Искусственный Интеллект

<u>Лекция 4:</u> Логистическая регрессия

Мартынюк Полина Антоновна

telegram: @PAMartynyuk
email: pa-martynyuk@yandex.ru

Модель

Task

Задача

Loss

Функция потерь

Optimizer

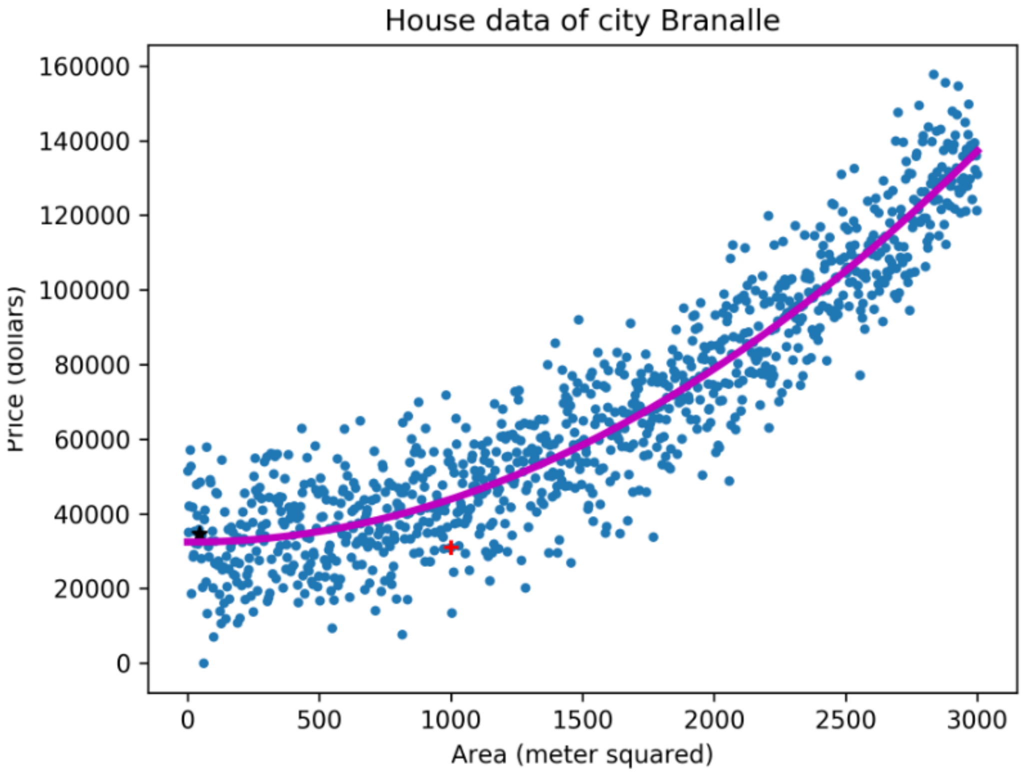
Оптимизатор

Регрессия

Регрессия (Regression)

- **Прогнозирование цены недвижимости** - определение стоимости домов или квартир на основе их характеристик, таких как площадь, количество комнат, район и другие факторы.

Площадь квартиры (X1, feature)	Количество комнат (X2, feature)	Цена квартиры (Y, label, target)
73	2	5.5
150	4	15.8
34	1	4.2
101	3	10.9



Модель: линейная регрессия

Task	**Loss**	**Optimizer**
Задача	Функция потерь	Оптимизатор
Регрессия	MSE	DG, SGD, …
$y = \sum_{i=1}^{p}(x_i w_i) + b$	$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$w = w - \alpha \nabla L(w)$ $b = b - \alpha \nabla L(b)$

Оптимизаторы бывают разные

https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html

ML Glossary

Search docs

BASICS
- Linear Regression
- Gradient Descent
- Logistic Regression
- Glossary

MATH
- Calculus
- Linear Algebra
- Probability (TODO)
- Statistics (TODO)
- Notation

NEURAL NETWORKS
- Concepts
- Forwardpropagation
- Backpropagation
- Activation Functions
- Layers
- Loss Functions
- Optimizers
  - Adagrad

Docs » Optimizers   Edit on GitHub

Optimizers

What is Optimizer ?

It is very important to tweak the weights of the model during the training process, to make our predictions as correct and optimized as possible. But how exactly do you do that? How do you change the parameters of your model, by how much, and when?

Best answer to all above question is *optimizers*. They tie together the loss function and model parameters by updating the model in response to the output of the loss function. In simpler terms, optimizers shape and mold your model into its most accurate possible form by futzing with the weights. The loss function is the guide to the terrain, telling the optimizer when it's moving in the right or wrong direction.

Below are list of example optimizers

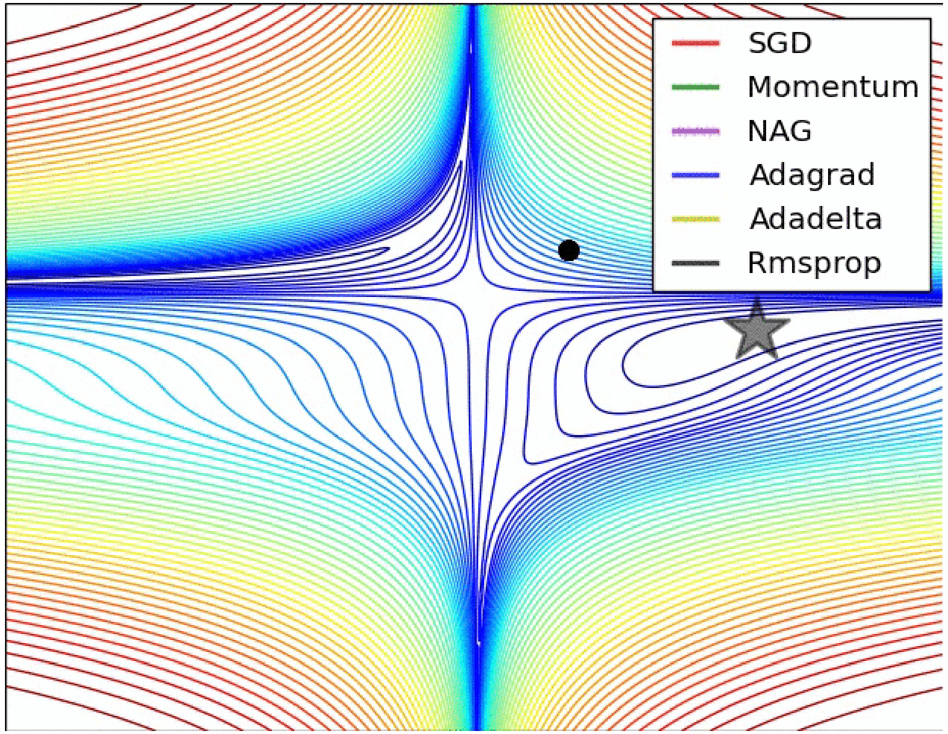
- Adagrad
- Adadelta
- Adam
- Conjugate Gradients
- BFGS
- Momentum
- Nesterov Momentum
- Newton's Method
- RMSProp
- SGD

