



# Введение в нейронные сети

Лектор – Троешестова Лидия

# План

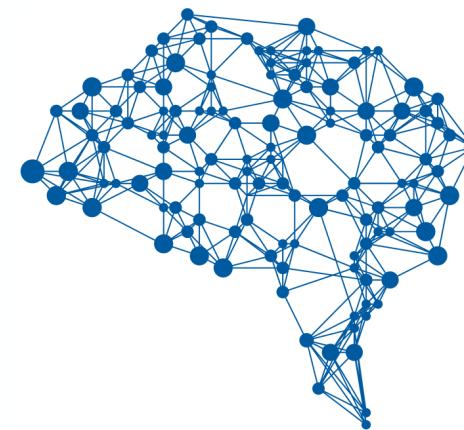
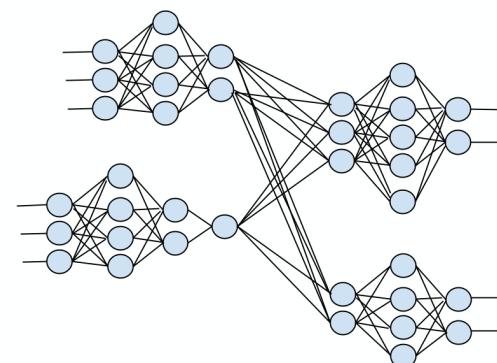
- **Введение, примеры применения**
- Математическая модель нейросети
- Приближение функции нейросетью
- Обучение нейросети



# Что такое эти ваши нейронные сети?

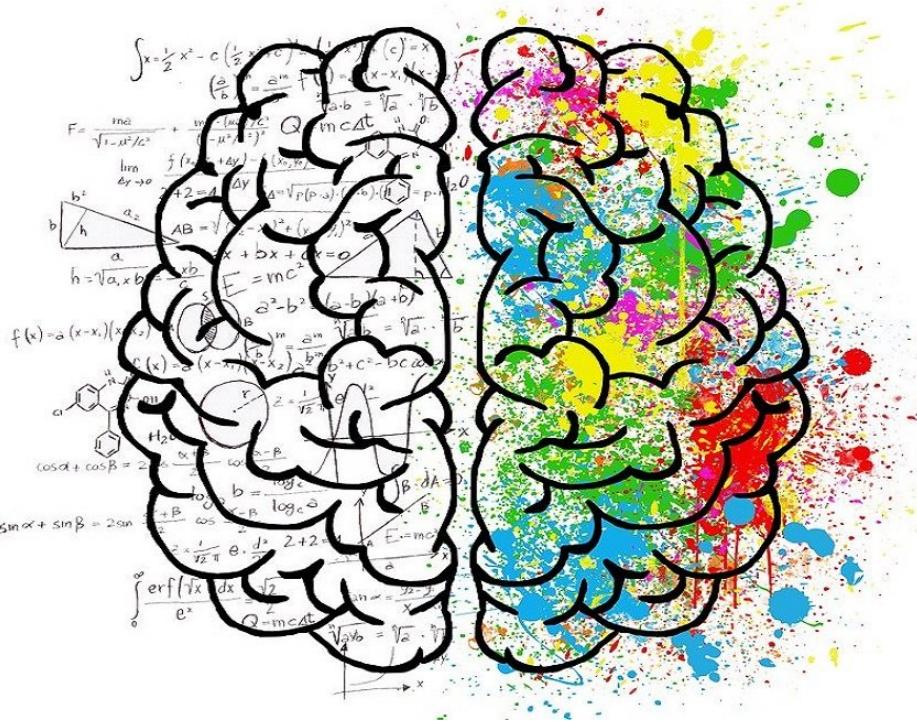
**Нейронная сеть** — математическая модель, а также ее программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

Это понятие возникло при изучении процессов, протекающих в мозге.





# Мозг



Мозг – это

- Система обработки информации
- Сложный, нелинейный, параллельный компьютер

Решает множество сложных задач из области распознавания образов, обработки сигналов и прочее



# Примеры



Задача распознавания  
знакомого лица в толпе  
займет **0.1-0.2 секунды**



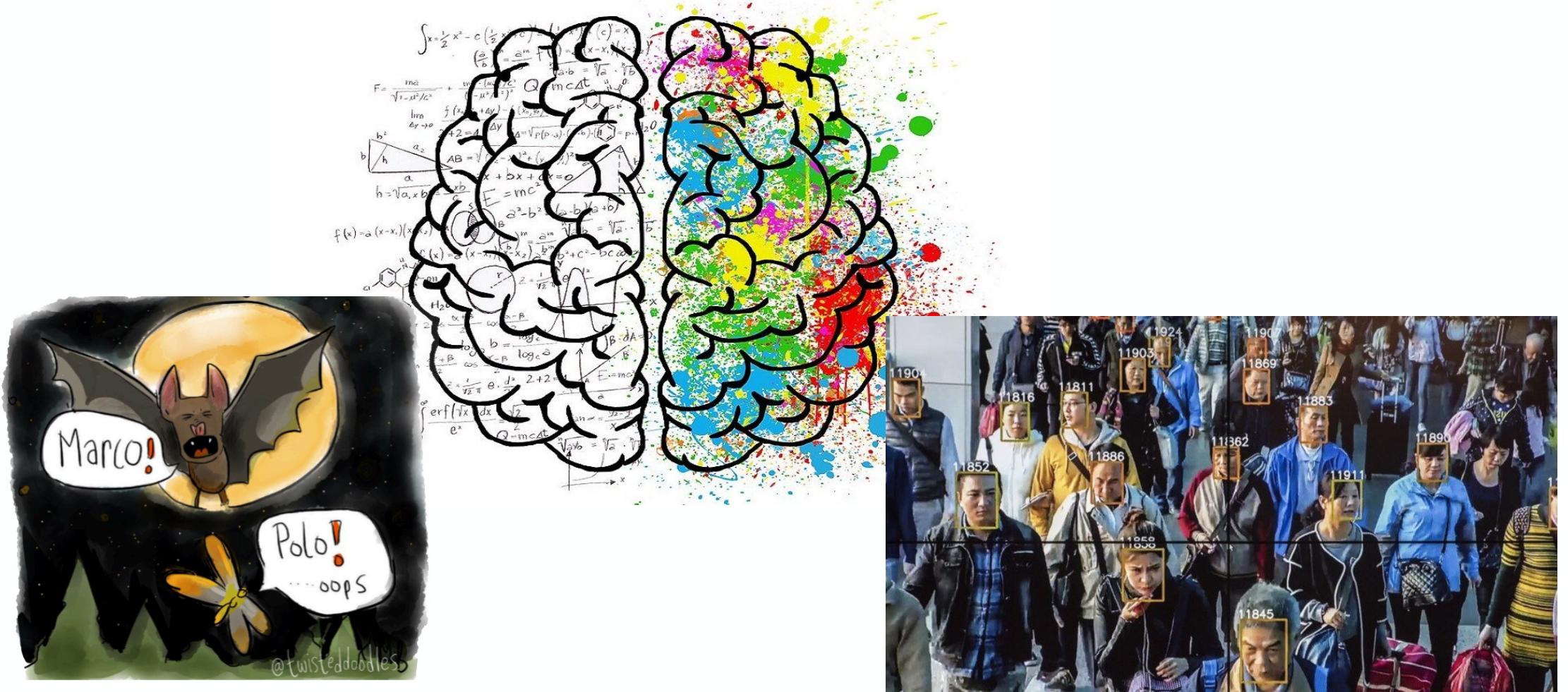
# Примеры

Эхо-локация. Летучая мышь получает информацию про положение, размер, скорость объекта.



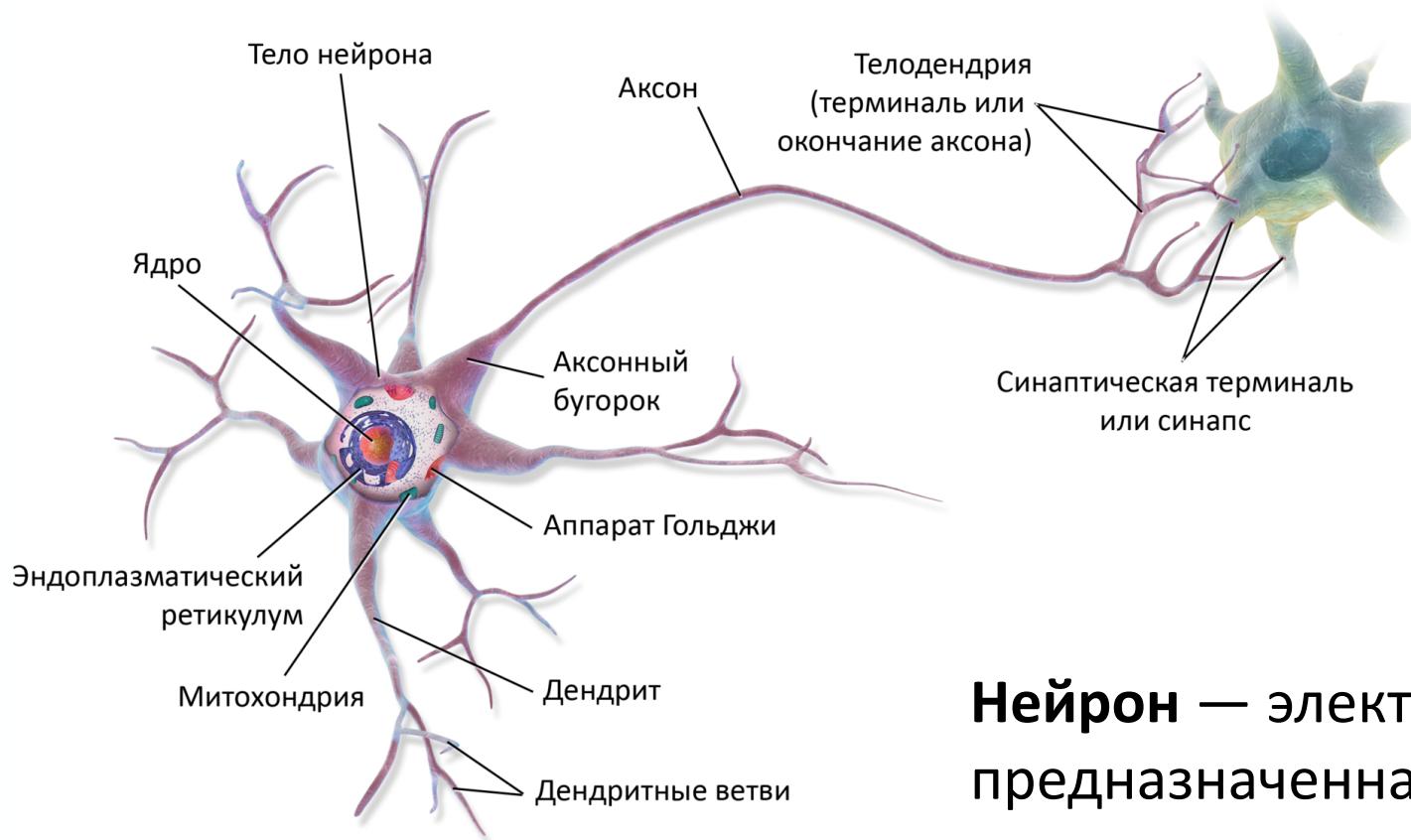


# Почему мы справляемся со столь сложными задачами?





# Биологический нейрон

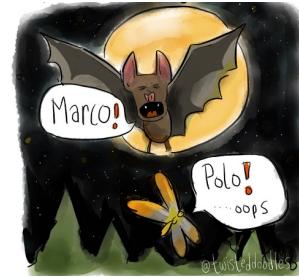


Структурно-функциональная единица нервной системы — это **нейрон**.

**Нейрон** — электрически возбудимая клетка, предназначенная для приёма, обработки, хранения, передачи и вывода информации с помощью электрических и химических сигналов.



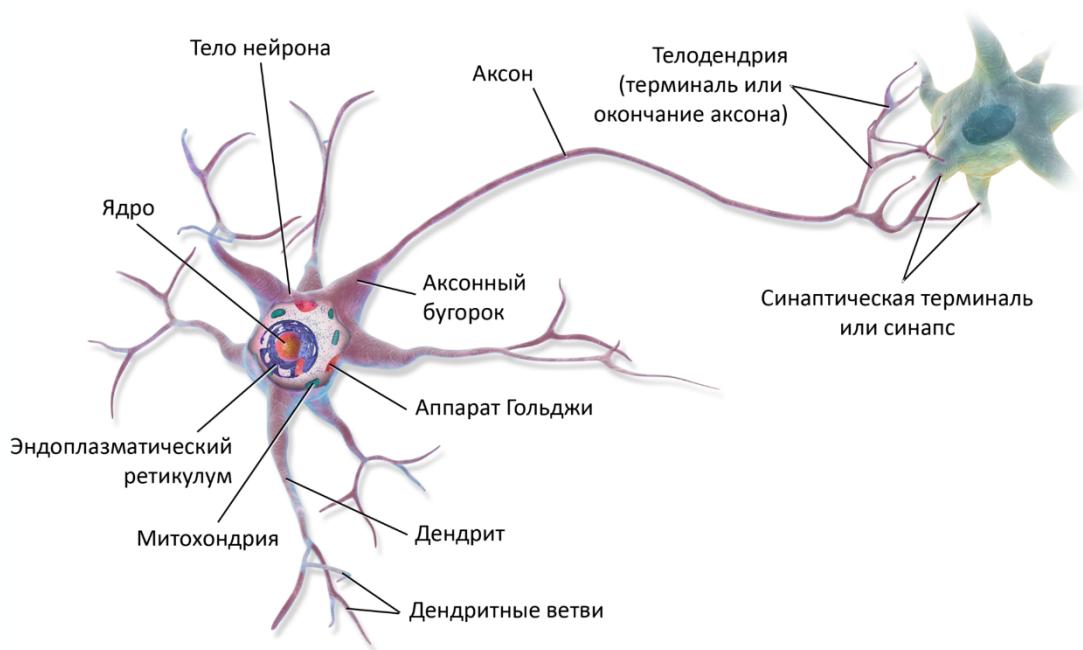
# Почему мы справляемся со столь сложными задачами?



