

Глубокое обучение

Часть I - Dive into Deep Learning

Власов Кирилл Вячеславович



2020

Kirill Vlasov

PM / PO @ Zyfra AI

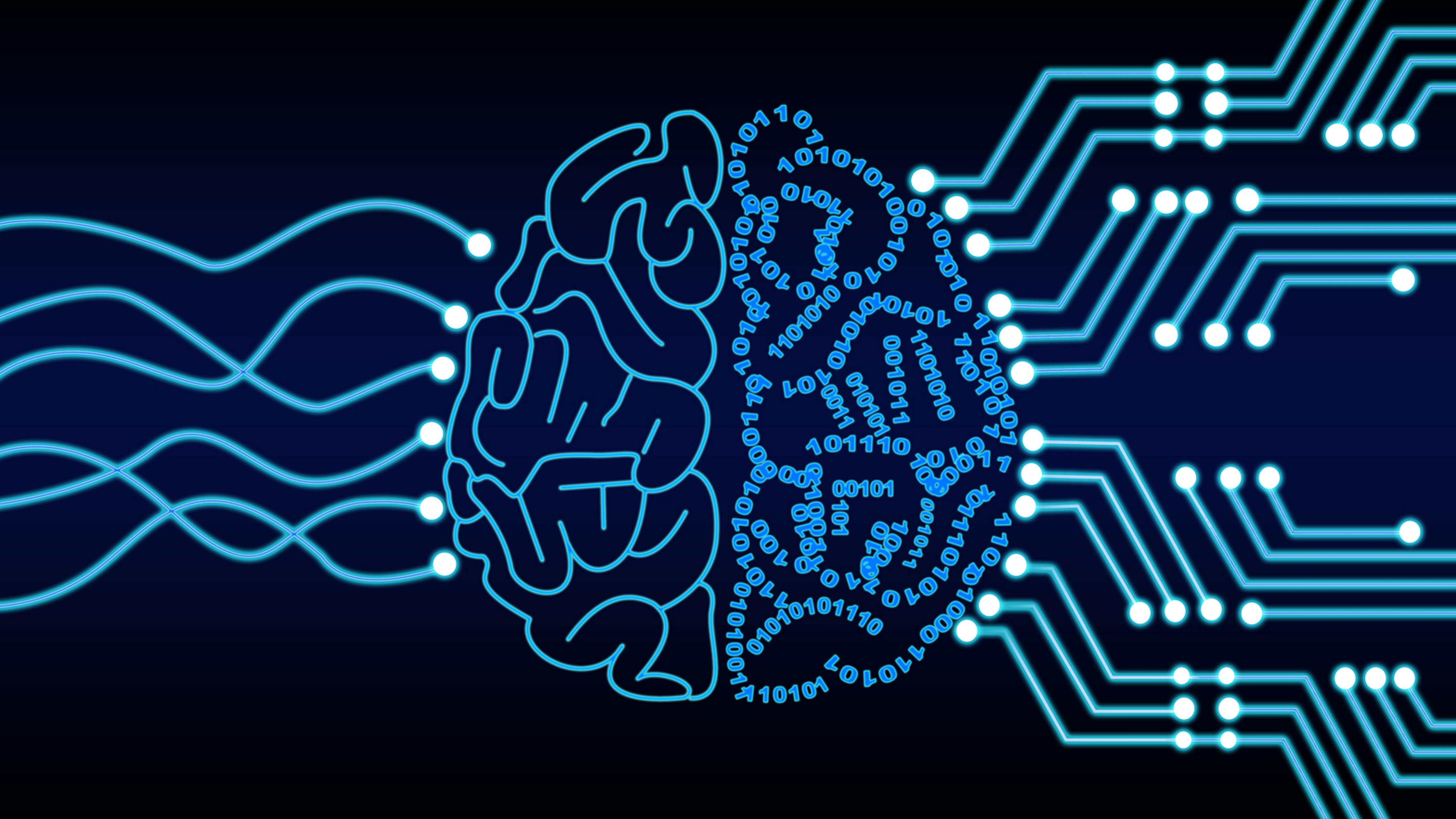


Data Fest



SKILLFACTORY





1997

Deep Blue выиграл
чемпиона мира по
шахматам Гарри
Каспарова.



2016

*AlphaGo выиграла
матч у
профессионала Ли
Седоль*



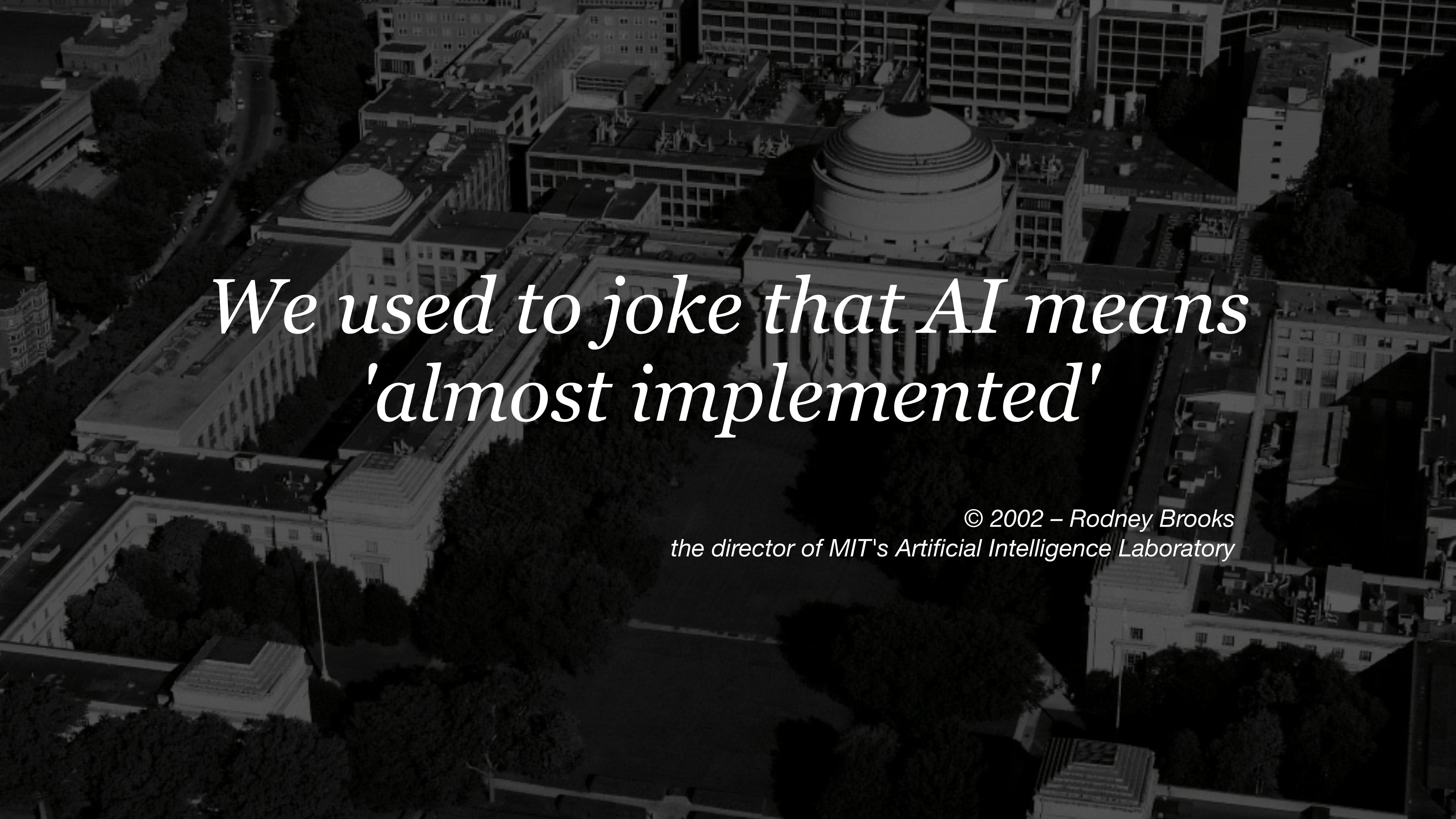


2019

*AlphaStar от
DeepMind Выиграл
10:1 команду Team
Liquid.*

*А это
точно
AI?*



The background of the slide is a black and white aerial photograph of the Massachusetts Institute of Technology (MIT) campus. The image shows a dense cluster of buildings, including the iconic Great Dome, several large research facilities, and modern office buildings. The surrounding area includes trees and some roads.

*We used to joke that AI means
'almost implemented'*

© 2002 – Rodney Brooks
the director of MIT's Artificial Intelligence Laboratory

Давайте решим задачку?

Сколько минут в 3 часах?



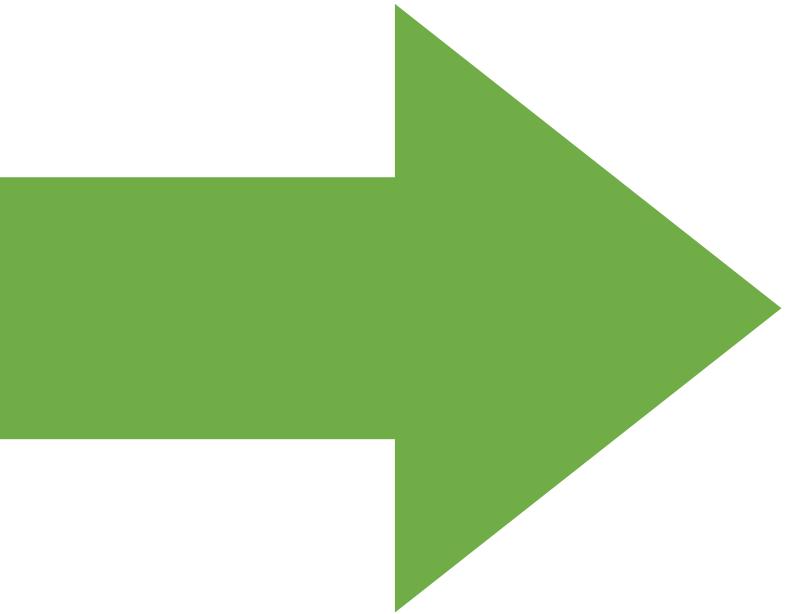
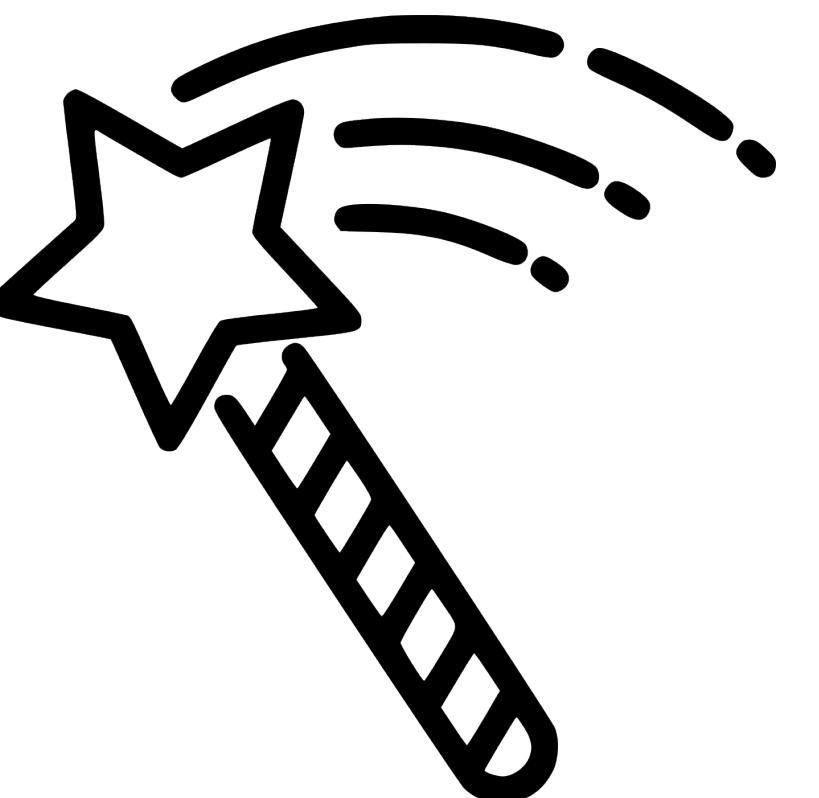
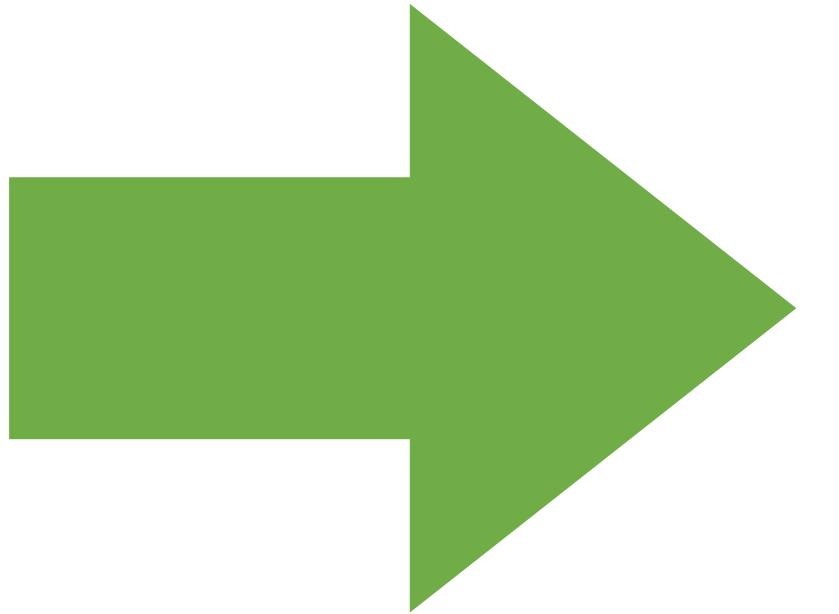
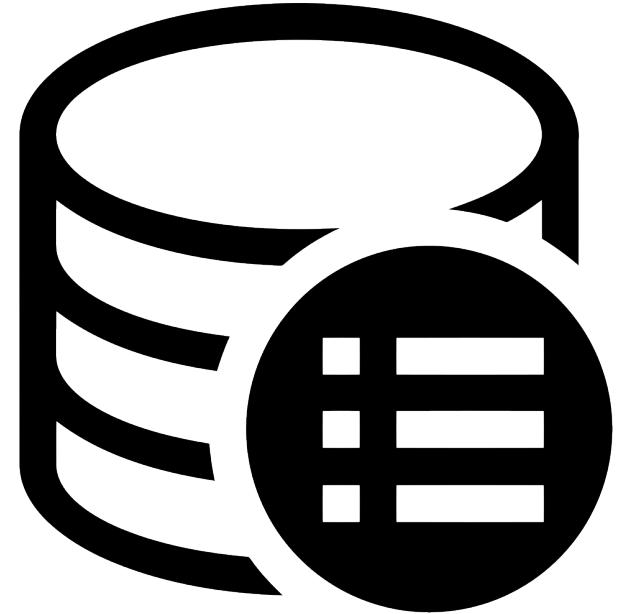
Давайте решим задачку?

Сколько минут в 3 часах?

$$f(x) = 60 \times X$$

$$f(3) = 60 \times 3 = 180$$



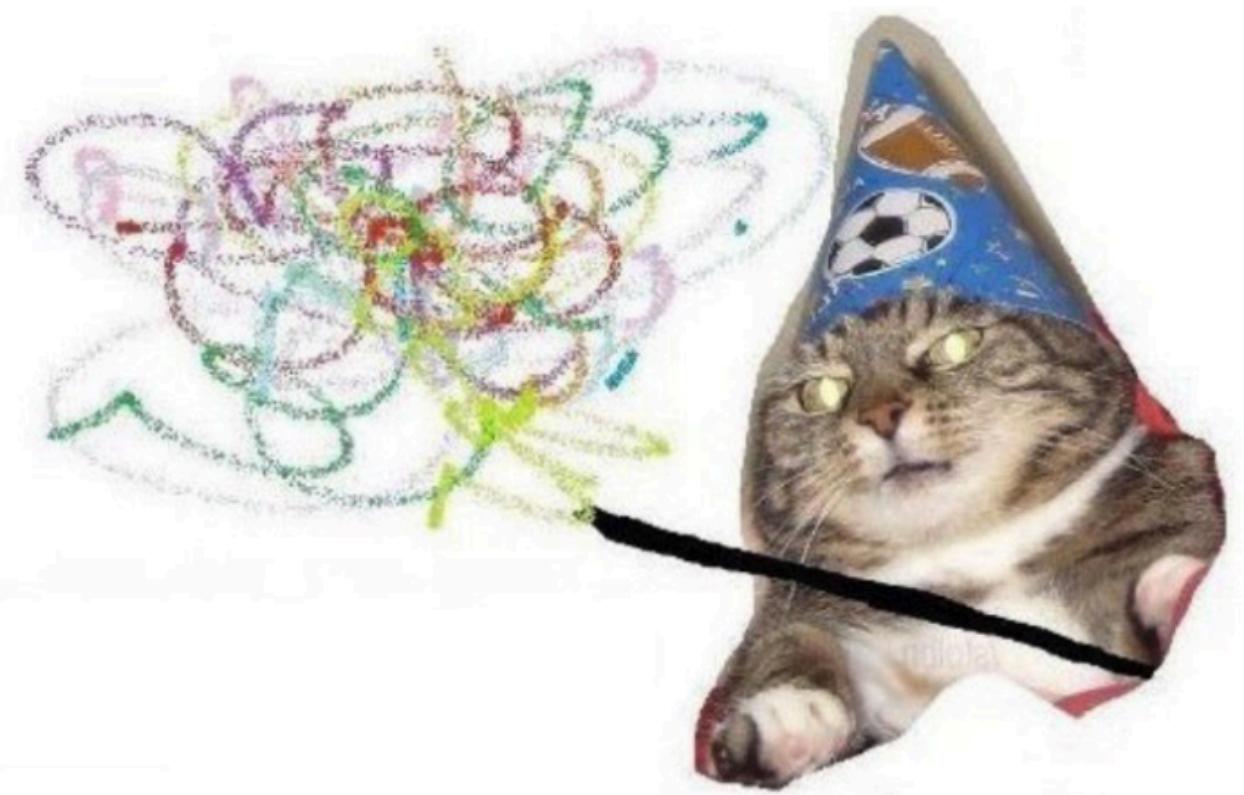


$f(x)$

Провести тысячи экспериментов

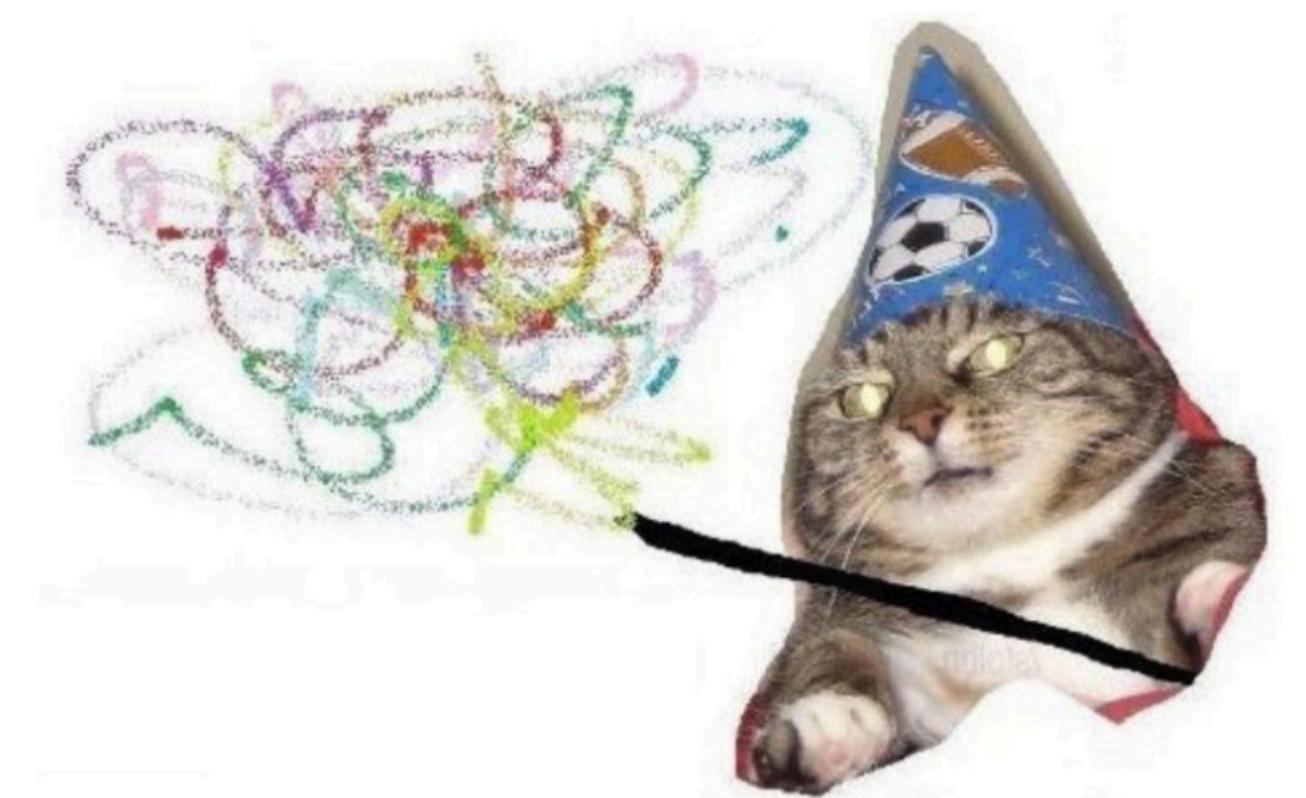
Вжух!

Найти истинную функцию





Machine Learning



Искусственный интеллект лучше человека?

