Современные методы анализа данных и машинного обучения

Тема 8. Лекция 11

Основы глубинного обучения. Введение в нейронные сети

Юрий Саночкин

ysanochkin@hse.ru

НИУ ВШЭ, 2024

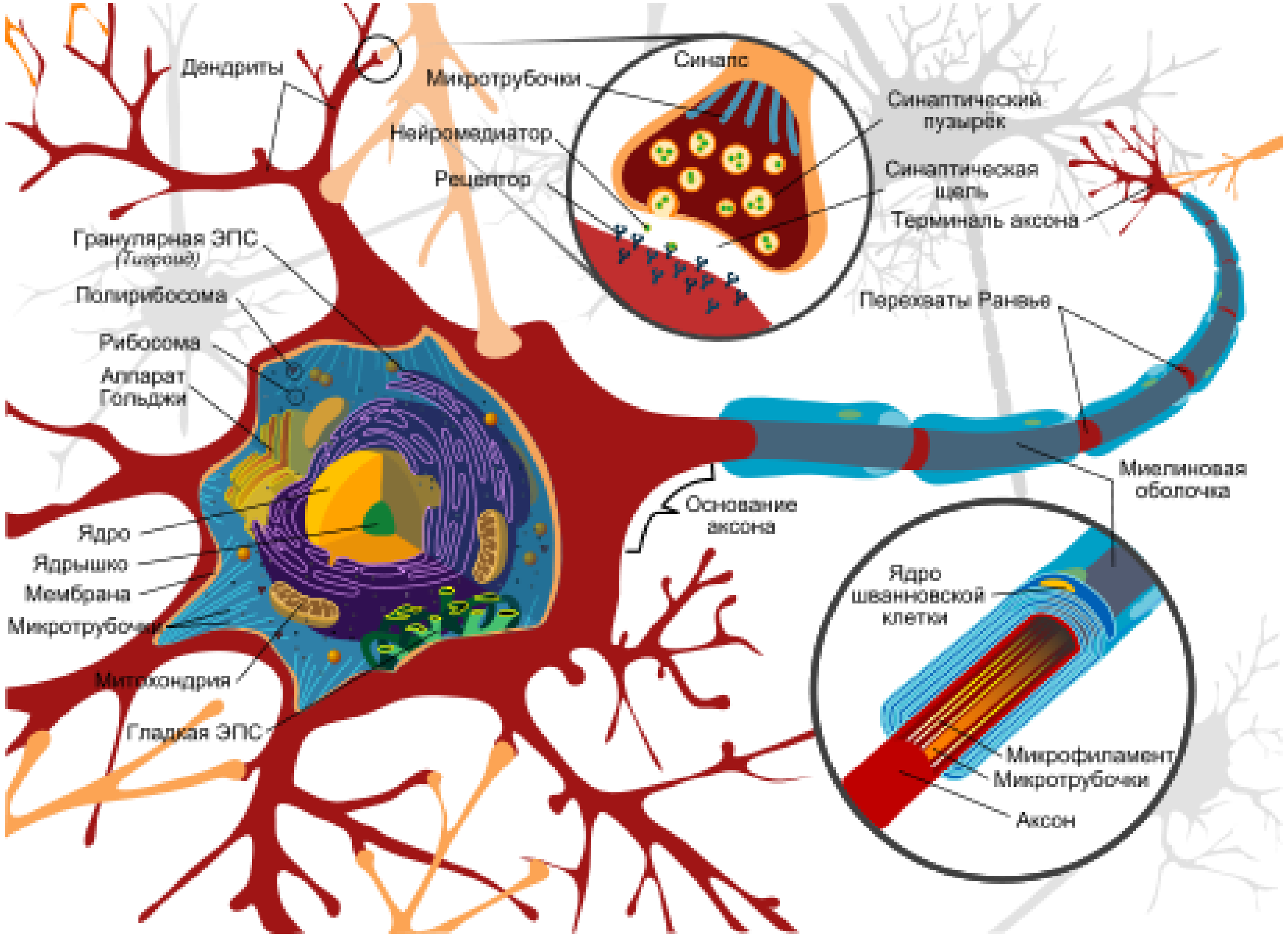
Нейронные сети

- Ну что же, вот и пришло время обсуждать великие и ужасные нейронные сети! :)
- Что вы знаете о них?

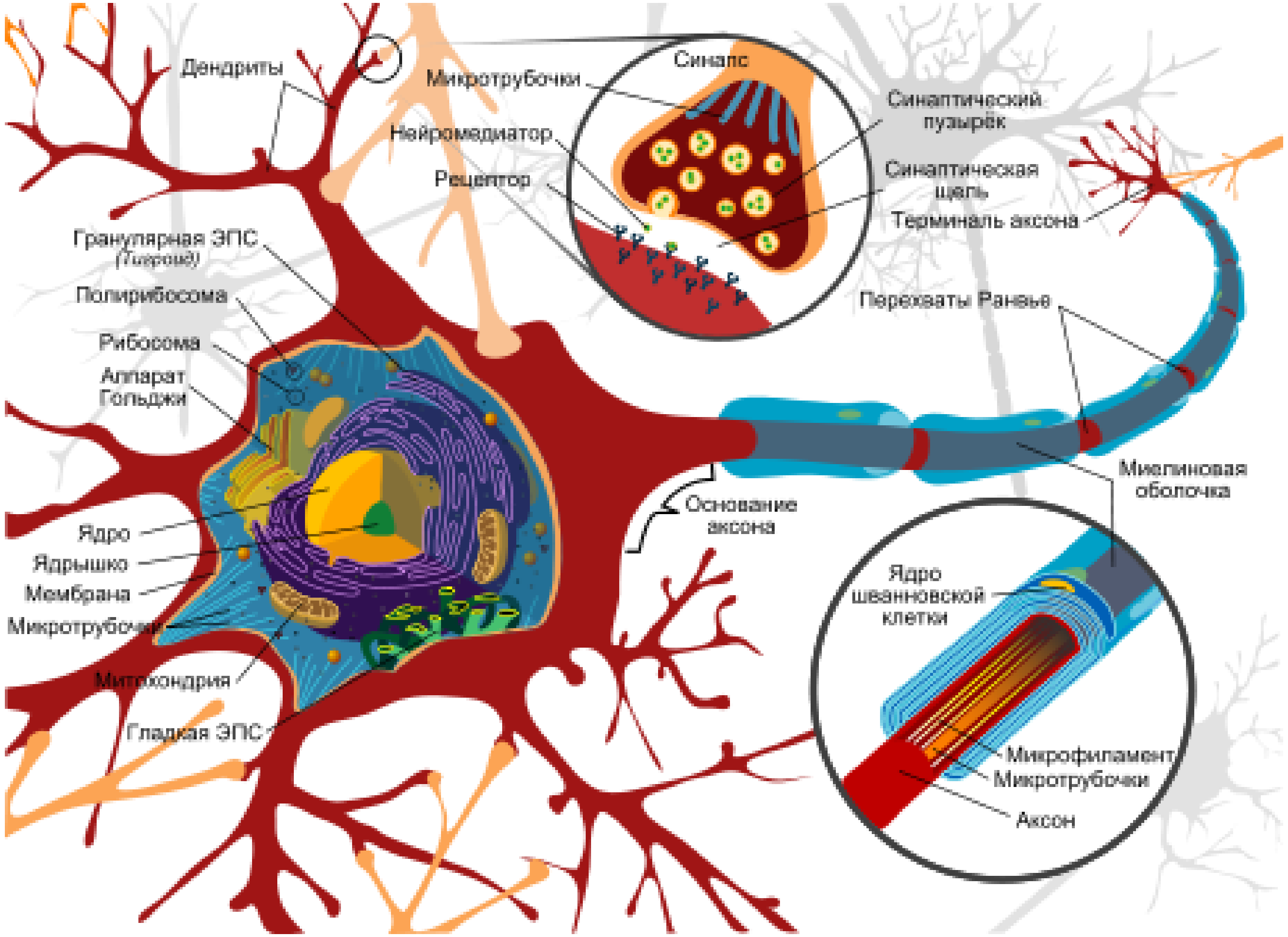
Нейронные сети

- Ну что же, вот и пришло время обсуждать великие и ужасные нейронные сети! :)
- Что вы знаете о них?
- А как вы вообще понимаете, что такое «нейрон»?

Биологический нейрон

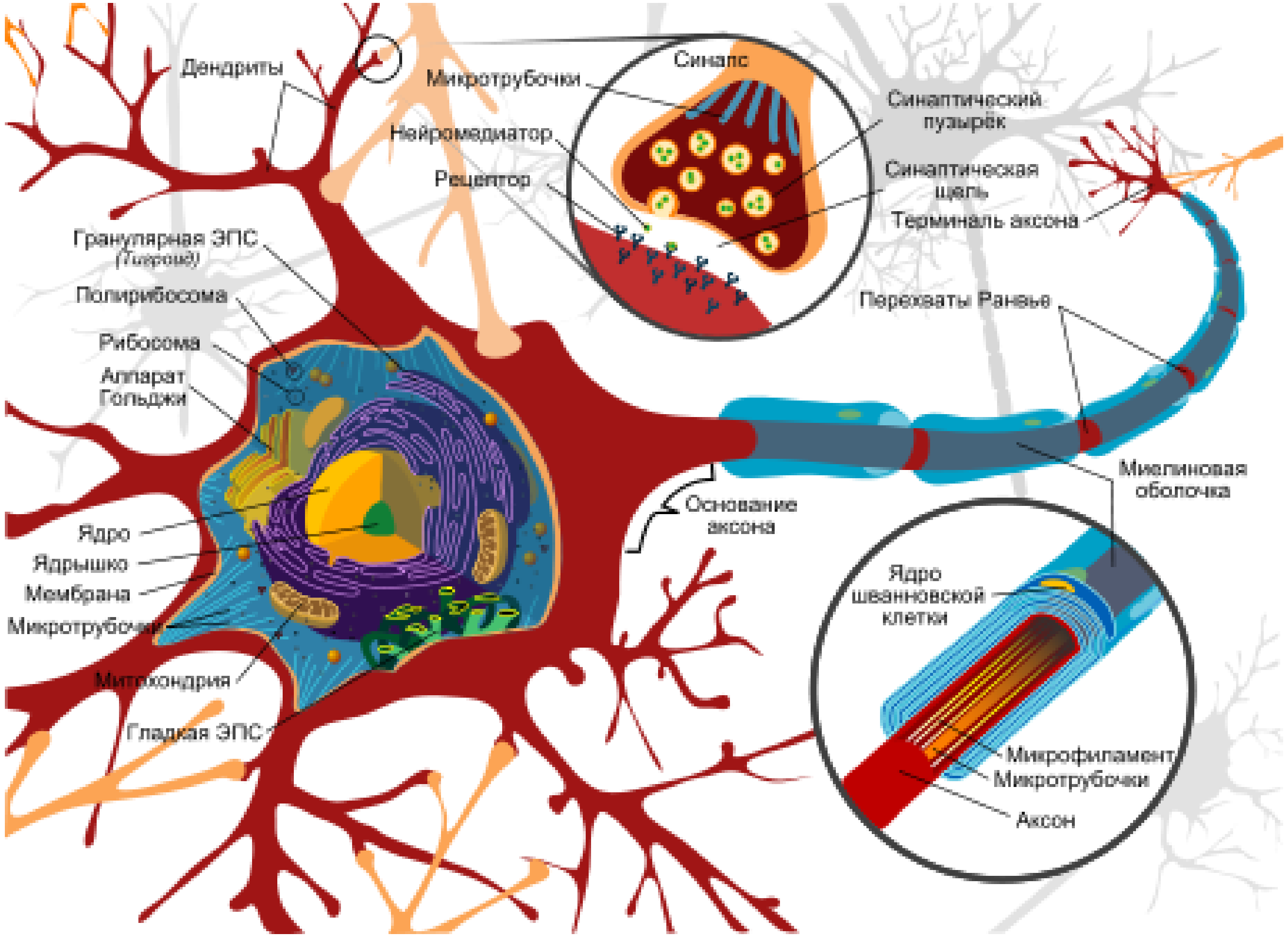


Биологический нейрон



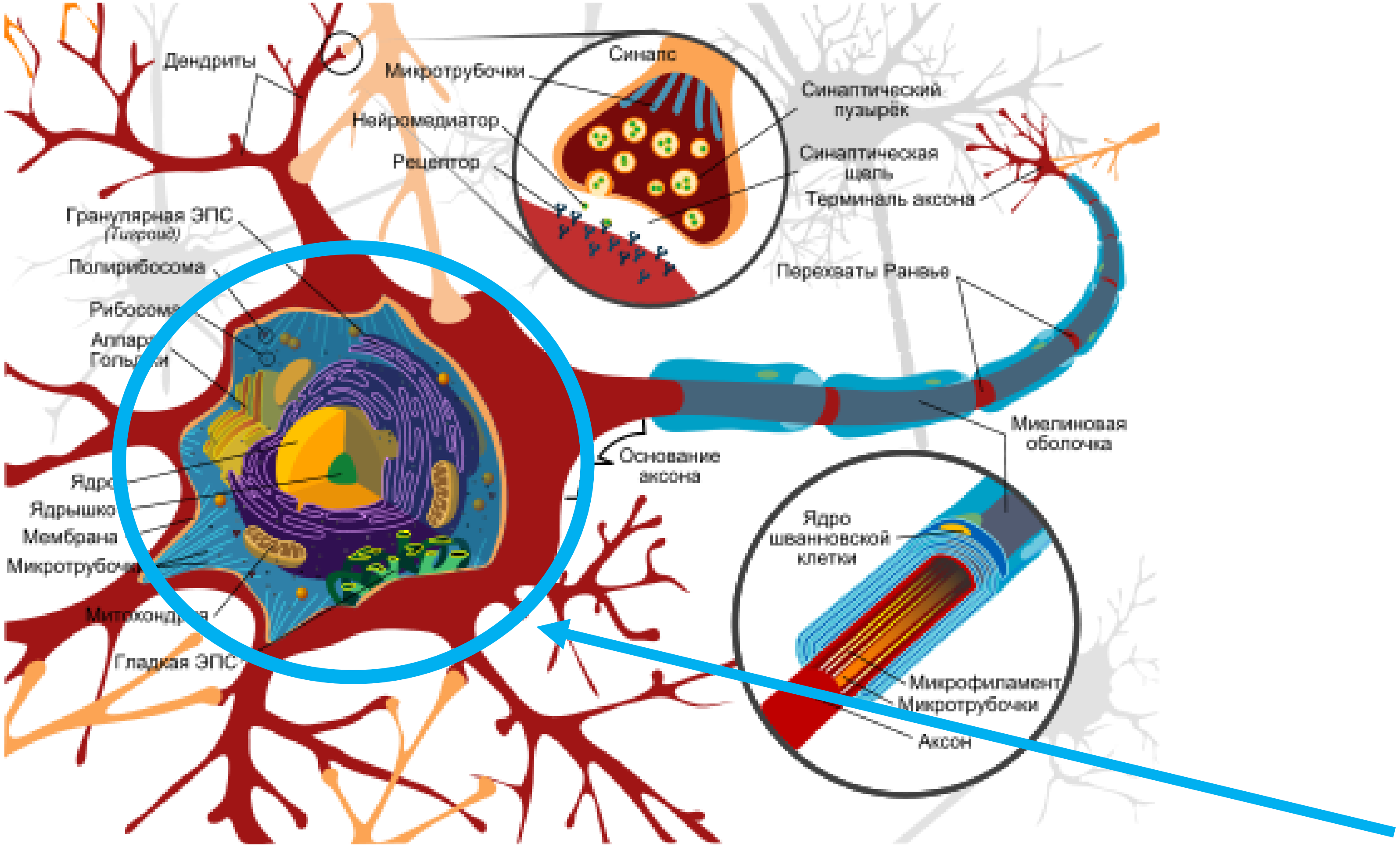
- Нейронные сети и их компоненты — нейроны — берут своё начало (и даже название) от понятия «нейрон» из биологии.

Биологический нейрон



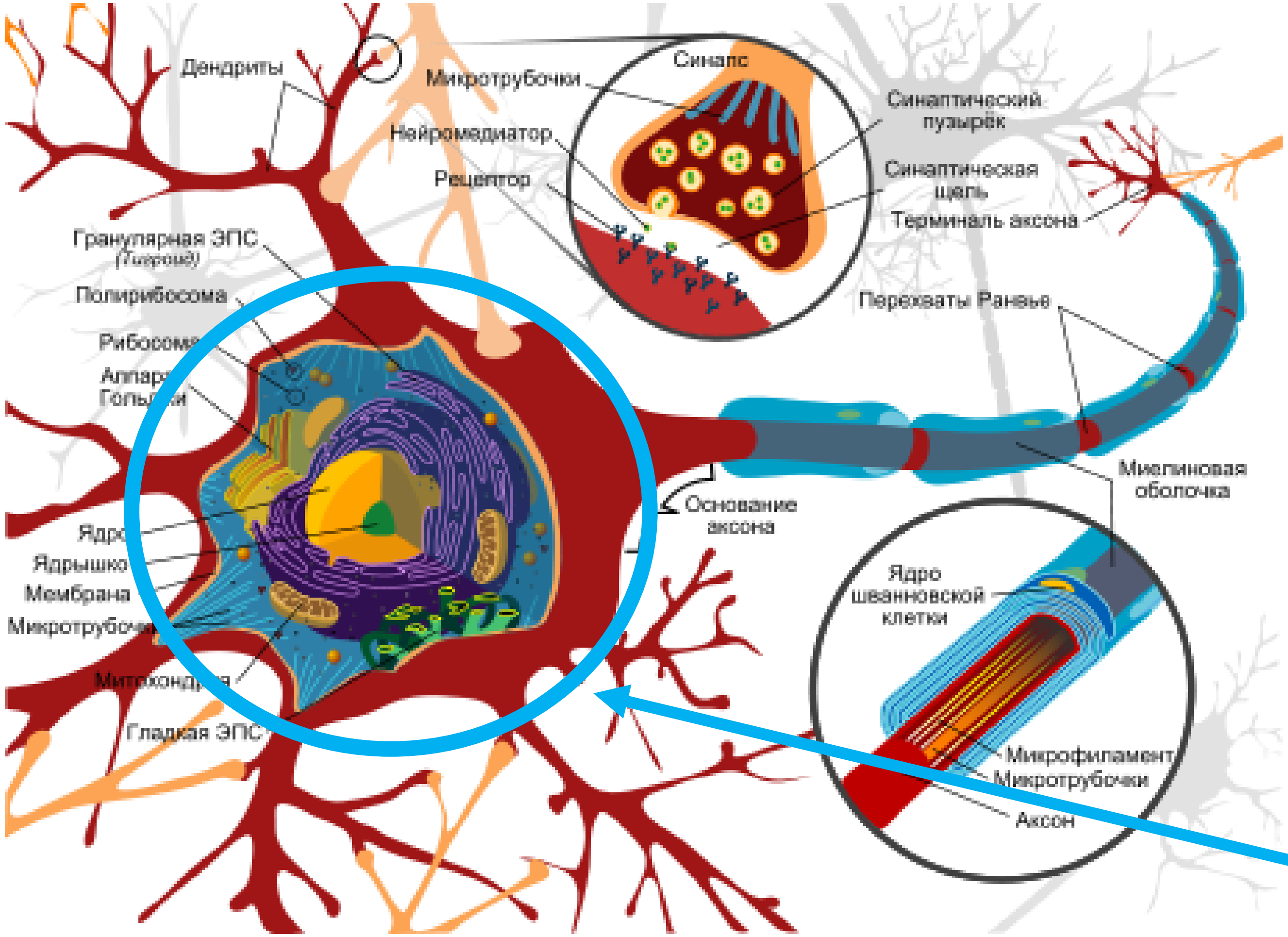
- Нейронные сети и их компоненты — нейроны — берут своё начало (и даже название) от понятия «нейрон» из биологии.
- Это не случайно и связано с очень близкой и похожей концепцией работы двух этих структур.