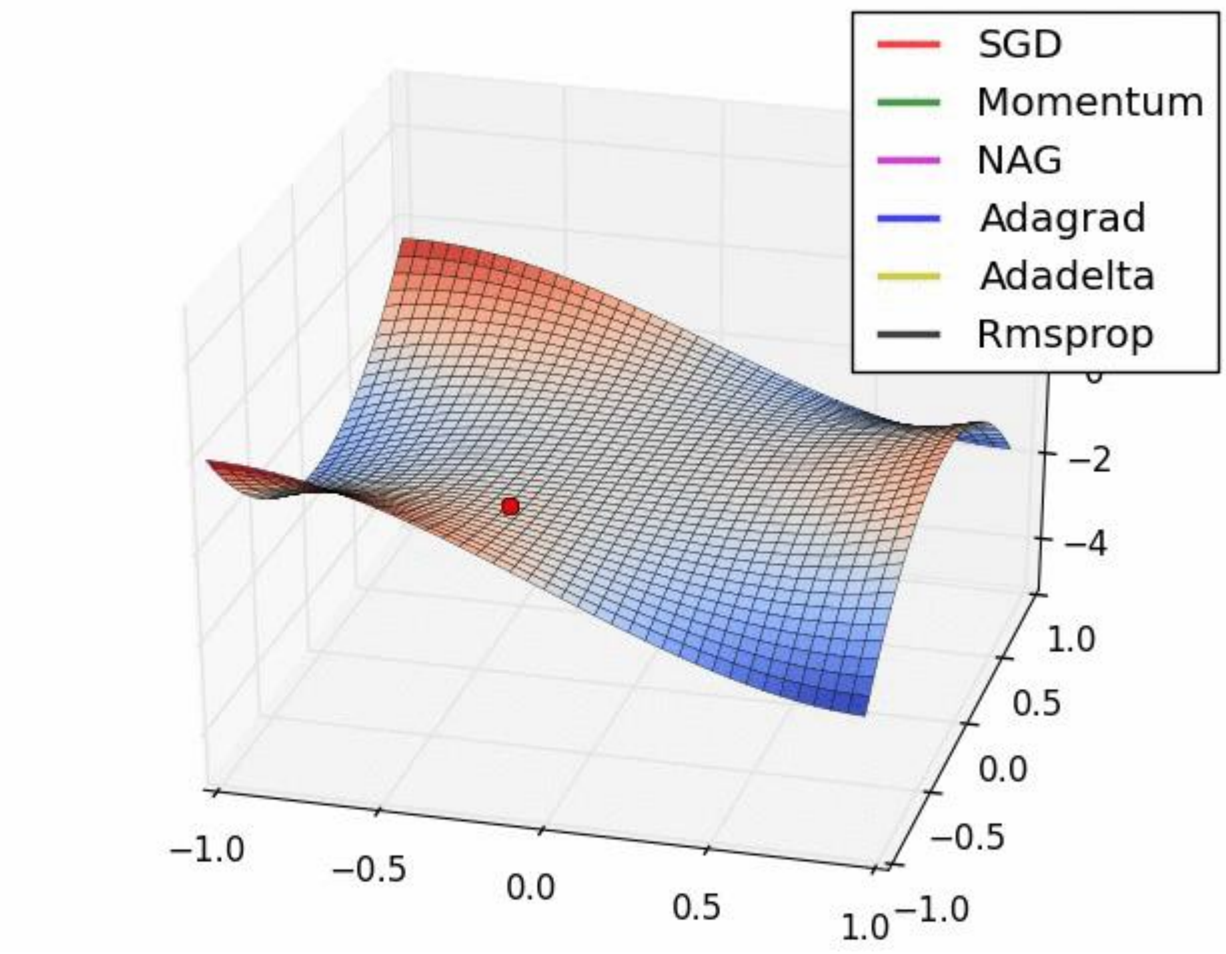
Below are list of example optimizers

- Adagrad
- Adadelta
- Adam
- Conjugate Gradients
- BFGS
- Momentum
- Nesterov Momentum
- Newton's Method
- RMSProp
- SGD

Оптимизаторы бывают разные

https://imgur.com/a/Hqolp#NKsFHJb





Модель

Task

Задача

Loss

Функция потерь

Optimizer

Оптимизатор

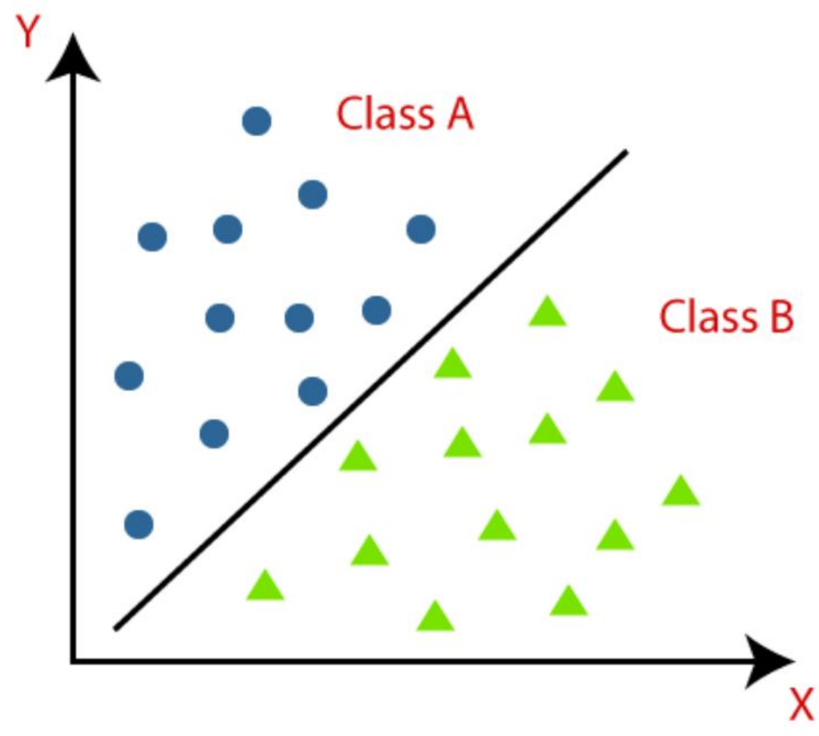
Классификация

Задачи обучения с учителем: Классификация

Классификация (Classification)

