Часть 2: Строим первую нейронную сеть

Романов Михаил, Игорь Слинько

Обучающая выборка

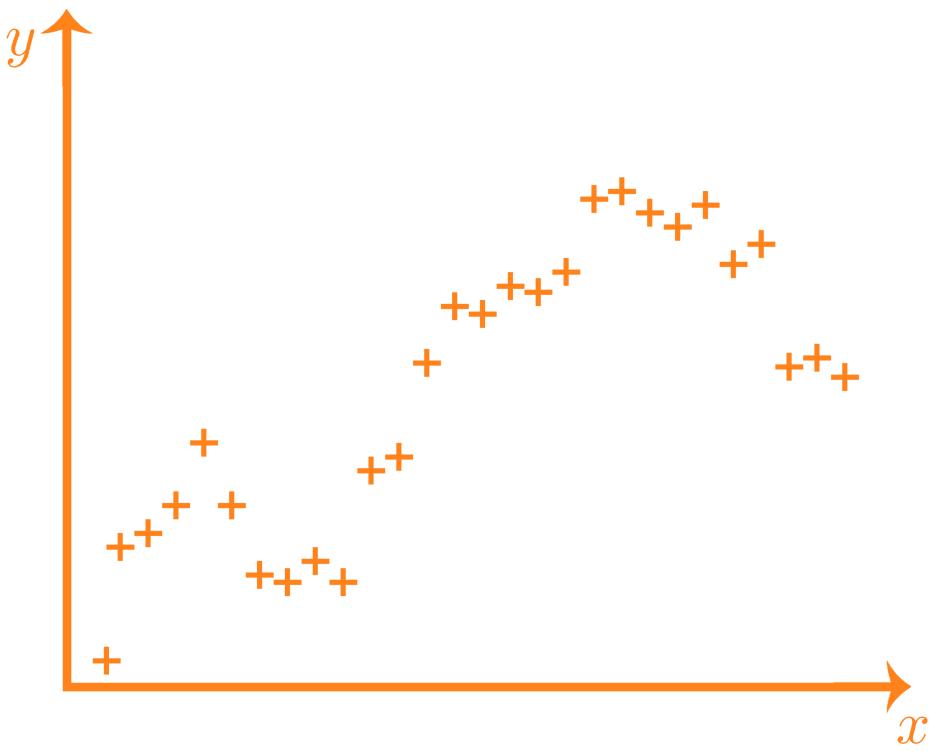
$\mathbf{x}_1$	$\mathbf{y}_1$
$\mathbf{x}_2$	$\mathbf{y}_2$
$\mathbf{x}_3$	$\mathbf{y}_3$
$\mathbf{x}_4$	$\mathbf{y}_4$
$\mathbf{x}_5$	$\mathbf{y}_5$
$\mathbf{x}_6$	$\mathbf{y}_6$
...	...
$\mathbf{x}_S$	$\mathbf{y}_S$

Обучающая выборка

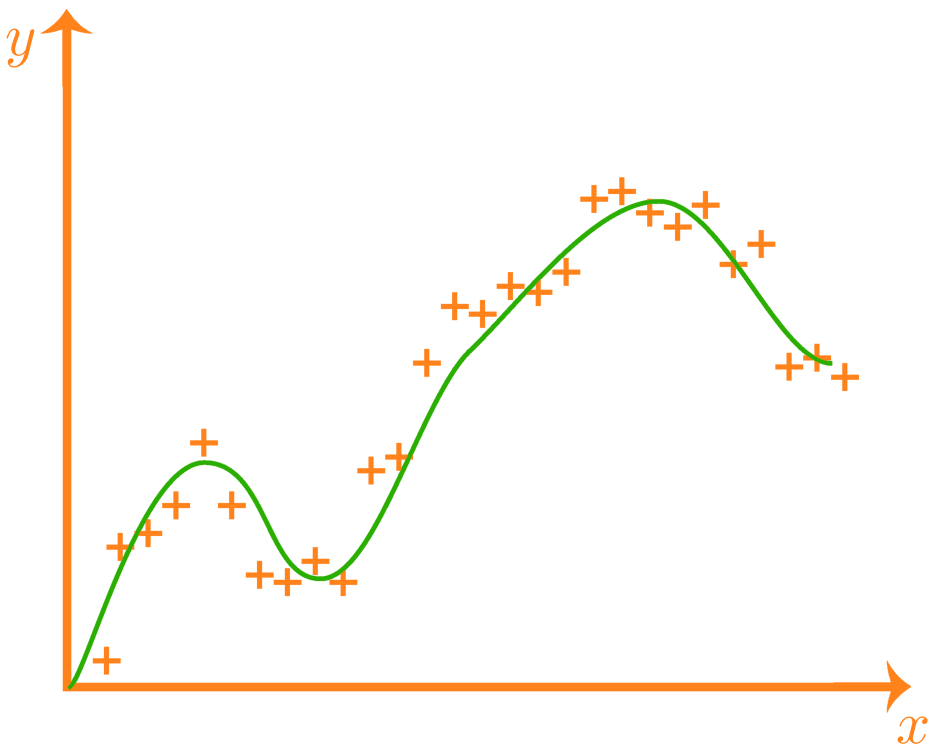
- Делится на train, test, valid

$\mathbf{x}_1$	$\mathbf{y}_1$
$\mathbf{x}_2$	$\mathbf{y}_2$
$\mathbf{x}_3$	$\mathbf{y}_3$
$\mathbf{x}_4$	$\mathbf{y}_4$
$\mathbf{x}_5$	$\mathbf{y}_5$
$\mathbf{x}_6$	$\mathbf{y}_6$
...	...
$\mathbf{x}_S$	$\mathbf{y}_S$

Пример задачи: восстановление зависимости

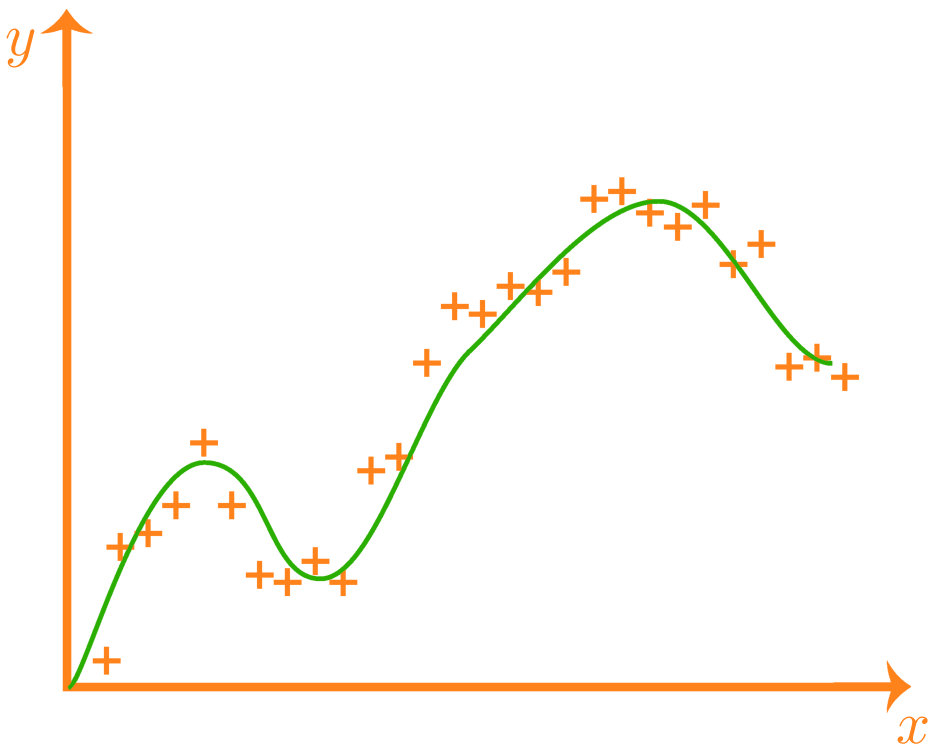


Пример задачи: восстановление зависимости



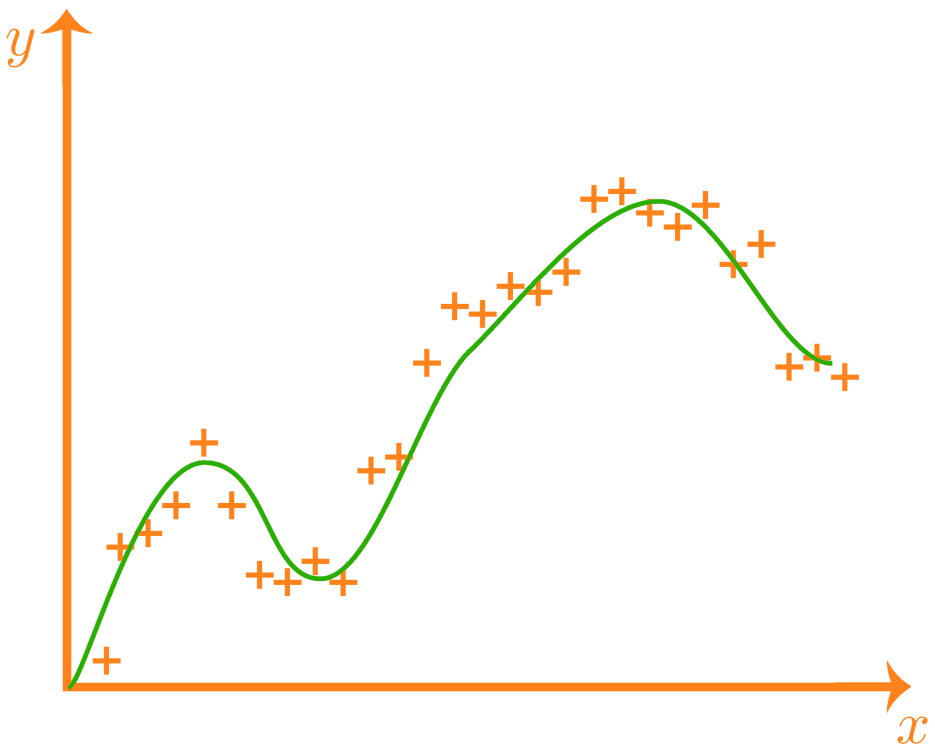
Пример задачи: восстановление зависимости

