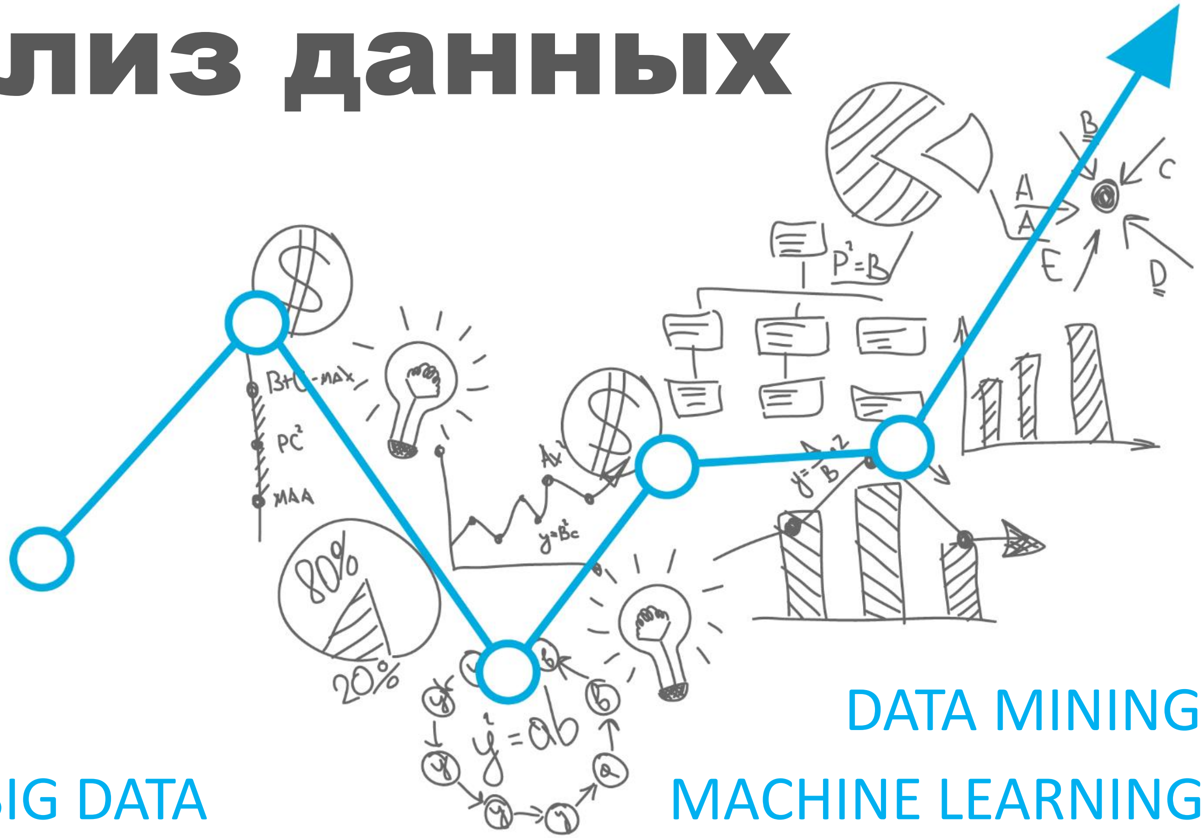
Интеллектуальный анализ данных



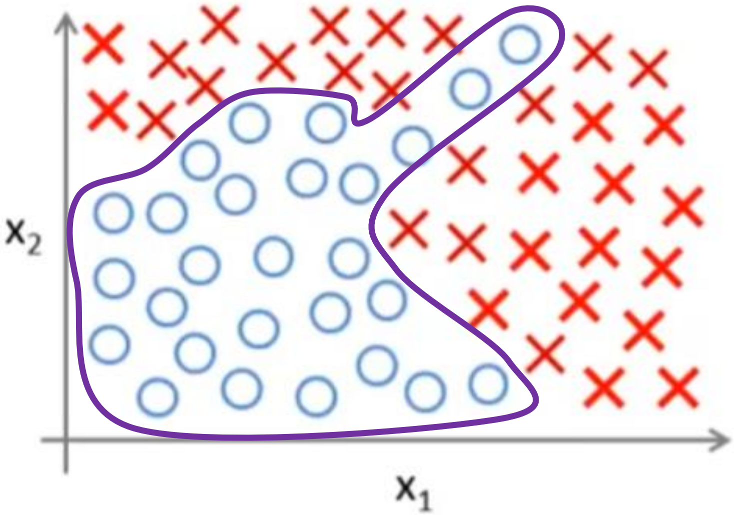
DATA MINING

BIG DATA

MACHINE LEARNING

Лекция 5. Нейронные сети

Сложная нелинейная классификация



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \\ +\theta_2 {x_1}^2 + \theta_3 x_1 x_2 + \cdots + \\ +\theta_8 {x_2}^2 + \theta_{15} x_1 {x_2}^4 + \cdots)$$

$$g(z) = \frac{1}{1+e^{-z}}$$ – сигмоида

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$n = 2$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \ldots \\ x_{100} \end{bmatrix}$$

– размер
– количество комнат
– возраст
– размер
…

$n = 100$

Квадратичные функции

$$\|\theta\| \approx O(n^2) \approx \frac{n^2}{2}$$

≈5 000 параметров

Кубические функции

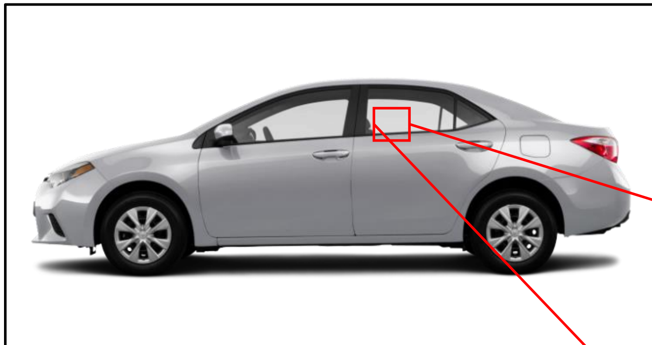
$$\|\theta\| \approx O(n^3)$$

≈170 000 параметров

Сложная нелинейная классификация

Задача компьютерного зрения

Что это?



Автомобиль

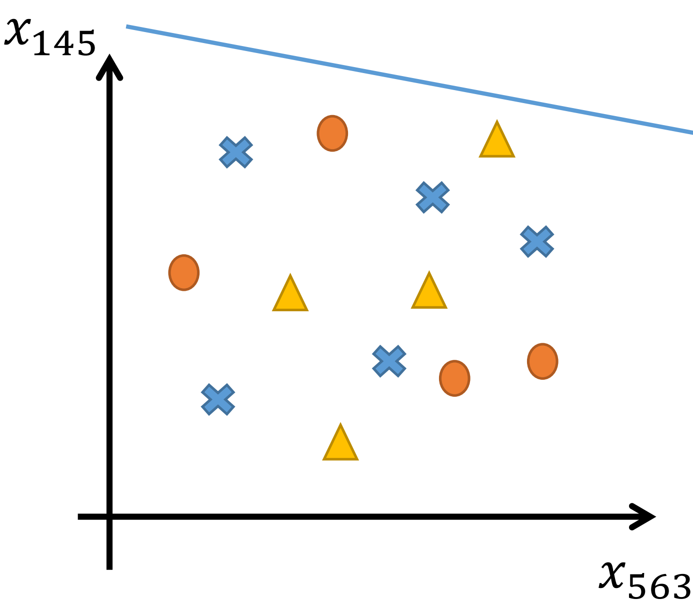
Велосипед

Жираф

Река

Человек

Котик



03	00	00	00	00	00	00	00	31	00	00	00	00	00	00	00
F6	00	00	00	01	00	00	00	24	2B	A5	32	3E	28	82	15
00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	B8	C0	07
8E	D8	E8	16	00	B8	00	0D	8E	C0	33	DB	C6	06	0E	00
10	E8	53	00	68	00	0D	68	6A	02	CB	8A	16	24	00	B4
08	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66
0F	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F
B7	C9	66	F7	E1	66	A3	20	00	C3	B4	41	BB	AA	55	8A
16	24	00	CD	13	72	0F	81	FB	55	AA	75	09	F6	C1	01
74	04	FE	06	14	00	C3	66	60	1E	06	66	A1	10	00	66

Сложная нелинейная классификация

Количество параметров модели?

