

Искусственный интеллект

Нейронные сети.

Нейронная сеть - математическая модель, а также её 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, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса[2]. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

С точки зрения **машинного обучения**, нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа;

С точки зрения **математики**, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации;

С точки зрения **кибернетики**, нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники;

С точки зрения **развития вычислительной техники и программирования**, нейронная сеть — способ решения проблемы эффективного параллелизма;

С точки зрения **искусственного интеллекта**, ИНС является основой философского течения коннекционизма и основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью комп

Нейронные сети не программируются в привычном смысле этого слова, они обучаются.

Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Нейроны в ней распределены по L уровням, на каждом из которых находятся L_k нейронов.

Для наглядности каждый нейрон, кроме входных и выходных, представлен в виде двух узлов: один суммирует входящие сигналы, второй преобразует их.

Обозначение 1. Нейроны и значение в каждом из них будем обозначать как U_{ij} , где i - номер уровня, а j - номер нейрона на уровне.

Обозначение 2. Запись W_{lij} будет обозначать связь между j -м нейроном уровня l с i -м нейроном уровня $l + 1$.

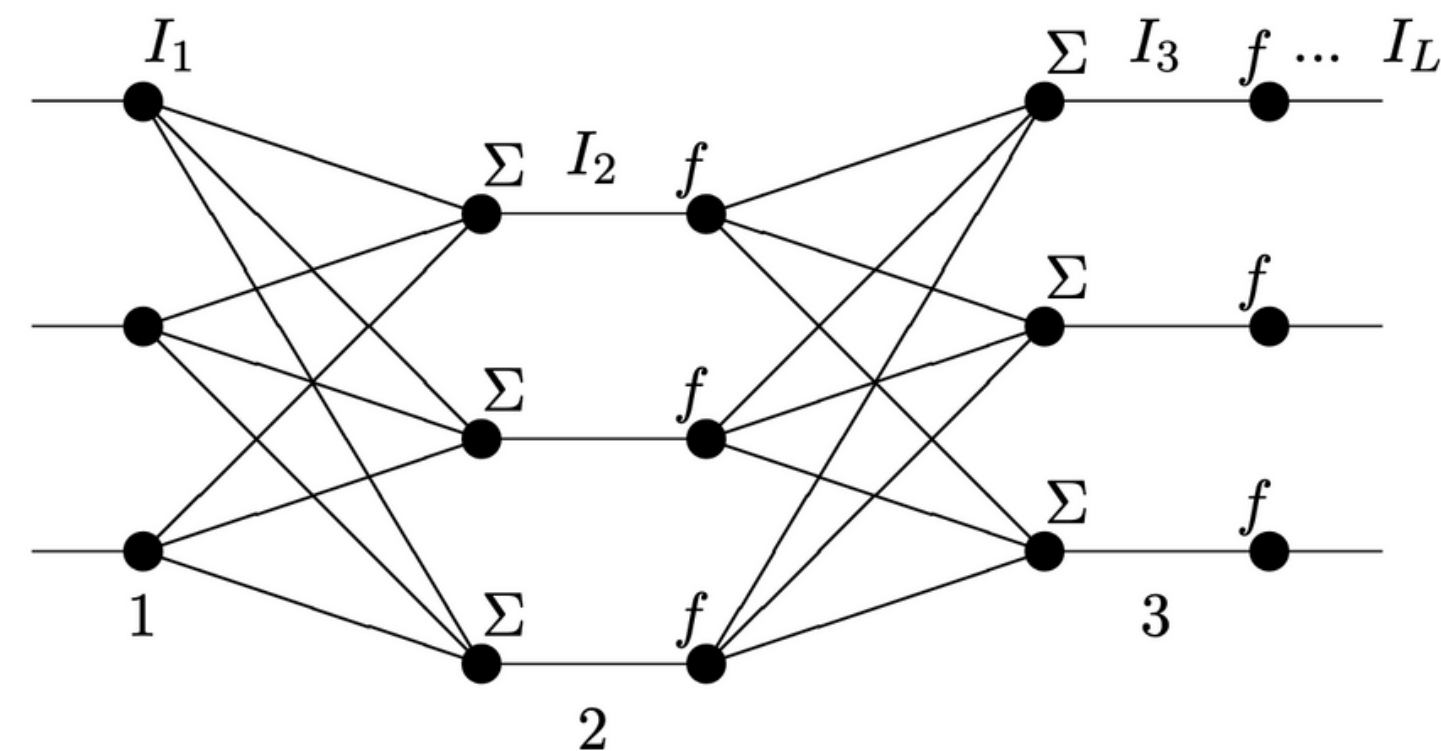
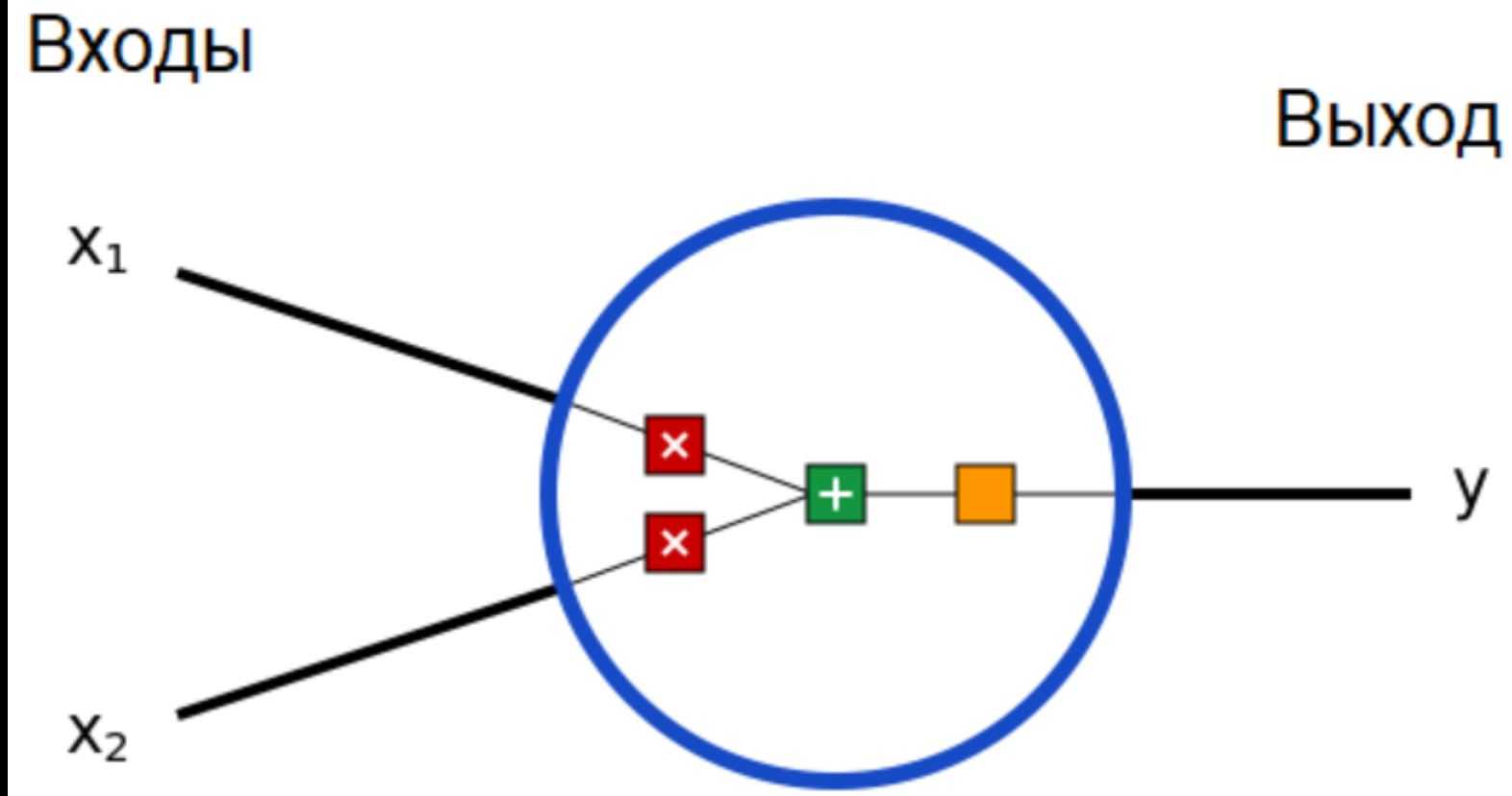