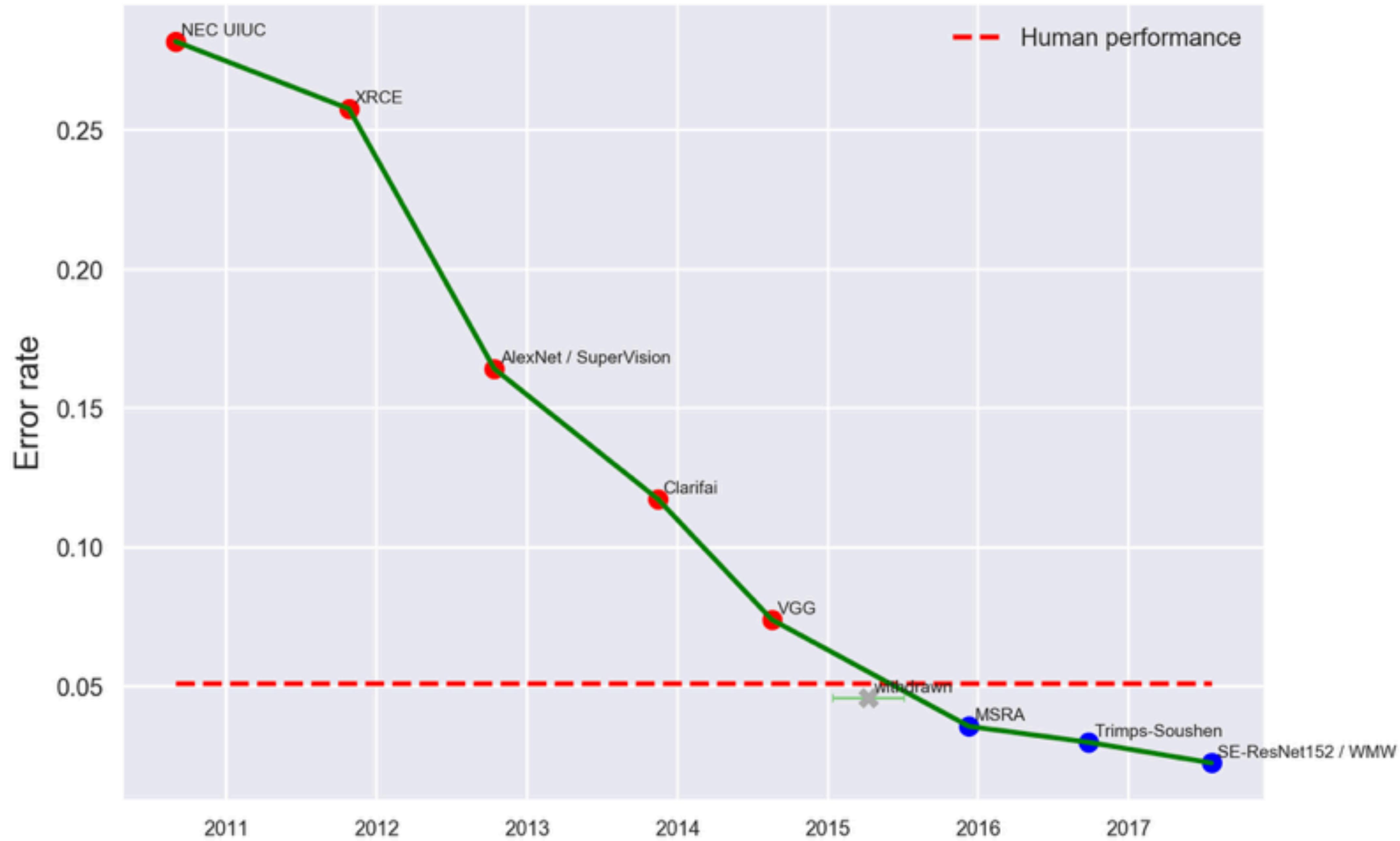


Распознавание изображений

ImageNet

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.





Превосходство NN над человеком



– Какой класс у этой картинки?
Например, есть классы "horse" и
"woman's clothing»...

Превосходство NN над человеком



– Какой класс у этой картинки?
Например, есть классы "horse" и
"woman's clothing". Но
правильный ответ, конечно, "hay".
– Разумеется. сена-то вон
сколько. а бабы с конем еле-еле!

(c) <https://habr.com/post/303196/>

Распознавание образов



Распознавание образов



Histogram of oriented gradients

Input image



Histogram of Oriented Gradients



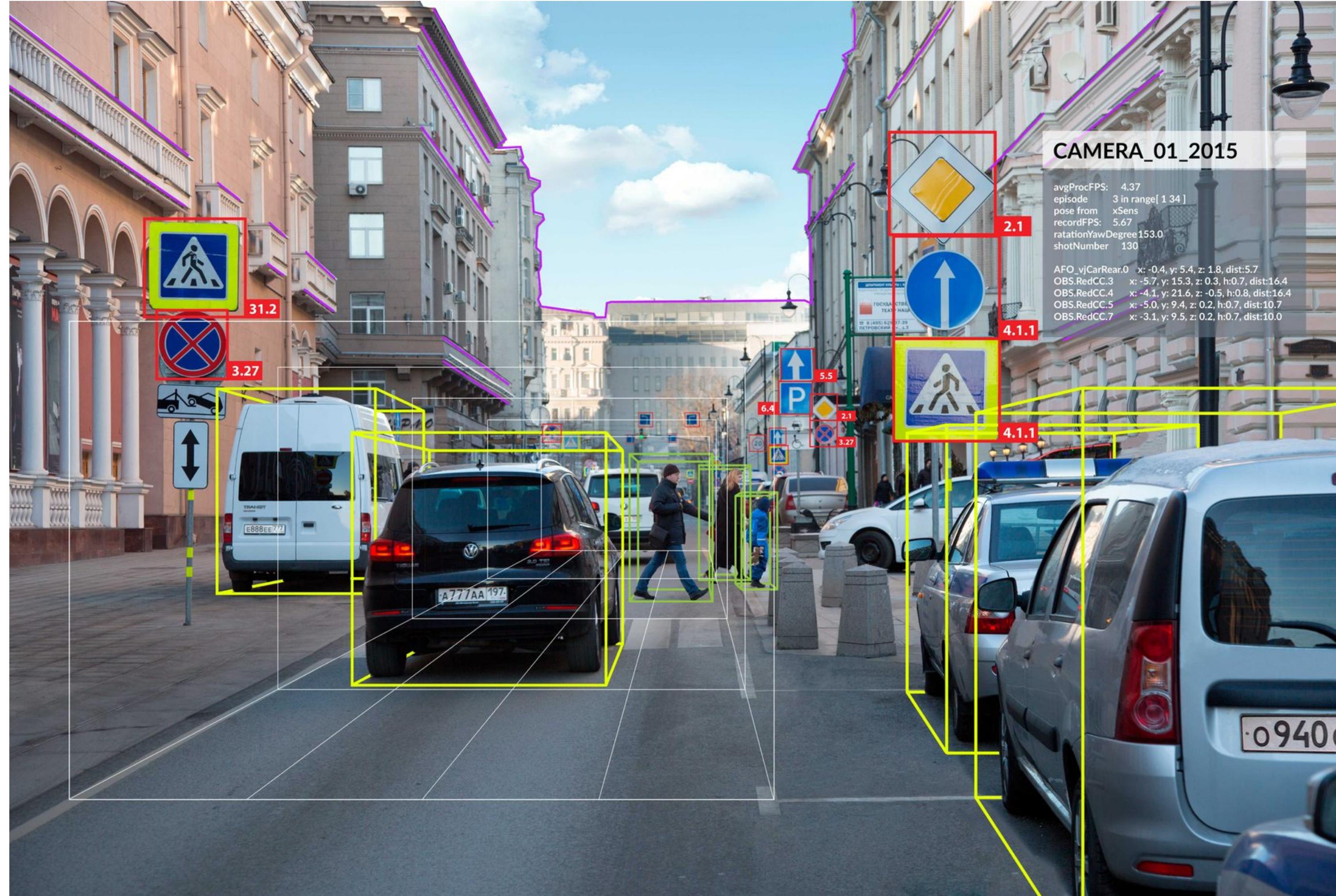
Распознавание образов



Распознавание образов



Сложности



Беспилотный автомобиль

Представьте, что мы обучили нейросеть определять объекты с видеорегистратора.

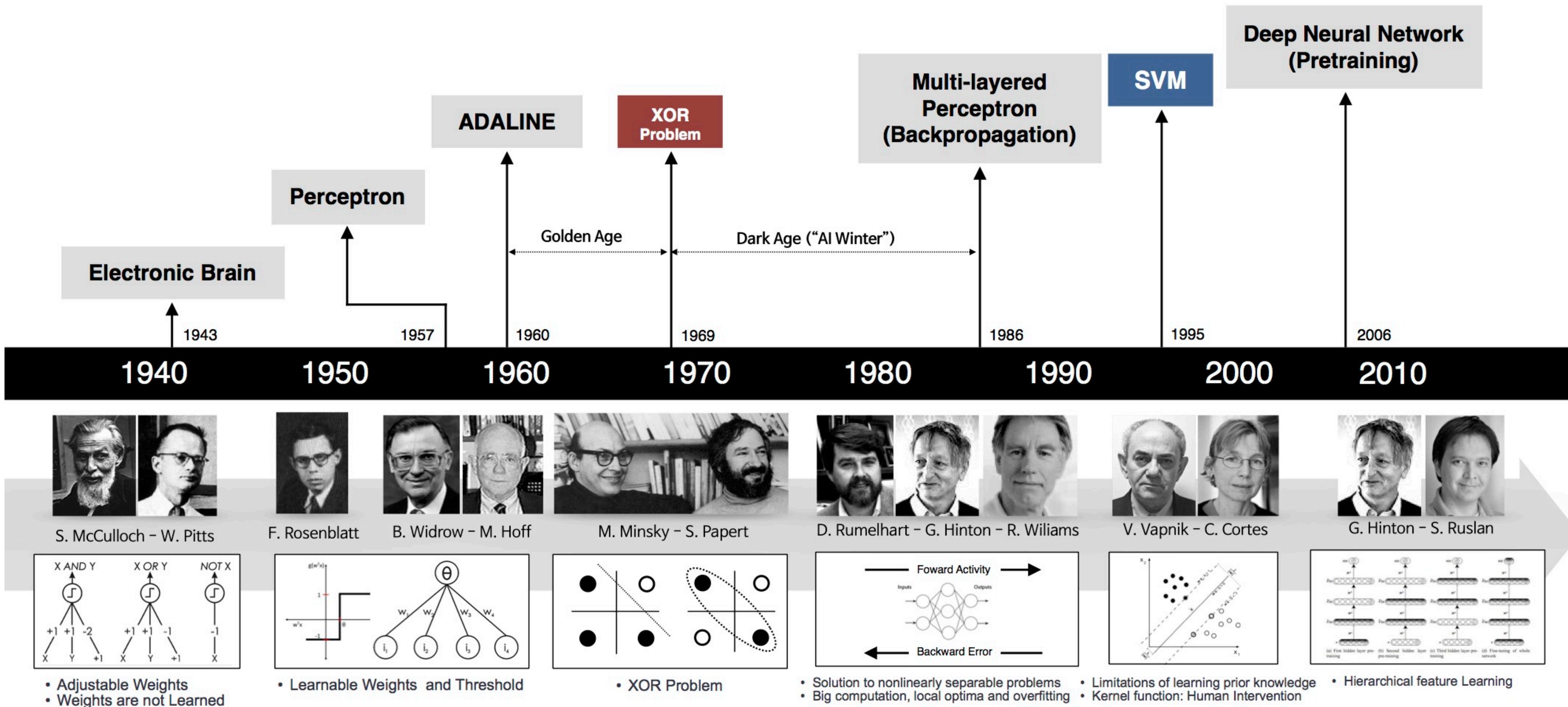
Сложности



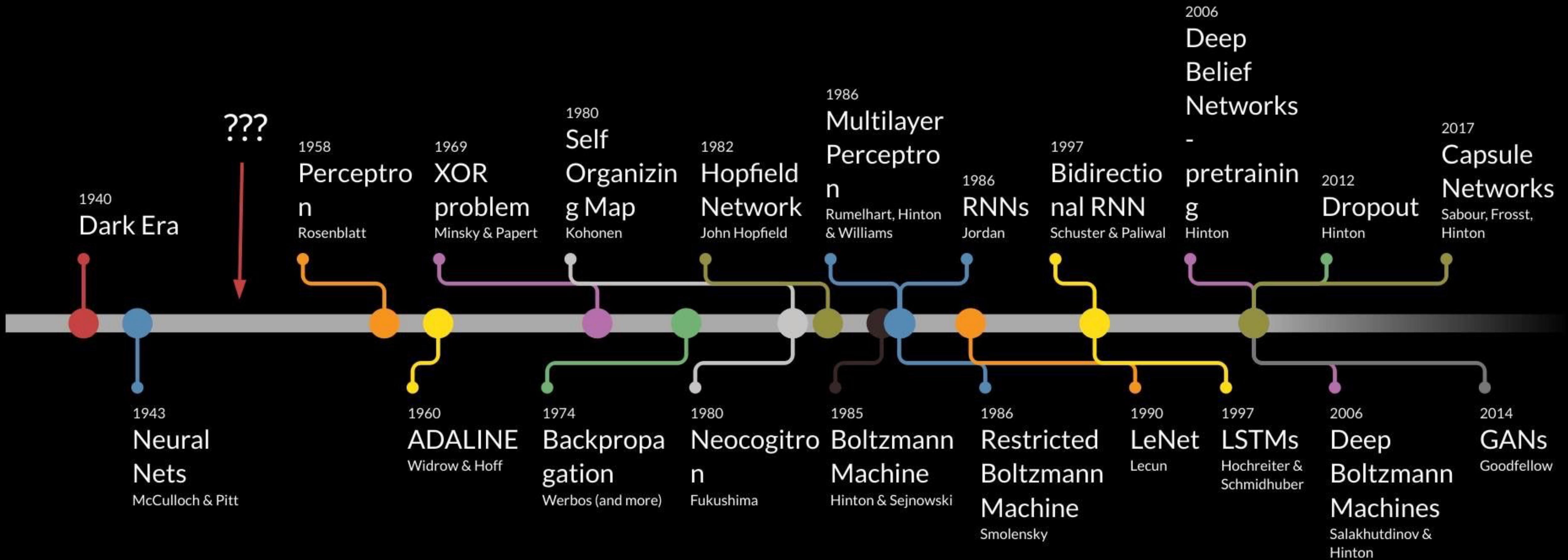
Беспилотный автомобиль

...что будет, если мы
встретим на дороге
такое?

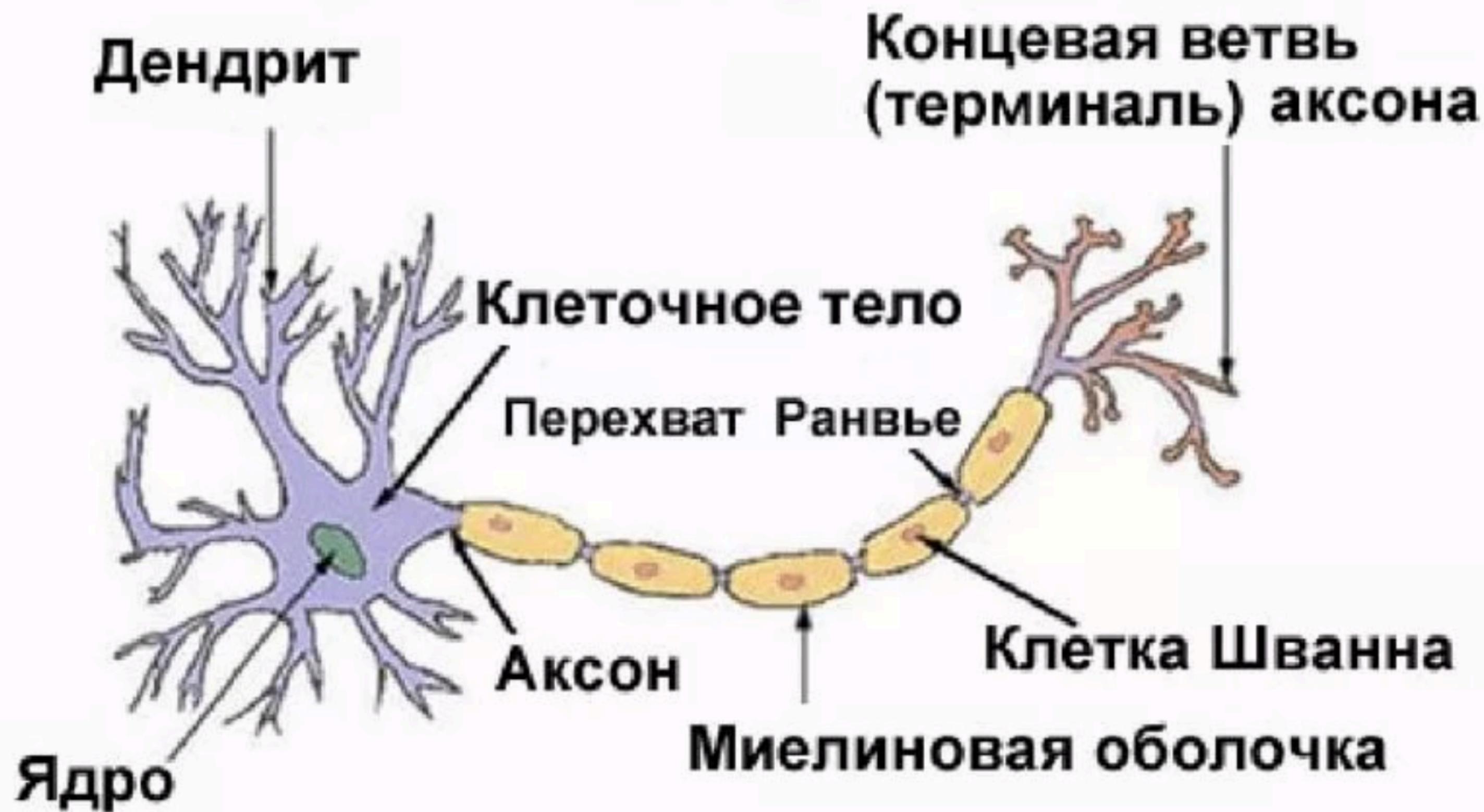
Deep Learning: Timeline



Deep Learning: Timeline



Аналогия с биологическим нейроном



Дендриты — короткие и сильно разветвлённые отростки нейрона, служащие главным местом для образования влияющих на нейрон возбуждающих и тормозных синапсов

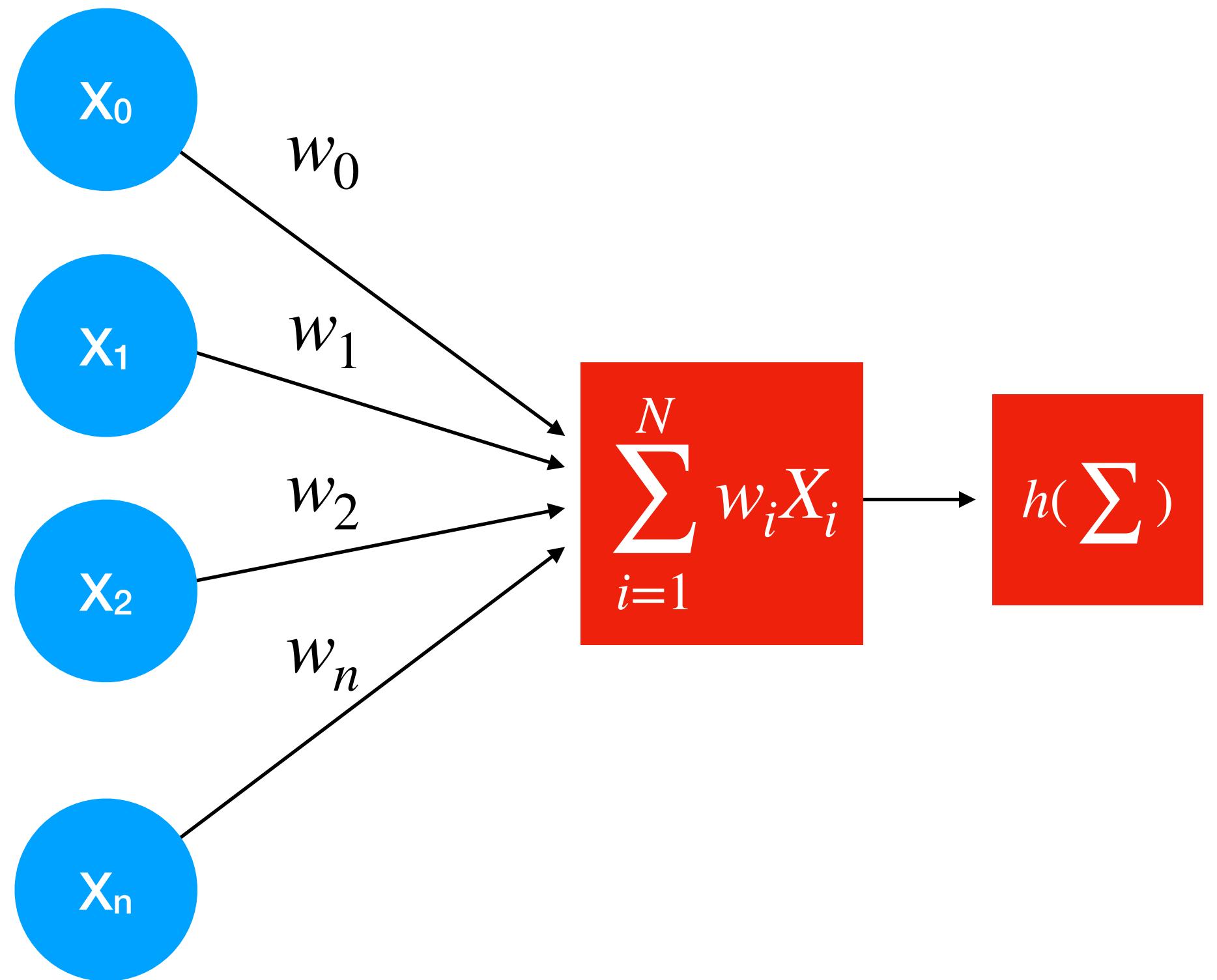
аксон генерирует и проводит по своему телу электрические сигналы называемые потенциалами действия. Эти сигналы образуются и распространяются благодаря электрически заряженным ионам натрия (Na^+), калия (K^+), хлора (Cl^-), кальция (Ca^{2+}).

На конце аксона находится синаптическое окончание — концевой участок терминали, контактирующий с клеткой-мишенью. Вместе с постсинаптической мембраной клетки-мишени синаптическое окончание образует синапс. Через синапсы передаётся возбуждение.