$$\mathbf{y}_s = f(\mathbf{x}_s)$$



Пример задачи: восстановление зависимости

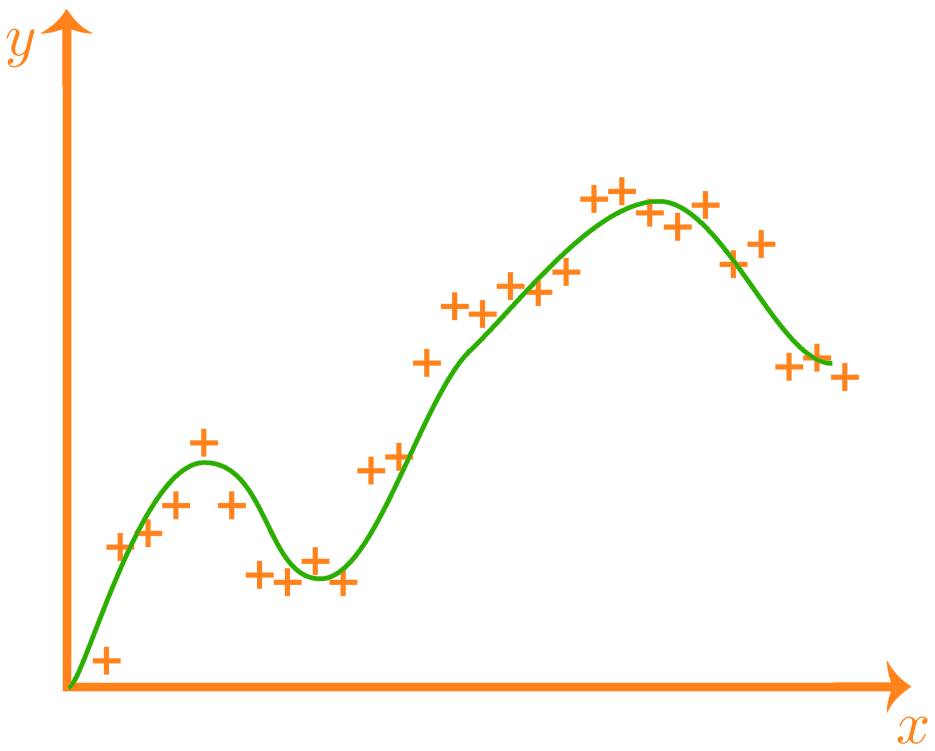
$$\mathbf{y}_s = f(\mathbf{x}_s) + \varepsilon_s$$



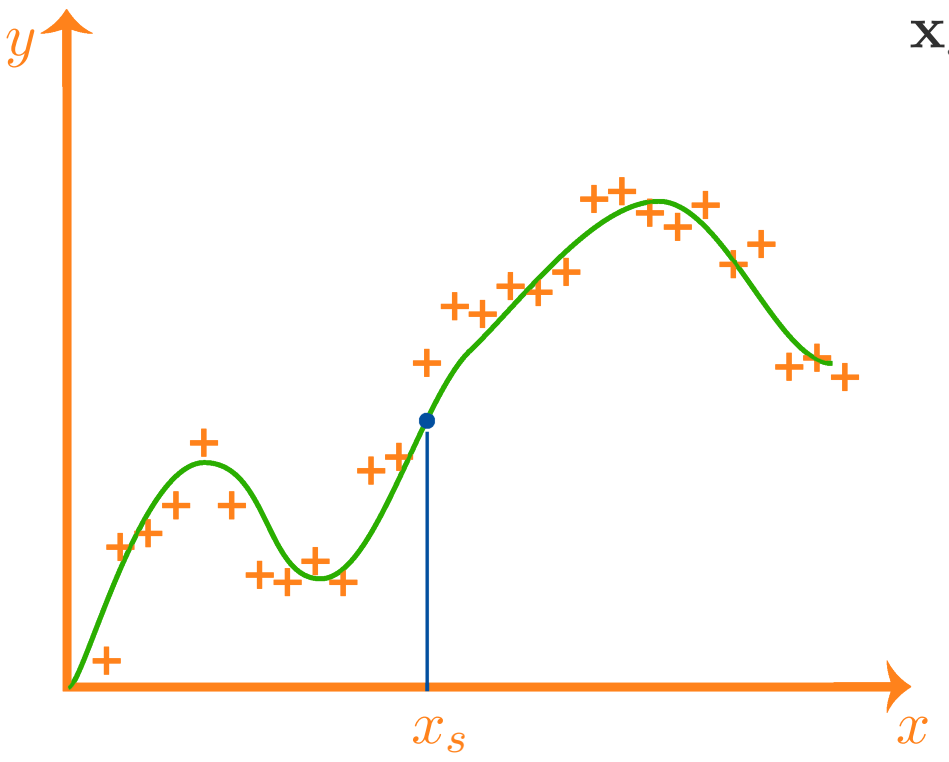
Пример задачи: восстановление зависимости

$$\mathbf{y}_s = f(\mathbf{x}_s) + \varepsilon_s$$

$f(\mathbf{x}_s)$ – скрытая зависимость



Пример задачи: восстановление зависимости

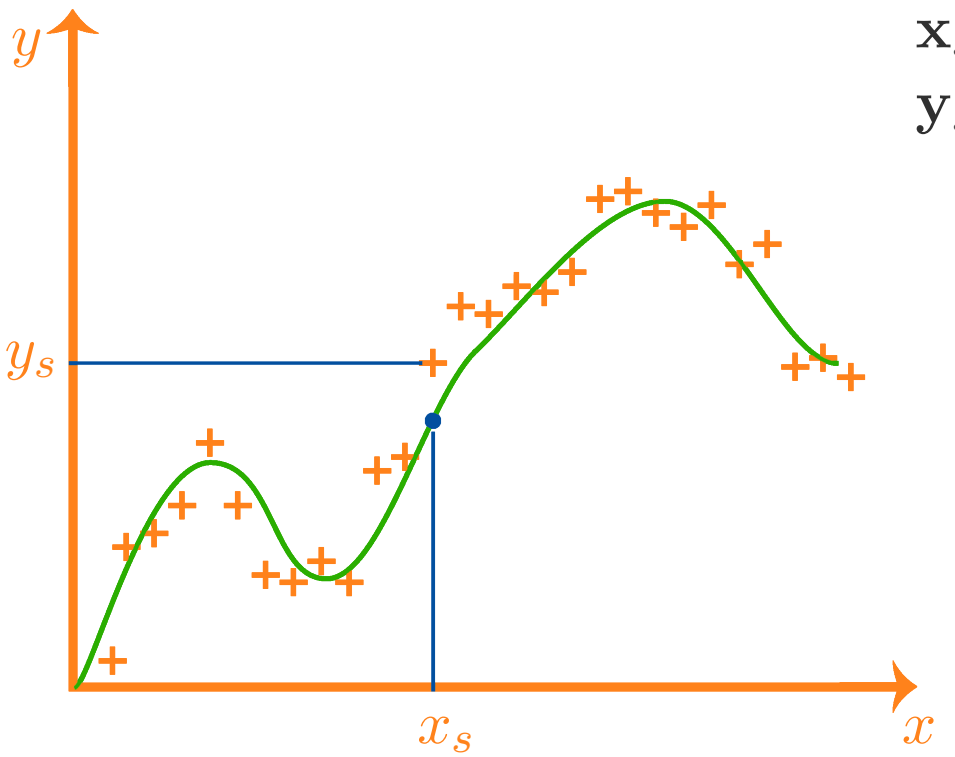


$\mathbf{y}_s = f(\mathbf{x}_s) + \varepsilon_s$

$f(\mathbf{x}_s)$ – скрытая зависимость

$\mathbf{x}_s$ – координата замера

Пример задачи: восстановление зависимости



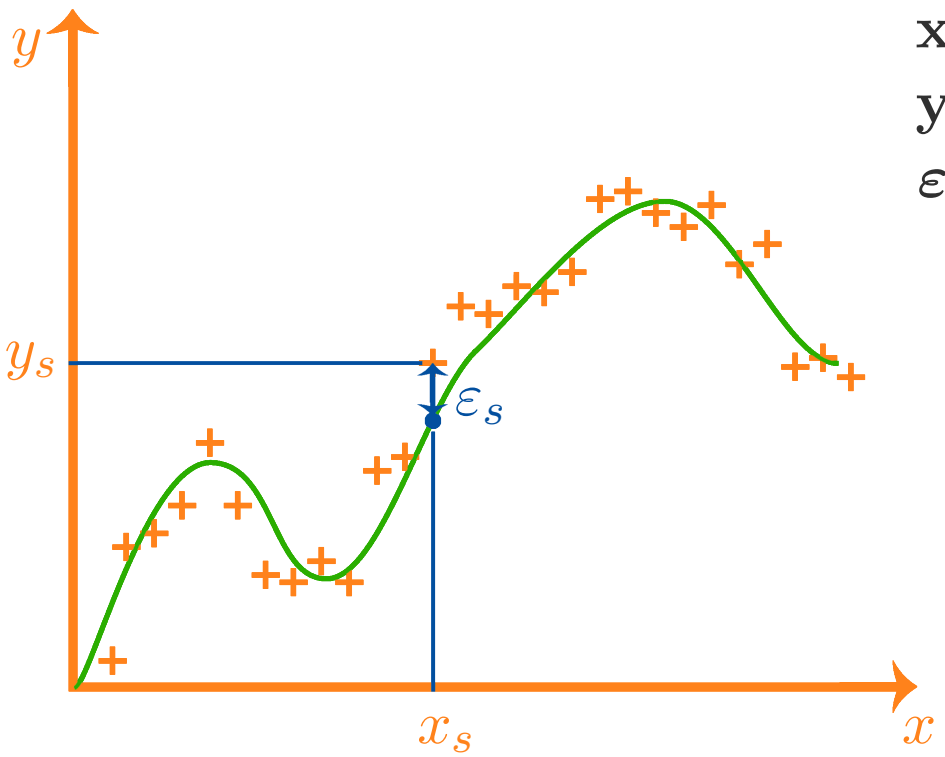
$\mathbf{y}_s = f(\mathbf{x}_s) + \varepsilon_s$

