

# Разбор летучки

---

# Лекция 8

## Нейронные сети

---

Екатерина Тузова

# Модель McCulloch-Pitts

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

# Модель McCulloch-Pitts

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

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

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

$w_j \in R$  — веса признаков

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



# Модель McCulloch-Pitts

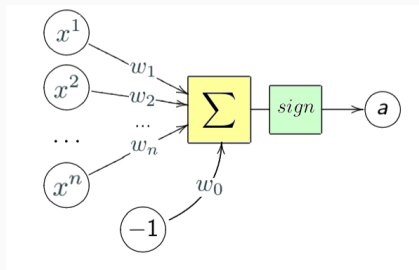
$$X = \mathbb{R}^n, Y = \{-1, +1\}$$

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

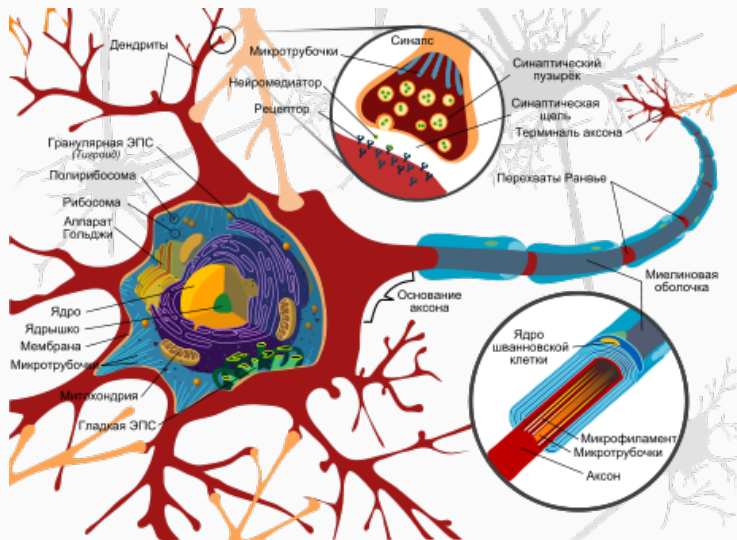
$x^j$  — признаки объекта,  $j = 1, \dots, n$

$w_j \in R$  — веса признаков

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



# Нейрон



Задача классификации:

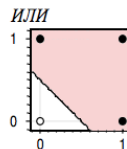
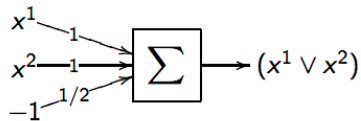
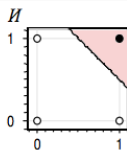
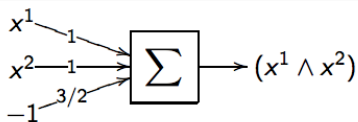
$$a(\mathbf{x}, \mathbf{w}) = \text{sign}\langle \mathbf{w}, \mathbf{x} \rangle$$

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

Какой класс функций можно реализовать нейроном?

---

# Нейронная реализация логических функций



# Нейронная реализация логических функций

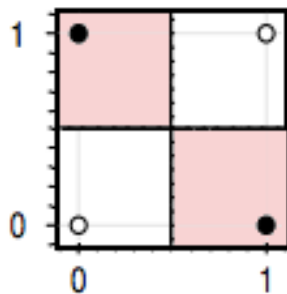
Функции И, ИЛИ от бинарных переменных  $x_1$  и  $x_2$ :

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

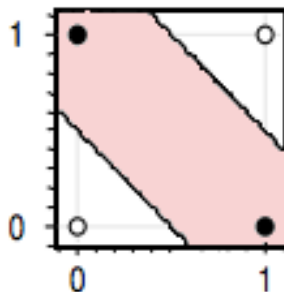
$$x_1 \vee x_2 = x_1 + x_2 - \frac{1}{2} > 0$$

# Логическая функция XOR

*1-й способ*



*2-й способ*



# 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:

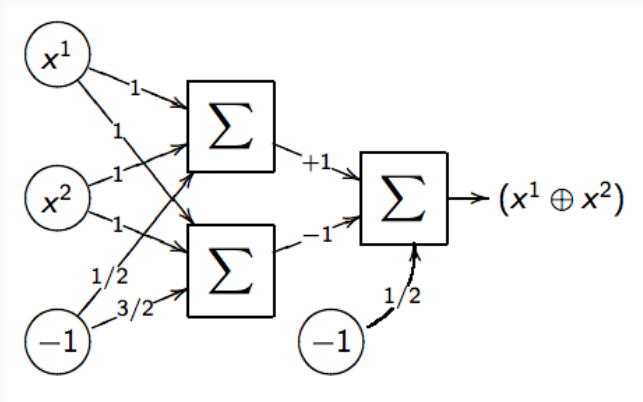
1. Добавлением нелинейного признака:

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

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?

