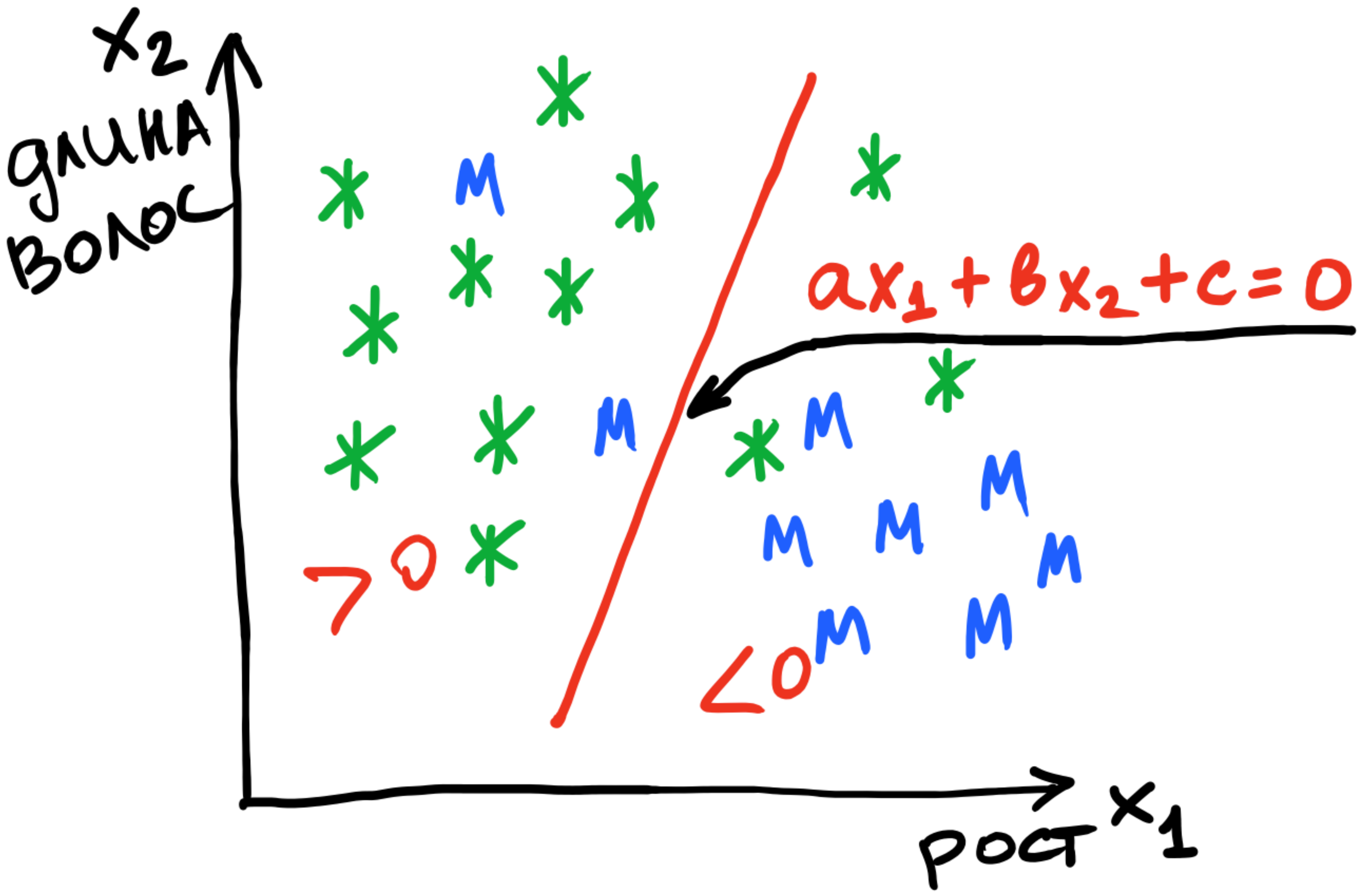
MML minor #4

Нейронные сети

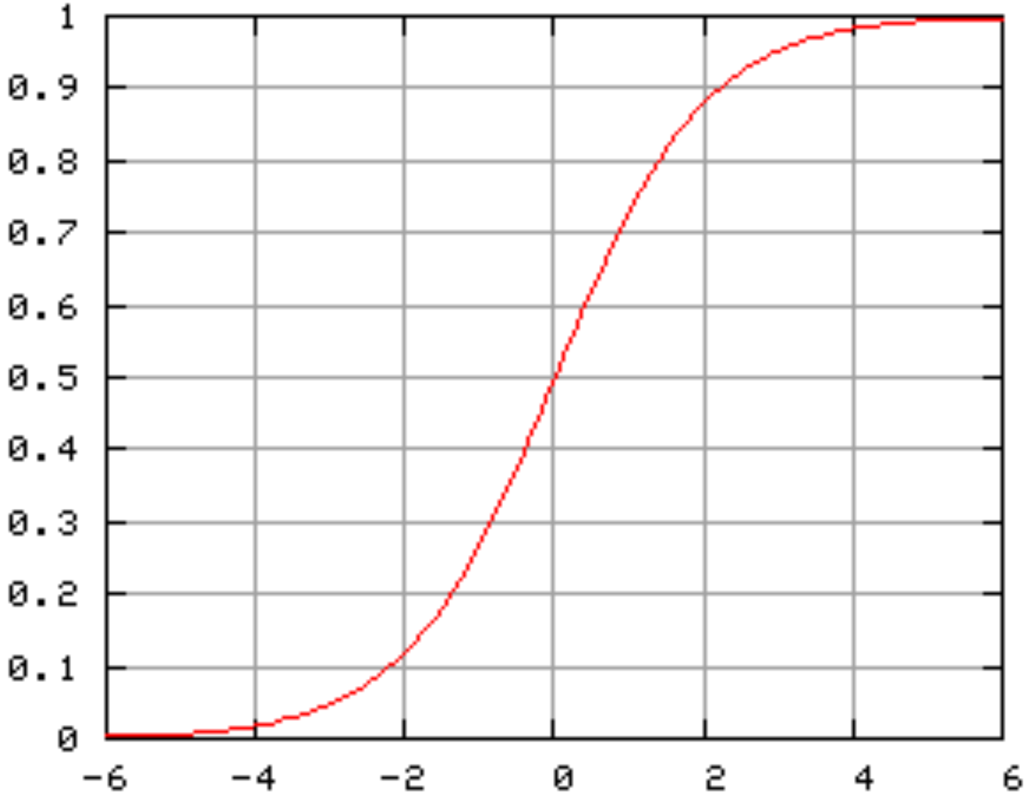
Линейная классификация



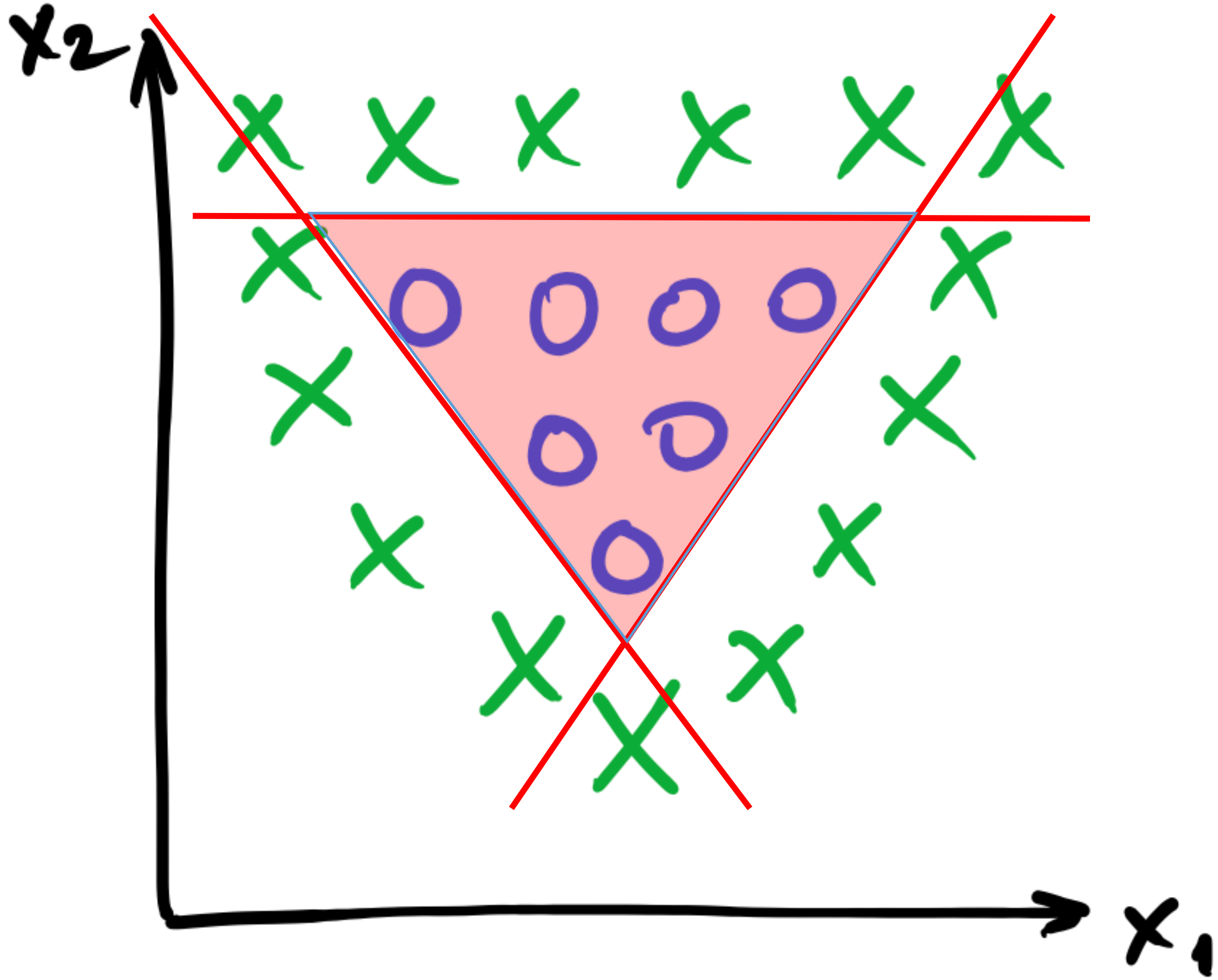
Напоминание: логистическая регрессия

Вероятность положительного класса: $y(x) = \sigma(w^T x) = \frac{1}{1+e^{-(w^T x)}}$

Логистическая функция $\sigma(x)$ взамен знака возвращает уверенность.