Биологический нейрон



Тело нейрона

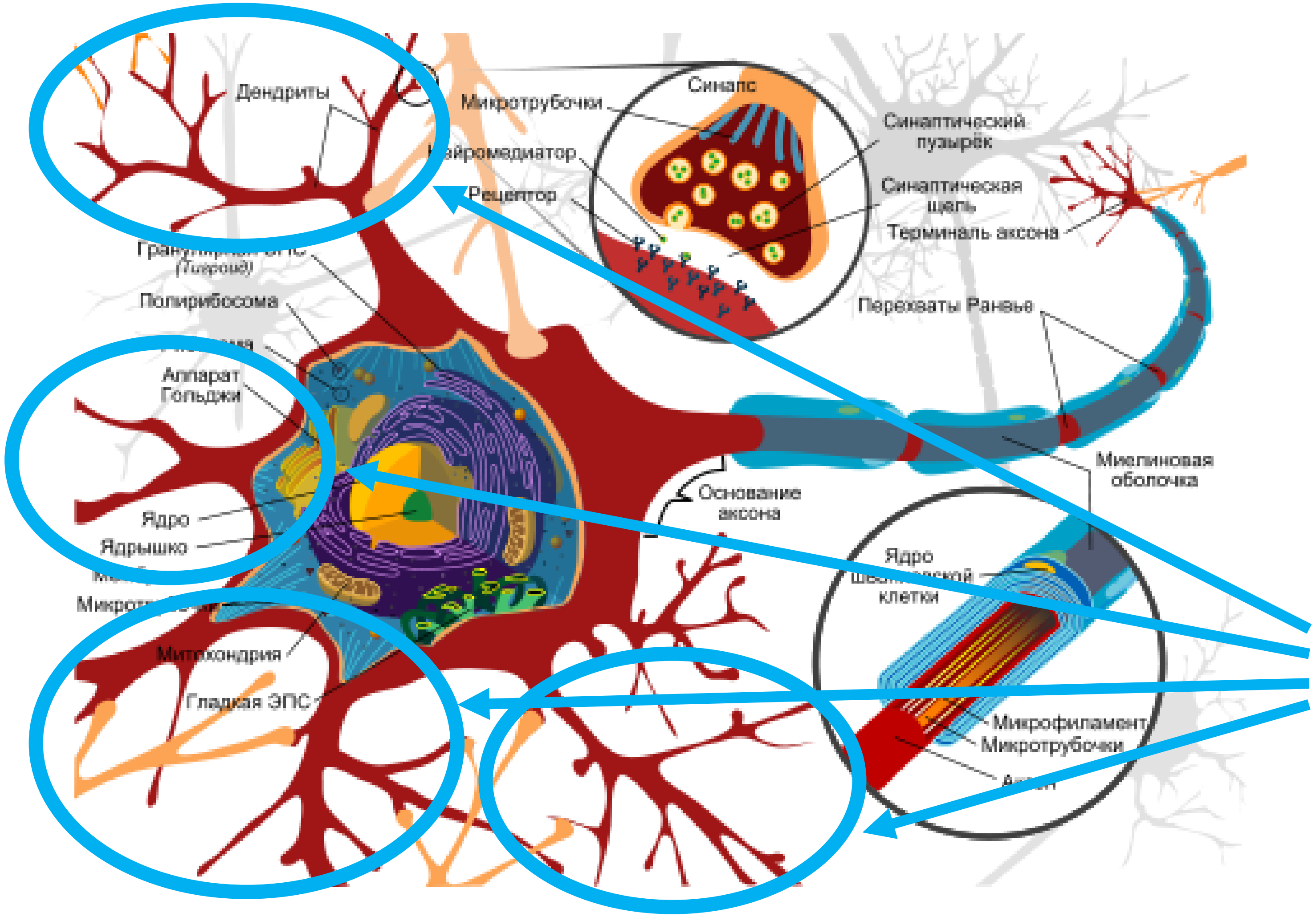
Биологический нейрон



- Свойство: накапливает электрический заряд

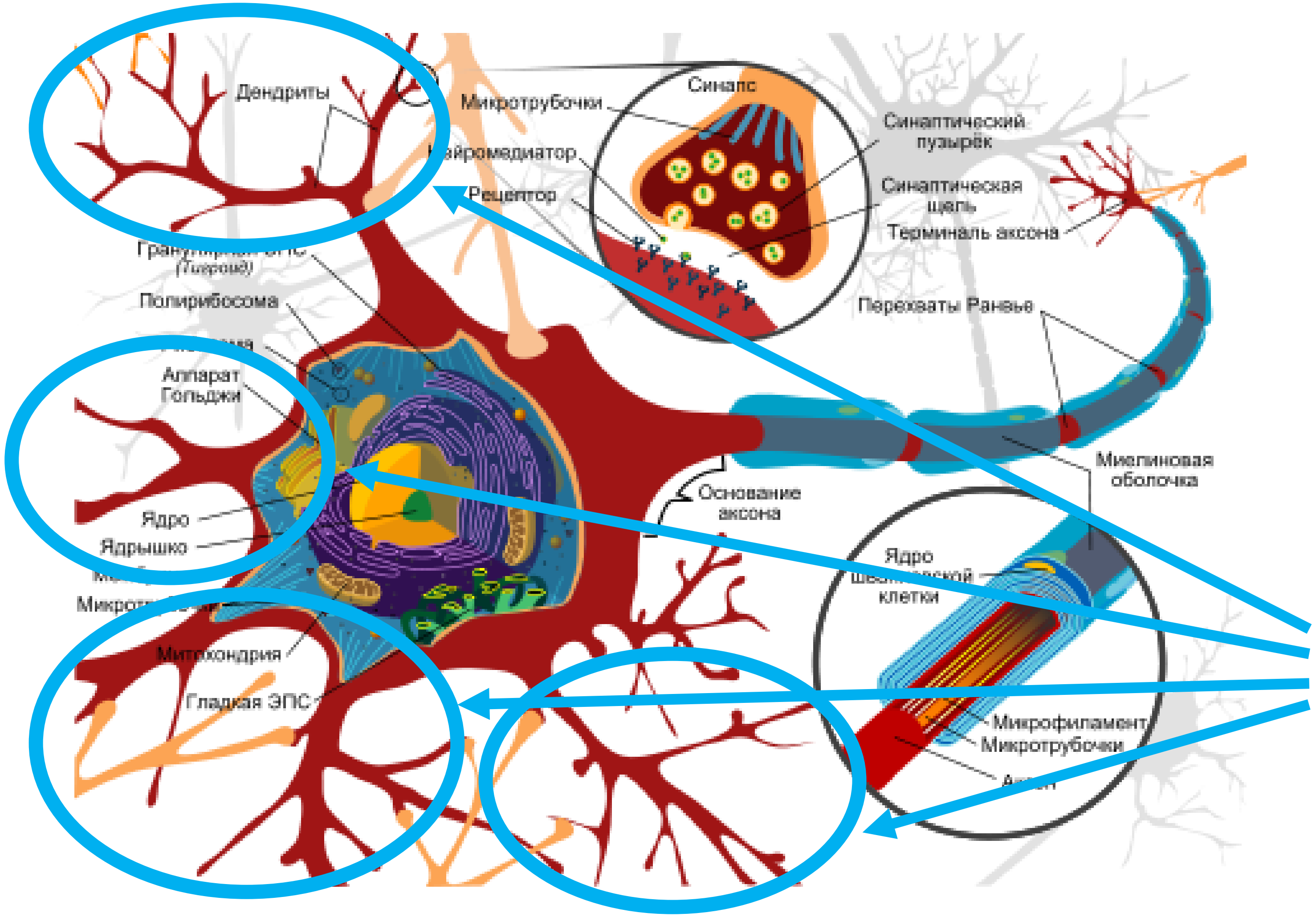
Тело нейрона

Биологический нейрон



Дендриты

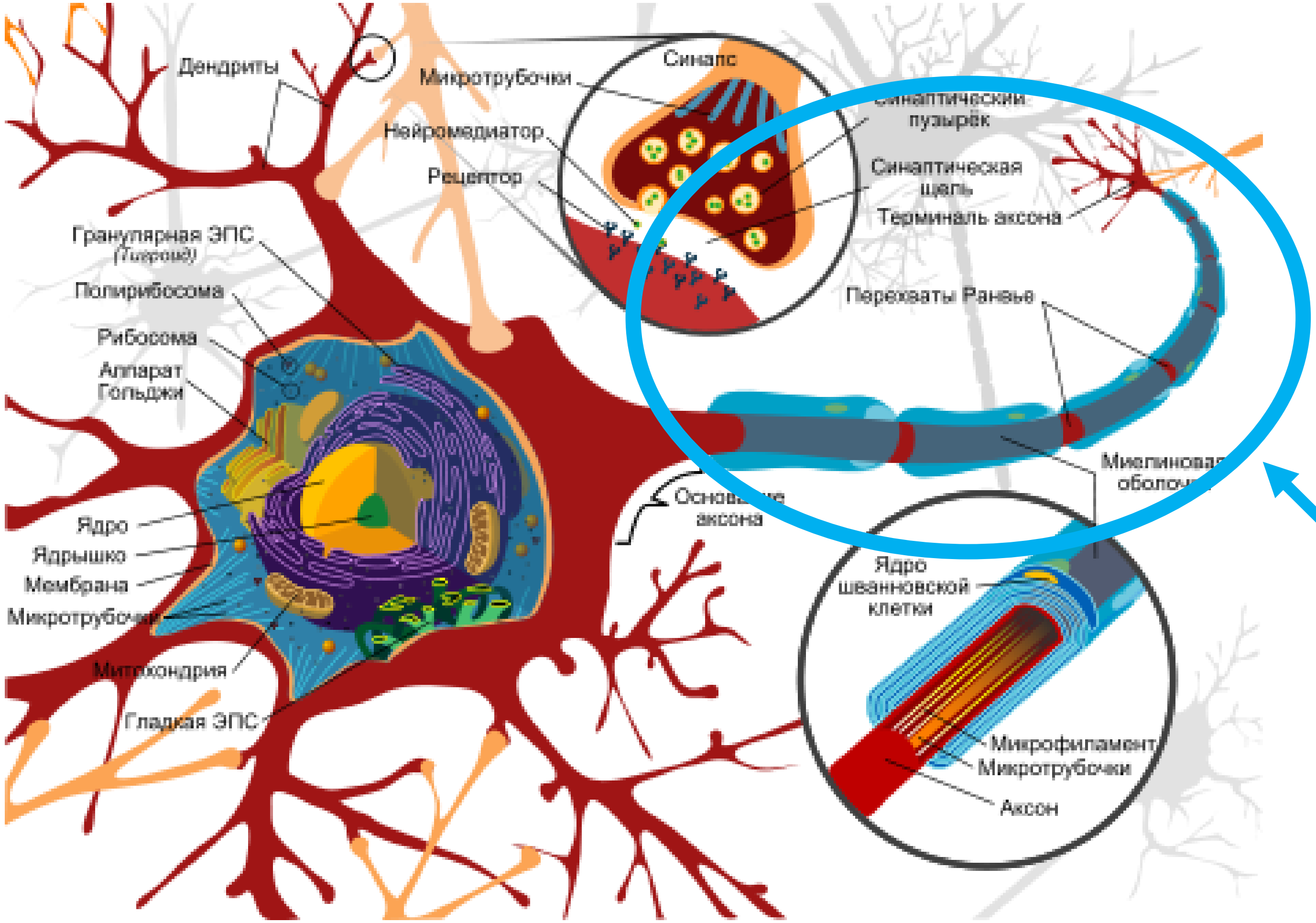
Биологический нейрон



Дендриты

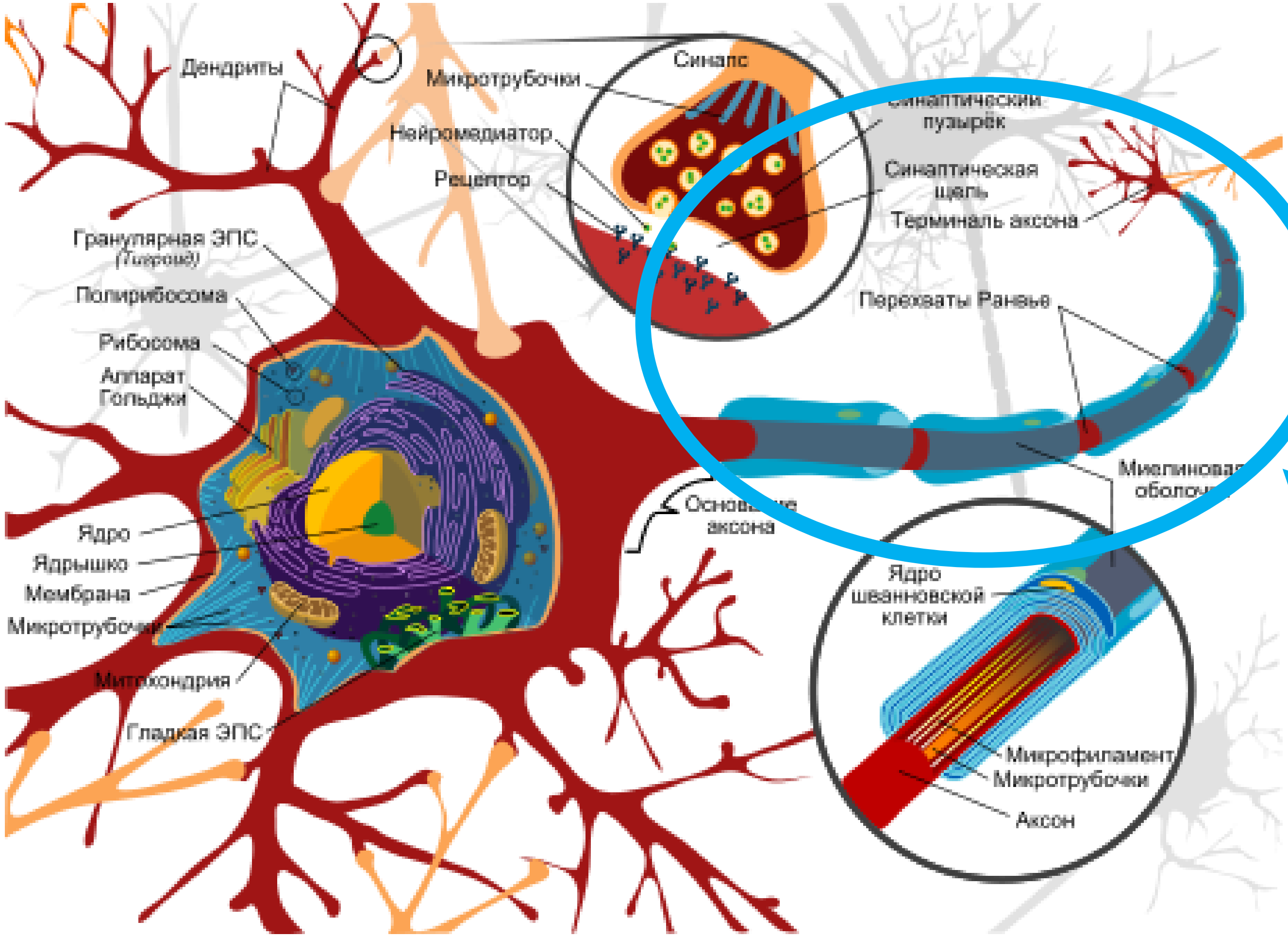
- Свойство: принимают сигналы от других нейронов

Биологический нейрон



Аксон

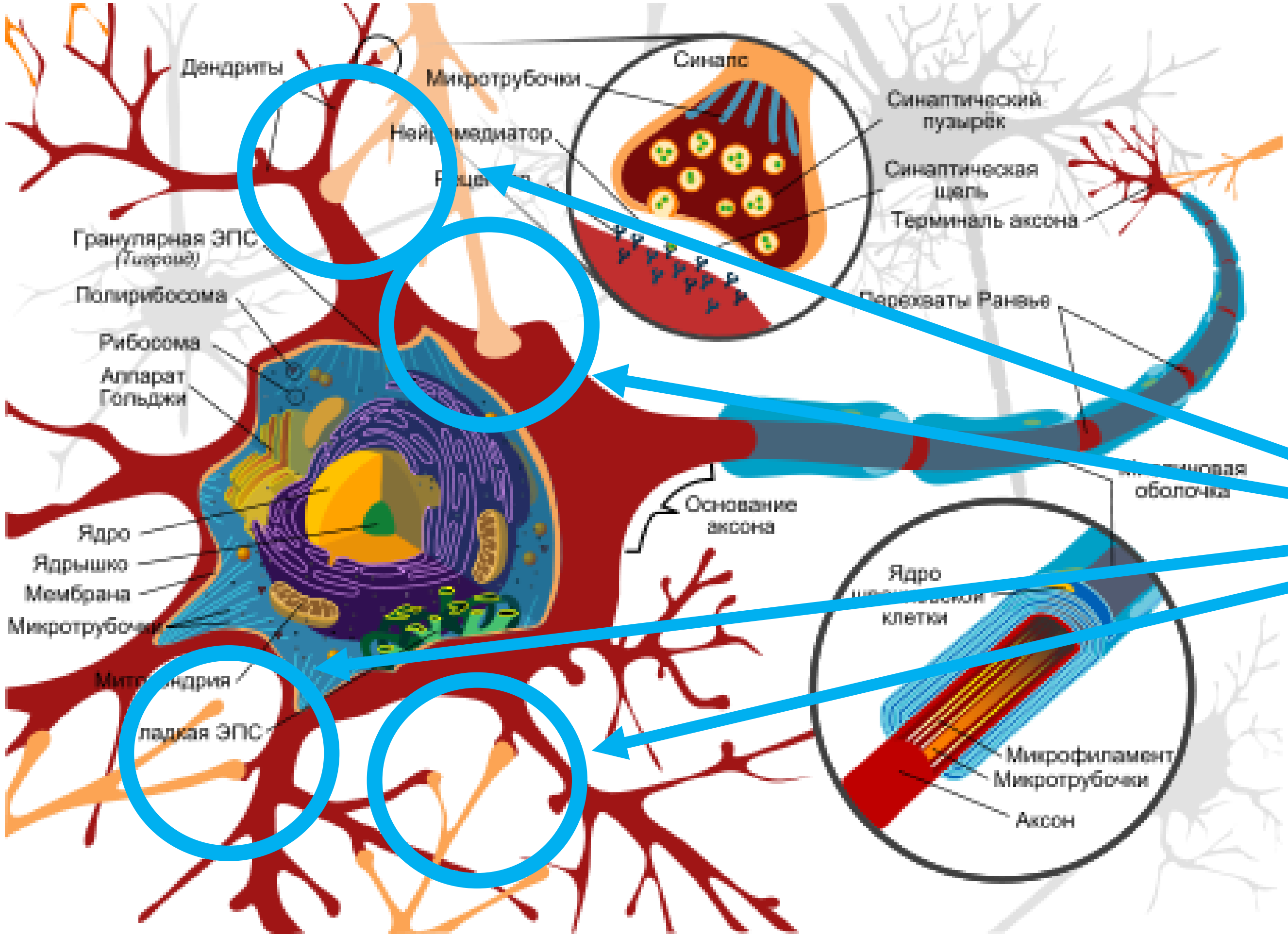
Биологический нейрон



- Свойство: передает сигнал к другому нейрону

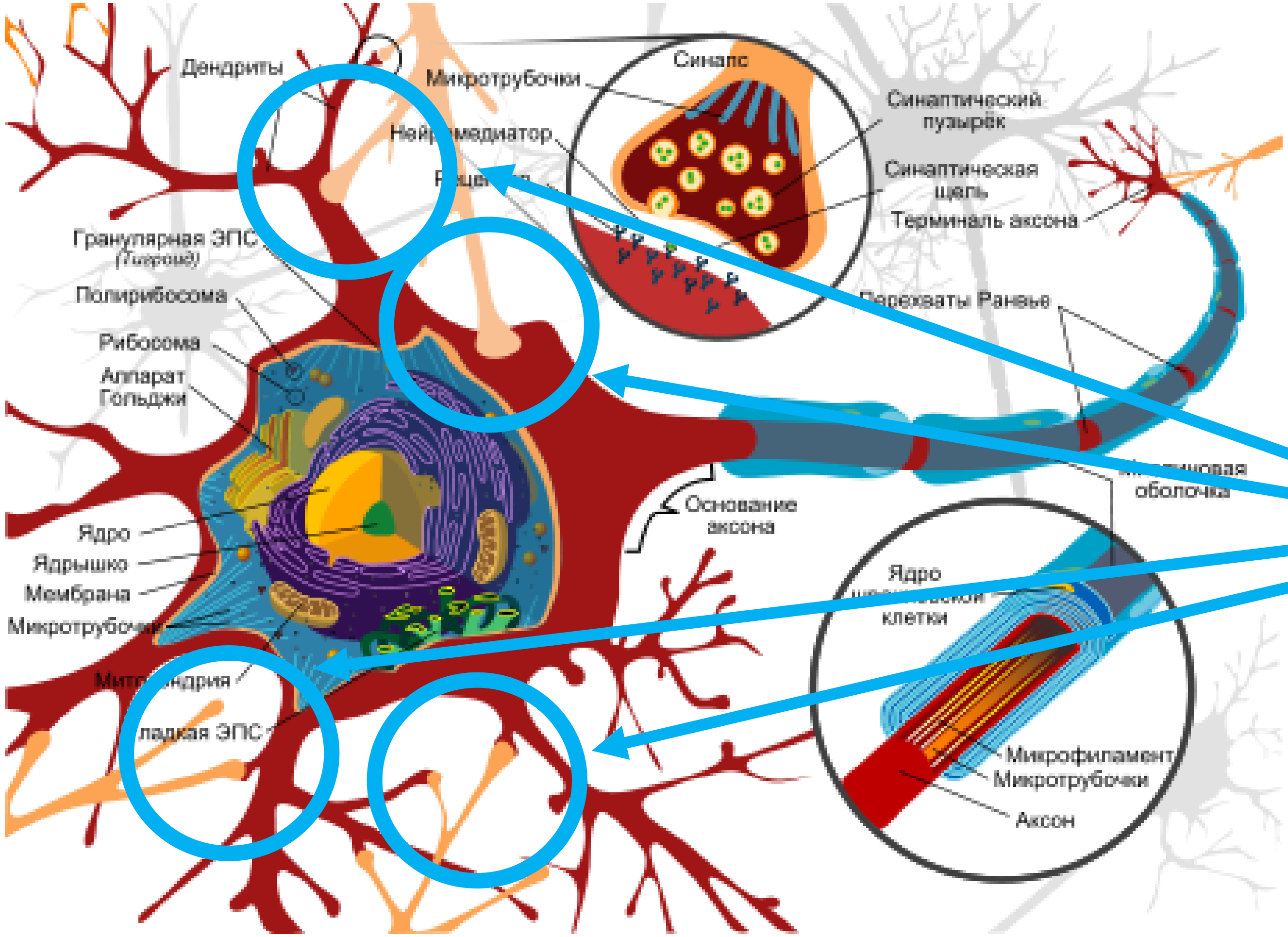
Аксон

Биологический нейрон



Синапсы

Биологический нейрон



- Свойство: связывают дендрит и аксон

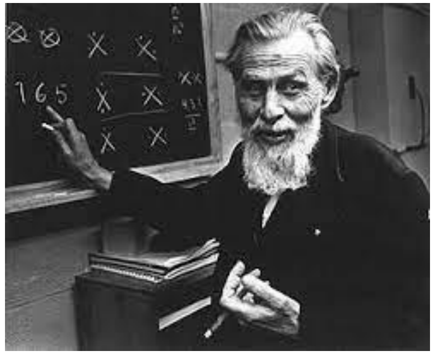
Синапсы

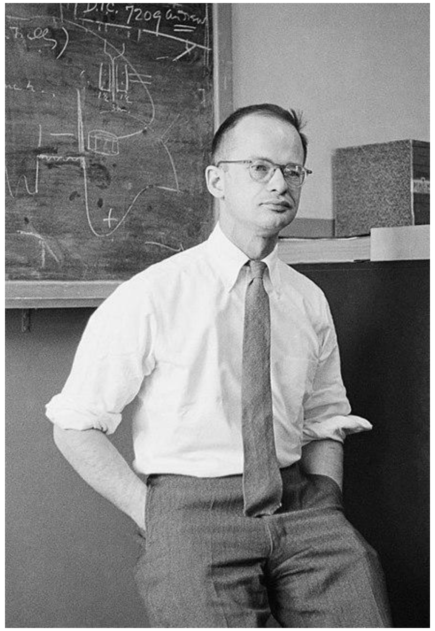
Нейронные сети

- Хотим реализовать систему, подобную биологическому нейрону.
- Необходимо накапливать и передавать сигналы.
- Хотим, чтобы наша модель была применима к различным задачам.

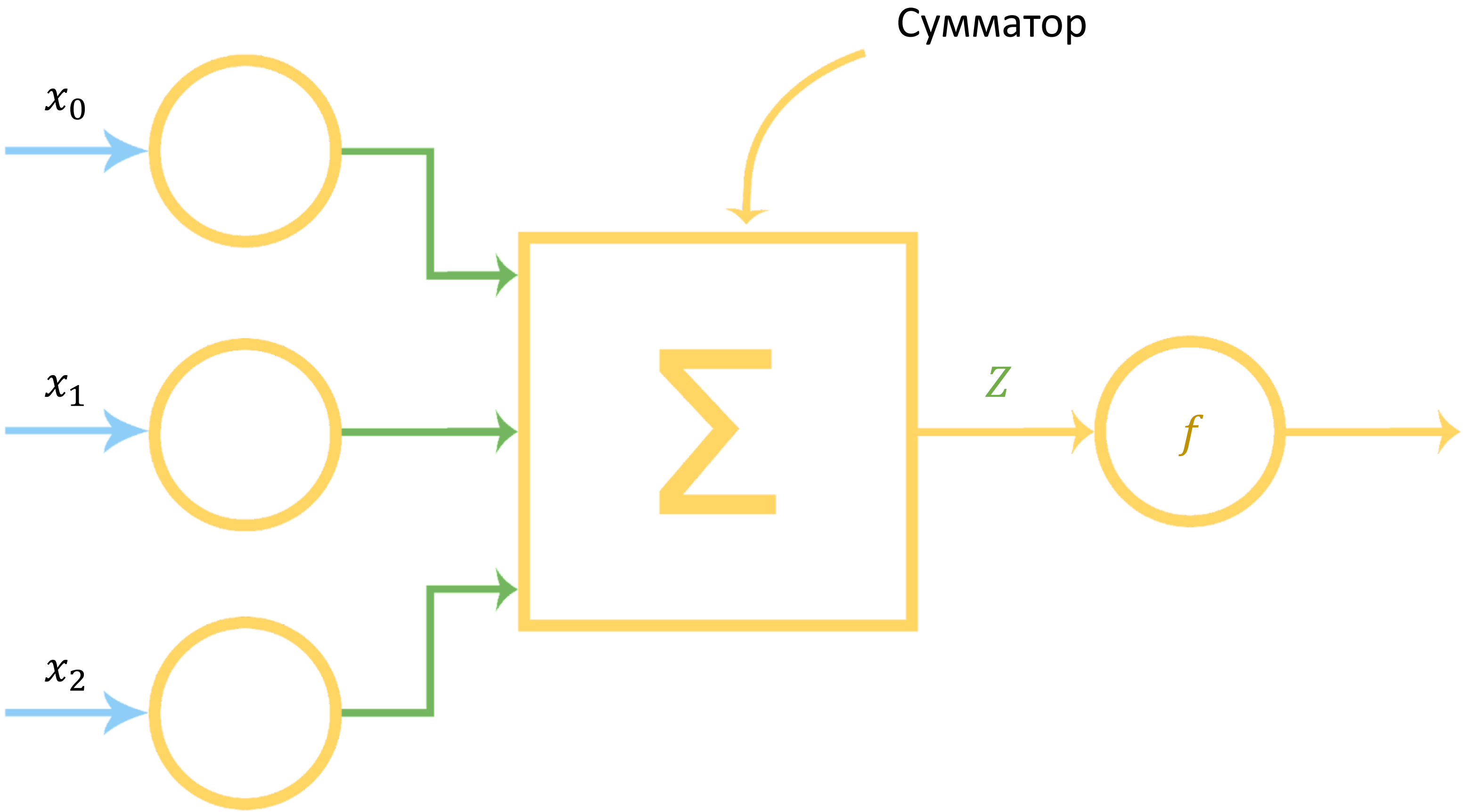
Математическая модель нейрона

- В качестве математической модели нейрона далее будет рассмотрена модель Маккаллока-Питтса — это они представлены на фотографиях справа.
- Очень важная ремарка: данная модель НЕ является реальной моделью биологического нейрона, а лишь отражает некоторые сходные процессы, происходящие в реальном нейроне, с помощью математических операций.