Искусственный нейрон

Математический нейрон Маккаллока – Питтса

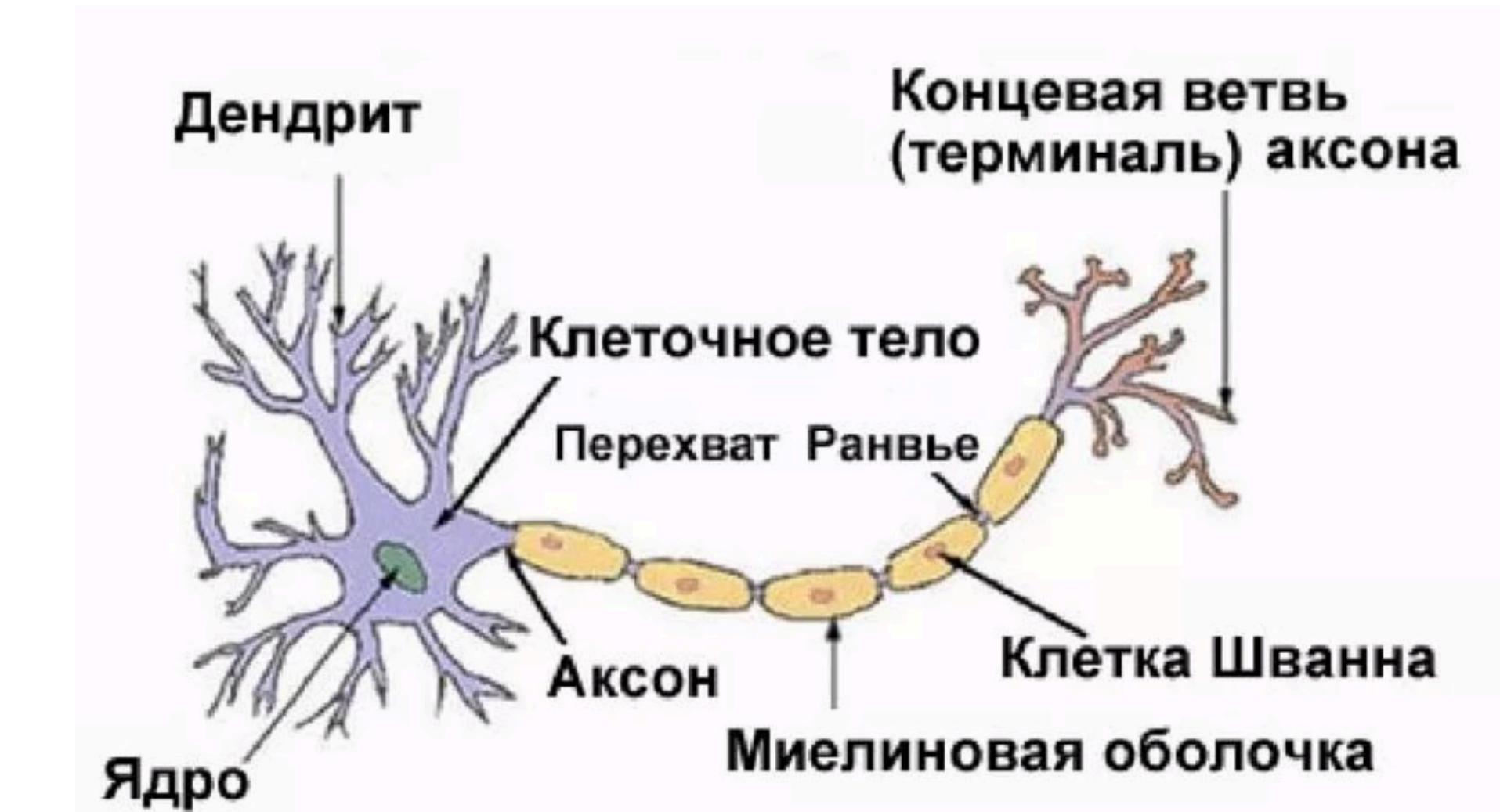
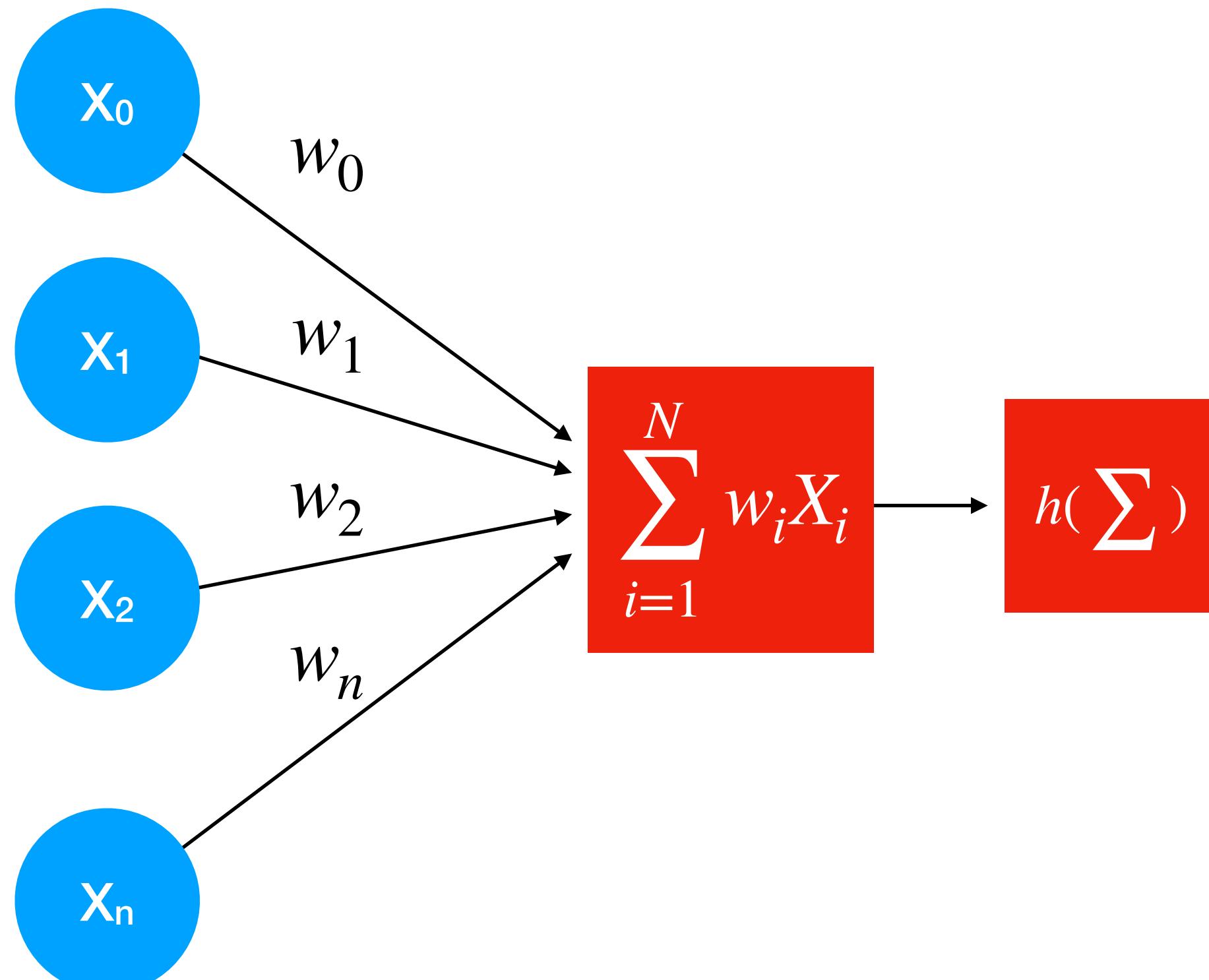
40-ые годы XX века



Искусственный нейрон

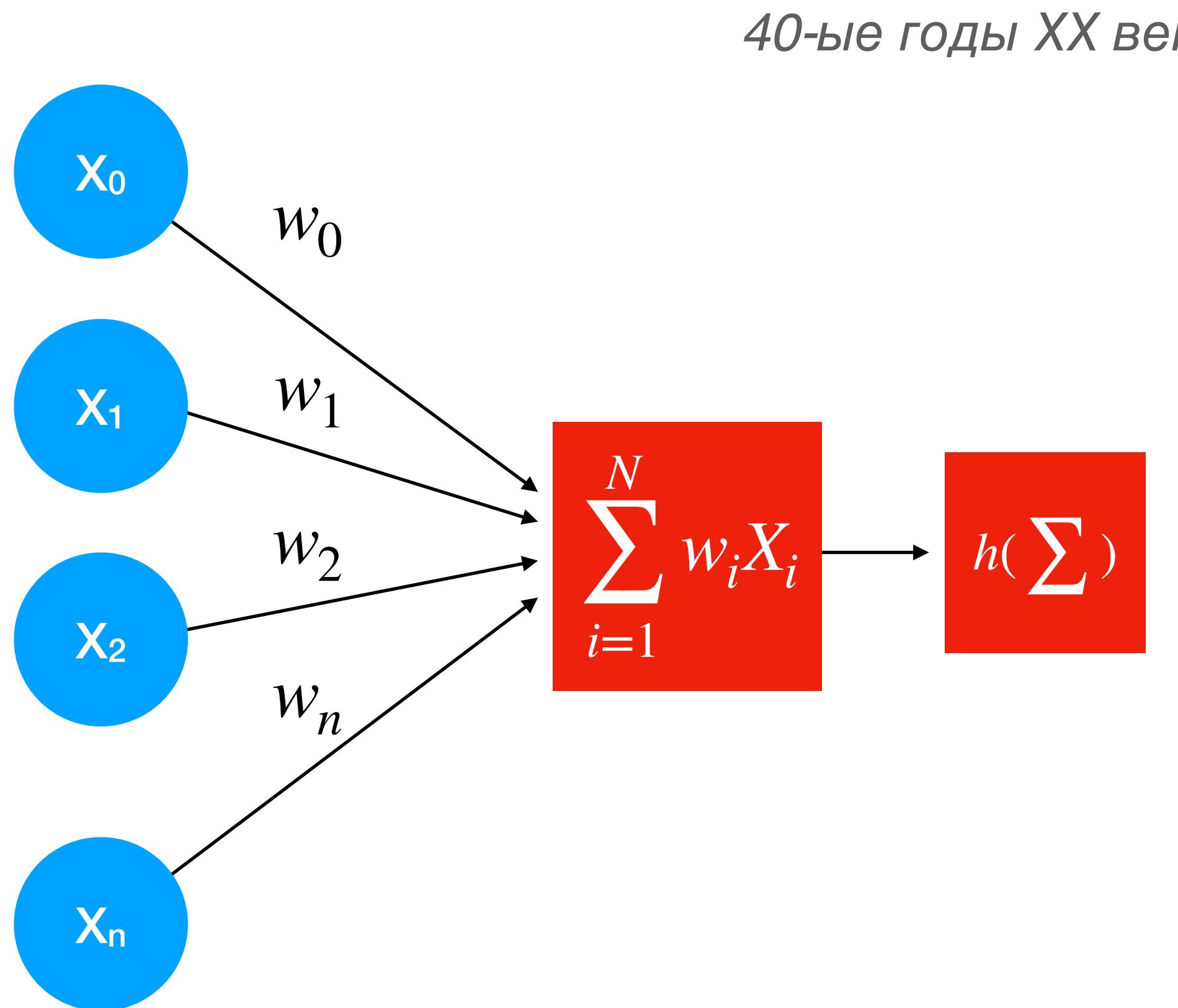
Математический нейрон Маккаллока – Питтса

40-ые годы XX века

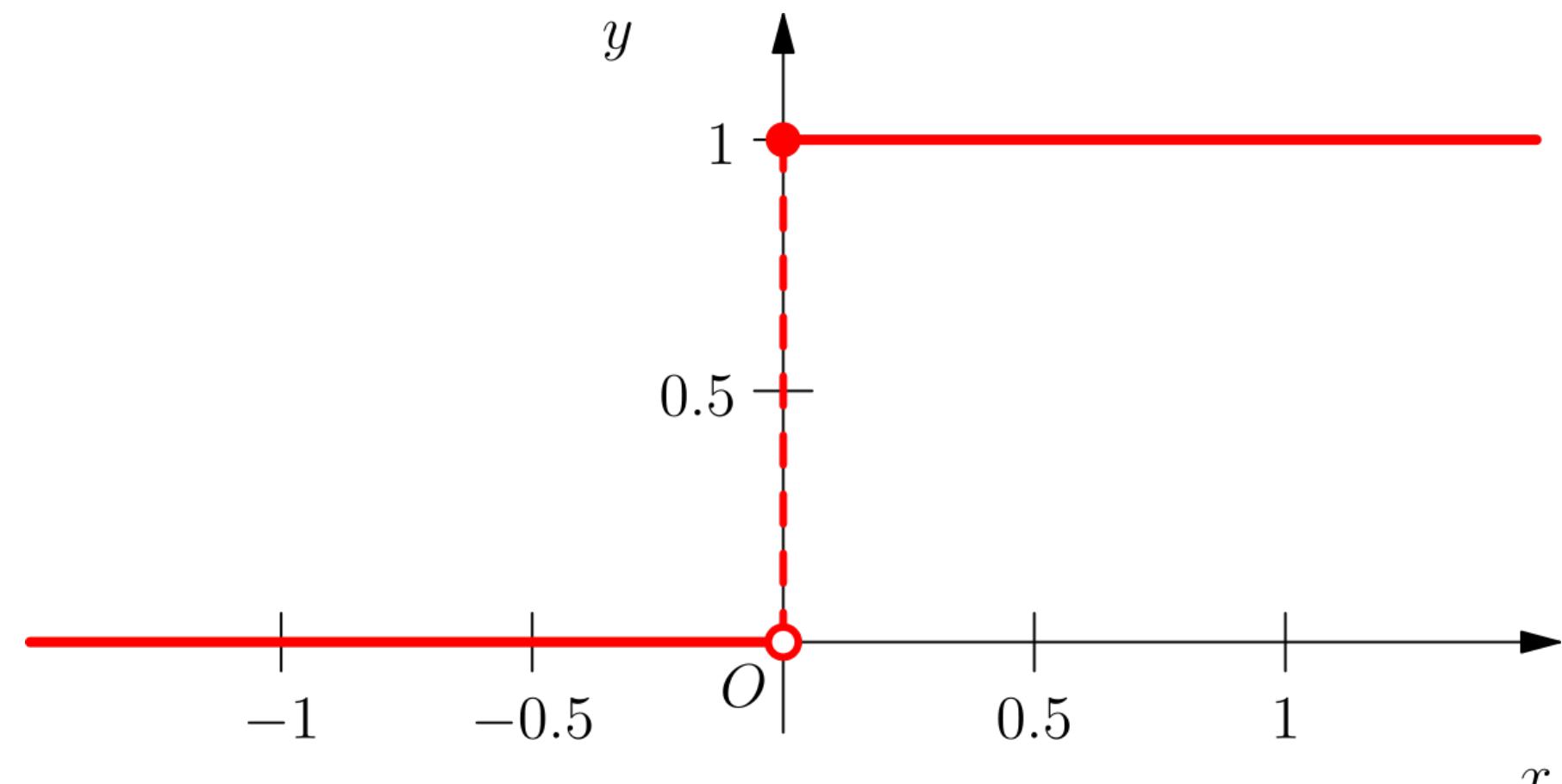


Искусственный нейрон

Математический нейрон Маккаллока – Питтса

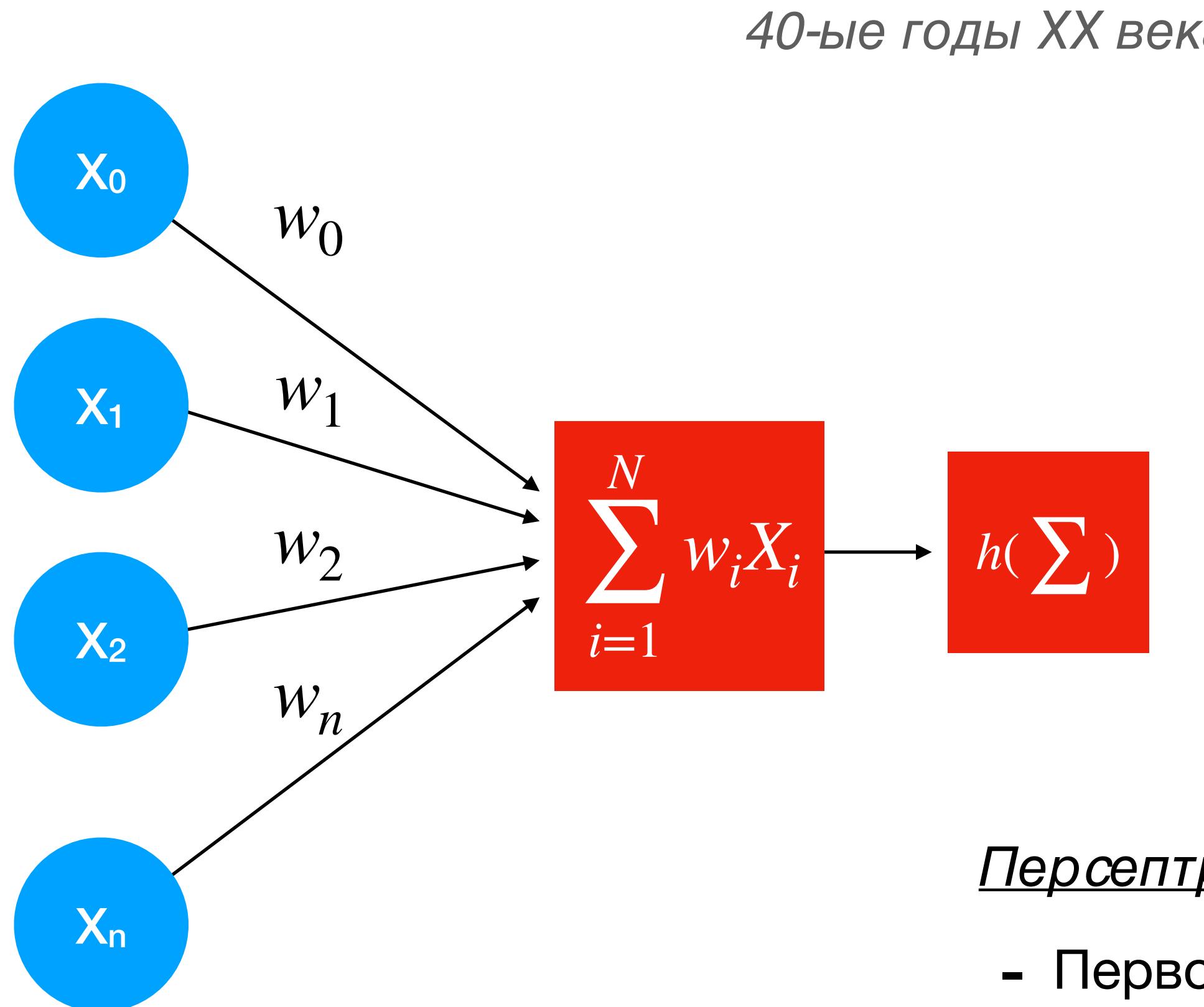


Функция Хевисайда

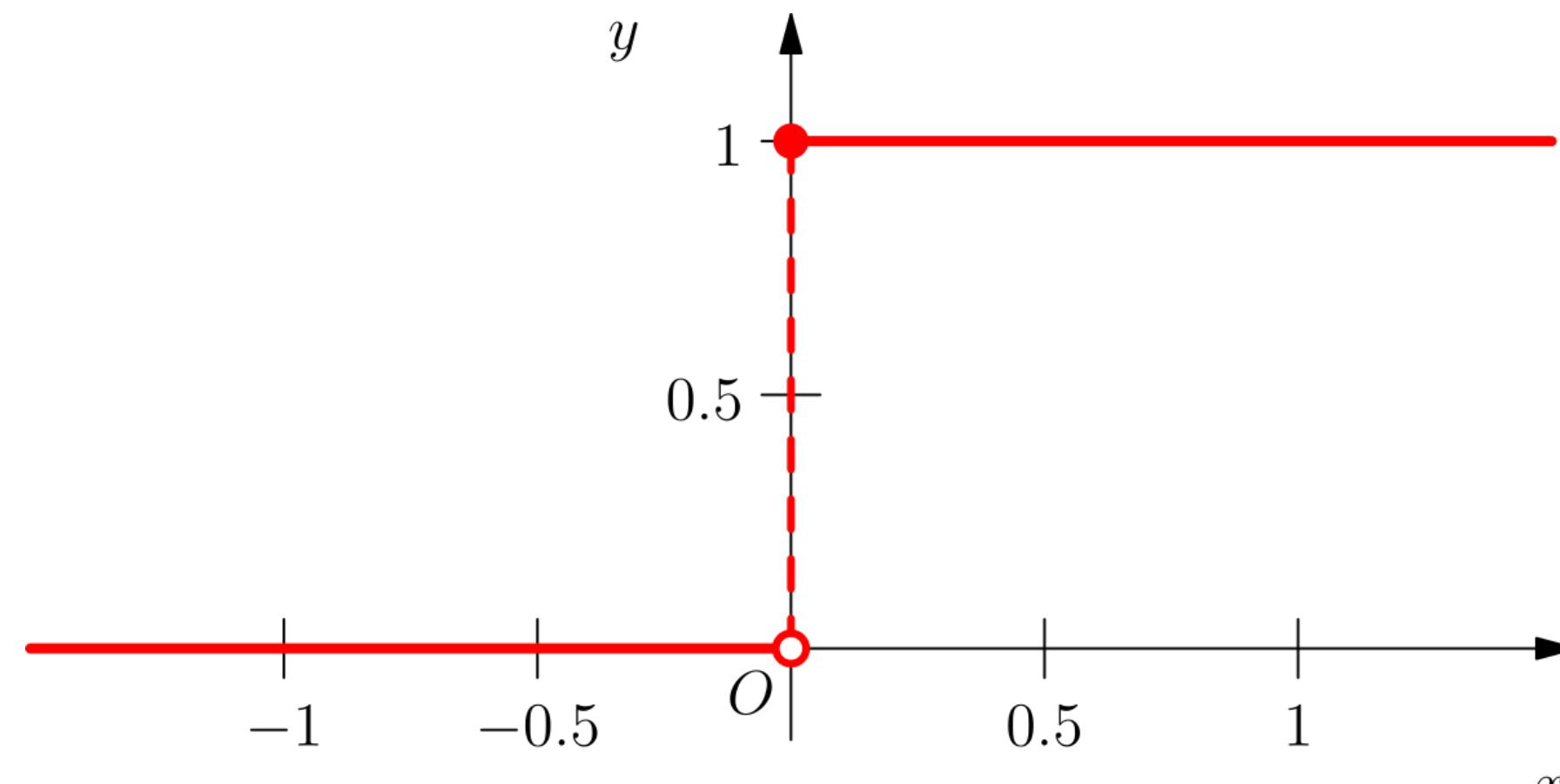


Искусственный нейрон

Математический нейрон Маккаллока – Питтса



Функция Хевисайда

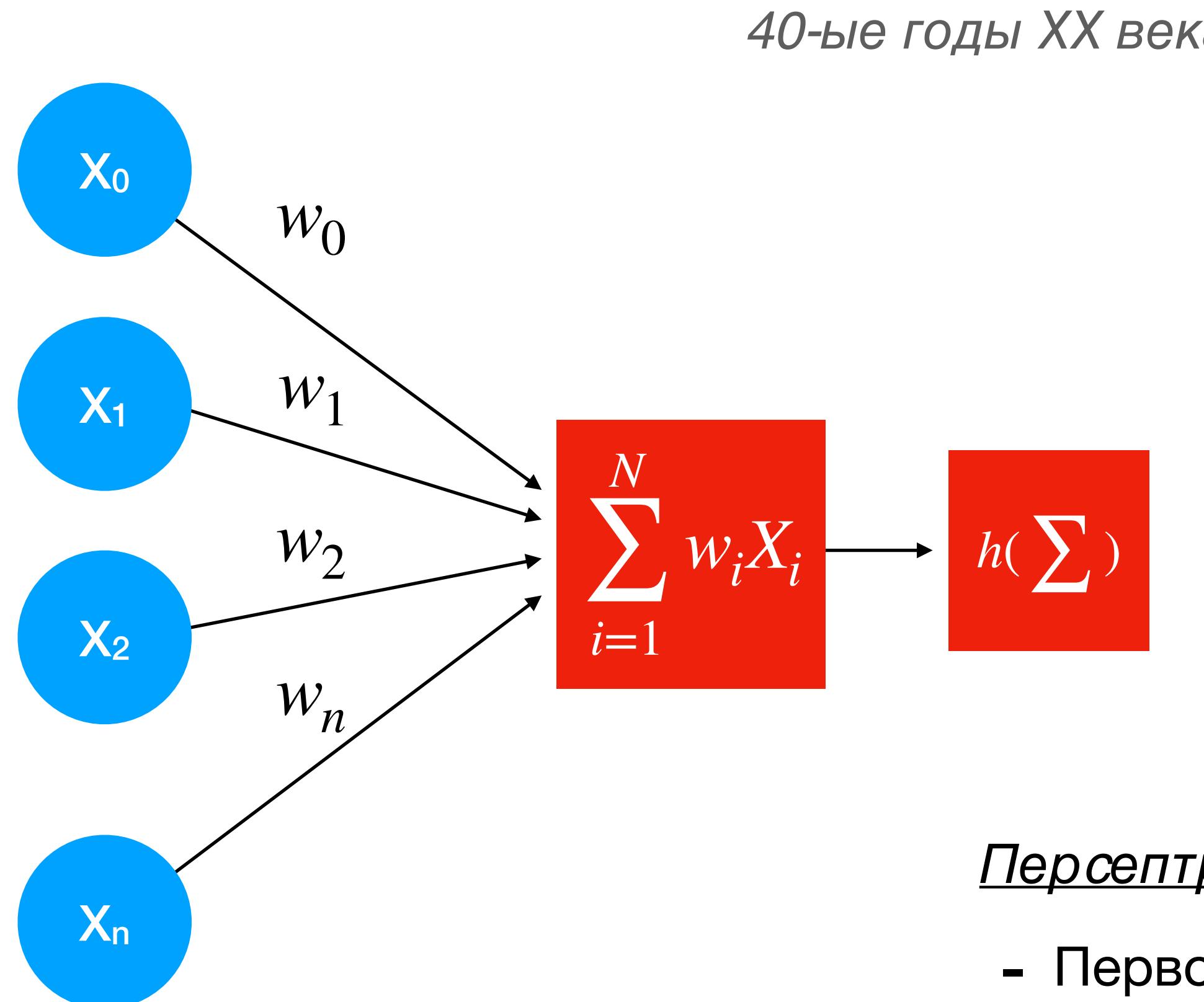


Персепtron Розенблatta

- Первое правило Хебба – *Если сигнал персептрана неверен и равен нулю, то необходимо увеличить веса тех входов, на которые была подана единица.*
- Второе правило Хебба – *Если сигнал персептрана неверен и равен единице, то необходимо уменьшить веса тех входов, на которые была подана единица.*

Искусственный нейрон

Математический нейрон Маккаллока – Питтса



Функция Хевисайда



Биологическая предпосылка:
Если нейрон срабатывает, то синоптическая связь укрепляется

Персепtron Розенблatta

- Первое правило Хебба – *Если сигнал персептрана неверен и равен нулю, то необходимо увеличить веса тех входов, на которые была подана единица.*
- Второе правило Хебба – *Если сигнал персептрана неверен и равен единице, то необходимо уменьшить веса тех входов, на которые была подана единица.*

Как обучить логистическую регрессию?