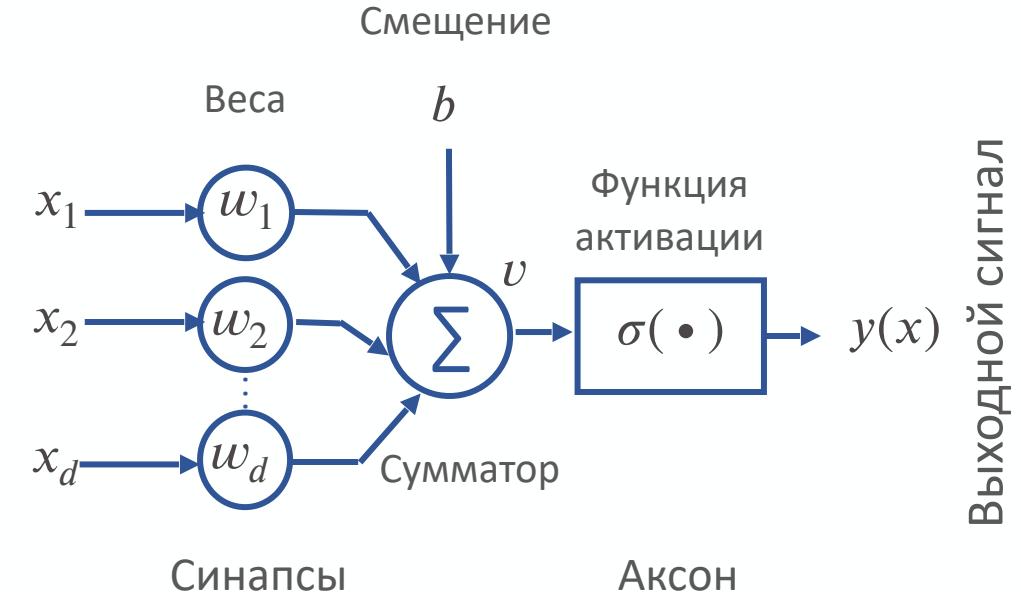


# Бионика: нейронные сети

## Биологический нейрон



## Искусственный нейрон (перцептрон)





# Историческая справка

1943 г. - McCulloch, Pitts

- Модель нейрона как «всё или ничего»
- Принципиальная возможность выполнить любые вычисления сетью нейронов

1949 г. – Hebb

- Открытие принципа формирования нейронных связей, взаимодействия нейронов

1958 г. – Rosenblatt

- Открытие «перцептрана» - устройства, моделирующего процесс человеческого восприятия



# Области применения

**Распознавание лиц / Face detection**

Цель — определить личность человека по снимку

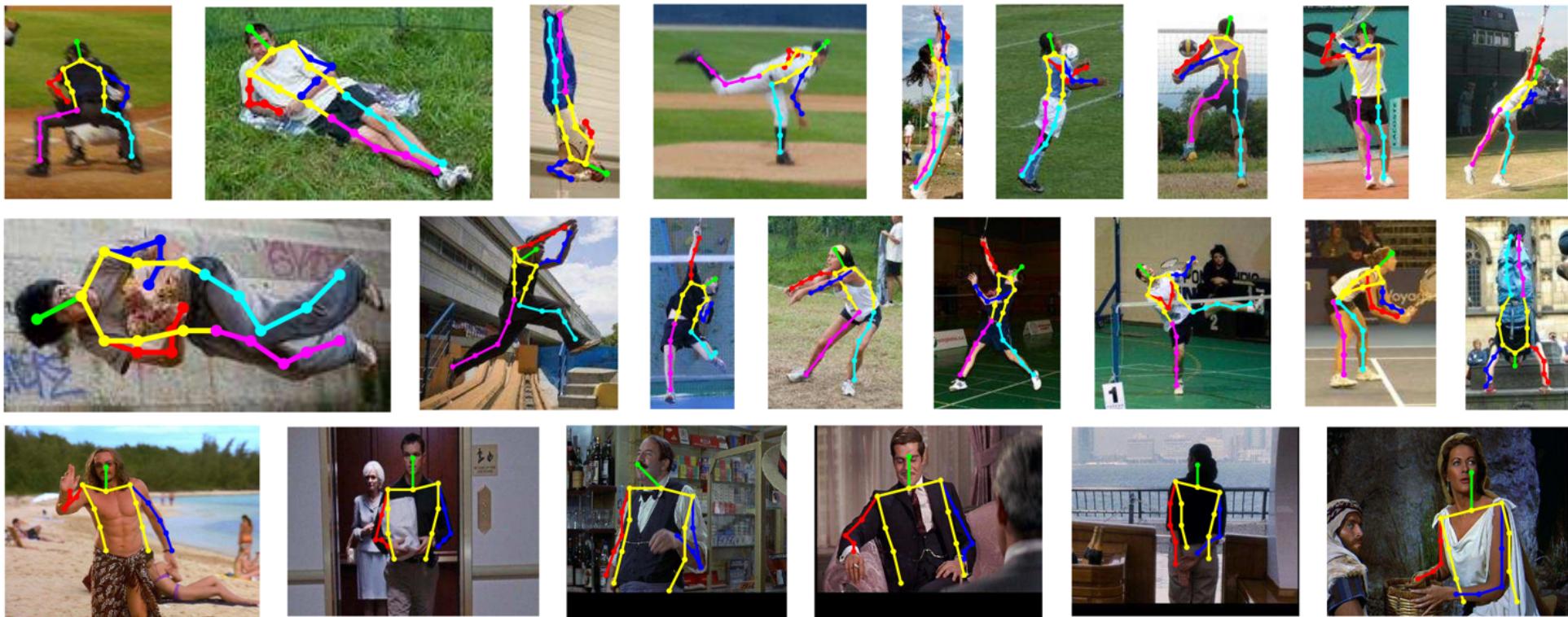




# Области применения

**Определение позы человека / Pose estimation**

Цель — определить линии скелета человека

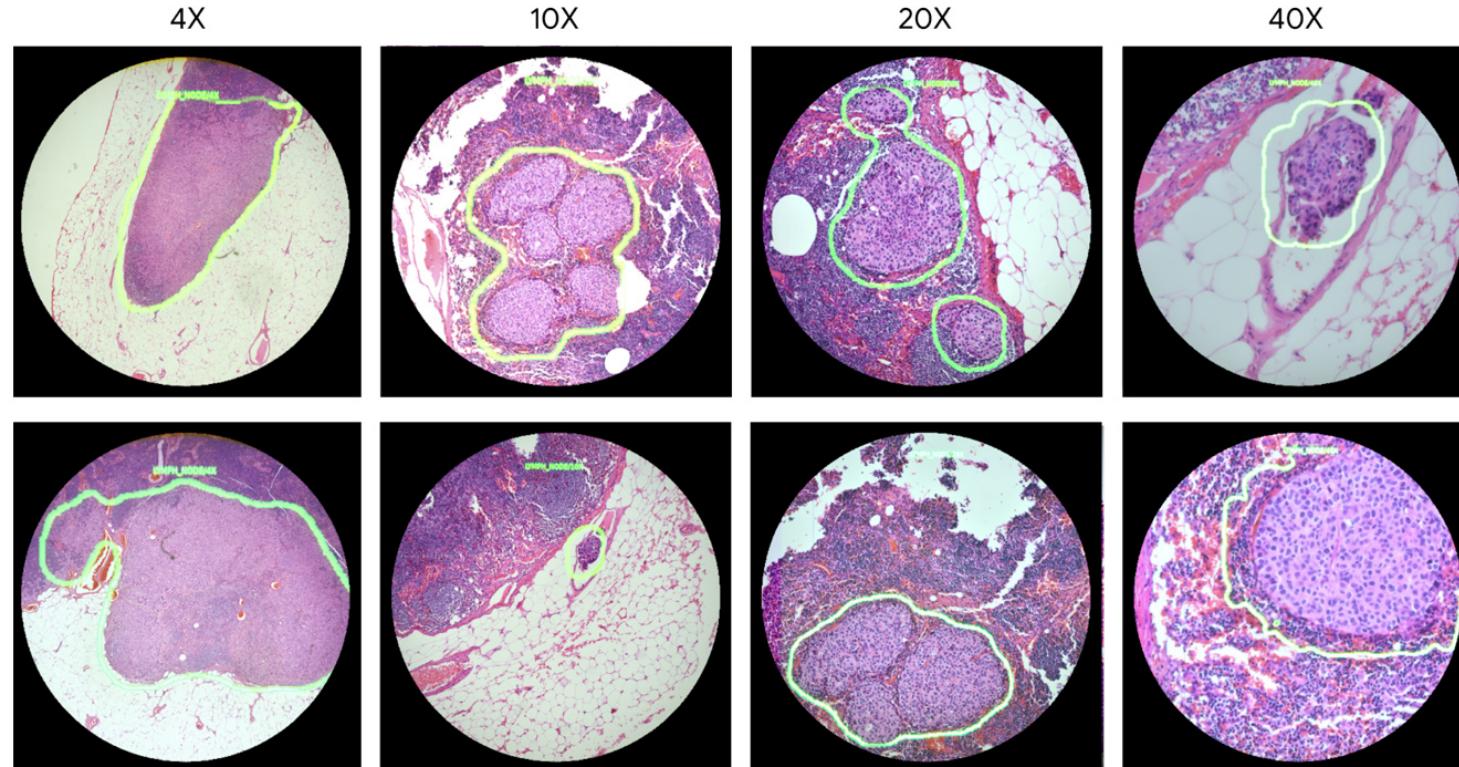




# Области применения

**Сегментация опухоли / Tumor segmentation**

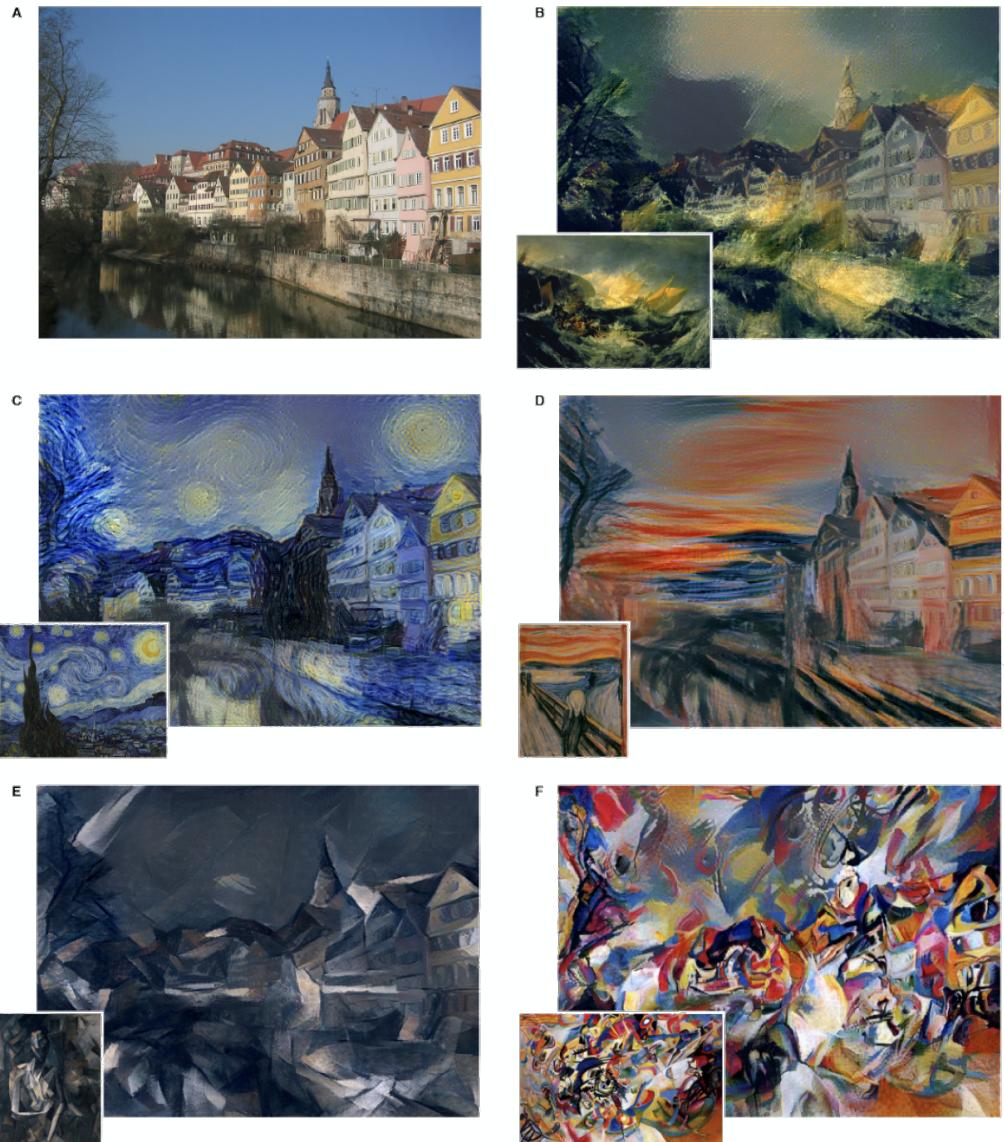
Цель — для каждого пикселя определить принадлежит ли он опухоли