Изображение 50×50 → 2500 пикселей

Полутоновое: $n = 2500$

Цветное (RGB): $n = 7500$

Линейная модель	Квадратичная модель	Кубическая модель
$n = 2500$	$n = 2500$	$n = 2500$
$\|\theta\| \approx O(n)$	$\|\theta\| \approx O(n^2)$	$\|\theta\| \approx O(n^3)$
2501	Более 6 млн	Более 15 млн

Изображение 500×500 — 250 000 пикселей — $\|\theta\| \approx O(n^3) \approx$15 625 000 000 000 000

Слишком много для логистической регрессии!

Сложная нелинейная классификация

Задача слишком сложна

Варианты?

Взять маленькое изображение
(5×5, 10×10, 15×20, …)

Низкое качество,
слабо применимо в
реальности

Выбор параметров,
усложнение функции
(FFT, PCA, VJD, …)

Индивидуальный
подход, не обобщается

Не использовать сложную модель, взять
множество простых со сложной структурой
связей между ними

Нейронная сеть, логистическое дерево, бустинг, …

Искусственные нейронные сети

1940-е — Первые идеи об искусственных нейронных сетях, модель перцептрона

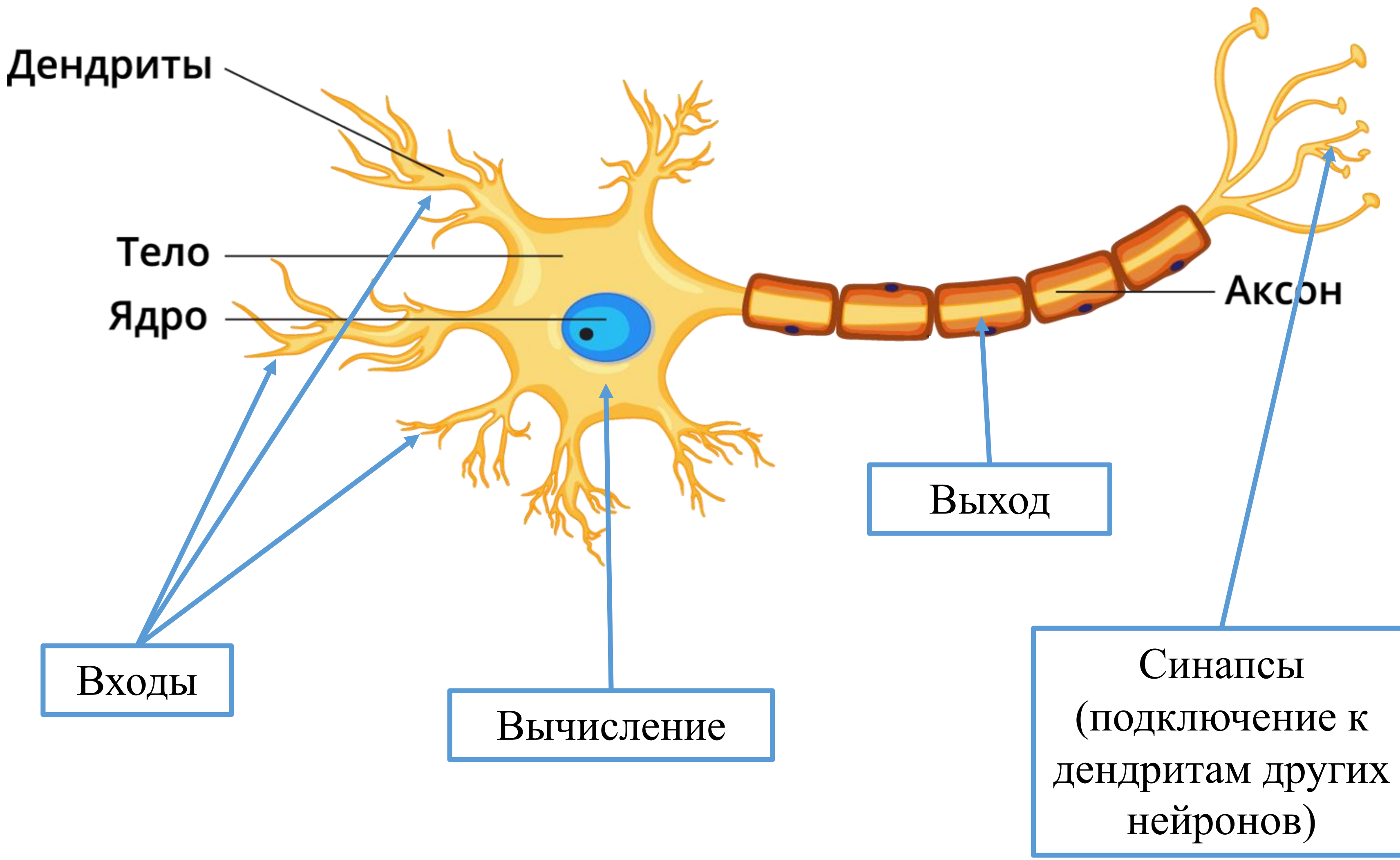
1980-е — Развитие теории нейросетей, различные функции активации, применения

1990-е конец — Алгоритм обучения (метод обратного распространения ошибки)