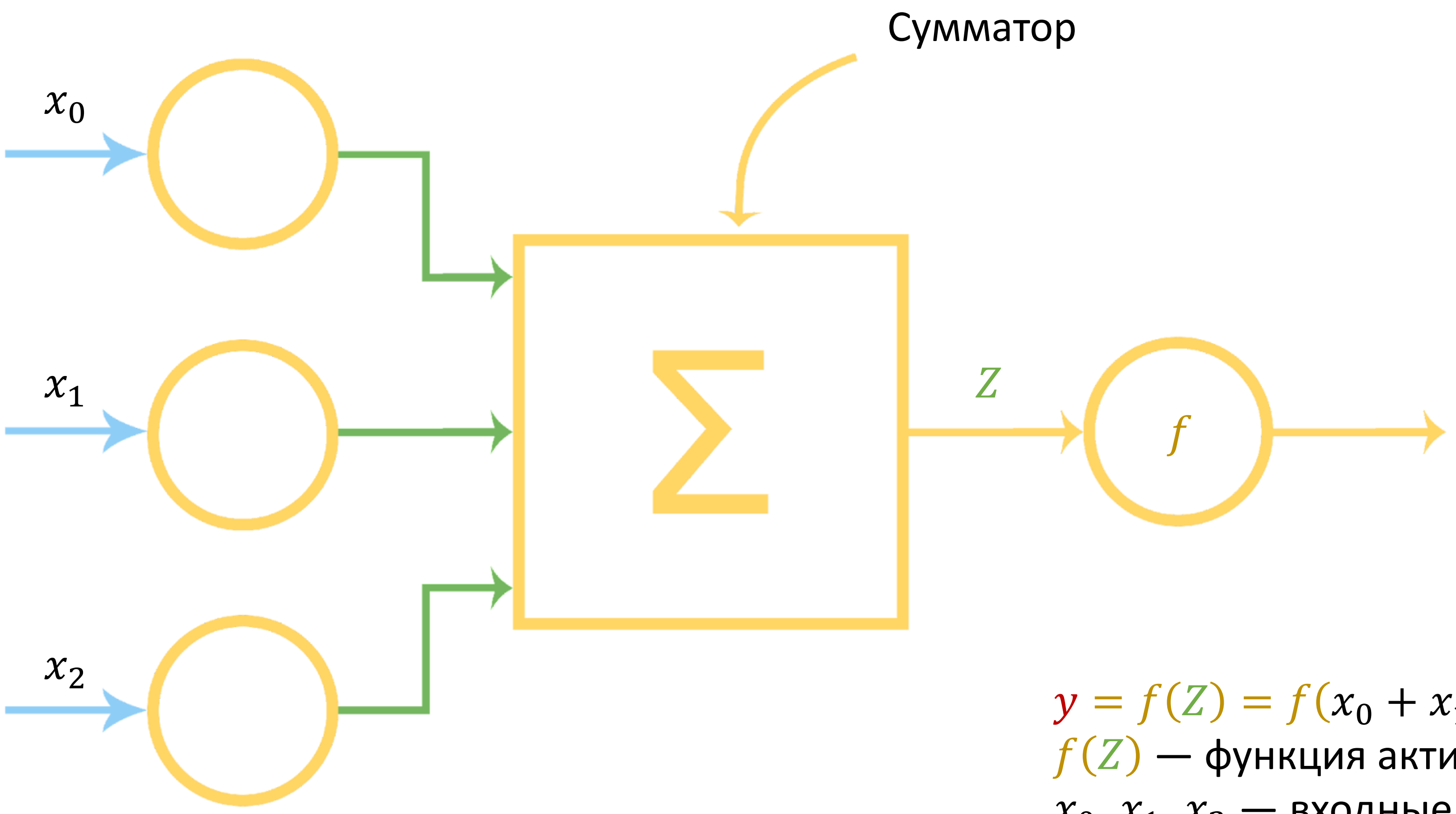




Математическая модель нейрона



Математическая модель нейрона

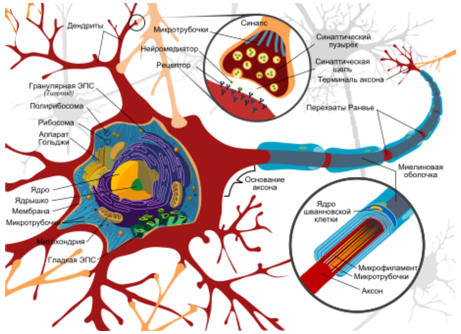


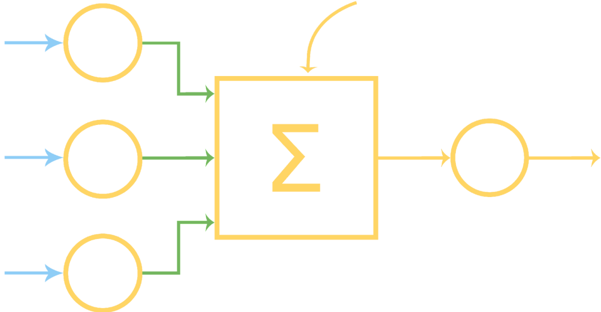
$y = f(Z) = f(x_0 + x_1 + x_2)$

$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

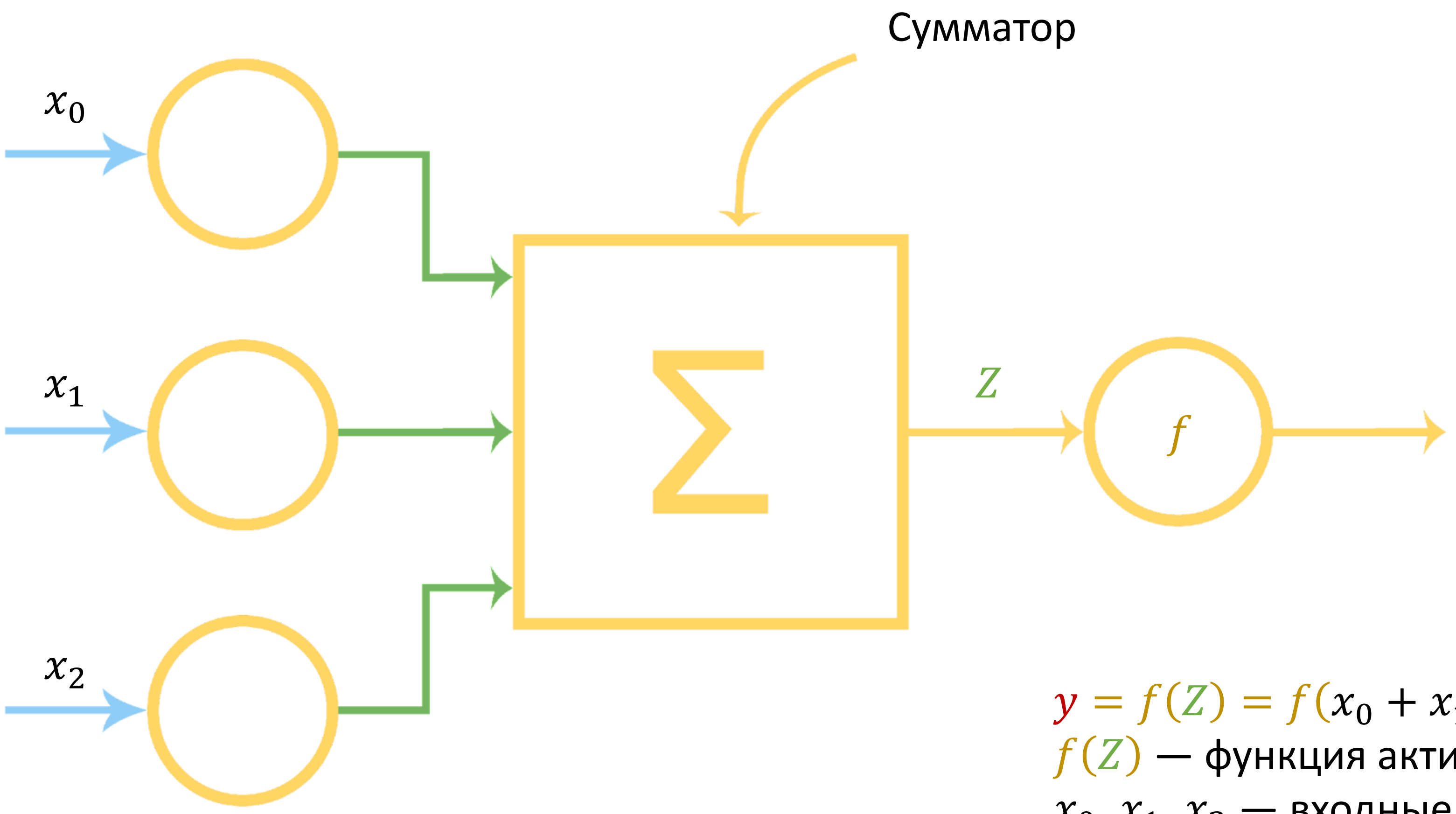
Математическая модель нейрона





- В математической модели нейрона тело нейрона заменяется на сумматор. Входы в сумматор заменяют дендриты. Аксон — заменяется на выход y.
- Биологический нейрон накапливает заряд до тех пор, пока этот заряд не достигнет какого-то значения, и только после этого этот заряд уходит по аксону к другим нейронам.
- Эту роль в нашей модели выполняет функция активации.

Математическая модель нейрона

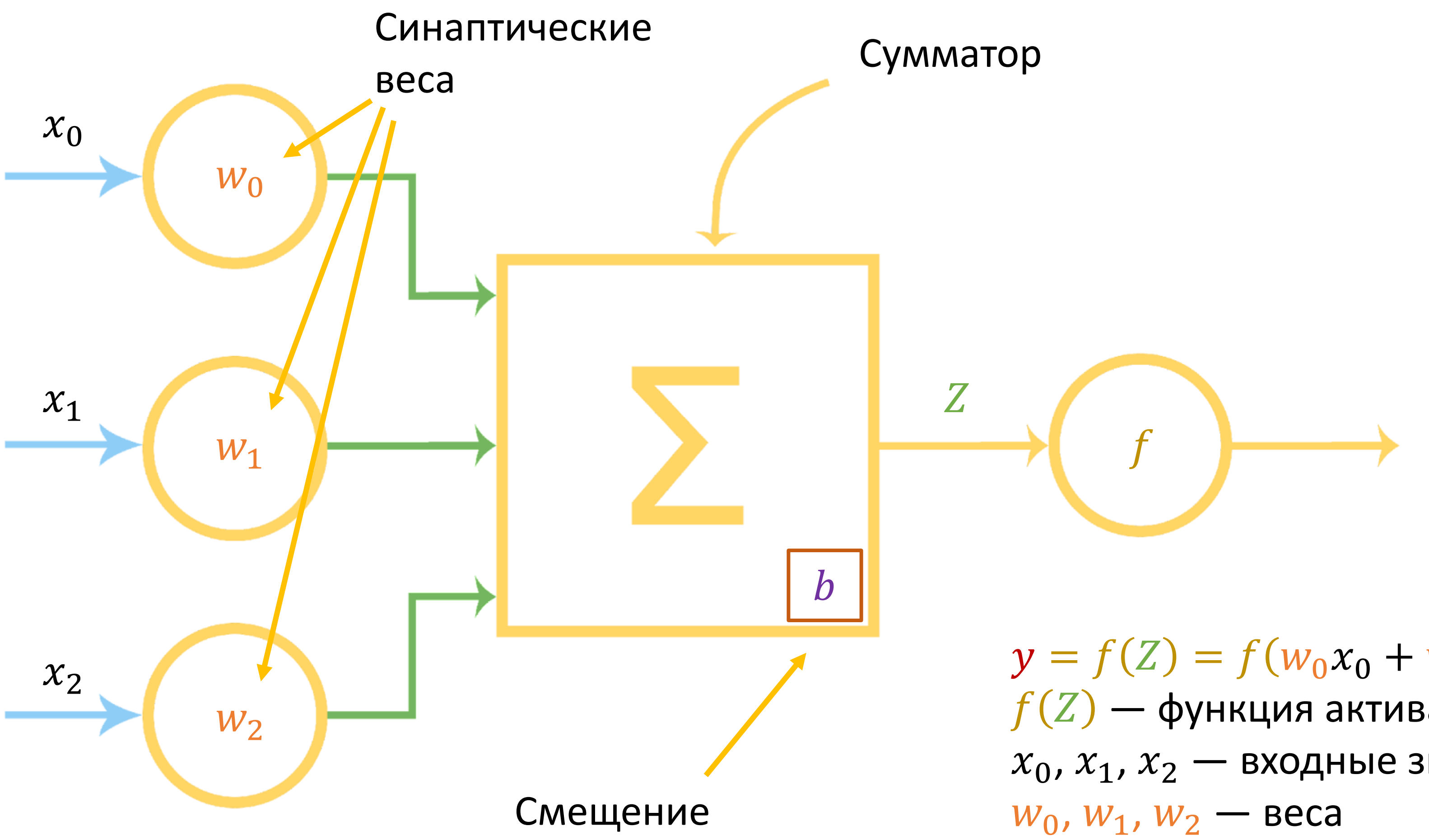


$y = f(Z) = f(x_0 + x_1 + x_2)$

$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

Математическая модель нейрона



$y = f(Z) = f(w_0 x_0 + w_1 x_1 + w_2 x_2 + b)$

$f(Z)$ — функция активации

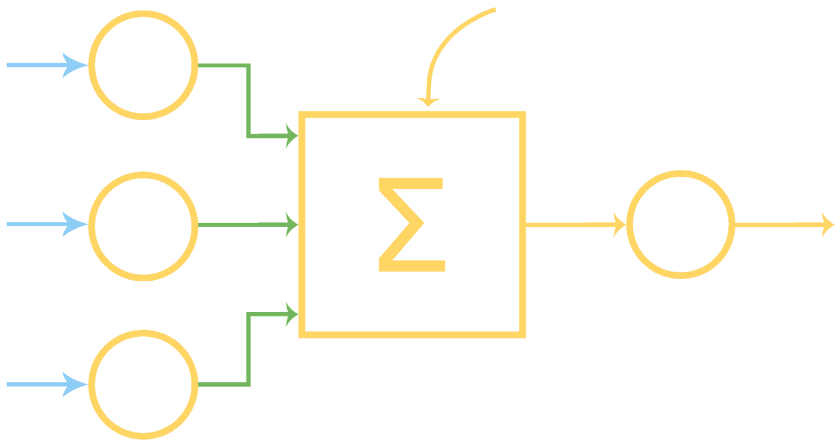
x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона

- Обучение биологической нейронной сети сводится к тому, что настраиваются синапсы, поэтому введём некоторые настраиваемые веса, которые навесим на входы в нейрон.
- Для того, чтобы описать все линейные функции, добавим смещение b. Это также обучаемый параметр.



$y = f(Z) = f(w_0 x_0 + w_1 x_1 + w_2 x_2 + b)$

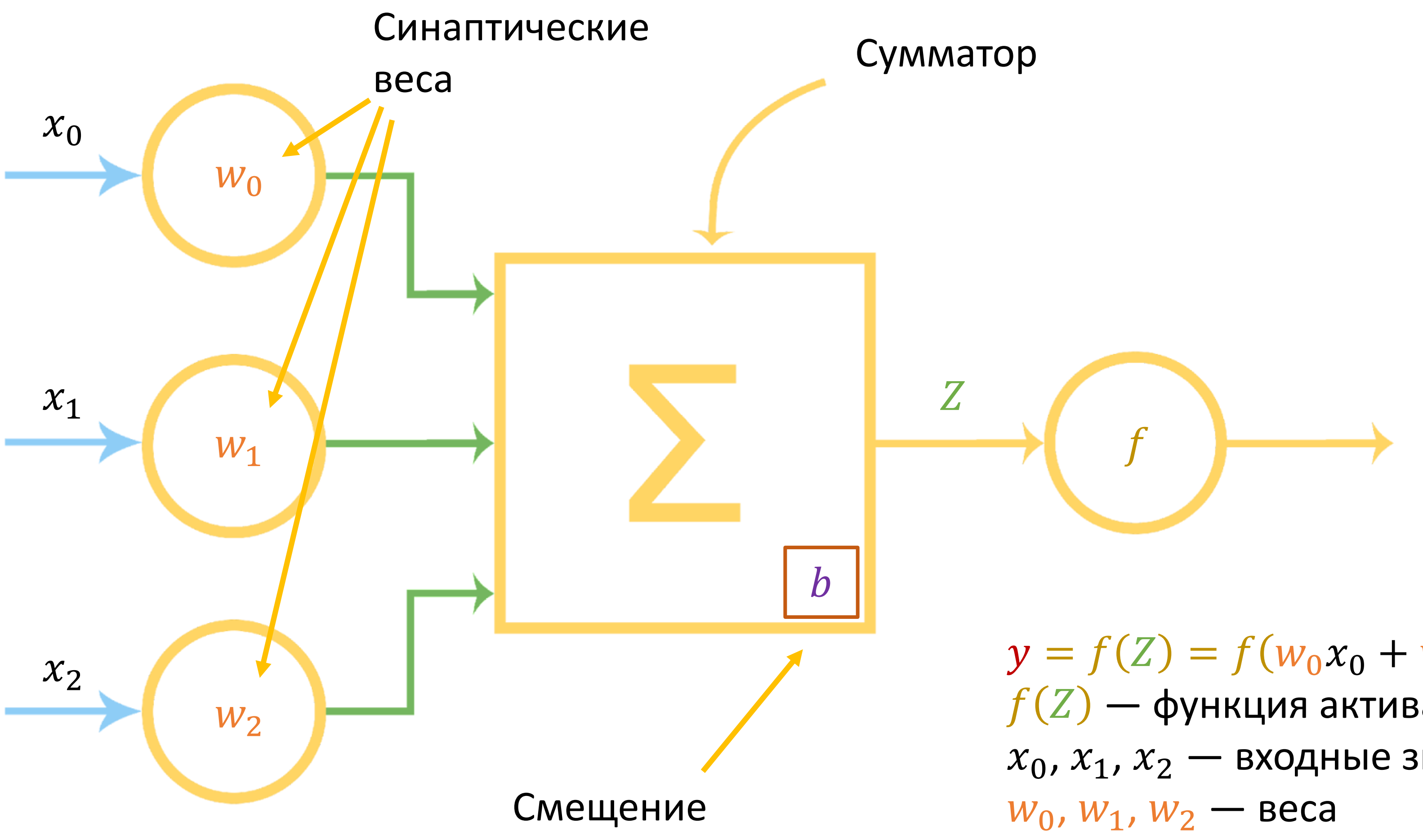
$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона



$y = f(Z) = f(w_0x_0 + w_1x_1 + w_2x_2 + b)$

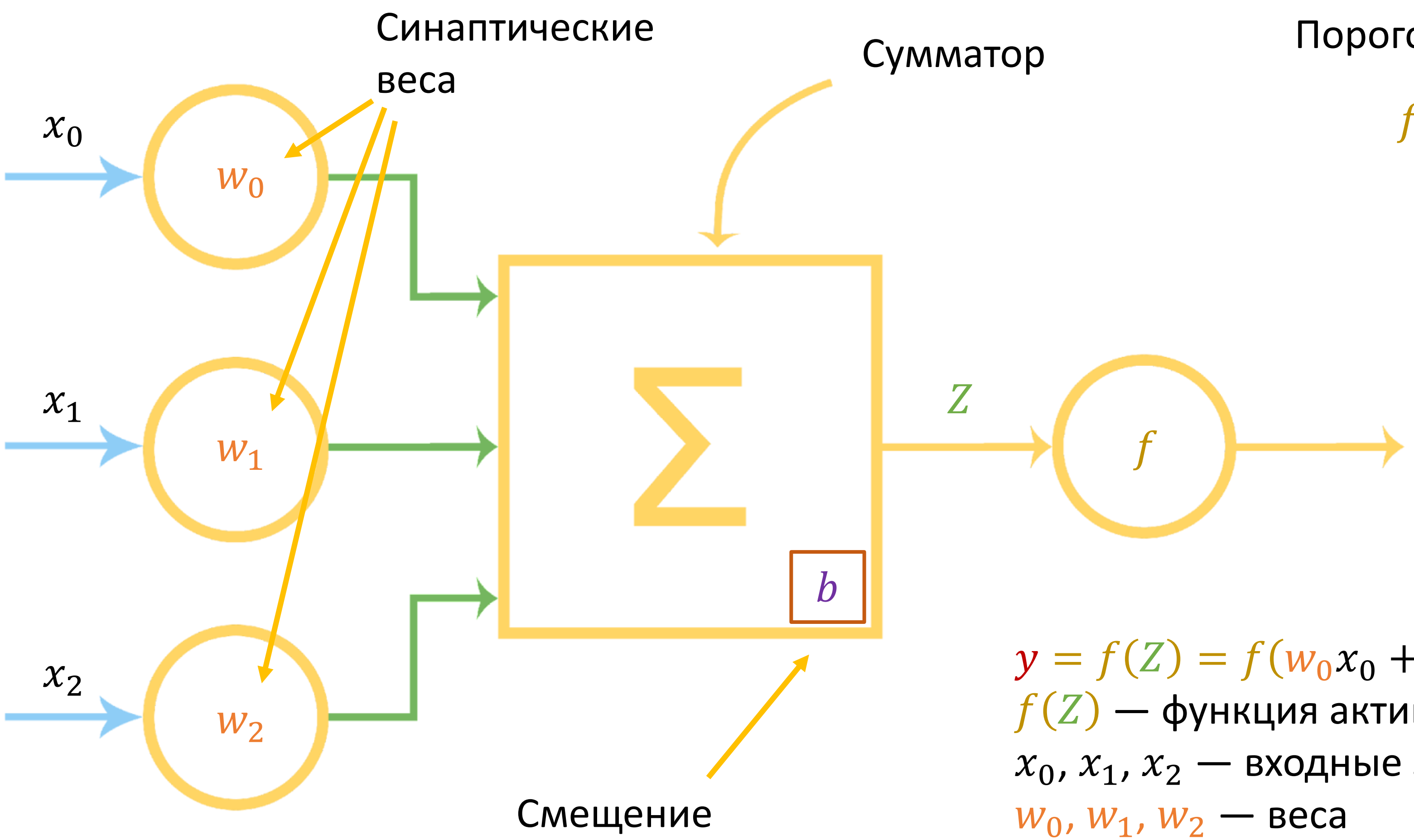
$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона



Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

$y = f(Z) = f(w_0 x_0 + w_1 x_1 + w_2 x_2 + b)$

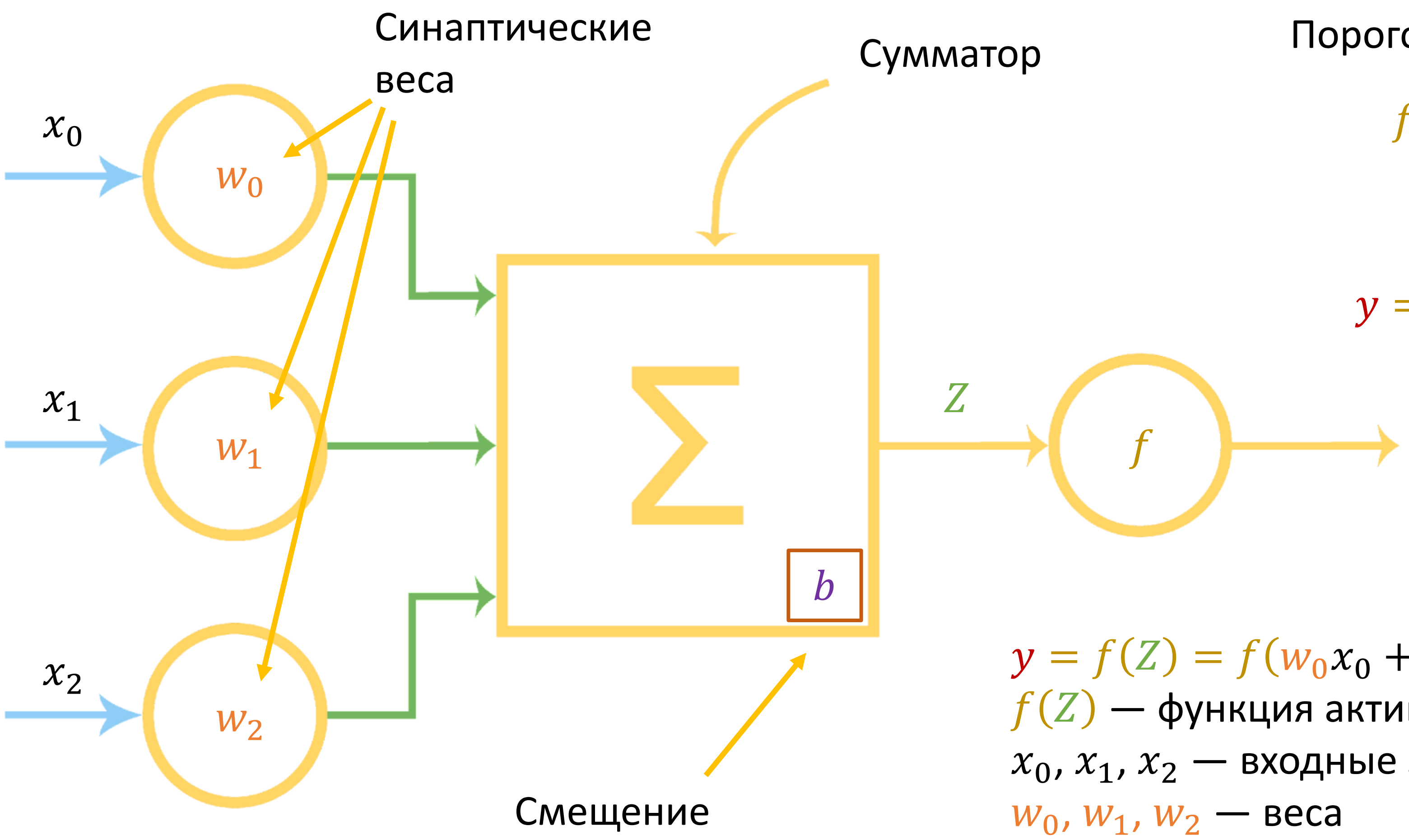
$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона



Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

$$y = f(Z) = f(\langle w, x \rangle + b)$$

$$y = f(Z) = f(w_0 x_0 + w_1 x_1 + w_2 x_2 + b)$$

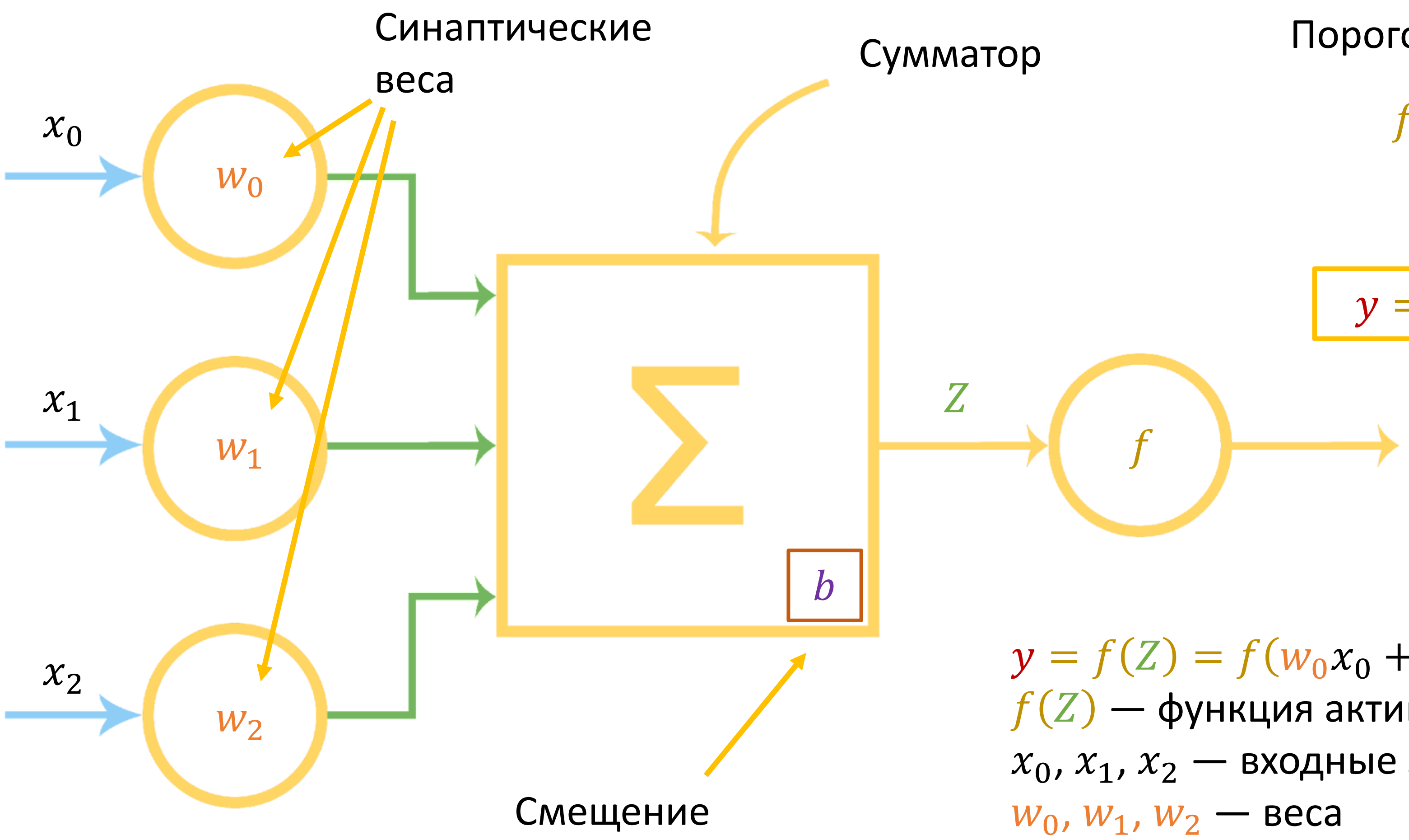
$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона



Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

$$\boxed{y = f(Z) = f(\langle w, x \rangle + b)}$$

$y = f(Z) = f(w_0 x_0 + w_1 x_1 + w_2 x_2 + b)$

$f(Z)$ — функция активации

x_0, x_1, x_2 — входные значения

w_0, w_1, w_2 — веса

b — смещение

Математическая модель нейрона

- Таким образом, мы смогли реализовать математическую модель нейрона и все её ключевые элементы: вес, смещение, функцию активации.
- Построили линейный решатель, который позволит нам получать различные результаты путем подстановки данных в него.

Булевы операции с нейроном

- Давайте попробуем сейчас решить несложную задачку и реализовать различные булевы операции через нейроны, а заодно потренируемся со структурой нейронных сетей.

Булевы операции с нейроном

x	$\bar{x}$
0	1
1	0

- Оператор ***НЕ***
- Задача: подобрать вес и смещение для выполнения логического ***НЕ***

Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

Булевы операции с нейроном

x	$\bar{x}$
0	1
1	0

- Оператор ***НЕ***
- Задача: подобрать вес и смещение для выполнения логического ***НЕ***

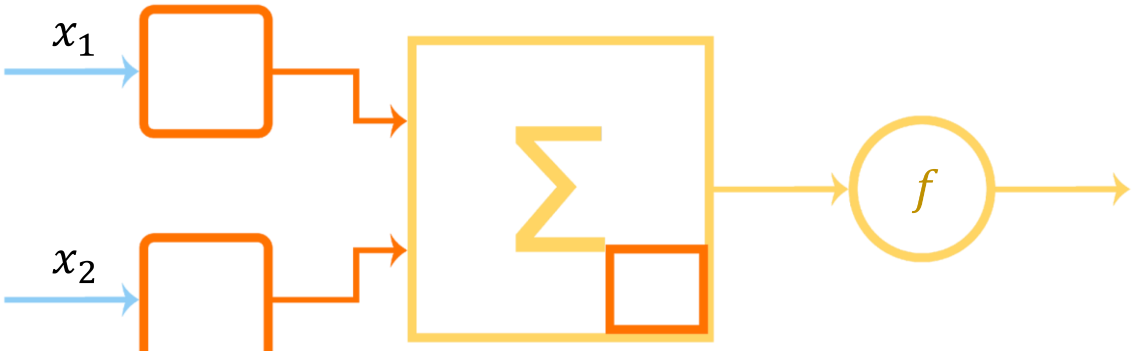
Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

Булевы операции с нейроном

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

- Оператор ***И***
- Задача: подобрать веса и смещение для реализации логического ***И***



Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \le 0 \\ 1, & Z > 0 \end{cases}$$

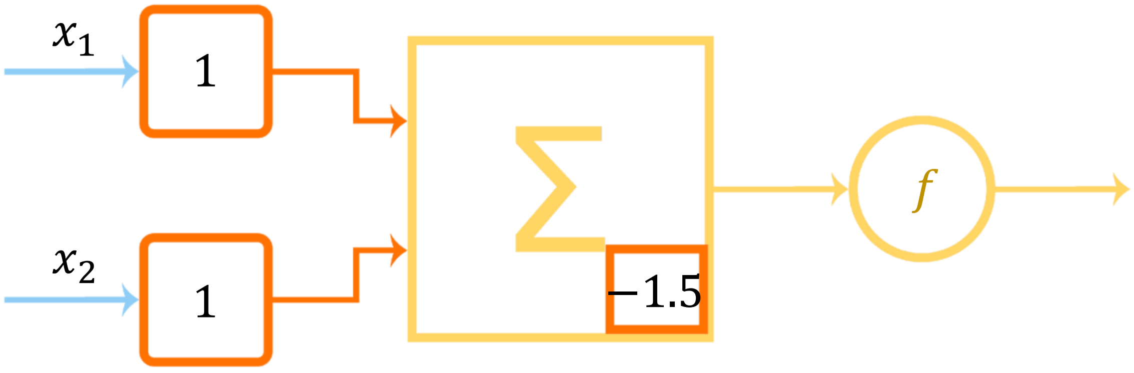
Булевы операции с нейроном

x_1	x_2	$x_1 \land x_2$
0	0	0
0	1	0
1	0	0
1	1	1

- Оператор ***И***
- Задача: подобрать веса и смещение для реализации логического ***И***

Пороговая функция активации:

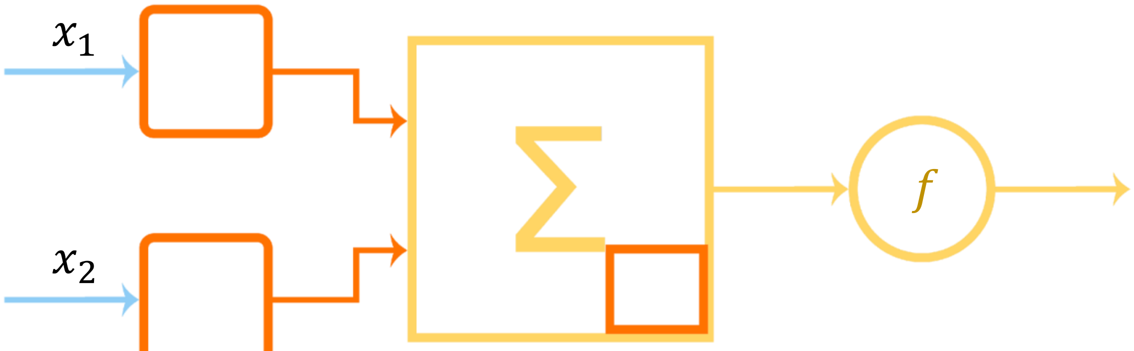
$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$



Булевы операции с нейроном

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

- Оператор ***ИЛИ***
- Задача: подобрать веса и смещение для реализации логического ***ИЛИ***



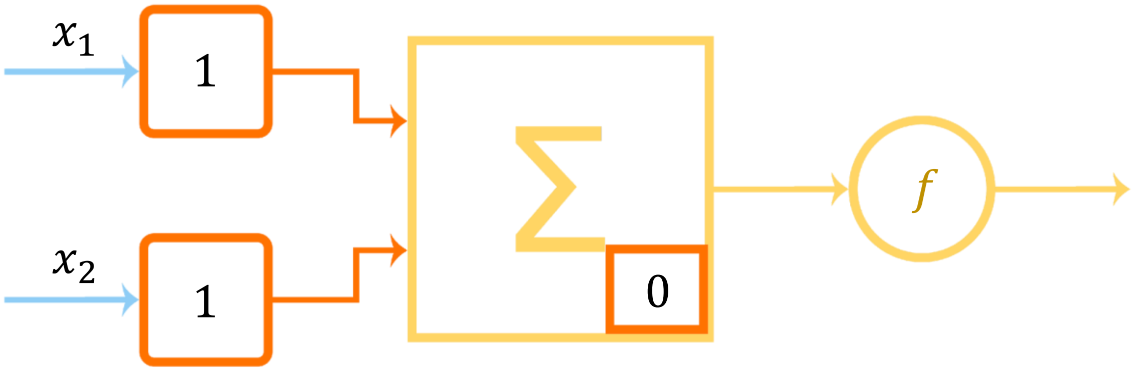
Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

Булевы операции с нейроном

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

- Оператор ***ИЛИ***
- Задача: подобрать веса и смещение для реализации логического ***ИЛИ***



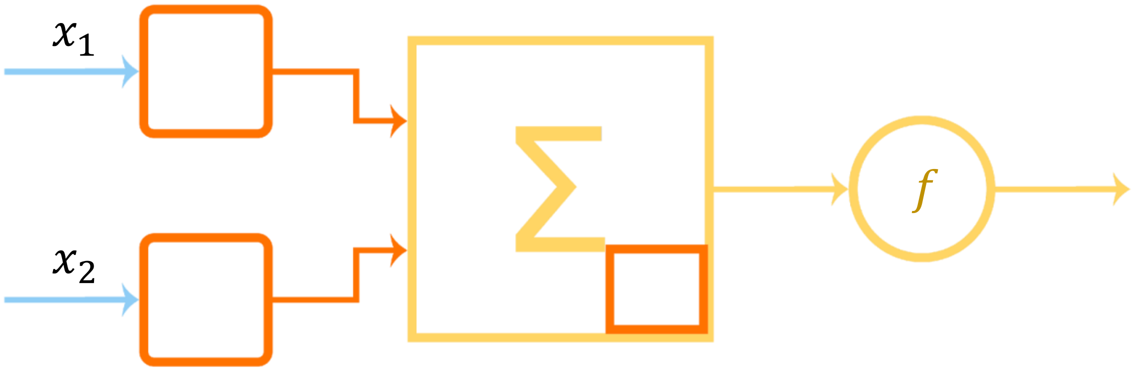
Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

Булевы операции с нейроном

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

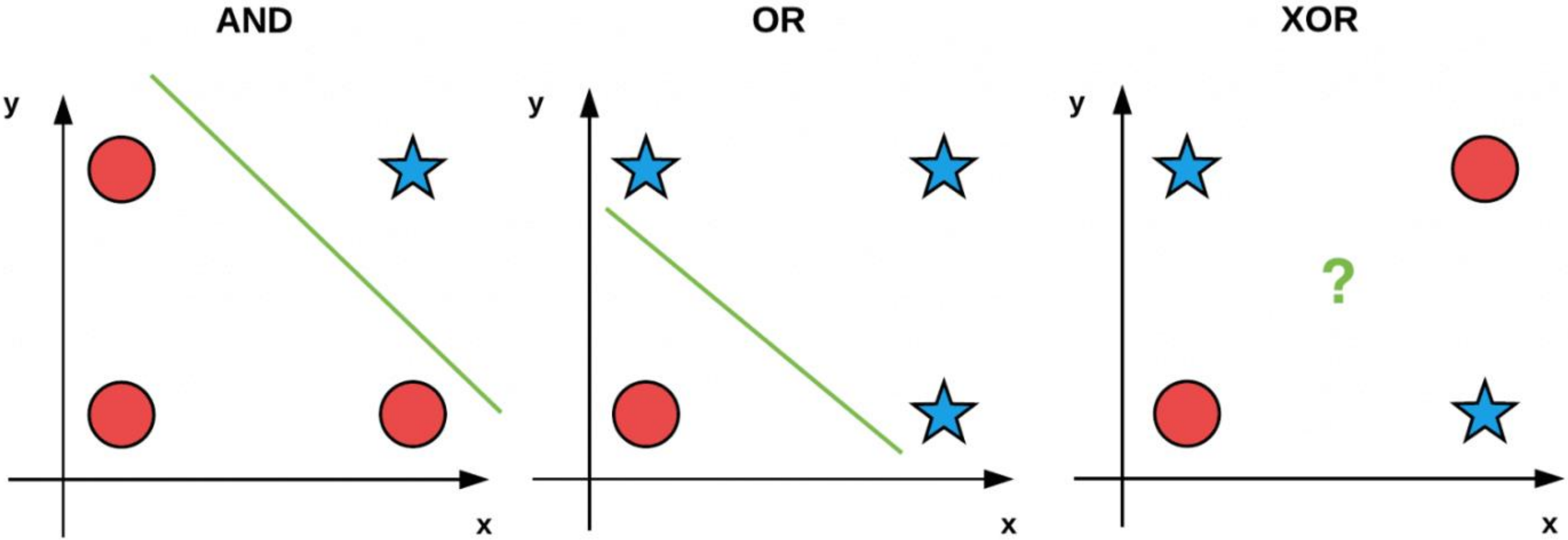
- Оператор ***XOR***
- Задача: подобрать веса и смещение для реализации логического ***XOR***



Пороговая функция активации:

$$f(Z) = \begin{cases} 0, & Z \leq 0 \\ 1, & Z > 0 \end{cases}$$

Булевы операции с нейроном



Разделяющая поверхность

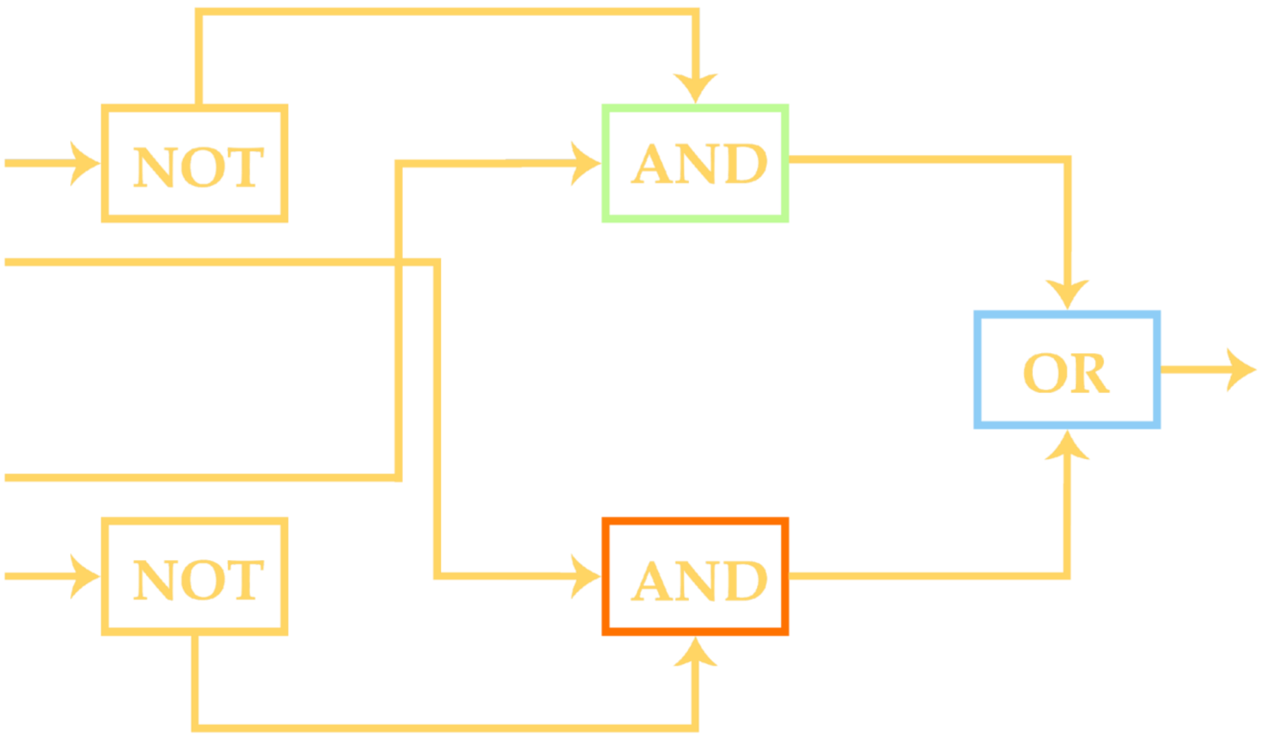
Булевы операции с нейроном

$$x_1 \oplus x_2 = (\overline{x_1} \wedge x_2) \vee (\overline{x_2} \wedge x_1)$$

- Оператор ***XOR***
- Задача: подобрать веса и смещение для реализации логического ***XOR***

Булевы операции с нейроном

$$x_1 \oplus x_2 = (\overline{x_1} \wedge x_2) \vee (\overline{x_2} \wedge x_1)$$



- Оператор ***XOR***
- Задача: подобрать веса и смещение для реализации логического ***XOR***

- С помощью 5 нейронов возможно реализовать нелинейную логическую операцию.
- Однако для этого вполне достаточно и 3 нейронов.

Булевы операции с нейроном

- Математический нейрон как таковой позволяет реализовывать булевы операции.
- Комбинация нейронов способна решить уже нелинейную булеву операцию!

Булевы операции с нейроном

- Математический нейрон как таковой позволяет реализовывать булевы операции.
- Комбинация нейронов способна решить уже нелинейную булеву операцию!
- Попробуем теперь объединить нейроны в несколько слоев и посмотрим, что из этого выйдет!