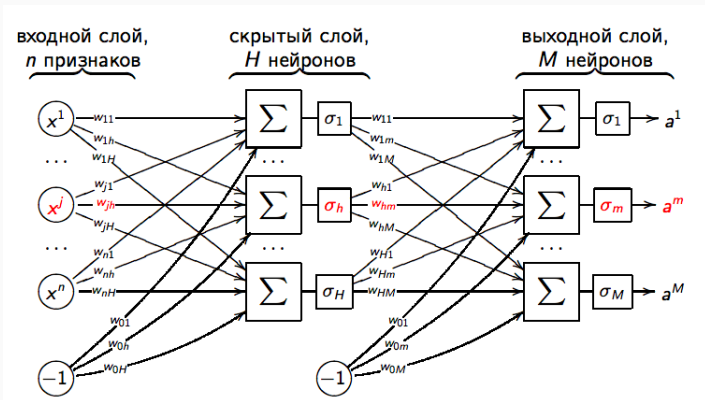
- Двухслойная сеть в  $\{0, 1\}^n$  позволяет реализовать произвольную булеву функцию.
- Двухслойная сеть в  $\mathbb{R}^n$  позволяет отделить произвольный выпуклый многогранник.
- Трёхслойная сеть  $\mathbb{R}^n$  позволяет отделить произвольную многогранную область, не обязательно выпуклую.

# 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  $\sigma$  можно приблизить любую непрерывную функцию с любой желаемой точностью

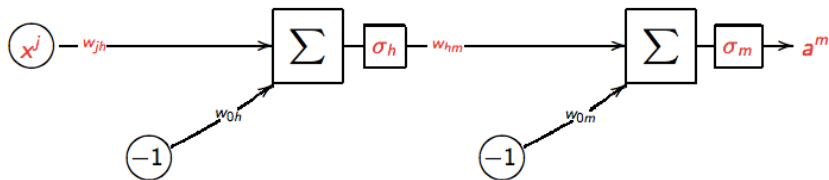
# Многослойная нейронная сеть

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





# Многослойная нейронная сеть



# Стохастический градиент

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

- 1 **function** STOCHASTIC\_GRADIENT( $X^l, \alpha, \eta$ )
- 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:** Сохранять некоторые промежуточные результаты в узлах сети.

# Задача дифференцирования суперпозиции функций

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

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

# Задача дифференцирования суперпозиции функций

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

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

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  $a^m(x_i)$ ,  $m = 1, \dots, M$  на объекте  $x_i$ :

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

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

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

# 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

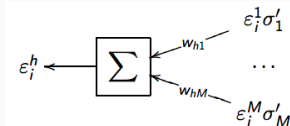
Ошибка на выходном слое:

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

Ошибка на скрытом слое:

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

$\varepsilon_i^h$  вычисляется по  $\varepsilon_i^m$ , если запустить сеть «задом наперёд»:



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?

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

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

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

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

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

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

# Алгоритм обратного распространения ошибки

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

# Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION( $X^l, H, \alpha, \eta$ )

2     ...

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

4         Взять  $x_i$  из  $X^l$

5

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



# Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION( $X^l, H, \alpha, \eta$ )

2 ...

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

4     Взять  $x_i$  из  $X^l$

5

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

6  $\left\{ \begin{array}{l} \varepsilon_i^h = \sum_{m=1}^M \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \dots, H \end{array} \right.$

# Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION( $X^l, H, \alpha, \eta$ )

2 ...