# Области применения

## Перенос стиля / Style transfer

Цель — изменить изображение так, чтобы новое изображение сохранило структуру исходного, но переняло стиль целевого изображения





# Области применения

**Генерация изображений / Image generation**

Цель — создание уникальных изображений

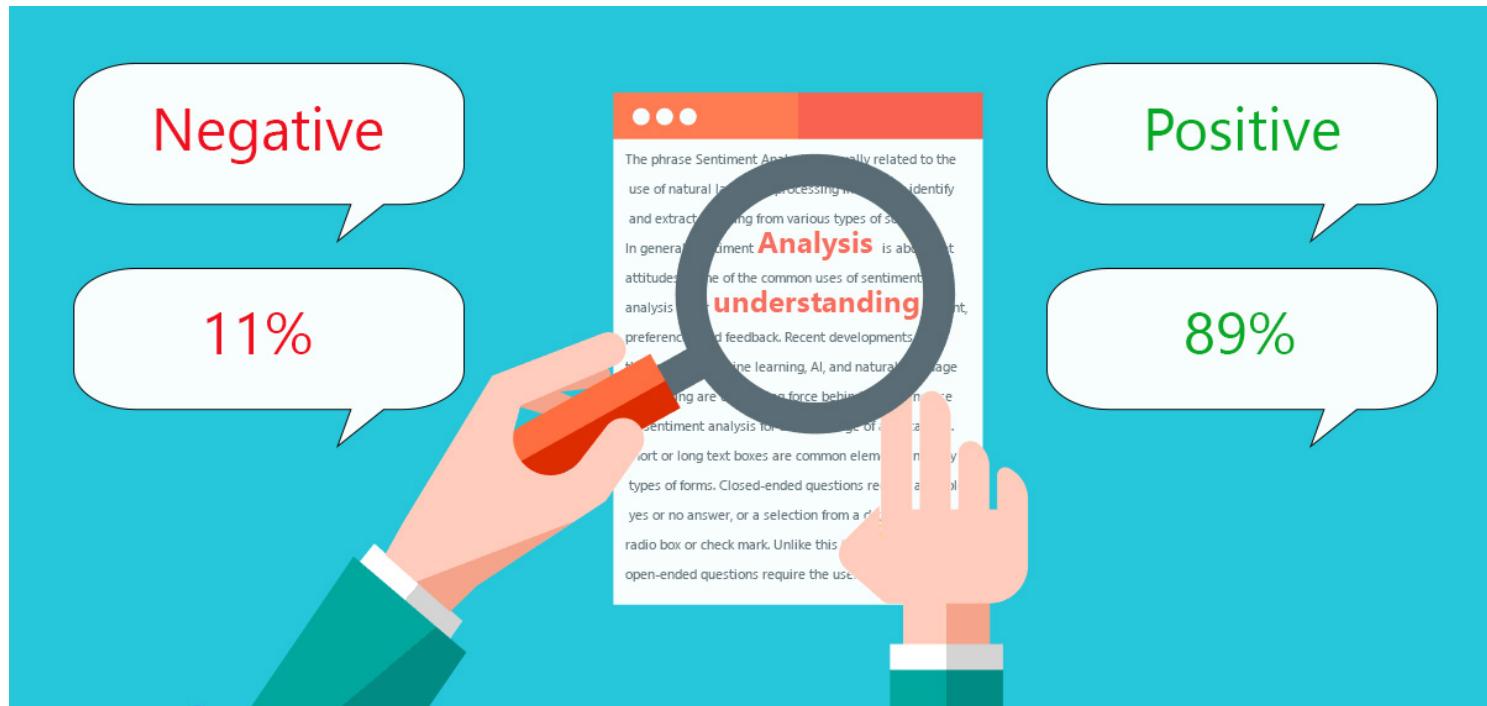




# Области применения

**Определение тональности текста / Sentiment analysis**

Цель — по определить является текст позитивным или негативным.





# Области применения

**Машинный перевод / Machine Language translation**

Цель — перевести текст на одном языке на другой язык

The screenshot shows the Yandex.Translate website interface. At the top, there's a navigation bar with 'Яндекс Переводчик' (Yandex Translator) and links for 'Текст' (Text), 'Сайты' (Websites), 'Документы' (Documents), and 'Картинки' (Images). On the right of the bar are buttons for 'Для бизнеса' (For business) and a user profile icon. Below the bar, the source language is set to 'АНГЛИЙСКИЙ' (English) and the target language is 'РУССКИЙ' (Russian), indicated by a double-headed arrow icon. The main translation area contains the English sentence 'neural networks are so cool!' on the left and its Russian translation 'нейронные сети - это так круто!' on the right. There are also icons for audio playback, a keyboard, and social sharing at the bottom.

Яндекс Переводчик Текст Сайты Документы Картинки Для бизнеса

● АНГЛИЙСКИЙ ↔ РУССКИЙ

neural networks are so cool! × нейронные сети - это так круто!

Перевести в Google



# Области применения

## Генерация текста / text generation

Цель — сгенерировать связный текст по соответствующей тематике

*Proof.* Omitted. □

**Lemma 0.1.** Let  $\mathcal{C}$  be a set of the construction.

Let  $\mathcal{C}$  be a gerber covering. Let  $\mathcal{F}$  be a quasi-coherent sheaves of  $\mathcal{O}$ -modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

*Proof.* This is an algebraic space with the composition of sheaves  $\mathcal{F}$  on  $X_{\text{étale}}$  we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where  $\mathcal{G}$  defines an isomorphism  $\mathcal{F} \rightarrow \mathcal{F}$  of  $\mathcal{O}$ -modules. □

**Lemma 0.2.** This is an integer  $\mathcal{Z}$  is injective.

*Proof.* See Spaces, Lemma ???. □

**Lemma 0.3.** Let  $S$  be a scheme. Let  $X$  be a scheme and  $X$  is an affine open covering. Let  $\mathcal{U} \subset \mathcal{X}$  be a canonical and locally of finite type. Let  $X$  be a scheme. Let  $X$  be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let  $X$  be a scheme. Let  $X$  be a scheme covering. Let

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

be a morphism of algebraic spaces over  $S$  and  $Y$ .

*Proof.* Let  $X$  be a nonzero scheme of  $X$ . Let  $X$  be an algebraic space. Let  $\mathcal{F}$  be a quasi-coherent sheaf of  $\mathcal{O}_X$ -modules. The following are equivalent

- (1)  $\mathcal{F}$  is an algebraic space over  $S$ .
- (2) If  $X$  is an affine open covering.

Consider a common structure on  $X$  and  $X$  the functor  $\mathcal{O}_X(U)$  which is locally of finite type. □

This since  $\mathcal{F} \in \mathcal{F}$  and  $x \in \mathcal{G}$  the diagram

$$\begin{array}{ccccc}
 S & \xrightarrow{\quad} & & & \\
 \downarrow & & & & \\
 \xi & \longrightarrow & \mathcal{O}_{X'} & & \\
 \uparrow \text{gor}_s & & \uparrow & & \\
 & & =\alpha' & \longrightarrow & \\
 & & \uparrow =\alpha' & & \\
 & & =\alpha' & \longrightarrow & \alpha \\
 & & & & \\
 \text{Spec}(K_\psi) & & \text{Mor}_{\text{Sets}} & & d(\mathcal{O}_{X/k}, \mathcal{G}) \\
 & & & & \downarrow X \\
 & & & & 
 \end{array}$$

is a limit. Then  $\mathcal{G}$  is a finite type and assume  $S$  is a flat and  $\mathcal{F}$  and  $\mathcal{G}$  is a finite type  $f_*$ . This is of finite type diagrams, and

- the composition of  $\mathcal{G}$  is a regular sequence,
- $\mathcal{O}_{X'}$  is a sheaf of rings.

□

*Proof.* We have see that  $X = \text{Spec}(R)$  and  $\mathcal{F}$  is a finite type representable by algebraic space. The property  $\mathcal{F}$  is a finite morphism of algebraic stacks. Then the cohomology of  $X$  is an open neighbourhood of  $U$ . □

*Proof.* This is clear that  $\mathcal{G}$  is a finite presentation, see Lemmas ??.

A reduced above we conclude that  $U$  is an open covering of  $\mathcal{C}$ . The functor  $\mathcal{F}$  is a “field”

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\bar{x}} \dashrightarrow (\mathcal{O}_{X_{\text{étale}}}) \longrightarrow \mathcal{O}_{X,\eta}^{-1} \mathcal{O}_{X,\lambda} (\mathcal{O}_{X,\eta}^{\text{pt}})$$

is an isomorphism of covering of  $\mathcal{O}_{X,\eta}$ . If  $\mathcal{F}$  is the unique element of  $\mathcal{F}$  such that  $X$  is an isomorphism.

The property  $\mathcal{F}$  is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme  $\mathcal{O}_X$ -algebra with  $\mathcal{F}$  are opens of finite type over  $S$ .

If  $\mathcal{F}$  is a scheme theoretic image points. □

If  $\mathcal{F}$  is a finite direct sum  $\mathcal{O}_{X,k}$  is a closed immersion, see Lemma ???. This is a sequence of  $\mathcal{F}$  is a similar morphism.



# Области применения

## Текстовые боты / text bots

GPT-4

You will be given a name of an object (such as Car, Chair, Elephant) and a letter in the alphabet. Your goal is to first produce a 1-line description of how that object can be combined with the letter in an image (for example, for an elephant and the letter J, the trunk of the elephant can have a J shape, and for the letter A and a house, the house can have an A shape with the upper triangle of the A being the roof). Following the short description, please create SVG code to produce this (in the SVG use shapes like ellipses, triangles etc and polygons but try to defer from using quadratic curves).

Round 1: A pig and the letter H.

Description: The pig can be lying on its back with its four legs forming the two vertical bars of the H, and its curly tail forming the horizontal bar.



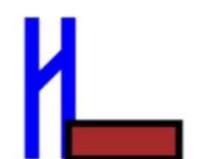
Round 2: A house and the letter U.

Description: A house with a U shape, where the two vertical lines of the U are the walls and the horizontal line is the floor. The roof is a triangle above the U.



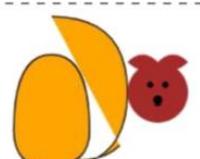
Round 3: A chair and the letter K.

Description: The back of the chair can have a K shape, with two vertical lines and a diagonal line connecting them. The seat of the chair can be a rectangle.



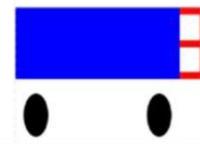
Round 4: A dog and the letter Q.

Description: The dog is sitting with its tail curled around its body, forming a Q shape. The Q is filled with orange and has a black outline.



Round 5: A car and the letter E.

Description: The car has an E shape on its front bumper, with the horizontal lines of the E being lights and the vertical line being the license plate.





# Области применения

**Распознавание речи / speech recognition**

Цель — перевести человеческую речь в текст





# Области применения

**Прогноз погоды / whether forecasting**

Цель — предсказать температуру, осадки, скорость и направление ветра и т.д.