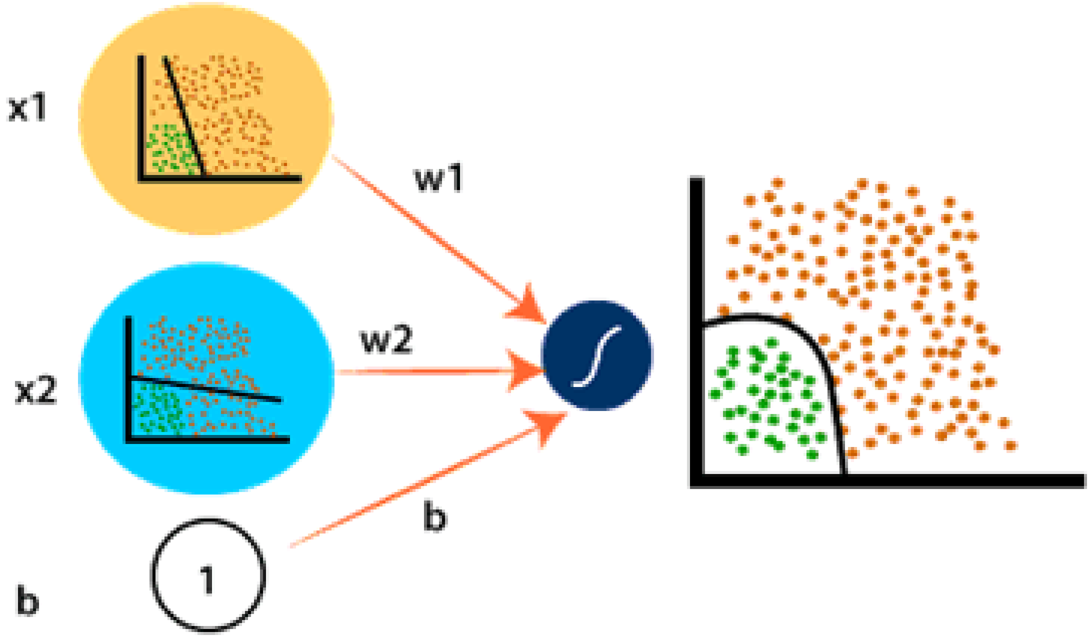
Нейронные сети

Нейронные сети

- Коротко о том, откуда в нейронных сетях появляется нелинейность:

Нейронные сети

- Коротко о том, откуда в нейронных сетях появляется нелинейность:



Нейронные сети

- Нейронная сеть — это структура, состоящая из множества нейронных слоев, где каждый слой включает в себя искусственные нейроны, взаимодействующие между собой посредством весов и функций активации.
- Такие взаимодействия позволяют сети обучаться на данных и решать различные задачи машинного обучения.

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь
  - Как мы хотим её штрафовать за ошибки

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь
  - Как мы хотим её штрафовать за ошибки
- Метод оптимизации

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь
  - Как мы хотим её штрафовать за ошибки
- Метод оптимизации
  - Как мы хотим обновлять веса

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь
  - Как мы хотим её штрафовать за ошибки
- Метод оптимизации
  - Как мы хотим обновлять веса
- Метрика

Нейронные сети

- Какие основные компоненты нейронной сети могут быть выделены?
- Архитектура нейронной сети
  - Из каких блоков она состоит
- Функция потерь
  - Как мы хотим её штрафовать за ошибки
- Метод оптимизации
  - Как мы хотим обновлять веса
- Метрика
  - Как будем оценивать качество работы

Нейронные сети

- Какие основные типы задач в глубинном обучении могут быть выделены?

Нейронные сети

- Какие основные типы задач в глубинном обучении могут быть выделены?
- Регрессия
- Классификация
- Компьютерное зрение (CV) – детекция, сегментация и т.д.
- Обработка естественного языка (NLP) – перевод, генерация и т.д.
- Metric learning
- Кластеризация
- Поиск аномалий

Функция потерь нейронной сети

- Напомните, что такое функция потерь?

Функция потерь нейронной сети

- Напомните, что такое функция потерь?
- Необходимо придумать некое правило для оценки качества предсказания во время обучения — функцию потерь. Функция потерь должна нам показывать, насколько хорошо мы сейчас решаем нашу задачу.

Функция потерь нейронной сети

- Напомните, что такое функция потерь?
- Необходимо придумать некое правило для оценки качества предсказания во время обучения — функцию потерь. Функция потерь должна нам показывать, насколько хорошо мы сейчас решаем нашу задачу.
- В качестве примера для регрессии можно вспомнить популярную и знакомую нам функцию потерь MSE — средний квадрат ошибки.

$$\frac{1}{n}\sum_{i}^{n}(\widetilde{y}_i - y_i)^2$$

Функция потерь нейронной сети

- А какая функция потерь используется в задачах классификации?

Функция потерь нейронной сети

- А какая функция потерь используется в задачах классификации?
- Для решения задачи классификации используют функцию потерь кросс-энтропия (в случае многоклассовой классификации) и бинарную кросс-энтропию для бинарной классификации.

Функция потерь нейронной сети

$$\underbrace{-\sum_{j=1}^{M} \overbrace{y_j}^{\text{Indicator variable}} log(\overbrace{p(y_j)}^{\text{Prob of class } j})}_{\text{Sum over classes}}$$

$$\overbrace{-\sum_{i=1}^{N}}^{\text{Sum over trials}} \overbrace{\underbrace{y_i}_{\text{Label}} log(\underbrace{p(y_i)}_{\text{Prob of positive class}}) + (1 - \underbrace{y_i}_{\text{Label}}) log(1 - \underbrace{p(y_i)}_{\text{Prob of positive class}})}^{\text{Sum over classes}}$$

Нейронные сети

- А вот теперь очень важное утверждение.
- Теорема Цыбенко или Универсальная теорема аппроксимации, теорема, доказанная Джорджем Цыбенко в 1989 году.

Нейронные сети

- А вот теперь очень важное утверждение.
- Теорема Цыбенко или Универсальная теорема аппроксимации, теорема, доказанная Джорджем Цыбенко в 1989 году.
- Искусственная нейронная сеть прямой связи с одним скрытым слоем и нелинейной функцией активации (например, сигмоида) может аппроксимировать…

Нейронные сети

- А вот теперь очень важное утверждение.
- Теорема Цыбенко или Универсальная теорема аппроксимации, теорема, доказанная Джорджем Цыбенко в 1989 году.
- Искусственная нейронная сеть прямой связи с одним скрытым слоем и нелинейной функцией активации (например, сигмоида) может аппроксимировать…

…любую ограниченную функцию многих переменных
с не более чем счетным числом точек разрыва
с любой точностью.

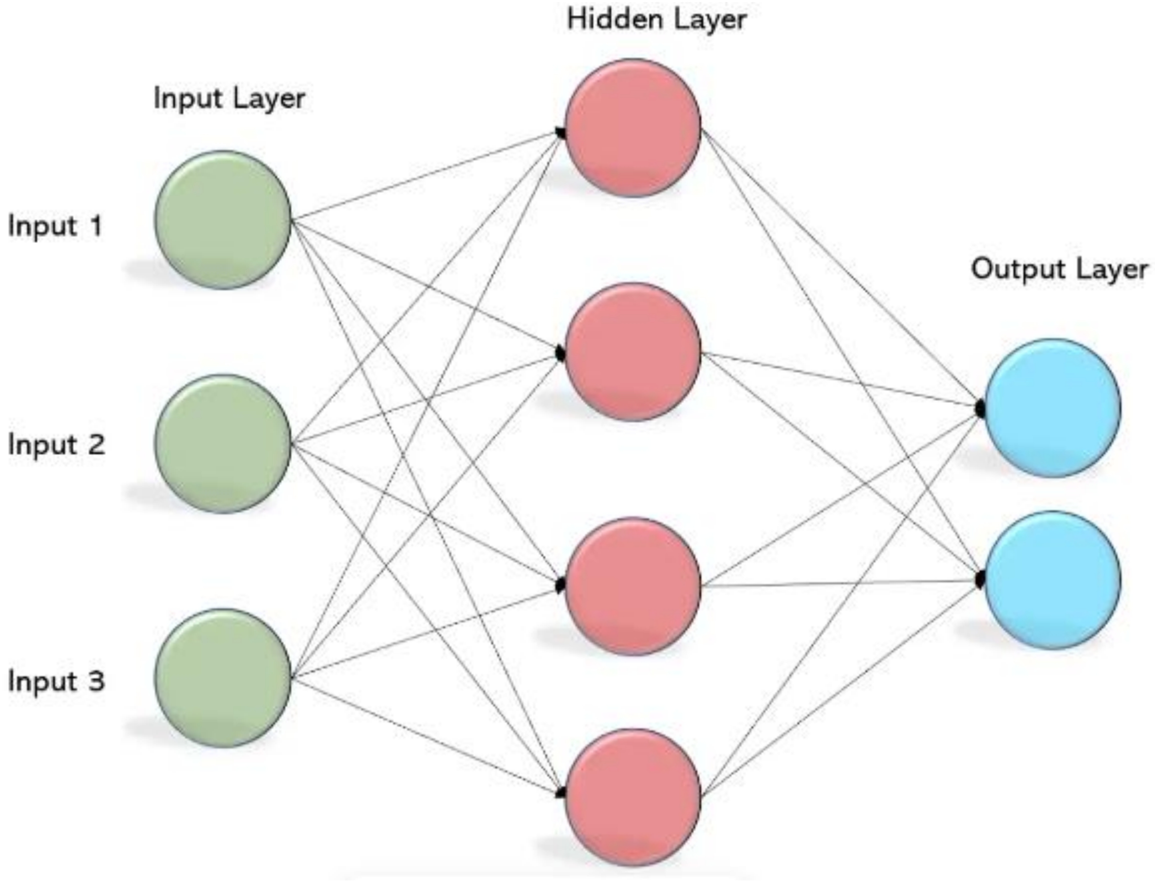
Нейронные сети

- А вот теперь очень важное утверждение.
- Теорема Цыбенко или Универсальная теорема аппроксимации, теорема, доказанная Джорджем Цыбенко в 1989 году.
- Искусственная нейронная сеть прямой связи с одним скрытым слоем и нелинейной функцией активации (например, сигмоида) может аппроксимировать…

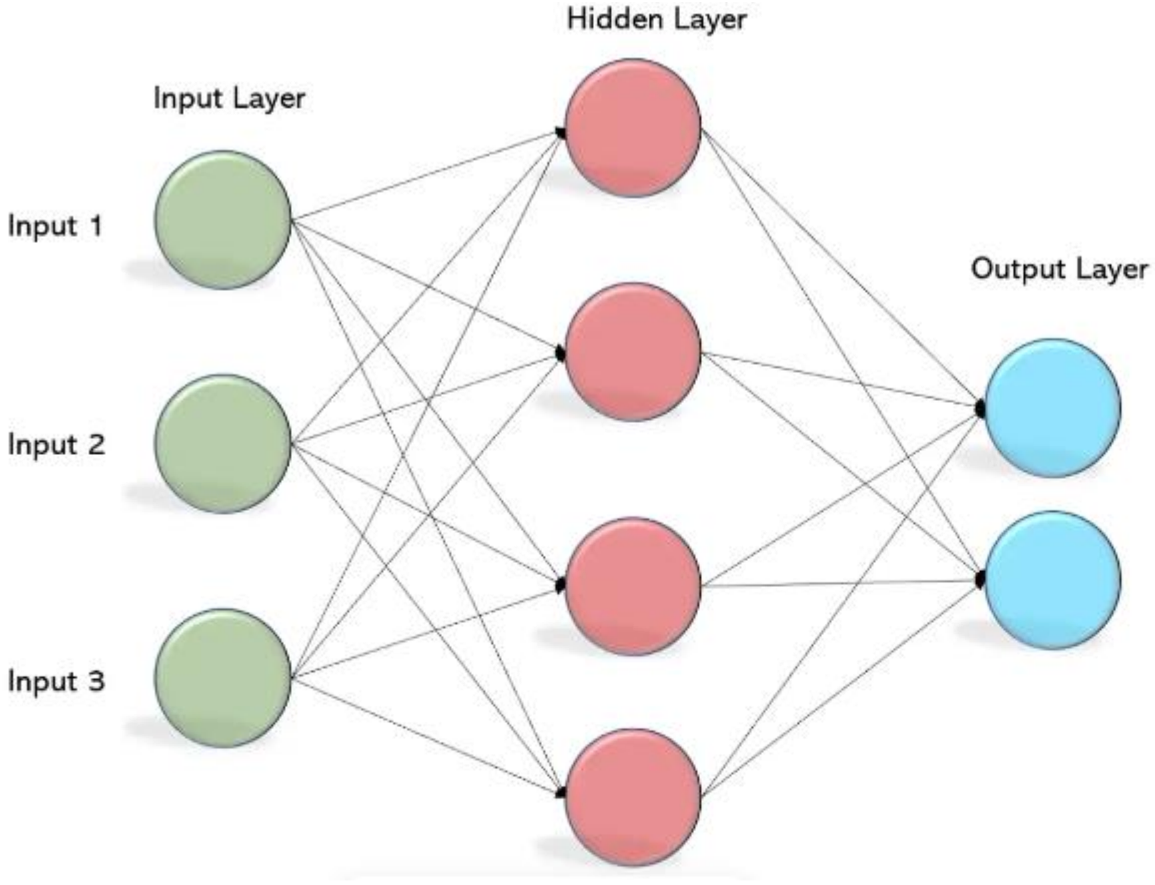
…любую ограниченную функцию многих переменных
с не более чем счетным числом точек разрыва
с любой точностью.

Условием только является достаточное количество нейронов скрытого слоя.

Нейронные сети

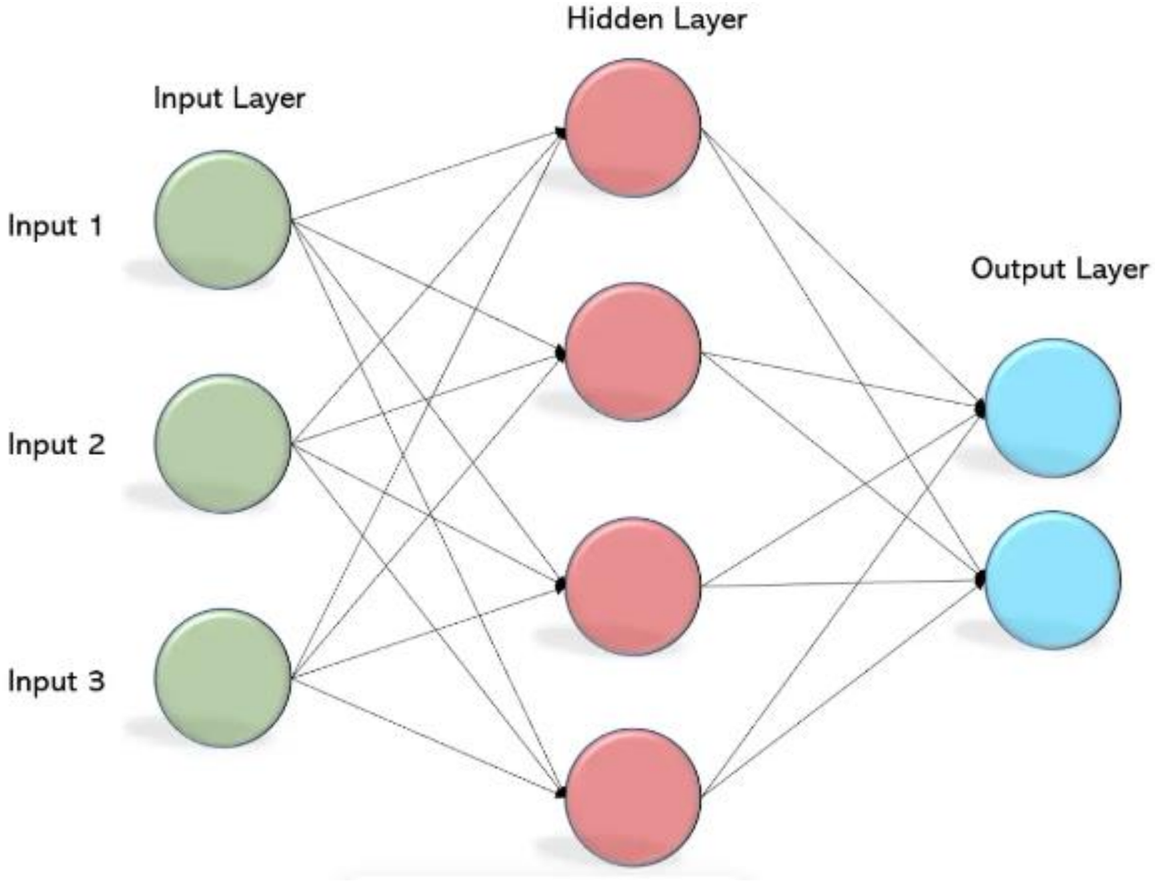


Нейронные сети



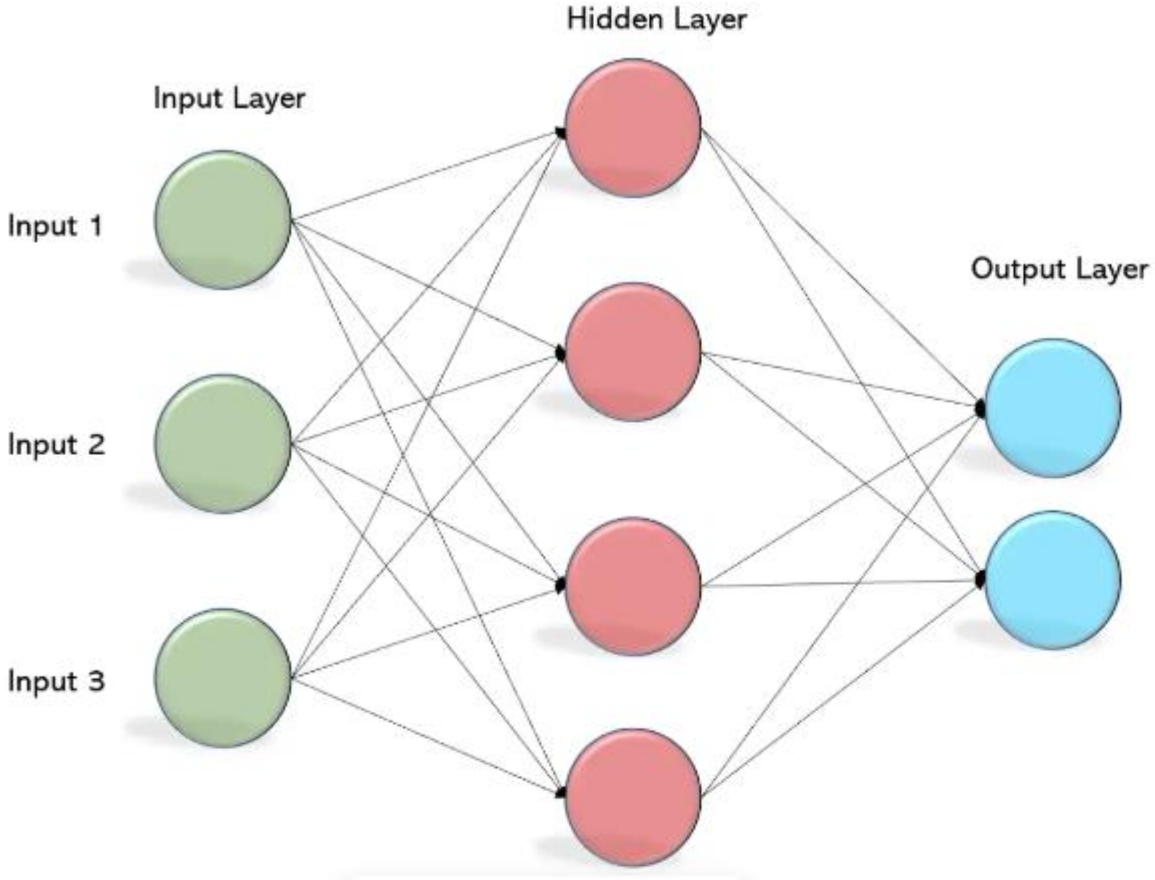
- Перцептрон Розенблатта

Нейронные сети



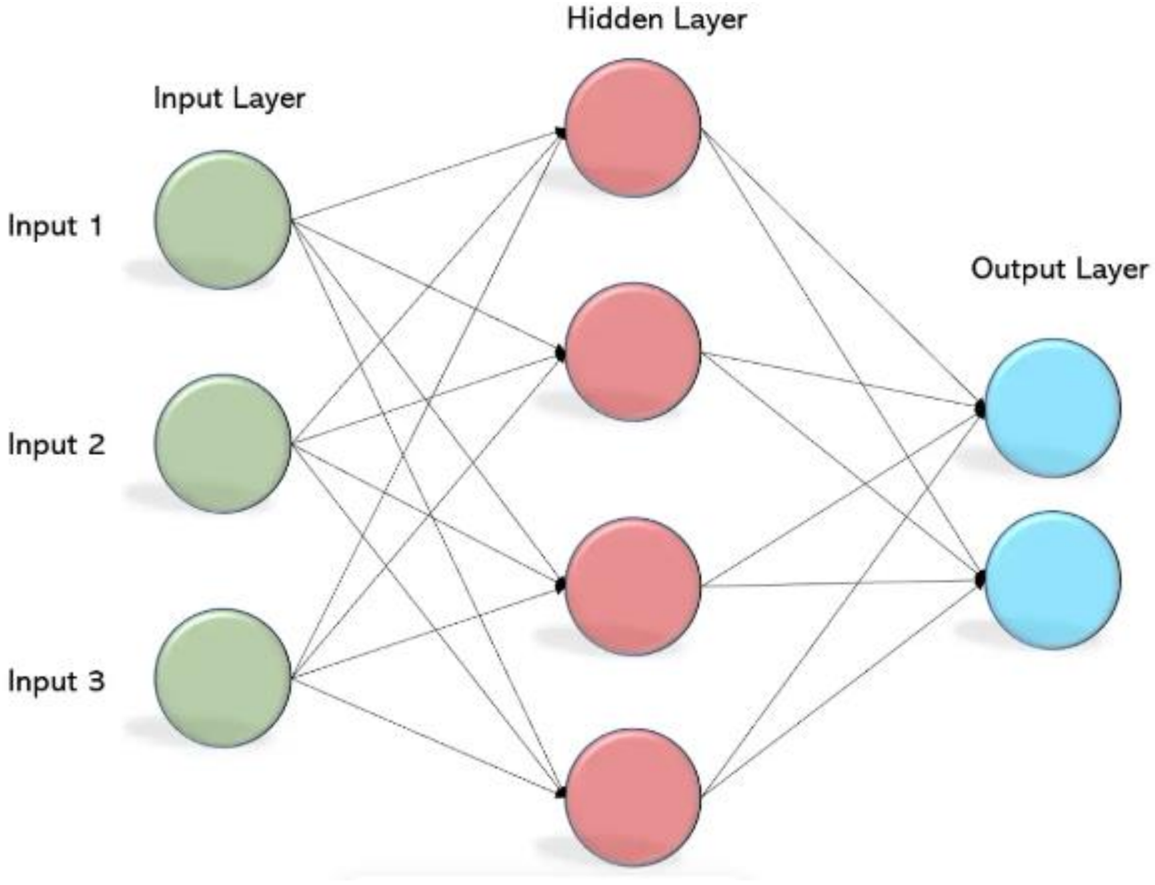
- Перцептрон Розенблатта
- Нейронная сеть с 1 скрытым слоем

Нейронные сети



- Перцептрон Розенблатта
- Нейронная сеть с 1 скрытым слоем
- Пороговая функция активации