Если строить несколько линейных моделей?



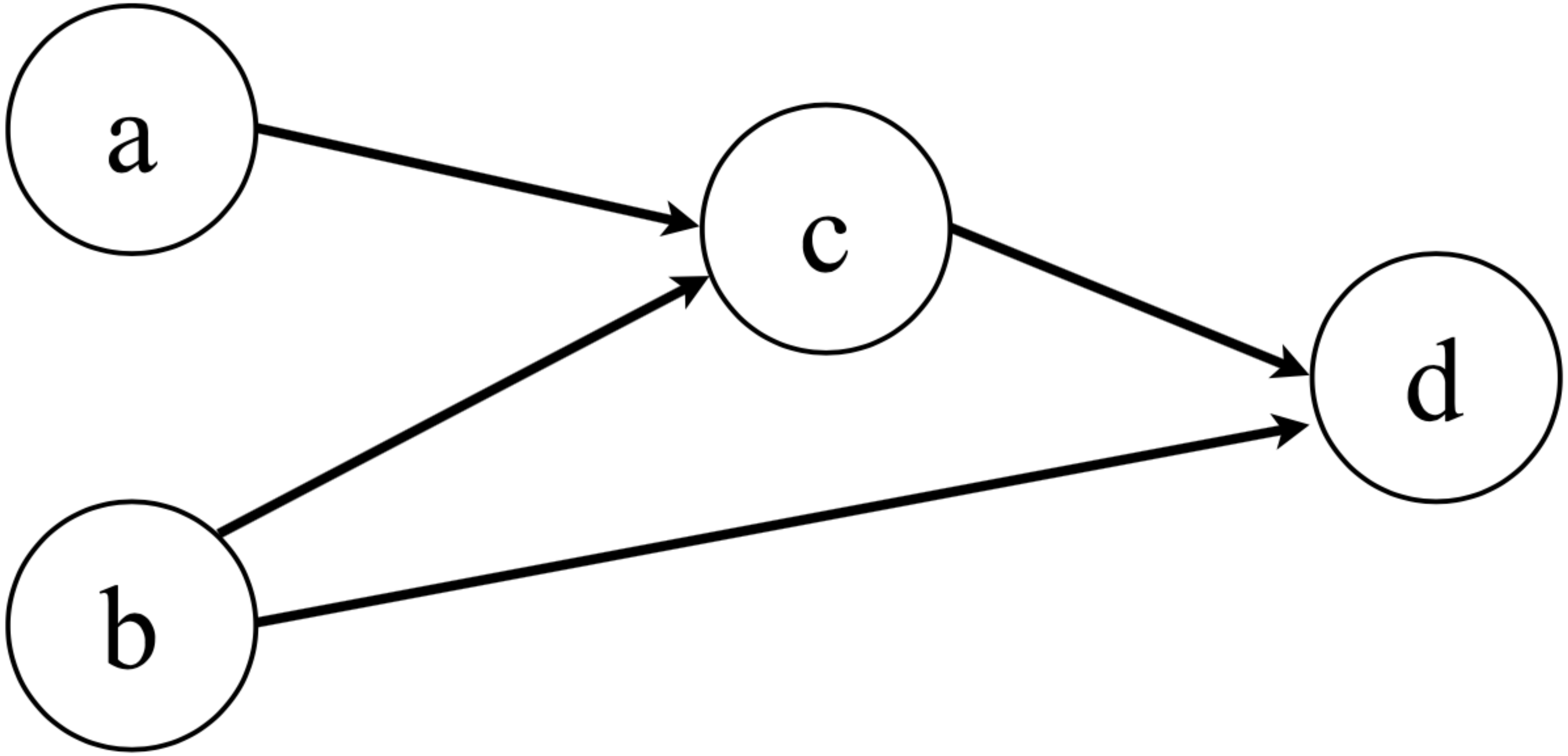
Построим три линейные модели, отвечающие за разделение в разных областях

На их предсказаниях построим финальную линейную модель

Осталось понять как…

Композиции в виде графов

$$d(c(a, b), b)$$



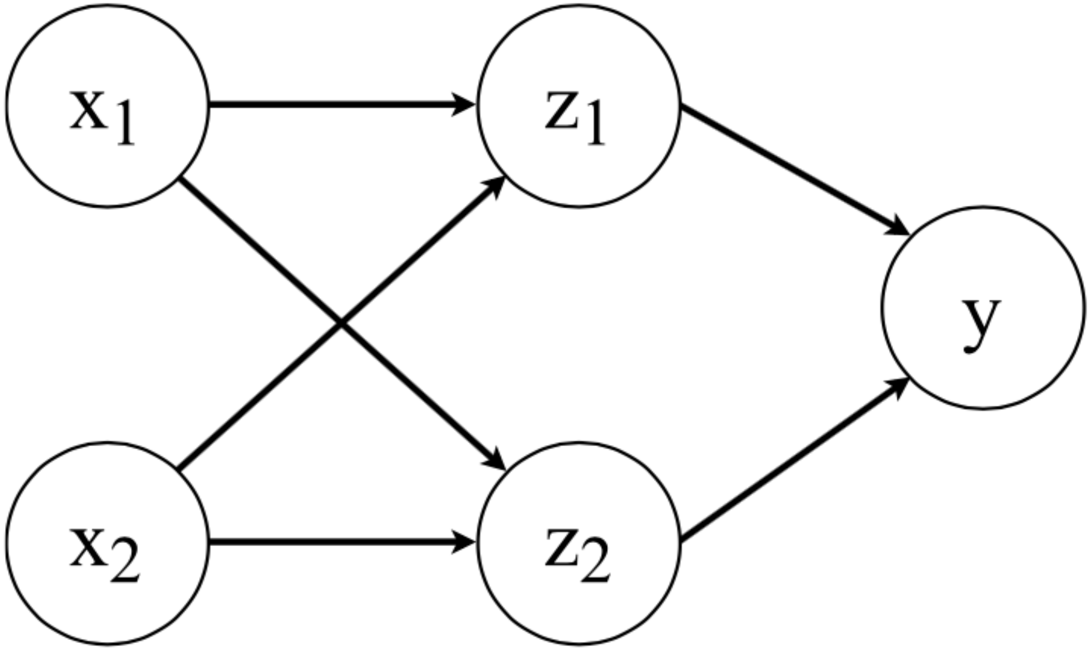
Композиции линейных функций

Идея: ставить в узел линейную функцию.