# Области применения

**Предсказание продаж / sales prediction**

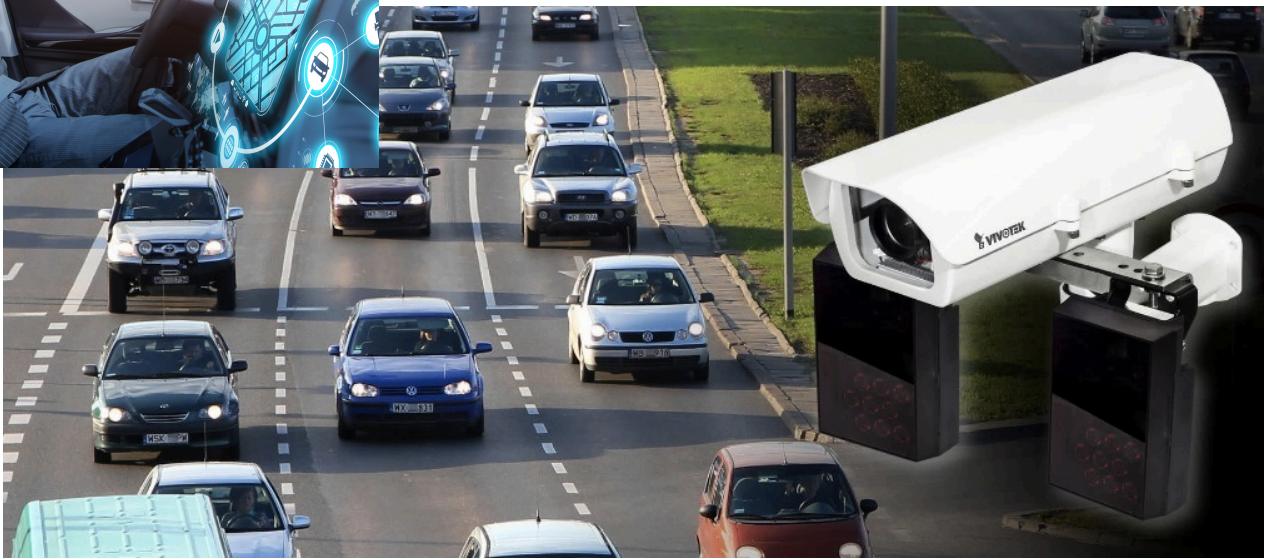
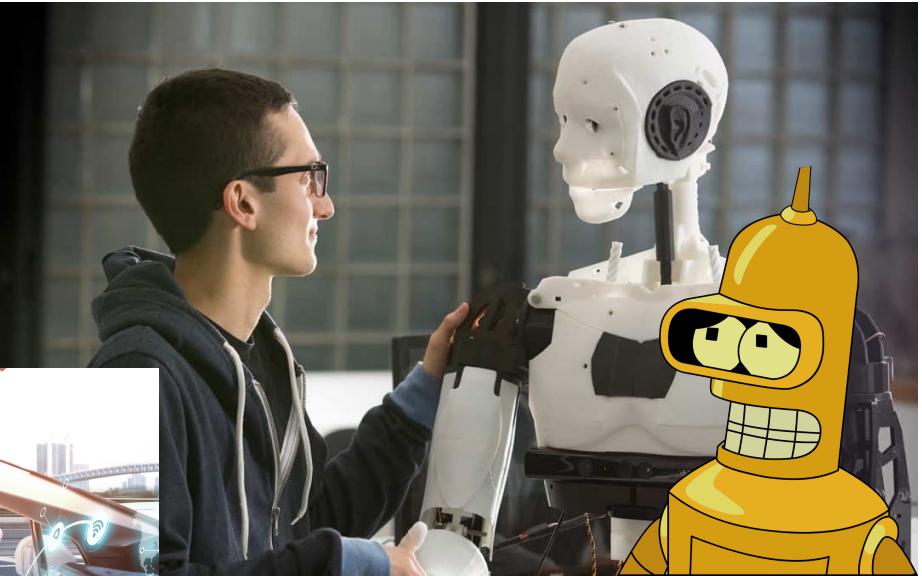
Цель — предсказать продажи товаров на некоторый период времени вперед





# Области применения

И многое другое...





# План

- Введение, примеры применения
- **Математическая модель нейросети**
- Приближение функции нейросетью
- Обучение нейросети

# Модель нейрона

Обозначим

$x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$  — один объект, где  $x_1, \dots, x_d$  — признаки;

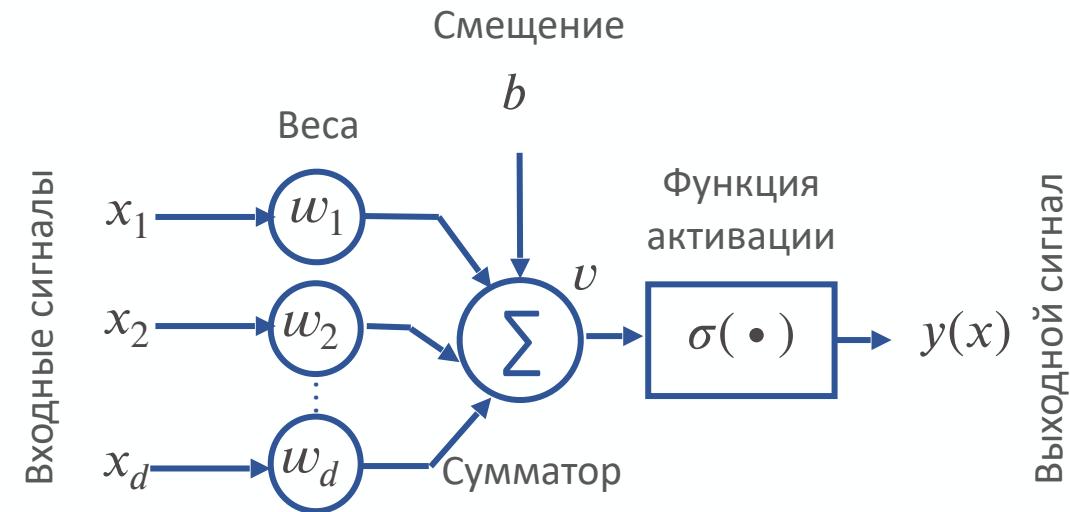
$w = (w_1, \dots, w_d)^T \in \mathbb{R}^d$  — вектор весов;

$b \in \mathbb{R}$  — смещение.

Выход нейрона —

$$y(x) = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d w_j x_j + b\right),$$

где  $\sigma$  — некоторая кус.-дифф. функция,  
назовем ее **функцией активации**.



# Что-то знакомое...

На что похожа эта формула?

$$y(x) = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d w_j x_j + b\right)$$

Линейная регрессия для 1 элемента:

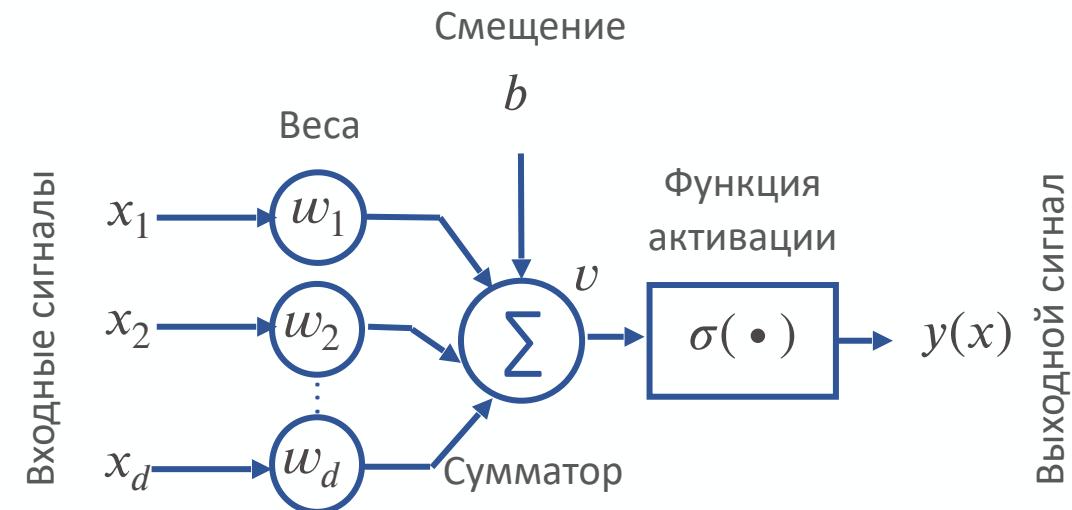
$$y(x) = \langle x, w \rangle + b = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d x_j w_j + b\right), \text{ где } \sigma(z) = z \text{ — линейная функция}$$

Логистическая регрессия для 1 элемента (будет на 3 курсе)

$$y(x) = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d x_j w_j + b\right), \text{ где } \sigma(z) = \frac{1}{1 + e^{-z}} \text{ — логистическая сигмоида}$$

Пуассоновская регрессия для 1 элемента (будет на 3 курсе)

$$y(x) = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d x_j w_j + b\right), \text{ где } \sigma(z) = e^z$$





# Решение задачи регрессии

Линейная регрессия:  $\hat{y} = Xw + b$ , т.е.  $\hat{y}_i = \sum_{j=1}^d x_{ij}w_j + b$

где  $X = (x_{ij})_{ij}$  — матрица входных данных,  $i \in \{1, \dots, n\}$ ,  $j \in \{1, \dots, d\}$

$\hat{y} = (\hat{y}_1, \dots, \hat{y}_d)^T$  — вектор предсказания,

$w = (w_1, \dots, w_d)^T$  — вектор весов,  $b$  — сдвиг.

Задачу можно решить аналитически. А можно с помощью **градиентного спуска**.

**Зададим функцию**, которую мы хотим минимизировать

$$L = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \text{ — MSE (Mean Squared Error)}$$

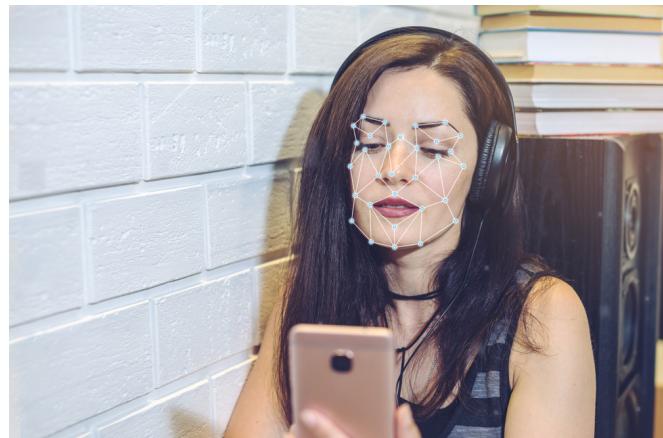
Пусть  $\theta = (w, b)$ , тогда оптимизация будет следующей:

$$\theta_t = \theta_{t-1} - \eta \nabla L(\theta_{t-1}), \quad \text{где } \eta \text{ — скорость обучения}$$



# Более сложные задачи

- У линейной и логистической регрессий ограниченная область применения
- Для того, чтобы решить нелинейную задачу,  
нужно делать сложные преобразования с признаками
- Один нейрон не справится со сложными задачами... 😞



*Proof.* Omitted. □

**Lemma 0.1.** Let  $\mathcal{C}$  be a set of the construction.  
Let  $\mathcal{C}$  be a gerber covering. Let  $\mathcal{F}$  be a quasi-coherent sheaves of  $\mathcal{O}$ -modules. We have to show that  
$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

*Proof.* This is an algebraic space with the composition of sheaves  $\mathcal{F}$  on  $X_{\text{étale}}$ , we have  
$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$
 where  $\mathcal{G}$  defines an isomorphism  $\mathcal{F} \rightarrow \mathcal{G}$  of  $\mathcal{O}$ -modules. □

**Lemma 0.2.** This is an integer  $Z$  is injective.  
*Proof.* See Spaces, Lemma ???. □

**Lemma 0.3.** Let  $S$  be a scheme. Let  $X$  be a scheme and  $X$  is an affine open covering. Let  $\mathcal{U} \subset X$  be a canonical and locally of finite type. Let  $X$  be a scheme. Let  $X$  be a scheme which is equal to the formal complex.  
The following to the construction of the lemma follows.  
Let  $X$  be a scheme. Let  $X$  be a scheme covering. Let  
$$b: X \rightarrow Y' \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$
 be a morphism of algebraic spaces over  $S$  and  $Y$ .  
Proof. Let  $X$  be a nonzero scheme of  $S$ . Let  $X$  be an algebraic space. Let  $\mathcal{F}$  be a quasi-coherent sheaf of  $\mathcal{O}_X$ -modules. The following are equivalent

- (1)  $\mathcal{F}$  is an algebraic space over  $S$ .
- (2) If  $X$  is an affine open covering.

Consider a common structure on  $X$  and  $X$  the functor  $\mathcal{O}_X(U)$  which is locally of finite type. □

This since  $\mathcal{F} \in \mathcal{F}$  and  $x \in \mathcal{G}$  the diagram  
$$\begin{array}{ccc} S & \xrightarrow{\quad} & \mathcal{O}_{X'} \\ \downarrow & \lhd & \downarrow \text{gr}_{x_1} \\ \mathcal{O}_{X'} & \xrightarrow{\quad} & \end{array}$$
  
$$\begin{array}{ccc} & & \alpha' \\ & \lhd & \downarrow \\ & \alpha' & \xrightarrow{\quad} \alpha \\ & \lhd & \downarrow \\ \text{Spec}(K_{\mathcal{G}}) & \xrightarrow{\quad} & \text{Mor}_{\mathcal{O}_{X'}} \\ & & \downarrow \text{d}(\mathcal{O}_{X'}, \mathcal{G}) \\ & & X \end{array}$$

is a limit. Then  $\mathcal{G}$  is a finite type and assume  $S$  is a flat and  $\mathcal{F}$  and  $\mathcal{G}$  is a finite type  $f$ . This is of finite type diagrams, and

- the composition of  $\mathcal{G}$  is a regular sequence,
- $\mathcal{O}_{X'}$  is a sheaf of rings.

*Proof.* We have see that  $X = \text{Spec}(U)$  and  $\mathcal{F}$  is a finite type representable by algebraic spaces. The property  $\mathcal{F}$  is a finite morphism of algebraic stacks. Then the cohomology of  $X$  is an open neighbourhood of  $U$ . □

*Proof.* This is clear that  $\mathcal{G}$  is a finite presentation, see Lemmas ???. A reduced above we conclude that  $U$  is an open covering of  $\mathcal{C}$ . The functor  $\mathcal{F}$  is a “field”

$$\mathcal{O}_{X, x} \rightarrow \mathcal{F}_x \dashv \text{Mor}_{\mathcal{O}_{X'}} \rightarrow \mathcal{O}_{X'}^{\vee} \mathcal{O}_{X, x}(\mathcal{O}_{X'}^{\vee})$$

is an isomorphism of covering of  $\mathcal{O}_{X, x}$ . If  $\mathcal{F}$  is the unique element of  $\mathcal{F}$  such that  $X$  is an isomorphism.

