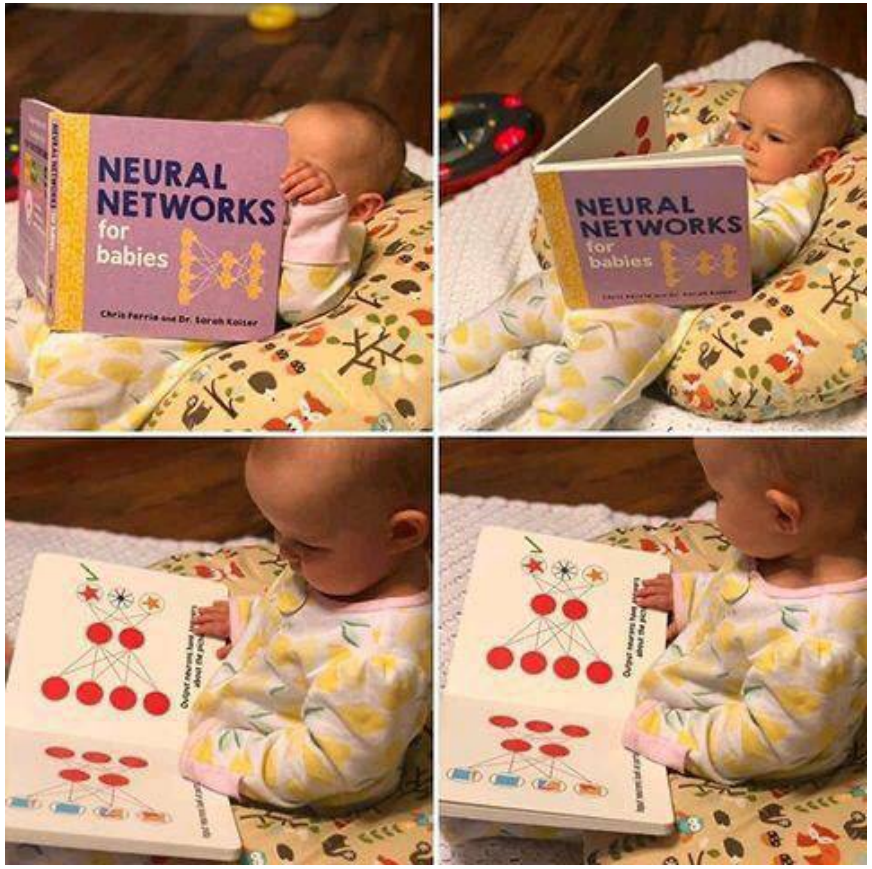
Введение в анализ данных



Введение в нейронные сети

Введение в нейронные сети

- **Определение нейронной сети**
- Модель нейронной сети
- Обучение нейронной сети
- Применение нейронных сетей

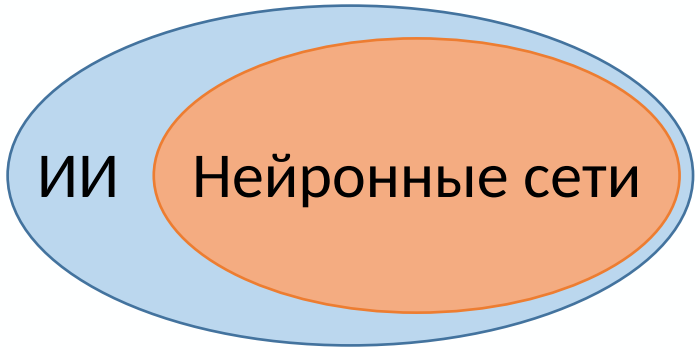
Искусственный интеллект

Что такое ИИ?

Искусственный интеллект (ИИ) — это область компьютерных наук. Специалисты по ИИ создают системы и программы, которые анализируют информацию и решают задачи аналогично тому, как это делает человек.

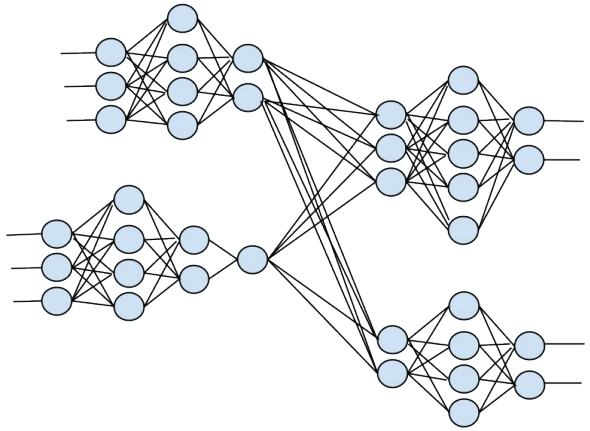
ИИ и нейронные сети

ИИ иногда путают с нейронными сетями. Но это лишь отчасти верно.
Нейросети — это один из наиболее популярных подходов достижения ИИ.

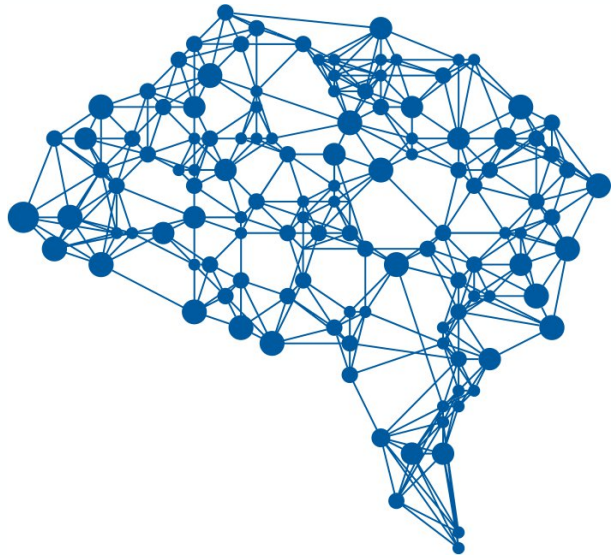


Что такое ~~эти ваши~~ нейронные сети?

Нейронная сеть / нейросеть — математическая модель (также ее программное или аппаратное воплощение), которая строится по принципу работы *мозга человека* и применяется для *анализа*, *интерпретации и генерации различных* данных, включая визуальную, текстовую и другие виды информации.



~

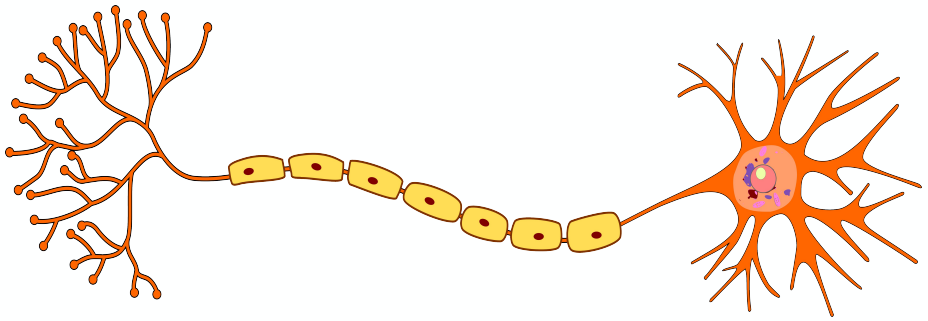


Как работает мозг?

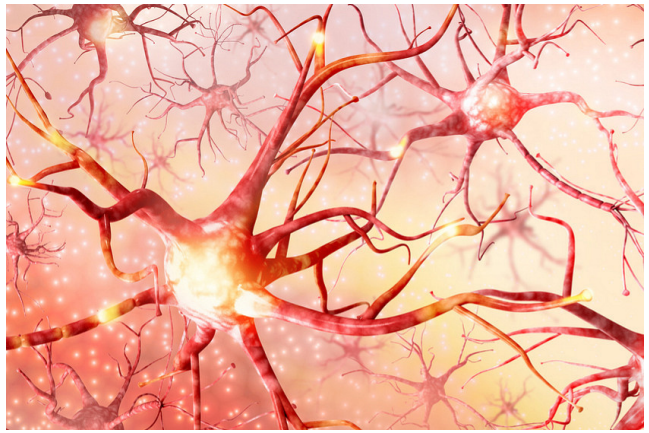
Основные клетки мозга — это **нейроны**.

- Они **хранят и передают информацию** через *электр. и хим. сигналы*.
- Каждый нейрон может образовывать тысячи связей с другими нейронами при помощи синапсов, образуя сложные **нейронные сети**.
- Нейронные сети **способны к обучению** — синапсы могут усиливаться или ослабляться в зависимости от сигнала. Мозг учится через обратную связь: успешные действия закрепляются, а ошибки приводят к изменению стратегий.