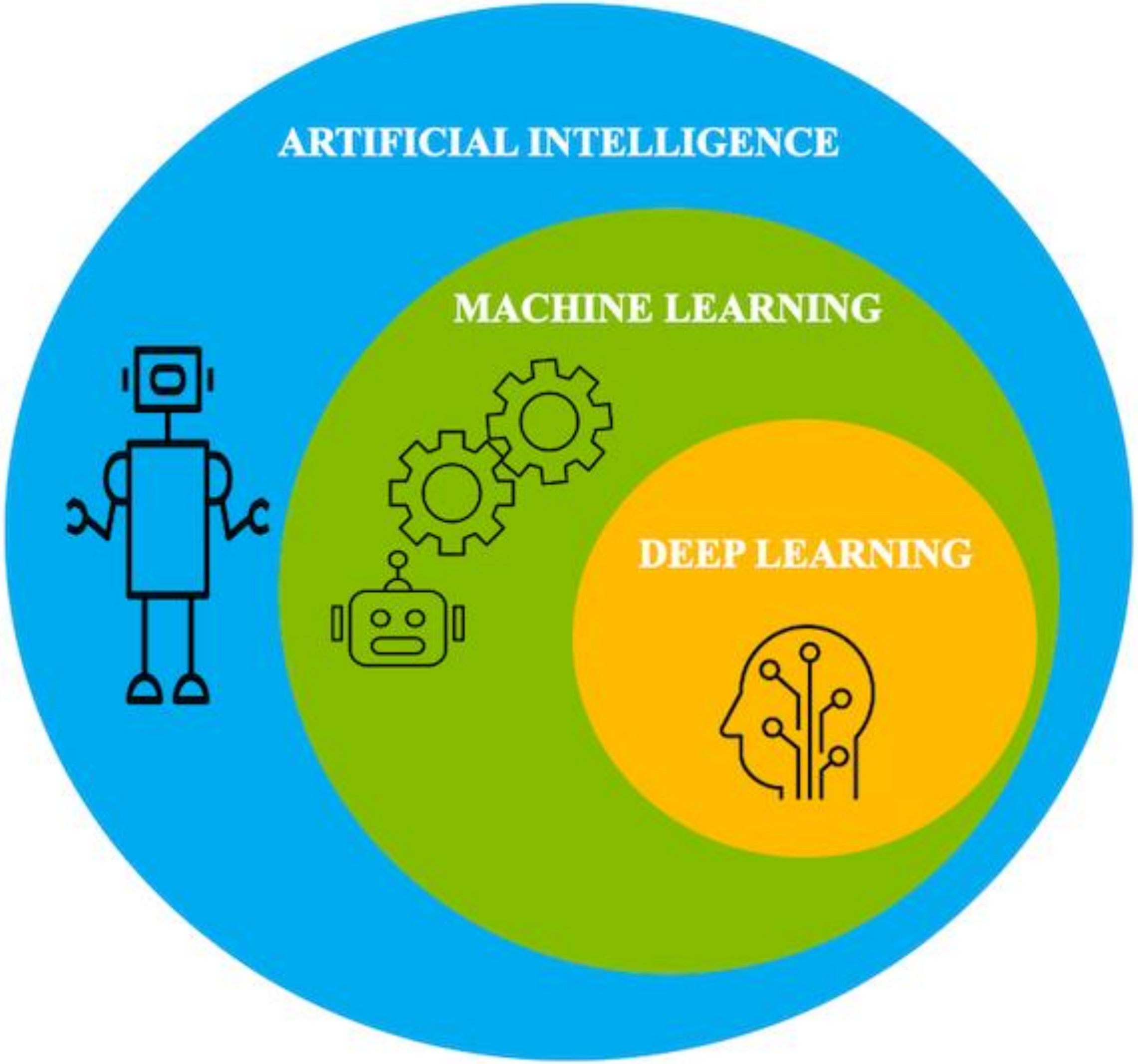
Нейронные сети



- Перцептрон Розенблатта
- Нейронная сеть с 1 скрытым слоем
- Пороговая функция активации
- Первая успешно работающая нейронная сеть (решала задачу распознавания английских букв)

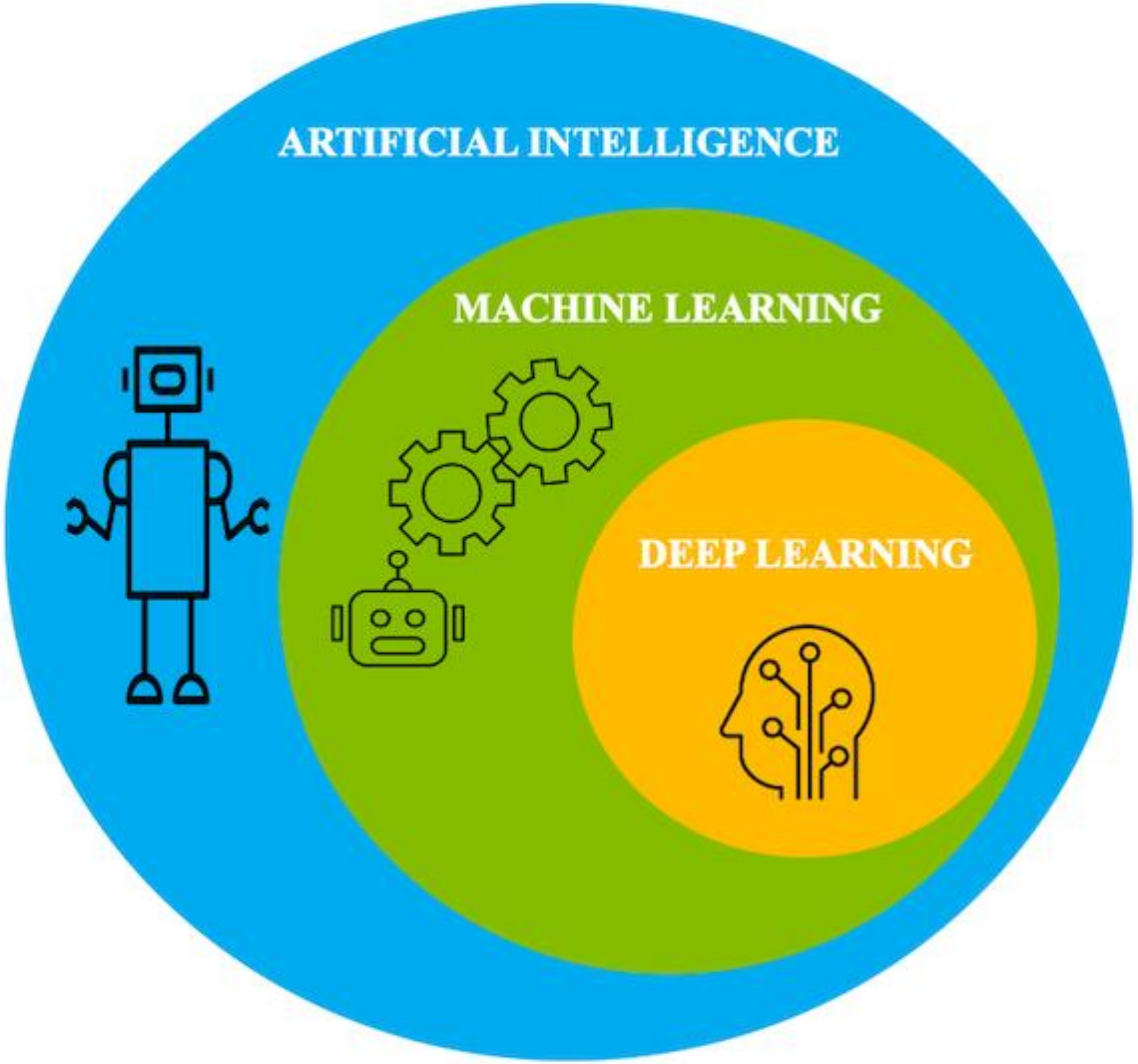
Глубинное обучение

- Под глубоким (или глубинным) обучением понимают одну из областей машинного обучения, связанную с большими нейронными сетями



Глубинное обучение

- Под глубоким (или глубинным) обучением понимают одну из областей машинного обучения, связанную с большими нейронными сетями
- Формально, сеть считается глубокой, если в ней 3 или более скрытых слоя



Математическая оптимизация нейронных сетей

Математическая оптимизация нейронных сетей

- Давайте вспомним очень важное понятие в контексте машинного обучения и математической оптимизации — градиент.

Математическая оптимизация нейронных сетей

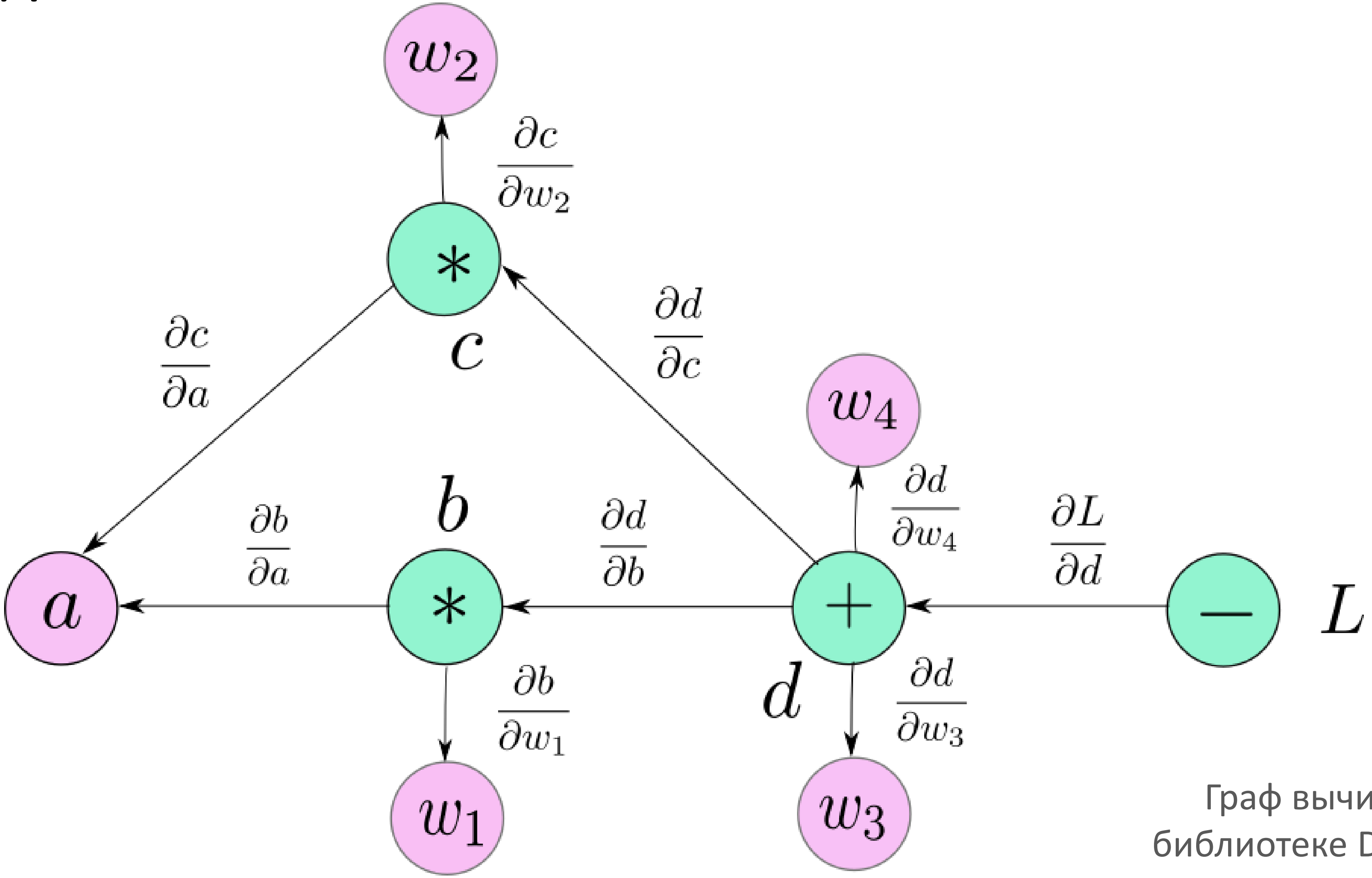
- Давайте вспомним очень важное понятие в контексте машинного обучения и математической оптимизации — градиент.
- Градиентом функции многих переменных называется вектор, составленный из всех частных производных.
- Обратите внимание: это именно вектор, не число!

Математическая оптимизация нейронных сетей

- Давайте вспомним очень важное понятие в контексте машинного обучения и математической оптимизации — градиент.
- Градиентом функции многих переменных называется вектор, составленный из всех частных производных.
- Обратите внимание: это именно вектор, не число!

$$\nabla F = \left(\frac{\partial F}{\partial x_1}(x_1, x_2, \dots, x_n), \frac{\partial F}{\partial x_2}(x_1, x_2, \dots, x_n), \dots, \frac{\partial F}{\partial x_n}(x_1, x_2, \dots, x_n) \right)$$

Математическая оптимизация нейронных сетей



Граф вычислений в библиотеке DL — PyTorch

Математическая оптимизация нейронных сетей

- Chain rule или правило цепочки — правило дифференцирования сложной функции

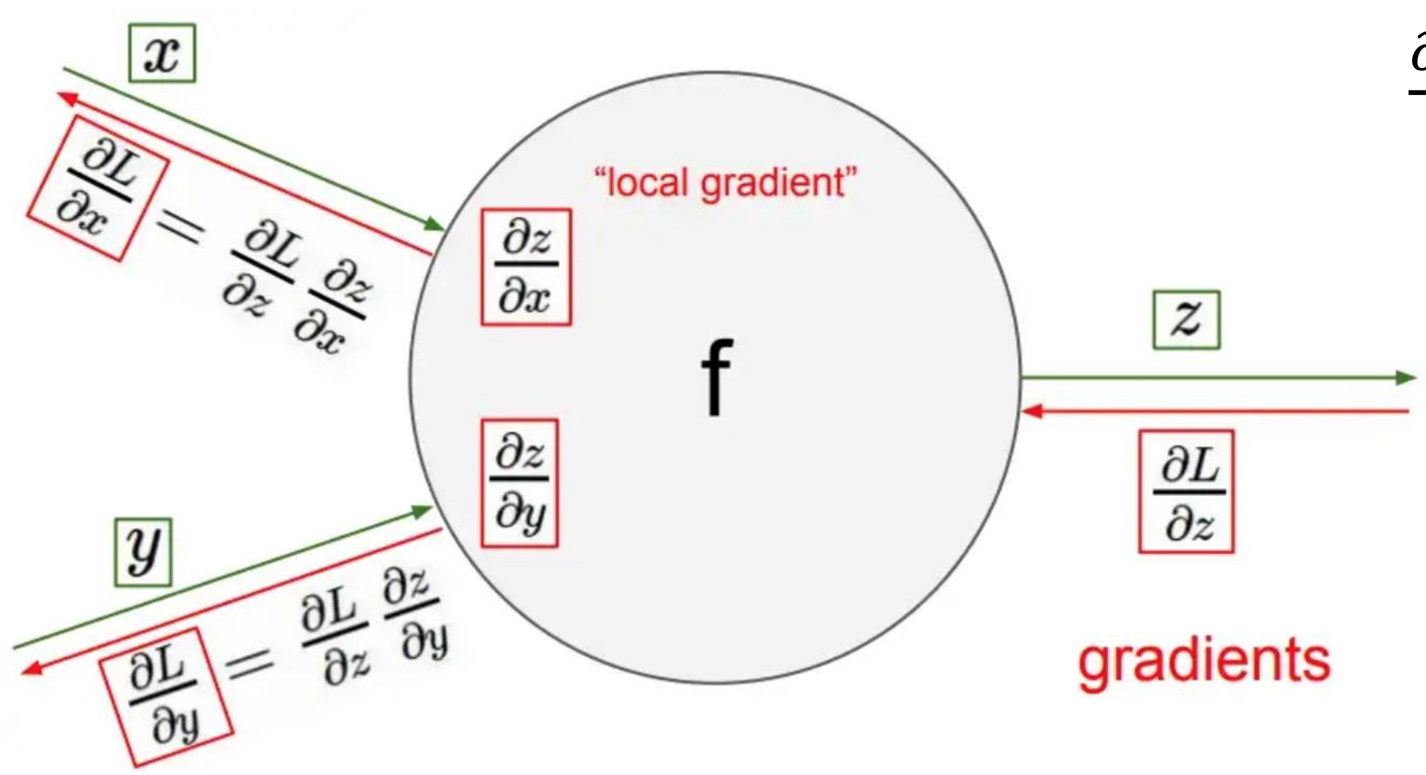
Математическая оптимизация нейронных сетей

- Chain rule или правило цепочки — правило дифференцирования сложной функции

$$\frac{\partial f(g(x), e(x))}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x} + \frac{\partial f}{\partial e} \cdot \frac{\partial e}{\partial x}$$

Математическая оптимизация нейронных сетей

- Chain rule или правило цепочки — правило дифференцирования сложной функции

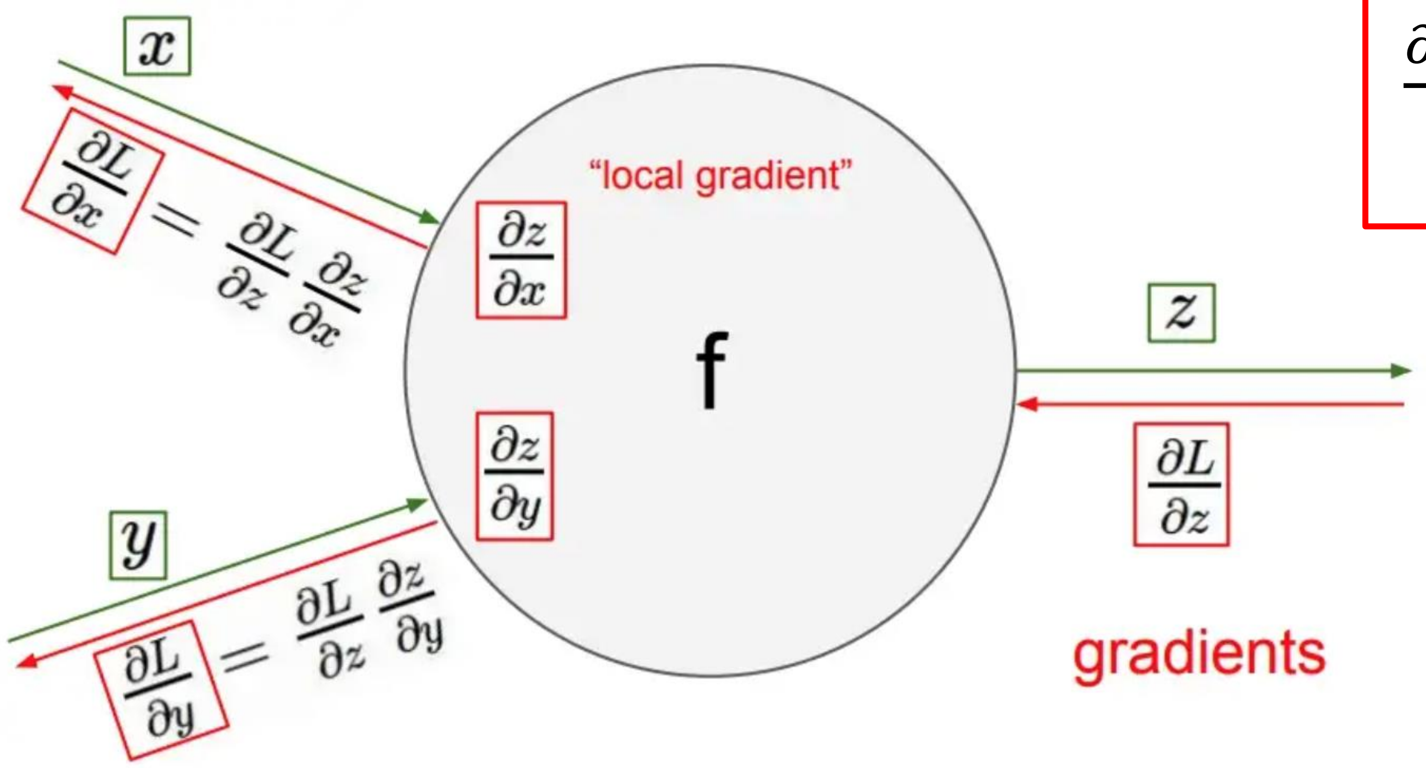


$$\frac{\partial f(g(x),e(x))}{\partial x} = \frac{\partial f}{\partial g}\cdot\frac{\partial g}{\partial x} + \frac{\partial f}{\partial e}\cdot\frac{\partial e}{\partial x}$$

Математическая оптимизация нейронных сетей

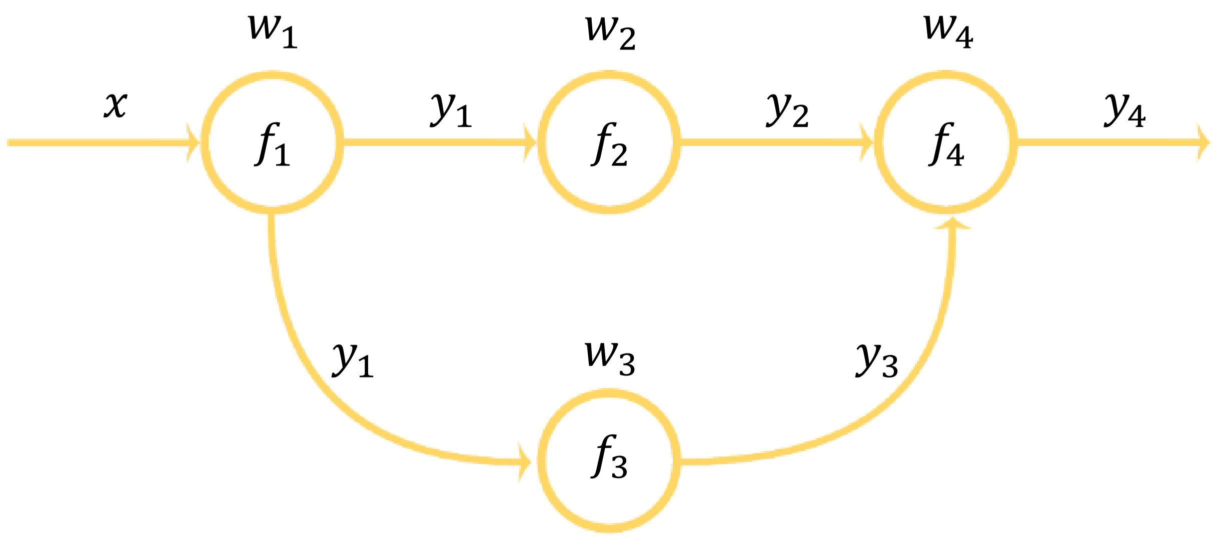
- Chain rule или правило цепочки — правило дифференцирования сложной функции

$$\boxed{\frac{\partial f(g(x),e(x))}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x} + \frac{\partial f}{\partial e} \cdot \frac{\partial e}{\partial x}}$$