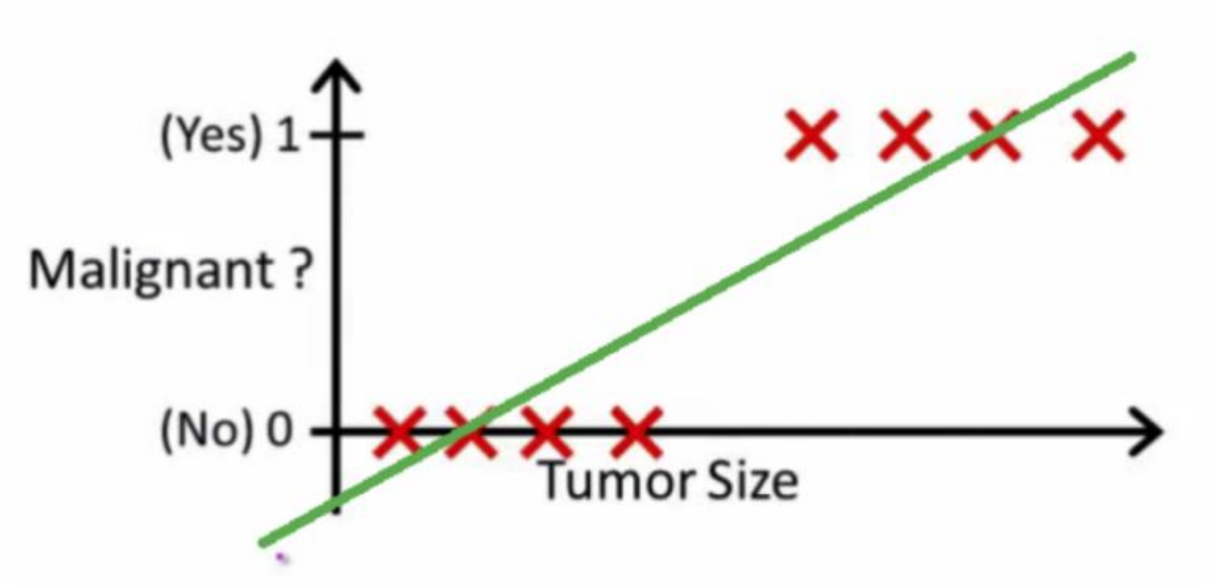
- предсказание категории или класса для входных данных. Выходная величина принимает значение одного из заранее определённых классов.

- **Бинарная классификация**: В задачах бинарной классификации метка может принимать одно из двух значений, например, 0 или 1, Да или Нет, Положительный или Отрицательный и т.д.
- **Многоклассовая классификация**: В многоклассовой классификации метка может относиться к одному из нескольких классов или категорий. Например, при классификации изображений метка может указывать на тип объекта (кошка, собака, автомобиль и так далее)



Почему бы не использовать линейную регрессию?

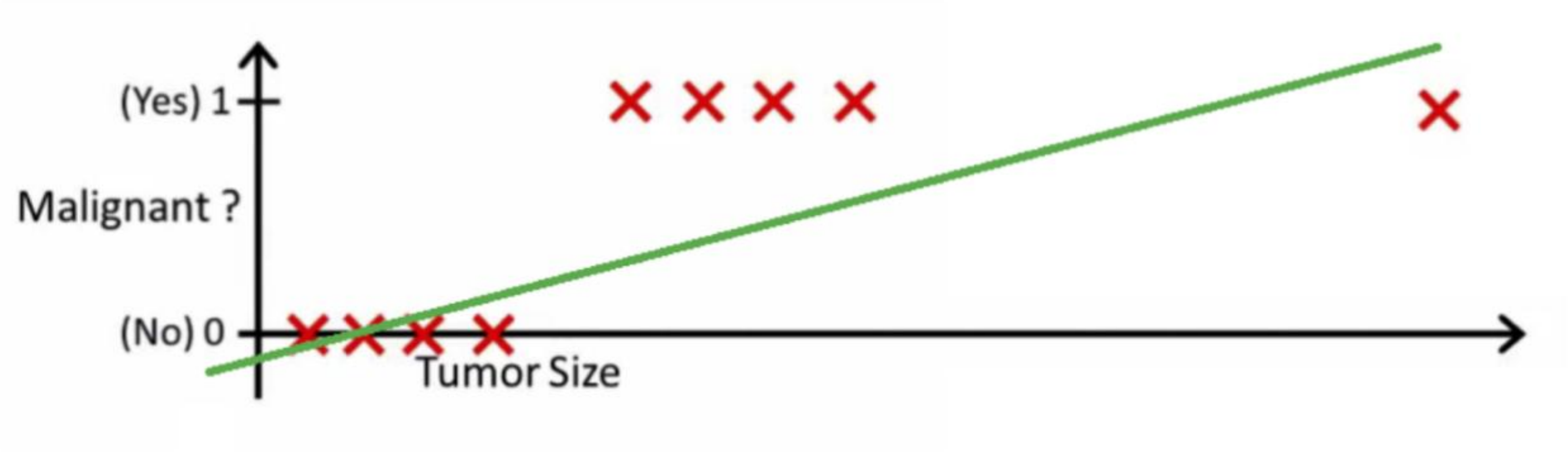
X (size)	y
1.5	0
4.2	0
8.1	1
2.4	0
7.1	1
4.0	1