Биологический нейрон

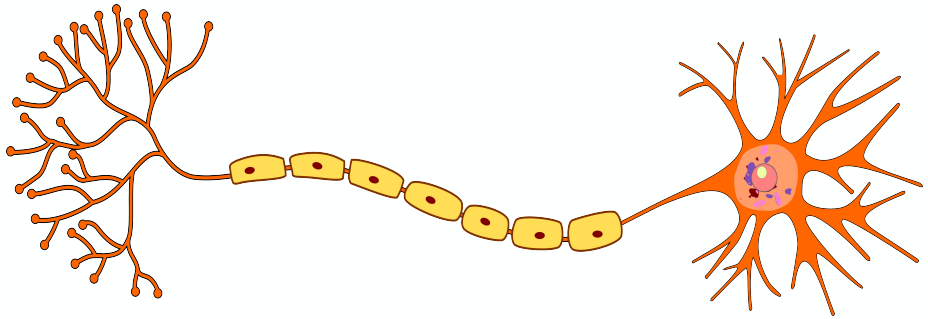


Биологическая нейронная сеть



Бионика: нейронные сети

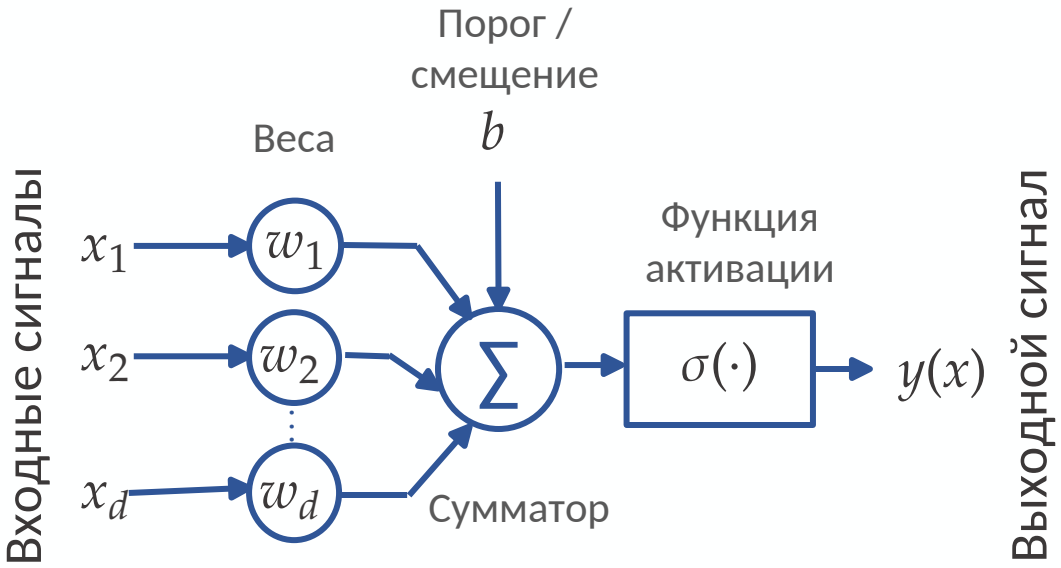
Биологический нейрон



Синапсы

Аксон

Искусственный нейрон



Синапсы

Аксон

Историческая справка

1943 г. — Мак-Каллок и Питц

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

1949 г. — Хебб

- Открытие принципа формирования биологических нейронных связей, взаимодействия нейронов.

1958 г. — Розенблатт

- Открытие «персептрона» — устройства, моделирующего процесс человеческого восприятия

Введение в нейронные сети

- Определение нейронной сети
- **Модель нейронной сети**
- Обучение нейронной сети
- Применение нейронных сетей

Модель нейрона

Обозначим

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

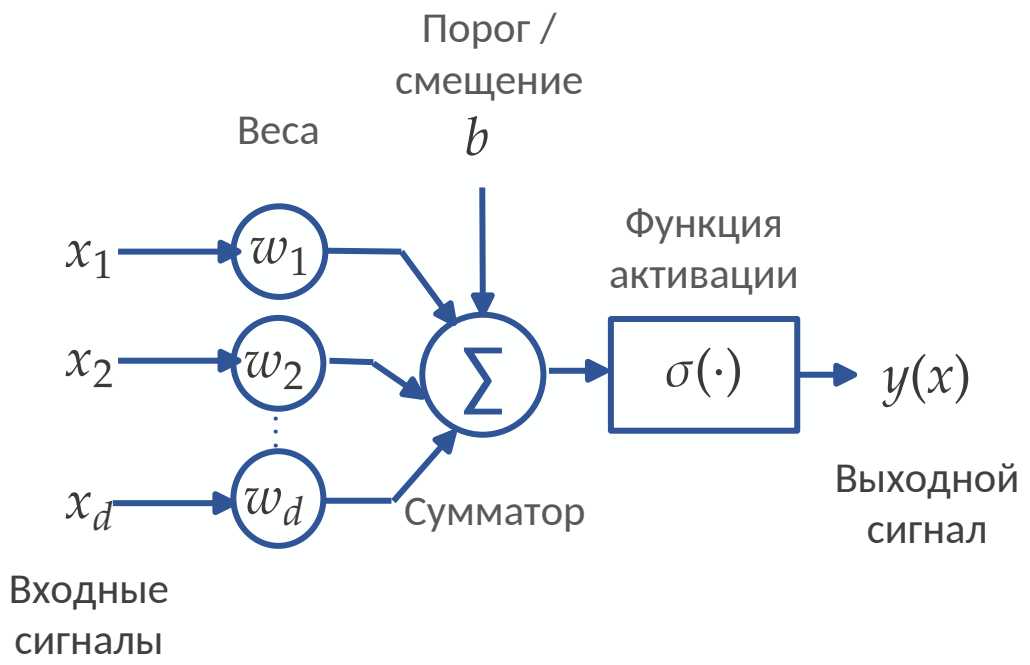
$w = (w_1, \ldots, w_d)^T \in \mathbb{R}^d$ — вектор весов;

$b \in \mathbb{R}$ — смещение.

Выход нейрона —

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

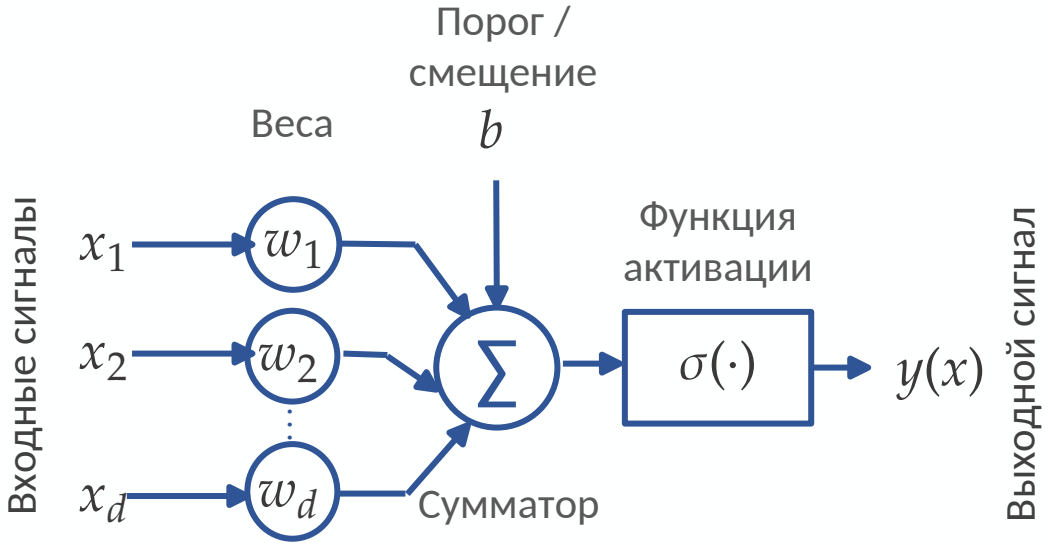
где σ — некоторая кус.-дифф. функция, назовем ее **функцией активации**.



Что-то знакомое...

На что похожа эта формула?

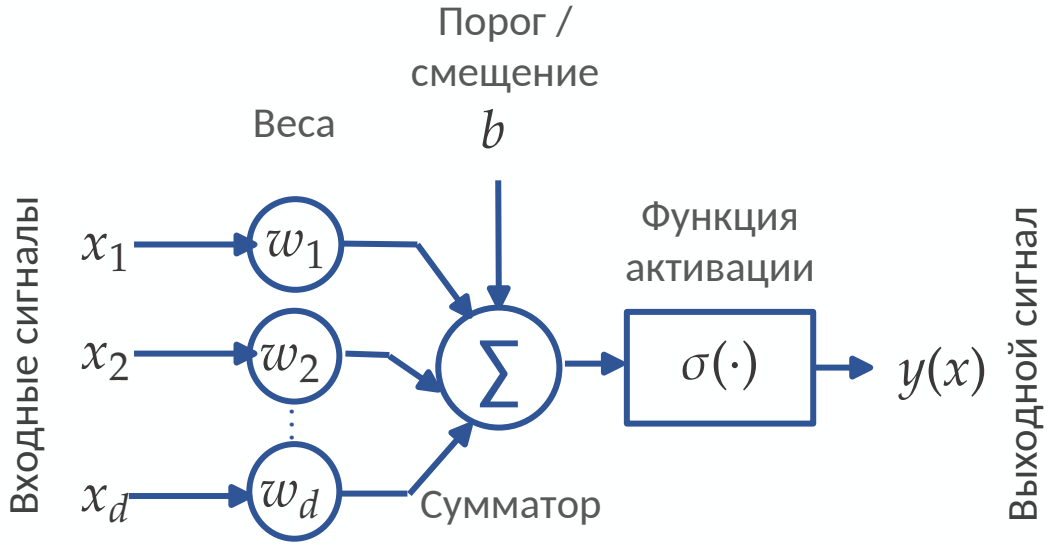
$$y(x) = \sigma(x^T w + b) = \sigma\left(\sum_{j=1}^{d} x_j w_j + b\right)$$



Что-то знакомое...

На что похожа эта формула?

$$y(x) = \sigma(x^T w + b)$$



Линейная регрессия

$$y(x) = x^T w + b = \sigma\left(x^T w + b\right),$$ где $\sigma(z) = z$ — линейная ф-я.

Логистическая регрессия

$$y(x) = \sigma\left(x^T w + b\right),$$ где $\sigma(z) = \frac{1}{1+e^{-z}}$ — логист. сигмоида.

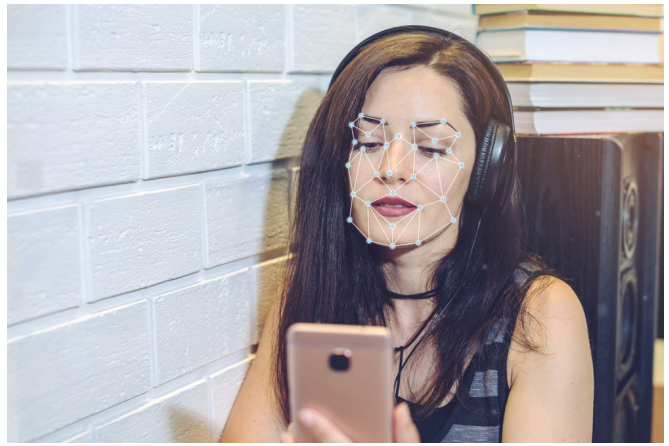
Более сложные задачи

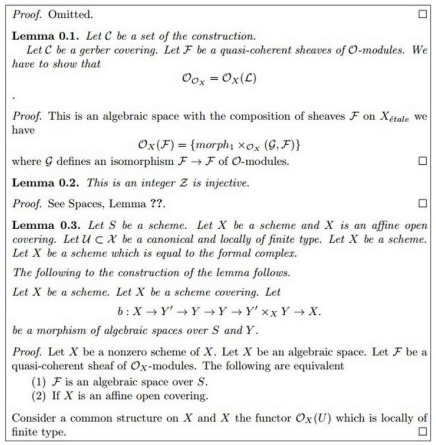
- У линейной и логистической регрессий ограниченная область применения.
- Для того, чтобы решить нелинейную задачу,

  нужно делать сложные преобразования с признаками.
- Один нейрон не справится со сложными задачами... ☹

Вспомним, что в нервной системе очень много нейронов.