$$f(x_1, \ldots, x_k) = \sum_{i=1}^{k} w_i x_i + w_0$$

(в каждом узле свой набор w_i)

Композиции линейных функций



$$z_1 = w_{11}x_1 + w_{12}x_2$$

$$z_2 = w_{21}x_1 + w_{22}x_2$$

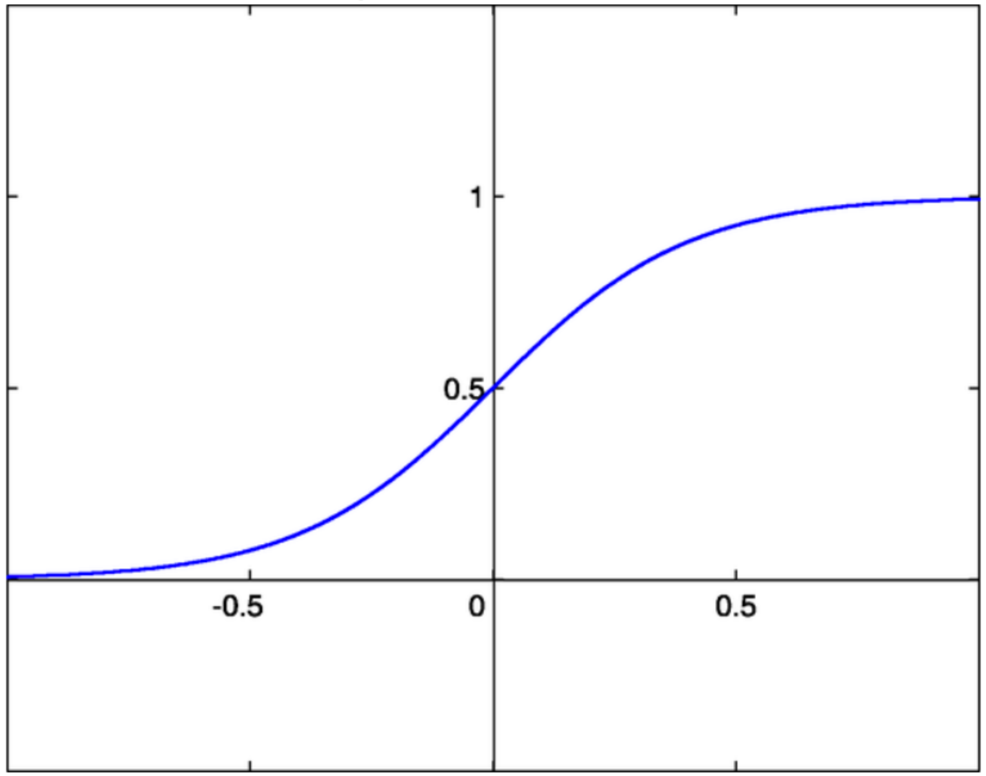
$$y = w_{31}z_1 + w_{32}z_2$$

Проблема: сохраняется линейность

$$y = (w_{31}w_{11} + w_{32}w_{21})x_1 + (w_{31}w_{12} \quad + w_{32}w_{22})x_2$$

Нужны нелинейности!

$h : \mathbb{R} \to \mathbb{R}$ — некоторая нелинейная функция



Пример: сигмоида

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

В каждом узле:

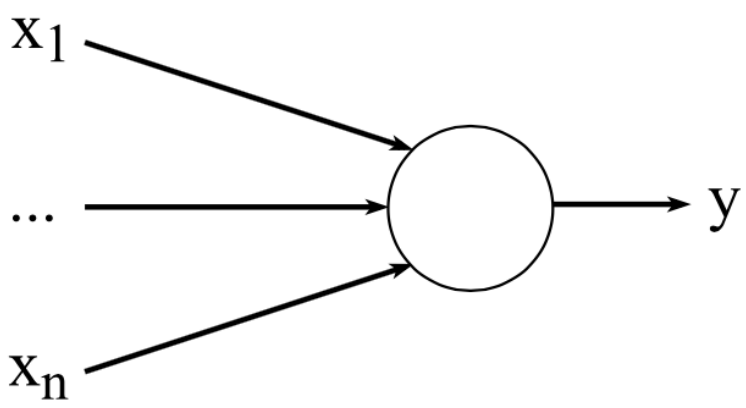
$$f(x_1, \ldots, x_k) = h\left(\sum_{i=1}^{k} w_i x_i + w_0\right)$$

Нейрон

$$y = h\left(\sum_{i=1}^{k} w_i x_i + w_0\right)$$

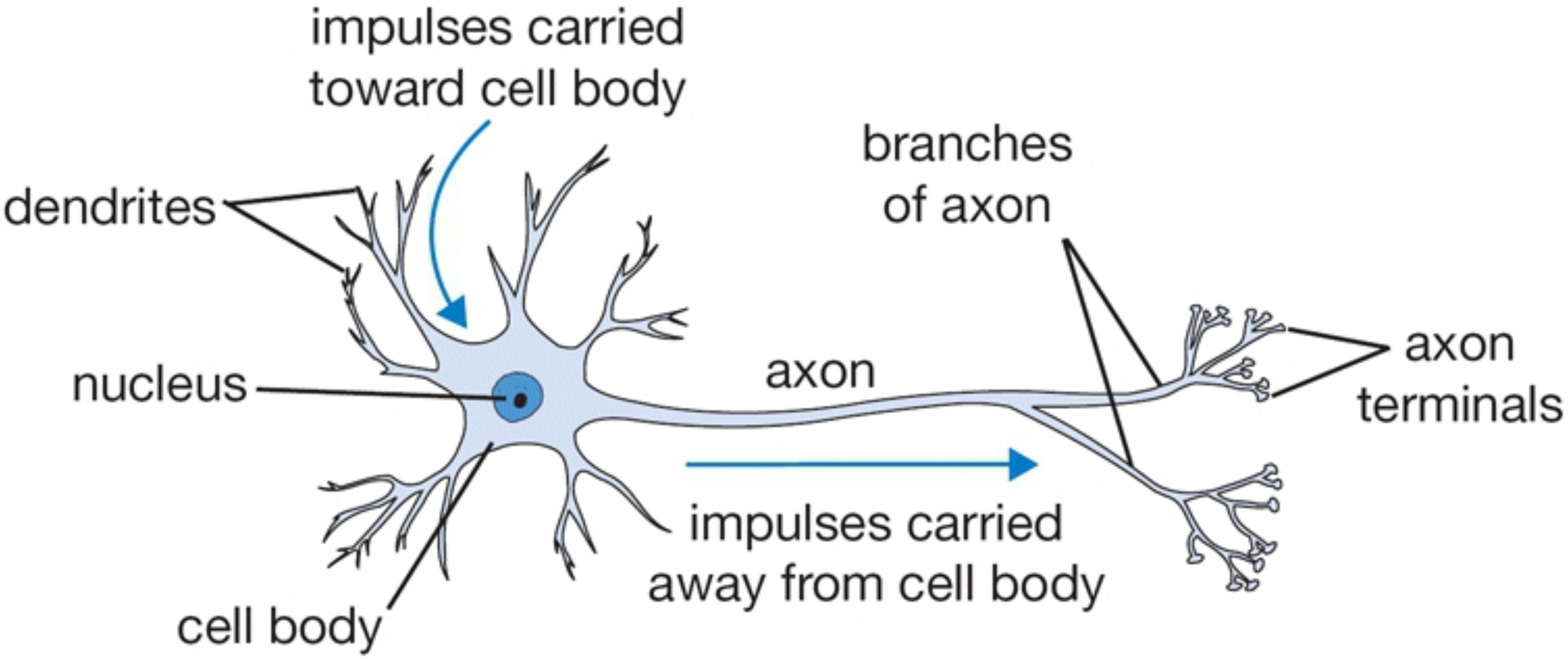
$w_0, w_1, \ldots, w_k$ — параметры нейрона