$f(\mathbf{x}_s)$ – скрытая зависимость

$\mathbf{x}_s$ – координата замера

$\mathbf{y}_s$ – измеренное значение

Пример задачи: восстановление зависимости



$\mathbf{y}_s = f(\mathbf{x}_s) + \varepsilon_s$

$f(\mathbf{x}_s)$ – скрытая зависимость

$\mathbf{x}_s$ – координата замера

$\mathbf{y}_s$ – измеренное значение

ε_s – шум

Starter Kit

- Архитектура нейронной сети

Starter Kit

- Архитектура нейронной сети
- Функция потерь

Starter Kit

- Архитектура нейронной сети
- Функция потерь
- Метод оптимизации

Starter Kit

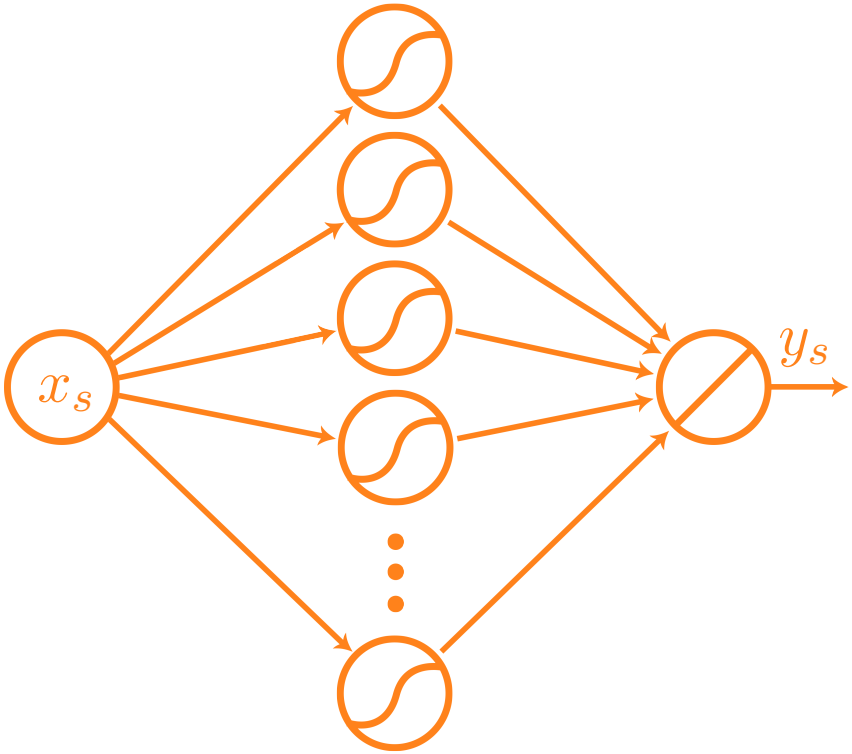
- Архитектура нейронной сети
- Функция потерь
- Метод оптимизации
- Метрики