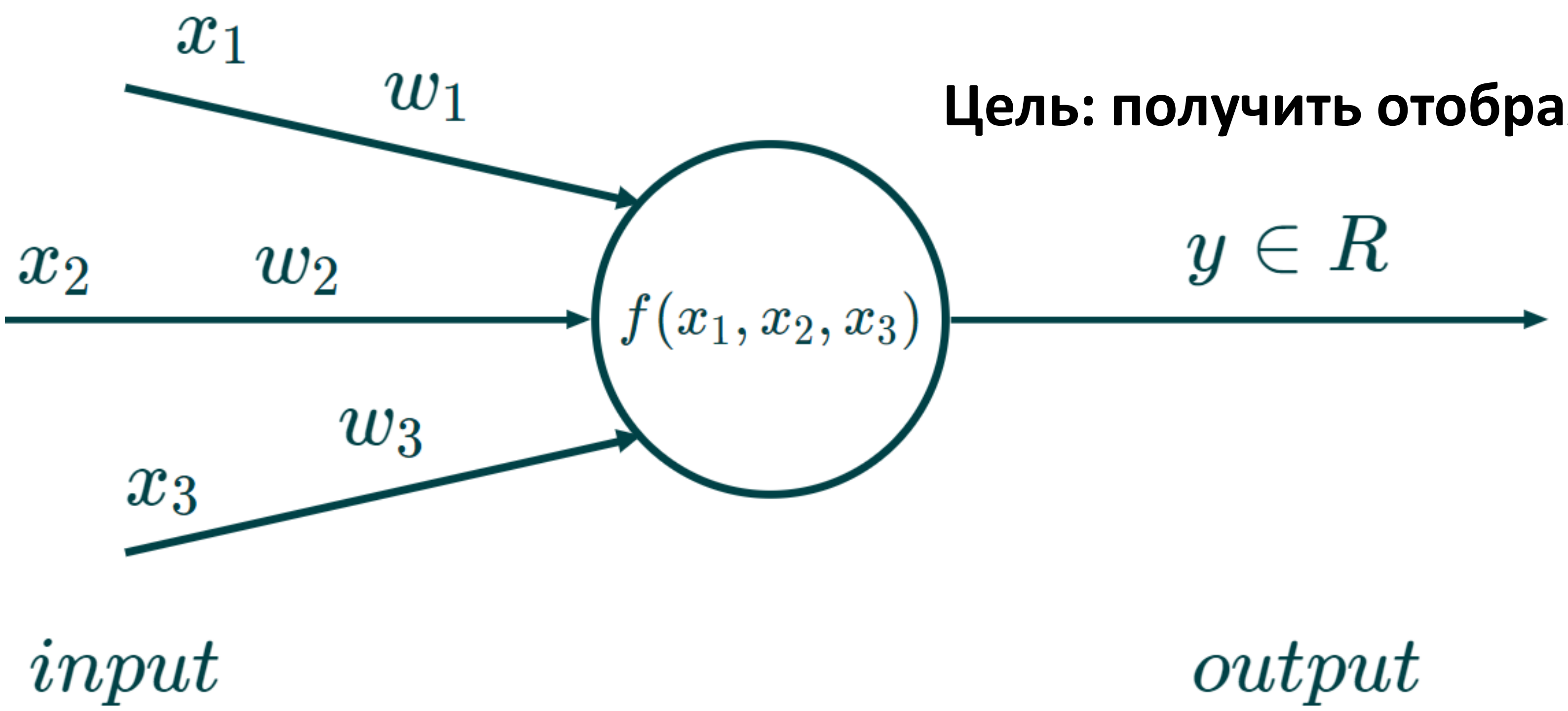
Но…

Почему бы не использовать линейную регрессию?