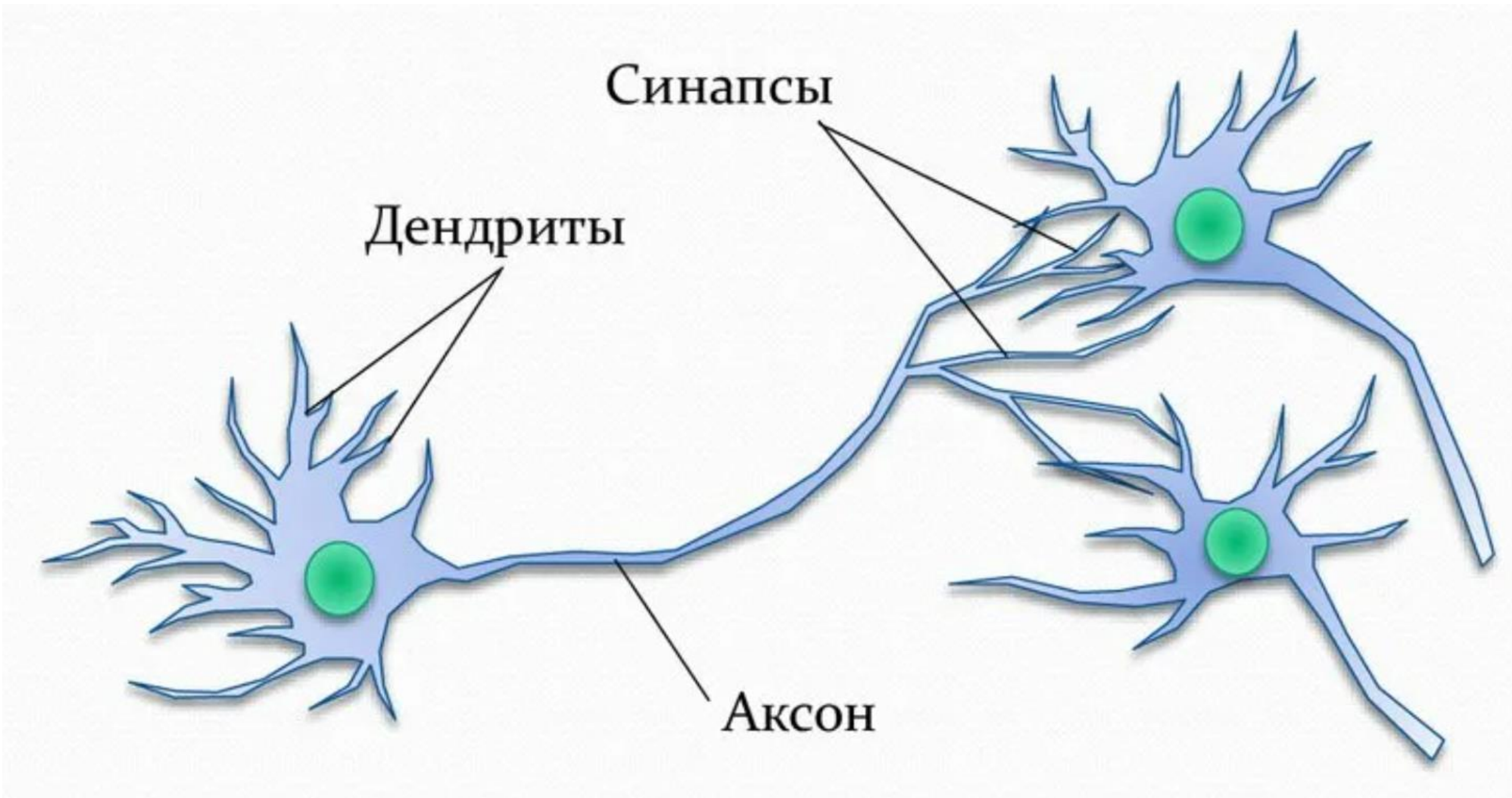
2000-е — Бурное развитие нейросетей, появление глубокого обучения, сверточных сетей, рекуррентных сетей и многое другое

2020-е — Появление первых приближений искусственного интеллекта (эффективные порождающие сети)

Нейрон


Дендриты
Тело
Ядро
Аксон
Входы
Вычисление
Выход
Синапсы (подключение к дендритам других нейронов)