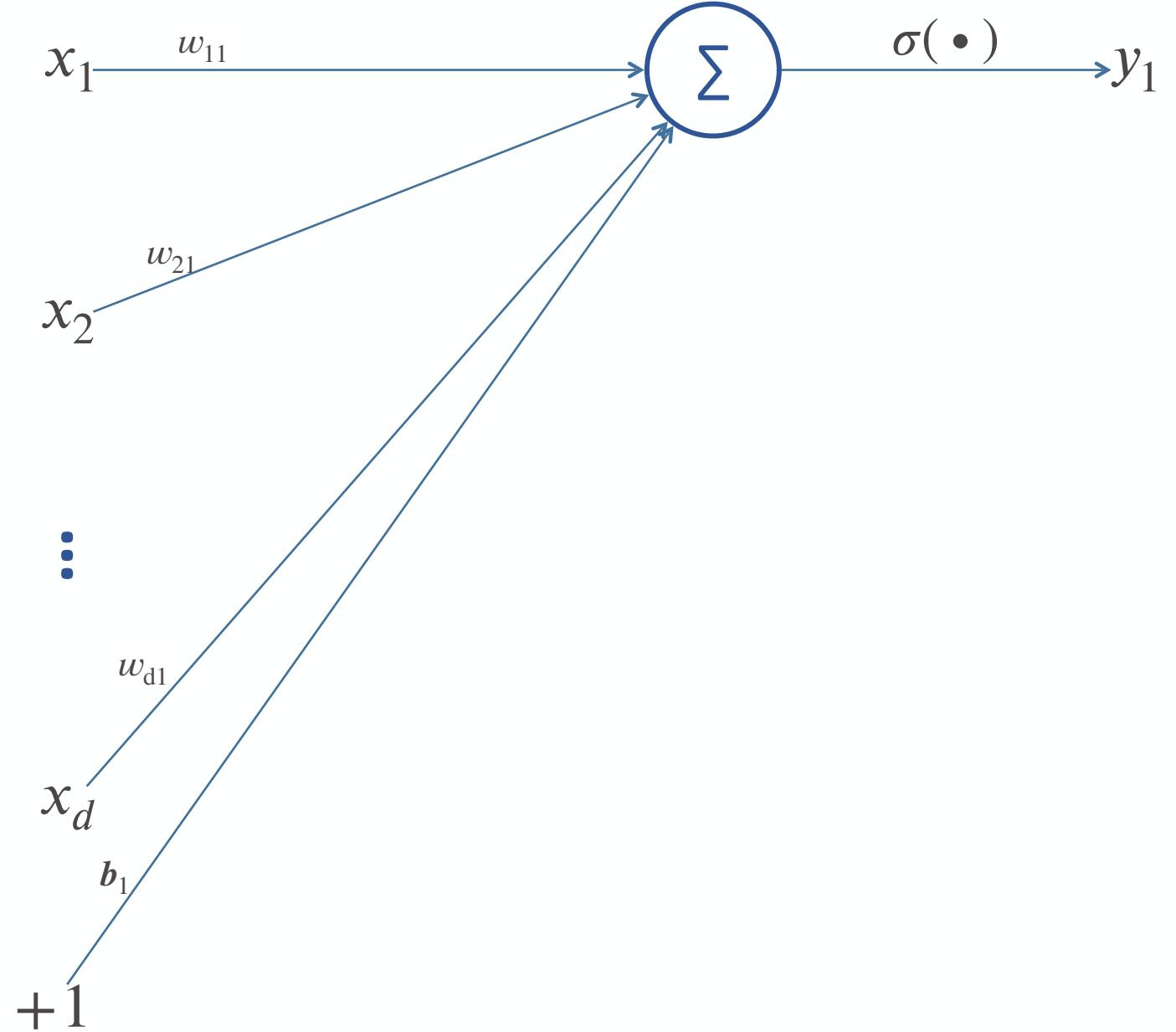
The property  $\mathcal{F}$  is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme  $\mathcal{O}_{X'}\text{-algebra}$  with  $\mathcal{F}$  are opens of finite type over  $S$ . If  $\mathcal{F}$  is a scheme theoretic image point.

If  $\mathcal{F}$  is a finite direct sum  $\mathcal{O}_{X, x}$  is a closed immersion, see Lemma ???. This is a sequence of  $\mathcal{F}$  is a similar morphism.



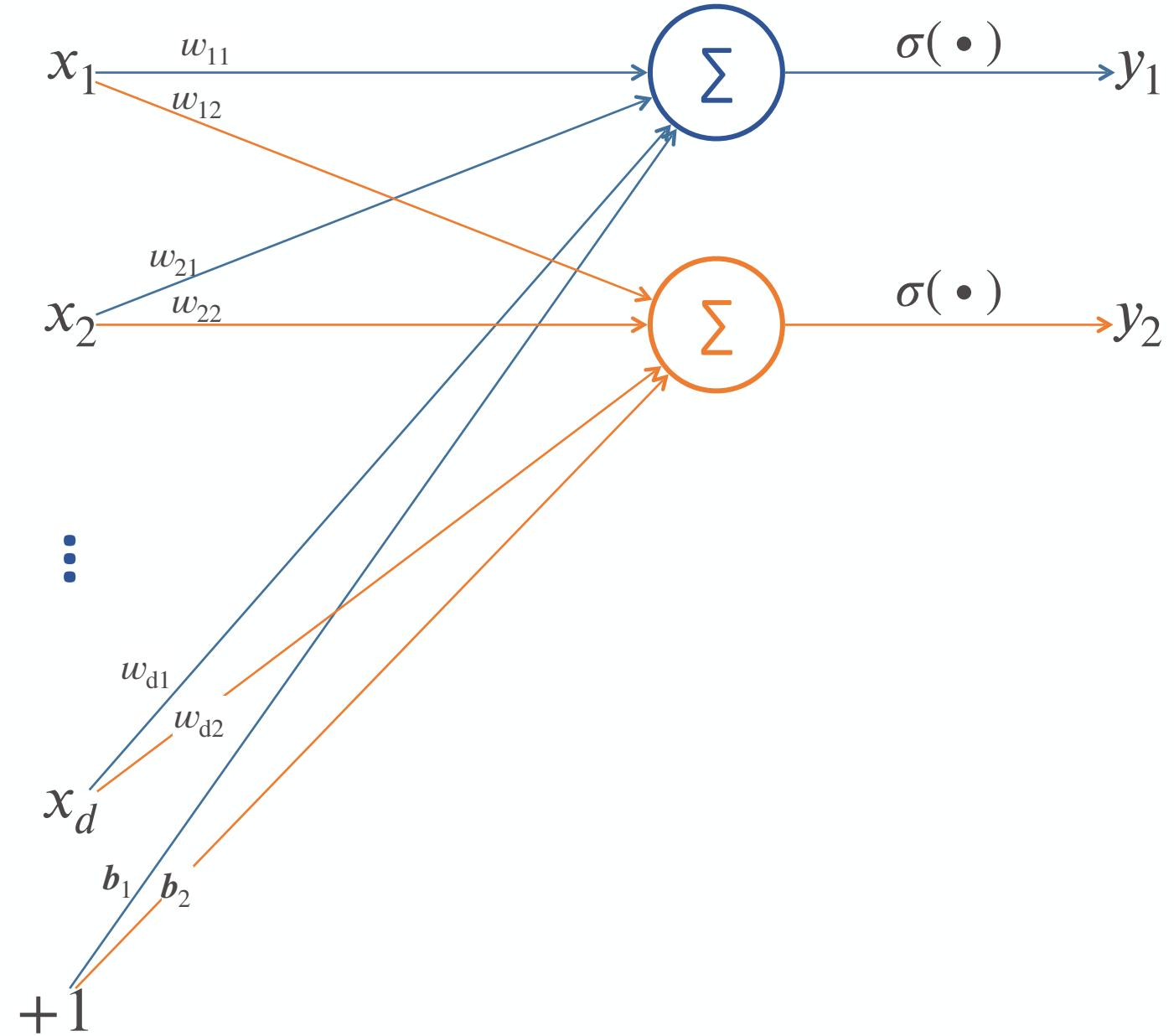
# Однослойная нейронная сеть

Один нейрон  
с весами  $w_{11}, \dots, w_{d1}$   
и смещением  $b_1$



# Однослойная нейронная сеть

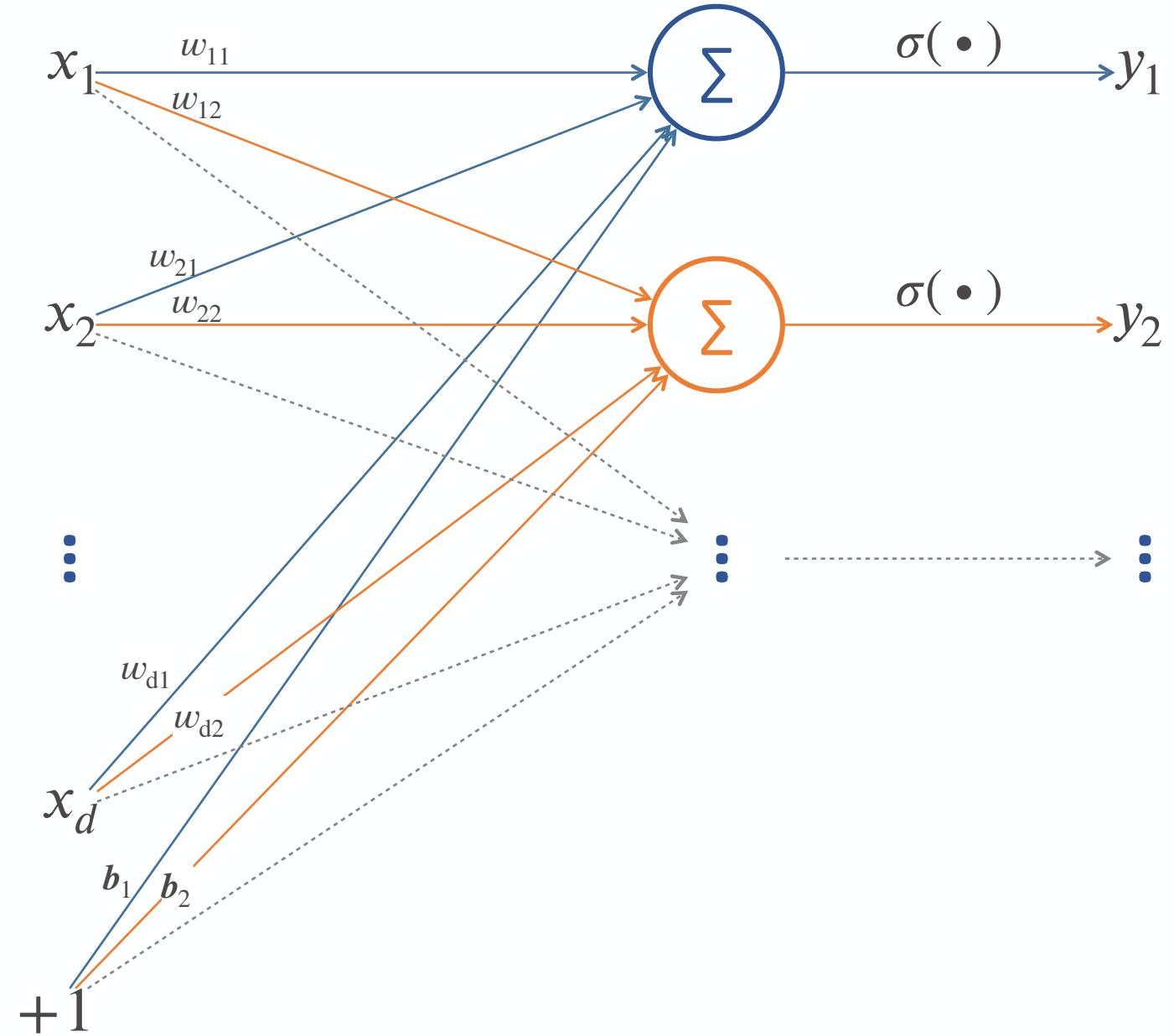
Два нейрона  
с весами  $w_{11}, \dots, w_{d1}$   
и  $w_{12}, \dots, w_{d2}$   
и смещениями  $b_1$  и  $b_2$