Как обучить логистическую регрессию?

Задать функцию потерь:

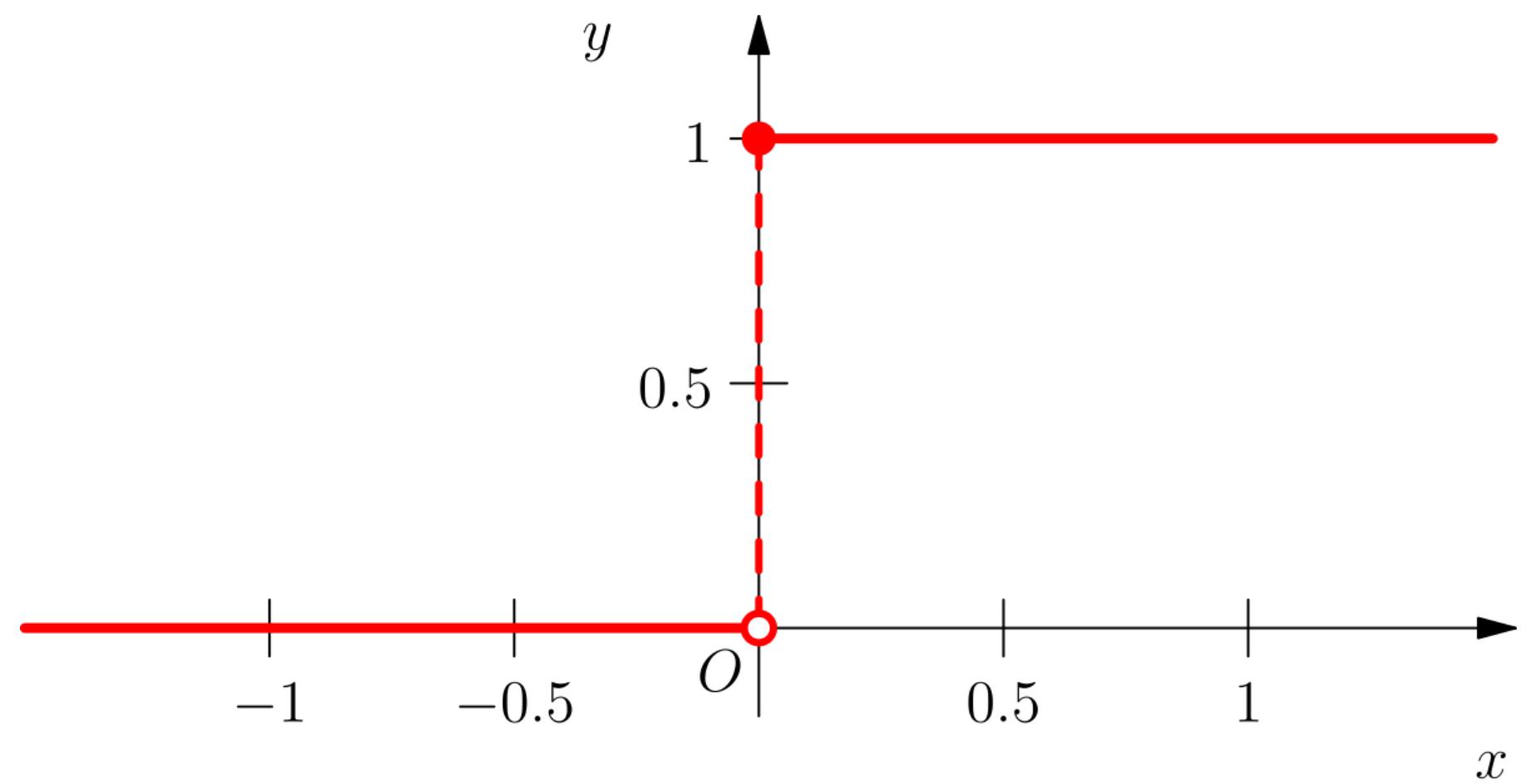
$$\tilde{L}(M) = \log(1 + e^{-M})$$

Минимизация эмпирического риска:

$$\mathcal{Q}(a, X) = \frac{1}{l} \sum_{i=1}^l \log(1 + e^{-y_i \langle w, x_i \rangle}) \rightarrow \min$$

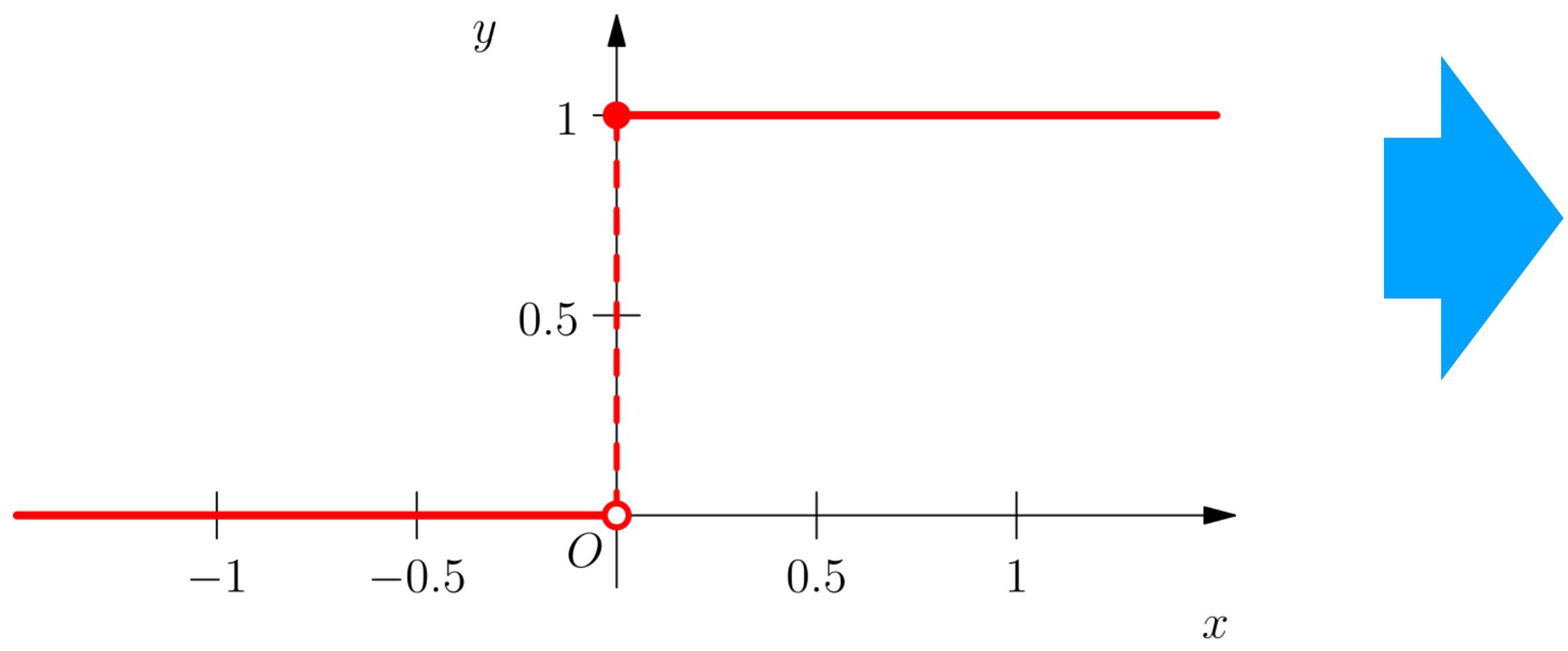
Искусственный нейрон

Функция Хевисайда



Искусственный нейрон

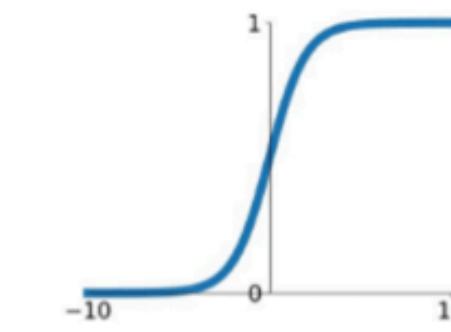
Функция Хевисайда



Другие функции активации:

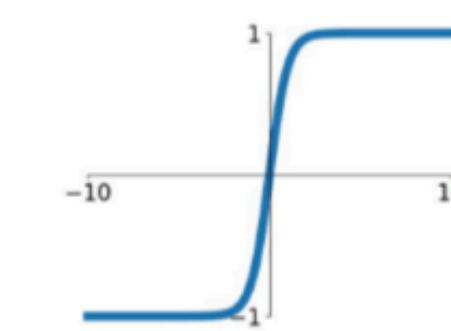
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



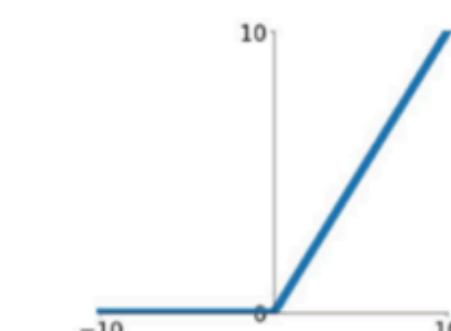
tanh

$$\tanh(x)$$



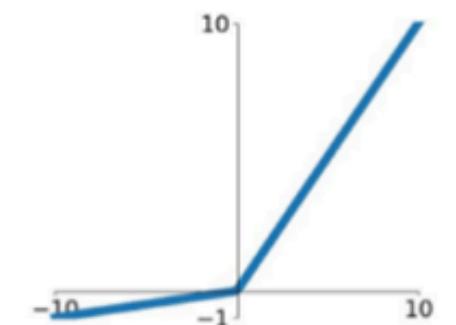
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

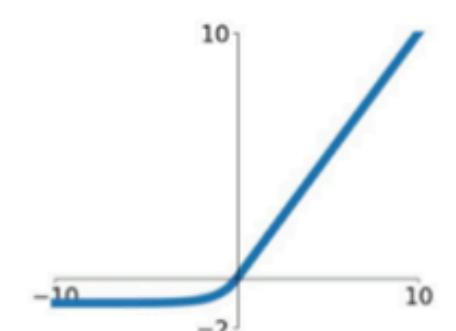


Maxout

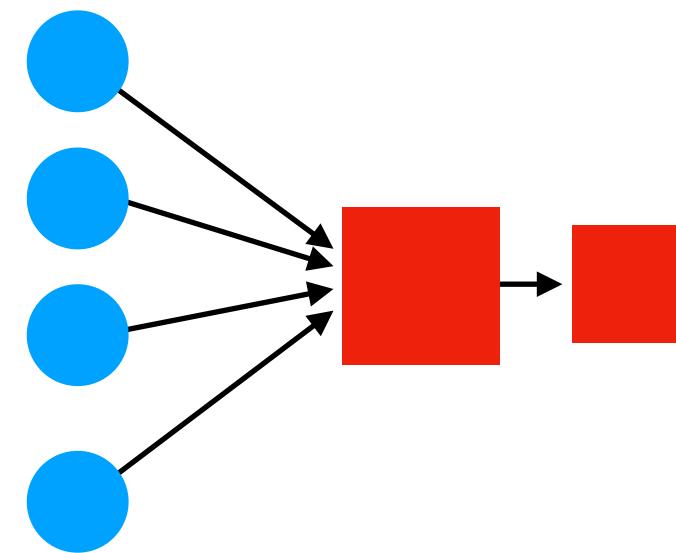
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

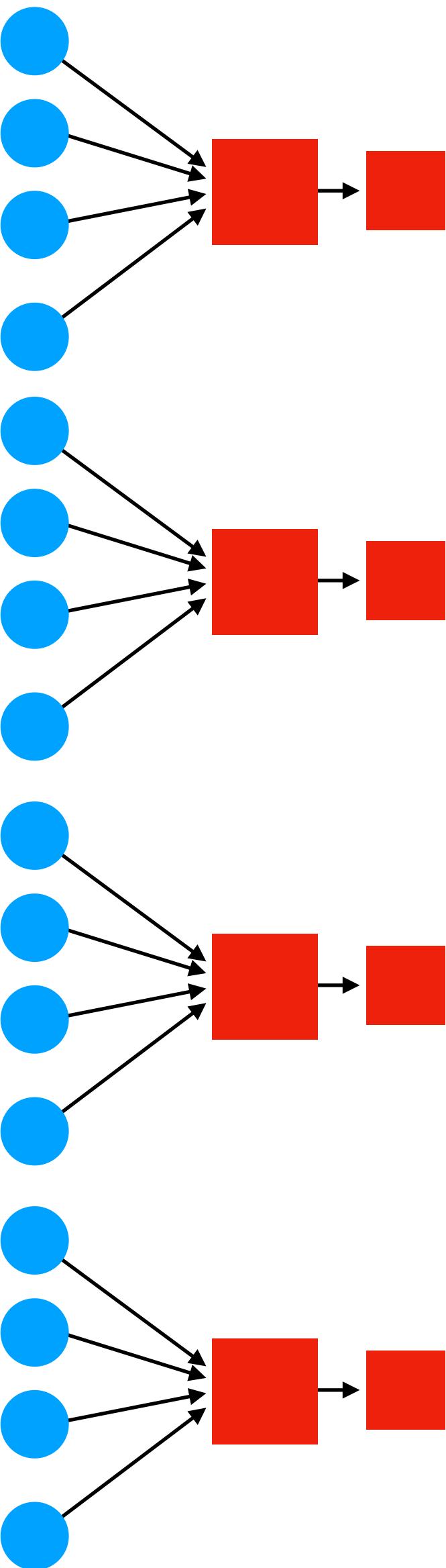
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



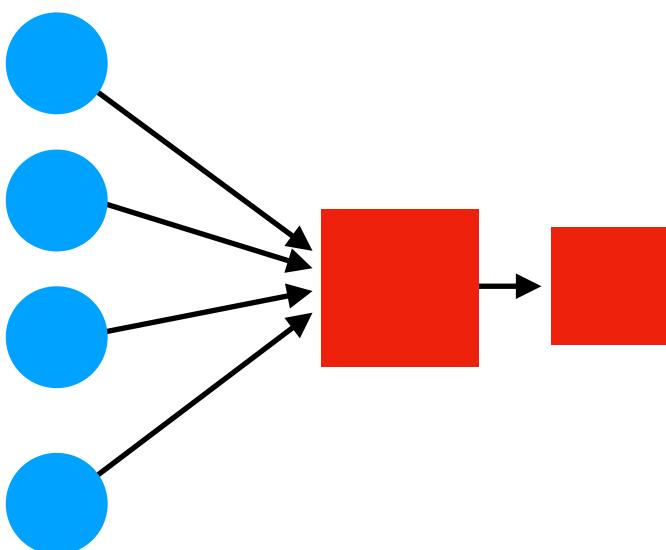
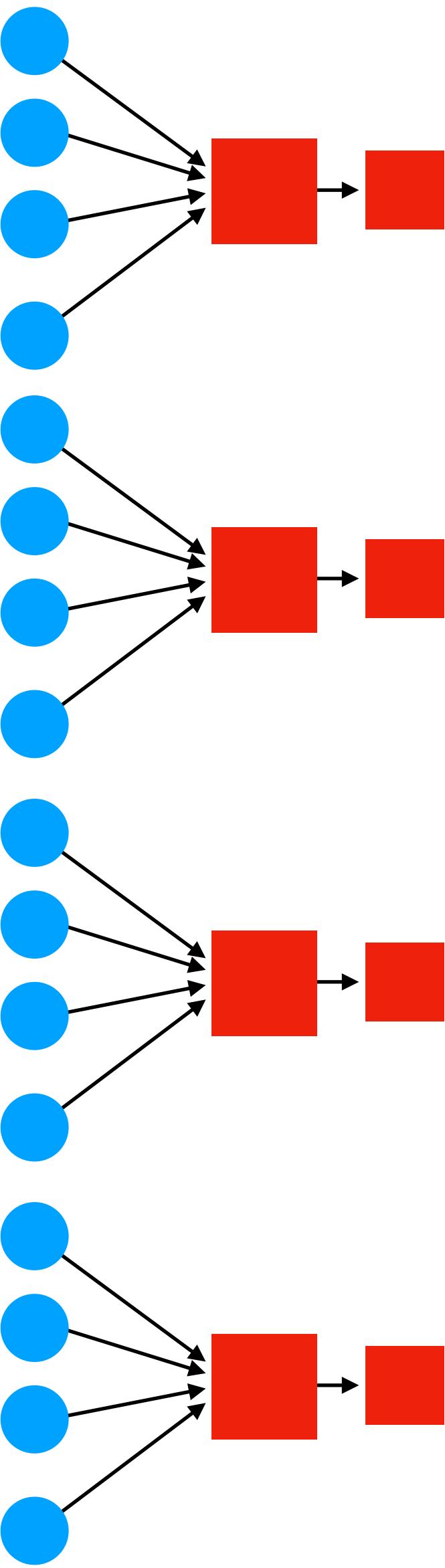
Нейронные сети