Математическая оптимизация нейронных сетей

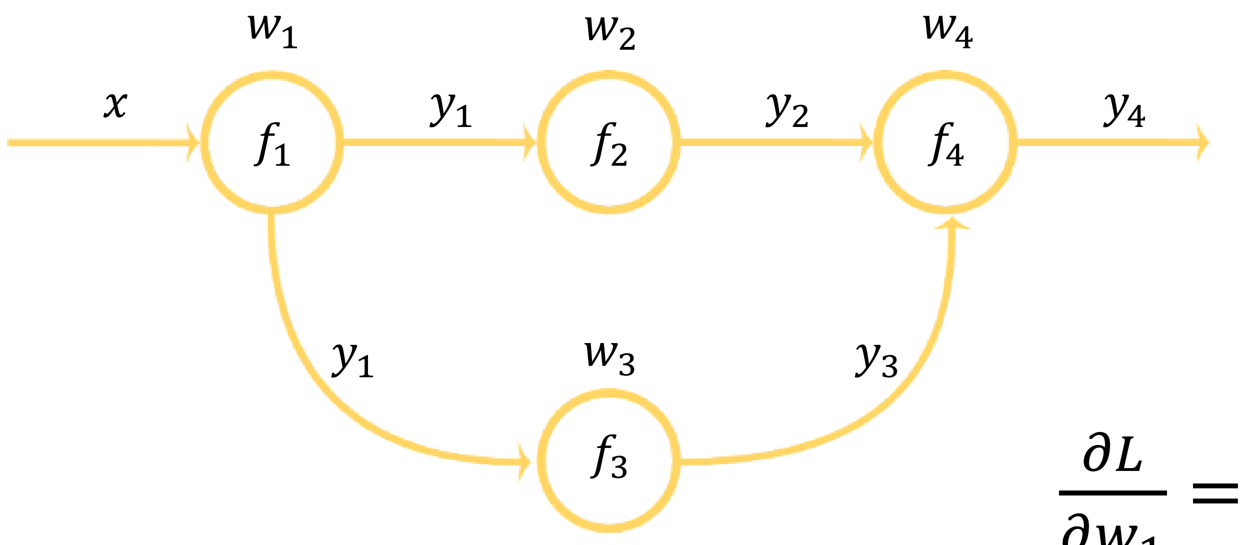
- Chain rule или правило цепочки — правило дифференцирования сложной функции



$$\boxed{\frac{\partial f(g(x),e(x))}{\partial x} = \frac{\partial f}{\partial g}\cdot\frac{\partial g}{\partial x} + \frac{\partial f}{\partial e}\cdot\frac{\partial e}{\partial x}}$$

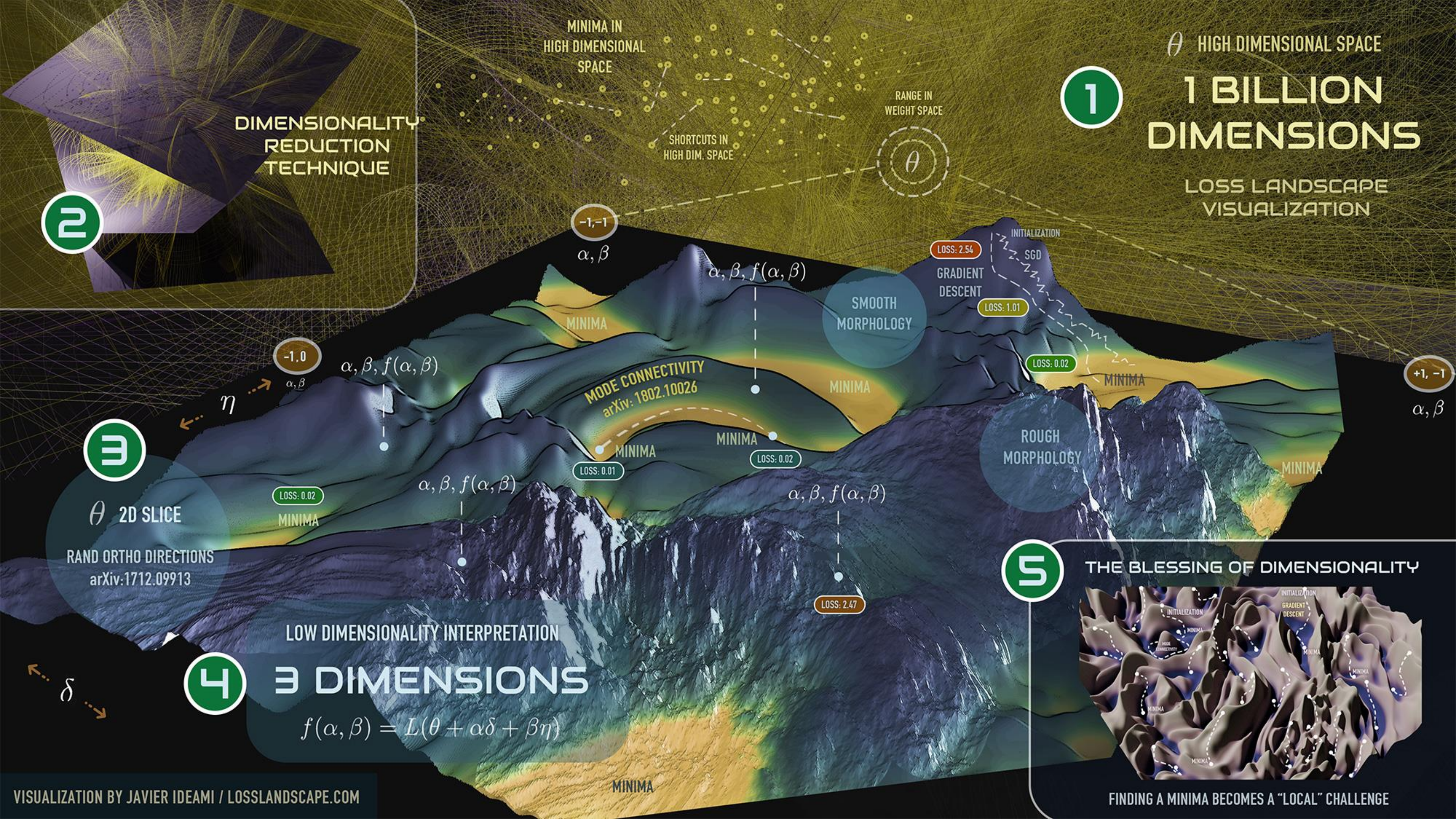
Математическая оптимизация нейронных сетей

- Chain rule или правило цепочки — правило дифференцирования сложной функции



$$\boxed{\frac{\partial f(g(x),e(x))}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x} + \frac{\partial f}{\partial e} \cdot \frac{\partial e}{\partial x}}$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}\left(\frac{\partial f_4}{\partial y_2} \cdot \frac{\partial f_2}{\partial y_1} + \frac{\partial f_4}{\partial y_3} \cdot \frac{\partial f_3}{\partial y_1}\right)\frac{\partial f_1}{\partial w_1}$$



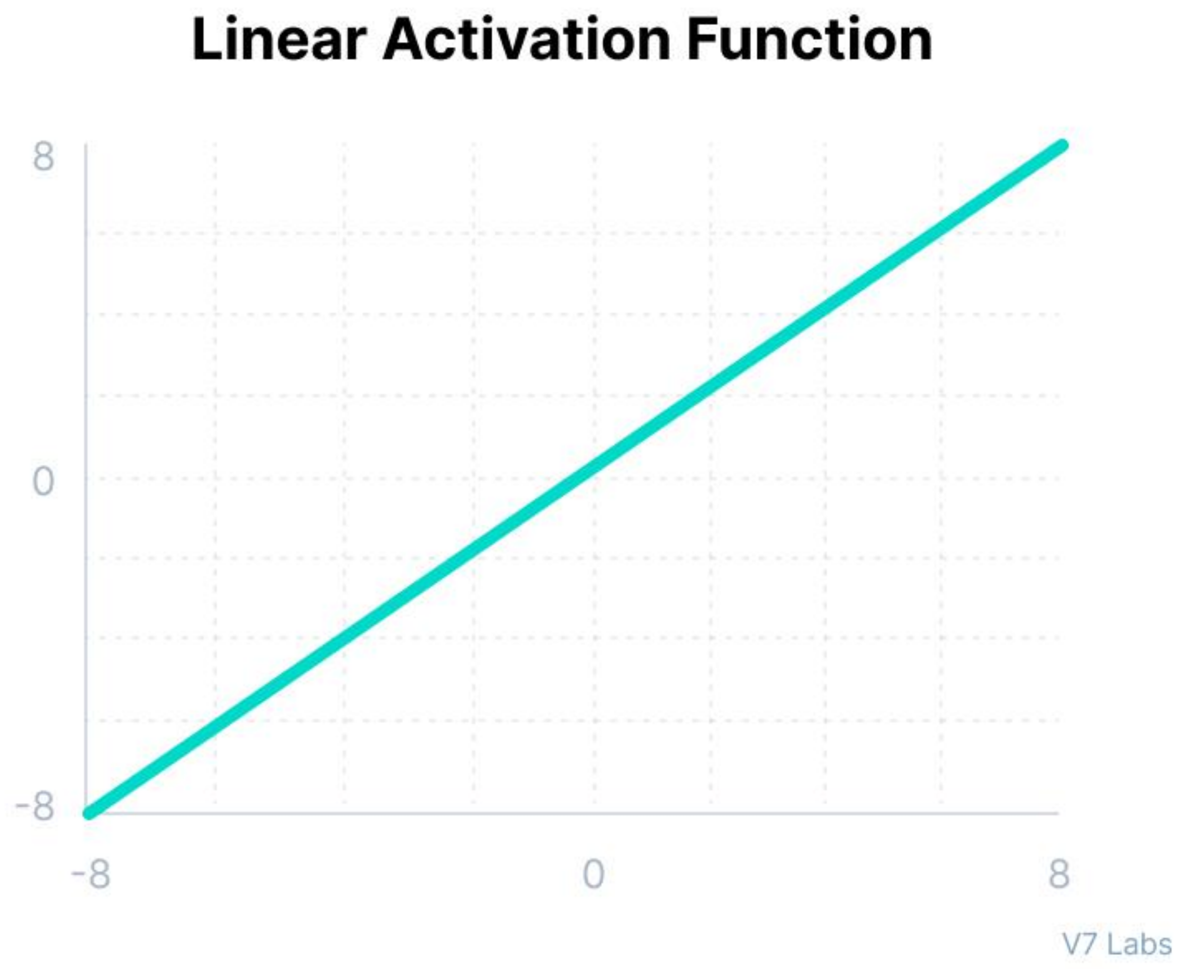
MINIMA IN HIGH DIMENSIONAL SPACE
SHORTCUTS IN HIGH DIM. SPACE
RANGE IN WEIGHT SPACE
θ HIGH DIMENSIONAL SPACE
1
1 BILLION DIMENSIONS
LOSS LANDSCAPE VISUALIZATION
2
DIMENSIONALITY REDUCTION TECHNIQUE
-1,-1
α, β
$\alpha, \beta, f(\alpha, \beta)$
INITIALIZATION
SGD
LOSS: 2.54
GRADIENT DESCENT
LOSS: 1.01
SMOOTH MORPHOLOGY
MINIMA
-1,0
η
MODE CONNECTIVITY
arXiv: 1802.10026
LOSS: 0.02
+1, −1
ROUGH MORPHOLOGY
LOSS: 0.01
3
θ 2D SLICE
RAND ORTHO DIRECTIONS
arXiv:1712.09913
LOSS: 2.47
δ
LOW DIMENSIONALITY INTERPRETATION
4
3 DIMENSIONS
$f(\alpha, \beta) = L(\theta + \alpha\delta + \beta\eta)$
5
THE BLESSING OF DIMENSIONALITY
FINDING A MINIMA BECOMES A "LOCAL" CHALLENGE
VISUALIZATION BY JAVIER IDEAMI / LOSSLANDSCAPE.COM

Математическая оптимизация нейронных сетей

- В реальной жизни ландшафт функции потерь чрезвычайно сложен.
- На картинке раннее представлен примерный ландшафт для модели с 1 млрд параметров (далеко не самая большая модель).
- Как видно, застрять можно где угодно, а значит нужно как-то модифицировать градиентный спуск.

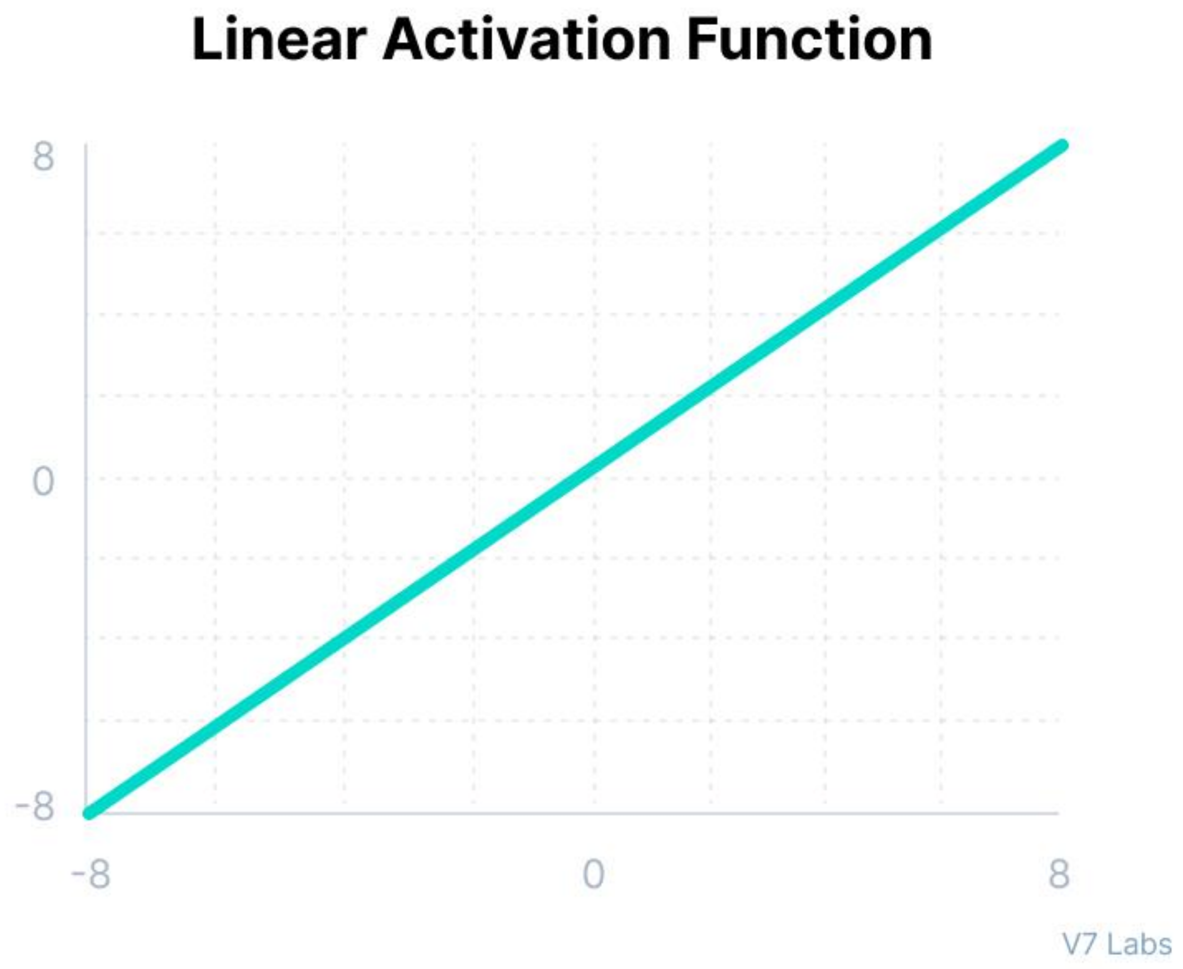
Функции активации

Функции активации

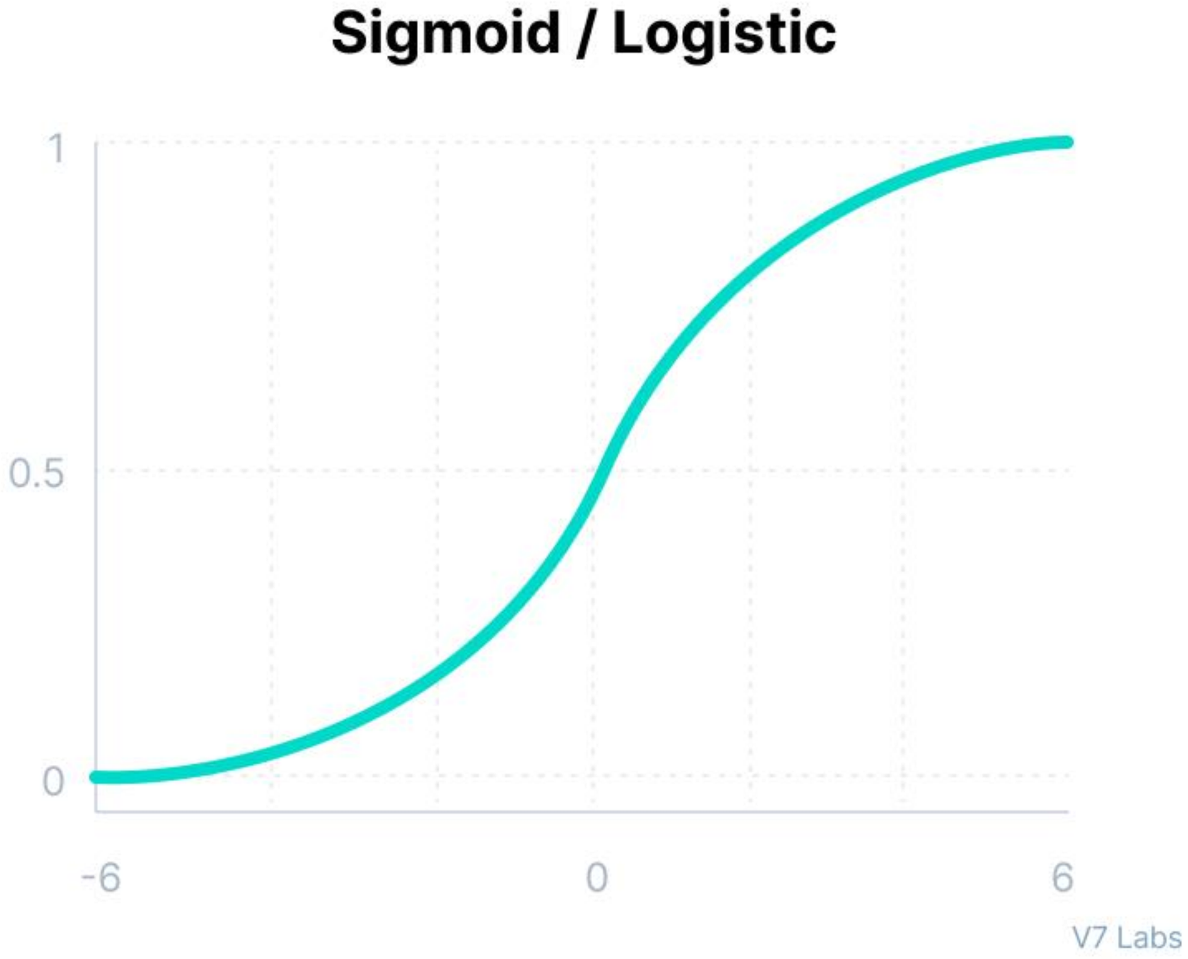


Функции активации

- Недостатки:
- Невозможно использовать обратное распространение ошибки, так как производная функции является константой
- Независимо от количества слоев в нейронной сети последний слой все равно будет линейной функцией первого слоя.