# Однослойная нейронная сеть

Больше нейронов



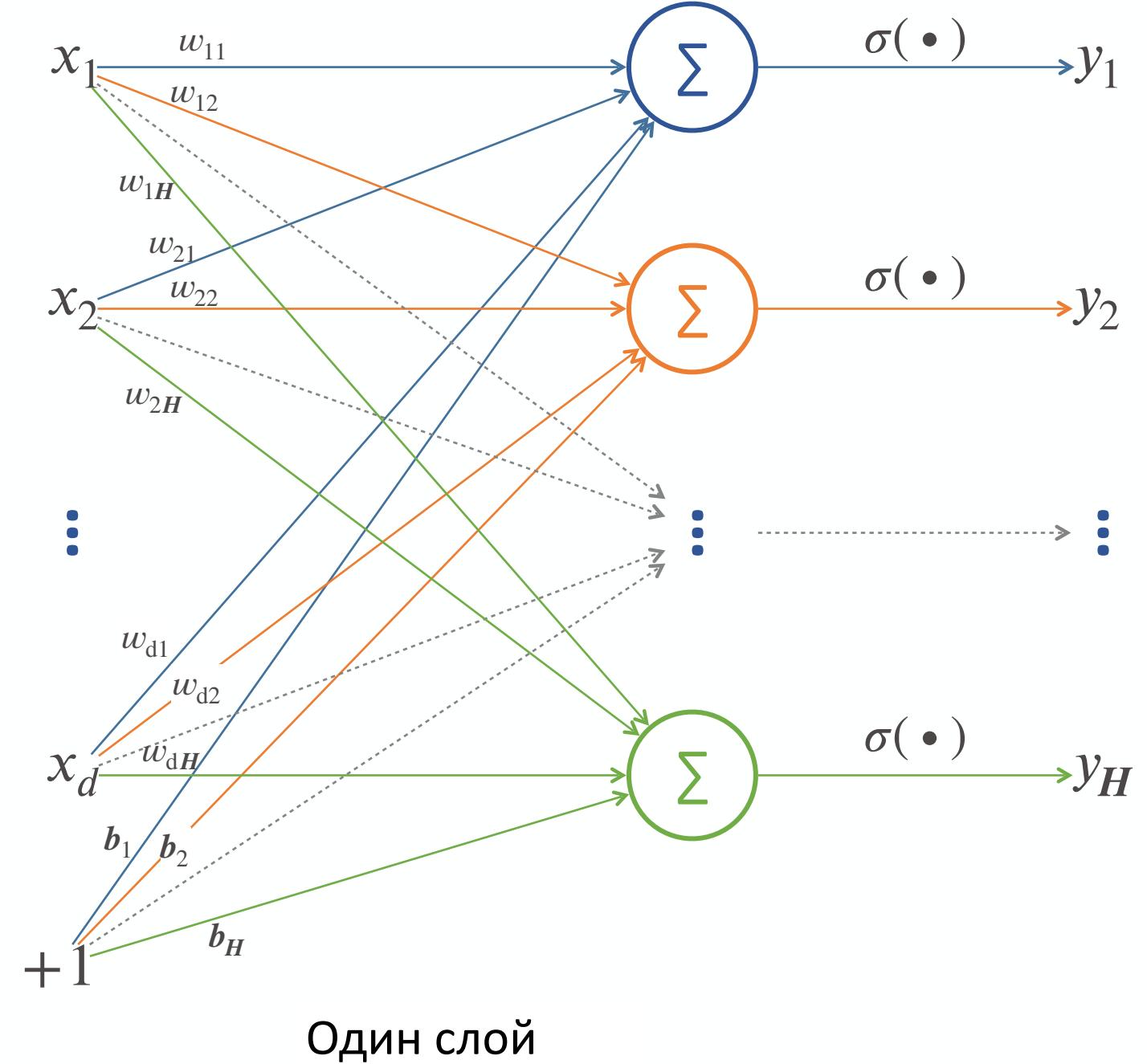


# Однослойная нейронная сеть

Слой размера  $H$   
— набор из  $H$  нейронов

$(w_{jh})_{jh} \in \mathbb{R}^{d \times H}$  и  $(b_h)_h$   
— параметры модели

Такую нейронную сеть  
называют  
**полносвязной**





# Однослойная нейронная сеть

## Матричное представление

Пусть  $x = (x_1, x_2, \dots, x_d)$  — элемент выборки.

$s = (s_1, s_2, \dots, s_H)$  — выходы нейронов до применения функции активации.

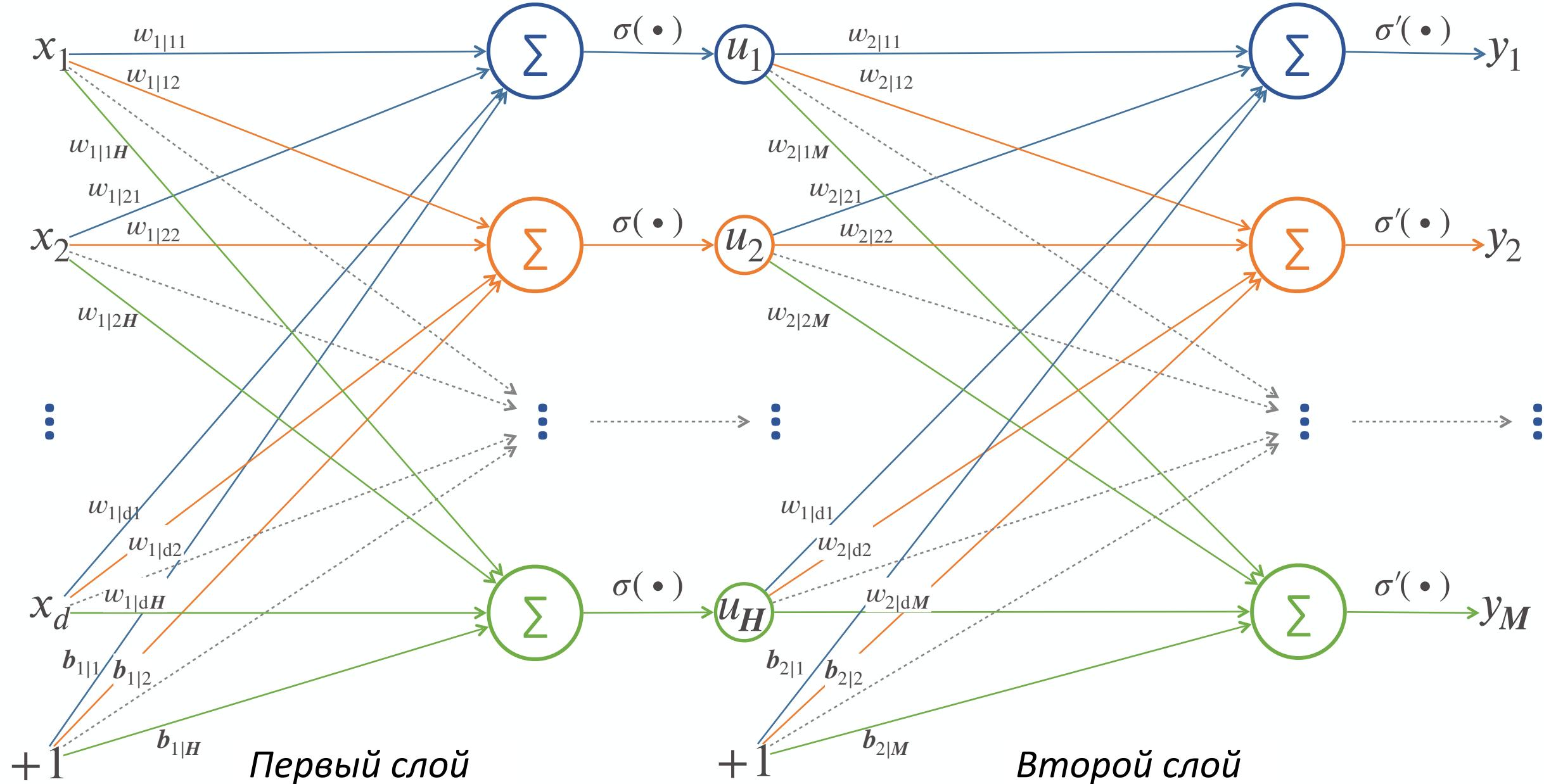
$y = (y_1, y_2, \dots, y_H)$  — выход слоя.

Тогда работу слоя можно описать операциями:

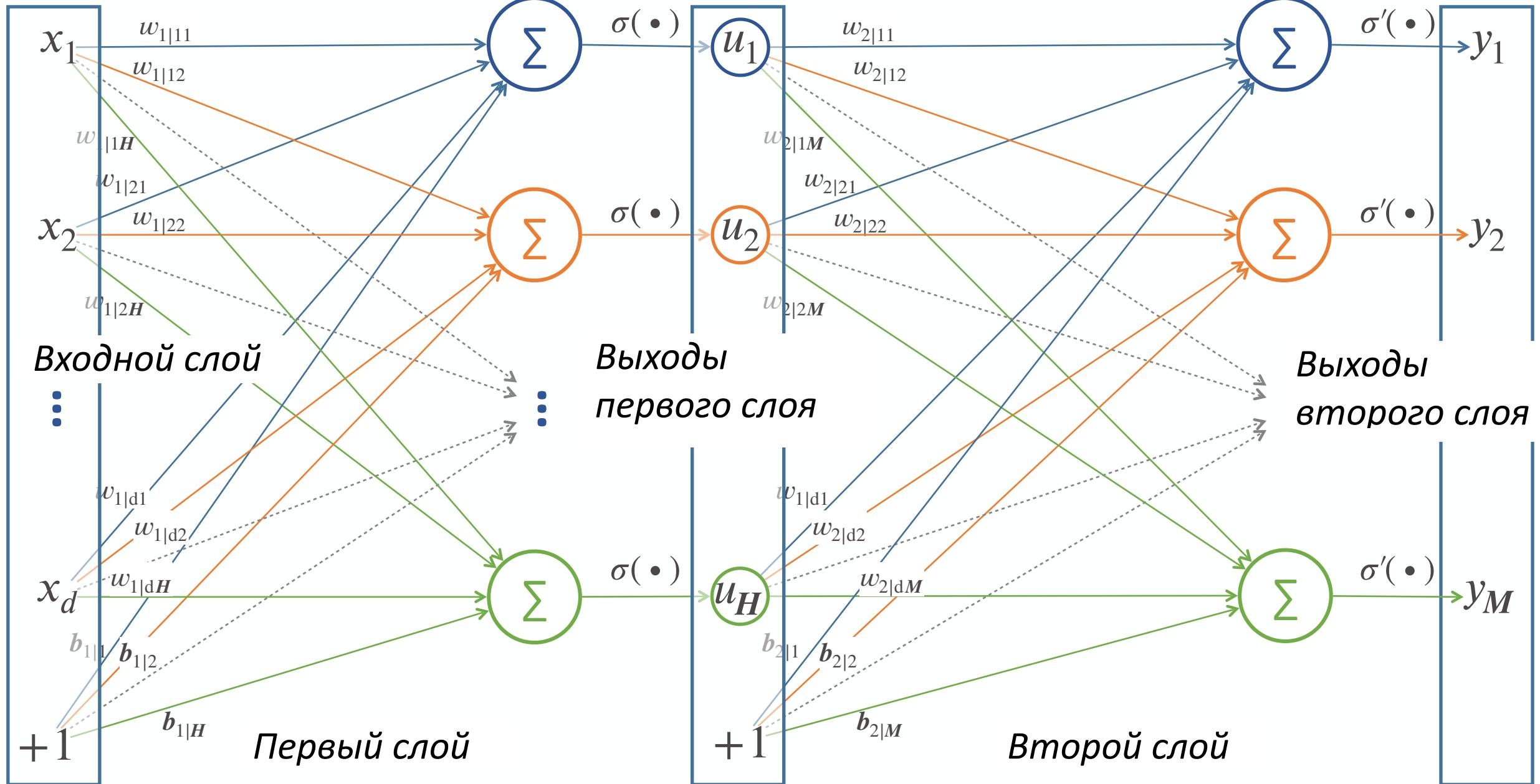
$$1) s = (x_1, x_2, \dots, x_d) \times \begin{pmatrix} w_{11} & \cdots & w_{1h} & \cdots & w_{1H} \\ w_{21} & \cdots & w_{2h} & \cdots & w_{2H} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{d1} & \cdots & w_{dh} & \cdots & w_{dH} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_H \end{pmatrix}^T = x^T \cdot W + b^T$$

$$2) y = (y_1, y_2, \dots, y_H) = (\sigma(s_1), \sigma(s_2), \dots, \sigma(s_H)) = \sigma(s) = \boxed{\sigma(x^T W + b^T)}$$

# Двухслойная нейронная сеть



# Двухслойная нейронная сеть





# Двухслойная нейронная сеть

## Матричное представление

Пусть  $x = (x_1, x_2, \dots, x_d)$  — элемент выборки,

$u = (s_1, s_2, \dots, s_H)$  — выход I слоя,  $y = (y_1, y_2, \dots, y_M)$  — выход II слоя,

$W_1 = (w_{1|jh})_{jh}$  — матрица весов I слоя,  $W_2 = (w_{2|hm})_{hm}$  — матрица весов II слоя,

$b_1 = (b_{1|1}, \dots, b_{1|H})^T$  — в-р сдвигов I слоя,  $b_2 = (b_{2|1}, \dots, b_{2|H})^T$  — вектор сдвигов II слоя.

Тогда работу двухслойной нейронной сети можно представить как:

$$1) \ u = \sigma_1(x^T W_1 + b_1^T)$$

$$2) \ y = \sigma_2(u^T W_2 + b_2^T) = \boxed{\sigma_2\left(\sigma_1\left(x^T W_1 + b_1^T\right)^T W_2 + b_2^T\right)}$$



# Двухслойная нейронная сеть

Назовем функцию  $\sigma(z)$  сигмоидой, если  $\lim_{z \rightarrow -\infty} \sigma(z) = 0$  и  $\lim_{z \rightarrow +\infty} \sigma(z) = 1$ .

$\sigma(z) = \frac{e^z}{1 + e^z}$  — логистическая сигмоида, один из примеров такой функции.