Значит, будем использовать **больше нейронов**!





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 $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ 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

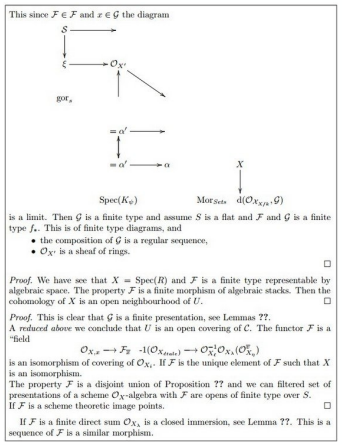
$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

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

$$\begin{tikzcd}
\mathcal{S} \arrow[r] \arrow[d] & \\
\xi \arrow[r] & \mathcal{O}_{X'} \arrow[dr] & \\
\mathrm{gor}_s \arrow[u] & & \\
= \alpha' \arrow[r] \arrow[d, leftrightarrow] & & \\
= \alpha' \arrow[r] & \alpha & X \arrow[d] \\
\mathrm{Spec}(K_\psi) & \mathrm{Mor}_{Sets} & \mathrm{d}(\mathcal{O}_{X_{X/k}}, \mathcal{G})
\end{tikzcd}$$

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 $X = \mathrm{Spec}(R)$ 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

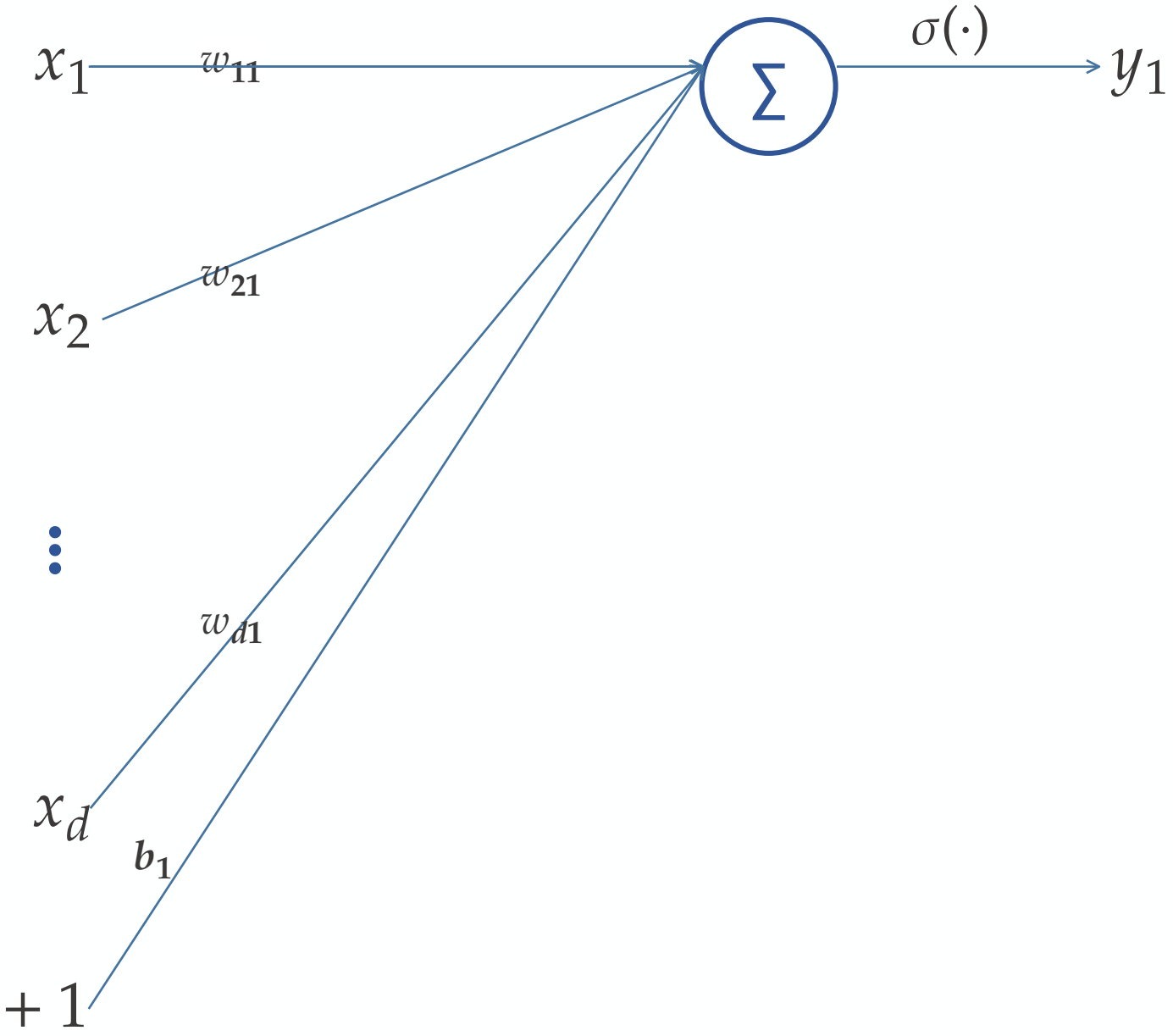
$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \text{ -1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{x}})$$

is an isomorphism of covering of $\mathcal{O}_{X_\ell}$. 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 $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