Нейронная сеть

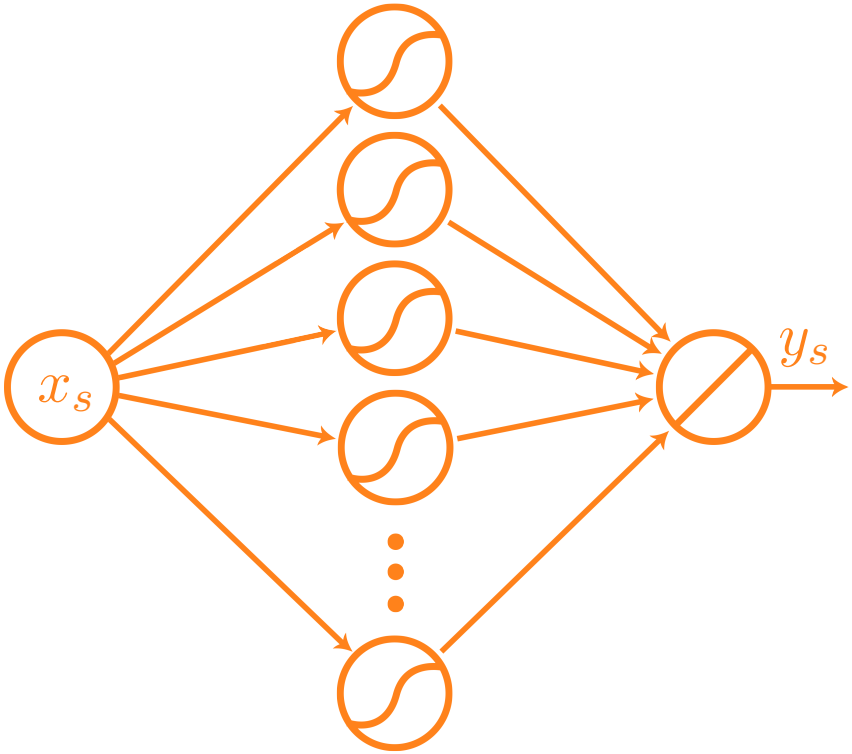
Нейронная сеть

Нейронная сеть

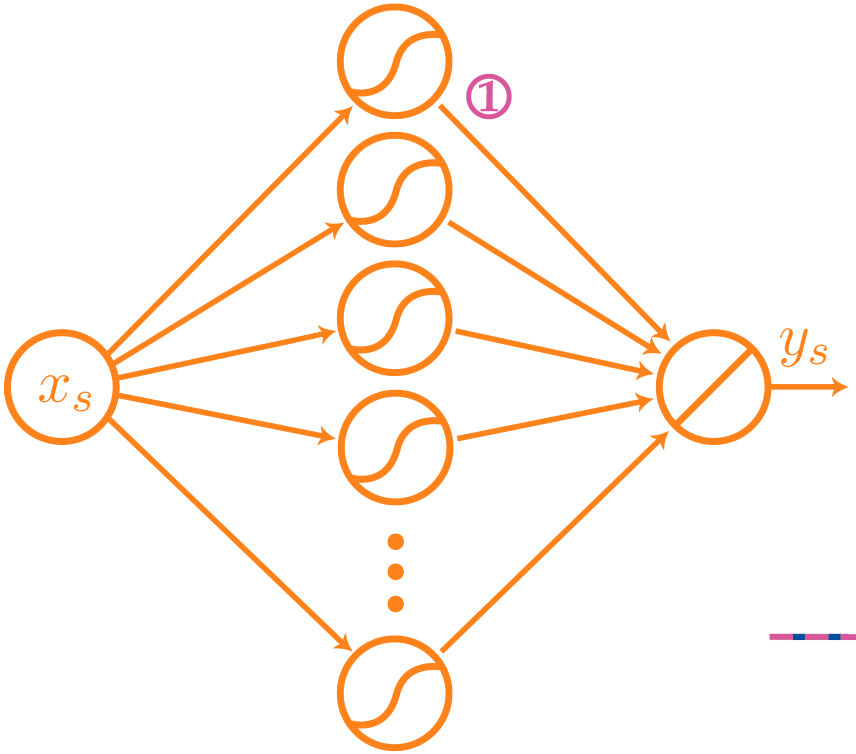
Нейронная сеть

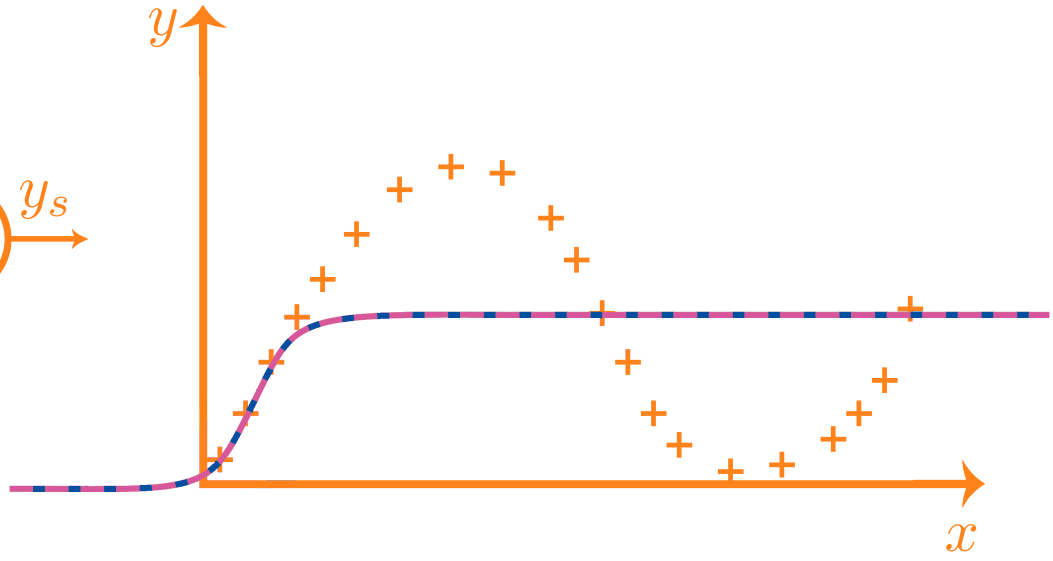


Нейронная сеть

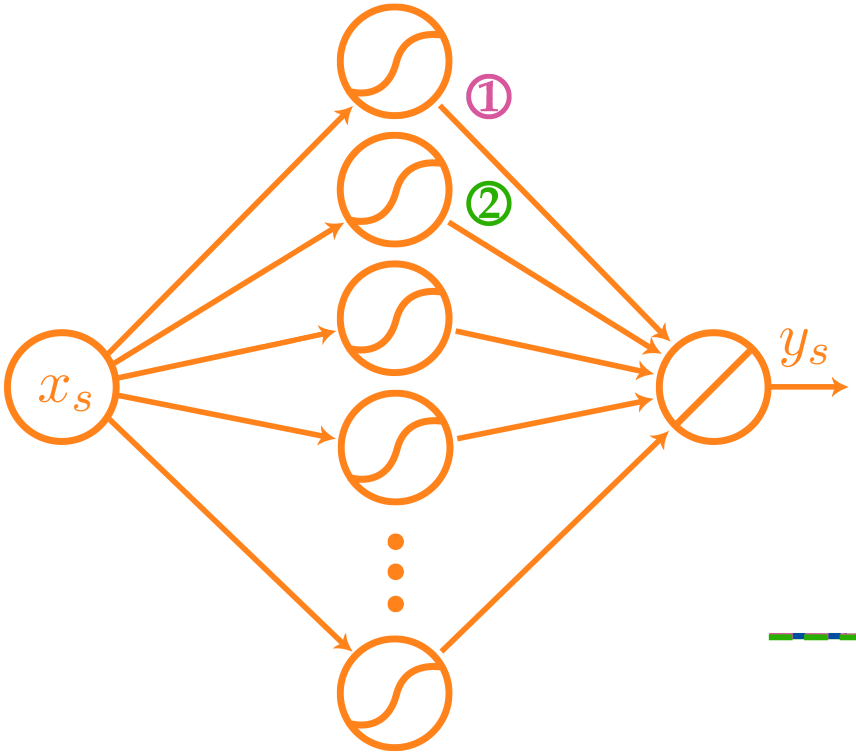


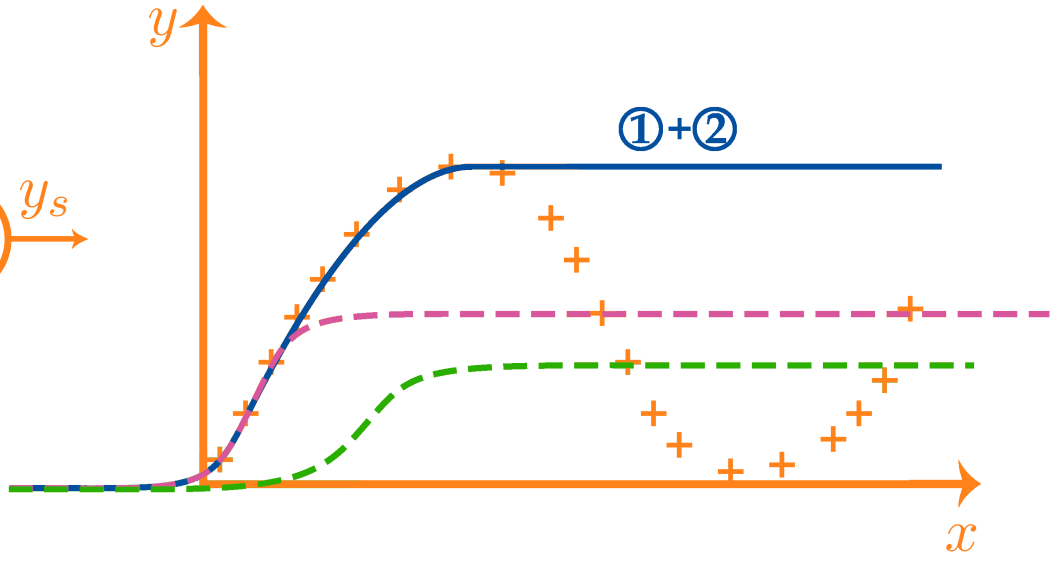
Нейронная сеть



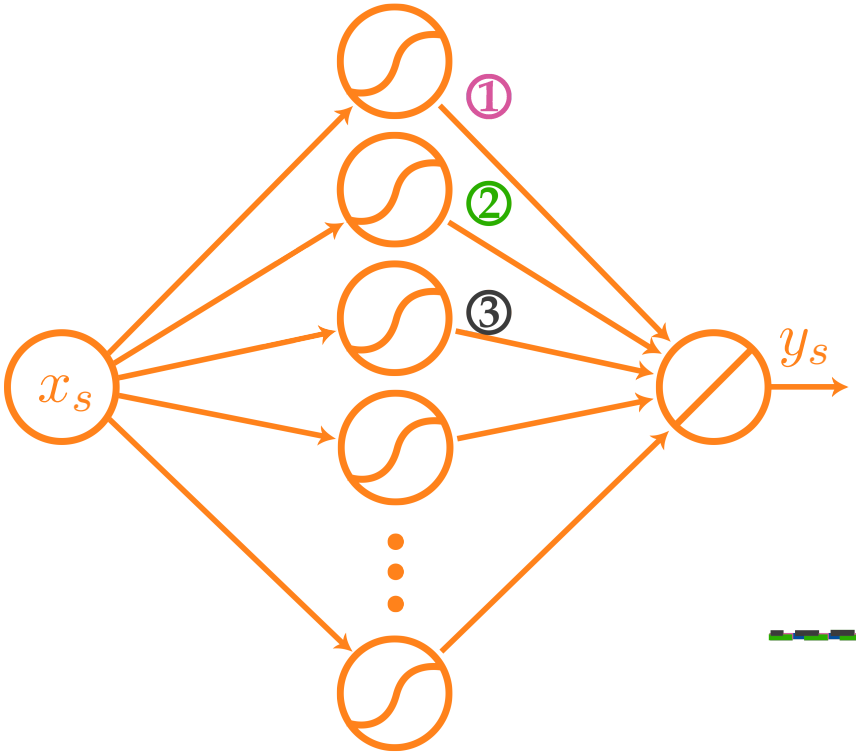


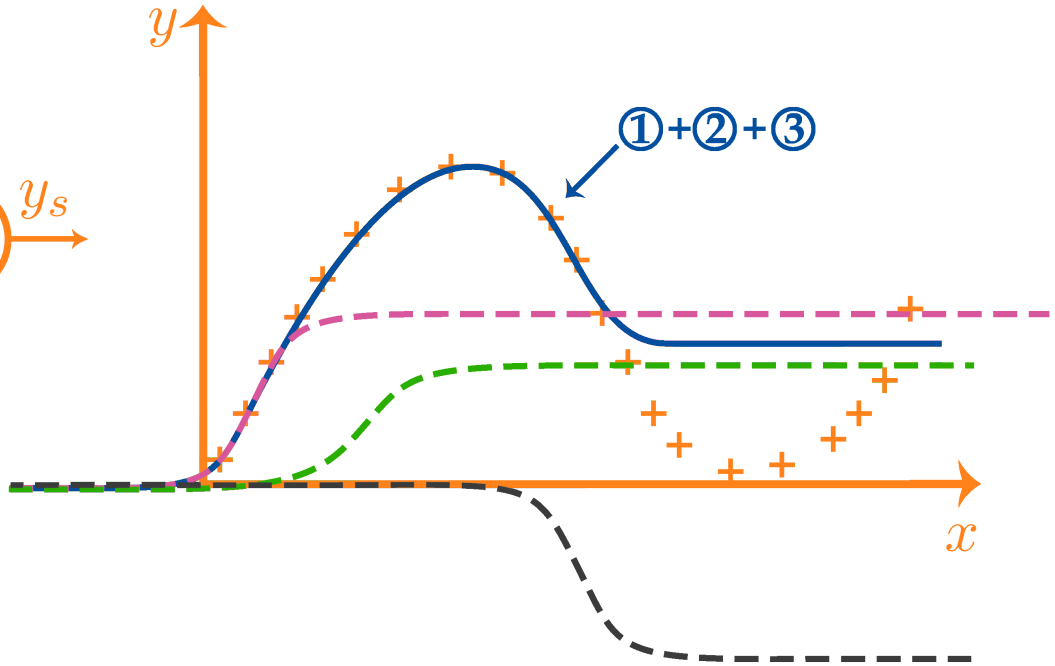
Нейронная сеть



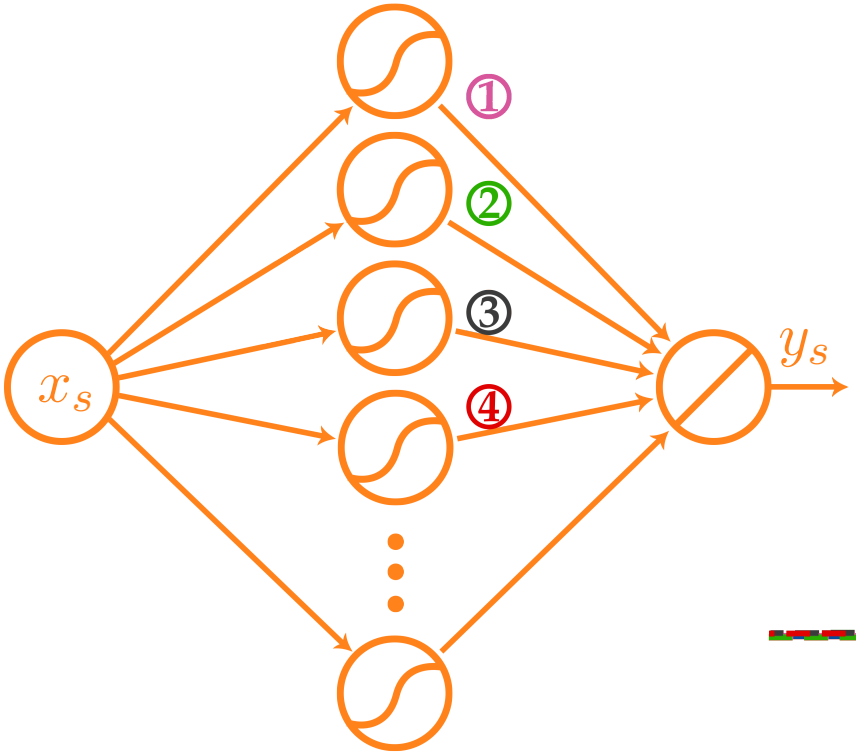


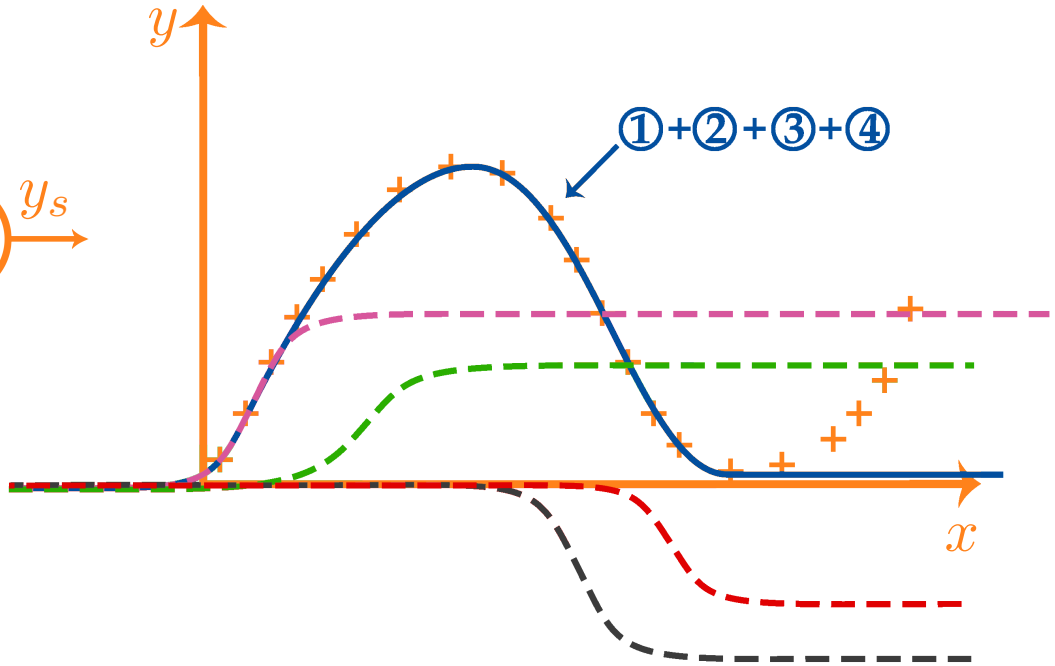
Нейронная сеть