## Теорема (Цыбенко, 1989)

Если  $\sigma(z)$  — непрерывная сигмоида, то для любой непрерывной на  $[0, 1]^d$  функции  $f(x)$  существуют такое  $H$  и значения параметров  $w_{1|h} \in \mathbb{R}^d$ ,  $w_{2|h} \in \mathbb{R}^d$ ,  $b \in \mathbb{R}$ ,

что двухслойная нейросеть  $y(x) = \sum_{h=1}^H w_{2|h} \cdot \sigma(x^T w_{1|h} + b)$

равномерно приближает  $f(x)$  с любой точностью  $\varepsilon$ :  $|y(x) - f(x)| < \varepsilon \quad \forall x \in [0, 1]^d$



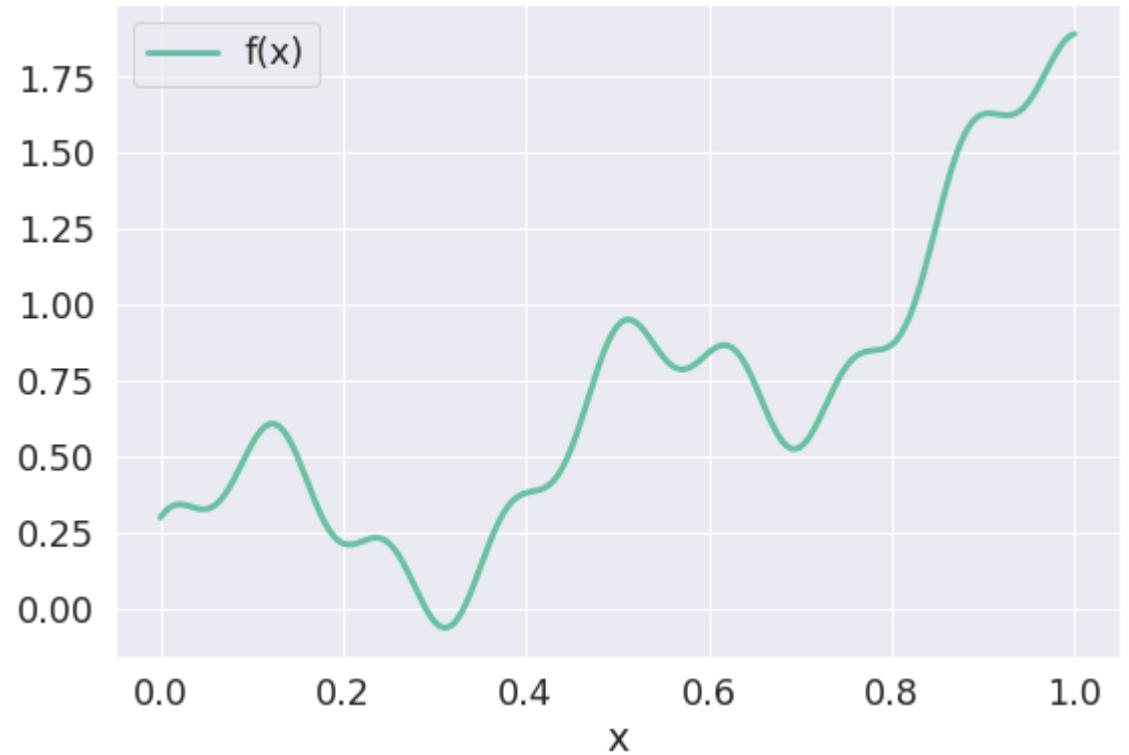
# План

- Введение, примеры применения
- Математическая модель нейросети
- **Приближение функции нейросетью**
- Обучение нейросети

# Пример

Возьмем  $f(x) = 0.2 + 0.4 x^2 + 0.3 x \sin(15 x) + 0.1 \cos(50 x)$ .

*Как аппроксимировать её нейронной сетью?*



# Пример

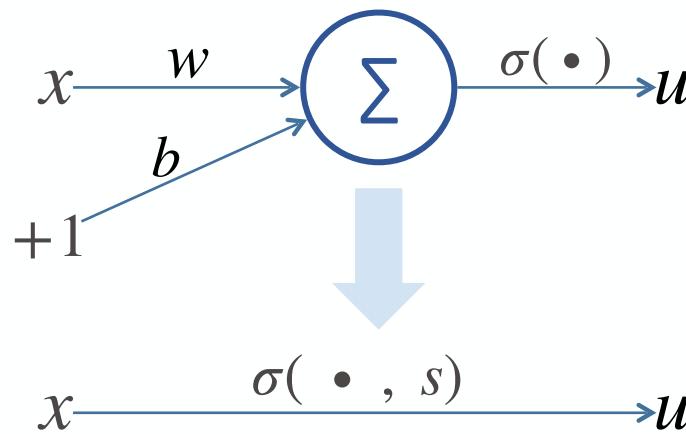
Как аппроксимировать  $f(x)$  нейронной сетью?

Рассмотрим **один нейрон**  $u = \sigma(wx + b)$  с функцией активации  $I\{z > 0\}$ .

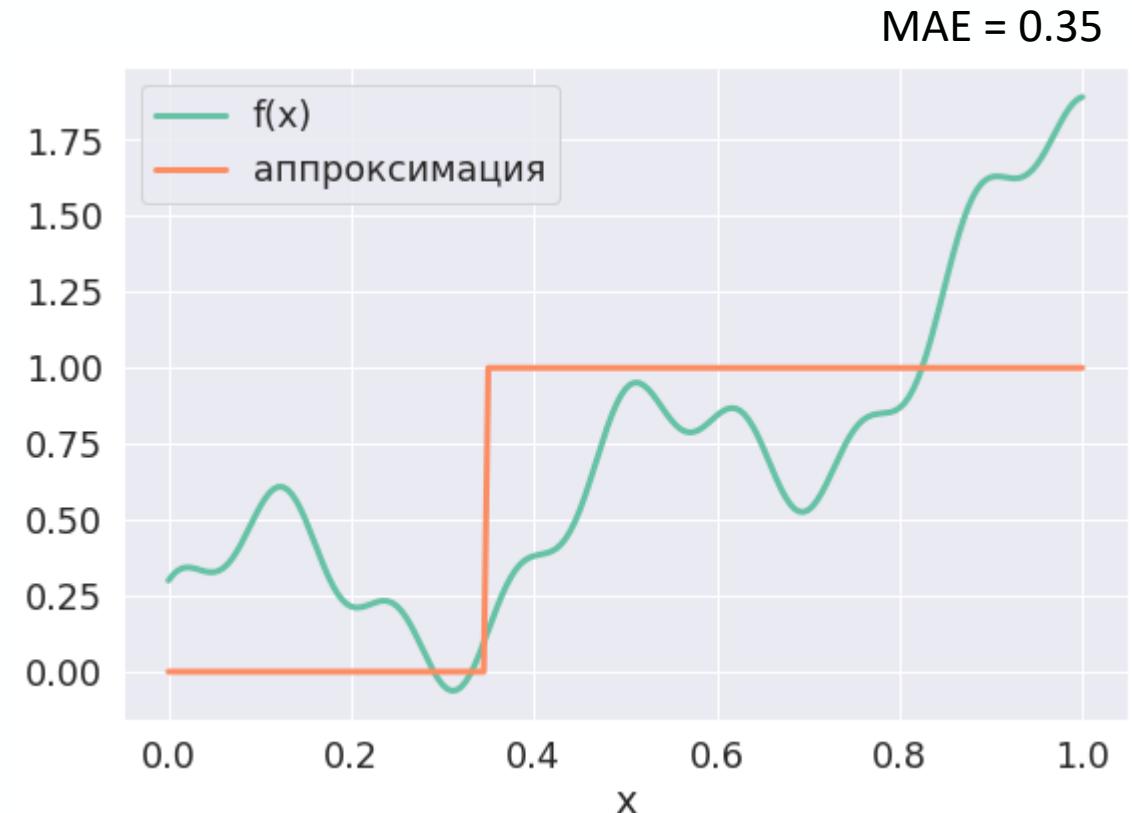
Заметим, что  $u > 0$  при  $x > -b/w$ . Обозначим  $s := -b/w$ .

Итоговая функция зависит только от  $s$

⇒ будем далее работать с  $s$ , а не с  $w, b$ .



Положим  $s = 0.35$ .

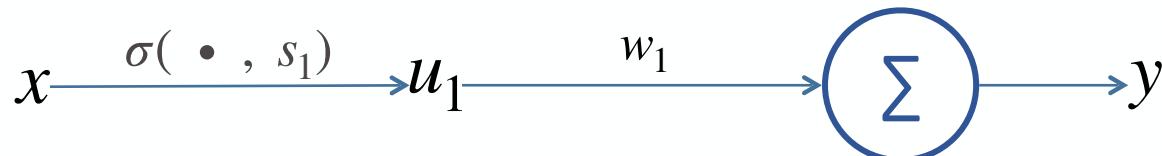


# Пример

Как аппроксимировать  $f(x)$  нейронной сетью?

**Добавим еще один слой.** Теперь у нас двухслойная нейронная сеть.

На втором слое один нейрон  $y = wi$ .



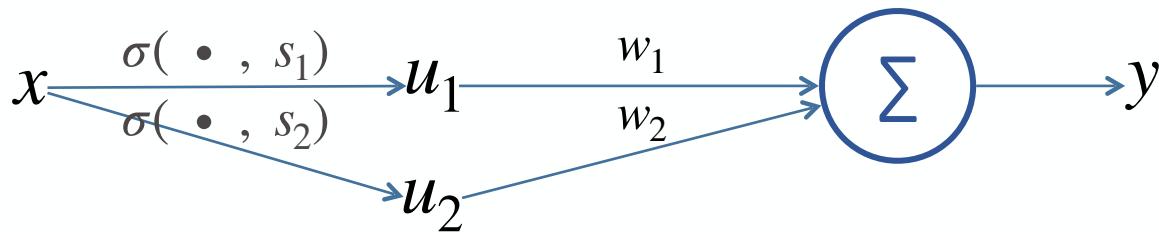
Положим  $w = 0.8$ .



# Пример

Как аппроксимировать  $f(x)$  нейронной сетью?

Добавим по нейрону на первом и втором слое.



Положим  $s_2 = 0.7$ ,  $w_2 = -w_1 = -0.8$ .

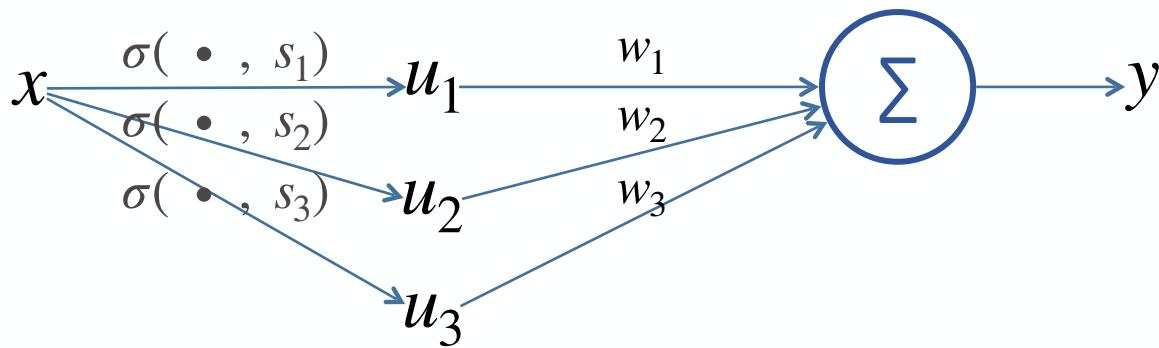
Получилась **ступенька**,  
которая приближает часть функции.



# Пример

Как аппроксимировать  $f(x)$  нейронной сетью?

Добавим еще по нейрону на первом и втором слое.



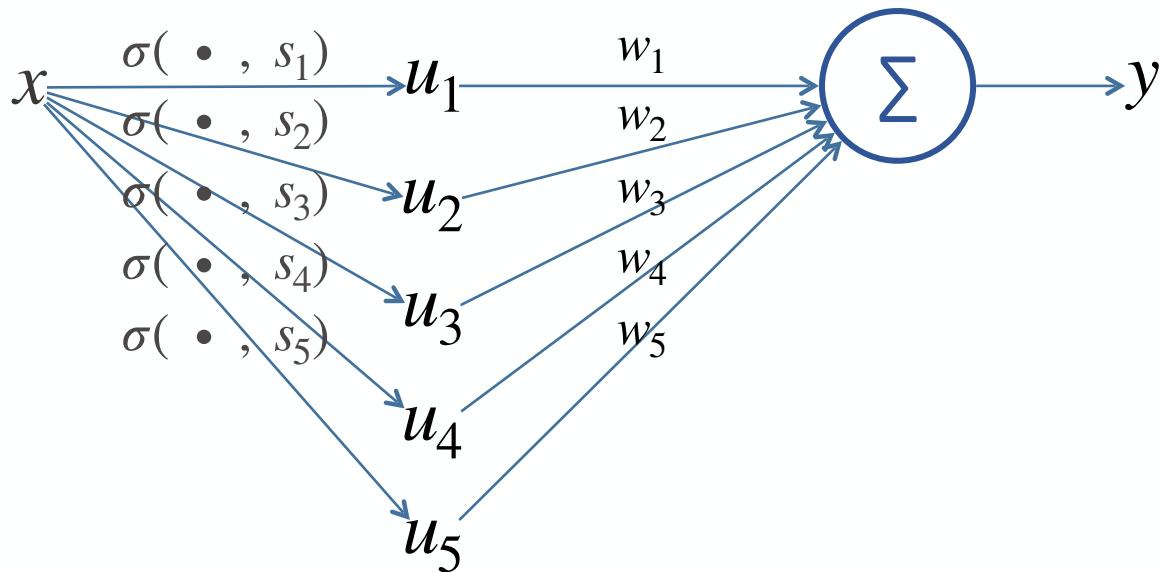
Положим  $s_3 = s_2 = 0.7, w_3 = 1.5$ .



# Пример

Как аппроксимировать  $f(x)$  нейронной сетью?

**Добавим еще по два нейрона на первом и втором слое.**



Положим  $s_4 = 0, s_5 = s_1 = 0.35,$

$$w_4 = -w_5 = 0.4 .$$





# Пример

## Вывод

- С помощью двухслойной нейронной сети получилось **аппроксимировать сложную функцию.**
- Мы **не прибегали к генерации сложных признаков**, которые понадобились бы, например, для линейной регрессии.
- Если **увеличить число нейронов**, то приближение получается **более точным**.