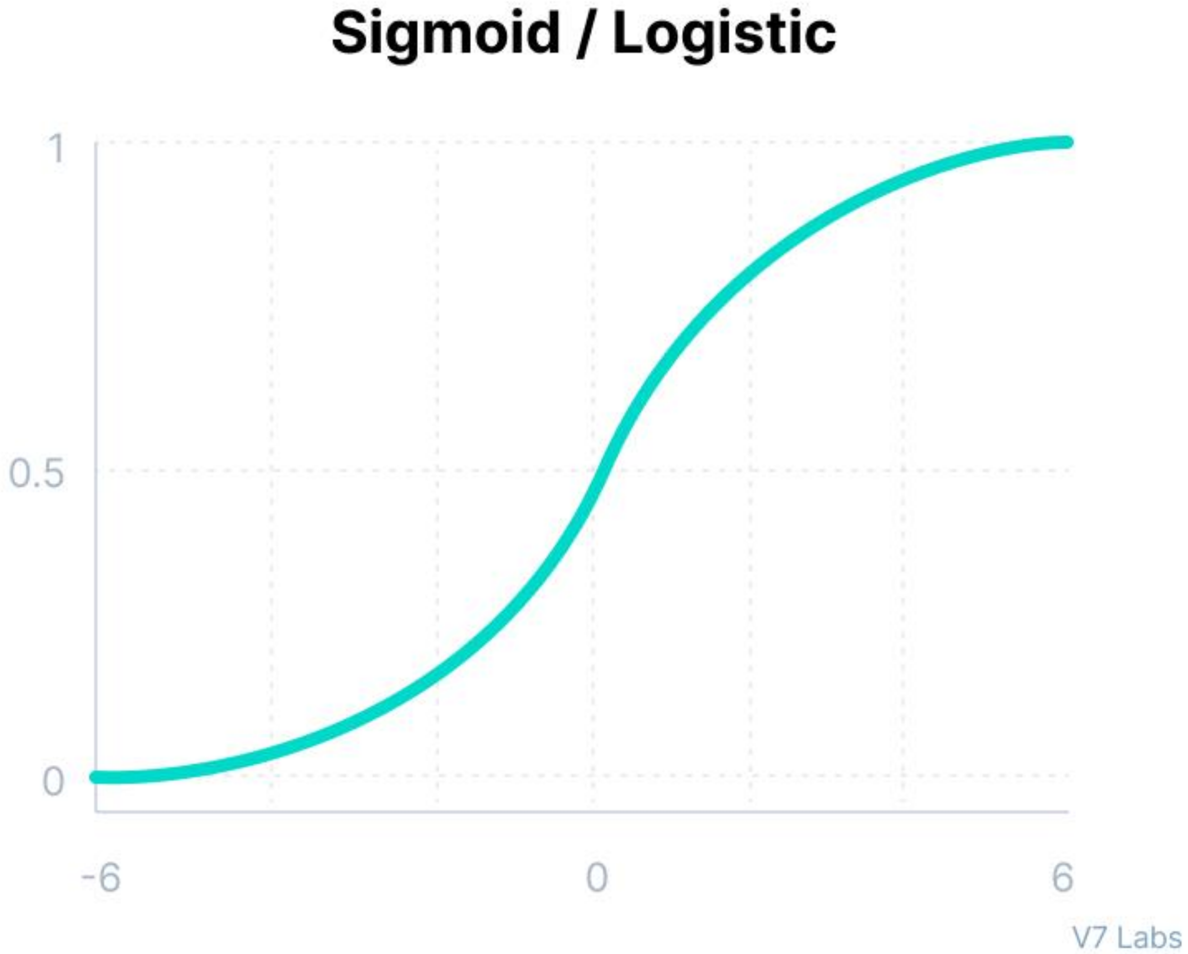


Функции активации

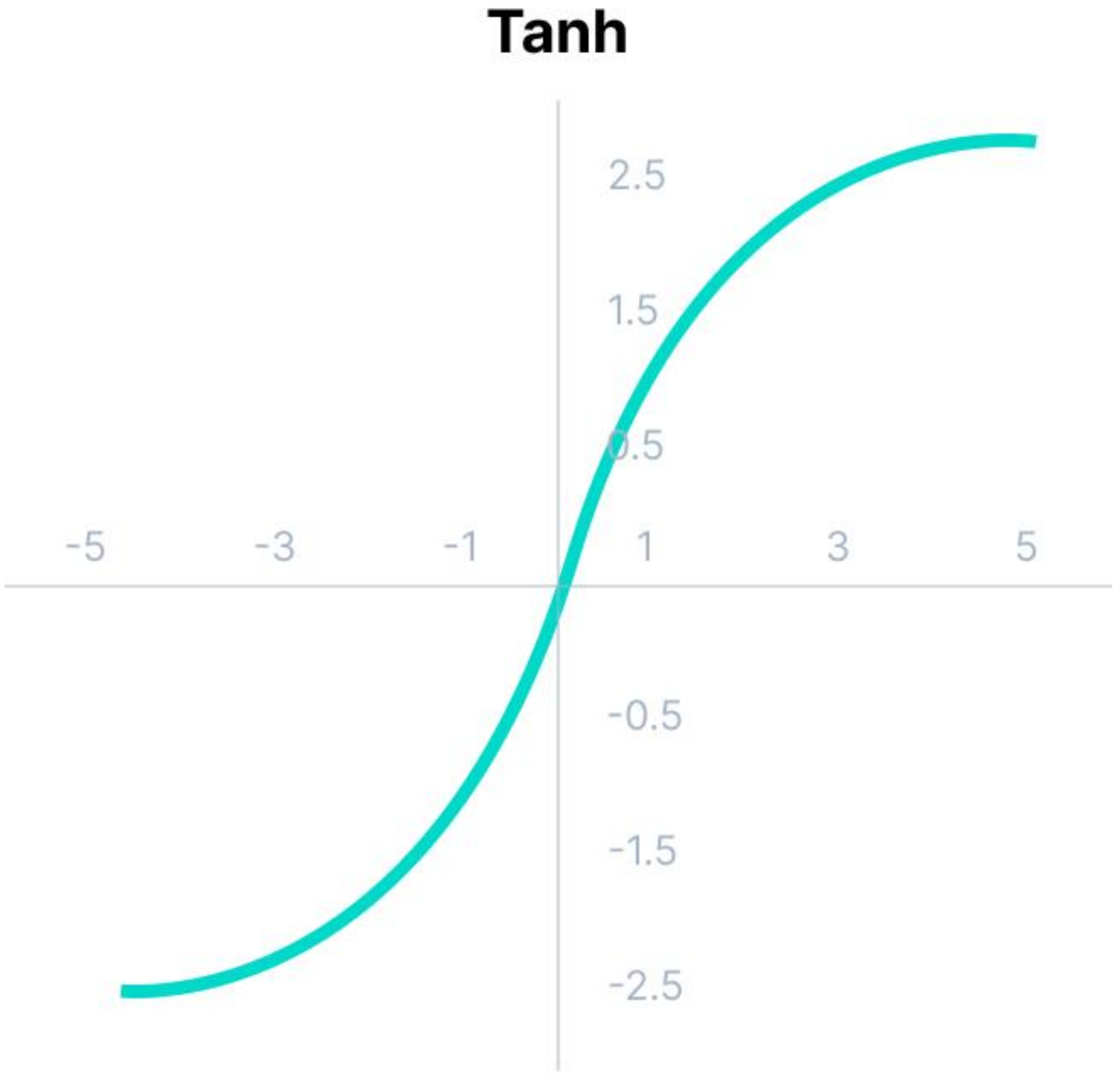


Функции активации

- Недостатки:
- Затухание градиентов
- Все выходы одного знака

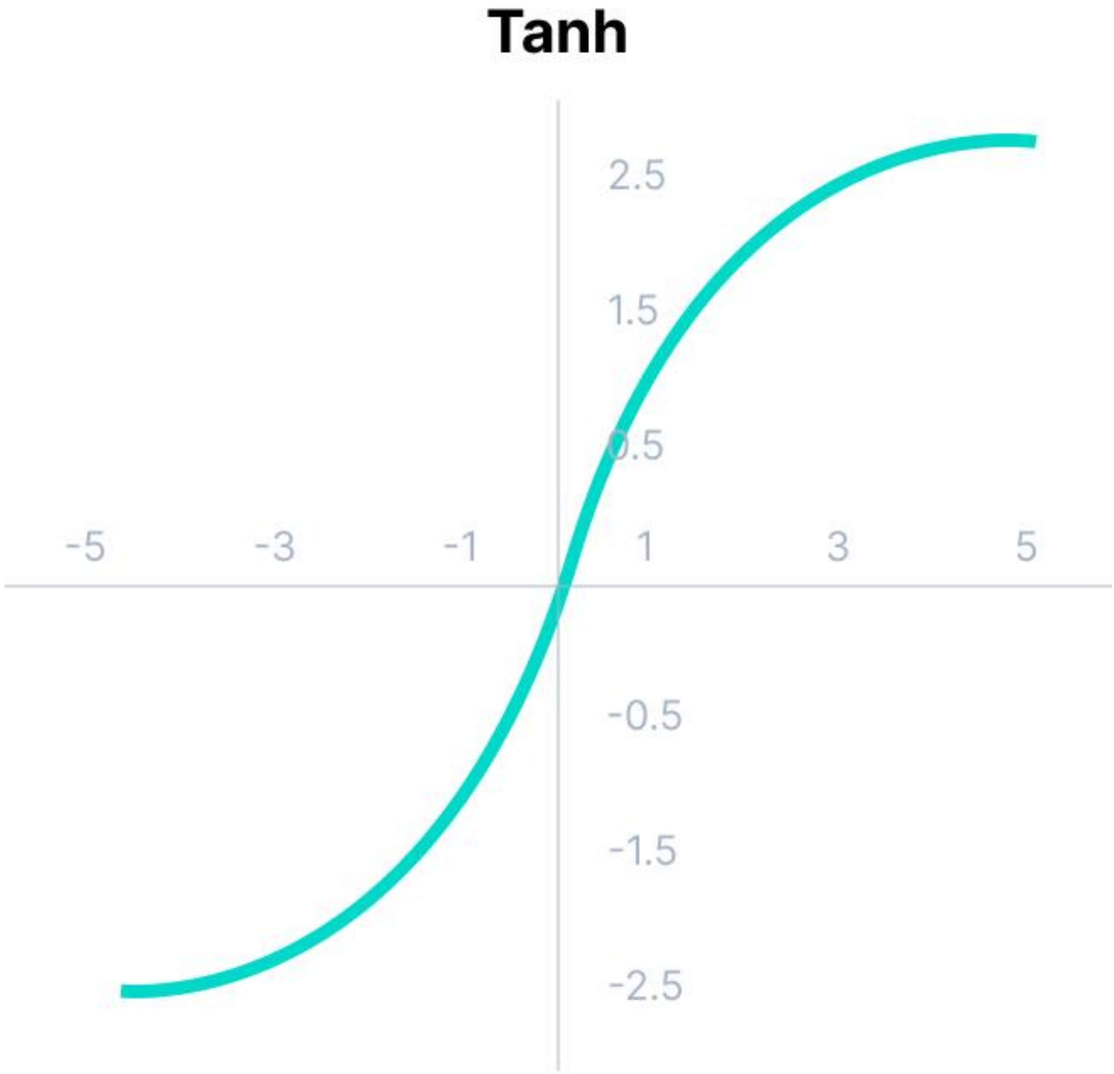


Функции активации

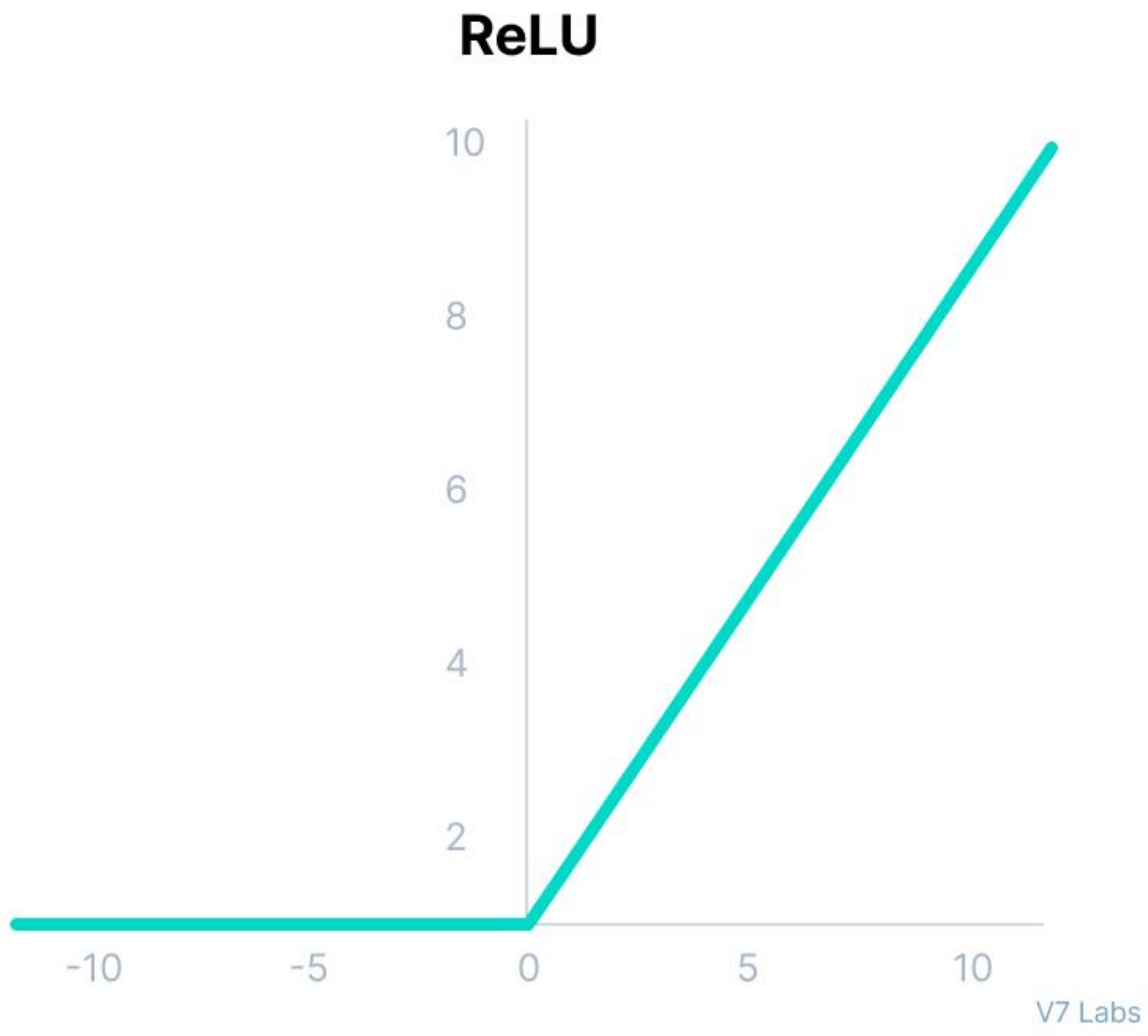


Функции активации

- Недостатки:
- Затухание градиентов
- Производная круче, чем у сигмоиды

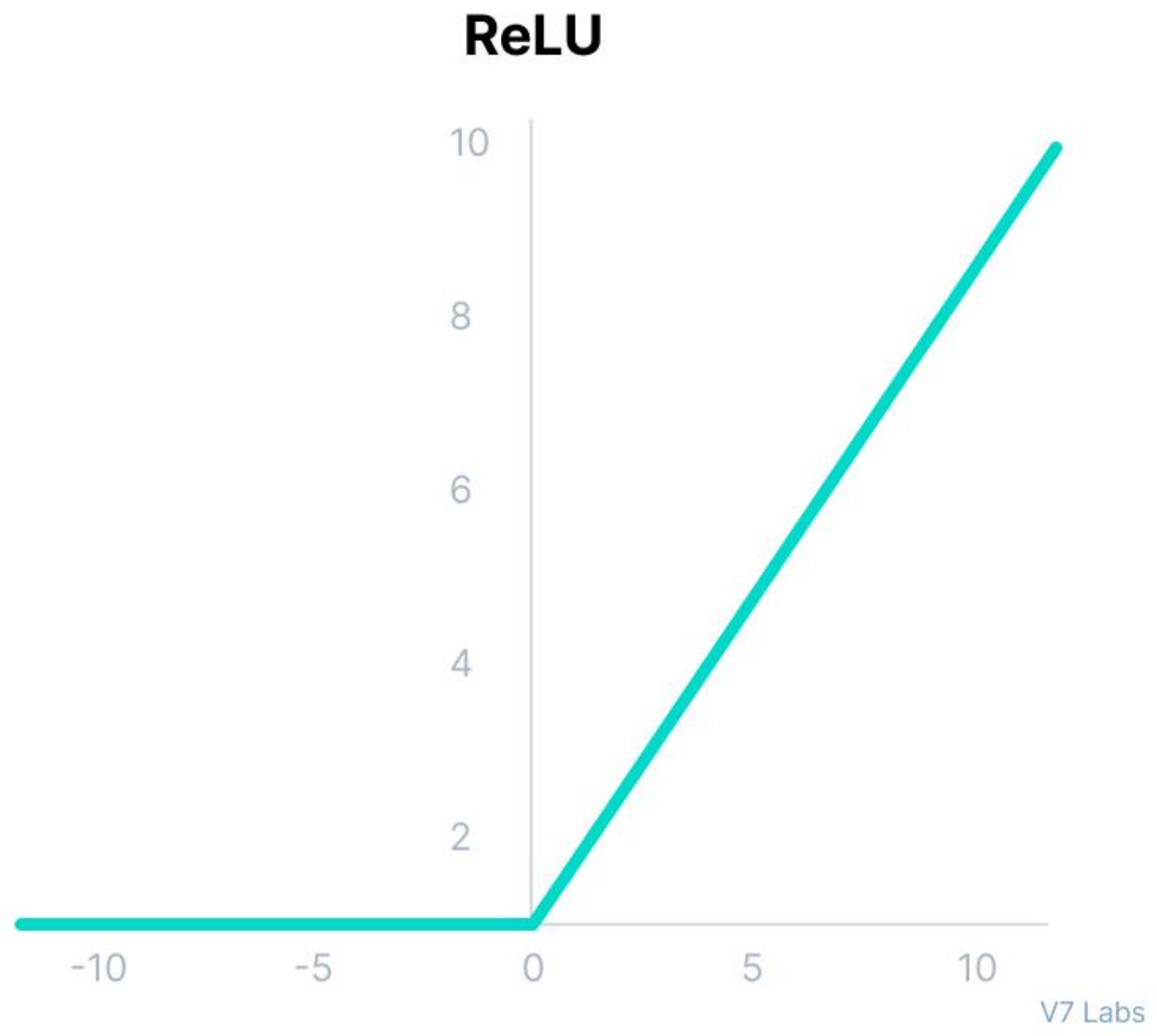


Функции активации



Функции активации

- Недостатки:
- Все отрицательные входные значения немедленно становятся равными нулю, что снижает способность модели правильно подбирать или обучать данные.

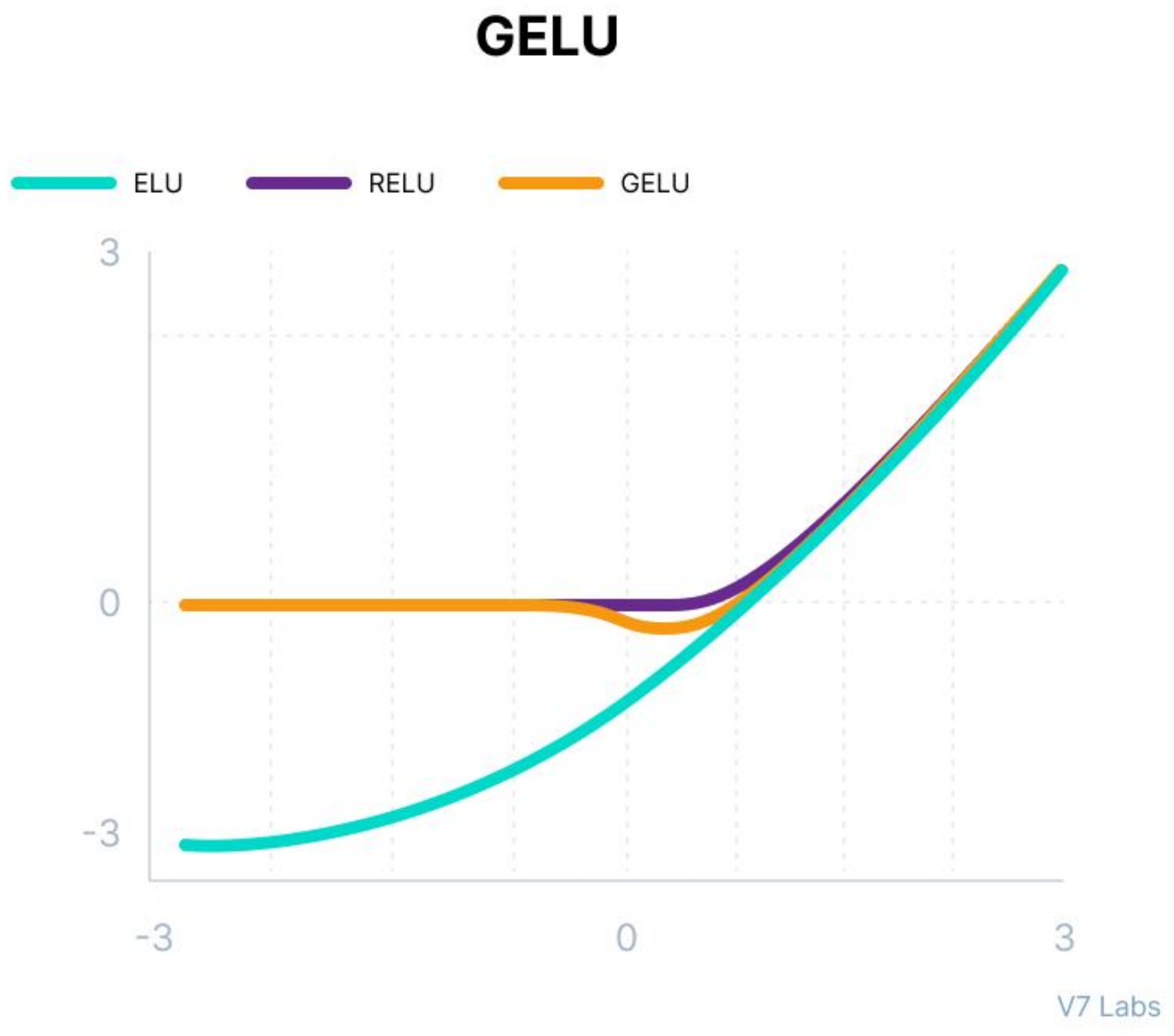


Функции активации

Функции активации

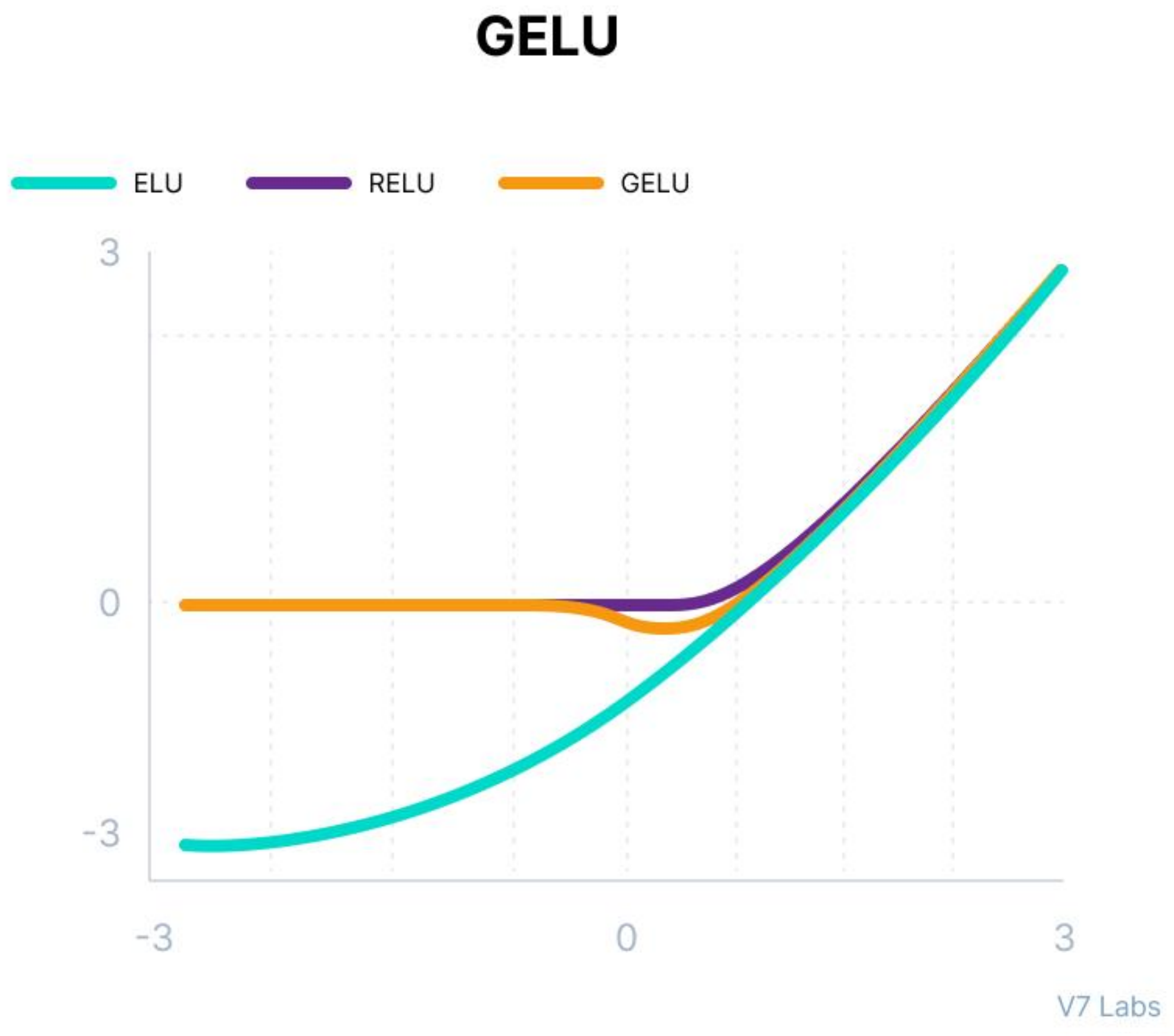
- Недостатки:
- Увеличение времени вычислений из-за экспоненты
- Возможен взрыв градиентов

Функции активации



Функции активации

- На данный момент одна из основных функций активации для больших и глубоких нейронных сетей



Нейронные сети

- Собираем все вместе

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации
- Функция потерь

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации
- Функция потерь
  - Определяем тип задачи и меру штрафа

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации
- Функция потерь
  - Определяем тип задачи и меру штрафа
- Метод оптимизации

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации
- Функция потерь
  - Определяем тип задачи и меру штрафа
- Метод оптимизации
  - Используем градиентный спуск или его модификацию

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
  - Выбираем слои, функции активации
- Функция потерь
  - Определяем тип задачи и меру штрафа
- Метод оптимизации
  - Используем градиентный спуск или его модификацию
- Метрика

Нейронные сети

- Собираем все вместе
- Архитектура нейронной сети
    - Выбираем слои, функции активации
- Функция потерь
    - Определяем тип задачи и меру штрафа
- Метод оптимизации
    - Используем градиентный спуск или его модификацию
- Метрика
    - В зависимости от типа задачи, может совпадать с функцией потерь

Нейронные сети

- Таким образом, мы детально разобрали алгоритм обратного распространения ошибки.
- Изучили различные функции активации, их преимущества и недостатки.
- Разобрали все основные этапы обучения нейронной сети.

Нейронные сети

- Таким образом, мы детально разобрали алгоритм обратного распространения ошибки.
- Изучили различные функции активации, их преимущества и недостатки.
- Разобрали все основные этапы обучения нейронной сети.
- Отличная работа!