$h(\cdot)$ — гиперпараметр нейрона



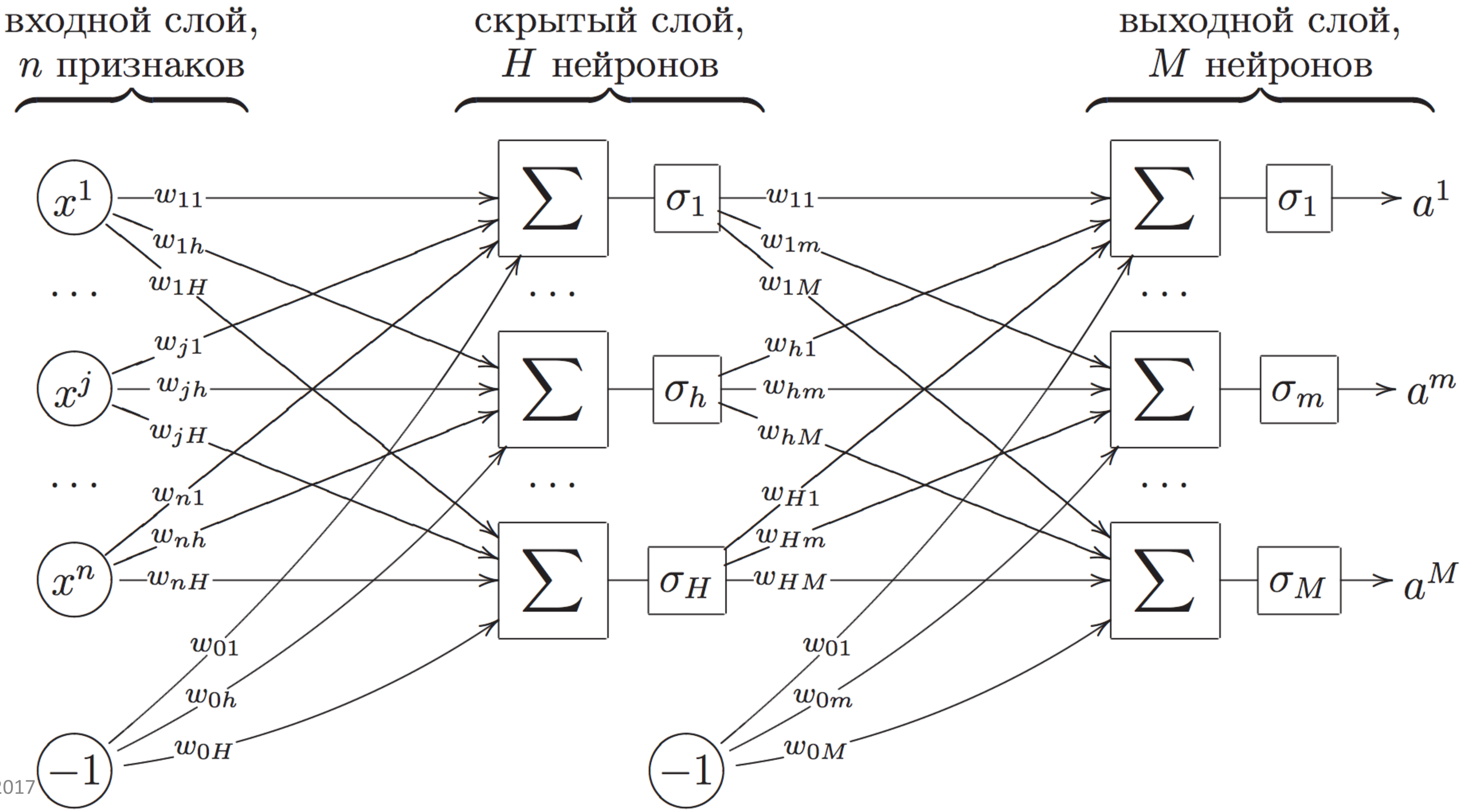
Нейронная сеть $=$ сеть (композиция) нейронов

Почему такое название



Упрощенное устройство нейрона человека

Как выглядит нейронная сеть



Обучение нейронной сети

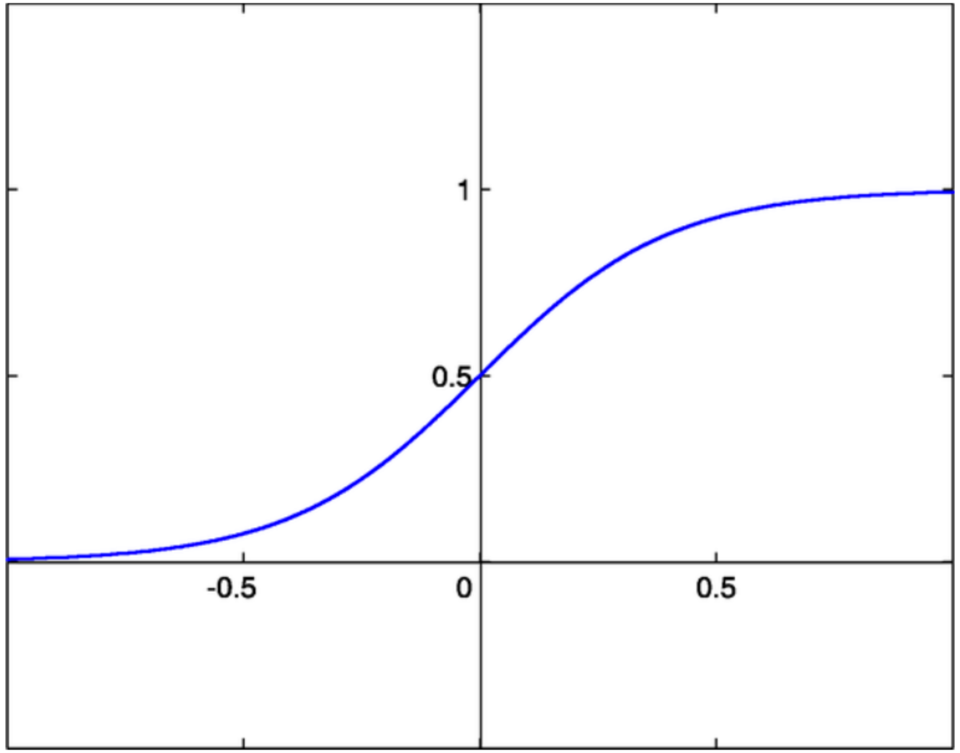
- $\{x_i, y_i\}_{i=1}^N$ — обучающая выборка
- $a(x; w)$ — модель
    - x — входной объект
    - w — параметры (веса входов нейронов)
    - архитектура сети — гиперпараметр
- $L(y, p)$ — функция потерь

$$Q(w) = \sum_{i=1}^{N} L(y_i, a(x_i; w)) \to \min_w$$

Оптимизация градиентным спуском: $\frac{\partial Q}{\partial w}$

Производная сигмоиды

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



$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Пример: один нейрон

$$a(x; w) = \sigma\left(\sum_{i=0}^{d} w_i x^i\right) = \sigma(w^T x)$$

$$L(y, p) = (y - p)^2$$

$$Q(w) = \sum_{i=1}^{N} L(y_i, a(x_i; w))$$

$$\frac{\partial Q(w)}{\partial w} = \sum_{i=1}^{N} \frac{\partial \big(y_i - \sigma(w^T x_i)\big)^2}{\partial w} = ?$$