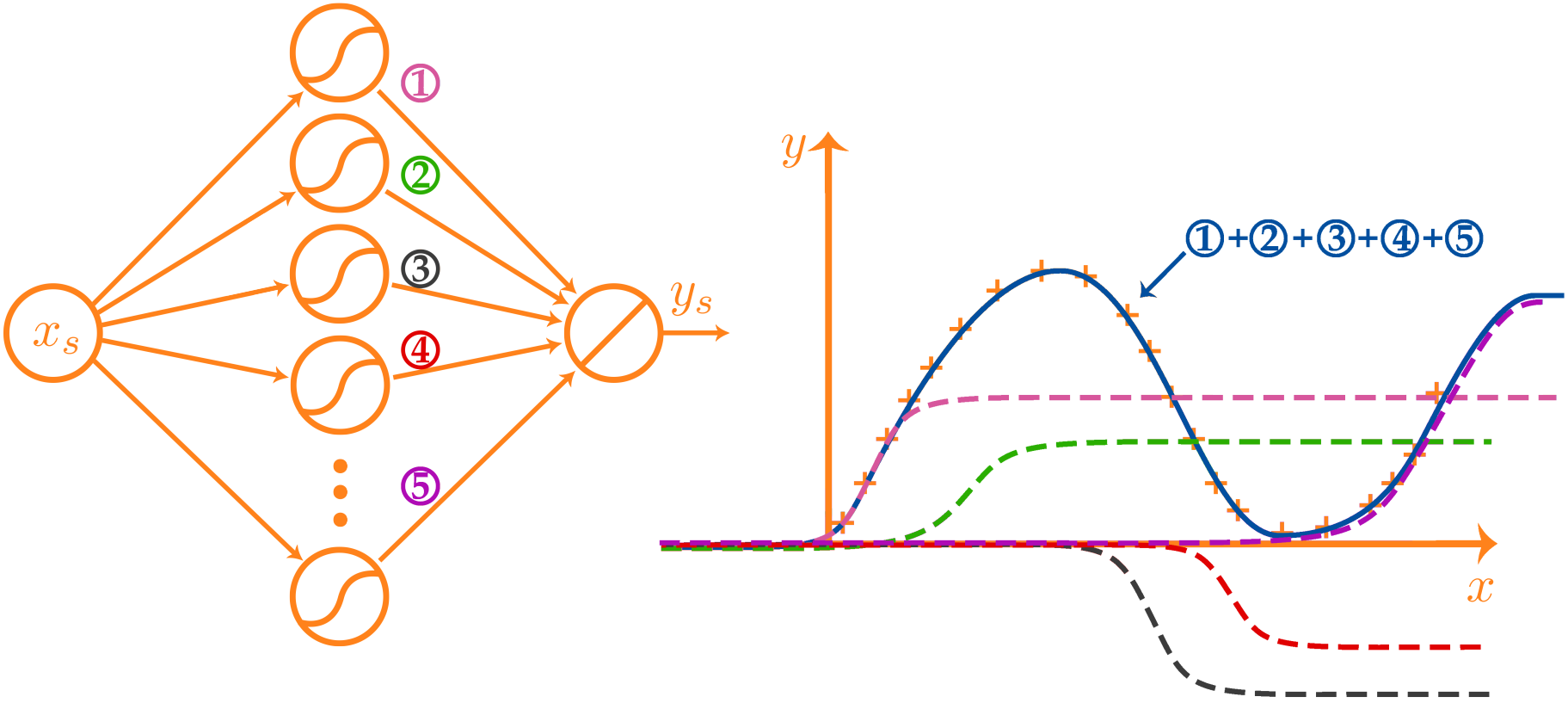


Нейронная сеть

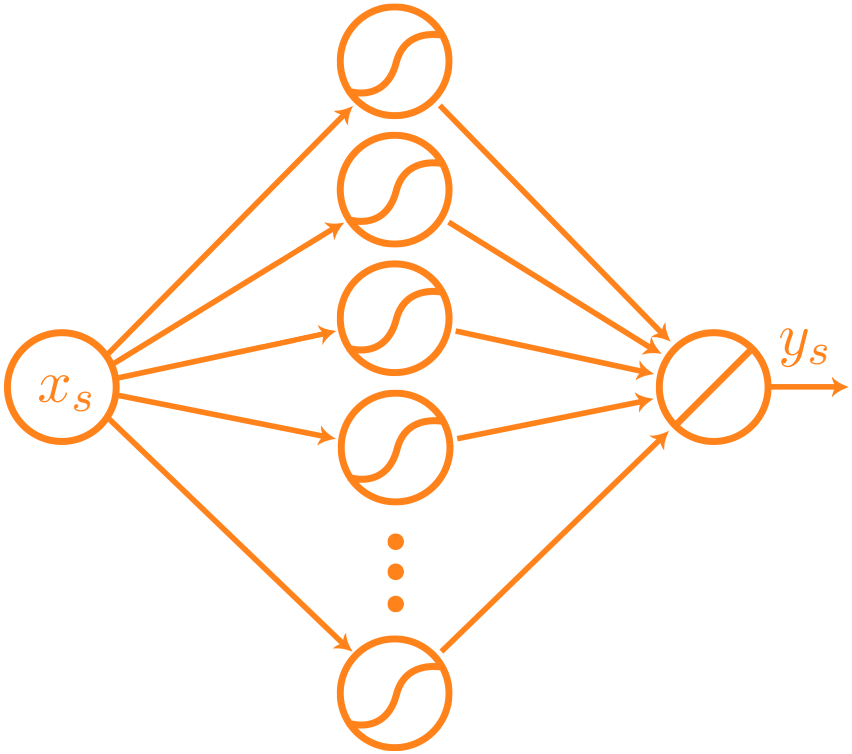




Нейронная сеть

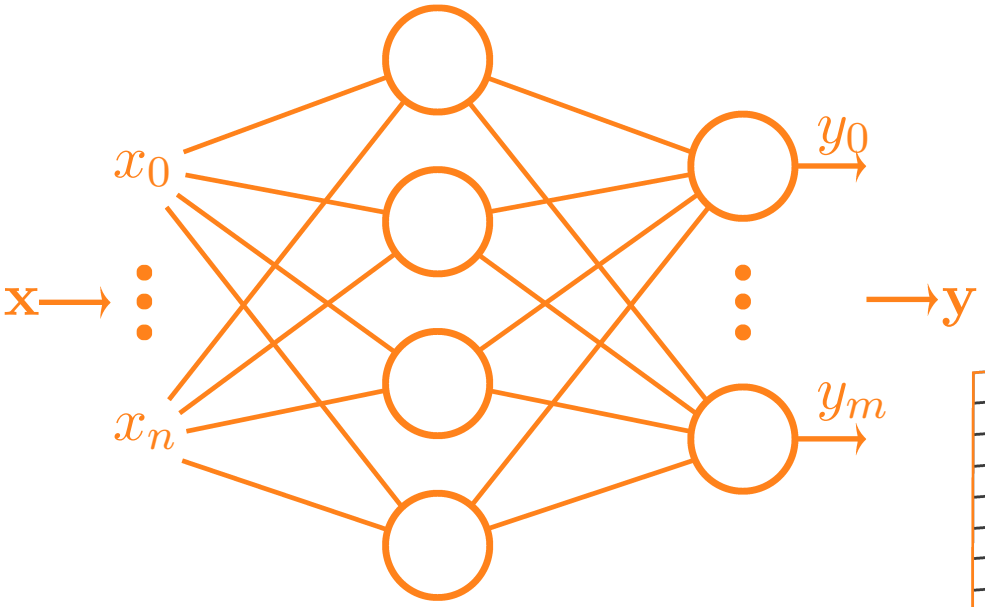


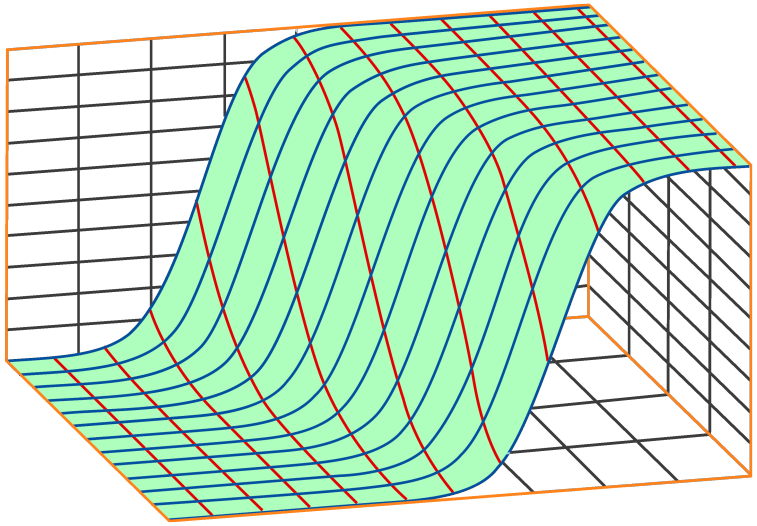
Нейронная сеть



При помощи линейной комбинации сигмоид можно с любой точностью приблизить любую ограниченную функцию с не более, чем счетным числом разрывов

Нейронная сеть





Функция потерь

Возьмем самую простую функцию потерь: среднеквадратичную ошибку или Mean Squared Error (MSE) для нашей задачи.

