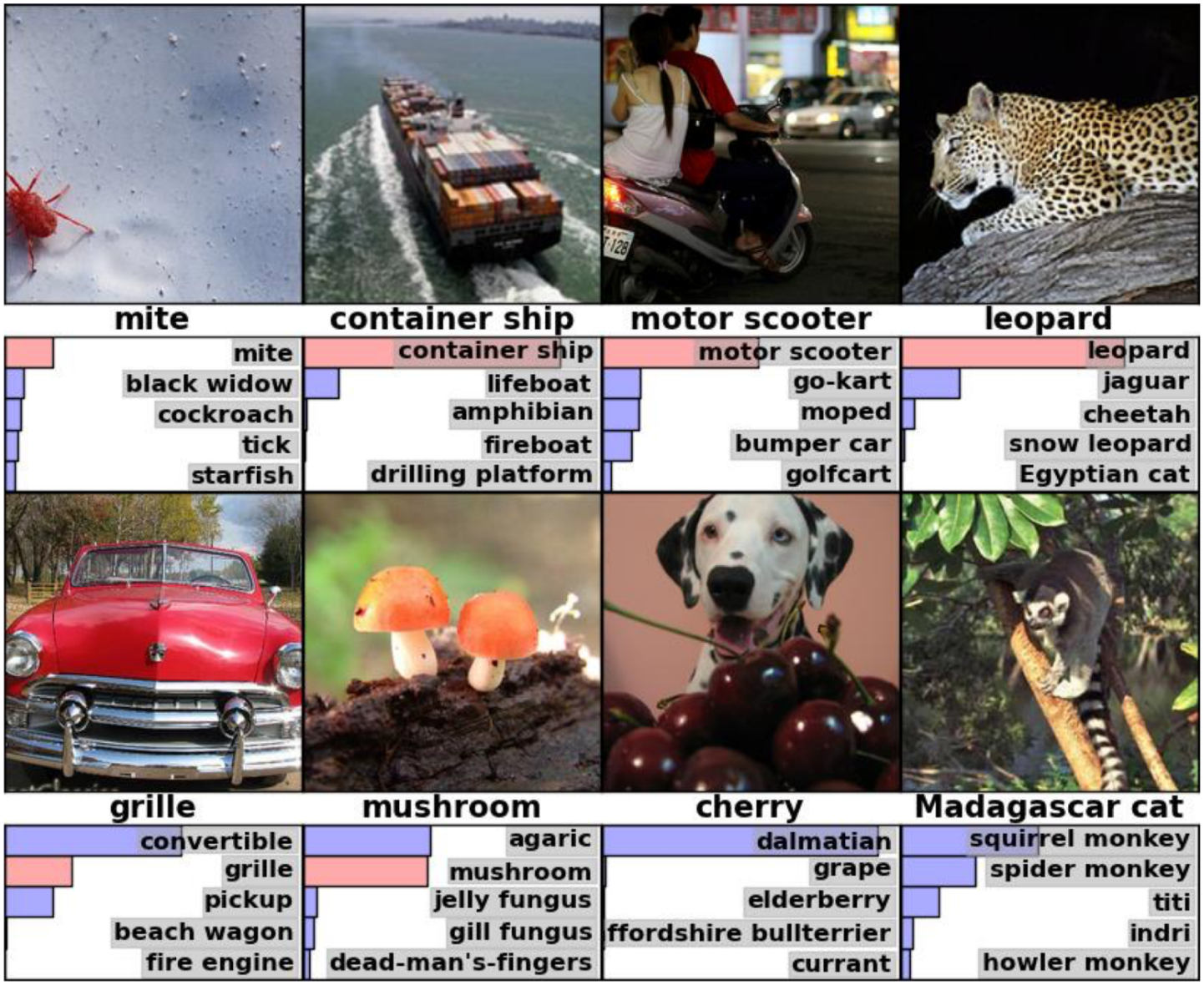


Свёрточная сеть

Извлечение признаков во время обучения.