…чувствительность к выбросам



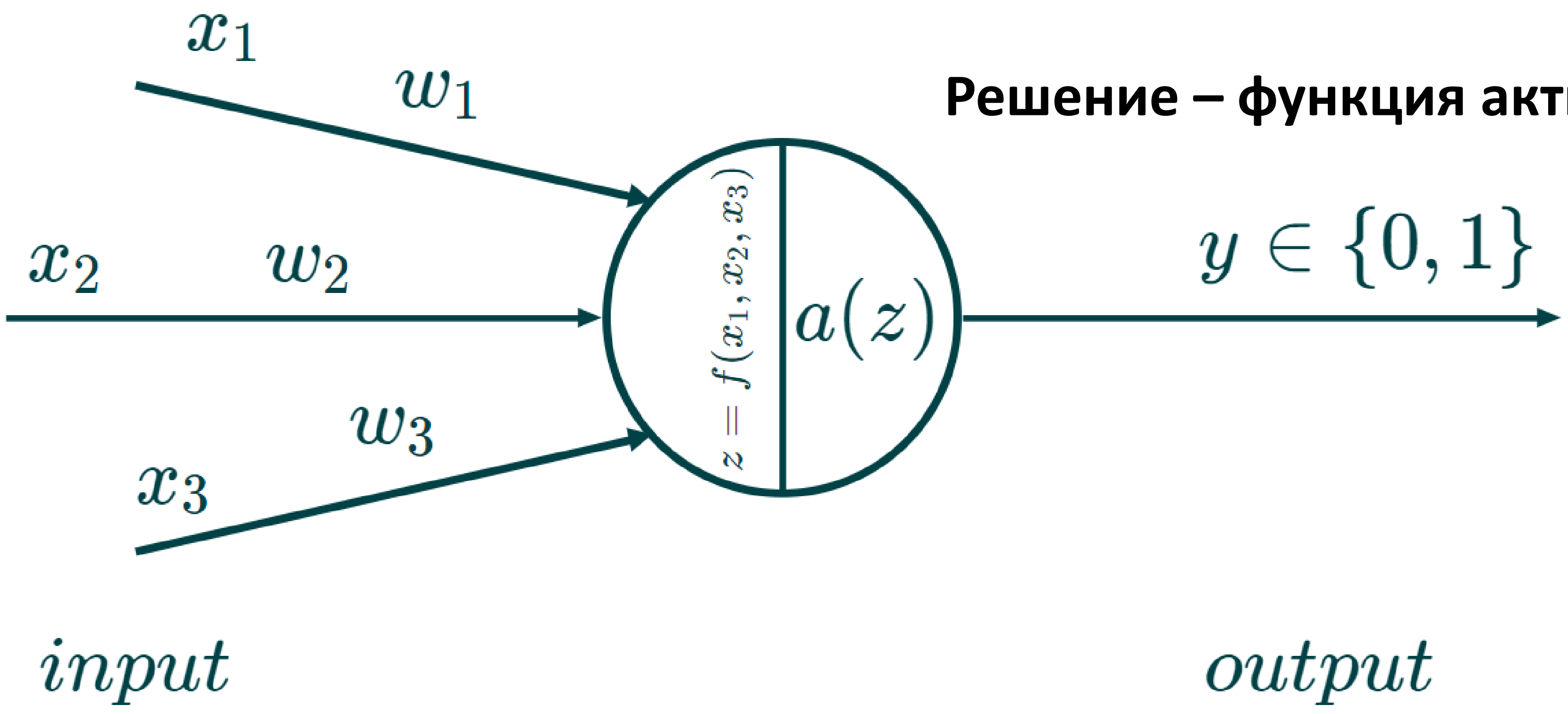
Перцептрон



Цель: получить отображение y в {0,1}

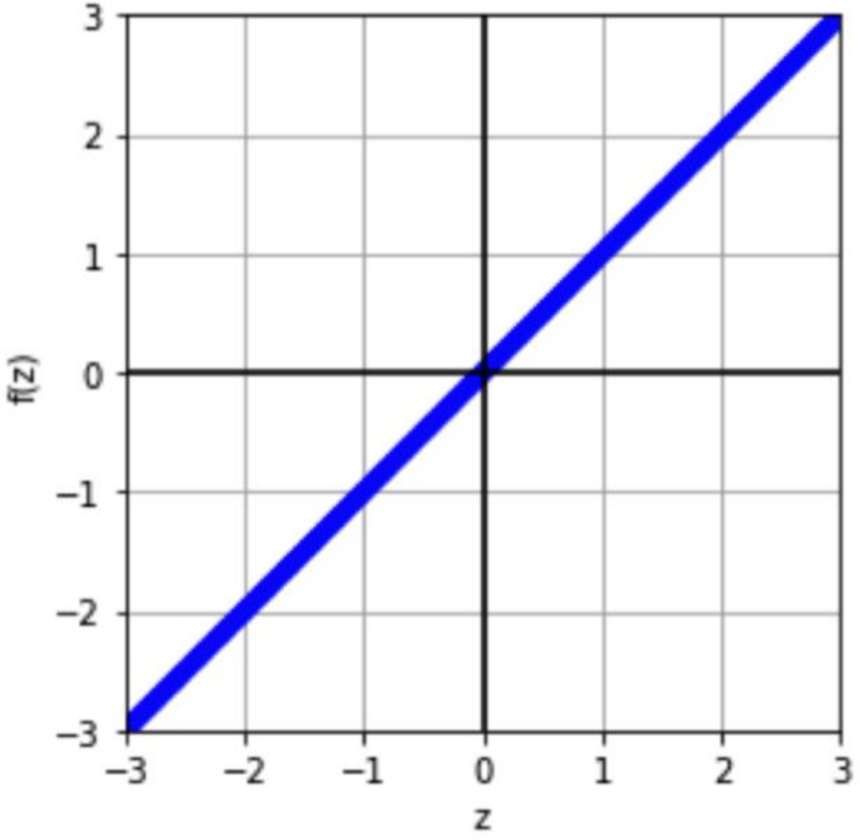
Перцептрон

Решение – функция активации



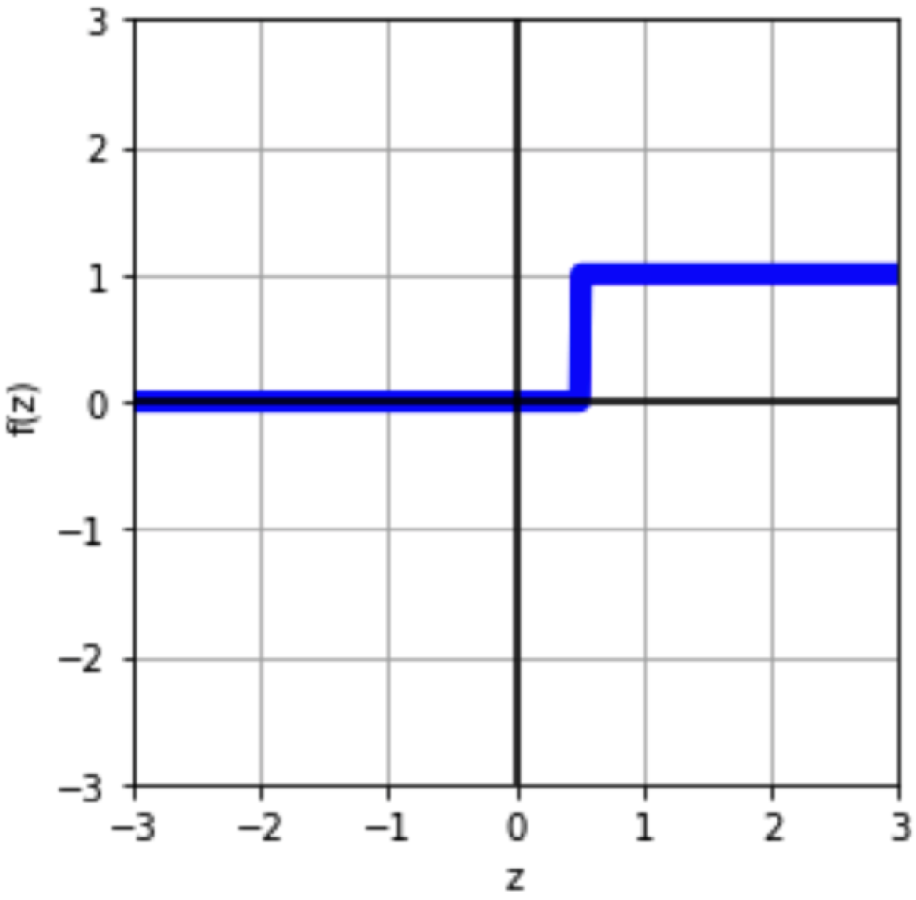
Линейная регрессия

$$f(z) = z$$



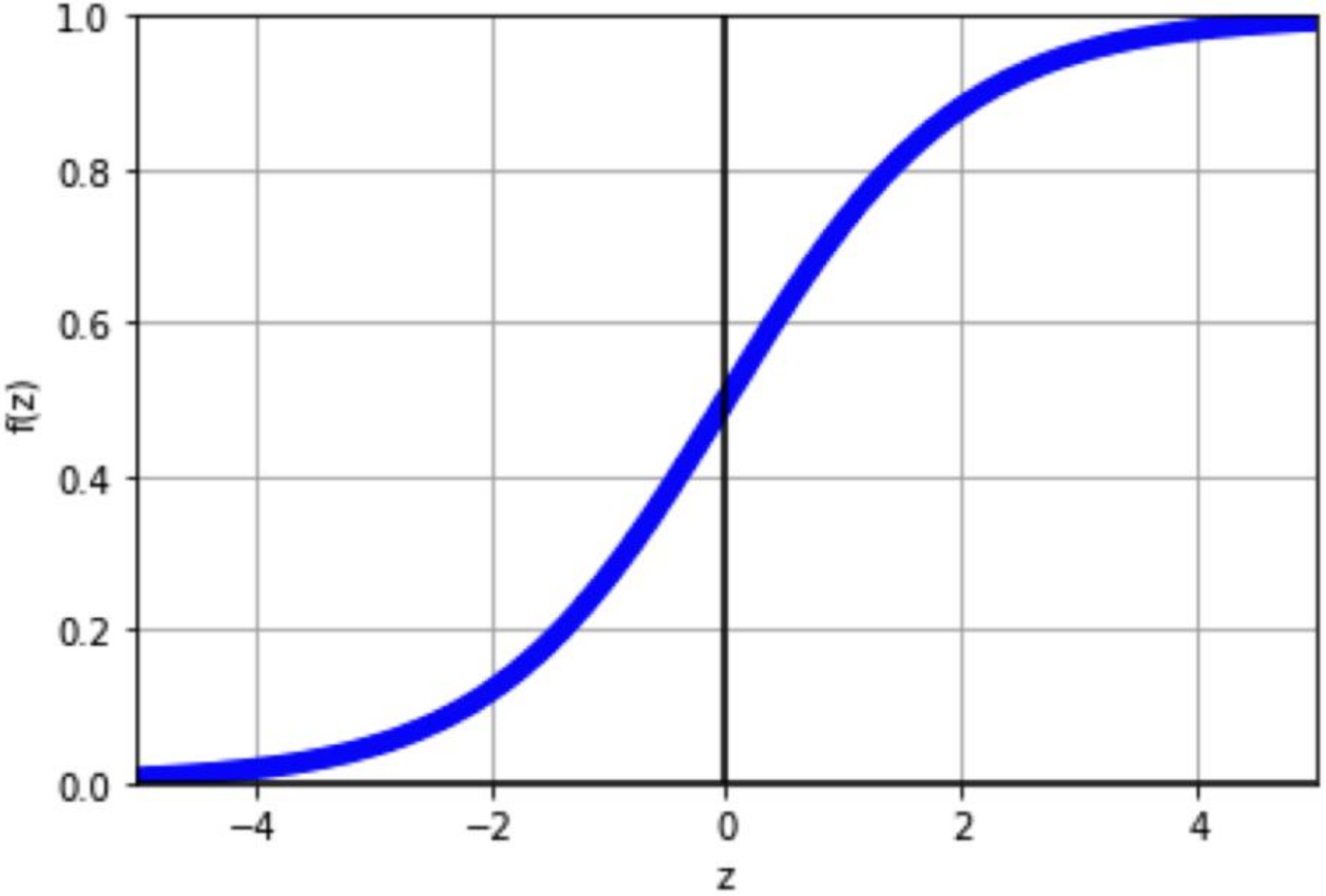
Граничное значение Treshold

$$f(z) = \begin{cases} 1, & \text{if } z > 0.5 \\ 0, & \text{otherwise} \end{cases}$$