Функция потерь

Возьмем самую простую функцию потерь: среднеквадратичную ошибку или Mean Squared Error (MSE) для нашей задачи.

$$MSE = \frac{1}{S} \sum_{s=1}^{S} (y_s - \widetilde{y}_s)^2$$

Функция потерь

Возьмем самую простую функцию потерь: среднеквадратичную ошибку или Mean Squared Error (MSE) для нашей задачи.

$$MSE = \frac{1}{S}\sum_{s=1}^{S}(y_s - \widetilde{y}_s)^2$$

Результат работы сети

Целевые значения

Метрики

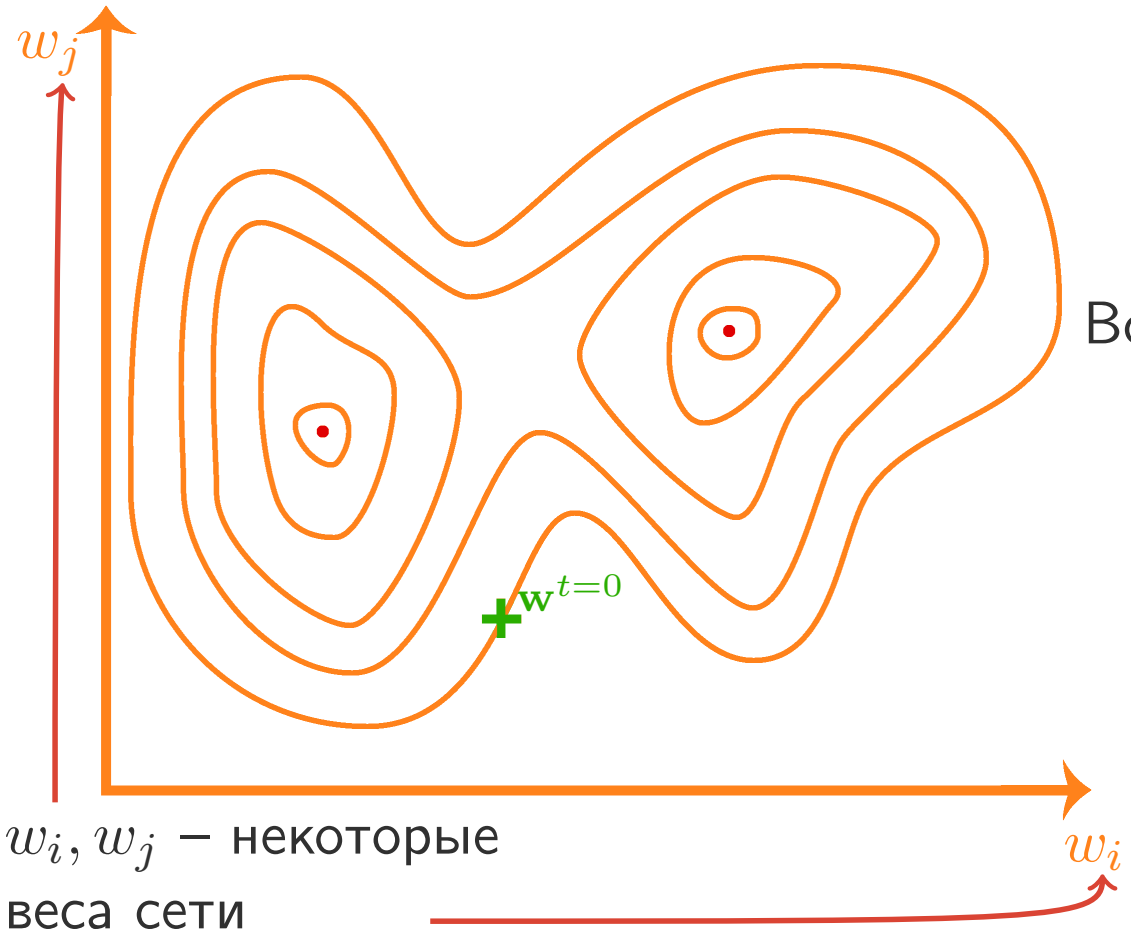
- Метриками можно выбрать любые функции, том числе функции потерь
- Например, точность, precision, recall, ROC AUC, top N
- Сделаем метрику «доля точек с отклонением меньше заданного»

Метрики

- Метриками можно выбрать любые функции, том числе функции потерь
- Например, точность, precision, recall, ROC AUC, top N
- Сделаем метрику «доля точек с отклонением меньше заданного»

$$Acc_\varepsilon = \frac{1}{S}\sum_{s=1}^{S}\left[|y_s - \widetilde{y}_s| < \varepsilon\right]$$

Настройка сети: градиентный спуск

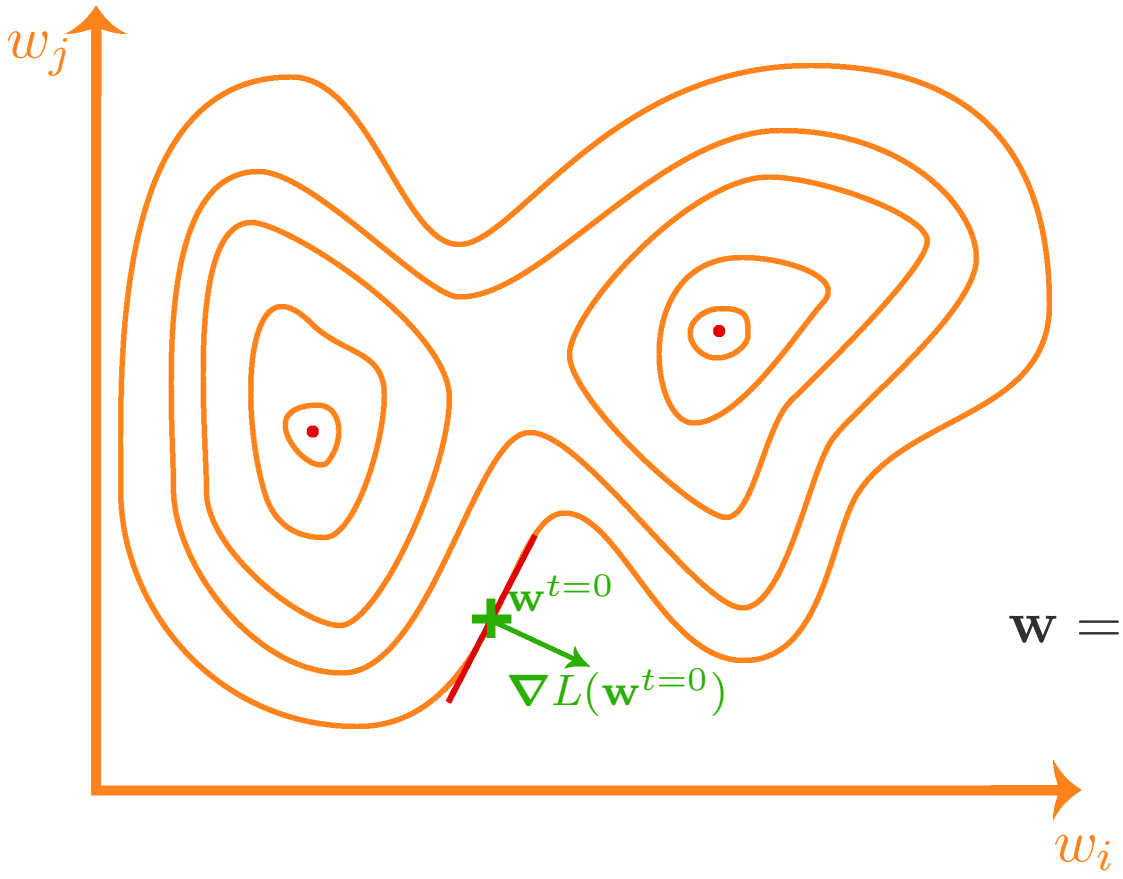


w_i, w_j – некоторые веса сети

Все веса сети:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix}$$

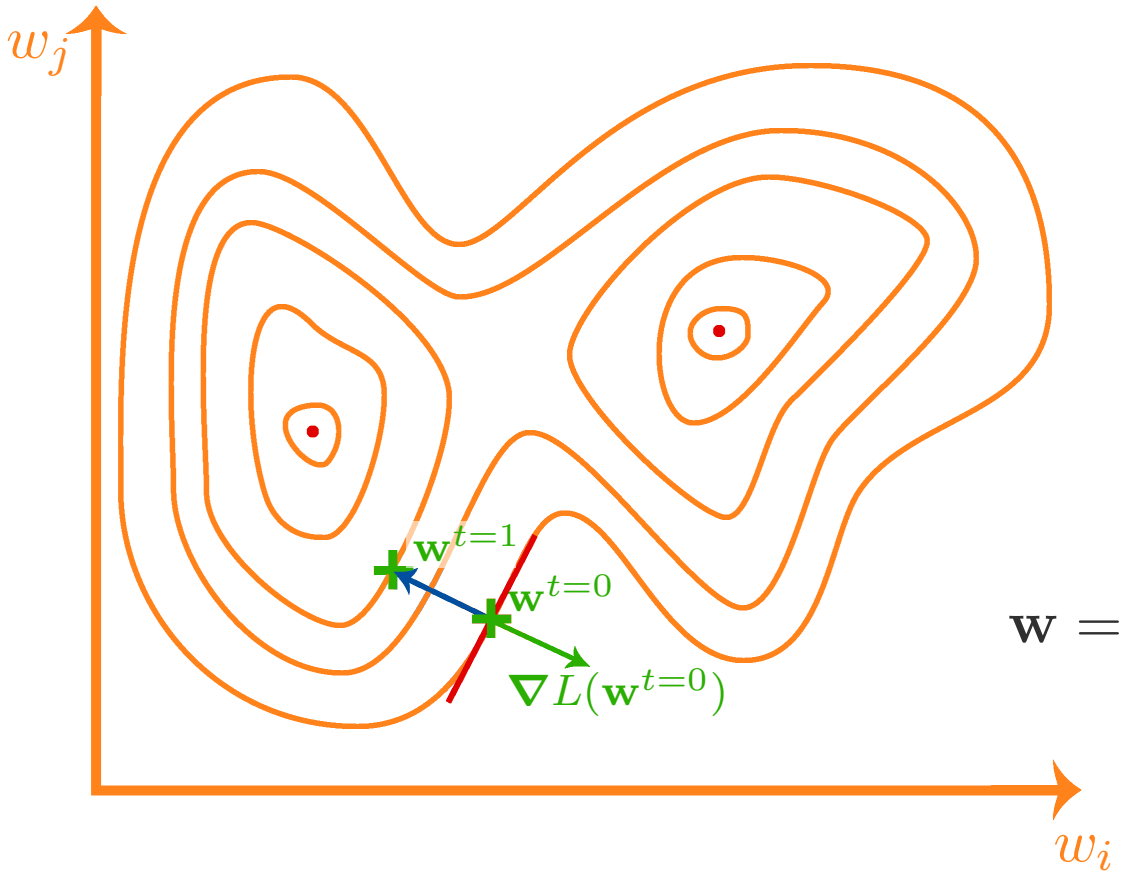
Настройка сети: градиентный спуск



$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix} \qquad \nabla L = \begin{bmatrix} \dfrac{\partial L}{\partial w_0} \\ \dfrac{\partial L}{\partial w_1} \\ \vdots \\ \dfrac{\partial L}{\partial w_{P-1}} \end{bmatrix}$$