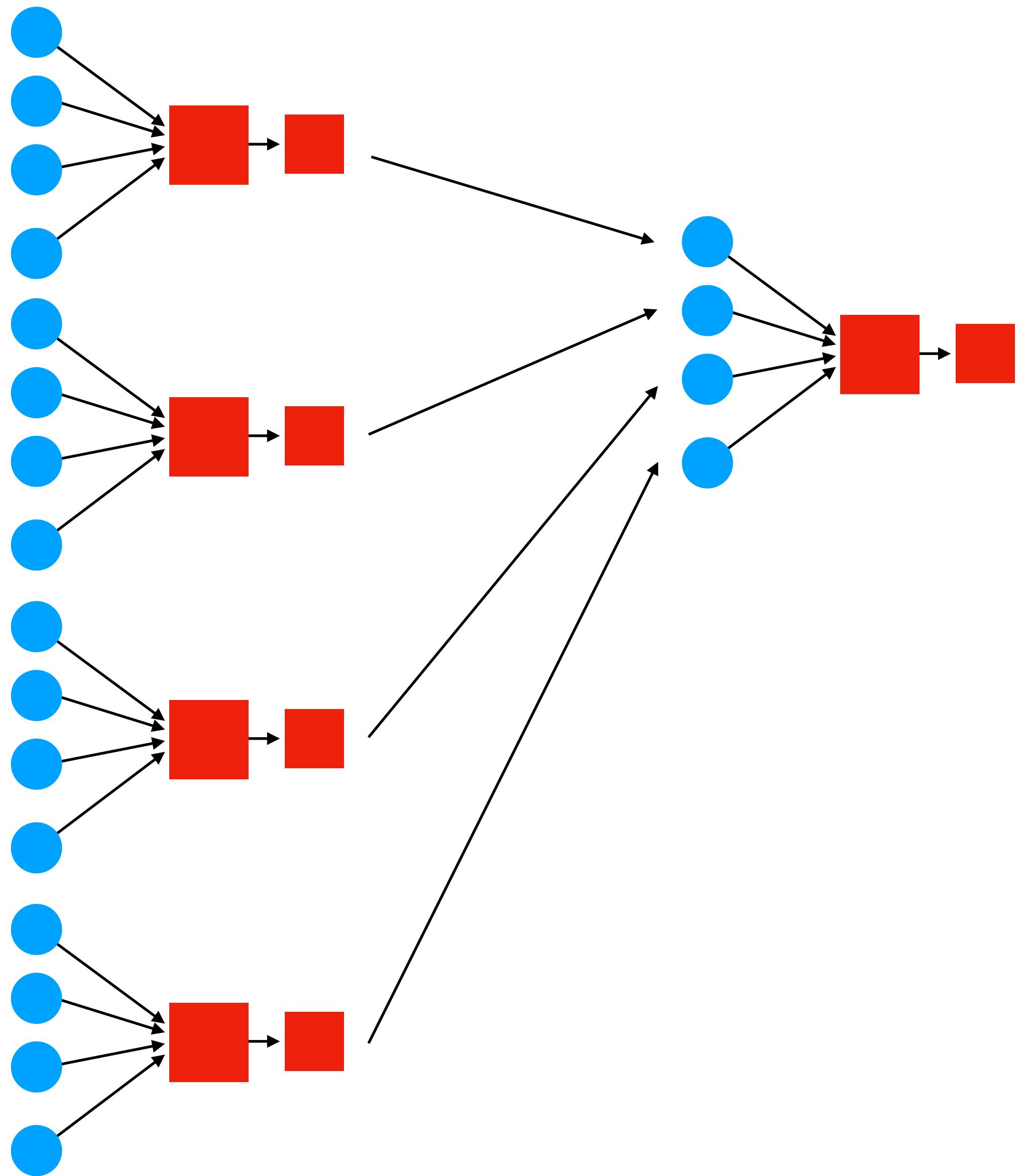
Нейронные сети



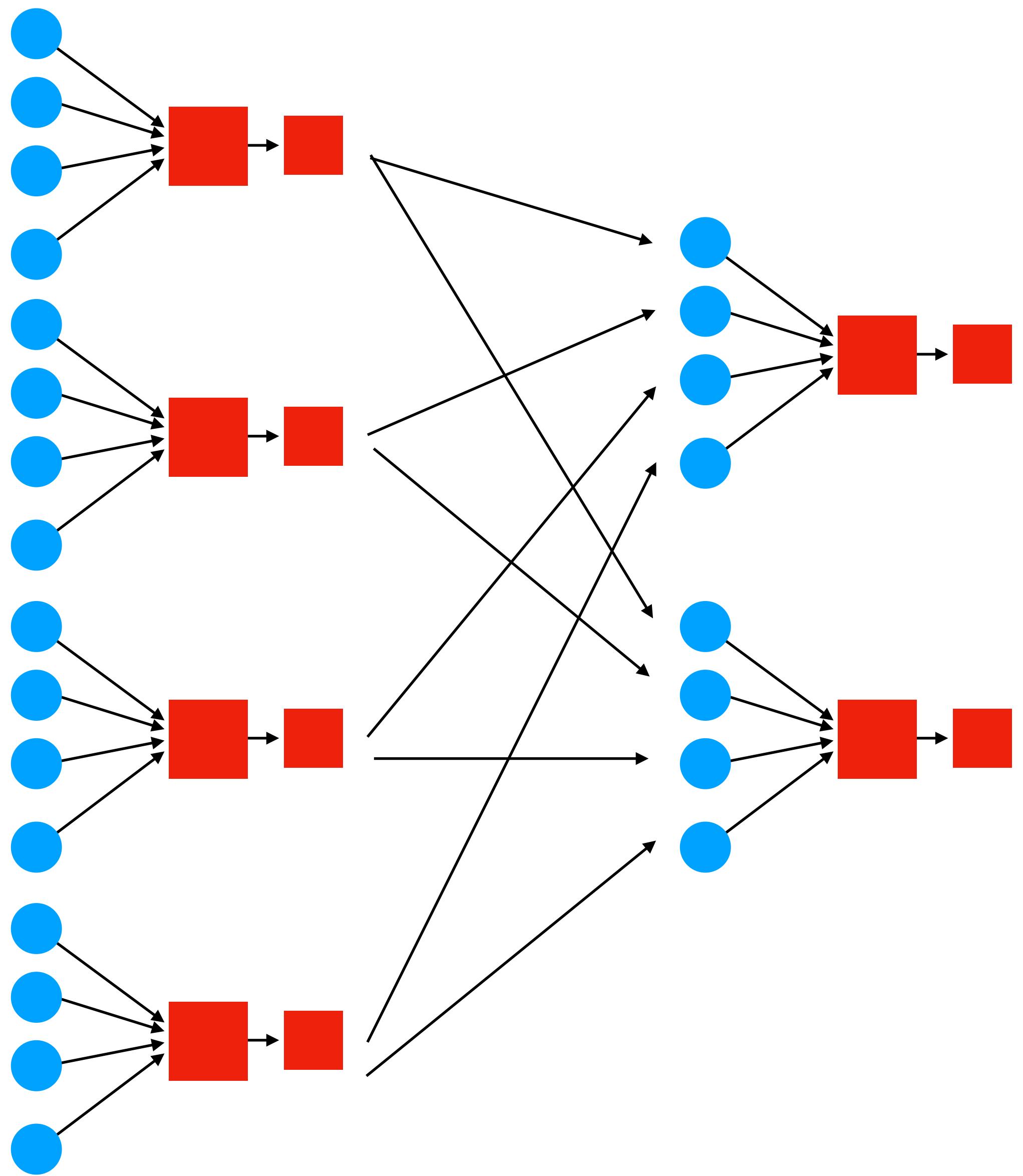
Нейронные сети



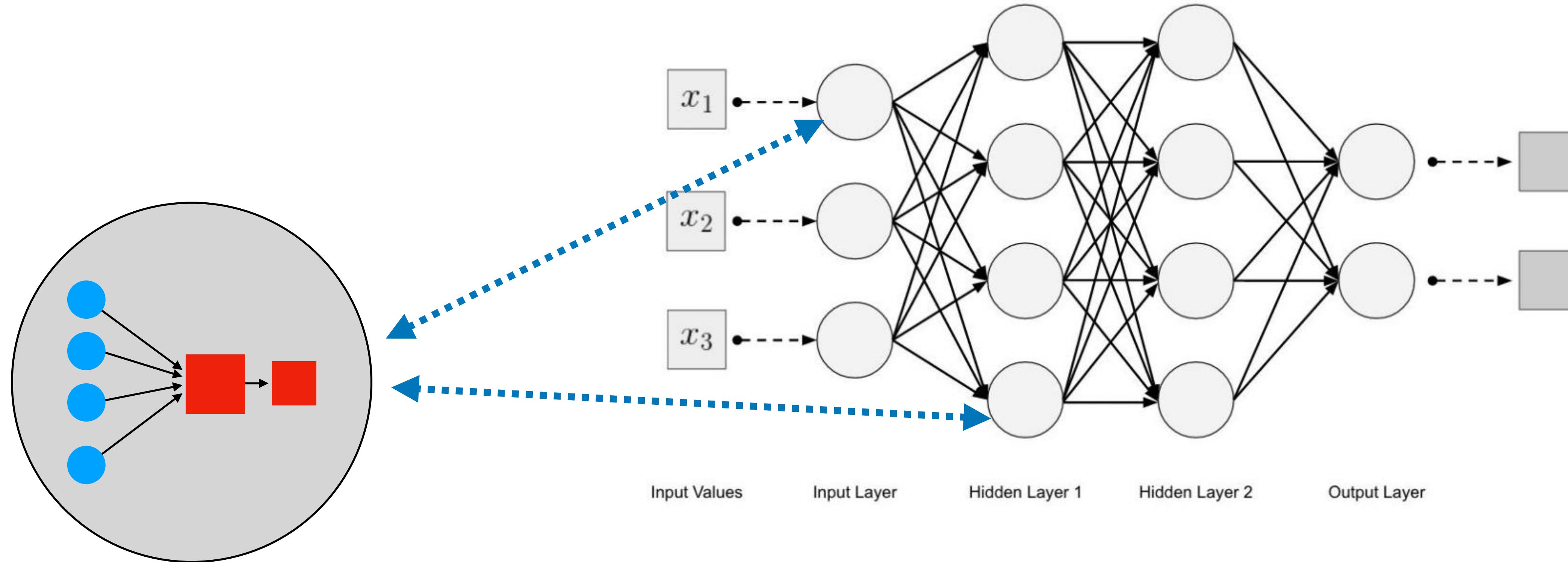
Нейронные сети



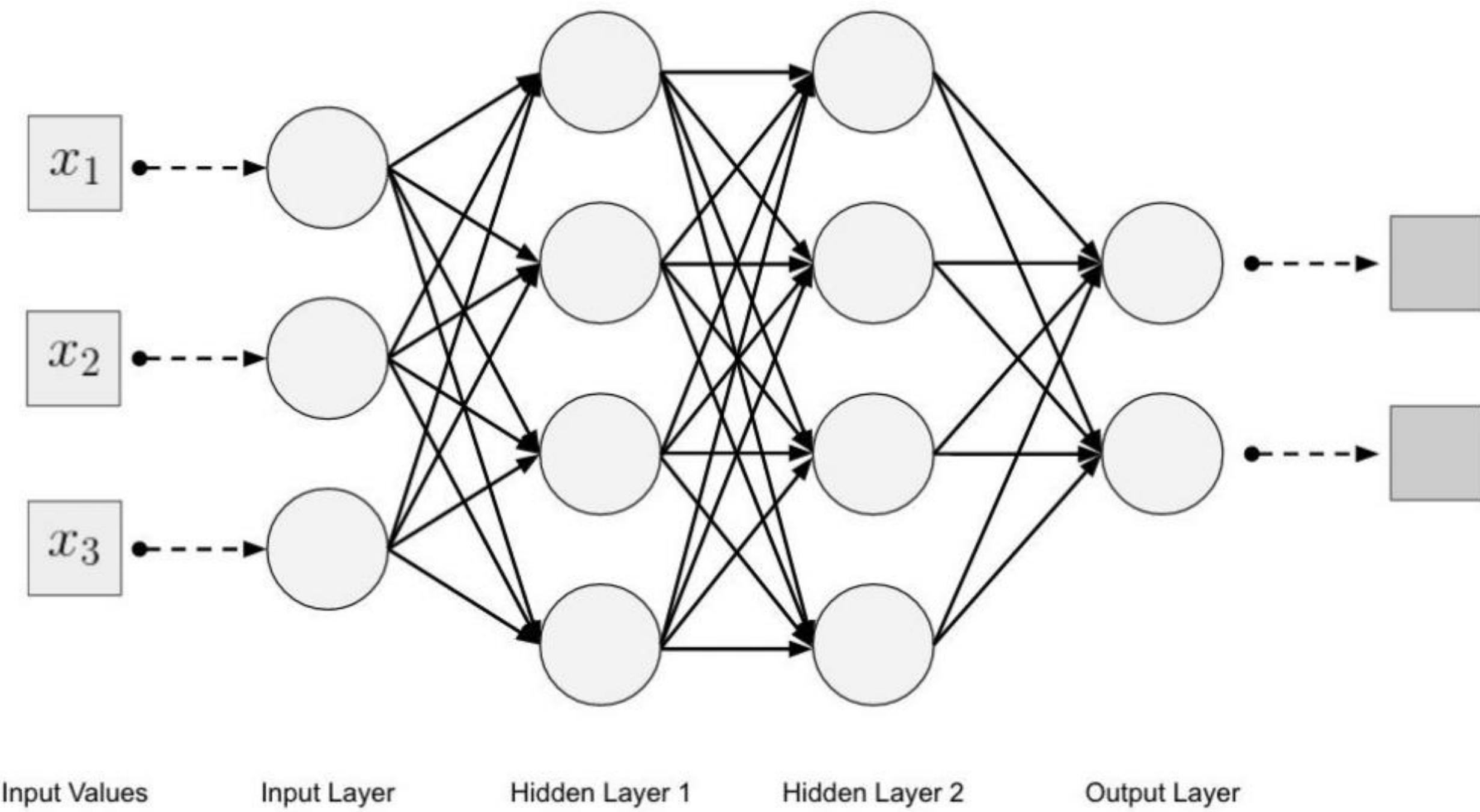
Нейронные сети



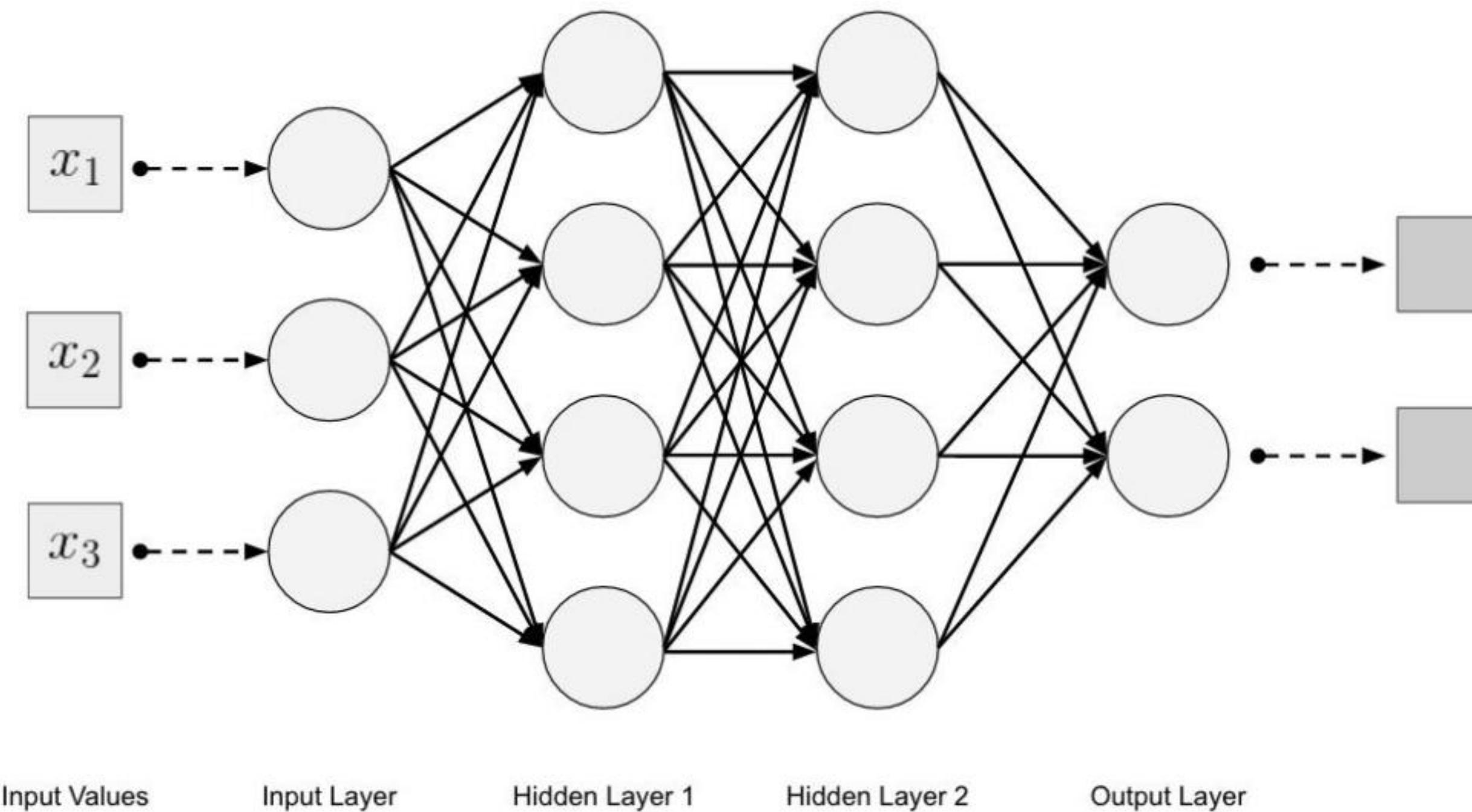
Нейронные сети



Нейронные сети



Нейронные сети



Утверждения:

Любая булева функция представима в виде нейронной сети с одним скрытым слоем с нелинейной функцией активации нейрона (но может потребоваться экспоненциально много нейронов в скрытом слое).

Любая непрерывная и ограниченная функция может быть сколь угодно точно аппроксимирована нейронной сетью с одним скрытым слоем с нелинейной функцией активации нейрона.

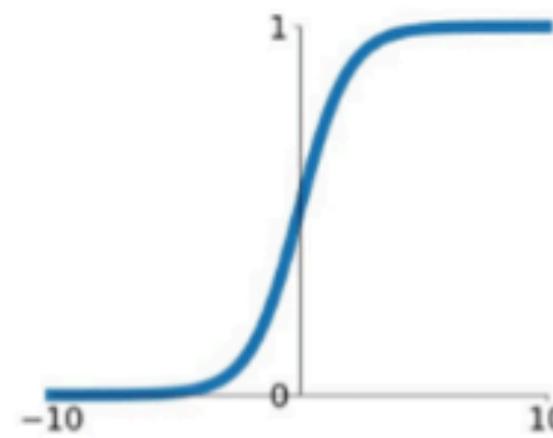
Любая функция может быть сколь угодно точно аппроксимирована нейронной сетью с двумя скрытыми слоями с нелинейной функцией активации нейрона.

Обучение нейронных сетей

ФУНКЦИИ АКТИВАЦИИ

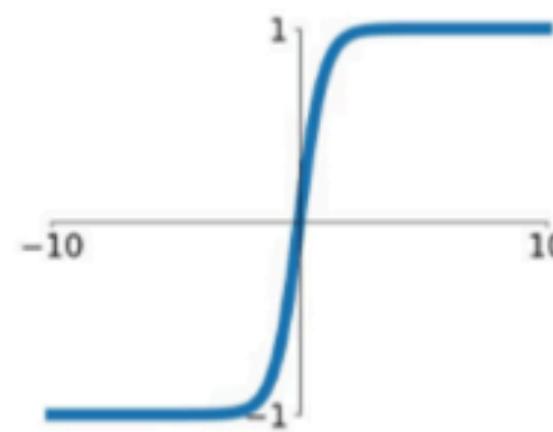
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



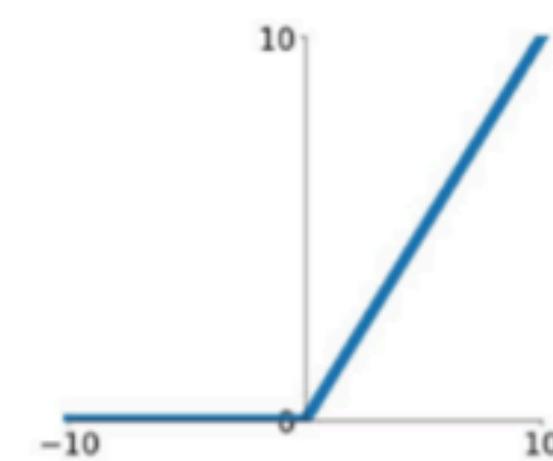
tanh

$$\tanh(x)$$



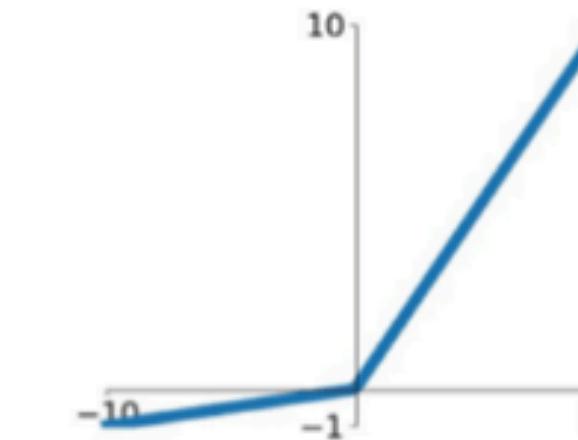
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