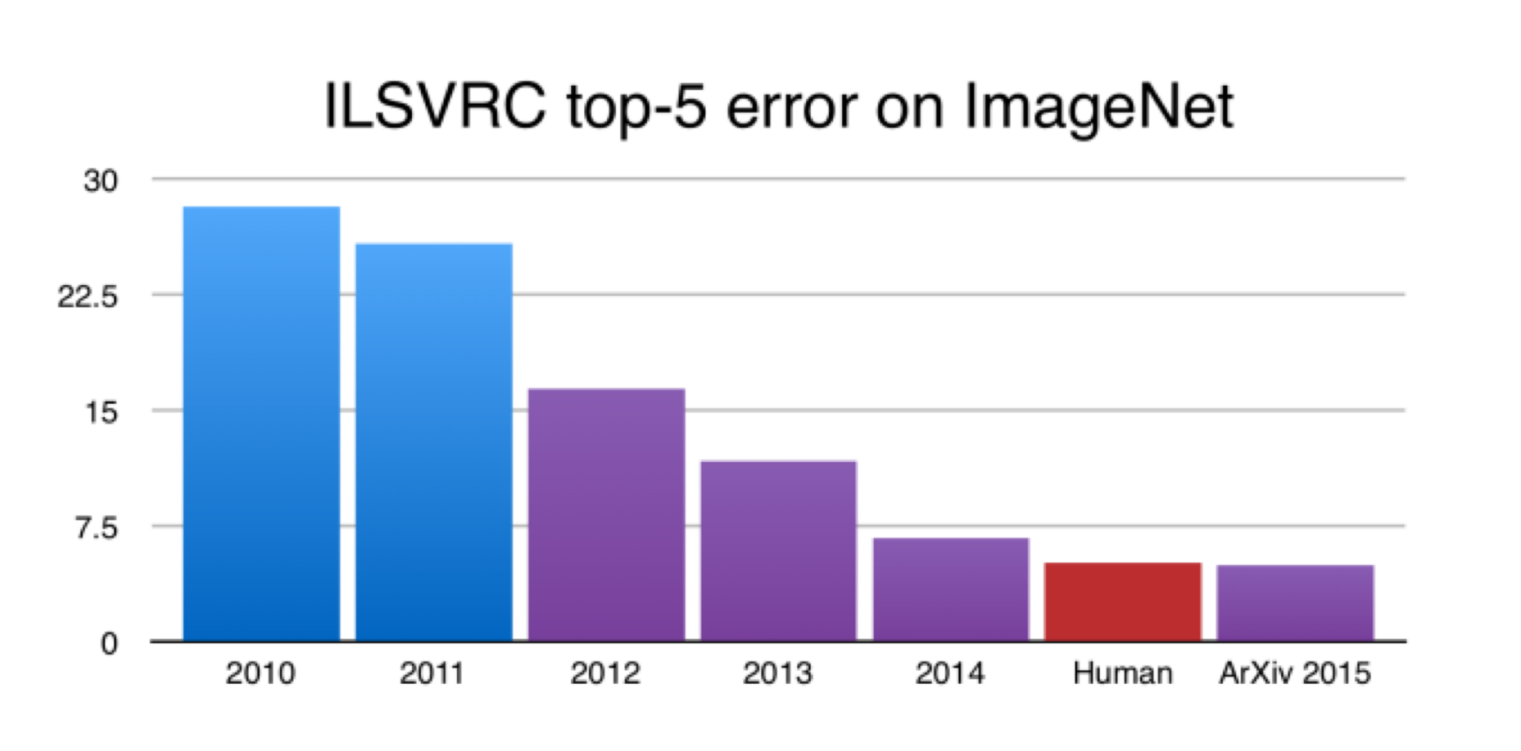


ImageNet



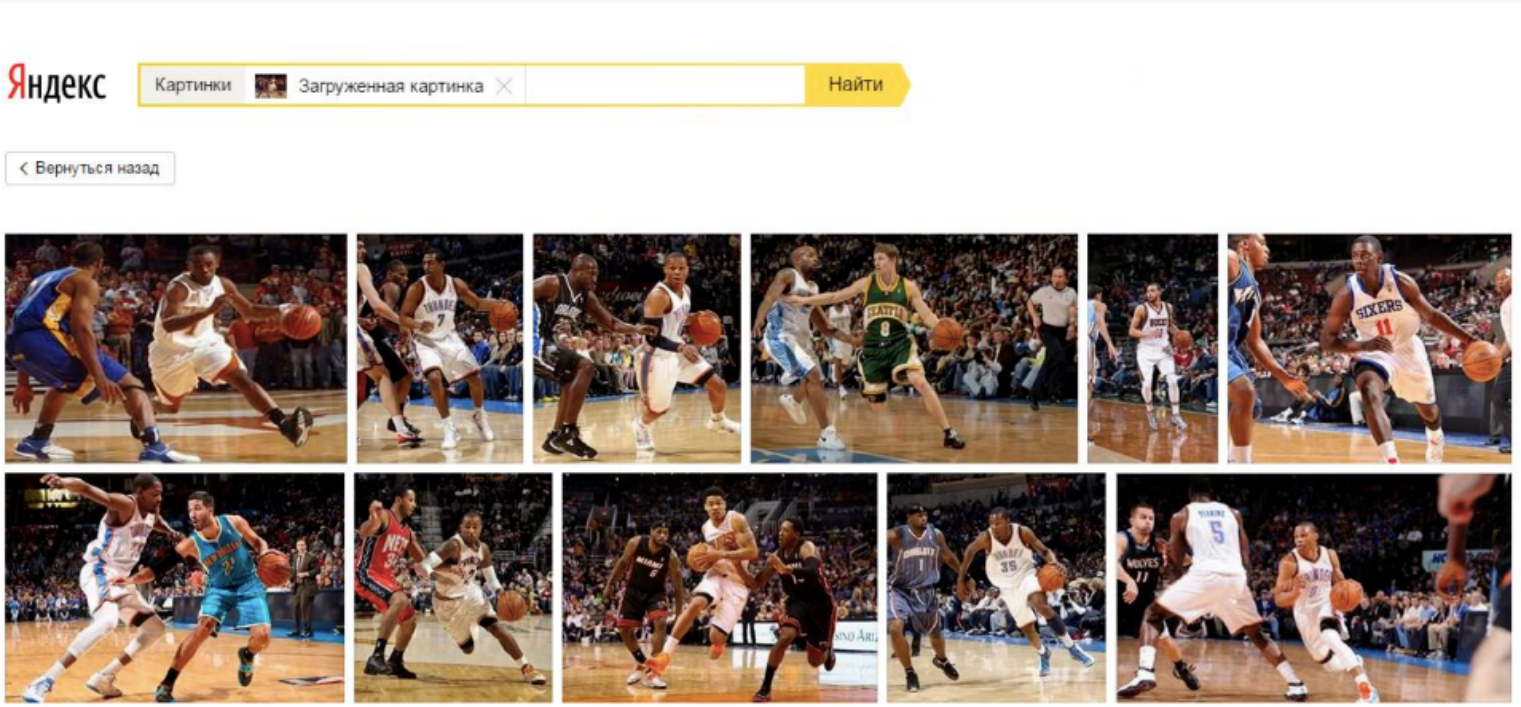
https://www.cs.toronto.edu/ fritz/absps/imagenet.pdf

ImageNet


ILSVRC top-5 error on ImageNet
30
22.5
15
7.5
0
2010
2011
2012
2013
2014
Human
ArXiv 2015