Но! Сейчас мы подбирали параметры самостоятельно.





# Двухслойная нейронная сеть

## Выводы

- С помощью линейных операций и функций активаций  $\sigma$  от одного аргумента можно вычислять любую непрерывную функцию на заданном интервале с любой желаемой точностью.
- Двух слоев в нейронной сети теоретически достаточно.

## Замечания

- Теорема ничего не говорит о количестве нейронов в каждом слое, о значении весов и сдвигов, и виде функции активации.
- Двумя слоями такая цель теоретически достигается, но сложно. Только в одной коре головного мозга число слоев равно 6.
- Дополнительные слои — удобный способ преобразования признаков, переход из одного признакового пространства в более удобное для решения задачи.



# План

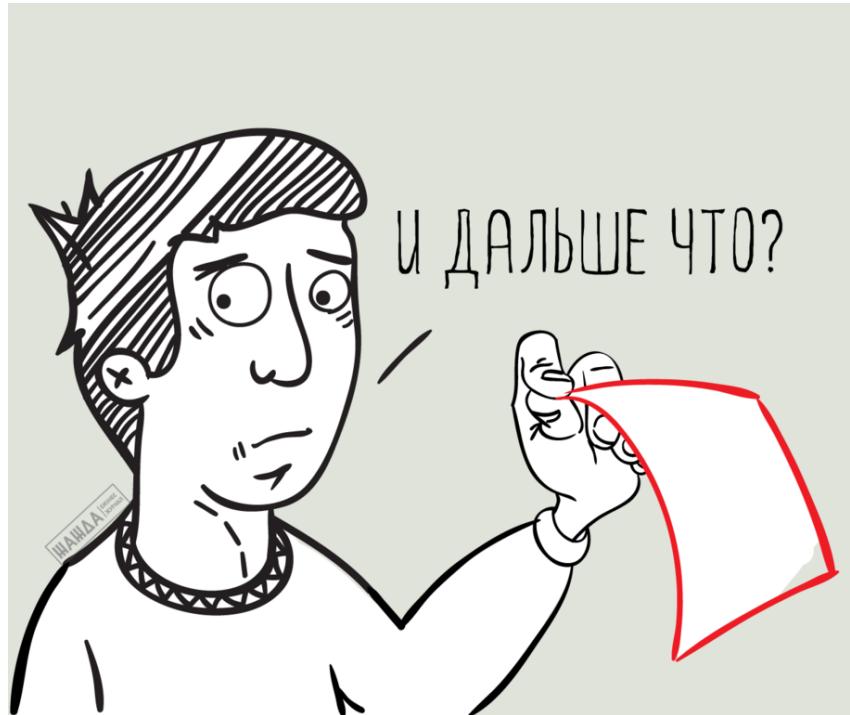
- Введение, примеры применения
- Математическая модель нейросети
- Приближение функции нейросетью
- **Обучение нейросети**



# Нейронная сеть

Мы параметризовали модель нейронной сети.

Из теоремы Цыбенко вытекает, что существуют параметры, при которых мы сможем аппроксимировать любую непр. функцию на заданном интервале.



Будем находить параметры!

А как?

С помощью **градиентного спуска**!  
По аналогии с решением задачи  
линейной регрессии.



# Обучение двухслойной нейронной сети. Пример

- Задаем модель:  $\hat{y} = \sigma_2 \left( \sigma_1 \left( x^T W_1 + b_1 \right)^T W_2 + b_2 \right)$  – формула для всех элементов выборки.
- Задаем функцию, которую будем минимизировать, например MSE:

$$L = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \text{ где } i \text{ — номер элемента выборки.}$$

- Обозначим все параметры сети как  $\theta$ .

Находим параметры с помощью градиентного спуска:

$$\theta_t = \theta_{t-1} - \eta \nabla L(\theta_{t-1}), \text{ где } \eta \text{ — скорость обучения.}$$



# Обучение нейронной сети

## Виды градиентного спуска

- **Batch Gradient Descent**

Разбиваем данные на блоки (батчи).

$$\theta_t = \theta_{t-1} - \eta \nabla \left( \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\hat{y}_{\theta_{t-1}}(x_i), Y_i) \right)$$

Для каждого блока считаем градиенты и накапливаем их.

Производим обновление параметров после подсчета градиента по всем данным.

- **Mini-batch Gradient Descent**

Разбиваем данные на блоки (батчи).

$$\theta_t = \theta_{t-1} - \eta \nabla \left( \frac{1}{b} \sum_{b=1}^B \mathcal{L}(\hat{y}_{\theta_{t-1}}(x_{i_b}), Y_{i_b}) \right)$$

Для каждого блока считаем градиент и обновляем параметры.

$(x_{i_1}, \dots, x_{i_b})$  — текущий батч. **Самый популярный метод.**

- **Стохастический градиентный спуск**

Выбираем один элемент  $x_i$  случайно.

Считаем градиент и производим обновление.

Эквивалентно Mini-Batch GD для размера батча, равного 1.

$$\theta_t = \theta_{t-1} - \eta \nabla (\mathcal{L}(\hat{y}_{\theta_{t-1}}(x_i), Y_i))$$

# Задачи (доска)

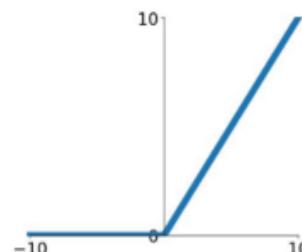
1. Данна нейронная сеть из 2 слоев в матричном виде:

$$\hat{y} = \tanh\left(\text{ReLU}\left(XW_1 + b_1^T\right)W_2 + b_2^T\right) \in \mathbb{R}^n, \text{ где}$$

батч  $X \in \mathbb{R}^{n \times d}$ ,  $W_1 \in \mathbb{R}^{d \times H}$ ,  $b_1 \in \mathbb{R}^H$ ,  $W_2 \in \mathbb{R}^{H \times 1}$ ,  $b_2 \in \mathbb{R}$ . Посчитайте правила обновления параметров  $W_i, b_i$  в Mini-Batch Gradient Descent.

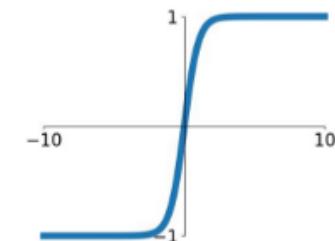
$$\text{ReLU}(t) = tI\{t \geq 0\}$$

$$\text{ReLU}'(t) = I\{t \geq 0\}$$



$$\tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$$

$$\tanh'(t) = 1 - \tanh^2(t)$$





# Задачи (доска)

1. Данна нейронная сеть из 2 слоев в матричном виде:

$$\hat{y} = \tanh \left( \text{ReLU} \left( XW_1 + b_1^T \right) W_2 + b_2^T \right) \in \mathbb{R}^n, \text{ где}$$

батч  $X \in \mathbb{R}^{n \times d}$ ,  $W_1 \in \mathbb{R}^{d \times H}$ ,  $b_1 \in \mathbb{R}^H$ ,  $W_2 \in \mathbb{R}^{H \times 1}$ ,  $b_2 \in \mathbb{R}$ . Посчитайте правила обновления параметров  $W_i, b_i$  в Mini-Batch Gradient Descent.

2. Рассмотрим нейронную сеть с 100 слоями и функцией активации  $\tanh$ . Найдите  $\frac{\partial L}{\partial X}$ , где  $X$  - входная матрица признаков.



# Решения

1. Параметры  $W_1, b_1, W_2, b_2$  обновляются по правилу  $W_i := W_i - \eta \cdot \frac{\partial L}{\partial W_i}; b_i := b_i - \eta \cdot \frac{\partial L}{\partial b_i}; i = 1, 2$  (\*)

Нужно найти градиенты, что проще делать в порядке от последних слоев к первым.

Зафиксируем  $L$  как MSE, тогда  $\frac{\partial L}{\partial \hat{y}} = \frac{2}{n}(\hat{y} - Y)^T$ . Обозначим  $Z_1 = XW_1 + b_1^T$ ;  $A_1 = \text{ReLU}(Z_1)$ ;  $Z_2 = A_1W_2 + b_2 \Rightarrow \hat{y} = \tanh(Z_2)$ .

1) Градиент 2 слоя:

$$\frac{\partial L}{\partial Z_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial Z_2} = \frac{\partial L}{\partial \hat{y}} \cdot (1 - \tanh^2(Z_2))$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial W_2} = \frac{\partial L}{\partial Z_2} \cdot A_1$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial b_2} = \sum_{j=1}^n \frac{\partial L}{\partial (Z_2^j)}$$

2) Градиент 1 слоя:

$$\frac{\partial L}{\partial Z_1} = \frac{\partial L}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1} = \left( \left( \frac{\partial L}{\partial Z_2} \right)^T \cdot W_2^T \cdot I\{x \geq 0\} \right)^T$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial W_1} = \frac{\partial L}{\partial Z_1} \cdot X$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial b_1} = \sum_{j=1}^n \frac{\partial L}{\partial (Z_1^j)}$$

Осталось подставить эти градиенты в формулу пересчета параметров (\*).

В формулах выше первое равенство - применение Chain Rule (формулы дифференцирования сложной функции). А второе равенство - выражение после подстановления значений градиентов в эту формулу, при этом на этом шаге мы разрешаем себе транспонировать некоторые выражения. Дело в том, что в общем случае производная матрицы по матрице - тензор размерности  $>2$ . Здесь же имеем дело с частными случаями, иногда приходится транспонировать матрицы, чтобы оставаться в 2-мерном пространстве.

А вообще, цель этой задачи – осознать, что градиенты выражаются друг через друга, если идти в порядке от последних слоев к первым.



# Решения

2. Рассмотрим нейронную сеть с 100 слоями и функцией активации  $\tanh$ . Давайте здесь найдем  $\frac{\partial L}{\partial W_1}$  вместо  $\frac{\partial L}{\partial X}$ : градиент лосса по весам более важен для обучения сети, а градиент лосса по входам больше нужен для анализа поведения сети. Наша цель – показать, что при использовании глубоких сетей с функцией активации  $\tanh$  на каждом слое градиенты, используемые для обновления сети, становятся чрезвычайно маленькими или «исчезают» при backward pass с выходных слоев на более ранние слои (Vanishing Gradient problem).

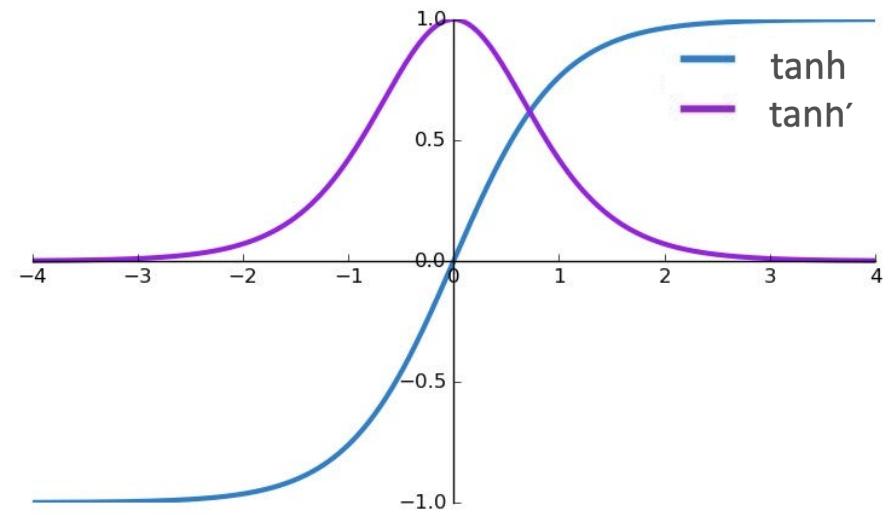
Введем обозначения:

$$\begin{cases} A_0 := X \\ Z_i := A_{i-1} \cdot W_i + b_i \\ A_i := \tanh(Z_i) \end{cases}$$

Здесь финальное предсказание  $\hat{y} \equiv A_N$ . Применим Chain Rule:

$$\begin{aligned} \frac{\partial L}{\partial W_1} &= \frac{\partial L}{\partial A_N} \cdot \frac{\partial A_N}{\partial Z_N} \cdot \frac{\partial Z_N}{\partial A_{N-1}} \cdot \frac{\partial A_{N-1}}{\partial Z_{N-1}} \cdot \dots \cdot \frac{\partial A_1}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial W_1} = \\ &= \frac{\partial L}{\partial A_N} \cdot \underbrace{\tanh'(Z_N)}_{\text{orange wavy line}} \cdot \underbrace{W_N^T}_{\text{orange wavy line}} \cdot \underbrace{\tanh'(Z_{N-1})}_{\text{orange wavy line}} \cdot \dots \cdot \underbrace{\tanh'(Z_1)}_{\text{orange wavy line}} \cdot X. \end{aligned}$$

Но  $\tanh'(t)$  очень быстро стремится к нулю при  $|t| \rightarrow \infty$  ( $\tanh'(t) < 10^{-8}$  при  $|t| > 10$ ). Поэтому значение градиента  $\frac{\partial L}{\partial W_1}$  с большой вероятностью станет зануляться во время обучения, что приведет к “параличу” сети. Таким же свойством обладает логистическая сигмоида, у которой тоже пологие концы. Поэтому популярны такие функции активации, как ReLU.





BCE!