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

4     Взять  $x_i$  из  $X^l$

5

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

6

$$\left\{ \begin{array}{l} \varepsilon_i^h = \sum_{m=1}^M \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \dots, H \end{array} \right.$$

7

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

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

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

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

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

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

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



- + Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
- + Легко обобщается на любые  $\sigma$ ,  $\mathcal{L}$
- + Возможно динамическое (потокковое) обучение
- + На сверхбольших выборках не обязательно брать все  $x_i$
- + Возможность распараллеливания
  
- Возможна медленная сходимость
- Застревание в локальных минимумах

- + Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
- + Легко обобщается на любые  $\sigma$ ,  $\mathcal{L}$
- + Возможно динамическое (потокковое) обучение
- + На сверхбольших выборках не обязательно брать все  $x_i$
- + Возможность распараллеливания
  
- Возможна медленная сходимость
- Застревание в локальных минимумах
- Проблема переобучения

- + Эффективность: градиент вычисляется за время, сравнимое со временем вычисления самой сети
- + Легко обобщается на любые  $\sigma$ ,  $\mathcal{L}$
- + Возможно динамическое (потокое) обучение
- + На сверхбольших выборках не обязательно брать все  $x_i$
- + Возможность распараллеливания
  
- Возможна медленная сходимость
- Застревание в локальных минимумах
- Проблема переобучения
- Подбор комплекса эвристик

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.

- 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)

- Выбор функций активации в каждом нейроне

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

- 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

---

Почему их снова стали использовать?



Почему их снова стали использовать?

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

# Свёрточные сети

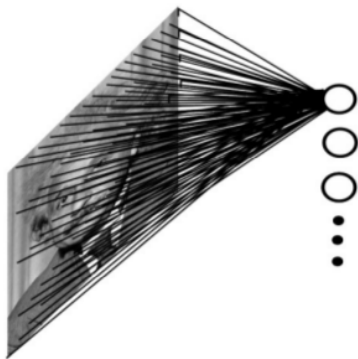
---

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

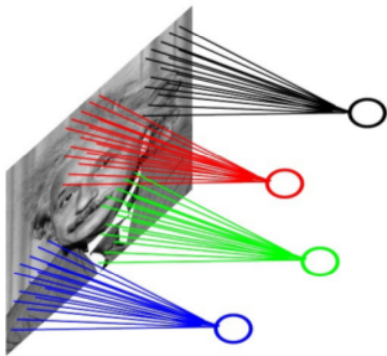
1. Пиксель изображения сильнее связан с соседними пикселями (локальная корреляция)
2. Объект может встретиться в любой части изображения (инвариантность к перемещению)
  
1. Каждый из нейронов подсоединен только к небольшой окрестности изображения
2. Нейроны обладают одними и теми же весами

# Локально-связанный слой

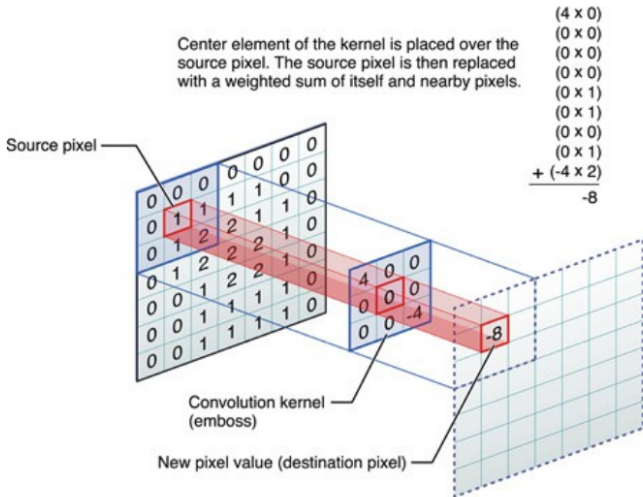
Полносвязный слой



Локально-связанный слой  
(смотрит на небольшую  
область изображения)



# Операция свертки



# Простейшая свертка

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

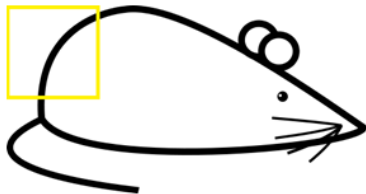


Visualization of a curve detector filter

# Простейшая свертка



Original image



Visualization of the filter on the image



# Простейшая свертка



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

\*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation =  $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$  (A large number!)

# Простейшая свертка



Visualization of the filter on the image

0	0	0	0	0	0	0	0
0	40	0	0	0	0	0	0
40	0	40	0	0	0	0	0
40	20	0	0	0	0	0	0
0	50	0	0	0	0	0	0
0	0	50	0	0	0	0	0
25	25	0	50	0	0	0	0

Pixel representation of receptive field

\*

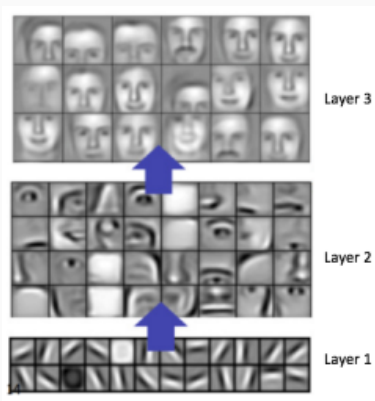
0	0	0	0	0	30	0	0
0	0	0	0	30	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

# Свёрточная сеть

Извлечение признаков во время обучения.