Сигмоида

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



Влияние на функцию потерь

На примере MSE

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Было:

$$L(w_1, \ldots, w_p) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (x_1 w_1 + \ldots + x_p w_p + b))^2$$

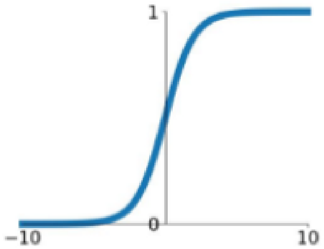
Стало:

$$L(w_1, \ldots, w_p) = \frac{1}{n} \sum_{i=1}^{n} (y_i - a(x_1 w_1 + \ldots + x_p w_p + b))^2$$

Другие функции активации

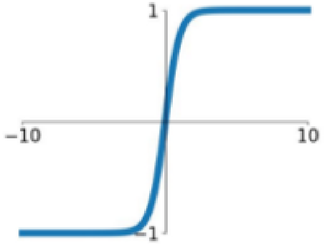
Sigmoid

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



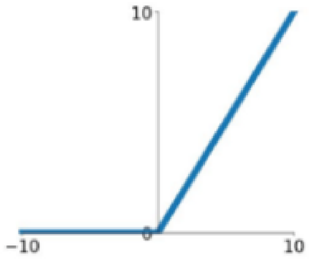
tanh

$\tanh(x)$



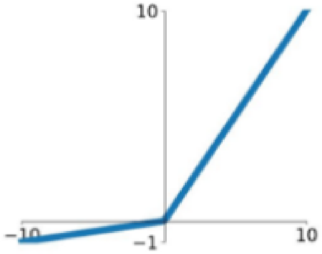
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

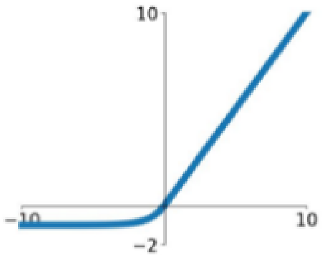


Maxout

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

ELU

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



https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

Новая функция потерь

MSE

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

Что нам нужно:

$$\begin{array}{ll} \hat{y} = 1 & 0.7 \\ \hat{y} = 0 & 0.3 \end{array}$$

→ Степень уверенности модели в своем ответе

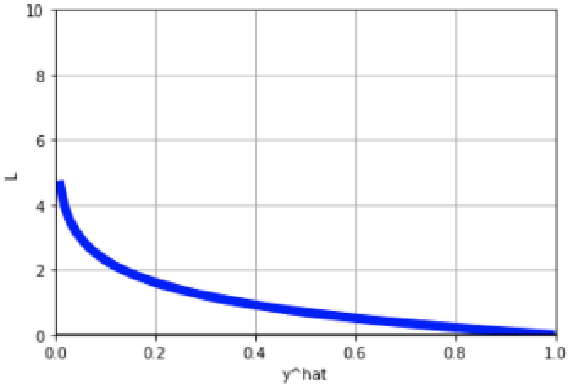
Log Loss

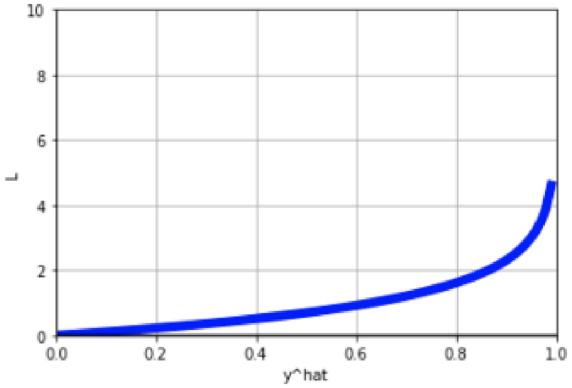
$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^{n} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