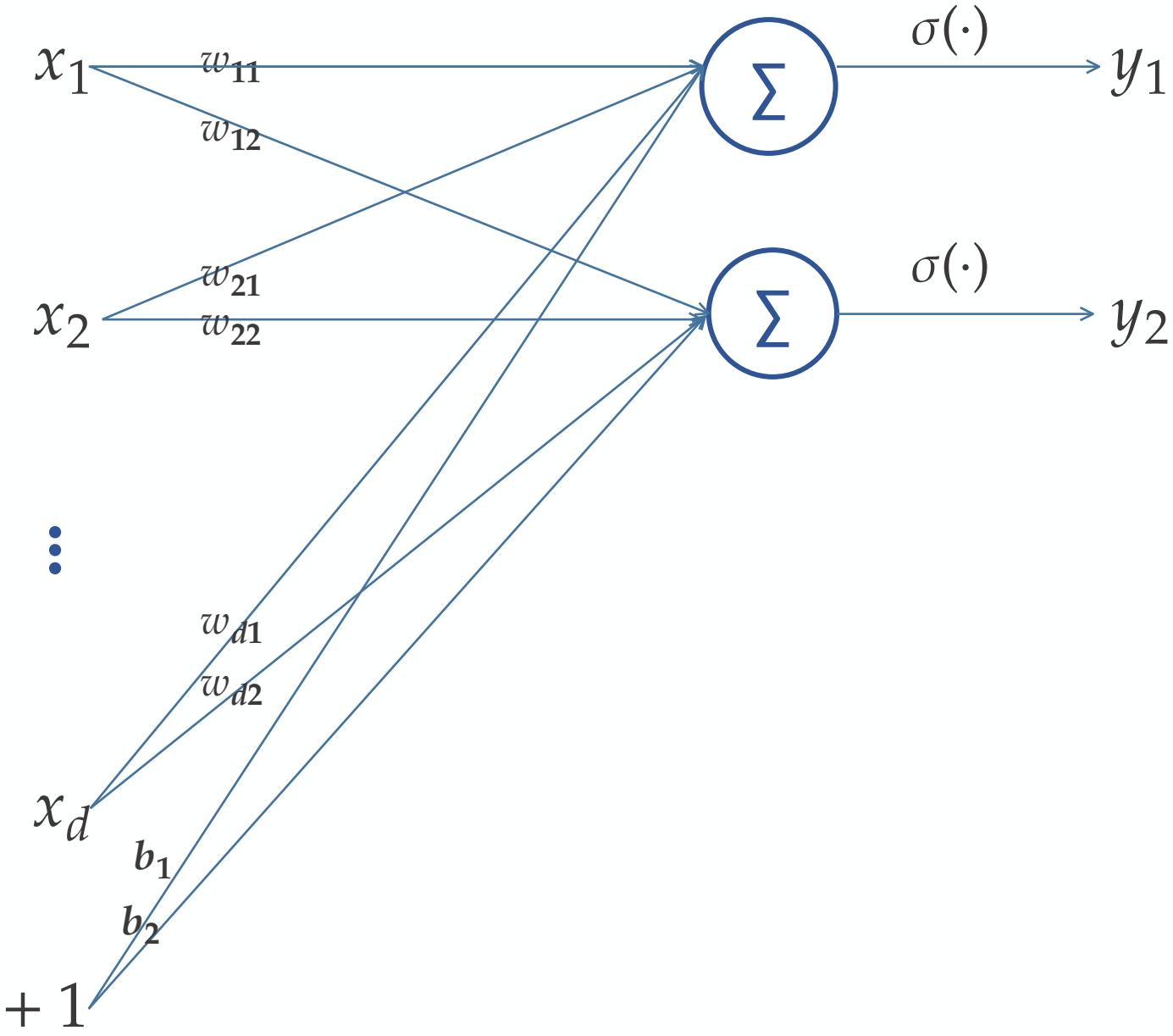
Однослойная нейронная сеть

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



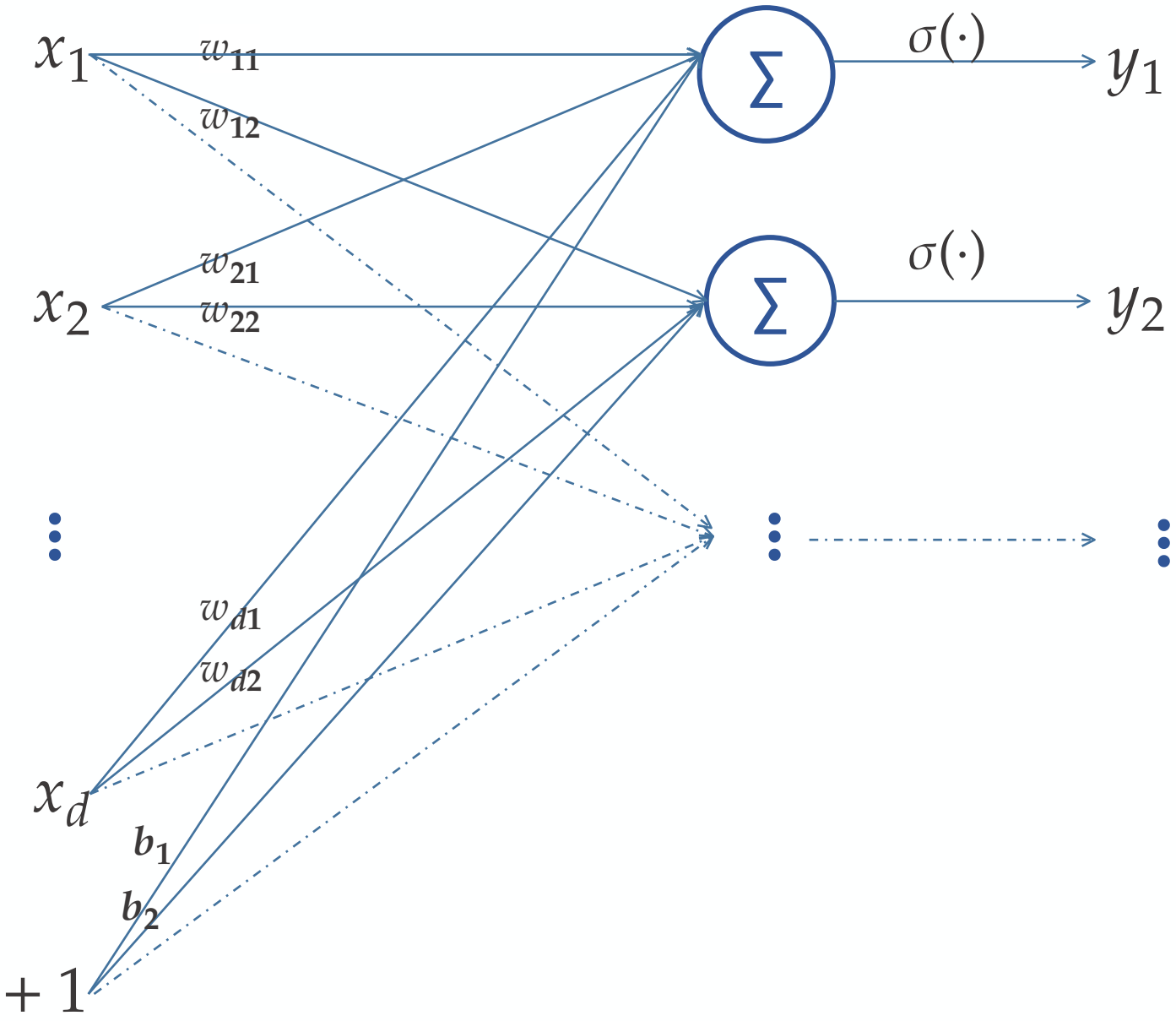
Однослойная нейронная сеть

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

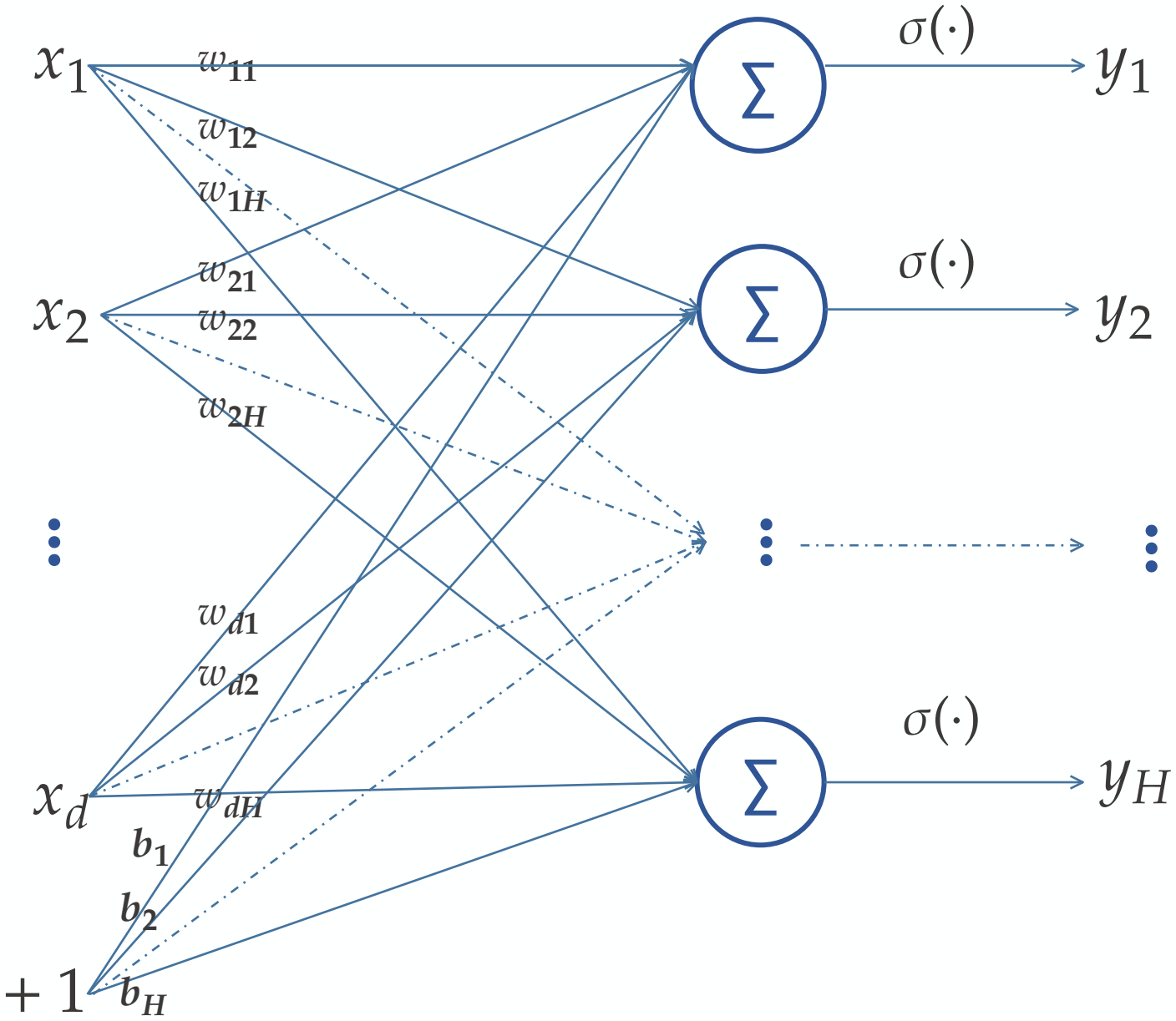


Однослойная нейронная сеть

Больше нейронов!



Однослойная нейронная сеть

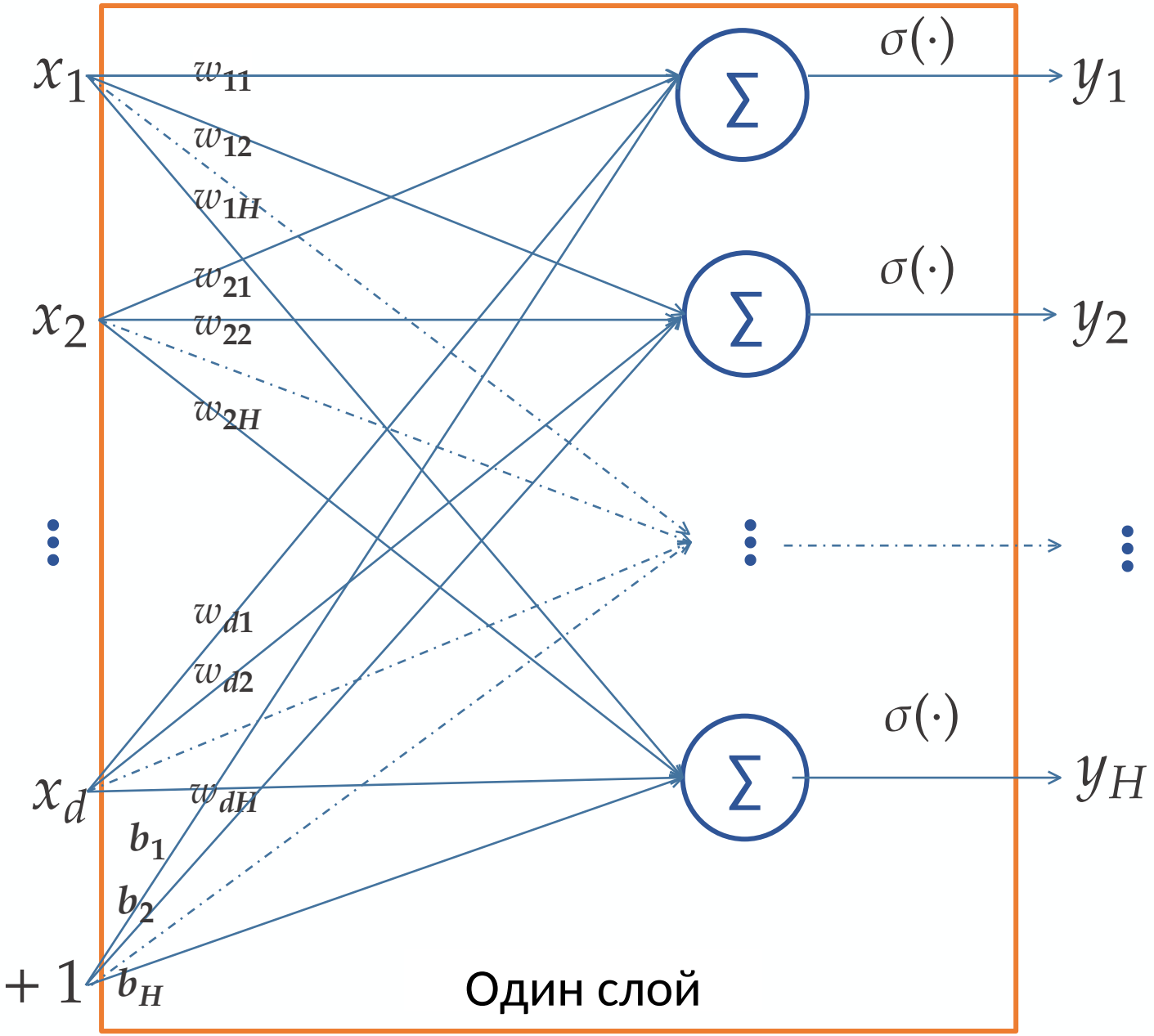


Однослойная нейронная сеть

Слой размера H :
набор из H нейронов.