Настройка сети: градиентный спуск

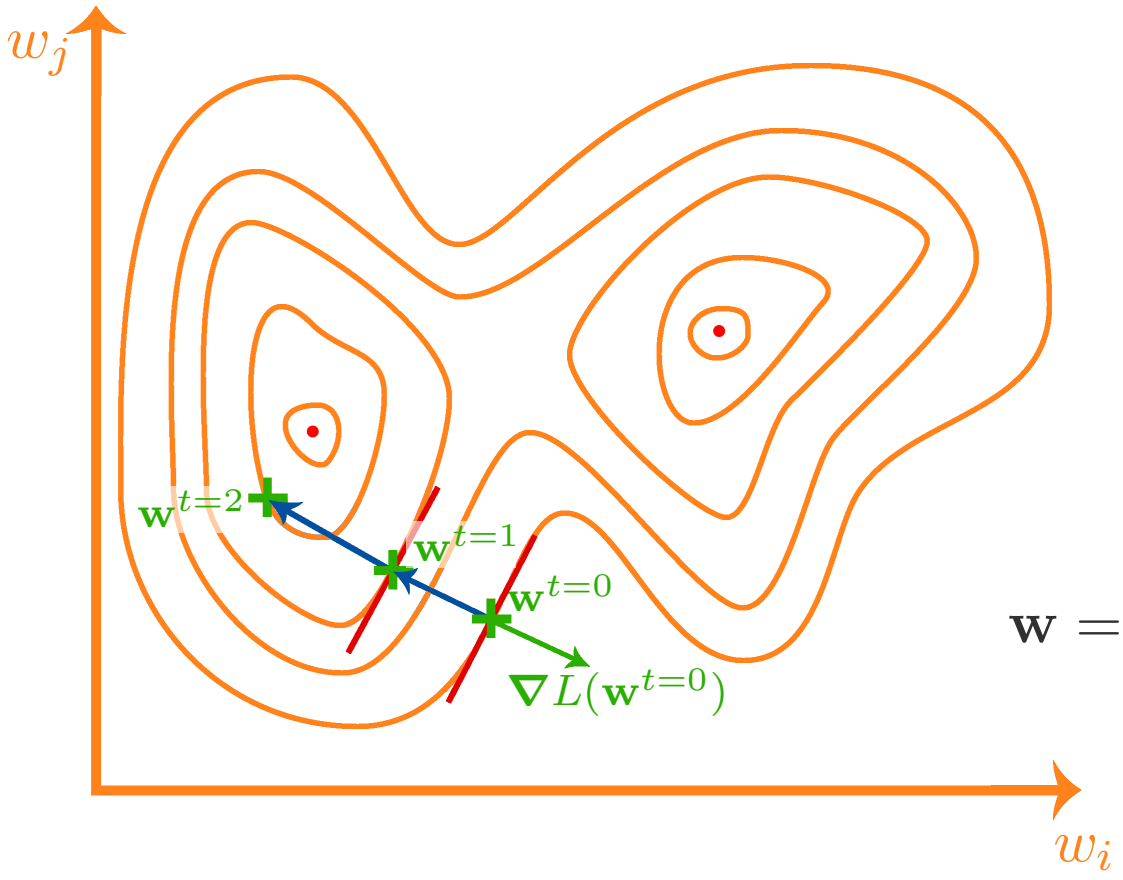
$$\mathbf{w}^{t=0}$$
$$\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=0})$$



$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix} \qquad \boldsymbol{\nabla} L = \begin{bmatrix} \dfrac{\partial L}{\partial w_0} \\ \dfrac{\partial L}{\partial w_1} \\ \vdots \\ \dfrac{\partial L}{\partial w_{P-1}} \end{bmatrix}$$

Настройка сети: градиентный спуск

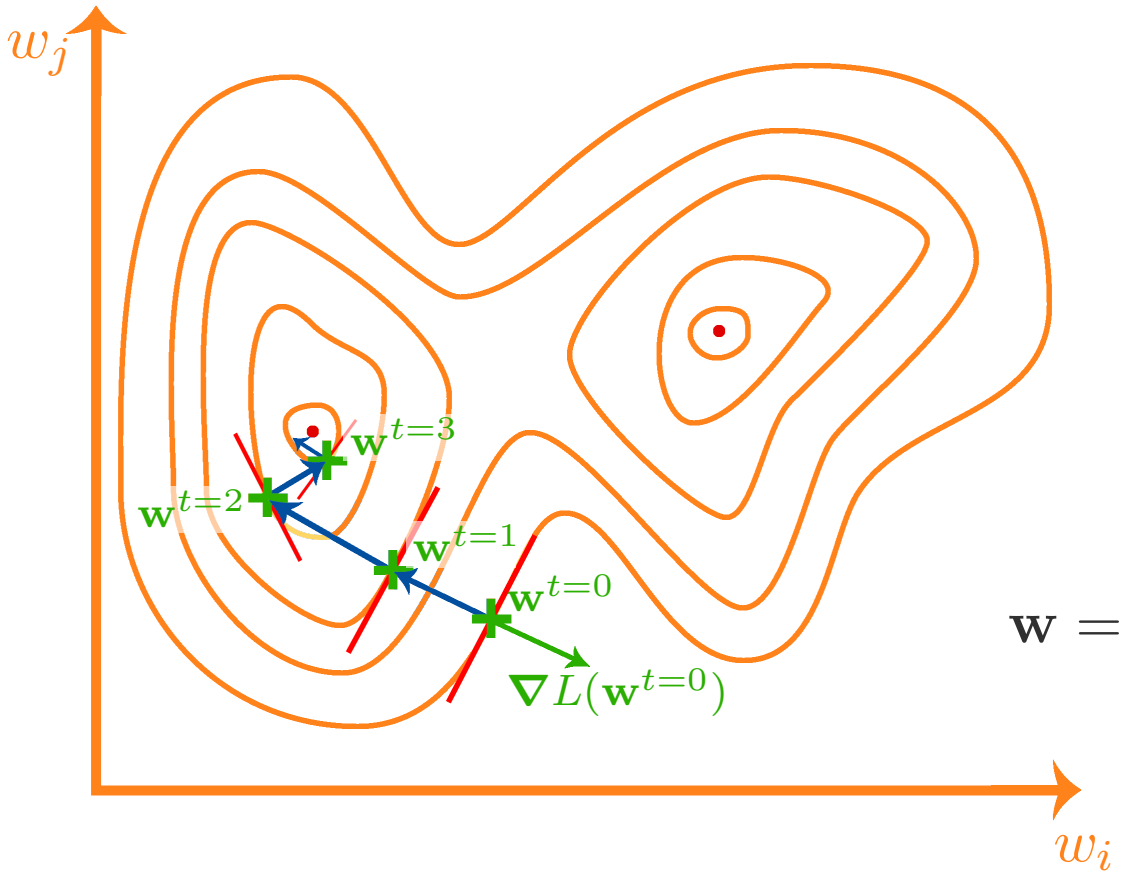
$$\mathbf{w}^{t=0}$$
$$\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=0})$$
$$\mathbf{w}^{t=2} = \mathbf{w}^{t=1} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=1})$$



$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix} \qquad \boldsymbol{\nabla} L = \begin{bmatrix} \dfrac{\partial L}{\partial w_0} \\ \dfrac{\partial L}{\partial w_1} \\ \vdots \\ \dfrac{\partial L}{\partial w_{P-1}} \end{bmatrix}$$

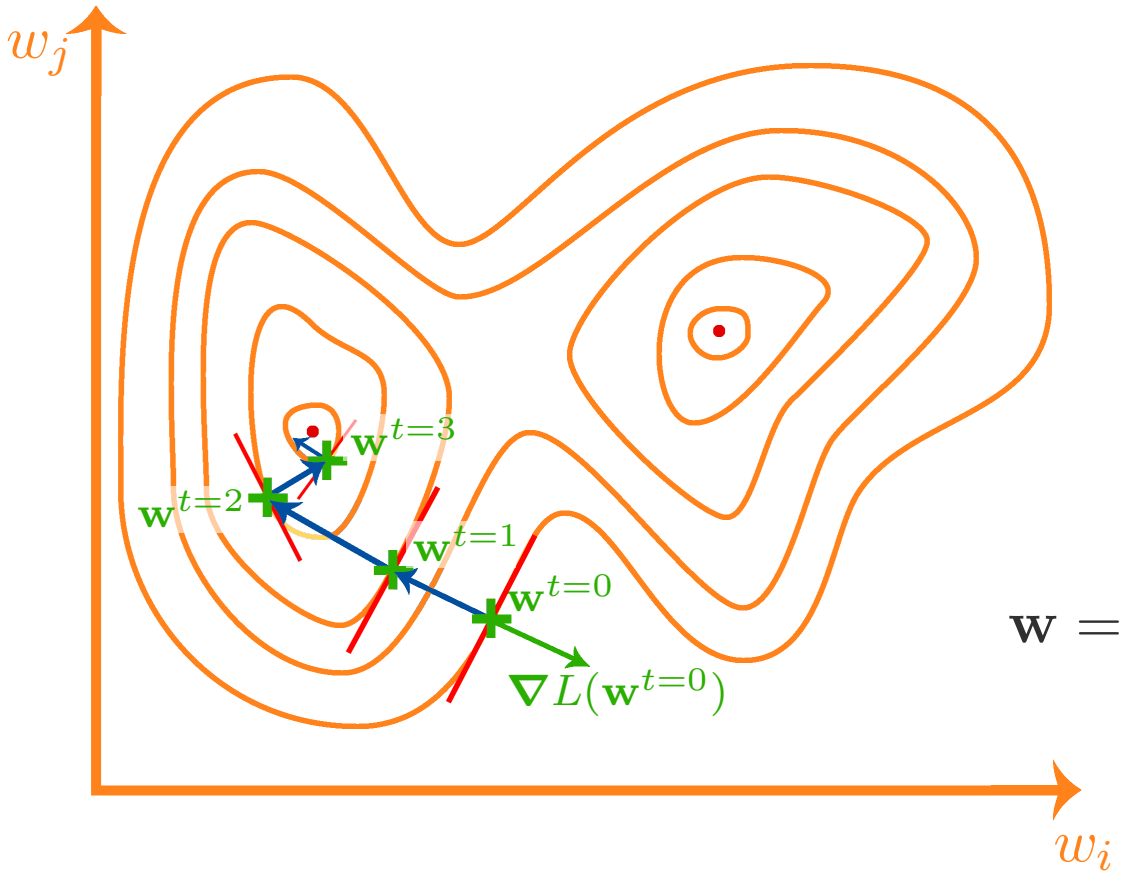
Настройка сети: градиентный спуск

$$
\begin{aligned}
&\mathbf{w}^{t=0} \\
&\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=0}) \\
&\mathbf{w}^{t=2} = \mathbf{w}^{t=1} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=1}) \\
&\mathbf{w}^{t=3} = \mathbf{w}^{t=2} - \alpha \boldsymbol{\nabla} L(\mathbf{w}^{t=2})
\end{aligned}
$$



$$
\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix} \qquad \boldsymbol{\nabla} L = \begin{bmatrix} \dfrac{\partial L}{\partial w_0} \\ \dfrac{\partial L}{\partial w_1} \\ \vdots \\ \dfrac{\partial L}{\partial w_{P-1}} \end{bmatrix}
$$