Log Loss

Слагаемое 1 **Слагаемое 2**

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i))$$



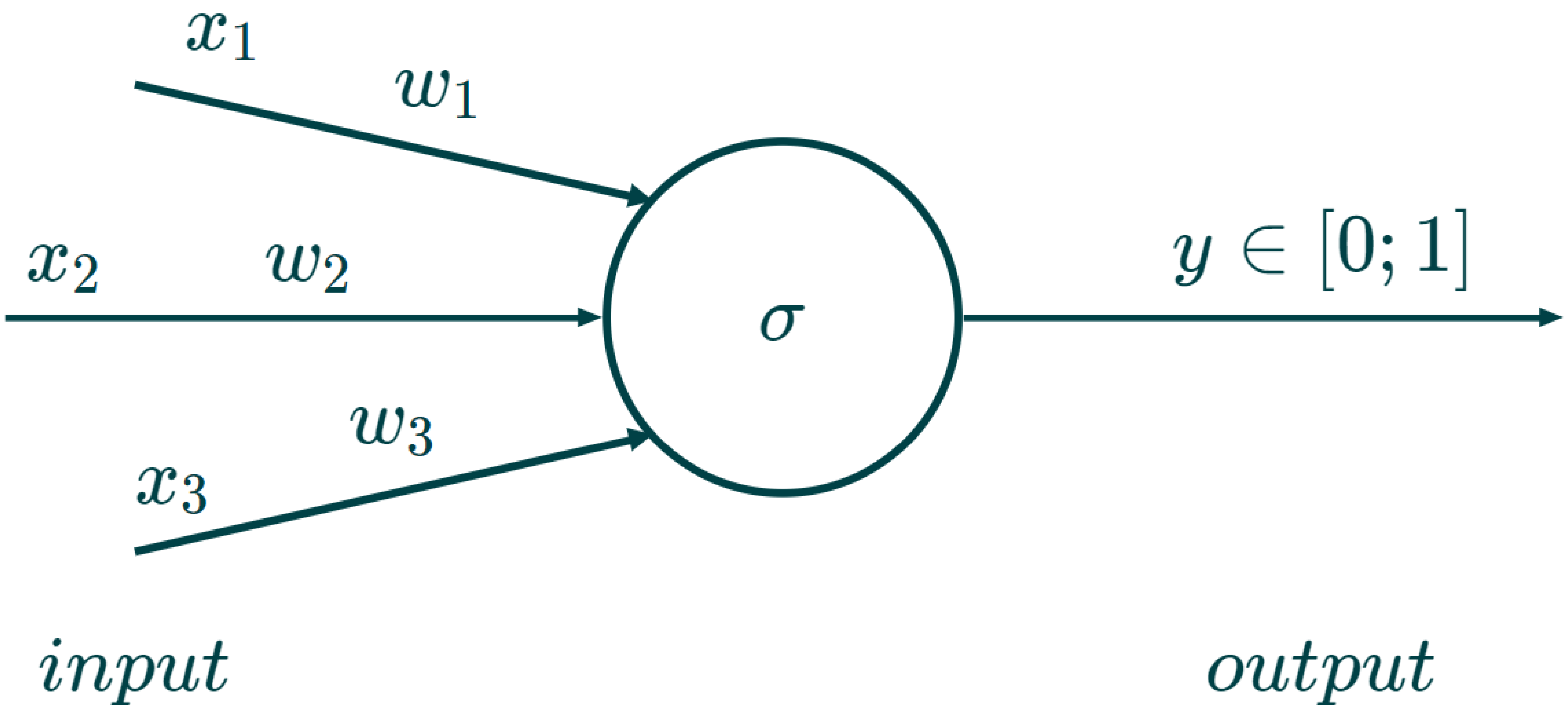


Производная Log Loss

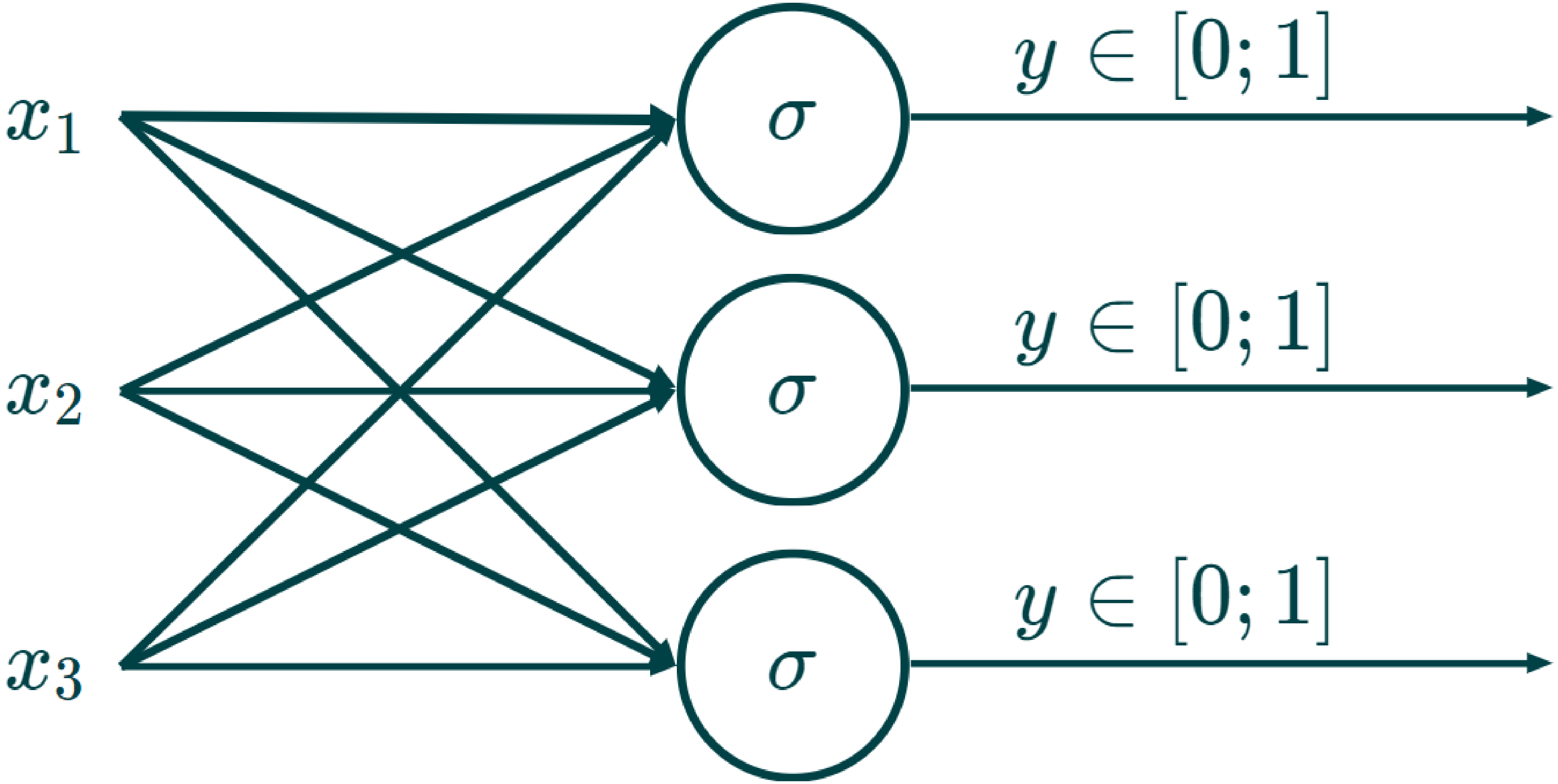
$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^{n} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

$$\frac{\partial L}{\partial w} = \frac{1}{n} \sum (\hat{y} - y)x \qquad \frac{\partial L}{\partial b} = \frac{1}{n} \sum (\hat{y} - y)$$