Производная композиции функций:

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g}\Big(g(x)\Big)\frac{\partial g}{\partial x}(x)$$

Производная сложной функции (chain rule)

$$\frac{\partial Q(w)}{\partial w} = \sum_{i=1}^{N} \frac{\partial \left(y_i - \sigma(w^T x_i)\right)^2}{\partial w} =$$

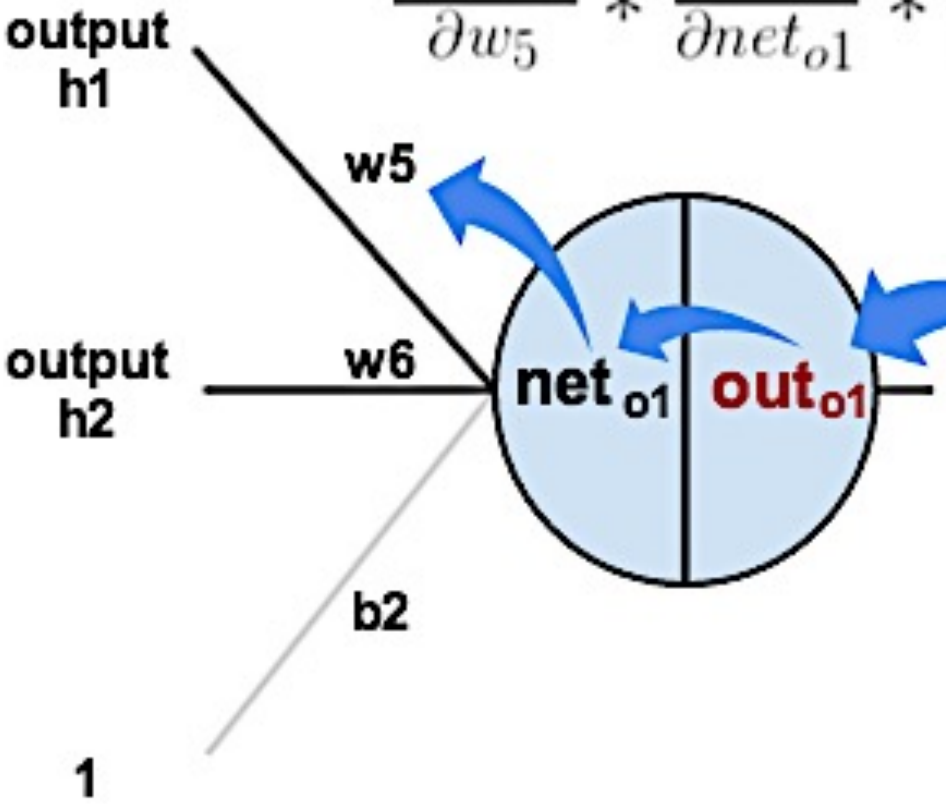
$$= \sum_{i=1}^{N} \boxed{-2(y_i - \sigma(w^T x_i))} \frac{\partial \sigma(w^T x_i)}{\partial w} =$$

$$= \sum_{i=1}^{N} \boxed{-2(y_i - \sigma(w^T x_i))} \boxed{\sigma(w^T x_i)(1 - \sigma(w^T x_i))} \frac{\partial (w^T x_i)}{\partial w} =$$

$$= \sum_{i=1}^{N} \boxed{-2(y_i - \sigma(w^T x_i))} \boxed{\sigma(w^T x_i)(1 - \sigma(w^T x_i))} x_i$$

То же самое визуально

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$



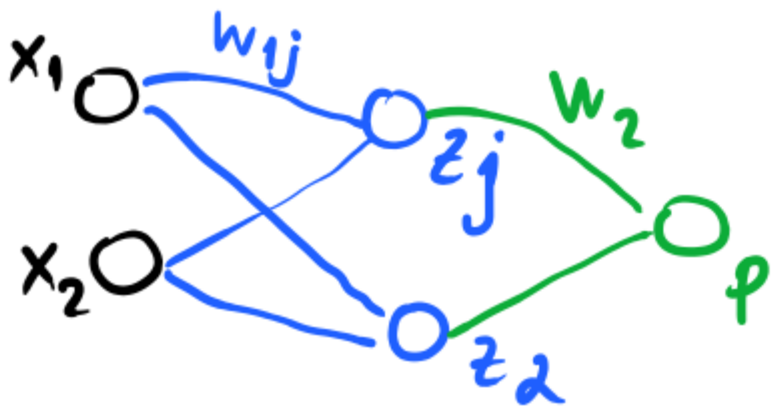
$$E_{o1} = \tfrac{1}{2}(target_{o1} - out_{o1})^2$$

$$E_{total} = E_{o1} + E_{o2}$$

Если добавить скрытый слой

$$z_j(x) = \sigma(w_{1,j}^T\, x), \qquad p(z) = \sigma(w_2^T z)$$

$$L(y, p) = (y - p)^2$$



$$Q(w) = \sum_{i=1}^{N} L(y_i, a(x_i; w))$$

$$\frac{\partial Q}{\partial w_2} = \sum_{i=1}^{N} \boxed{\frac{\partial L}{\partial p}} \frac{\partial p}{\partial w_2}$$