Похожие изображения

Похожие изображения

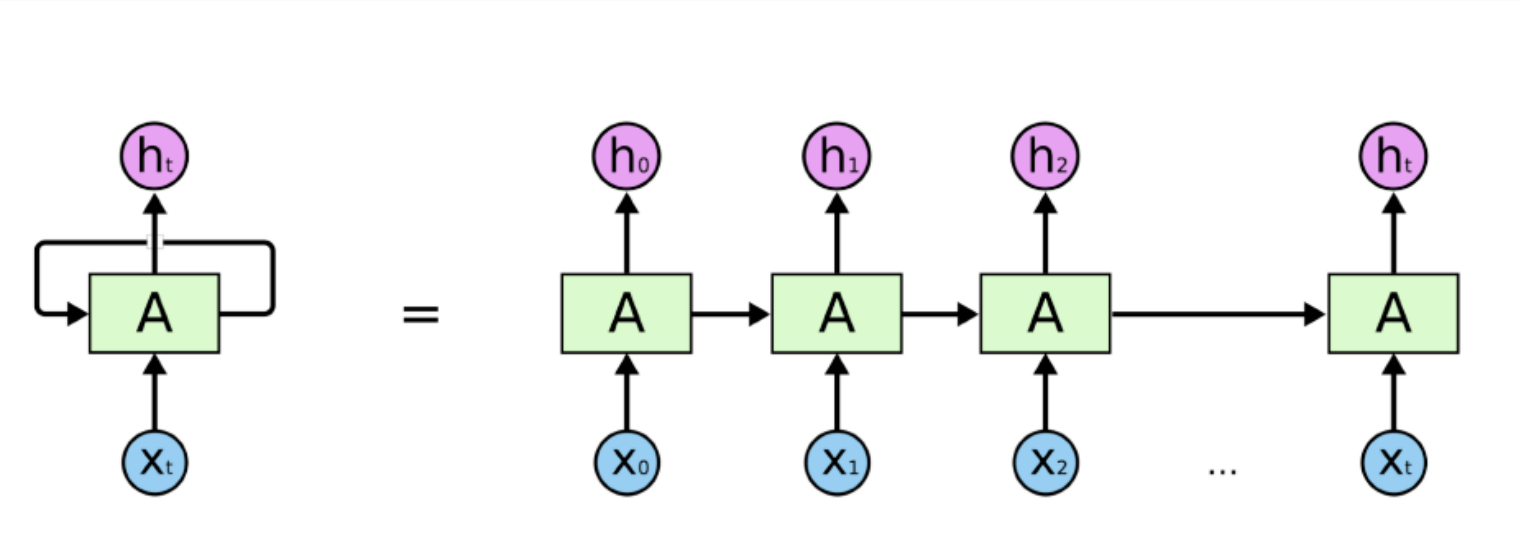


Стилизация изображений

https://arxiv.org/pdf/1508.06576v1.pdf

Рекуррентные нейронные сети

RNN



Пример

Текст состоит из последовательности слов. Хотелось бы подавать слова на вход нейросети по очереди, но так, чтобы сеть помнила контекст.