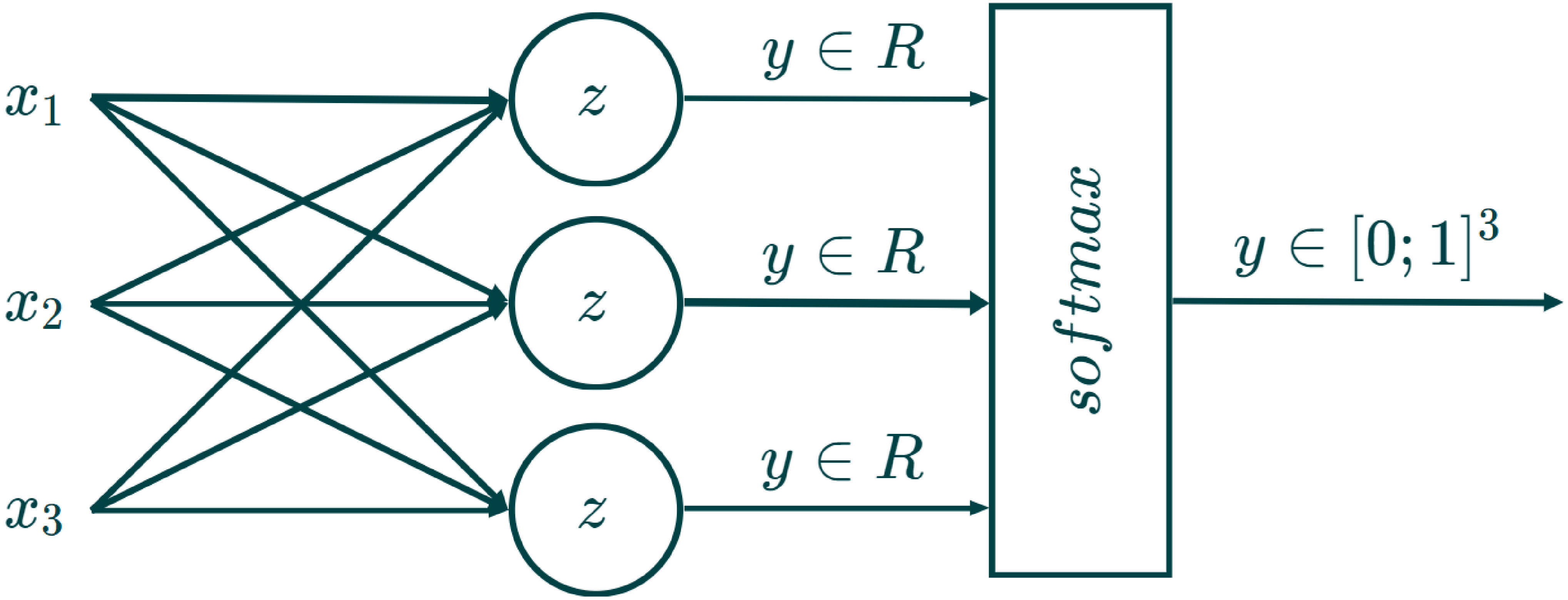
Бинарная классификация



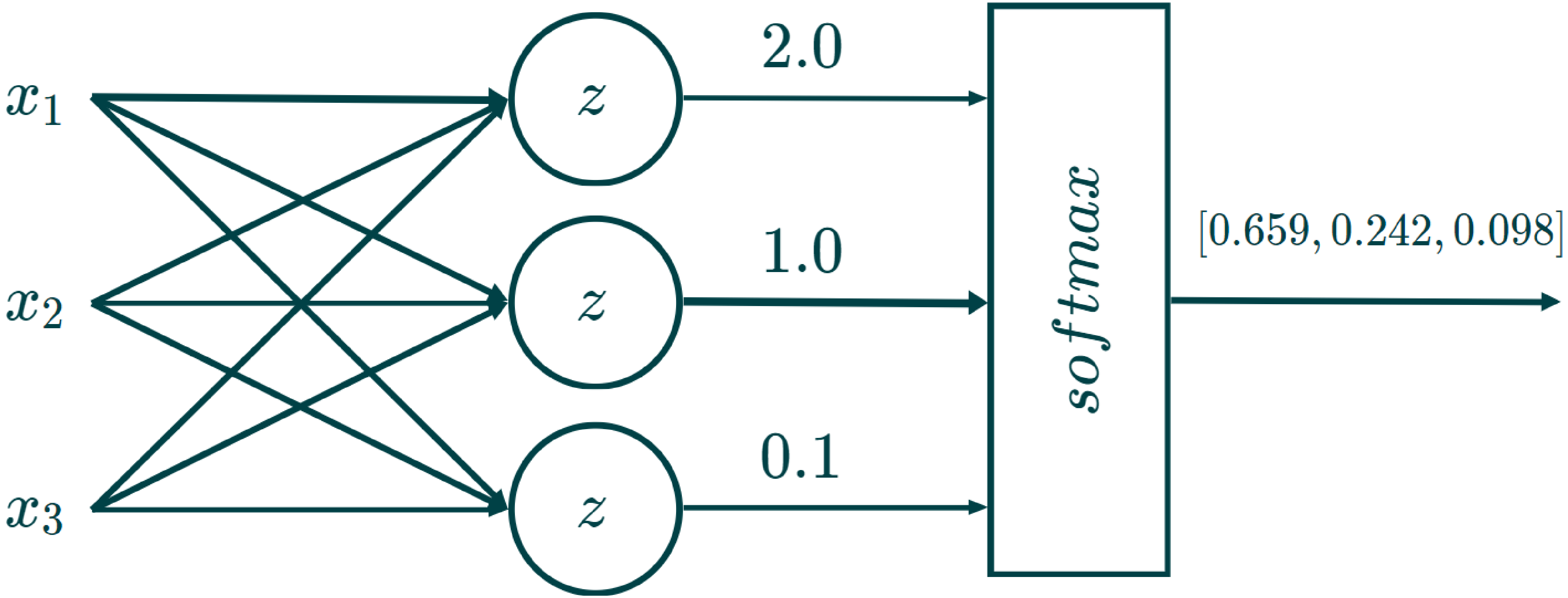
Многоклассовая классификация



Многоклассовая классификация



Многоклассовая классификация



Softmax

$$y = \frac{e^{y_i}}{\sum_{i=0}^{num\_of\_classes} e^{y_i}}$$

Матричные вычисления

$$y = \sum_{i=1}^{p} (x_i w_i) + b$$

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_p w_p + b$$

Как можно представить выполняемые вычисления?

Матричные вычисления

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_p w_p + b$$

$$y = [x_1 \ldots x_p] * \begin{bmatrix} w_1 \\ \ldots \\ w_p \end{bmatrix} + b$$

$$y = XW + b$$

Матричные вычисления

$$y = [x_1 \ldots x_p] * \begin{bmatrix} w_1 \\ \ldots \\ w_p \end{bmatrix} + b$$

$$\begin{bmatrix} y^{(1)} \\ \ldots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} {x_1}^{(1)} & \ldots & {x_p}^{(1)} \\ \ldots & \ldots & \ldots \\ {x_1}^{(n)} & \ldots & {x_p}^{(n)} \end{bmatrix} * \begin{bmatrix} w_1 \\ \ldots \\ w_p \end{bmatrix} + b$$

Градиентный спуск

Было:

$$w_1, \dots, w_p := 0$$
$$b := 0$$
$$for\ i\ in\ range(n\_iter):$$
$$\quad w_1 := w_1 - \alpha \frac{\partial L}{\partial w_1}$$
$$\quad \dots$$
$$\quad w_p := w_p - \alpha \frac{\partial L}{\partial w_p}$$
$$\quad b := b - \alpha \frac{\partial L}{\partial b}$$

Стало:

$$W := [0, \dots, 0]$$
$$b := 0$$
$$for\ i\ in\ range(n\_iter):$$
$$\quad W := W - \alpha \frac{\partial L}{\partial W}$$
$$\quad \dots$$
$$\quad b := b - \alpha \frac{\partial L}{\partial b}$$

Спасибо за внимание!

Конец Лекции 4