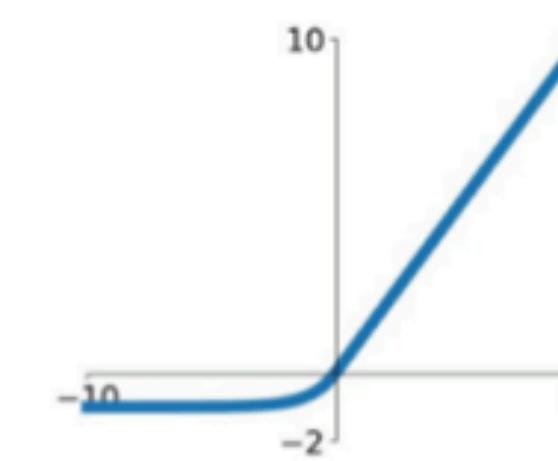


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



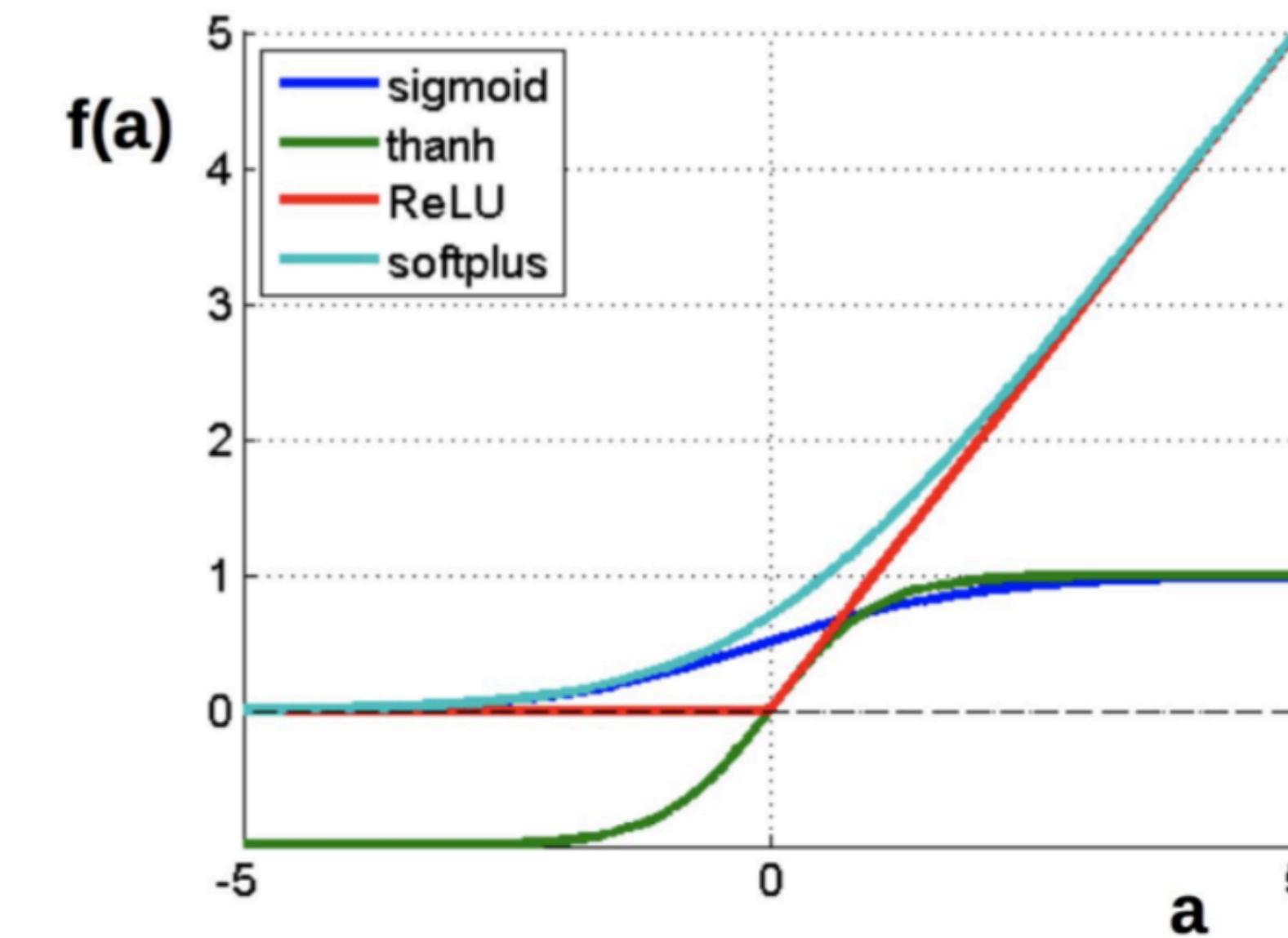
ФУНКЦИИ АКТИВАЦИИ

$$f(a) = \frac{1}{1 + e^a}$$

$$f(a) = \tanh(a)$$

$$f(a) = \max(0, a)$$

$$f(a) = \log(1 + e^a)$$

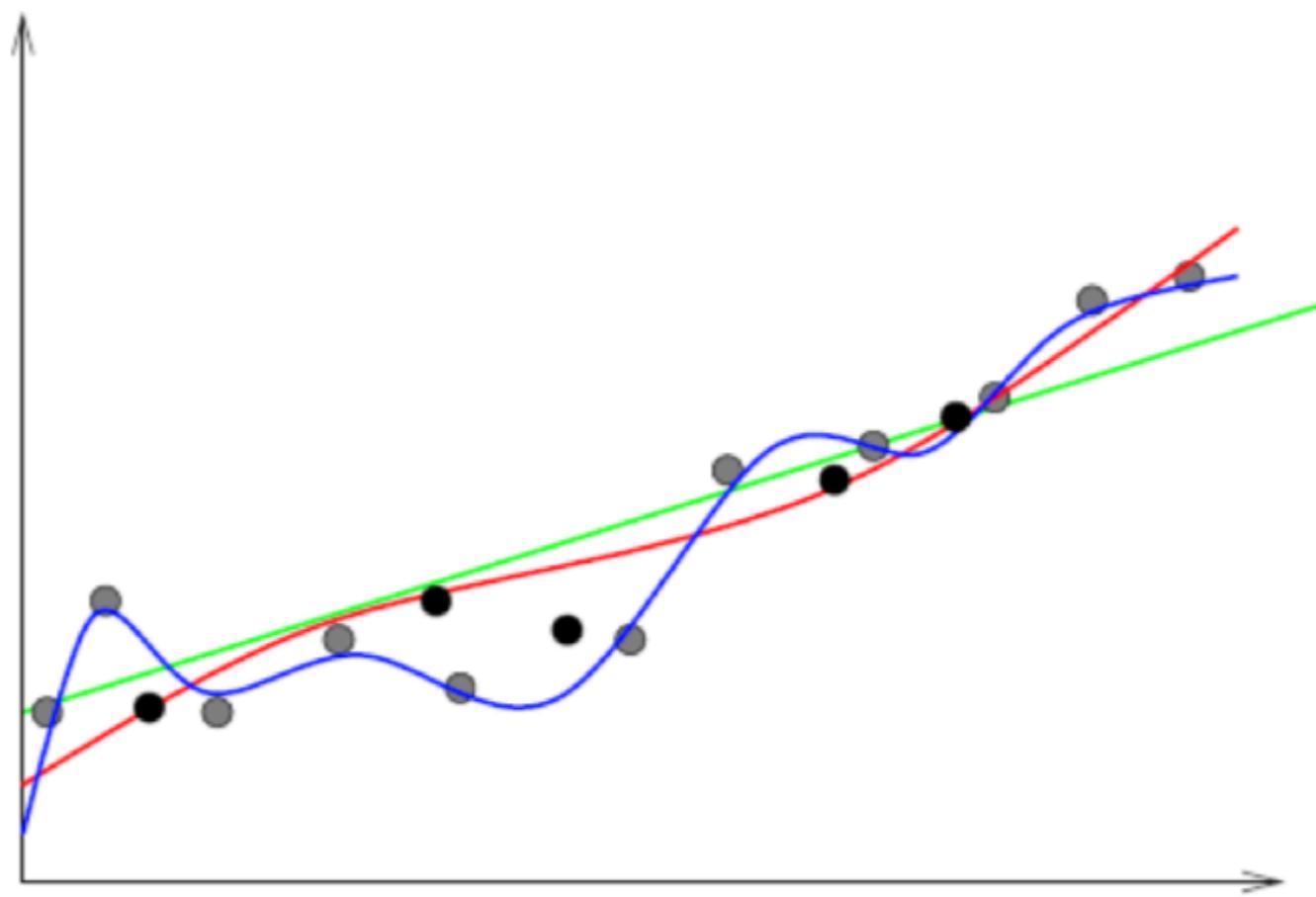


Interactive playground

playground.tensorflow.org/

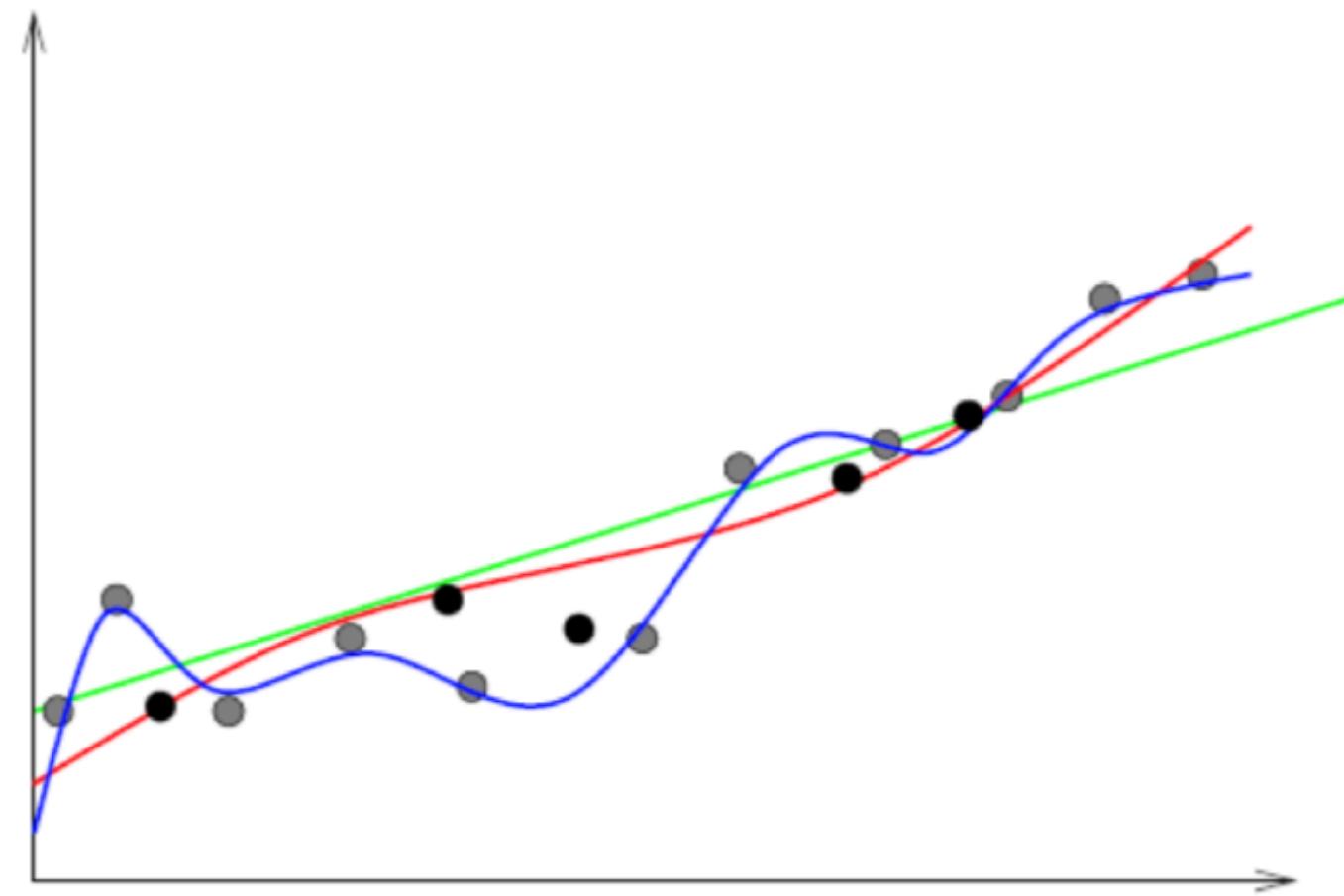
Последний слой нейронной сети

Восстановление регрессии



Последний слой нейронной сети

Восстановление регрессии



Выход сети:

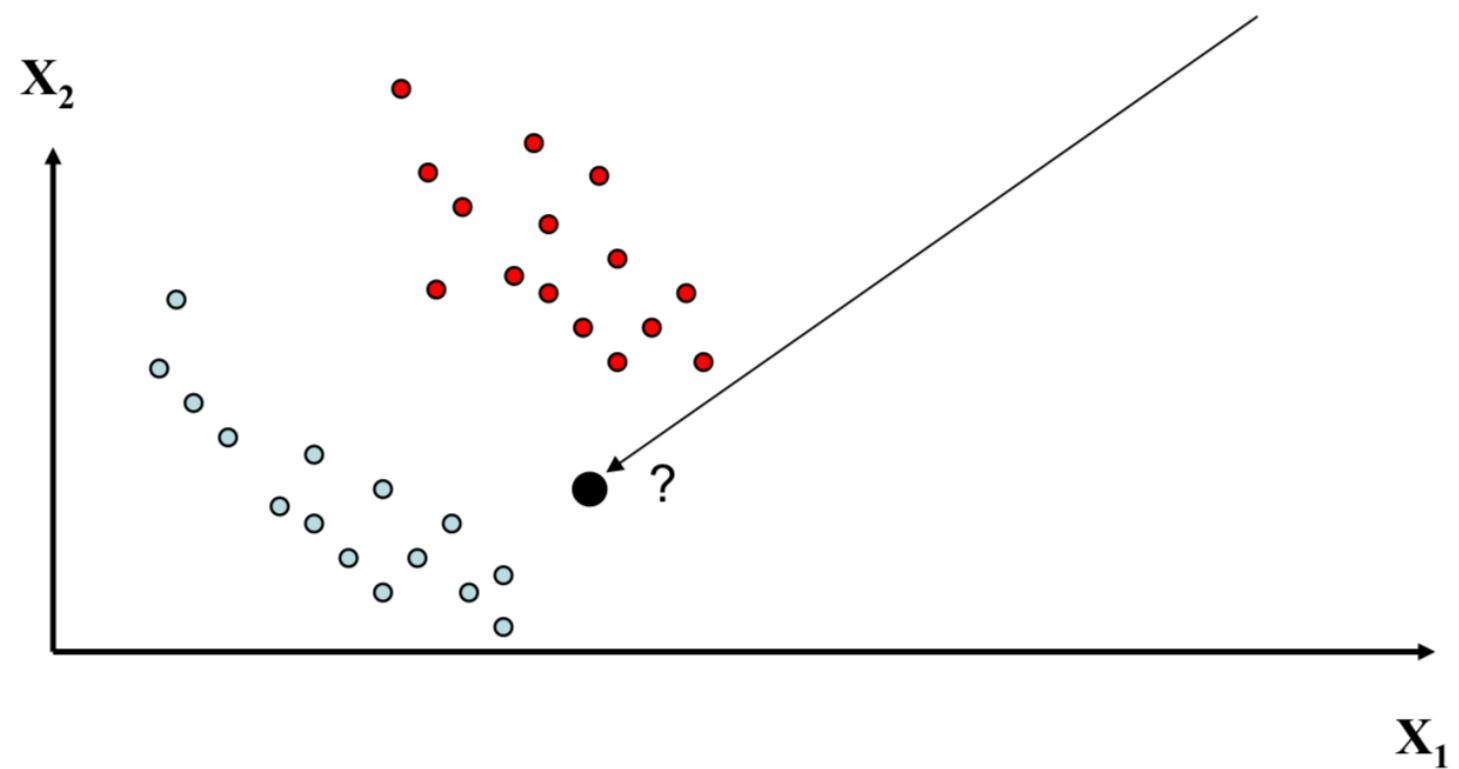
Linear

$$\sigma(z) = \sum \omega x$$

$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^l (f(x_i) - y_i)^2$$

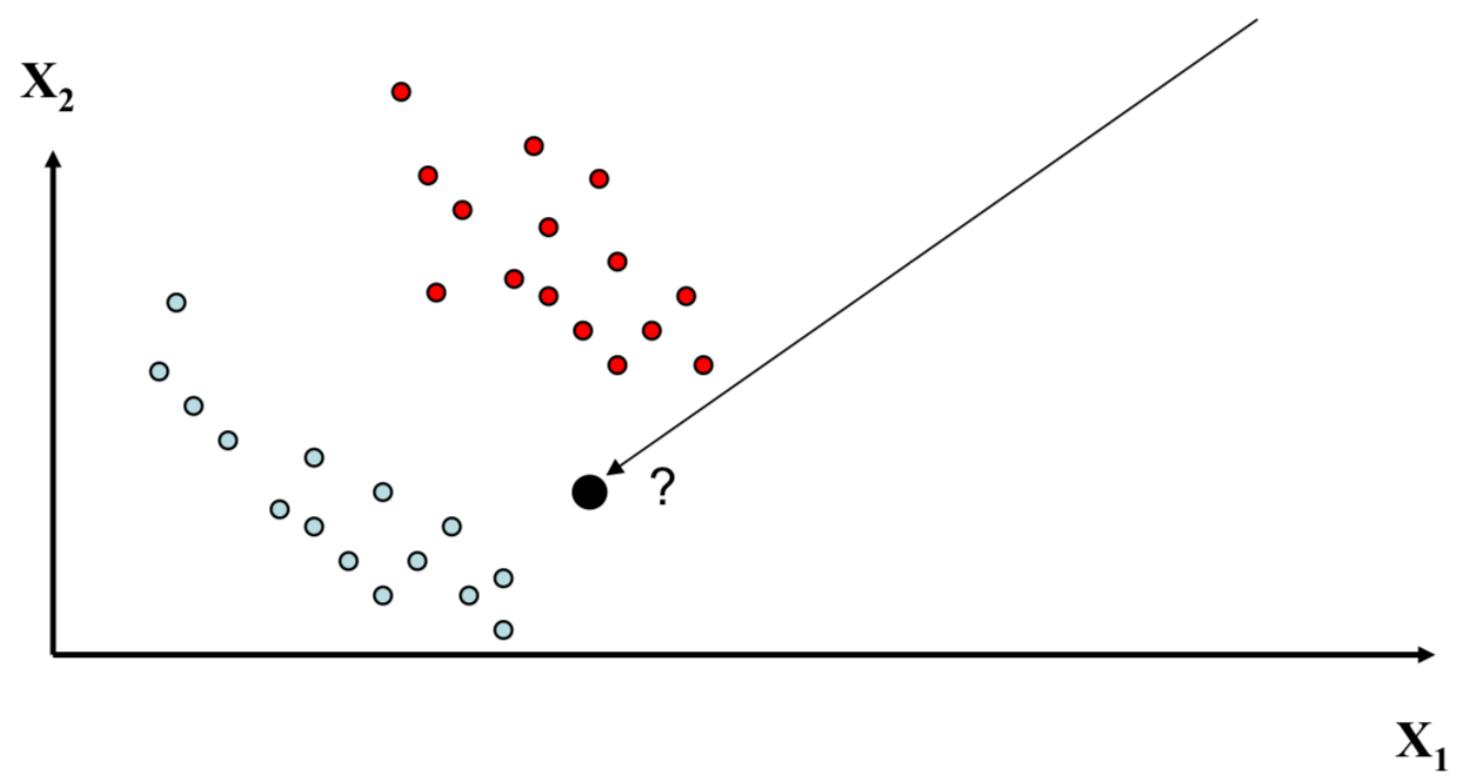
Последний слой нейронной сети

Классификация



Последний слой нейронной сети

Классификация



Выход сети:

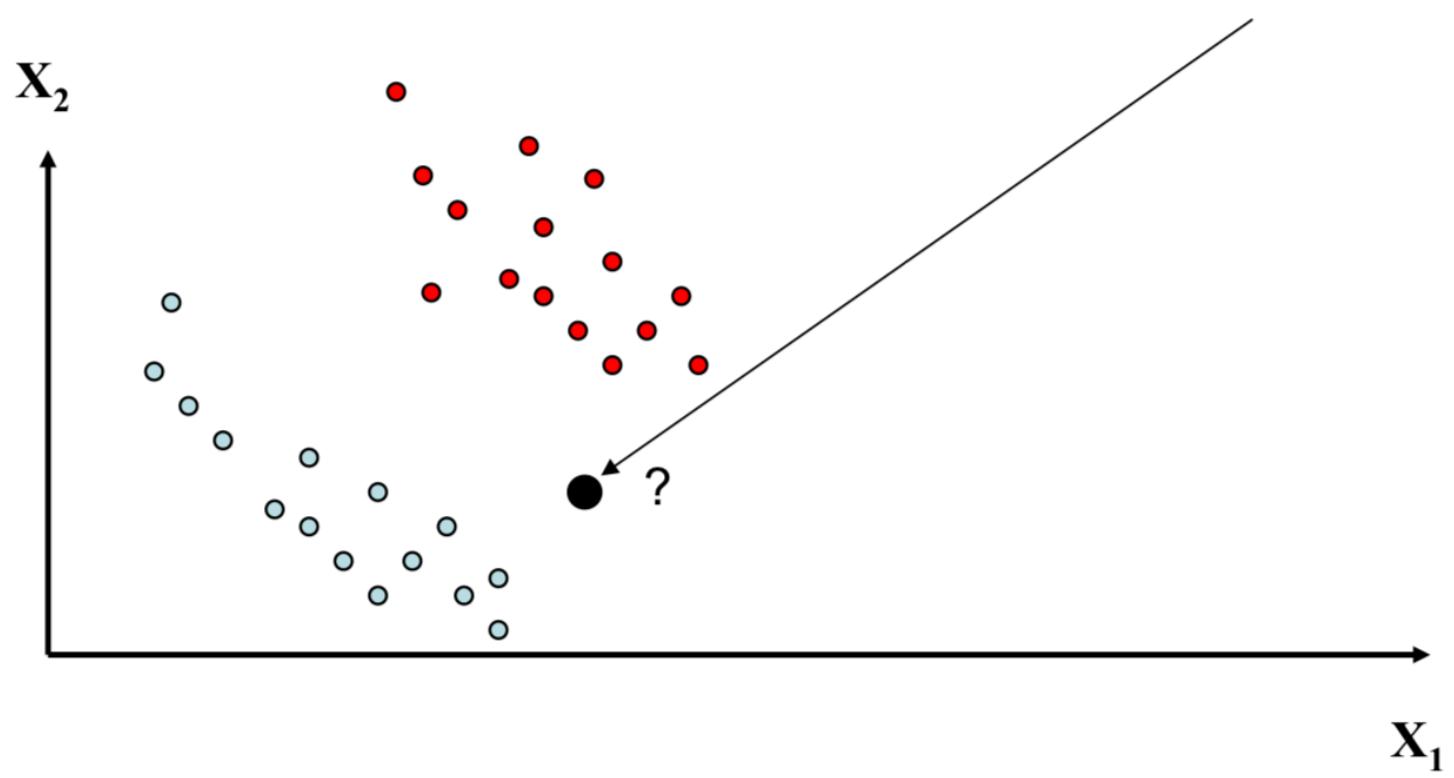
Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}} \in [0; 1]$$

$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^l -y_i \log a_i - (1 - y_i) \log(1 - a_i)$$

Последний слой нейронной сети

Классификация

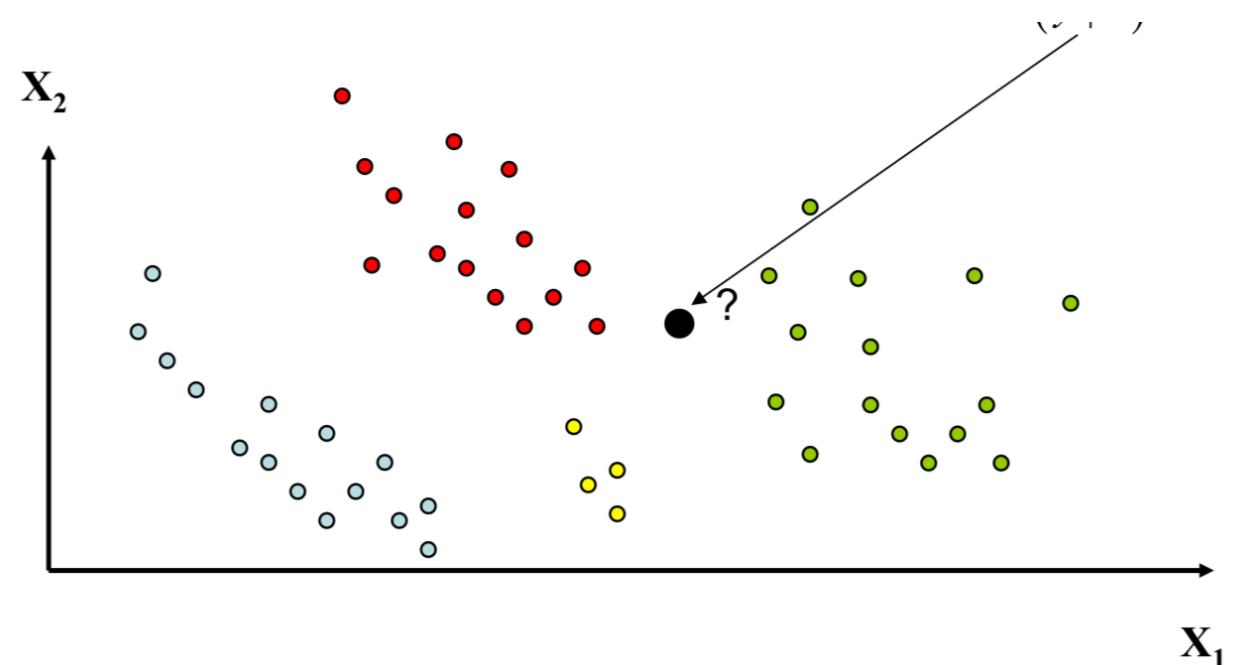


Выход сети:

Sigmoid

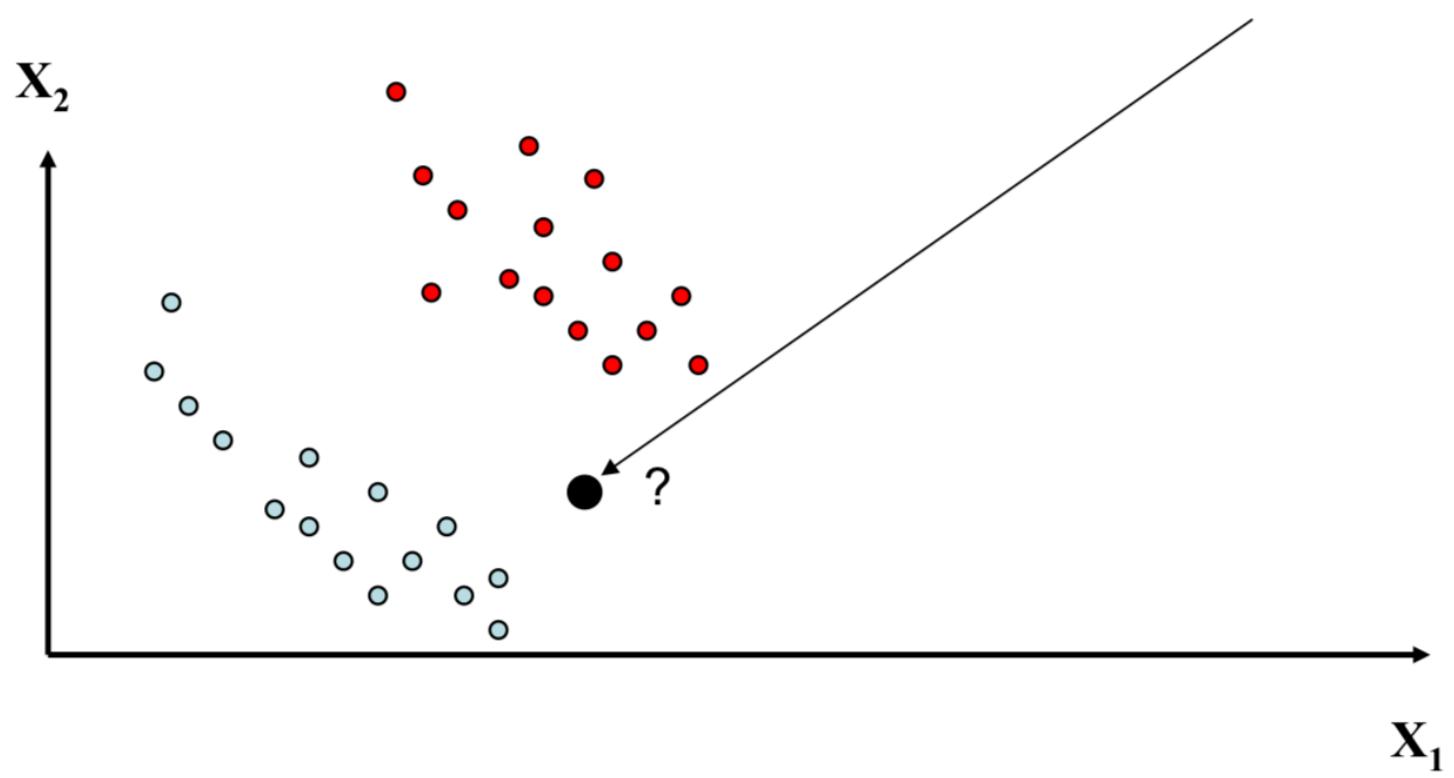
$$\sigma(z) = \frac{1}{1 + e^{-z}} \in [0; 1]$$

$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^l -y_i \log a_i - (1 - y_i) \log(1 - a_i)$$



Последний слой нейронной сети

Классификация

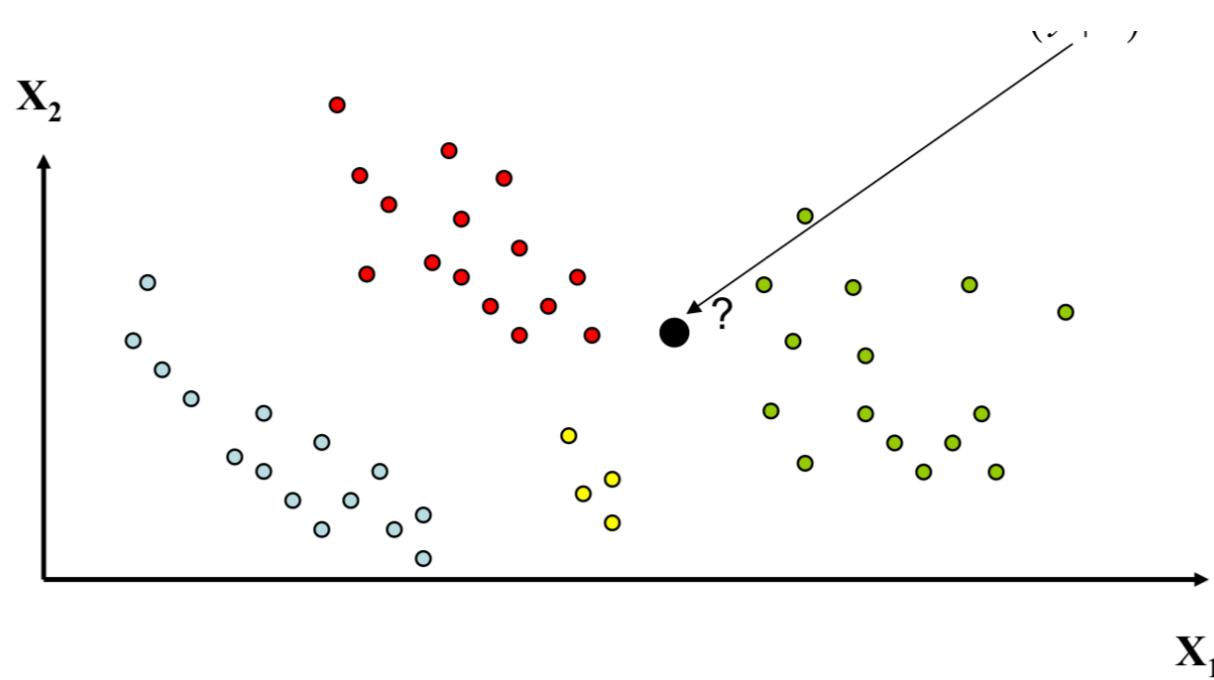


Выход сети:

Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}} \in [0; 1]$$

$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^l -y_i \log a_i - (1 - y_i) \log(1 - a_i)$$

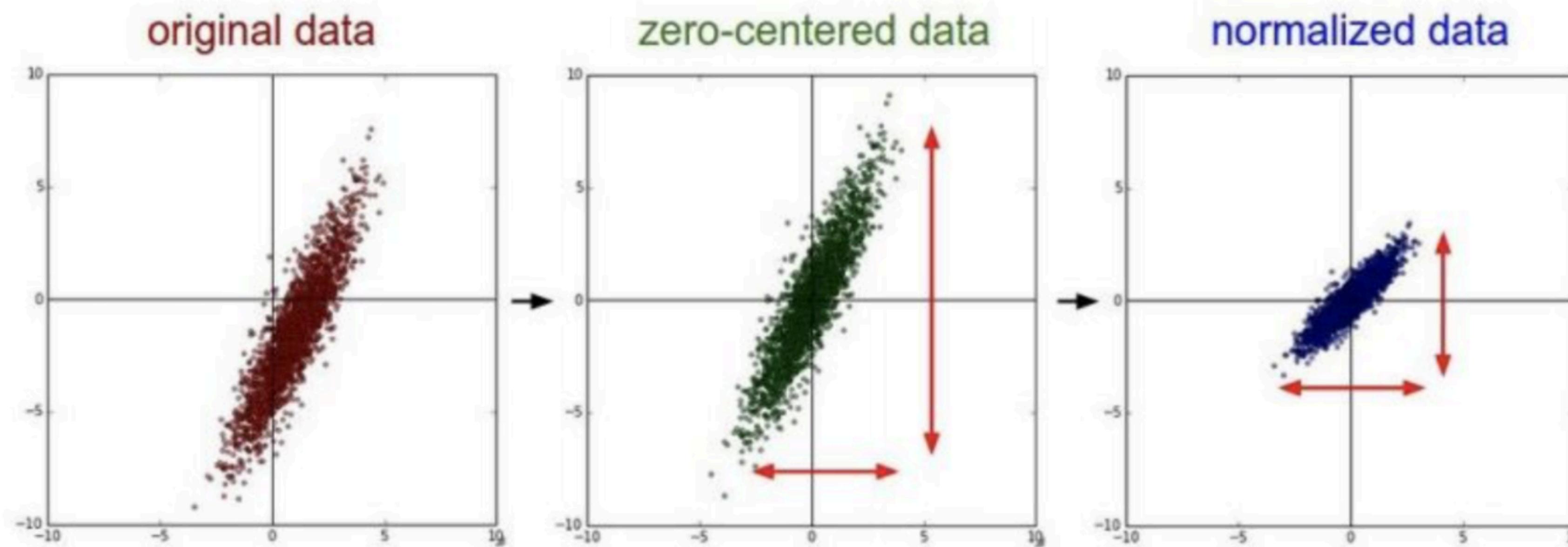


Softmax

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \in [0; 1]$$

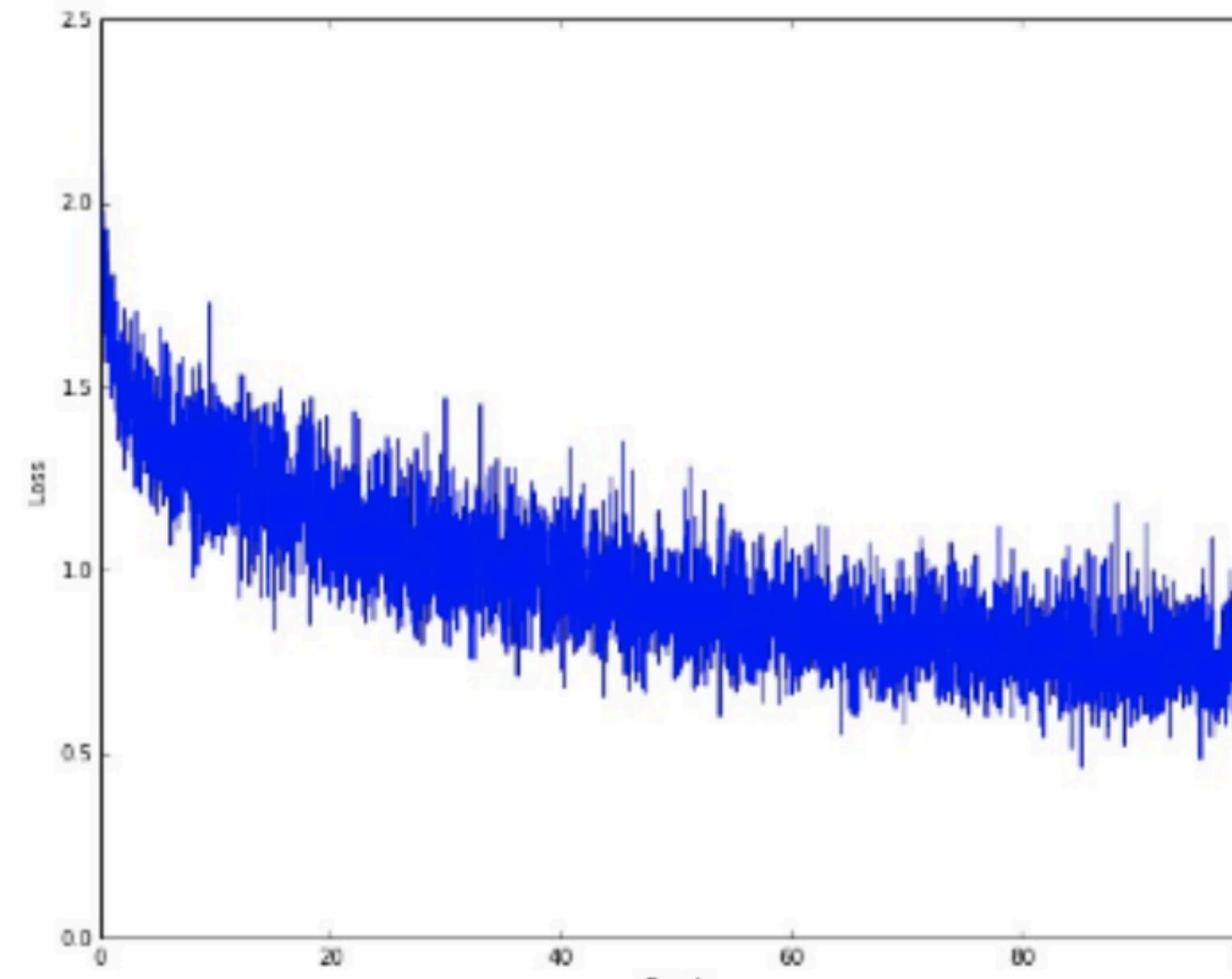
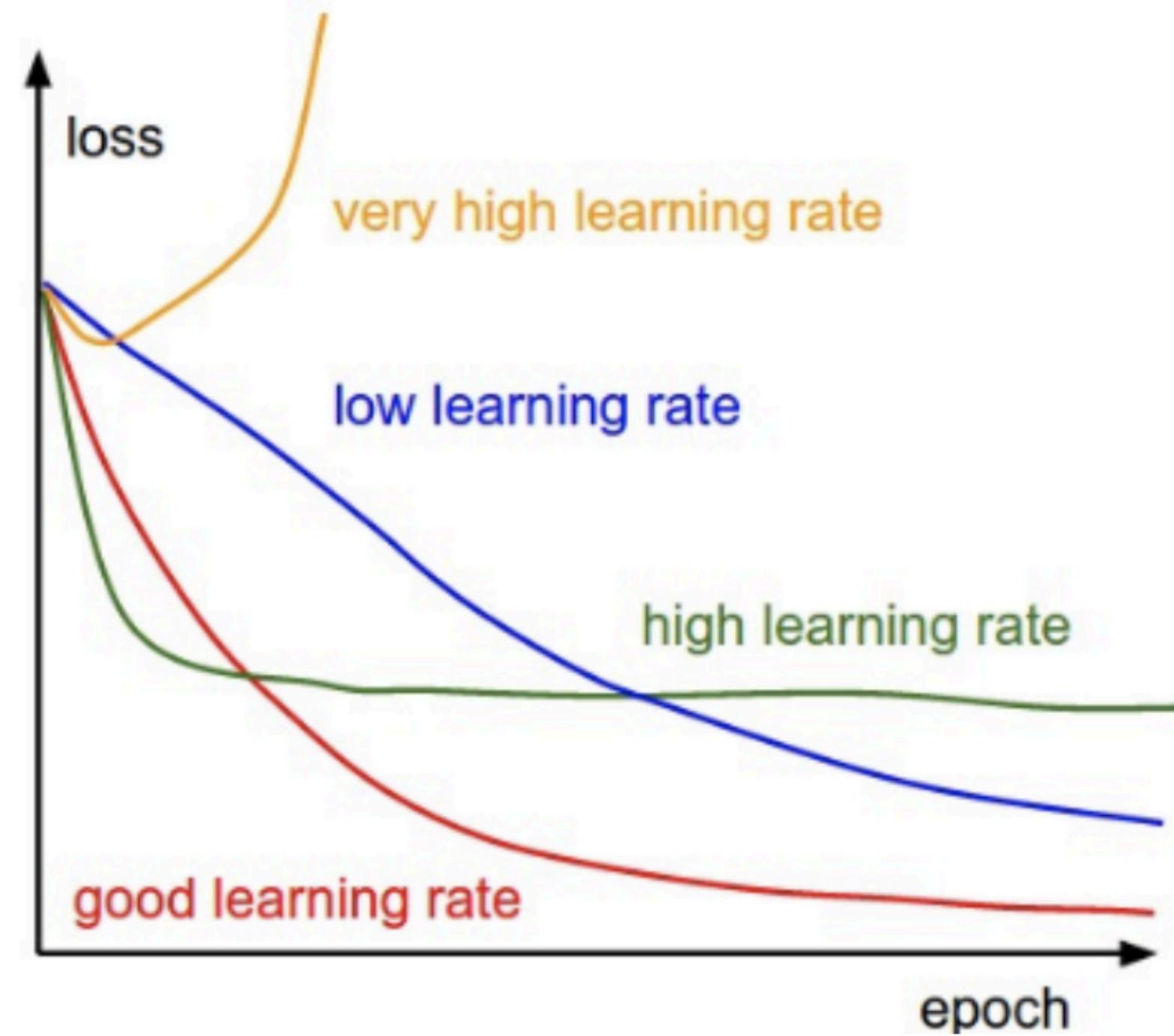
$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^l \sum_{k=1}^K y_{ik} \log a_{ik}$$

Нормализация данных



Обновление весов

$$x_{t+1} = x_t - \text{learning rate} \cdot dx$$

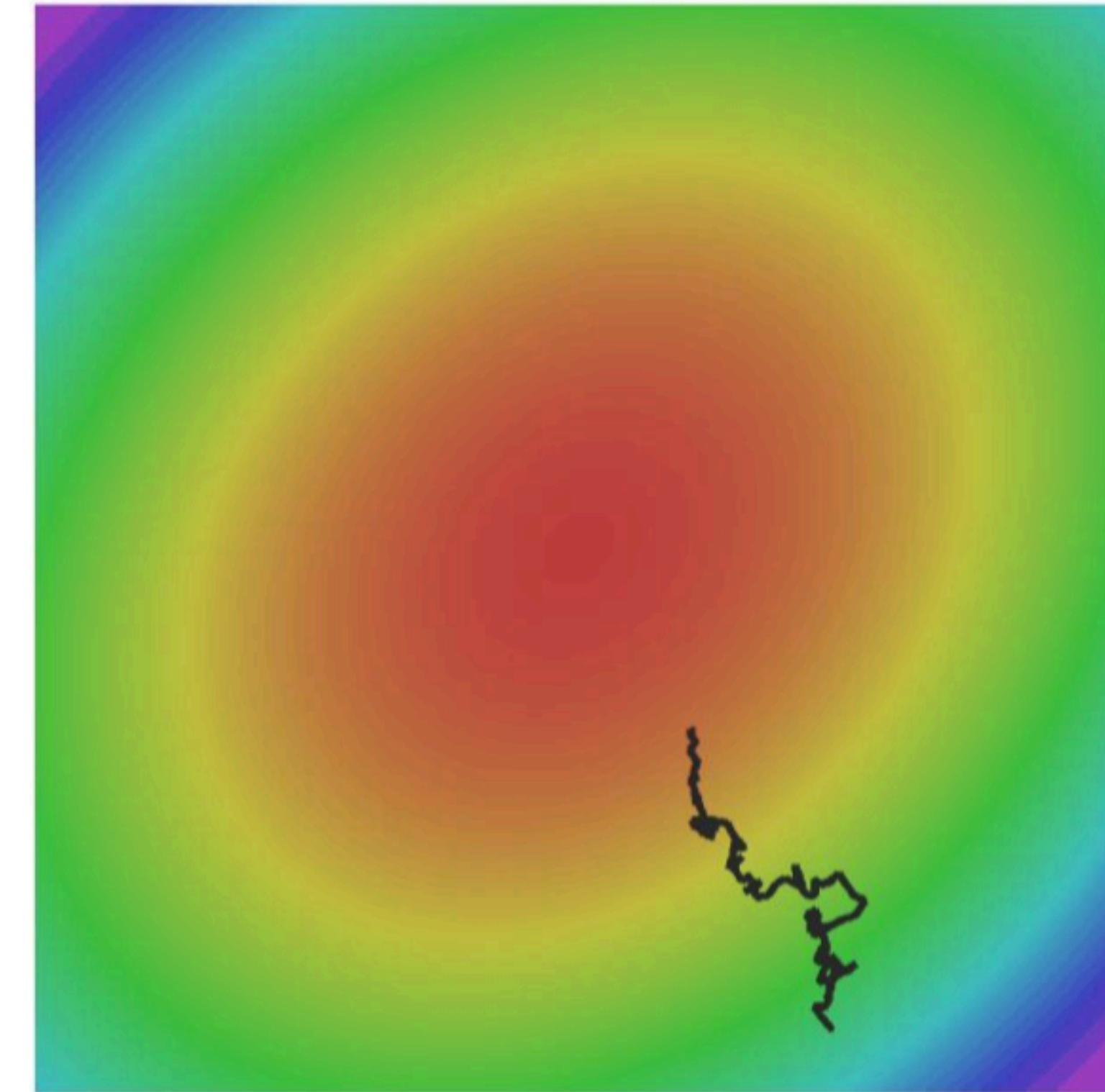


Обновление весов

Our gradients come from minibatches so they can be noisy!

$$L(W) = \frac{1}{N} \sum_{i=1}^N L_i(x_i, y_i, W)$$

$$\nabla_W L(W) = \frac{1}{N} \sum_{i=1}^N \nabla_W L_i(x_i, y_i, W)$$



Обновление весов

SGD

$$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

```
while True:  
    dx = compute_gradient(x)  
    x += learning_rate * dx
```