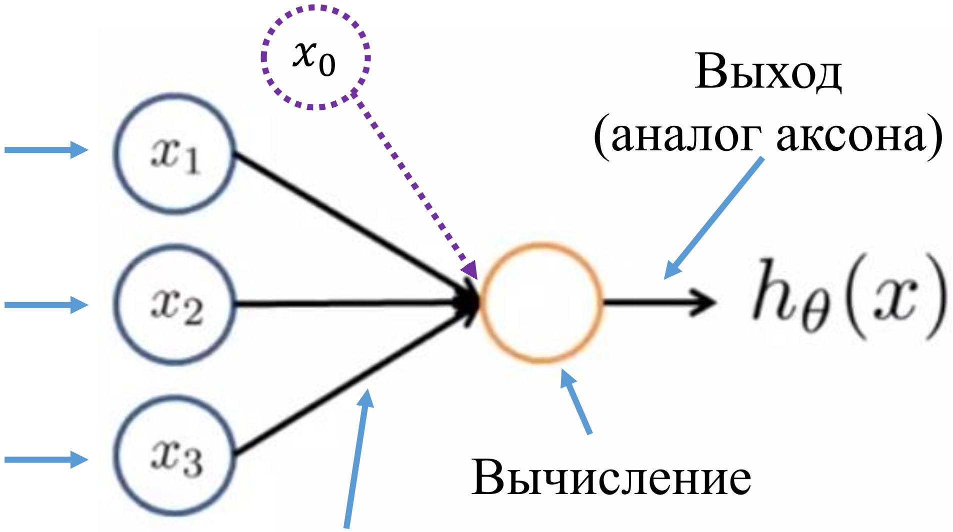
Нейронная сеть



Искусственный нейрон

Модель нейрона: логистический элемент

Входы



Аналог
дендритов

Сигмоидная
(логистическая)
функция
активации

$$g(z) = \frac{1}{1 + e^{-z}}$$

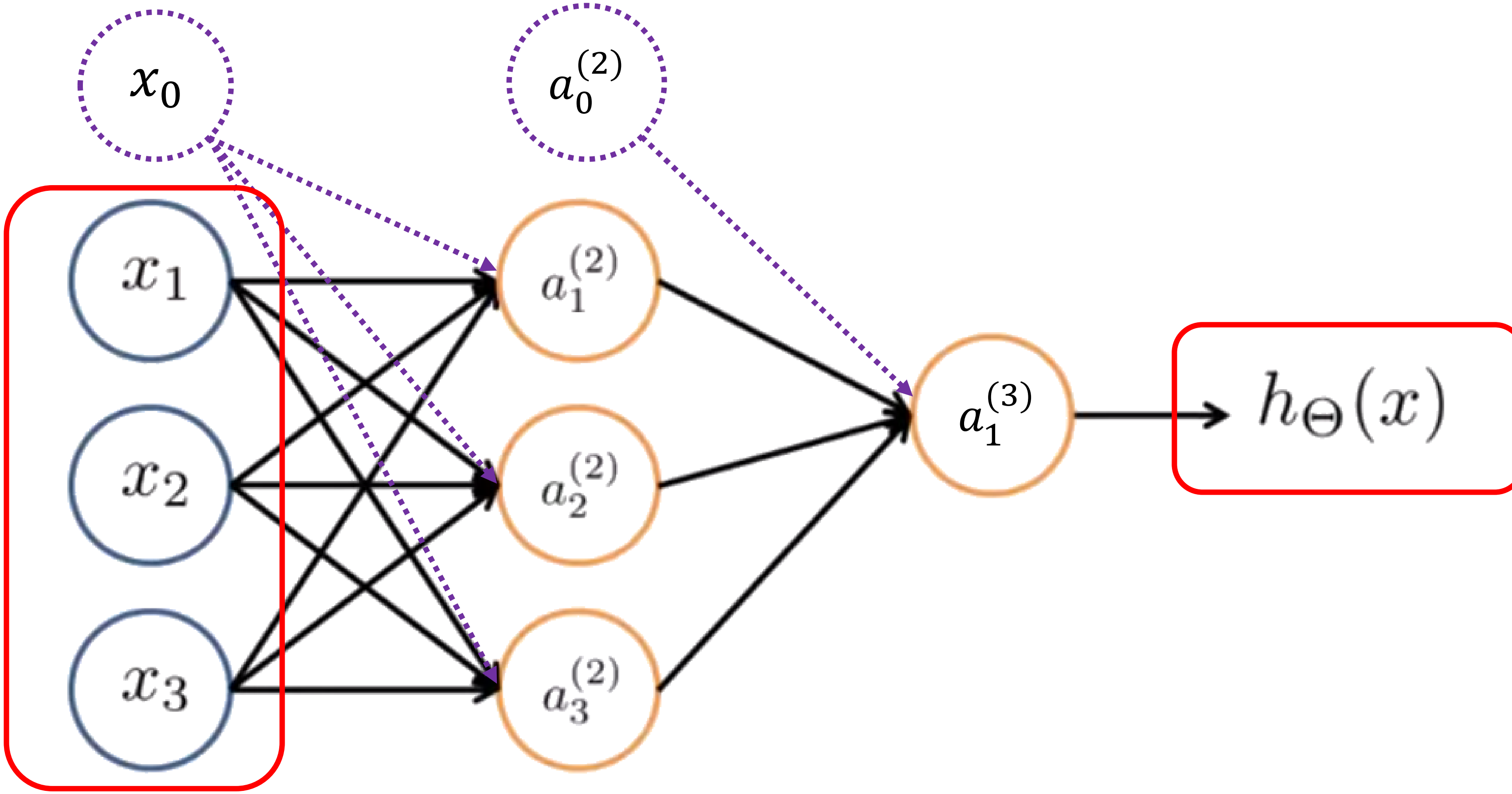
Параметры, веса

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$x \in \mathbb{R}^{n+1} = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \qquad \theta \in \mathbb{R}^{n+1} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix}$$

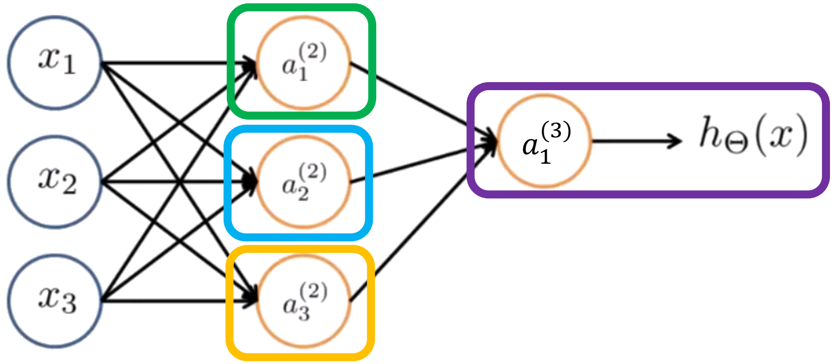
$$x_0 = 1$$

Искусственная нейронная сеть



Слой 1	Слой 2	Слой 3
Входной	Скрытый (м.б. несколько)	Выходной

Искусственная нейронная сеть



$a_i^{(j)}$ – активация (выход) нейрона i в слое j

$\Theta^{(j)}$ – матрица весов между слоем j и слоем j+1

$$a_1^{(2)} = g(\Theta_{10}^{(1)} + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

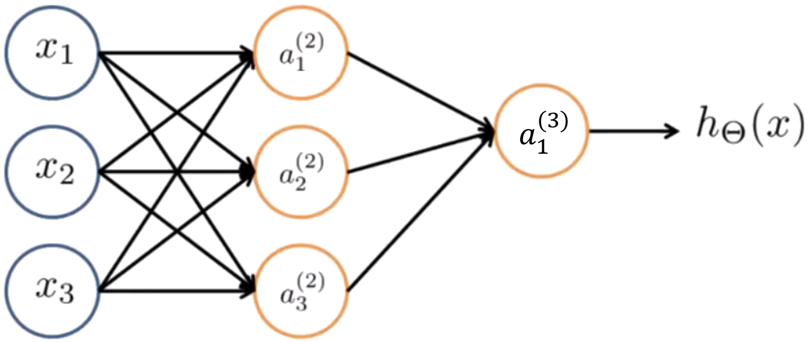
$$\Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} & \Theta_{13}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} & \Theta_{23}^{(1)} \\ \Theta_{30}^{(1)} & \Theta_{31}^{(1)} & \Theta_{32}^{(1)} & \Theta_{33}^{(1)} \end{bmatrix}$$

$$h_\Theta(x) = g(\Theta_{10}^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

$$\Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} & \Theta_{13}^{(2)} \end{bmatrix}$$

Один нейрон искусственной сети с сигмоидной функцией активации эквивалентен линейной логистической регрессии!

Искусственная нейронная сеть



$$a_0^{(1)} = x_0 = 1$$
$$a_1^{(1)} = x_1$$
$$a_2^{(1)} = x_2$$
$$a_3^{(1)} = x_3$$

$$h_\Theta(x) = a_1^{(3)}$$

s_j – количество нейронов в слое j $(s_2 = 3, s_3 = 1)$

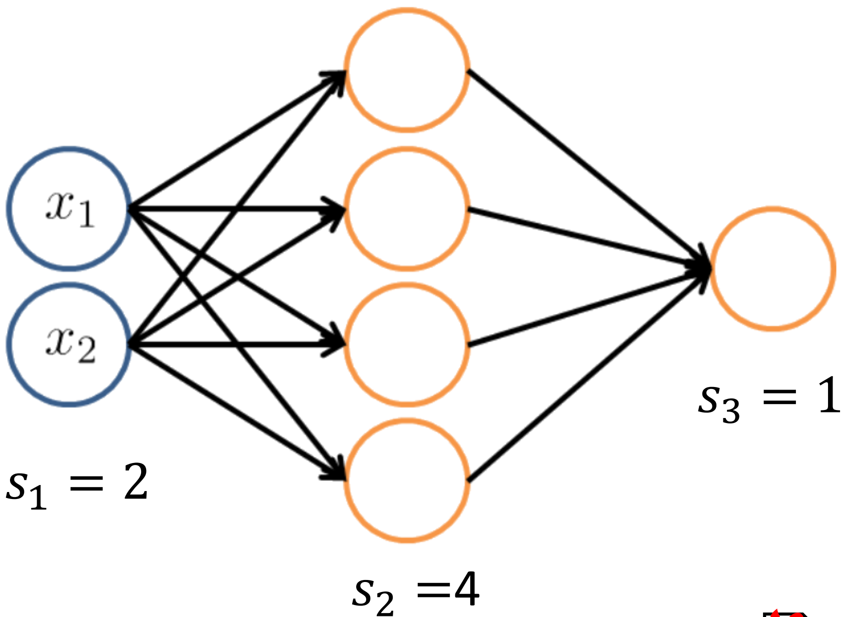
$$a_i^{(j+1)} = g(\Theta_{i0}^{(j)} a_0^{(j)} + \Theta_{i1}^{(j)} a_1^{(j)} + \cdots + \Theta_{is_{j+1}}^{(j)} a_{s_{j+1}}^{(j)})$$

$\Theta^{(j)} = \mathbb{R}^{s_{j+1} \times (s_j+1)}$ $\quad a^j = \mathbb{R}^{s_j+1}$ $\quad a_0^j = 1$ $\quad g(z) = \frac{1}{1+e^{-z}}$

$$\boxed{a^{(j+1)} = g(\Theta^{(j)} a^{(j)})}$$

Можно вычислить активацию всех нейронов слоя за одну матричную операцию умножения, не важно сколько в нем нейронов