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

Такую нейронную сеть или слой нейронной сети называют **полносвязной**(ым)



Однослойная нейронная сеть

Матричное представление

Пусть $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$ — элемент выборки.

$s = (s_1, \dots, s_H)^T \in \mathbb{R}^H$ — выходы нейронов до применения функции активации.

$y = (y_1, \dots, y_H)^T \in \mathbb{R}^H$ — выход слоя,

$W = (w_{jh})_{jh} \in \mathbb{R}^{d \times H}$ — м-ца весов,

$b = (b_1, \dots, b_H)^T \in \mathbb{R}^H$ — сдвиг.

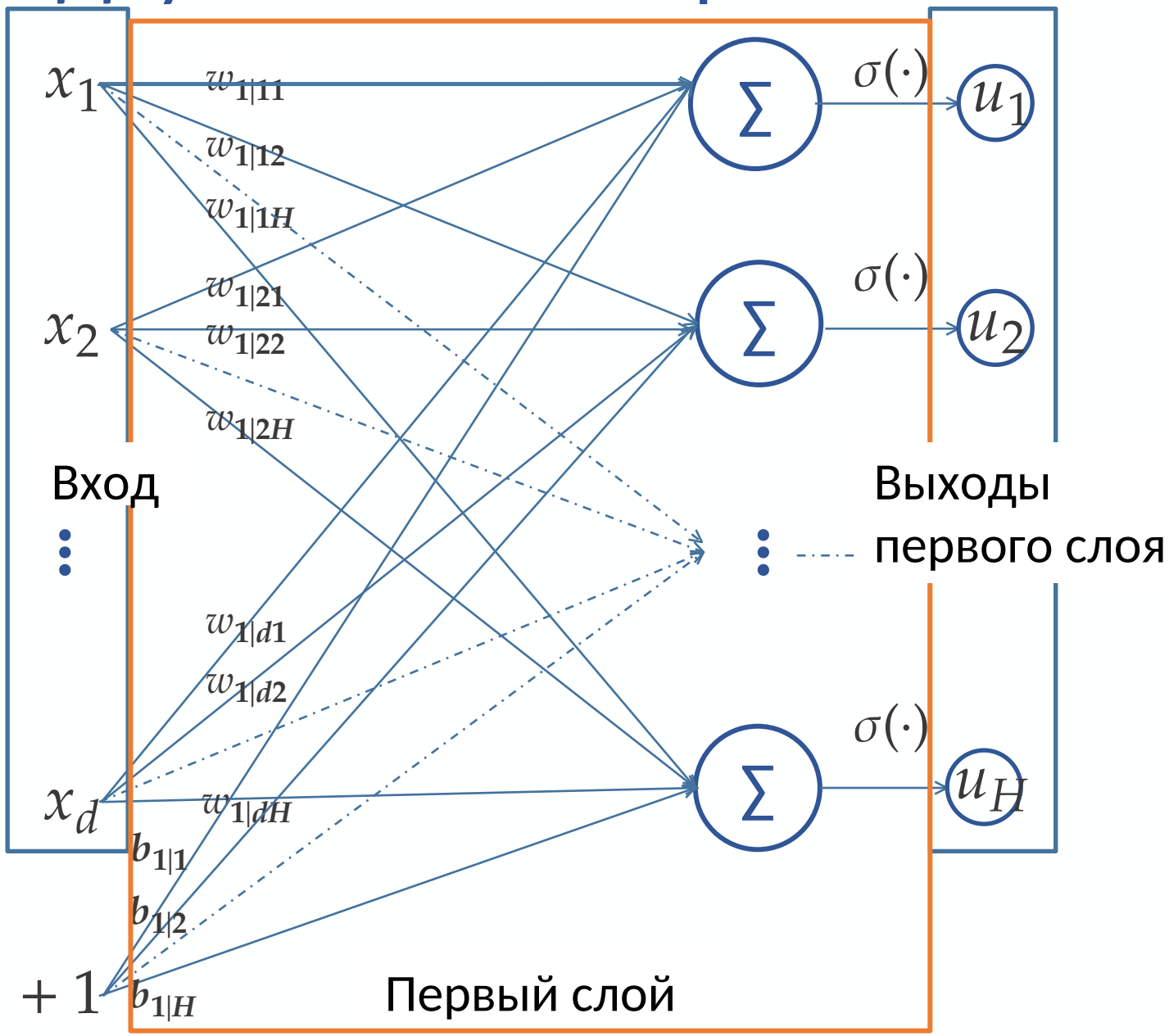
Тогда работу слоя можно описать операциями:

1) $s^T = (x_1, \cdots, x_d) \times \begin{pmatrix} w_{11} & \cdots & w_{1H} \\ \vdots & \ddots & \vdots \\ w_{d1} & \cdots & w_{dH} \end{pmatrix} + (b_1, \dots, b_H) = x^T W + b^T$

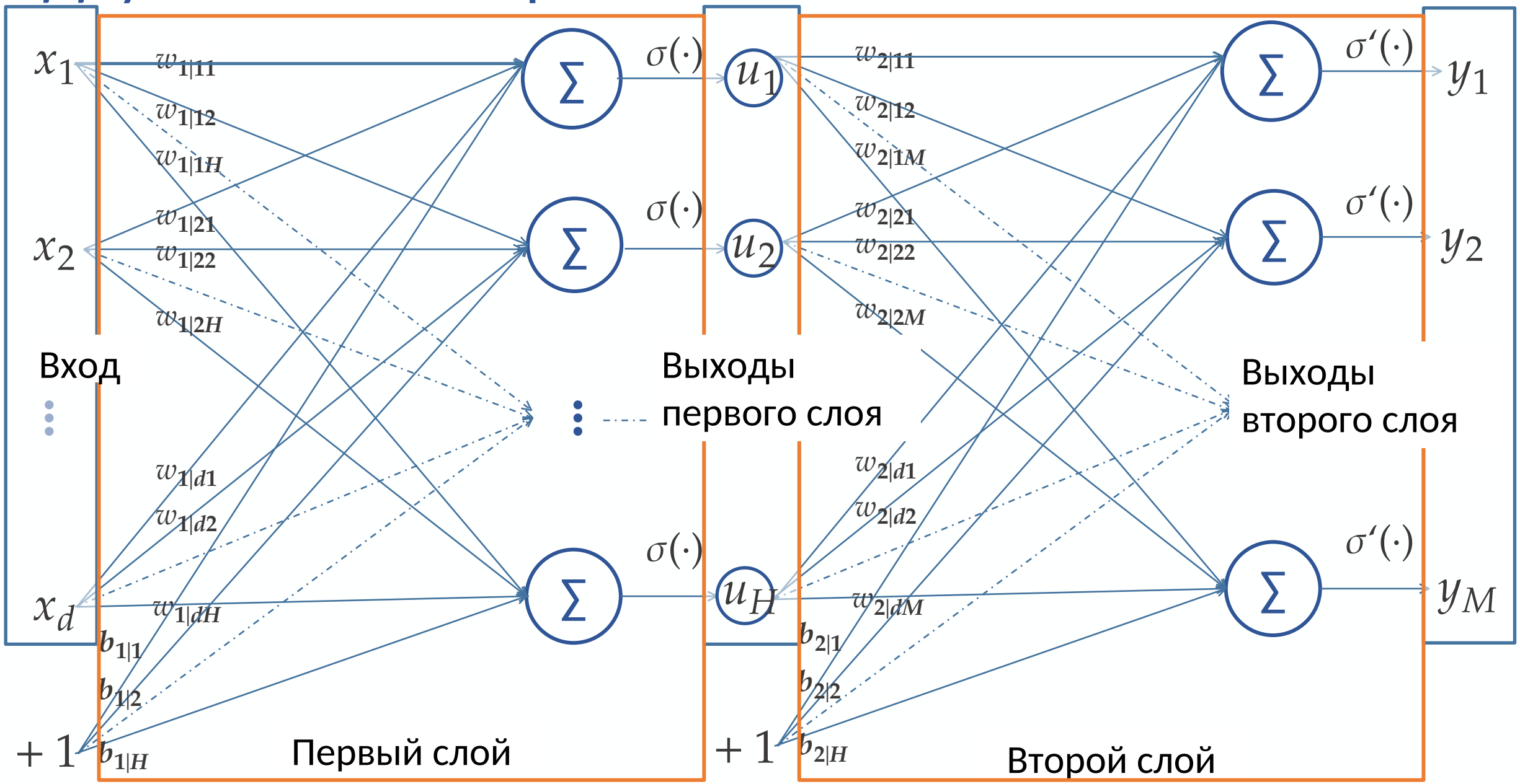
2) $y^T = (y_1, \dots, y_H) = (\sigma(s_1), \sigma(s_2), \dots, \sigma(s_H)) = \sigma\left(s^T\right) = \sigma(x^T W + b^T)$ ← *Здесь и далее активации применяем поэлементно*

Таким образом модель 1-сл. нейронной сети — это **функция** $\boxed{y(x) = \sigma(x^T W + b^T)}$.