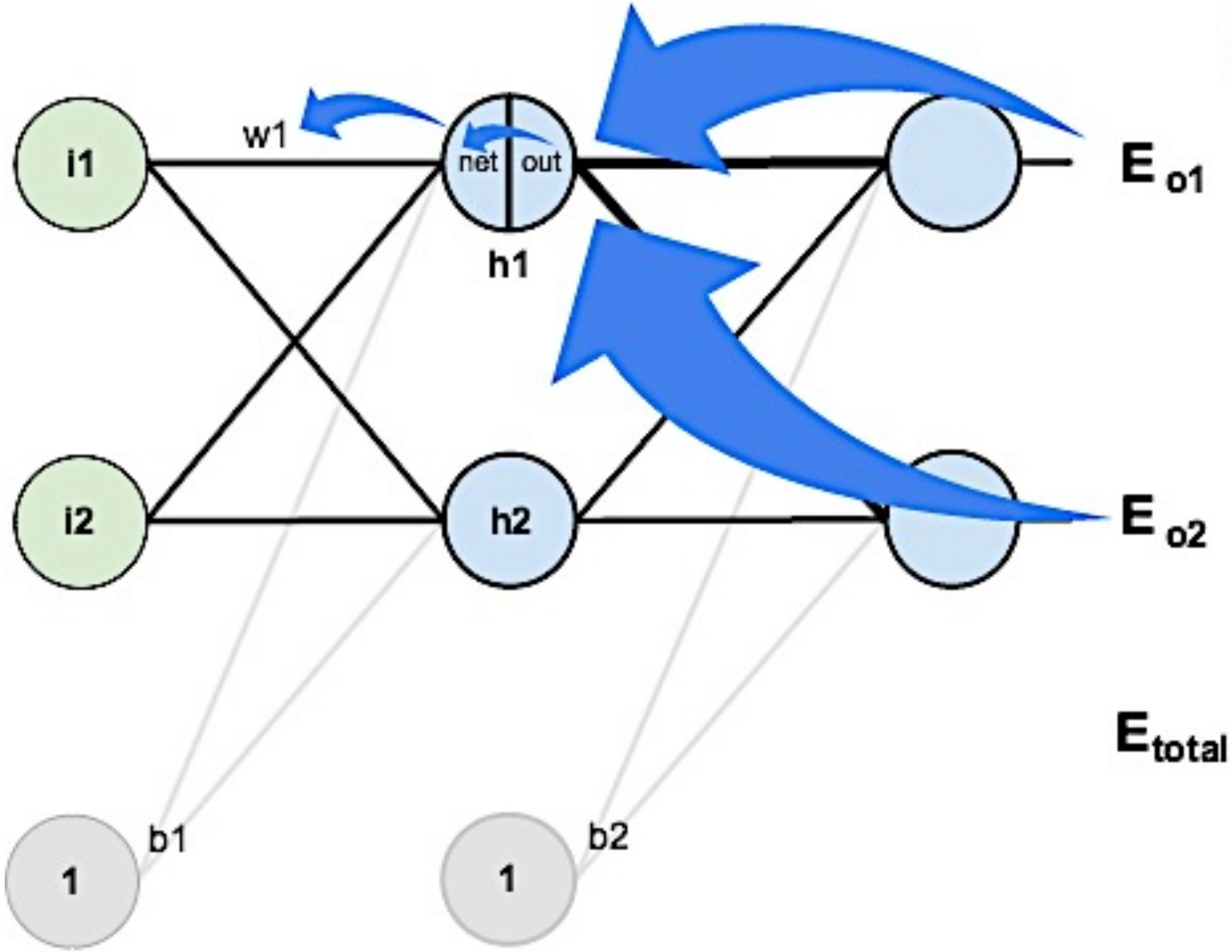
Это уже умеем считать

$$\frac{\partial Q}{\partial w_{1,j}} = \sum_{i=1}^{N} \boxed{\frac{\partial L}{\partial p}} \frac{\partial p}{\partial w_{1,j}} = \sum_{i=1}^{N} \boxed{\frac{\partial L}{\partial p}} \sum_{s=1}^{J} \frac{\partial p}{\partial z_s} \frac{\partial z_s}{\partial w_{1,j}} =$$

Только одно не ноль

$$= \sum_{i=1}^{N} \boxed{\frac{\partial L}{\partial p}} \frac{\partial p}{\partial z_j} \frac{\partial z_j}{\partial w_{1,j}}$$

Похожий пример визуально

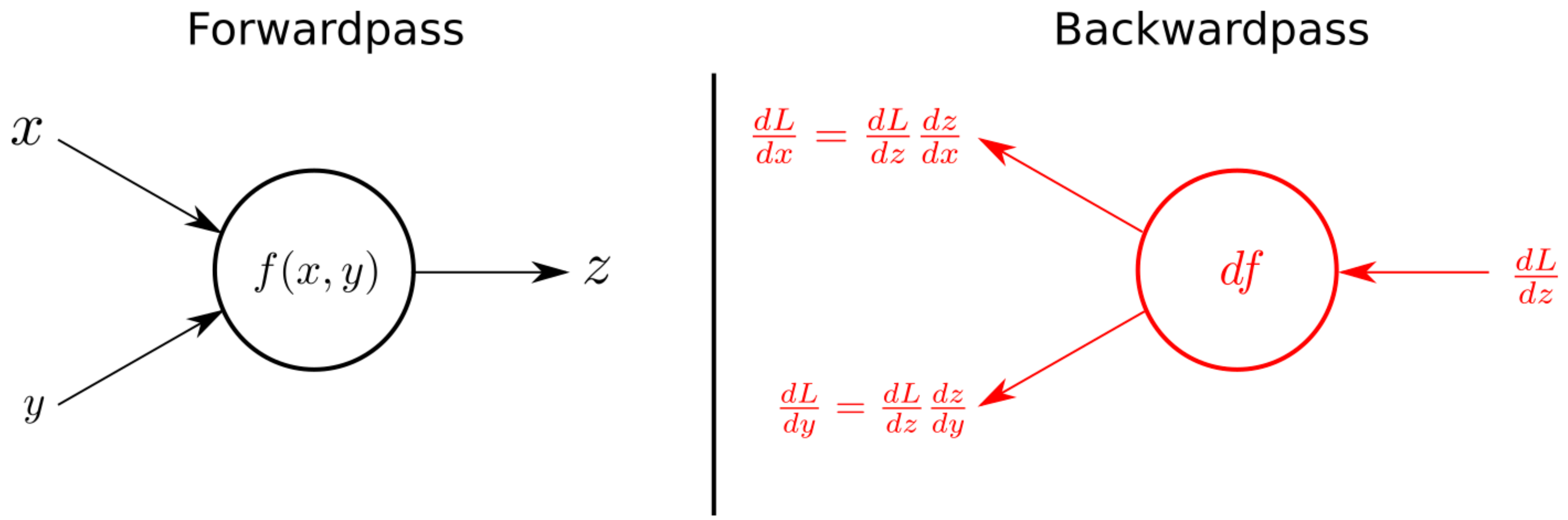


$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$E_{total} = E_{o1} + E_{o2}$$

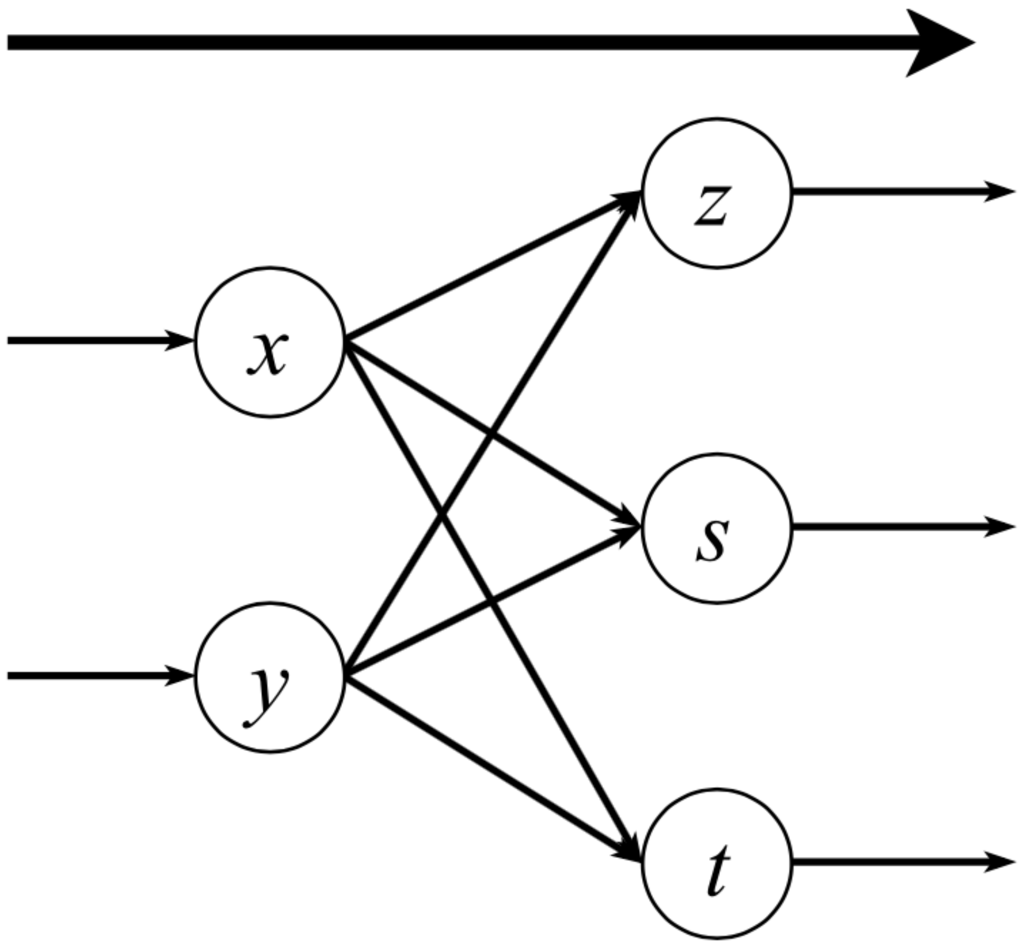
Обратное распространение ошибки (back-prop)