Шекспир

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

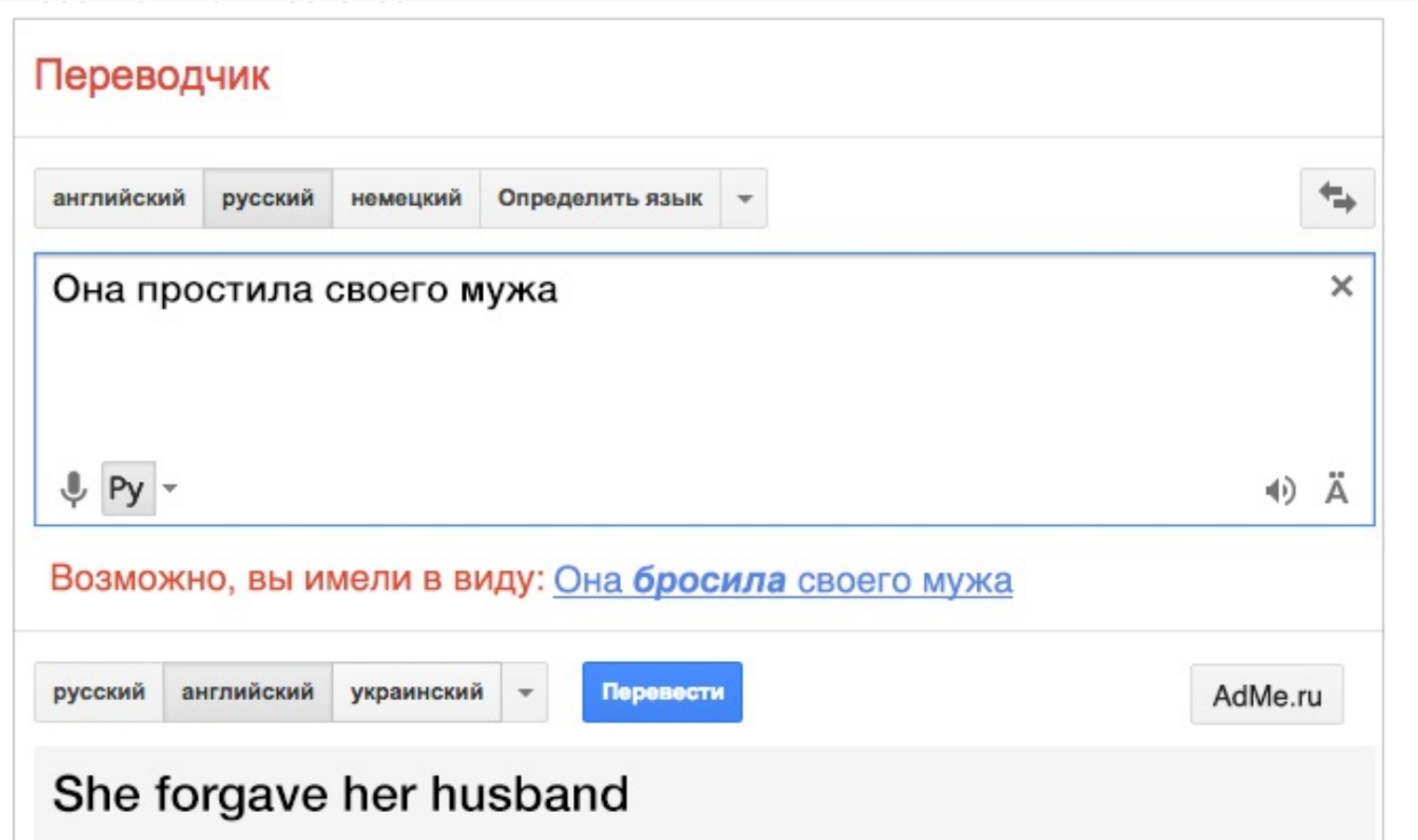
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Генерация кода