Двухслойная нейронная сеть



Двухслойная нейронная сеть



Двухслойная нейронная сеть

Матричное представление

Пусть $x = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ — элемент выборки,

$W_1 = (w_{1|jh})_{jh} \in \mathbb{R}^{d \times H}$ — м-ца весов I слоя,

$b_1 = (b_{1|1}, \ldots, b_{1|H})^T \in \mathbb{R}^H$ — сдвиг I слоя,

$u = (s_1, \ldots, s_H)^T \in \mathbb{R}^H$— выход I слоя,

$W_2 = (w_{2|hm})_{hm} \in \mathbb{R}^{H \times M}$ — м-ца весов II слоя,

$b_2 = (b_{2|1}, \ldots, b_{2|M})^T \in \mathbb{R}^M$ — сдвиг II слоя,

$y = (y_1, \ldots, y_M)^T \in \mathbb{R}^M$ — выход II слоя.

Тогда работу двухслойной нейронной сети можно представить как:

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

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

Таким образом модель 2-сл. нейронной сети — это **функция** $\boxed{y(x) = \sigma_2\left(\sigma_1\left(x^T W_1 + b_1{}^T\right) W_2 + b_2{}^T\right)}$.

Двухслойная нейронная сеть

Назовем функцию $\sigma(z)$ **сигмоидой**, если $\lim\limits_{z\to-\infty} \sigma(z) = 0$ и $\lim\limits_{z\to+\infty} \sigma(z) = 1$.

$\sigma(z) = \frac{1}{1+e^{-z}}$ — **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 $w_{1|j} \in \mathbb{R}^d$, $w_{2|h} \in \mathbb{R}^d$, $b \in \mathbb{R}$,

что двухслойная нейросеть $y(x) = \sum\limits_{h=1}^{H} w_{2|h} \cdot \sigma\left(\sum\limits_{j=1}^{d} x_j w_{1|j} + b\right)$

равномерно приближает $f(x)$ с любой точностью ε:

$|y(x) - f(x)| < \varepsilon$, для всех $x \in [0, 1]^d$

George Cybenko. Approximation by Superpositions of a Sigmoidal functions.
Mathematics of Control, Signals, and Systems. 1989.

Двухслойная нейронная сеть

Выводы

- С помощью линейных операций и функций активаций σ от одного аргумента можно приближать **любую непрерывную функцию** на заданном интервале с любой желаемой точностью.
- **Двух слоев** в нейронной сети **<u>теоретически</u> достаточно**.

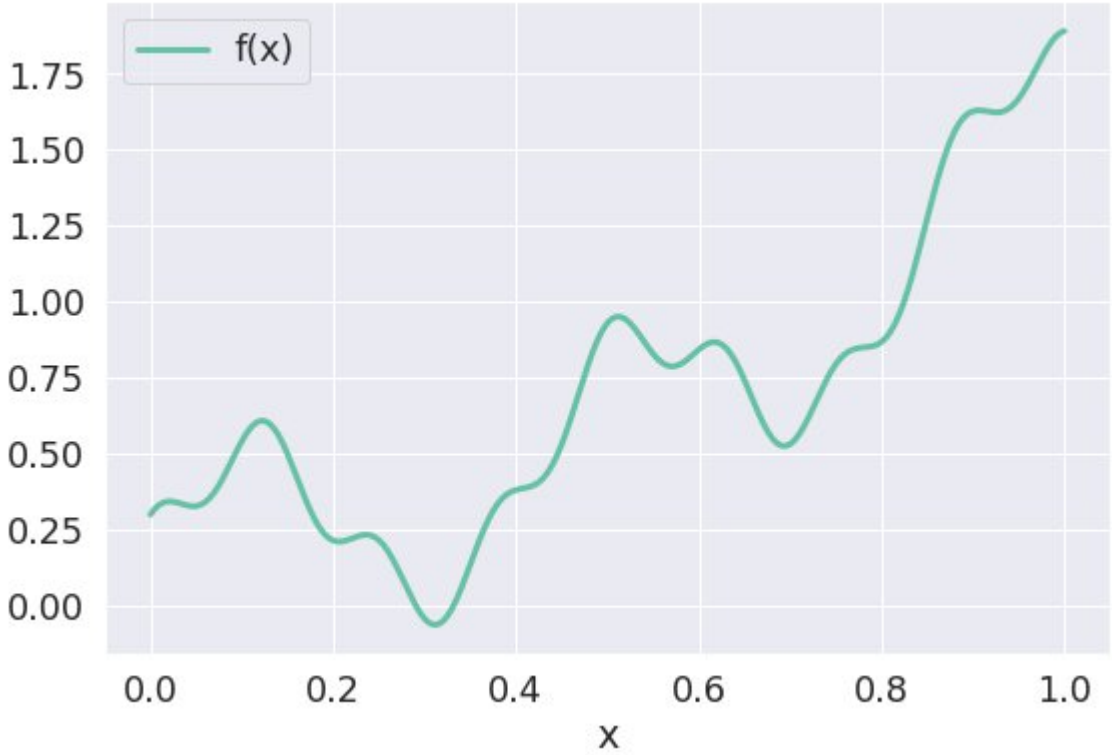
Замечания

- 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.

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?

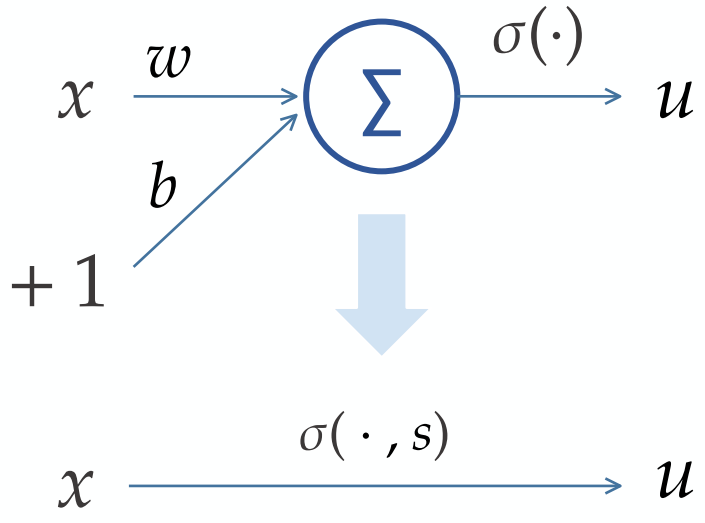
Рассмотрим **один нейрон** $u = \sigma(wx + b)$ с функцией активации $I\{z > 0\}$.

Для удобства

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

Итоговая функция зависит только от s

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



MAE = 0.35

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 $y = wu$.

$$x \xrightarrow{\sigma(\,\cdot\,,\, s_1 = 0.35)} u_1 \xrightarrow{w_1 = 0.8} \Sigma \rightarrow y$$

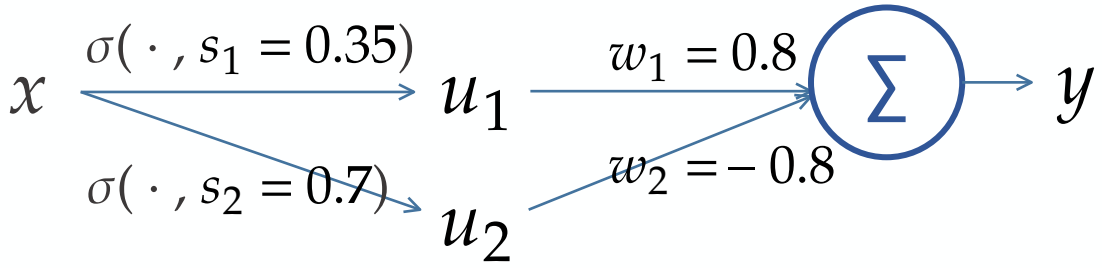
Положим $w = 0.8$.

MAE = 0.32

Пример

Как аппроксимировать $f(x)$ нейронной сетью?

Добавим по нейрону на первом и втором слое.