Прямой ход до конца

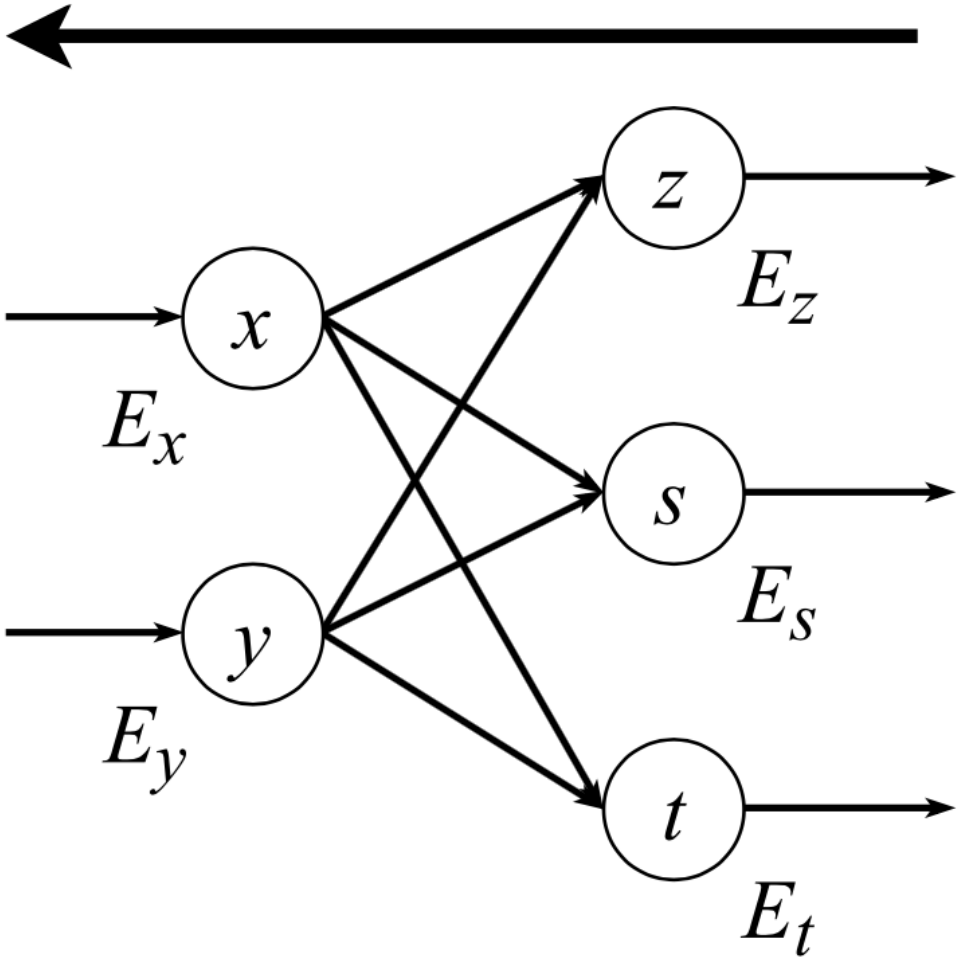
Потом обратный ход полностью

И еще раз: прямой ход



$$z = h(w_{z,x}x + w_{z,y}y + w_{z,0})$$

И обратный ход



$$E_\alpha = \frac{\partial Q}{\partial \alpha}$$

$$\alpha = x, y, z, s, t, ..$$

$$\frac{\partial Q}{\partial w_\alpha} = E_\alpha \frac{\partial \alpha}{\partial w_\alpha}$$

$$E_x = \frac{\partial Q}{\partial x} = \frac{\partial Q}{\partial z}\frac{\partial z}{\partial x} + \frac{\partial Q}{\partial s}\frac{\partial s}{\partial x} + \frac{\partial Q}{\partial t}\frac{\partial t}{\partial x} = E_z \frac{\partial z}{\partial x} + E_s \frac{\partial s}{\partial x} + E_t \frac{\partial t}{\partial x}$$

Обратное распространение ошибки (back-prop)

- $a(x; w)$ — сеть

  (композиция дифференцируемых функций)

- $L(y, p)$ — функция потерь

  (дифференцируемая функция)

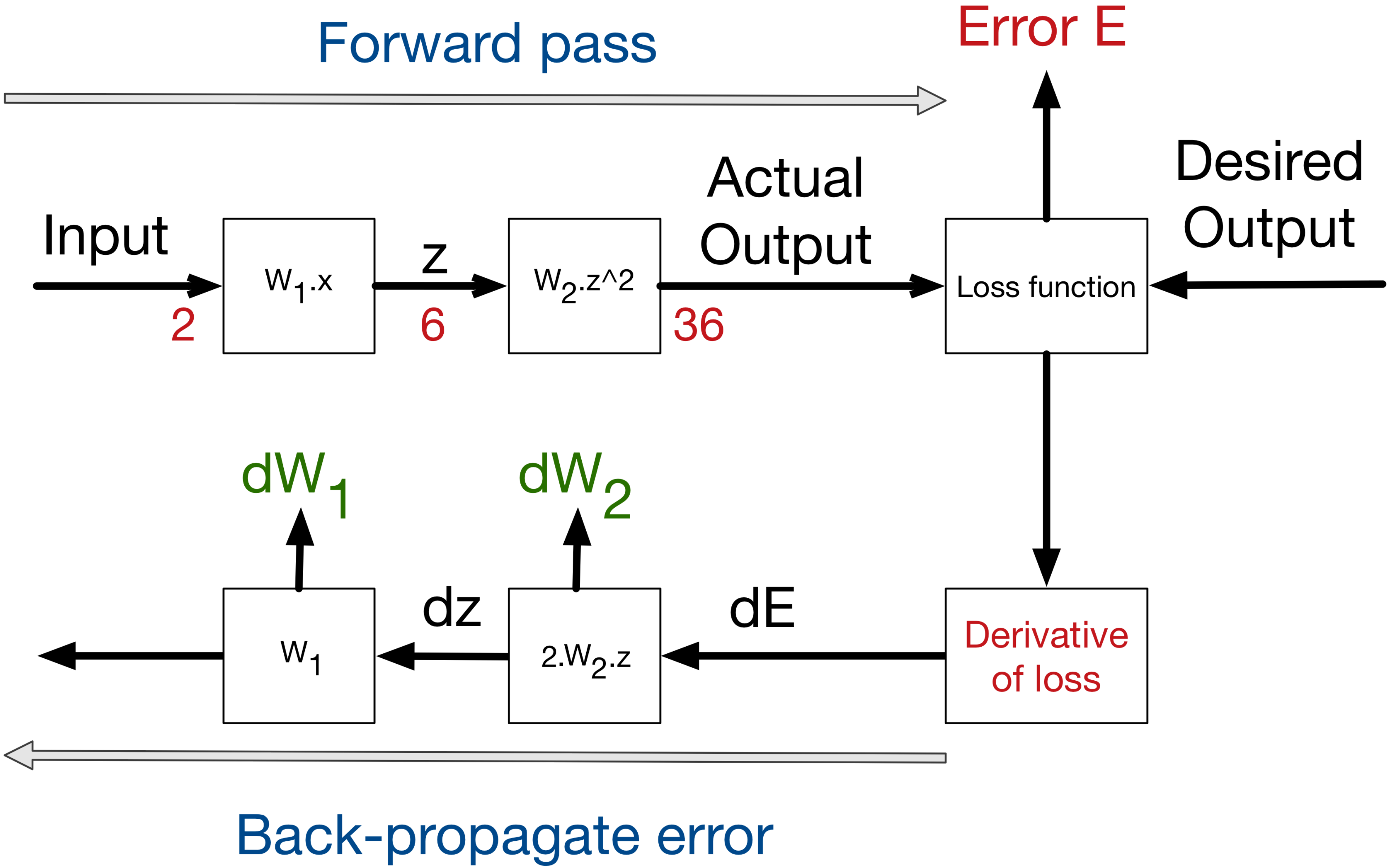
- $Q(w) = \sum_{i=1}^{N} L(y_i, a(x_i; w))$ — потери на выборке

  (композиция дифференцируемых функций)

$$\frac{\partial Q(\alpha_1, \alpha_2, \ldots)}{\partial w_j} = \sum_s \frac{\partial Q}{\alpha_s} \frac{\partial \alpha_s(\beta_1, \beta_2, \ldots)}{\partial w_j} =$$