Искусственная нейронная сеть

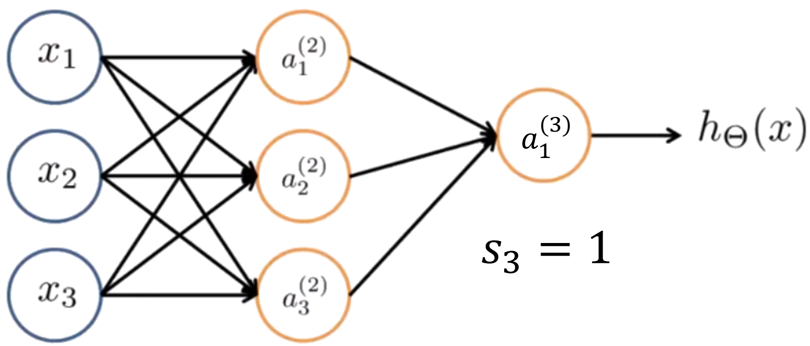


Какие размеры имеет матрица $\Theta^{(1)}$?

$$\Theta^{(j)} = \mathbb{R}^{s_{j+1} \times (s_j + 1)}$$

- ☒ 2×4
- ☒ 4×2
- ☒ 3×4
- ☑ 4×3

Алгоритм прямого распространения



$s_1 = 3$ $s_2 = 3$

$N = 3$ – количество слоев

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$\Theta^{(j)} = \mathbb{R}^{s_{j+1} \times (s_j + 1)}$$

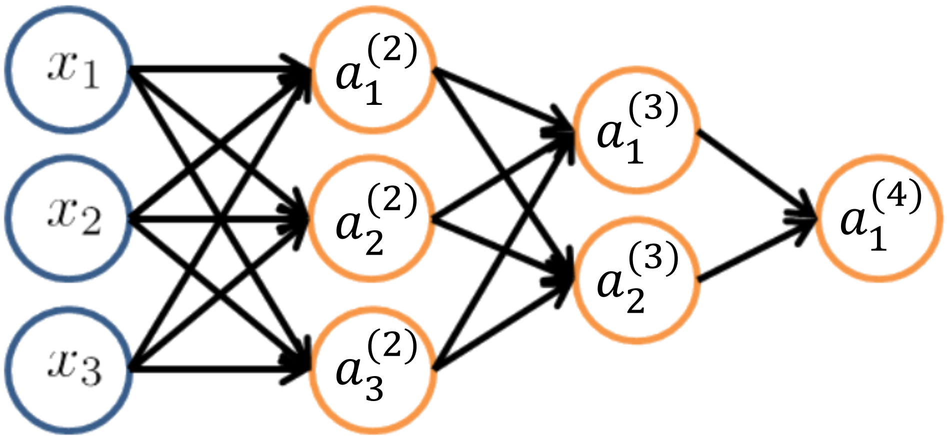
$a^{(1)} = [1 \quad x_1 \quad x_2 \quad \ldots \quad x_{s_1}]^T$
для $j = 1 \ldots (N-1)$ выполнить:
$\quad z^{(j+1)} = \Theta^{(j)} a^{(j)}$
$\quad a^{(j+1)} = \boxed{[1 \mid g(z^{(j+1)})]^T}$
конец
$h_\Theta(x) = a^{(N)}$

Конкатенация
(добавляем слева 1)

$$[1 \quad g_1 \quad g_2 \quad \ldots \quad g_k]$$

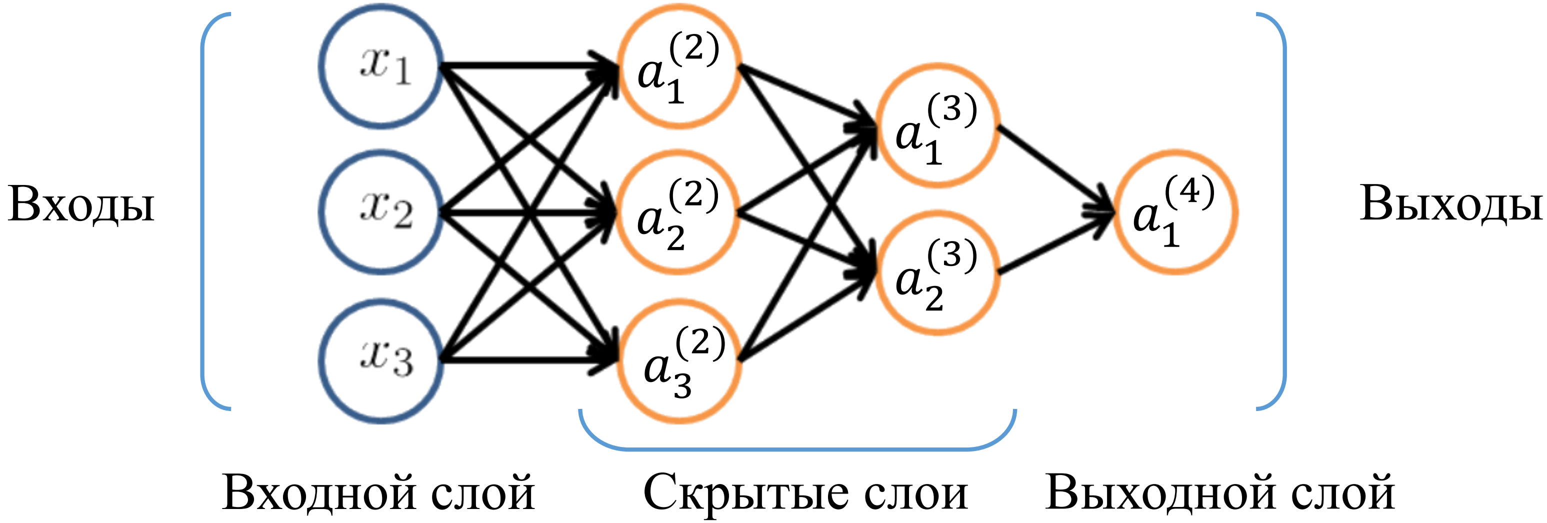
$$[1] \mid [g_1 \quad g_2 \quad \ldots \quad g_k]$$

Алгоритм прямого распространения



Как вычислить $a^{(2)}$?

- ☒ $a^{(2)} = \Theta^{(1)} a^{(1)}$
- ☒ $z^{(2)} = \Theta^{(2)} a^{(1)}$; $a^{(2)} = \left[1 \mid g(z^{(2)})\right]$
- ☑ $z^{(2)} = \Theta^{(1)} a^{(1)}$; $a^{(2)} = \left[1 \mid g(z^{(2)})\right]$
- ☒ $z^{(2)} = \Theta^{(1)} g(a^{(1)})$; $a^{(2)} = g(z^{(2)})$

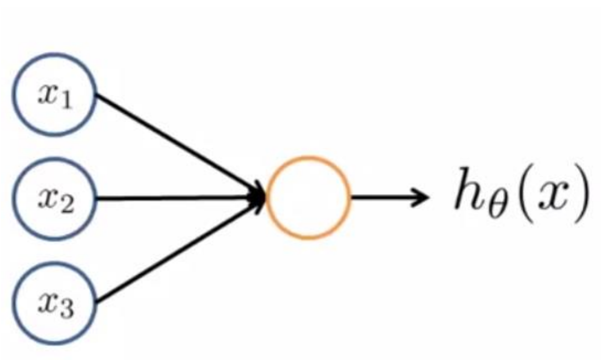


Архитектура определяет:

- Количество нейронов входного слоя
- Количество нейронов входного слоя
- Количество слоев
- Типы слоев — Плотный, Сверточный, …
- Типы связей в слоях — Прямые, Обратные
- Типы функций активации — Сигмоидная, Гиперболическая, Линейная, …