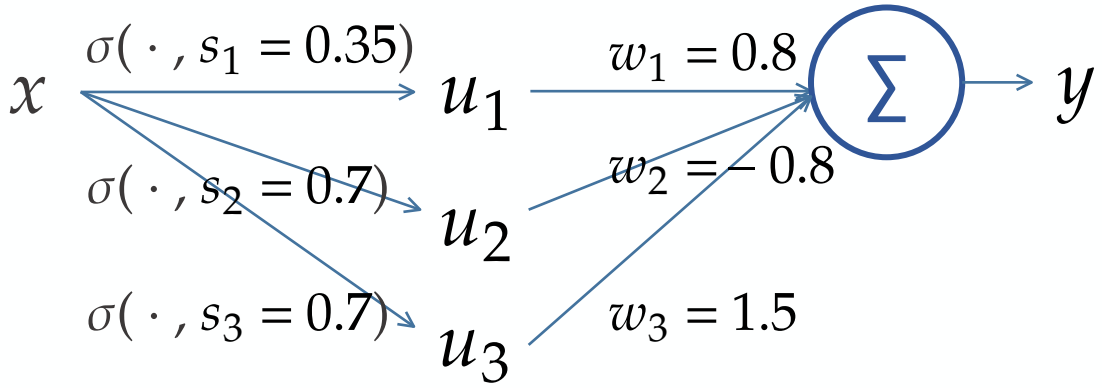


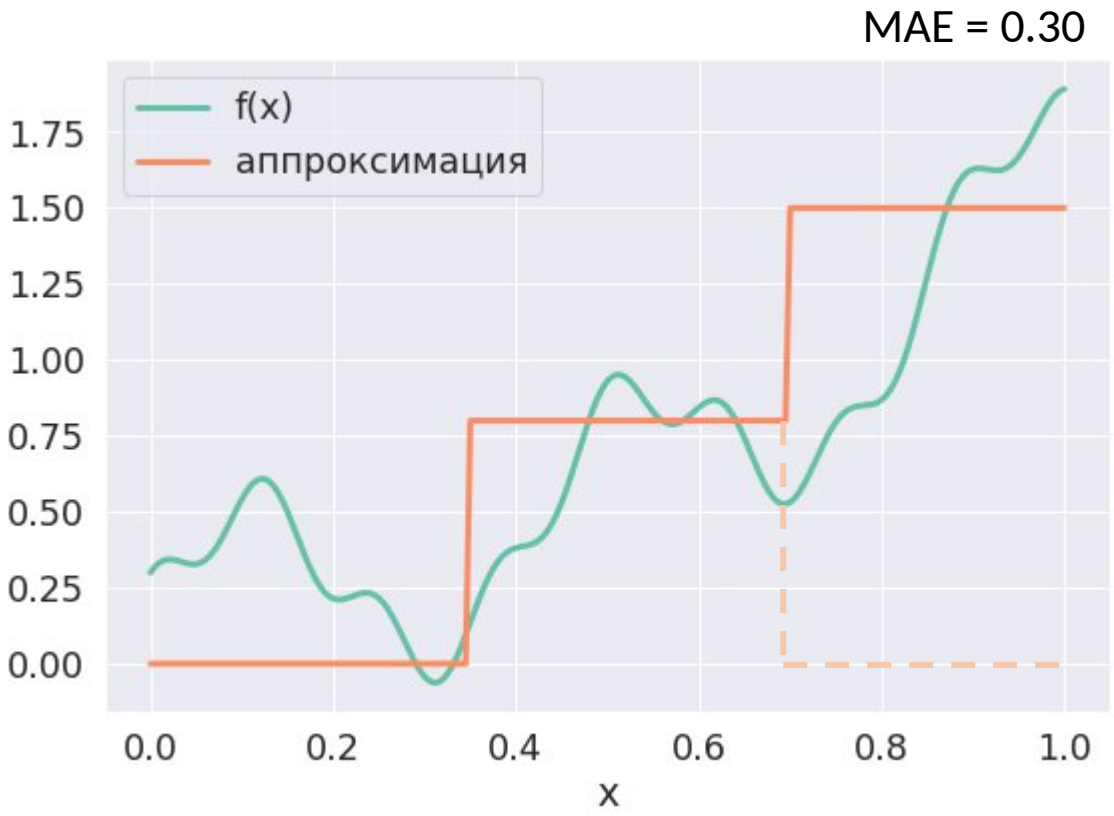
Получилась **ступенька**,
которая приближает часть функции.

Пример

Как аппроксимировать $f(x)$ нейронной сетью?

Добавим еще по нейрону на первом и втором слое.

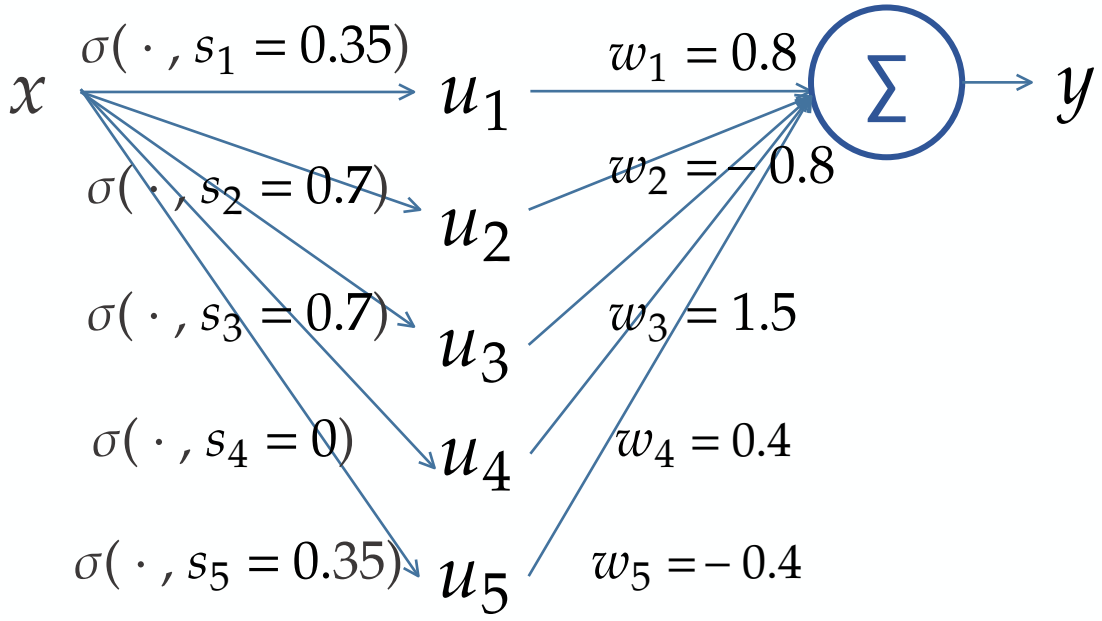




Пример

Как аппроксимировать $f(x)$ нейронной сетью?

Добавим еще по два нейрона на первом и втором слое.



Пример

Вывод

- С помощью 2-сл. нейросети 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.

Введение в нейронные сети

- Определение нейронной сети
- Модель нейронной сети
- **Обучение нейронной сети**
- Применение нейронных сетей

Подбор параметров или обучение нейросети

Как найти параметры нейросети?

Вспомним, что **лин. и лог. регрессии аналогичны нейрону** с тожд. и лог. функциями активации соответственно.

Но у нейросетей не может быть аналитического решения из-за нелинейностей.

У лог. регрессии тоже.
Значит нужно использовать другой подход — **метод градиентного спуска**.

Линейные модели. Напоминание

Пусть $X = (x_{ij})_{ij} \in \mathbb{R}^{n \times d}$ — матрица входных данных, $Y \in \mathbb{R}^n$ — таргет.

Линейная регрессия

- Модель
$$y(x) = x^T w + b,$$
$w = (w_1, \ldots, w_d)^T$ — вектор весов, b — сдвиг.

- Минимизируем квадратичную ошибку:
$$\mathscr{L} = \sum_{i=1}^{n} (\widehat{Y}_i - Y_i)^2 = \left\| \widehat{Y} - Y \right\|^2,$$
где $\widehat{Y} = Xw + b$ — предсказания модели.

Логистическая регрессия

- Модель
$$y(x) = \sigma\left(x^T w + b\right),$$
$w = (w_1, \ldots, w_d)^T$ — вектор весов, b — сдвиг,
$\sigma(z) = \frac{1}{1+e^{-z}}$ — логистическая сигмоида.

- Минимизируем кросс-энтропию:
$$\mathscr{L} = \sum_{i=1}^{n} \left(Y_i \log \widehat{Y}_i + (1 - Y_i) \log\left(1 - \widehat{Y}_i\right)\right),$$
где $\widehat{Y} = \sigma(Xw + b)$ — предсказания модели.

Обозначим все параметры сети как $\theta = (w_1, \ldots, w_d, b)^T$, положим $\tilde{X} = \begin{pmatrix} x_{11} & \cdots & x_{1d} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \cdots & x_{nd} & 1 \end{pmatrix}$.

Находим параметры с помощью градиентного спуска: $\theta_{t+1} = \theta_t - \eta \nabla \mathscr{L}(\theta_t)$, где η — скорость обучения

где $\nabla \mathscr{L}(\theta) = 2\tilde{X}^T(\tilde{X}\theta - y)$

где $\nabla \mathscr{L}(\theta) = \tilde{X}^T(S(\theta) - Y) = \sum_{i=1}^{n} \left(y_i - \sigma\left(\tilde{x}_i \theta^T\right)\right)\tilde{x}_i,$

$S(\theta) = \sigma\left(\tilde{X}\theta\right) = \left(\sigma\left(\tilde{x}_1^T \theta\right), \ldots, \sigma\left(\tilde{x}_n^T \theta\right)\right)^T$

Обучение нейронной сети. Пример

Пусть $X = (x_{ij})_{ij} \in \mathbb{R}^{n \times d}$ — матрица входных данных, $Y \in \mathbb{R}^n$ — таргет.

- Модель — двухслойная нейронная сеть: $y(x) = \sigma_2\left(\sigma_1\left(x^T W_1 + b_1{}^T\right) W_2 + b_2{}^T\right)$,

  $W_1 \in \mathbb{R}^{d \times H}$ и $b_1 \in \mathbb{R}^H$ — м-ца весов и сдвиг I слоя,

  $W_2 \in \mathbb{R}^{H \times M}$ и $b_2 \in \mathbb{R}^M$ — м-ца весов и сдвиг II слоя,

  σ_1 и σ_2 — функции активации.

- Задаем ф-ю $\mathscr{L}$, которую будем минимизировать, например, MSE.

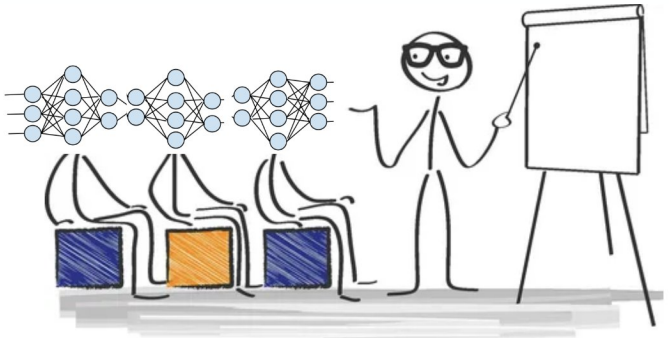
Обозначим все параметры сети как θ.

Находим параметры с помощью градиентного спуска:

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

Процесс поиска оптимальных параметров нейросети называется **обучением**.

Цель — нужно найти $\nabla \mathscr{L}$ по каждому параметру модели.



Задача

Пусть $y(x) = \tanh(\left(\mathrm{ReLU}_1\left(x^T W_1 + b_1\right) W_2 + b_2\right)$ — двухслойная нейронная сеть,

где $W_1 \in \mathbb{R}^{d \times H}$, $b_1 \in \mathbb{R}^H$, $W_2 \in \mathbb{R}^{H \times M}$, $b_2 \in \mathbb{R}$.