Рис. 1. Нейронная сеть.

Нейрон принимает несколько входов, выполняет над ними кое-какие математические операции, а потом выдает один выход. Вот как выглядит нейрон с двумя входами.



Внутри нейрона происходят три операции. Сначала значения входов умножаются на **веса**:

$$x_1 \rightarrow x_1 * w_1, x_2 \rightarrow x_2 * w_2$$

Затем взвешенные входы складываются, и к ним прибавляется значение **порога** b :

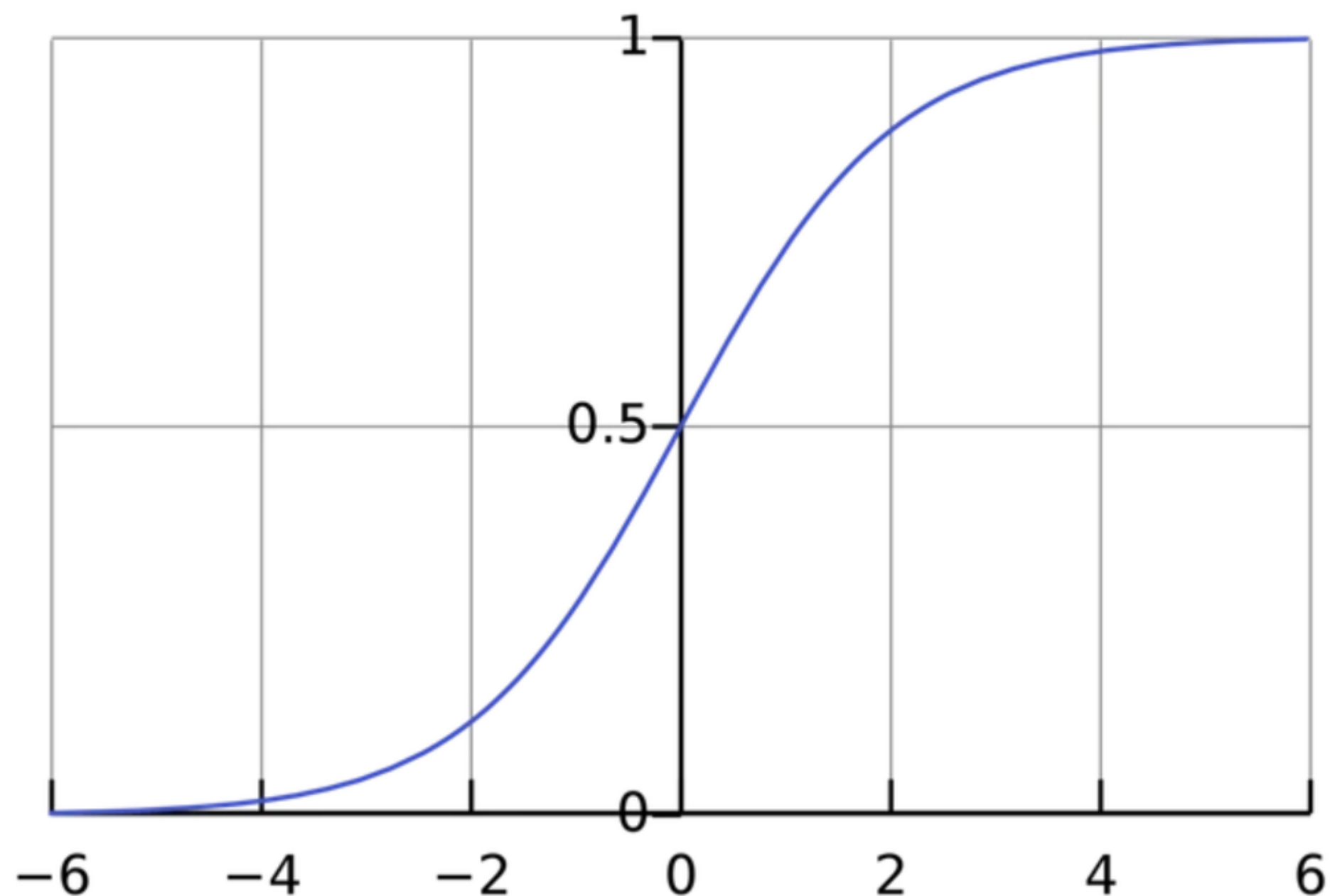
$$x_1 * w_1 + x_2 * w_2 + b$$

Наконец, полученная сумма проходит через **функцию активации**:

$$y = f(x_1 * w_1 + x_2 * w_2 + b)$$

Сигмоида выдает результаты в интервале $(0, 1)$. Можно представить, что она «упаковывает» интервал от минус бесконечности до плюс бесконечности в $(0, 1)$: большие отрицательные числа превращаются в числа, близкие к 0, а большие положительные – к 1.

Функция активации преобразует неограниченные значения входов в выход, имеющий ясную и предсказуемую форму. Одна из часто используемых функций активации – **сигмоида**:



Сигмоида

Простой пример

Допустим, наш двухвходовой нейрон использует сигмоидную функцию активации и имеет следующие параметры:

$$w = [0, 1] \quad b = 4$$

$w=[0, 1]$ – это всего лишь запись $w_1=0$, $w_2=1$ в векторном виде. Теперь зададим нашему нейрону входные данные: **$x=[2, 3]$** . Мы используем **скалярное произведение векторов**, чтобы записать формулу в сжатом виде:

$$(w \cdot x) + b = ((w_1 * x_1) + (w_2 * x_2)) + b = 0 * 2 + 1 * 3 + 4 = 7$$

$$y = f(w \cdot x + b) = f(7) = 0.999$$

Наш нейрон выдал 0.999 при входах $x=[2, 3]$. Вот и все! Процесс передачи значений входов дальше, чтобы получить выход, называется **прямой связью (feed forward)**.

Создание простой нейронной сети

NumPy

NumPy
Why NumPy? Powerful n-dimensional arrays.
Numerical computing tools. Interoperable.
Performant. Open source.

```
import numpy as np

def sigmoid(x):
    # Наша функция активации:  $f(x) = 1 / (1 + e^{-x})$ 
    return 1 / (1 + np.exp(-x))

class Neuron:
    def __init__(self, weights, bias):
        self.weights = weights
        self.bias = bias

    def feedforward(self, inputs):
        # Умножаем входы на веса, прибавляем порог, затем используем функцию активации
        total = np.dot(self.weights, inputs) + self.bias
        return sigmoid(total)

weights = np.array([0, 1]) # w1 = 0, w2 = 1
bias = 4                  # b = 4
n = Neuron(weights, bias)

x = np.array([2, 3])      # x1 = 2, x2 = 3
print(n.feedforward(x))   # 0.9990889488055994
```

Дополнительные практические задания

<https://proglib.io/p/pishem-neyroset-na-python-s-nulya-2020-10-07>