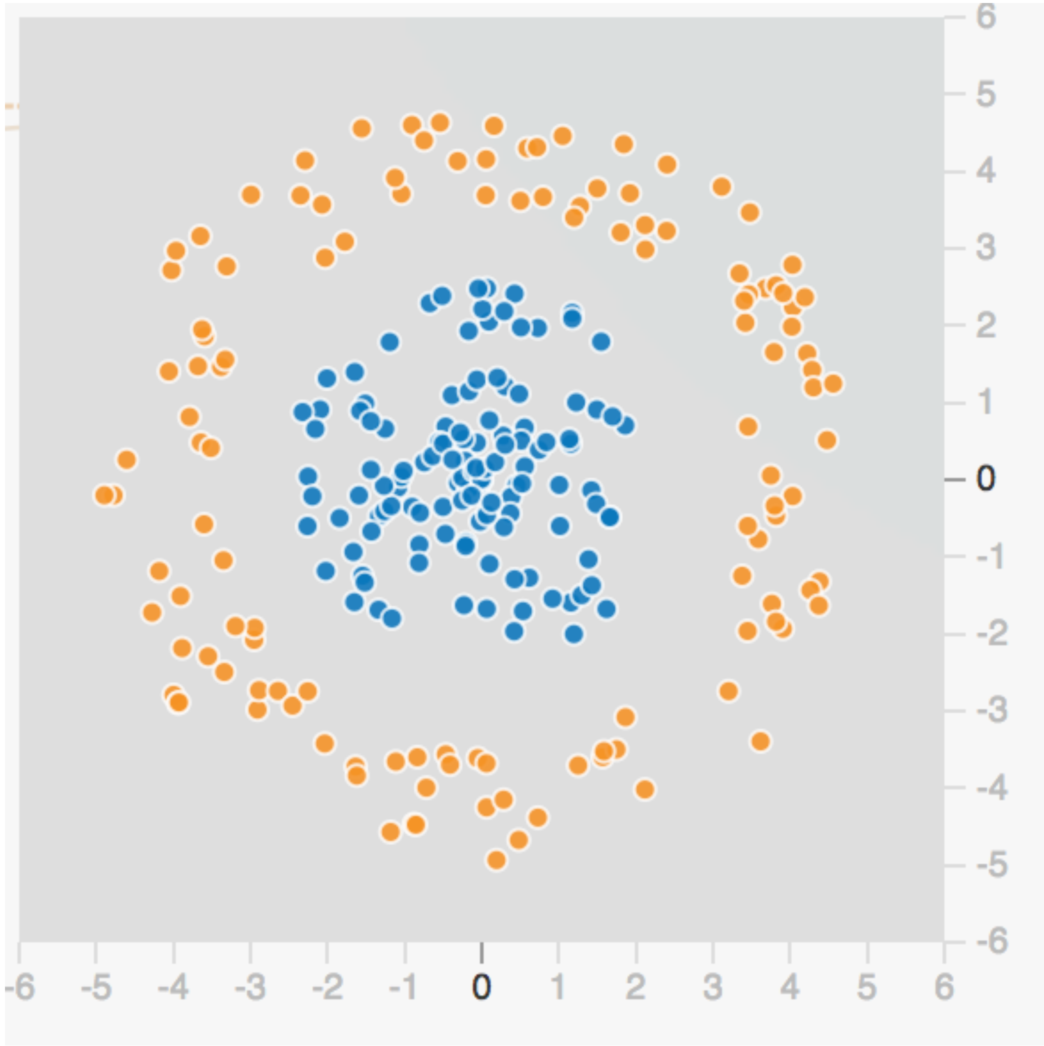
$$= \sum_s \frac{\partial Q}{\alpha_s} \sum_t \frac{\partial \alpha_s}{\partial \beta_t} \frac{\partial \beta_t(\gamma_1, \gamma_2, \ldots)}{\partial w_j} = \ldots$$

Обратные шаги это тоже граф

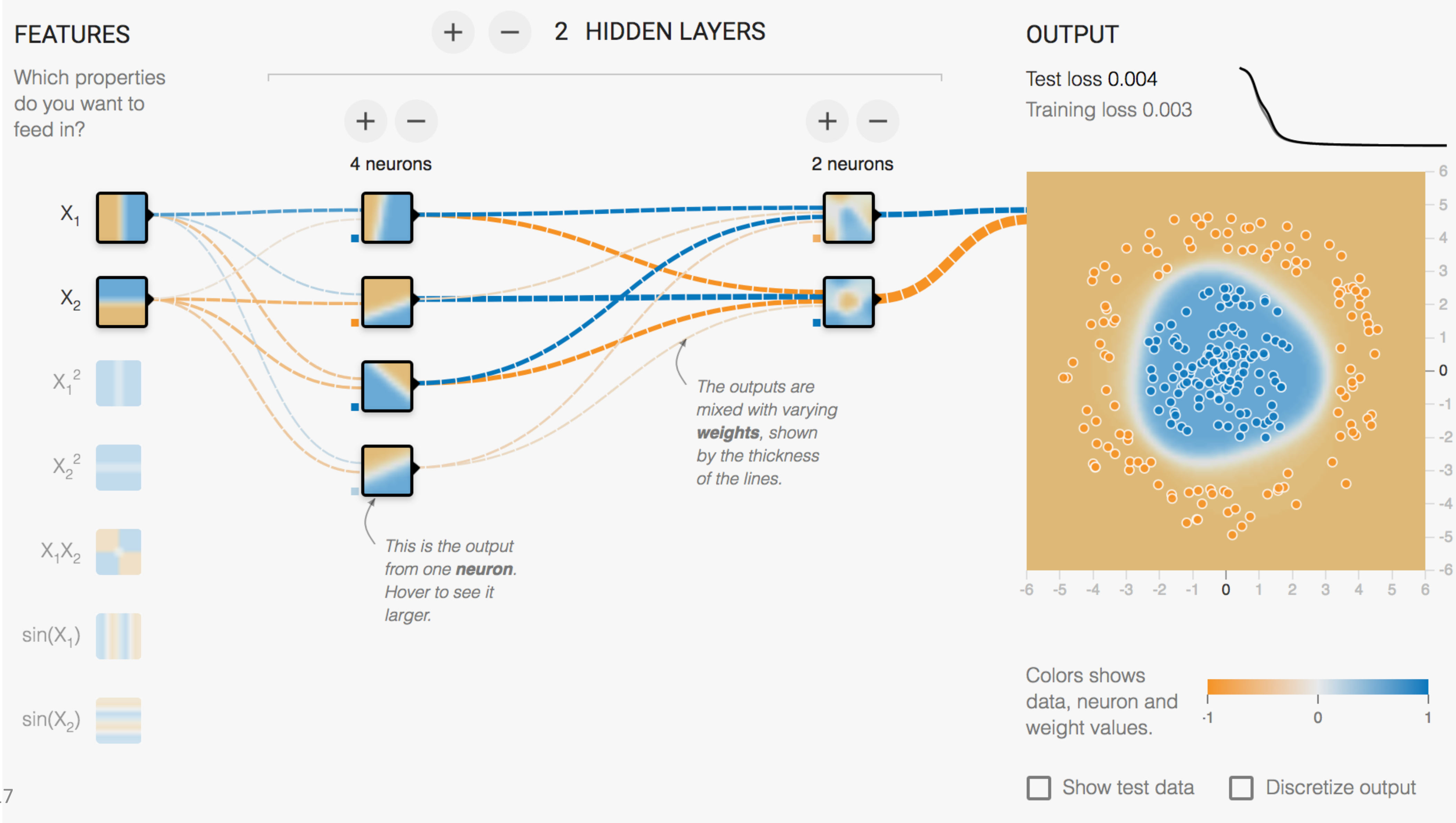


Демо: нейросети в TensorFlow Playground

- http://playground.tensorflow.org



Демо: нейросети в TensorFlow Playground



Резюме

- **Плюсы:**
  - Универсальные аппроксиматоры (приближают сложные функции)
  - Сложные композиции простых функций (легко дифференцировать)

- **Минусы:**
  - Архитектуру надо подбирать руками
  - Сильное переобучение (нужна регуляризация)
  - Проблемы с затухающими или взрывающимися градиентами

Ссылки

- https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
- http://www.machinelearning.ru/wiki/images/c/c2/Voron-ML-NeuralNets-slides.pdf