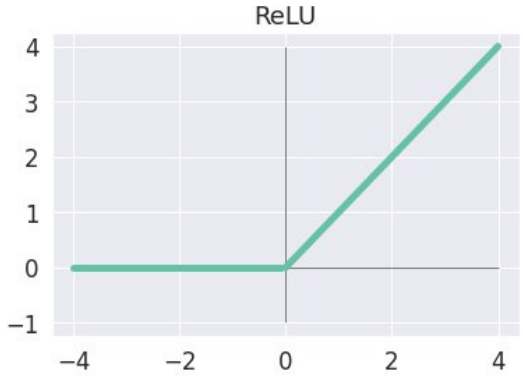
Положим $X = (x_{ij})_{ij} \in \mathbb{R}^{n \times d}$ — матрица входных данных, $Y \in \mathbb{R}^{n \times M}$ — отклик.

Сеть обучается путем минимизации квадратичной ошибки.

Посчитайте правила обновления параметров.

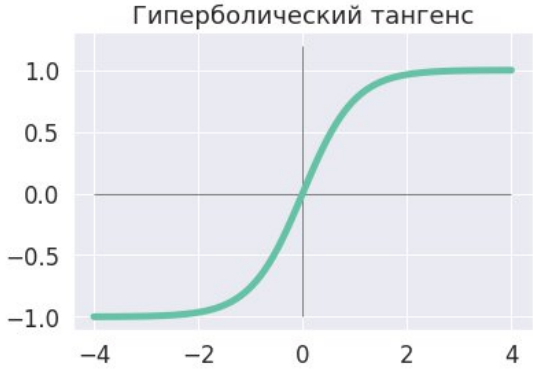
$$\mathbf{ReLU}(t) = tI\{t \geq 0\}$$

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



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

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



Решение на доске

Обучение нейронной сети

Задача

Пусть у нас имеется нейронная сеть из $\boldsymbol{K}$ **слоев** $y = f_K(f_{K-1}(\ldots(f_1(x))\ldots))$.
Для удобства обозначим x_k, y_k, θ_k как вход, выход и параметризация слоя k,
т.е. $y_k = f_k(x_k, \theta_k)$.

Хотим найти $\frac{\partial \mathscr{L}}{\partial \theta_k}$ для всех $k \in \{1, \ldots, K\}$.

Наивный подход

Посчитать для каждого параметра градиент отдельно.

Но! Такой подсчет одного градиента линеен по количеству параметров.

$\Rightarrow$ **Сложность** такой процедуры **квадратична** по количеству параметров.

Метод обратного распространения ошибки

Градиенты будем вычислять **последовательно**,

используя **формулу производной сложной функции**.

Прямой проход

$$x_1 \to \dots \to \boxed{\begin{array}{c} y_k = f_k(x_k, \theta_k) \\ x_k = y_{k-1} \end{array}} \to \boxed{\begin{array}{c} y_{k+1} = f_{k+1}(x_{k+1}, \theta_{k+1}) \\ x_{k+1} = y_k \end{array}} \to \dots \to y_K$$

Обратное распространение ошибки

$$\frac{\partial \mathscr{L}}{\partial x_k} = \frac{\partial \mathscr{L}}{\partial y_k} \cdot \frac{\partial y_k}{\partial x_k} \leftarrow \frac{\partial \mathscr{L}}{\partial y_k} = \frac{\partial \mathscr{L}}{\partial x_{k+1}} \leftarrow \frac{\partial \mathscr{L}}{\partial x_{k+1}} = \frac{\partial \mathscr{L}}{\partial y_{k+1}} \cdot \frac{\partial y_{k+1}}{\partial x_{k+1}} \leftarrow \frac{\partial \mathscr{L}}{\partial y_{k+1}} = \frac{\partial \mathscr{L}}{\partial x_{k+2}} \leftarrow$$

$$\frac{\partial \mathscr{L}}{\partial \theta_k} = \frac{\partial \mathscr{L}}{\partial y_k} \cdot \frac{\partial y_k}{\partial \theta_k} \qquad \frac{\partial \mathscr{L}}{\partial \theta_{k+1}} = \frac{\partial \mathscr{L}}{\partial y_{k+1}} \cdot \frac{\partial y_{k+1}}{\partial \theta_{k+1}}$$

вычисляется за 1 шаг

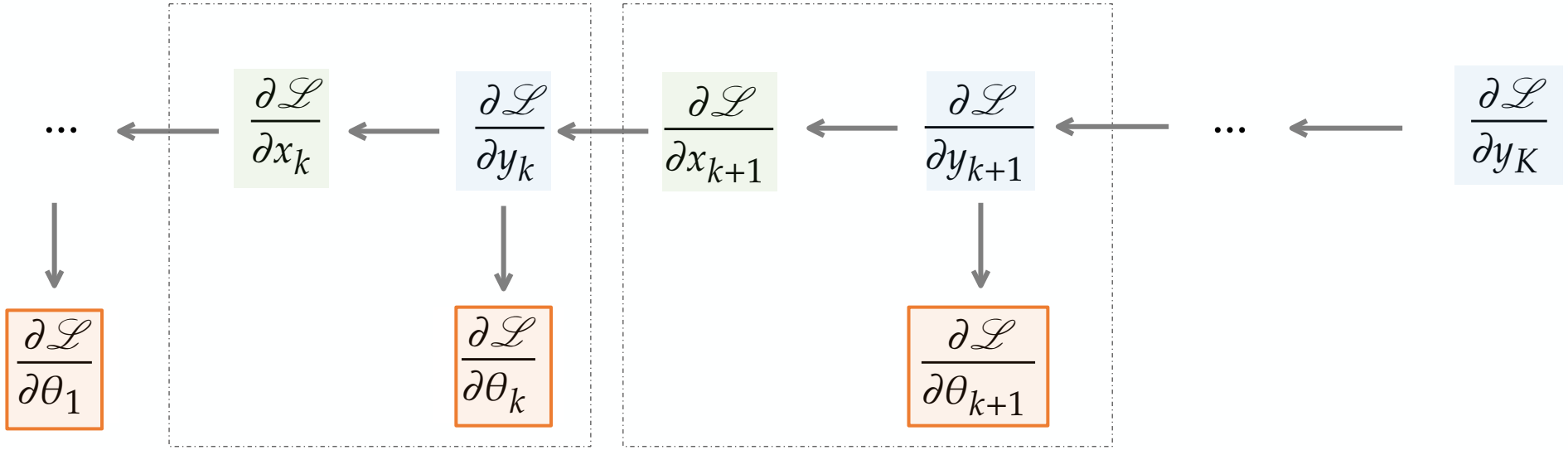
Метод обратного распространения ошибки

Обновляем параметры сети слой за слоем, начиная с последнего.

Для вычисления производных слоя k нам требуется производная $\mathscr{L}$ по входу слоя $k+1$.

Общая сложность получается **линейной!**

Обратное распространение ошибки — back propagation

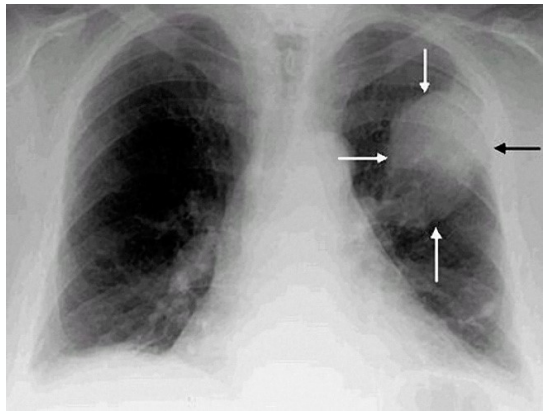


Введение в нейронные сети

- Определение нейронной сети
- Модель нейронной сети
- Обучение нейронной сети
- **Применение нейронных сетей**

Применение нейронных сетей. Примеры

Здравоохранение:
определение
рака легкого
по рентгенографии
грудной клетки



Образование:
адаптация
языковых уроков
под студентов

duolingo

Транспорт:
автономные
автомобили

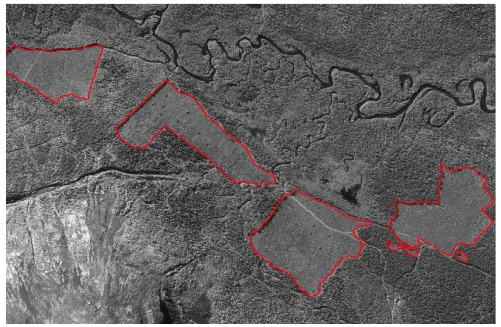
Творчество:
генерация
изображений

Применение нейронных сетей. Примеры

Умные устройства:
голосовые помощники

Игры и развлечения:
боты для игр

Экология:
анализ спутниковых снимков
для отслеживания вырубки лесов



Бизнес и маркетинг:
персональные рекомендации

Применение нейронных сетей. Примеры

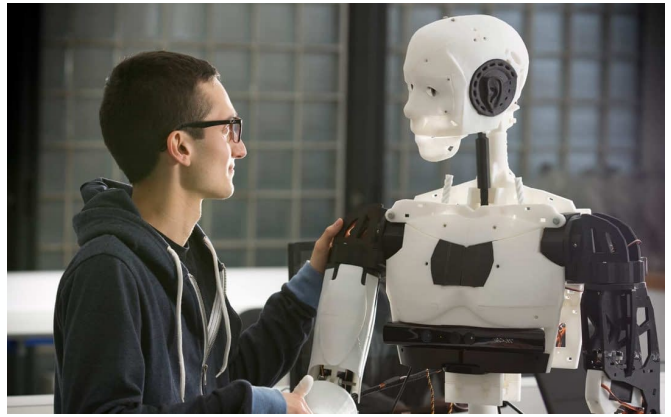
Чат-боты

YandexGPT

deepseek

Финансы:
прогнозирование рынков

Робототехника:
управление роботами



И многое-многое другое ...

ВСЁ!