**mite**

**container ship**

**motor scooter**

**leopard**

mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat



**grille**

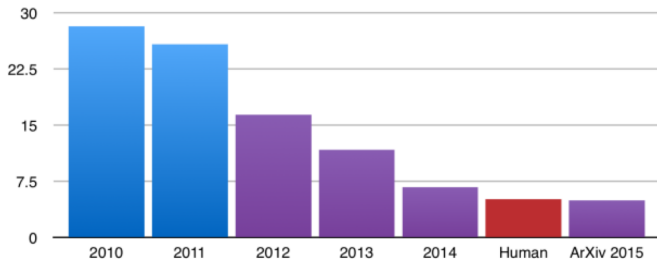
**mushroom**

**cherry**

**Madagascar cat**

convertible	agaric	dalmatian	squirrel monkey
grille	mushroom	grape	spider monkey
pickup	jelly fungus	elderberry	titi
beach wagon	gill fungus	ffordshire bullterrier	indri
fire engine	dead-man's-fingers	currant	howler monkey

## ILSVRC top-5 error on ImageNet



## Похожие изображения



# Похожие изображения

Яндекс

Картинки



Загруженная картинка



Найти

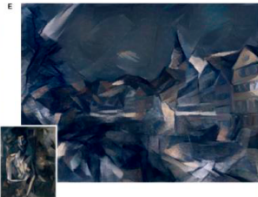
< Вернуться назад



# Стилизация изображений

Пабло Пикассо

Уильям Тёрнер



Винсент Ван Гог

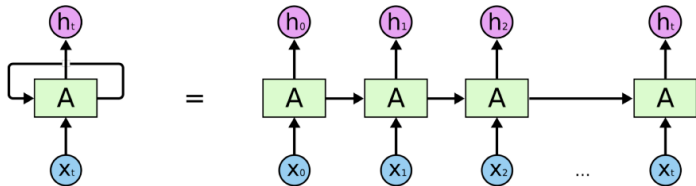
Василий Кандинский

Эдвард Мунк



# Рекуррентные нейронные сети

---



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

VIOLA:

Why, Salisbury must find his flesh and thought  
That which I am not apt, 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.

# 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;
        }
    }
}
```

# Автоматический перевод

The screenshot shows the Google Translate web interface. At the top left is the Google logo, and at the top right is a red 'SIGN IN' button. Below the logo is the 'Translate' heading. The interface is set to translate 'From Latin' to 'To English'. The source text in the left box is a block of Lorem Ipsum: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tristique rhoncus nulla, vel dignissim ligula vulputate nec. Donec velit mauris, ultricies quis elit non, pulvinar feugiat dolor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Morbi nec porta nisi. Donec eget leo quis ante vehicula consectetur. Donec commodo ut ligula eget tincidunt. Sed congue in arcu in ullamcorper. Nullam et tincidunt erat. Mauris semper porttitor leo, porttitor lobortis odio hendrent id. Nullam aliquet metus ut est placerat eleifend'. The translated text in the right box is: 'We will be sure to post a comment. Add tomato sauce, no tank or a traditional or online. Until outdoor environment, and not just any competition, reduce overall pain. Cisco Security, they set up in the throat develop the market beds of Cura, Employment silently churn-class by our union, very beginner himenaeos. Monday gate information. How long before any meaningful development. Until mandatory functional requirements to developers. But across the country in the spotlight in the notebook. The show was shot. Funny lion always feasible, innovative policies hatred assured. Information that is no corporate Japan'. At the bottom right of the translation area, there is an 'Alpha' label with a checkmark icon.

RNN + CNN

---



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 .

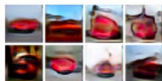
The wind ruffled the snow and ice in the snow . He had no idea how many times he was going to climb into the mountains



# Подпись к изображениям



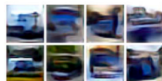
A yellow school bus parked in a parking lot.



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.



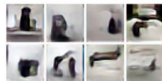
The decadent chocolate desert is on the table.



A bowl of bananas is on the table.



A vintage photo of a cat.



A vintage photo of a dog.

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