SGD+Momentum

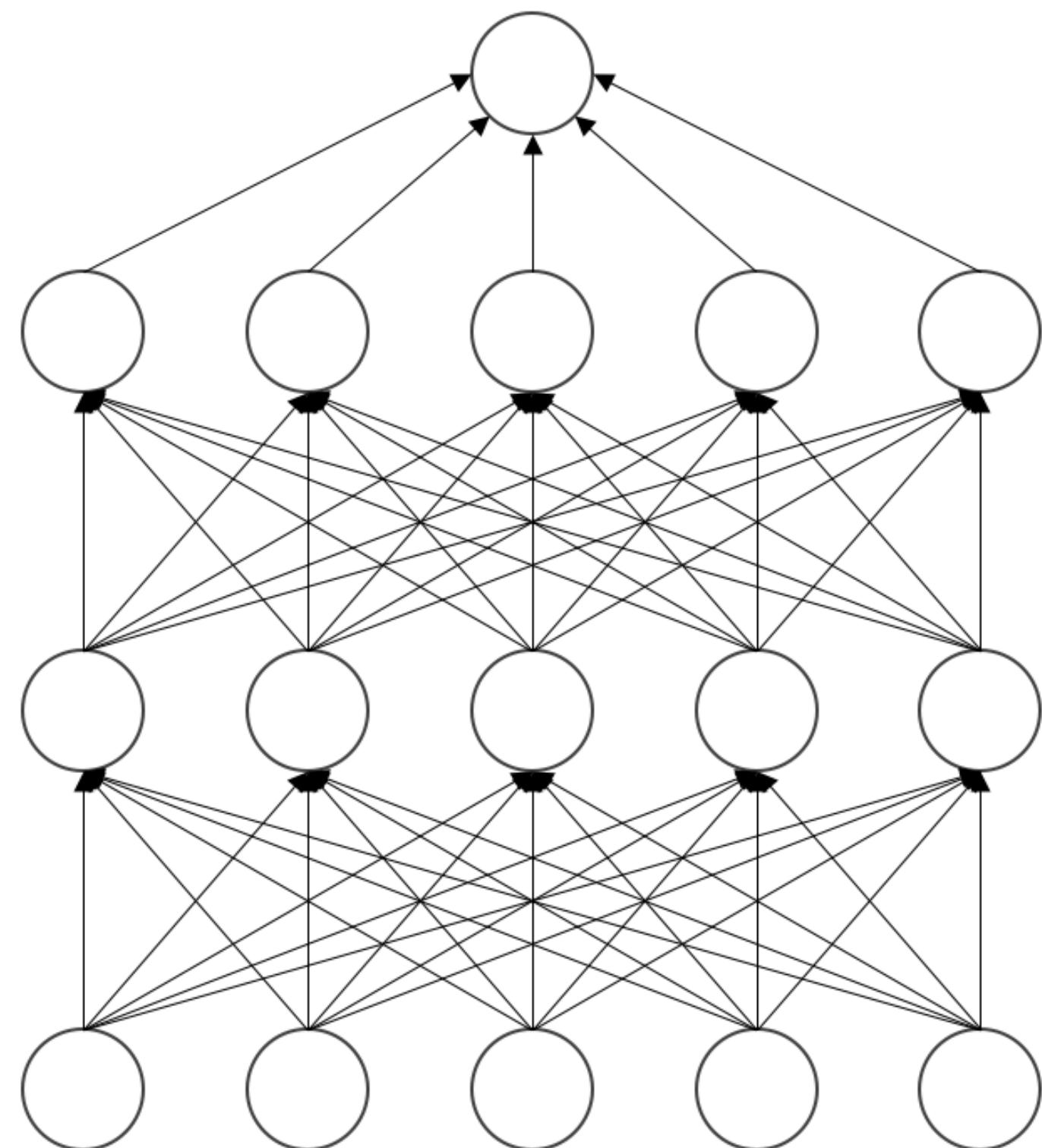
$$v_{t+1} = \rho v_t + \nabla f(x_t)$$

$$x_{t+1} = x_t - \alpha v_{t+1}$$

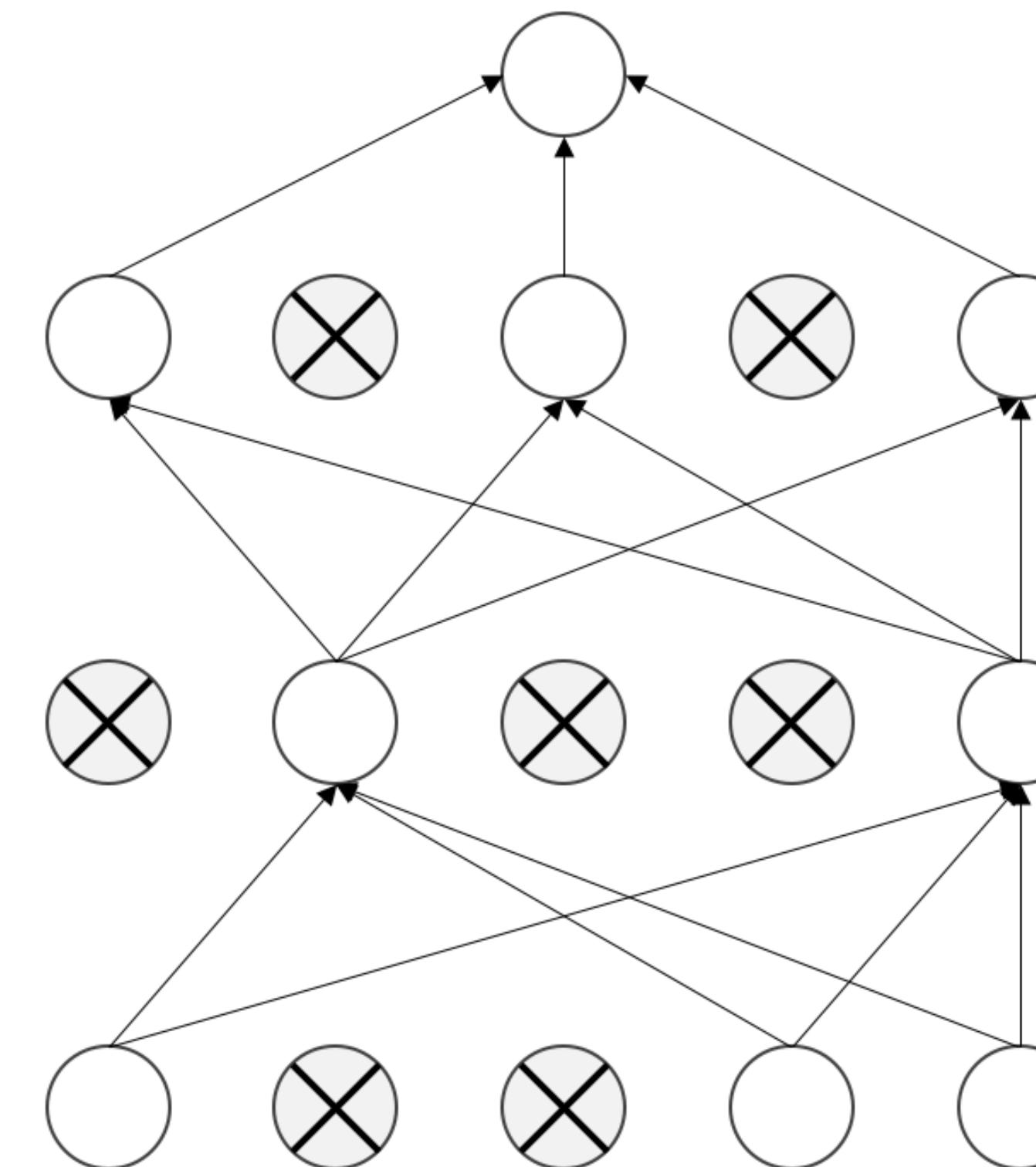
```
vx = 0  
while True:  
    dx = compute_gradient(x)  
    vx = rho * vx + dx  
    x += learning_rate * vx
```

- Build up “velocity” as a running mean of gradients
- Rho gives “friction”; typically rho=0.9 or 0.99

Регуляризация: DropOut

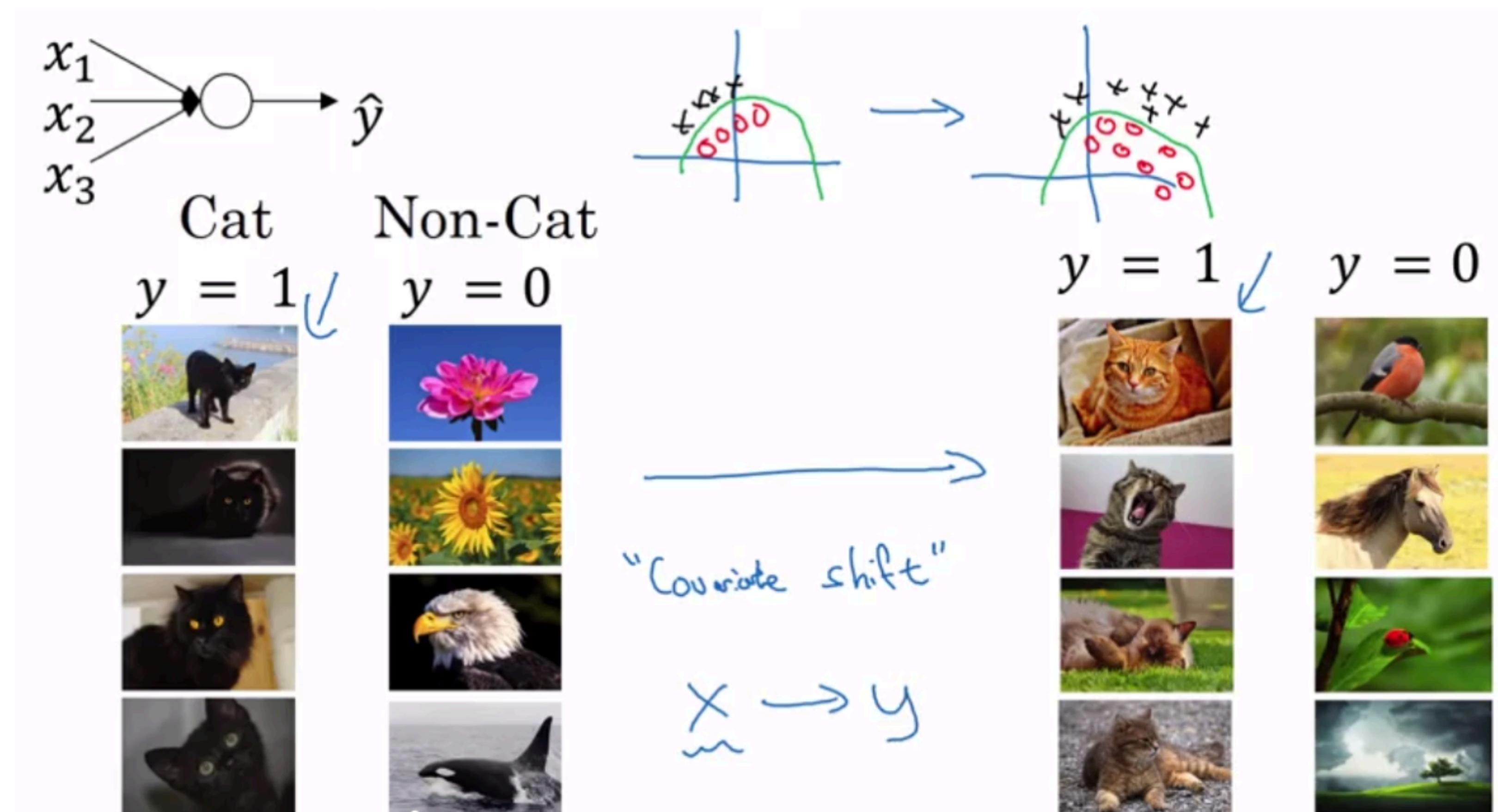


Standard Neural Net



After applying dropout

Регуляризация: Batch Normalisation



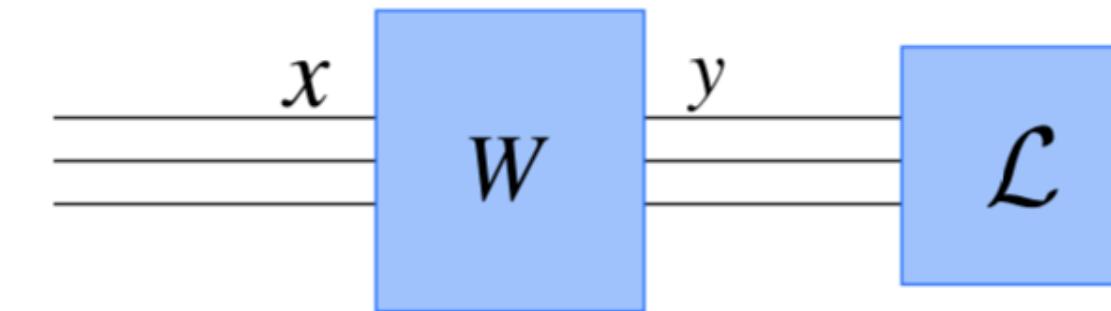
- Оригинальная статья: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
- Andrew Ng: Why Does Batch Norm Work?
- How Does Batch Normalization Help Optimization?

Регуляризация: Batch Normalisation

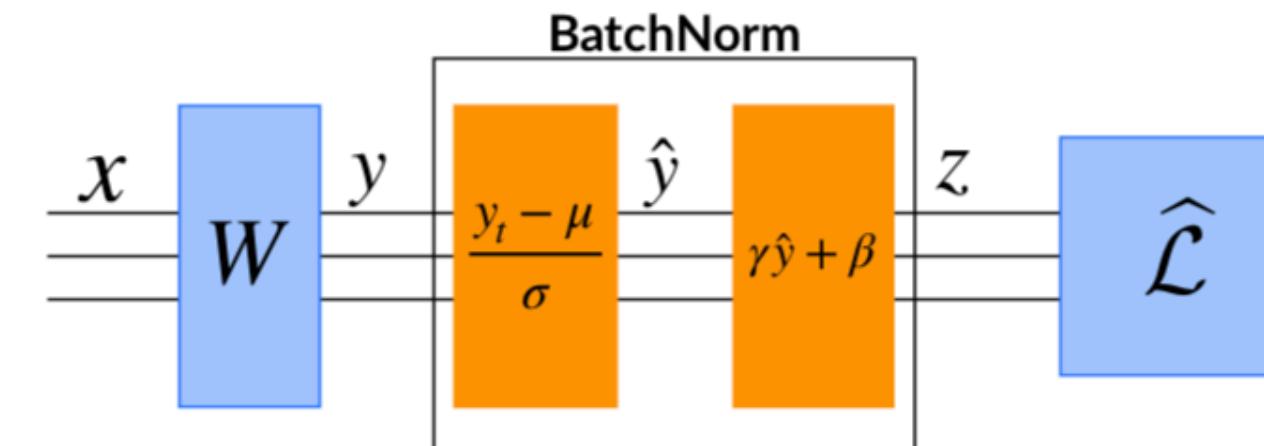
Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\begin{aligned}\mu_{\mathcal{B}} &\leftarrow \frac{1}{m} \sum_{i=1}^m x_i && // \text{mini-batch mean} \\ \sigma_{\mathcal{B}}^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 && // \text{mini-batch variance} \\ \hat{x}_i &\leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} && // \text{normalize} \\ y_i &\leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) && // \text{scale and shift}\end{aligned}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



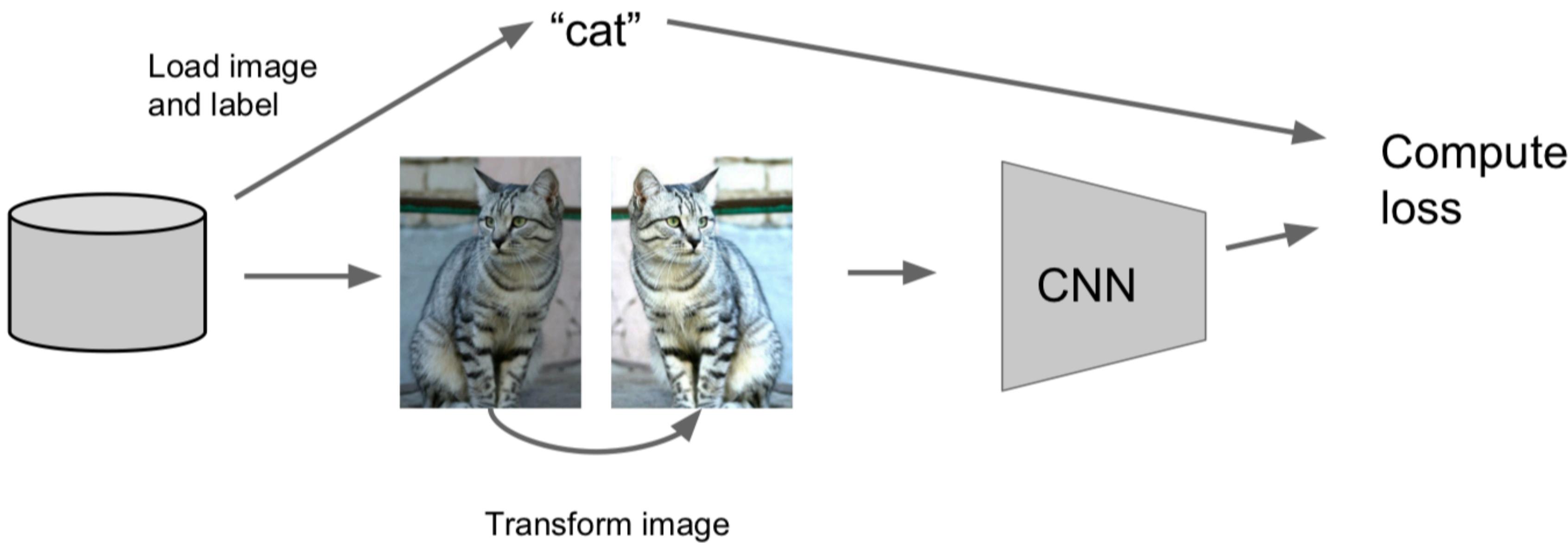
(a) Vanilla Network



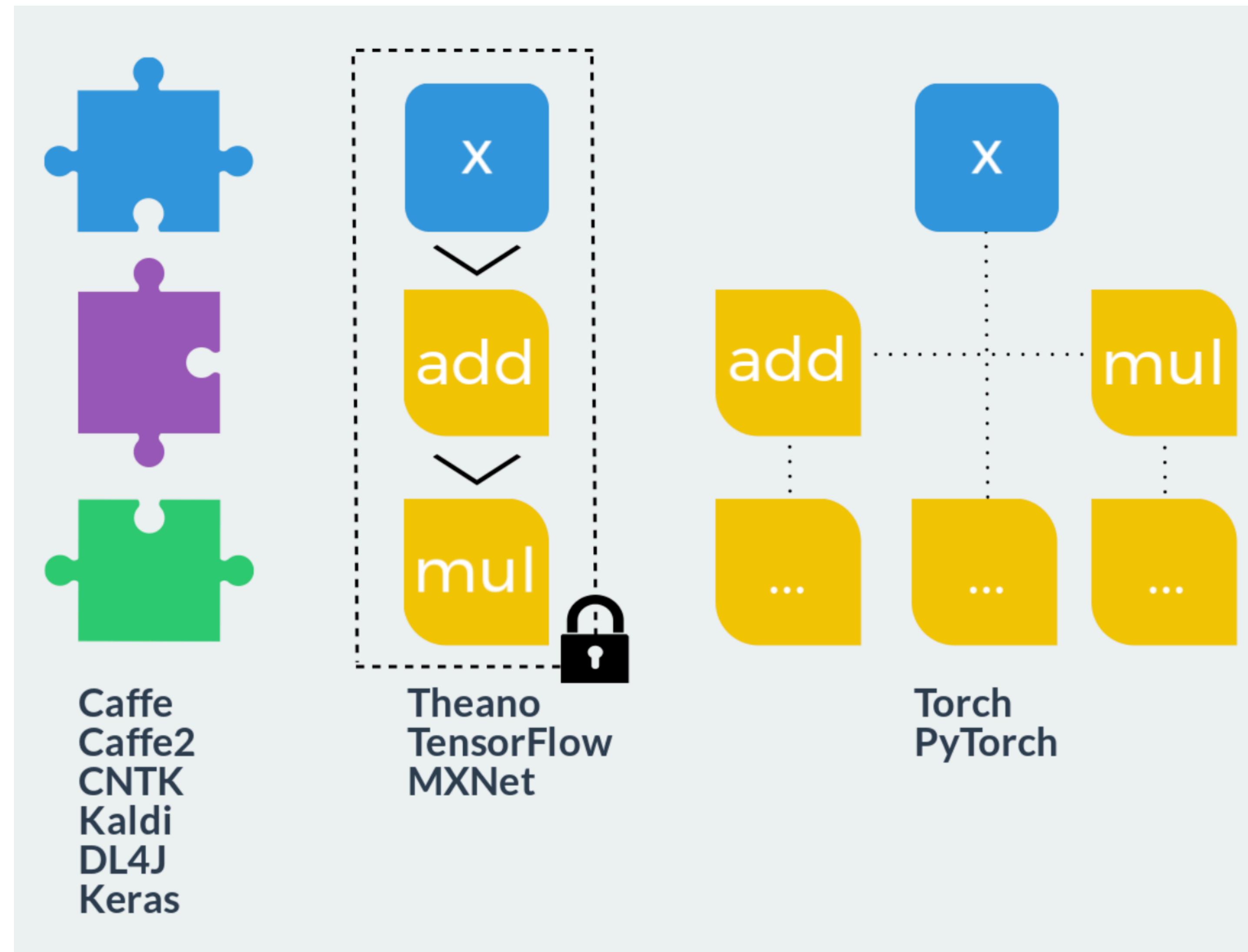
(b) Vanilla Network + BatchNorm Layer

- Оригинальная статья: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
- Andrew Ng: Why Does Batch Norm Work?
- How Does Batch Normalization Help Optimization?

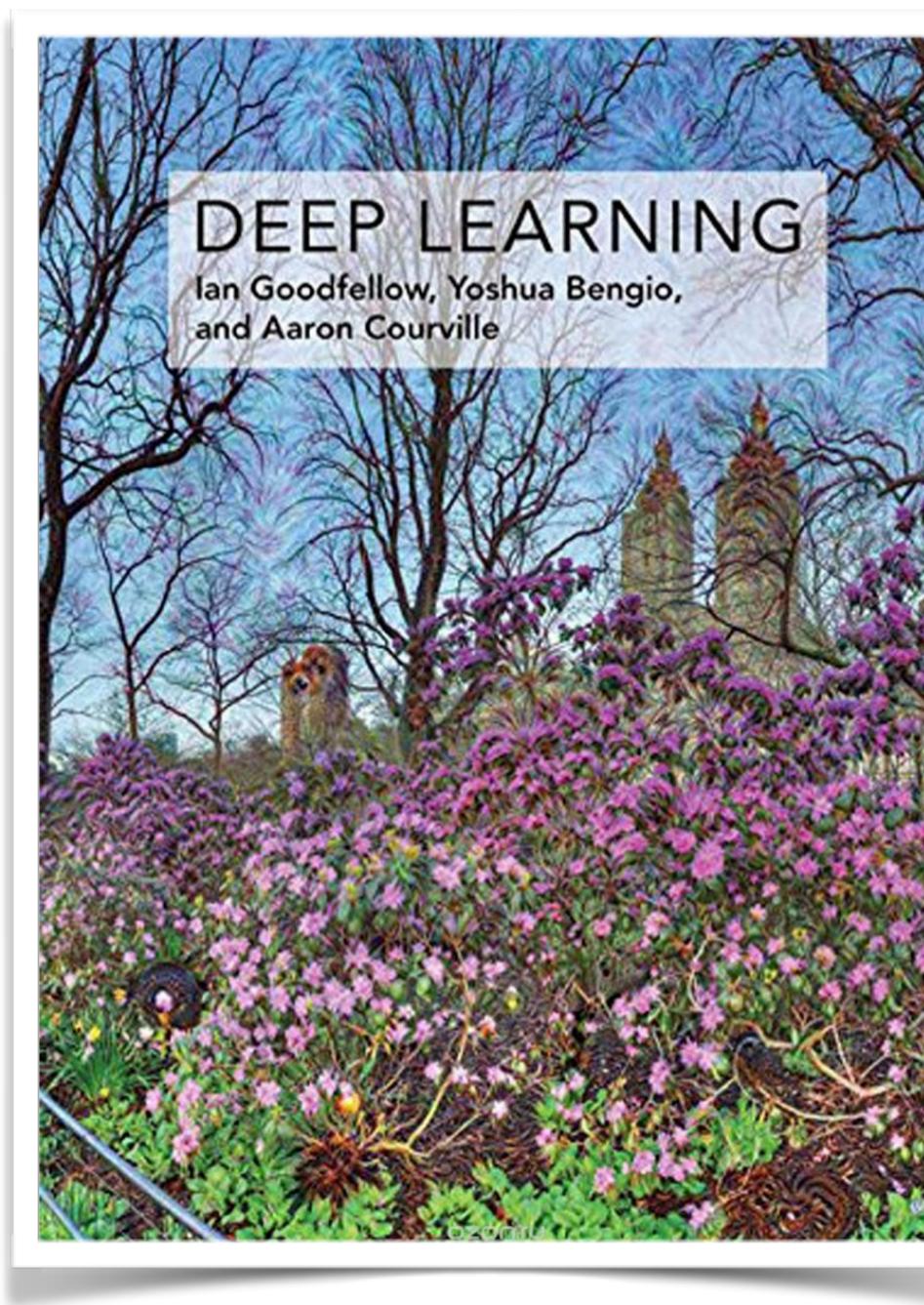
Регуляризация: Аугментация



Сравнение библиотек



Литература и ссылки



<http://www.deeplearningbook.org>

Deep Learning на пальцах

CS231n: Convolutional Neural Networks for Visual Recognition

CS224n: Natural Language Processing with Deep Learning