Сравнение нейросети и логистической регрессии

Логистическая регрессия



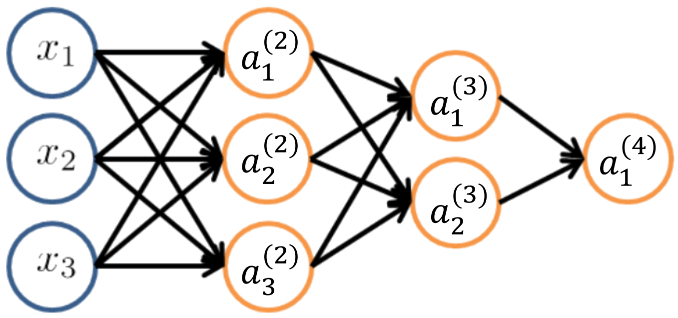
Повышаем сложность модели, увеличиваем число параметров

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \\ +\theta_2 {x_1}^2 + \theta_3 x_1 x_2 + \cdots + \\ +\theta_8 {x_2}^2 + \theta_{15} x_1 {x_2}^4 + \cdots)$$

Очень быстро считается

Очень долго обучается

Нейронная сеть



Увеличиваем количество слоев и нейронов, входной вектор не изменяется

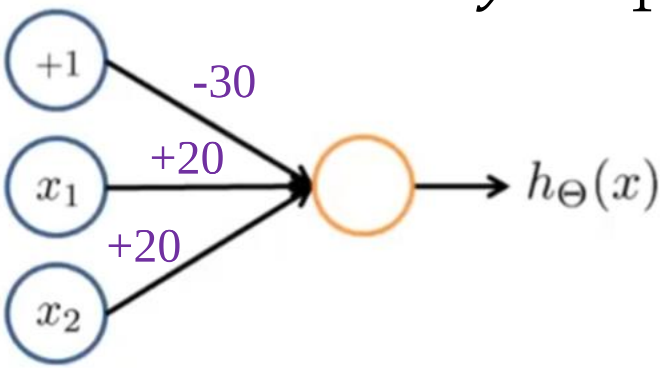
Быстро считается (алгоритм прямого распространения)

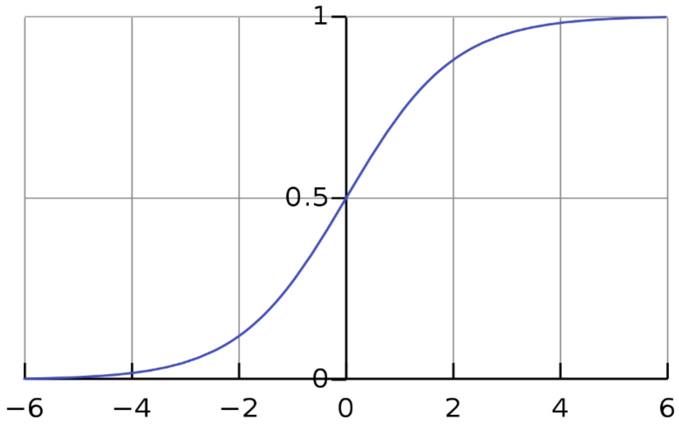
Относительно быстро обучается (алгоритм обратного распространения ошибки)

Логическая (булева) функция И (AND, ∧ , &)

$$x_1, x_2 \in \{0, 1\}$$

$$y = x_1 \wedge x_2$$





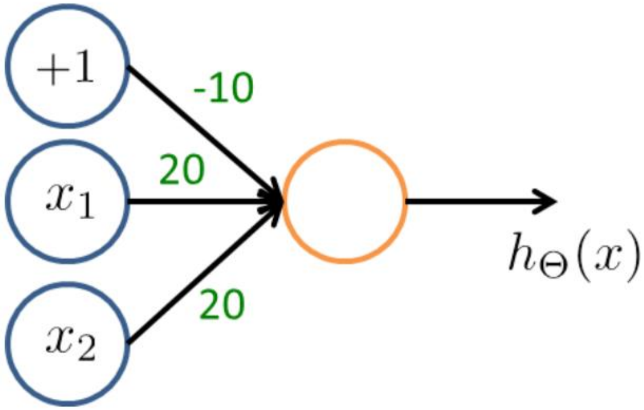
$$h_\Theta(x) = g(-30 + 20x_1 + 20x_2)$$

$\Theta^{(1)}_{10}$ $\Theta^{(1)}_{11}$ $\Theta^{(1)}_{12}$

x_1	x_2	$h_\Theta(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(+10) \approx 1$

$$h_\Theta(x) \approx x_1 \wedge x_2$$

Логические функции



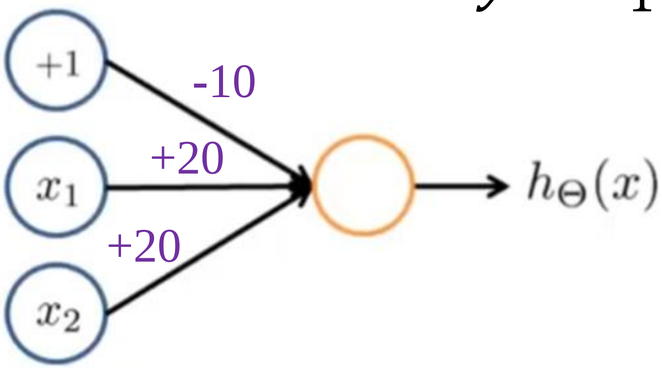
Пусть $x_1, x_2 \in \{0, 1\}$. Какую булеву функцию приближенно вычисляет представленная нейронная сеть?

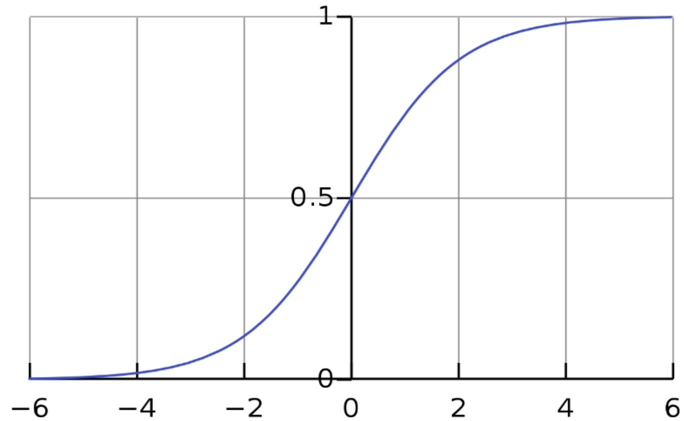
- $x_1\ AND\ x_2$
- $x_1\ OR\ x_2$
- (NOT x_1) OR (NOT x_2)
- (NOT x_1) AND (NOT x_2)

Логическая (булева) функция ИЛИ (OR, ∨, |)

$$x_1, x_2 \in \{0, 1\}$$

$$y = x_1 \vee x_2$$





$$h_\Theta(x) = g(\underbrace{-10}_{\Theta_{10}^{(1)}} + \underbrace{20}_{\Theta_{11}^{(1)}}x_1 + \underbrace{20}_{\Theta_{12}^{(1)}}x_2)$$