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
```

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Автоматический перевод



RNN + CNN

Подпись к изображениям

Later on the eighth day , Billy was a friend of a man who lived on his own . He did n't know how far away they were , and if he was to survive the fall . His mind raced , trying not to show any signs of weakness .

The wind ruffled the snow and ice in the snow . He had no idea how many times he was going to climb into the mountains

Подпись к изображениям

A yellow school bus parked in a parking lot.

A red school bus parked in a parking lot.

A green school bus parked in a parking lot.

A blue school bus parked in a parking lot.

The decadent chocolate desert is on the table.

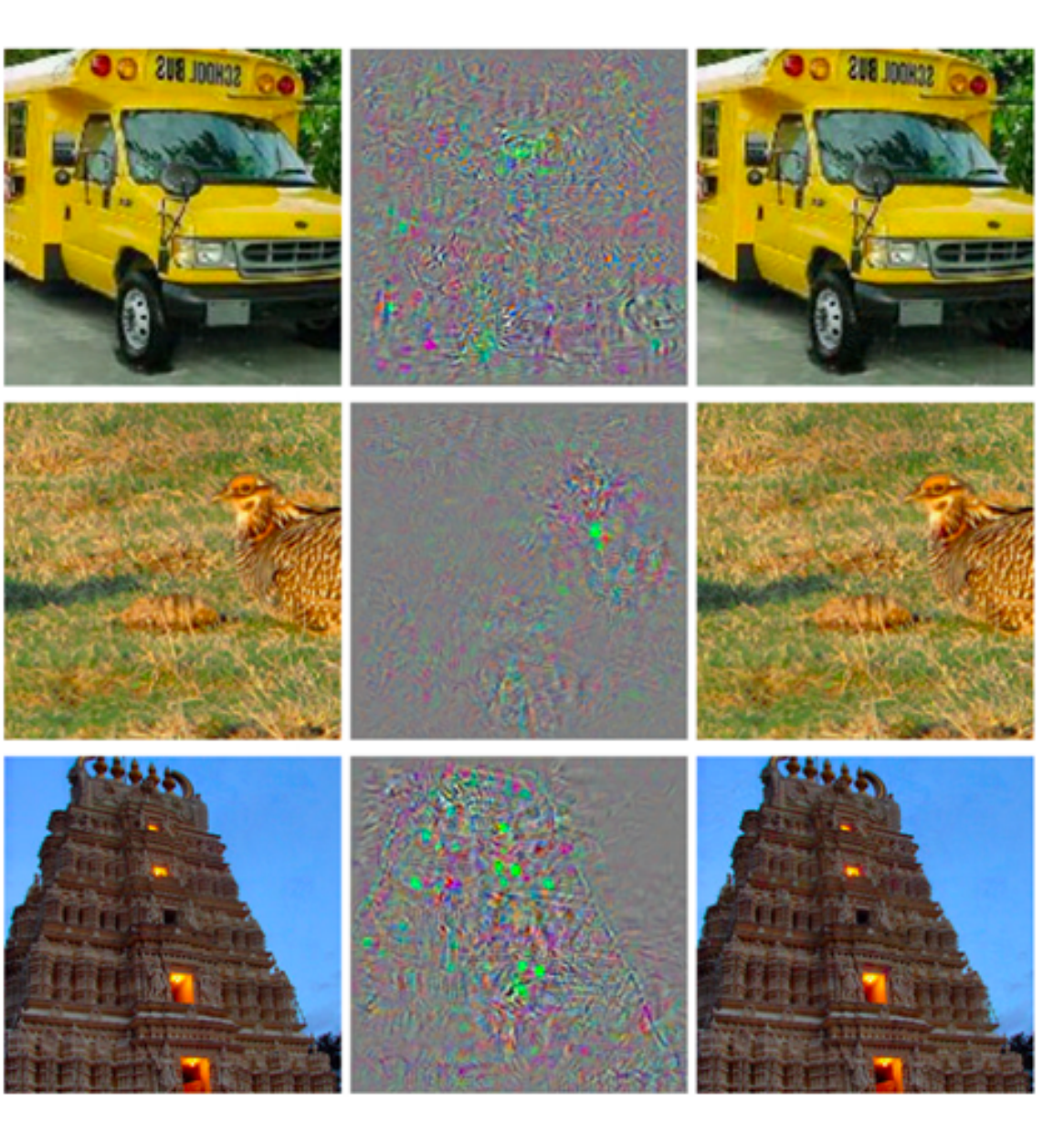
A bowl of bananas is on the table.

A vintage photo of a cat.

A vintage photo of a dog.

Generative Adversarial Networks

Adversarial Networks



https://arxiv.org/pdf/1701.00160.pdf

Вопросы?

Что почитать по этой лекции

- Andrej Karpathy lecture notes (cs231n)
- Tom Mitchell "Machine Learning" Chapter 4.5

На следующей лекции

- Задача максимизации зазора - аналог классификатора с регуляризацией
- Двойственная задача
- Что такое опорный вектор
- Регуляризация
- Решение для неразделимых выборок
- Kernel trick