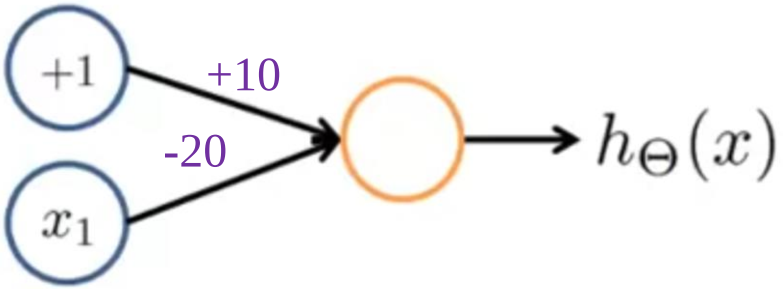
x_1	x_2	$h_\Theta(x)$
0	0	$g(-10) \approx 0$
0	1	$g(+10) \approx 1$
1	0	$g(+10) \approx 1$
1	1	$g(+30) \approx 1$

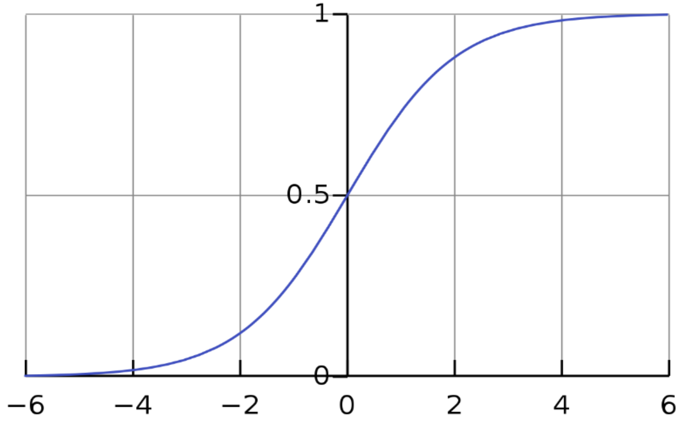
$$h_\Theta(x) \approx x_1 \vee x_2$$

Логическое отрицание НЕ (NOT, $\neg$)

$$x_1 \in \{0, 1\}$$

$$y = \neg x_1$$





$$h_\Theta(x) = g(10 - 20x_1)$$

x_1	$h_\Theta(x)$
0	$g(+10) \approx 1$
1	$g(-10) \approx 0$

$$h_\Theta(x) \approx \neg x_1$$

Логические функции

Пусть $x_1, x_2 \in \{0, 1\}$. Мы хотим приближенно вычислить булеву функцию:

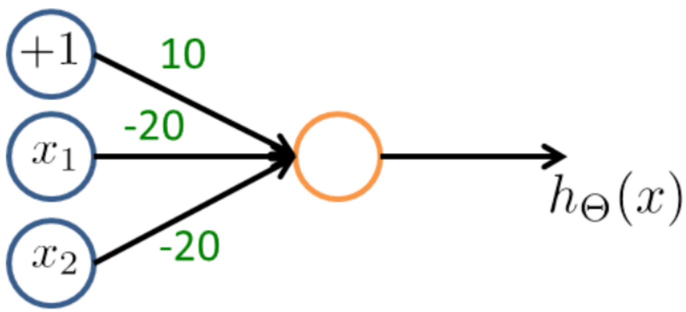
$$y = \neg x_1 \wedge \neg x_2$$

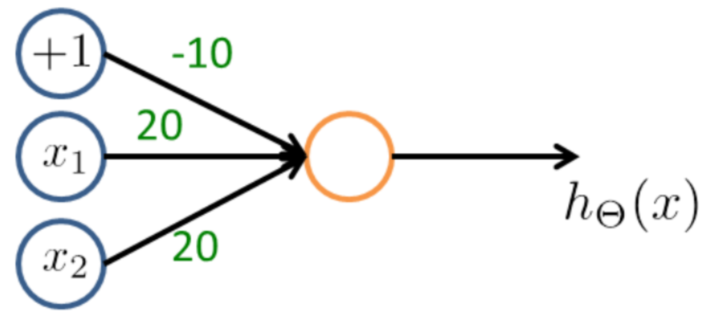
Или в другой записи:

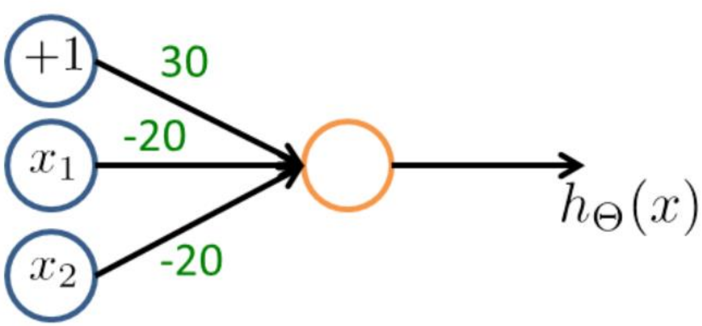
$$y = (NOT\ x_1)\ AND\ (NOT\ x_2)$$

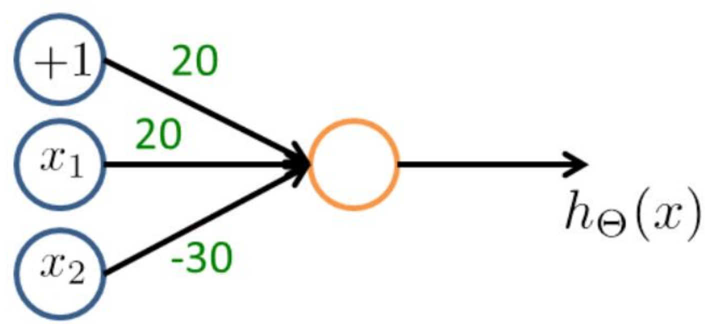
Какая из представленных нейронных сетей решает эту задачу?

x_1	x_2	$\neg x_1$ & $\neg x_2$
0	0	1
0	1	0
1	0	0
1	1	0









ИСКЛЮЧАЮЩЕЕ ИЛИ

XOR, $\oplus$

$x_1 \oplus x_2 = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

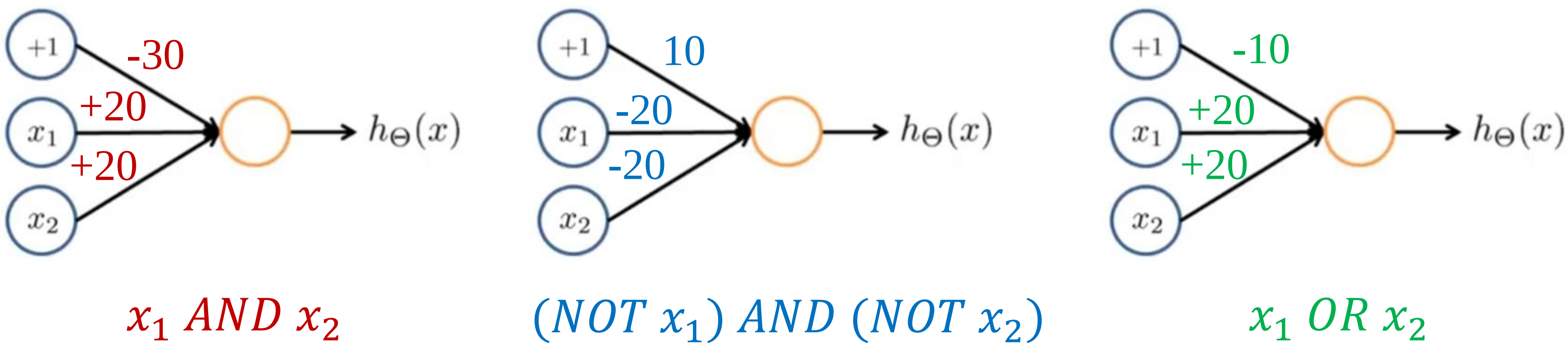
XNOR

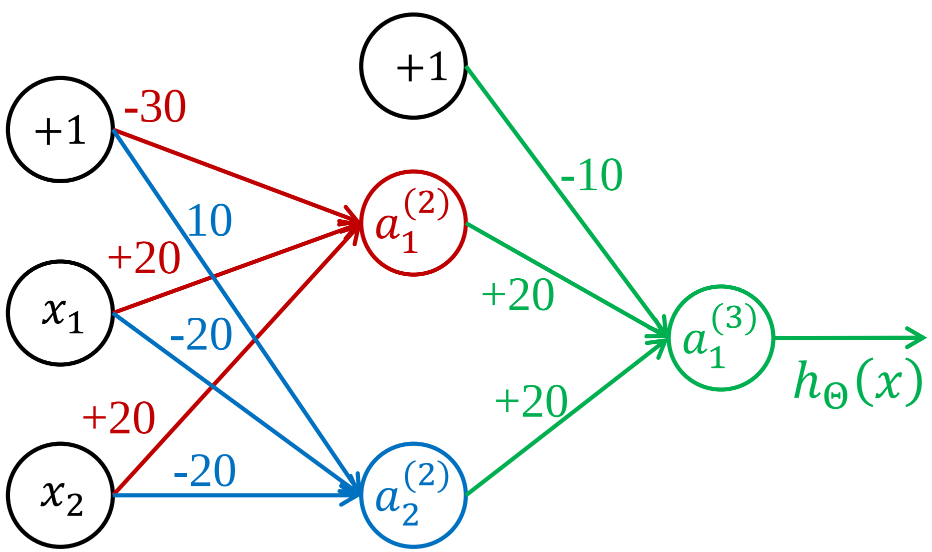
$\neg(x_1 \oplus x_2)$

x_1	x_2	$\neg(x_1 \oplus x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Логические функции

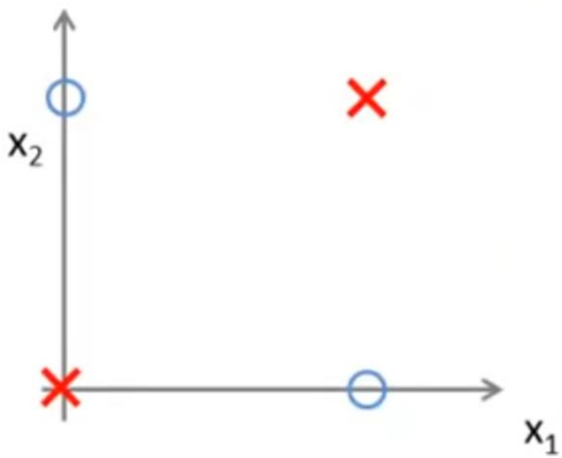
Логическая (булева) функция $x_1\ XNOR\ x_2$

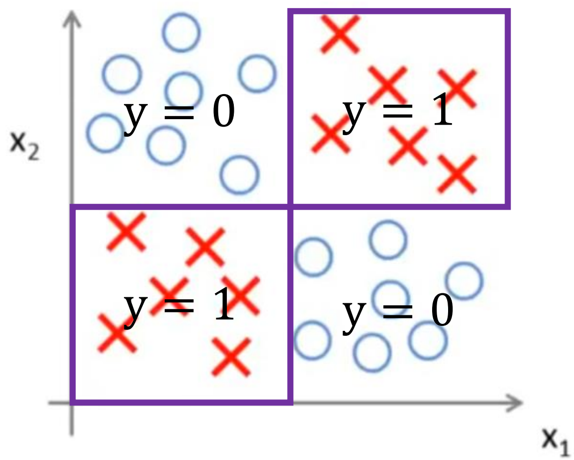




x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Логическая (булева) функция $y = x_1 \ XNOR \ x_2$





x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

Нельзя реализовать логистической регрессией

Легко реализуется нейронной сетью

Многоклассовая классификация

Принцип «**один-против-всех**»

Автомобиль

Пешеход

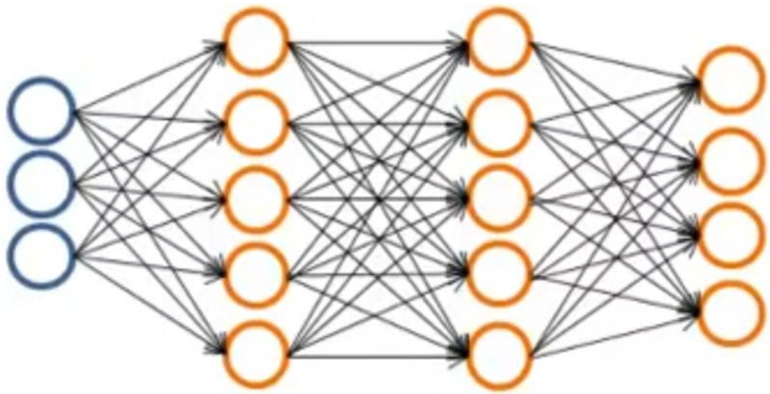
Велосипед

Паровоз

Входной слой

Изображение 400×300

$$x = [x_1, \dots, x_{120\,000}]$$



Выходной слой

$$h_\Theta(x) \in \mathbb{R}^4$$

Автомобиль

$$h_\Theta(x) = \begin{bmatrix} \boxed{1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Пешеход

$$h_\Theta(x) = \begin{bmatrix} 0 \\ \boxed{1} \\ 0 \\ 0 \end{bmatrix}$$

Велосипед

$$h_\Theta(x) = \begin{bmatrix} 0 \\ 0 \\ \boxed{1} \\ 0 \end{bmatrix}$$

Паровоз

$$h_\Theta(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \boxed{1} \end{bmatrix}$$

Многоклассовая классификация

Принцип «**один-против-всех**»

$$h_\Theta(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

В реальности нейросеть выдает:

$$h_\Theta(x) = \begin{bmatrix} 0.94 \\ 0.03 \\ 0.09 \\ 0.20 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0.04 \\ 0.89 \\ 0.12 \\ 0.07 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0.18 \\ 0.01 \\ 0.67 \\ 0.43 \end{bmatrix} \qquad h_\Theta(x) = \begin{bmatrix} 0.31 \\ 0.06 \\ 0.15 \\ 0.91 \end{bmatrix}$$

Принцип «**один-против-всех**»

$$h_\Theta(x) \in \mathbb{R}^4 \qquad h_\Theta(x) = \begin{bmatrix} P(x - \text{автомобиль}) \\ P(x - \text{пешеход}) \\ P(x - \text{велосипед}) \\ P(x - \text{паровоз}) \end{bmatrix}$$

Как определить, к какому классу k относится x?

$$\boxed{k = \operatorname{argmax} h_\Theta(x)}$$

Функция argmax возвращает индекс элемента вектора, значение которого максимально

$k = 1$ – автомобиль $\quad k = 3$ – велосипед

$k = 2$ – пешеход $\quad k = 4$ – паровоз

Многоклассовая классификация

В задаче классификации на 10 классов мы используем нейронную сеть с тремя слоями. Скрытый (второй) слой имеет 5 нейронов. Если используется подход «один-против-всех», сколько элементов содержит матрица $\Theta^{(2)}$?

- ☒ 50
- ☑ 60
- ☒ 55
- ☒ 65
- ☒ 10