Настройка сети: градиентный спуск



$$
\begin{aligned}
&\mathbf{w}^{t=0} \\
&\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \nabla L(\mathbf{w}^{t=0}) \\
&\mathbf{w}^{t=2} = \mathbf{w}^{t=1} - \alpha \nabla L(\mathbf{w}^{t=1}) \\
&\mathbf{w}^{t=3} = \mathbf{w}^{t=2} - \alpha \nabla L(\mathbf{w}^{t=2}) \\
&\dots \\
&\mathbf{w}^{T} = \mathbf{w}^{T-1} - \alpha \nabla L(\mathbf{w}^{T-1})
\end{aligned}
$$

$$
\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{P-1} \end{bmatrix} \qquad \nabla L = \begin{bmatrix} \dfrac{\partial L}{\partial w_0} \\ \dfrac{\partial L}{\partial w_1} \\ \vdots \\ \dfrac{\partial L}{\partial w_{P-1}} \end{bmatrix}
$$

Настройка сети: градиент функции потерь

Для того, чтобы можно было осуществить такой спуск, нам и нужна дифференцируемая функция потерь

$$MSE = \frac{1}{S}\sum_{s=1}^{S}(y_s - \widetilde{y}_s)^2$$

Настройка сети: градиент функции потерь

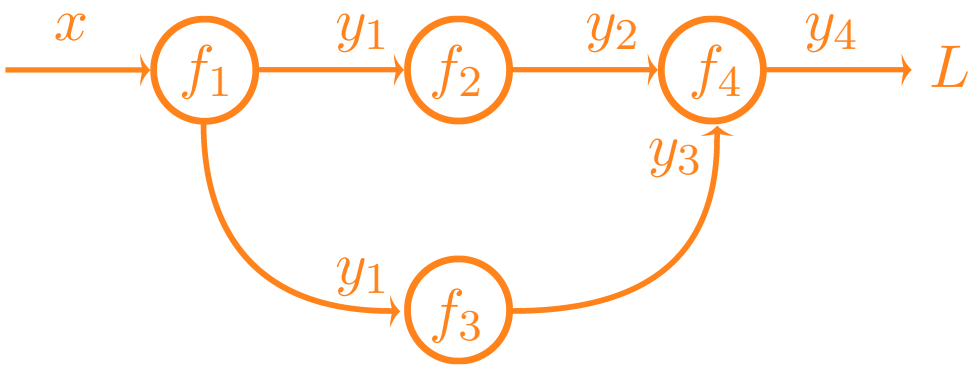
Для того, чтобы можно было осуществить такой спуск, нам и нужна дифференцируемая функция потерь

$$MSE = \frac{1}{S}\sum_{s=1}^{S}(y_s - \widetilde{y}_s)^2$$

$$\frac{\partial}{\partial y_s}MSE = -\frac{2}{S}(y_s - \widetilde{y}_s)$$

Рассчитываем градиенты: правило цепочки

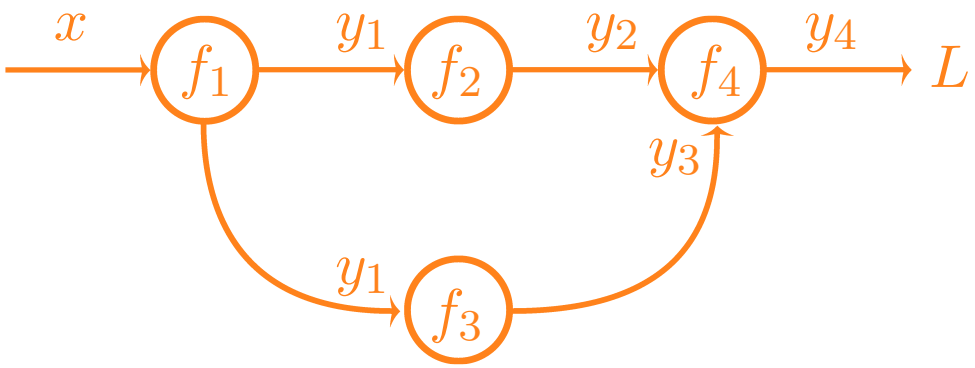
Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



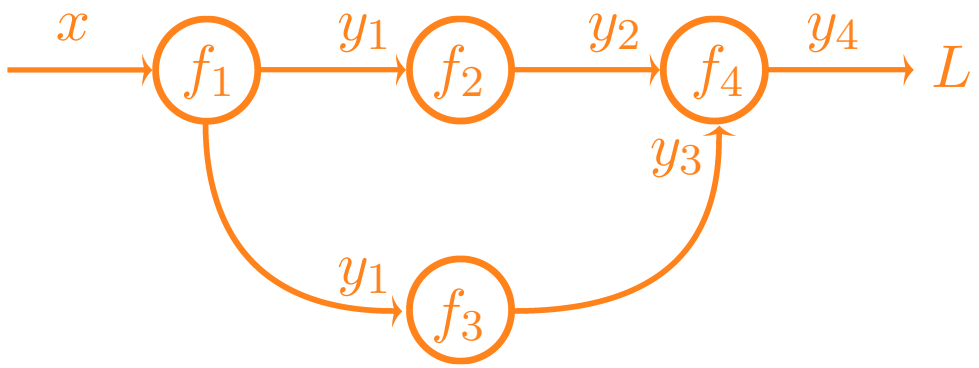
$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

Правило дифференцирования:

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

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

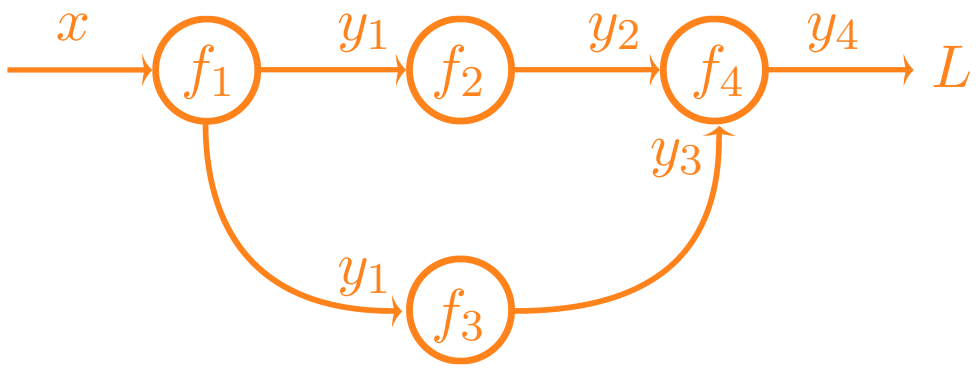
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_2} =$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

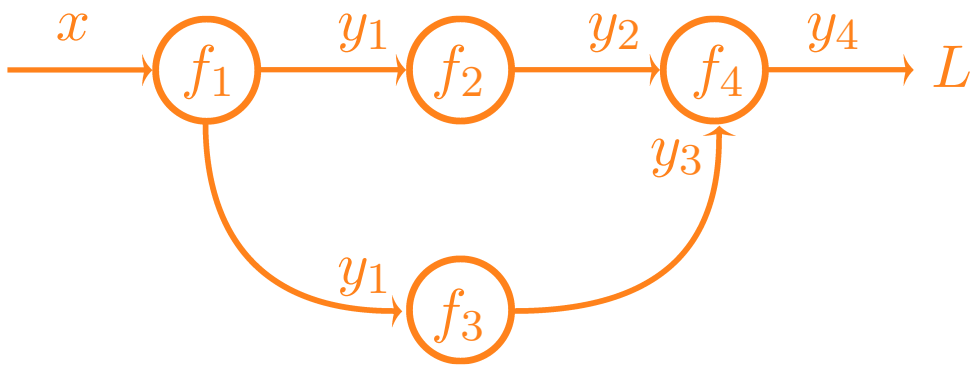
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y_4}$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

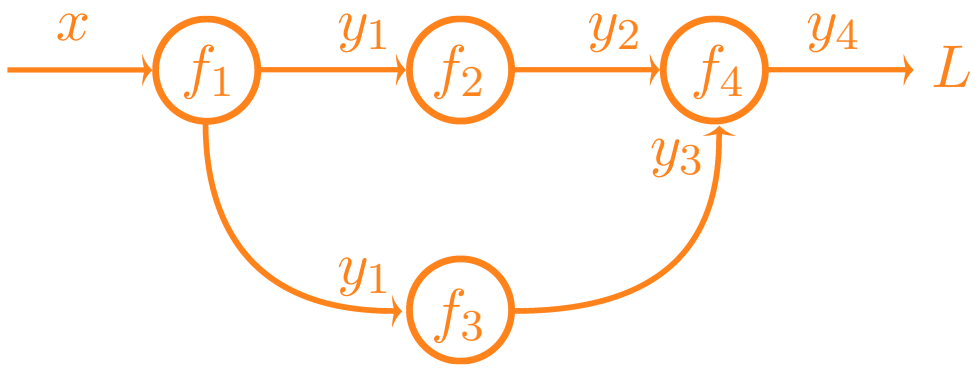
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y_4} \cdot \frac{\partial f_4}{\partial y_2}$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

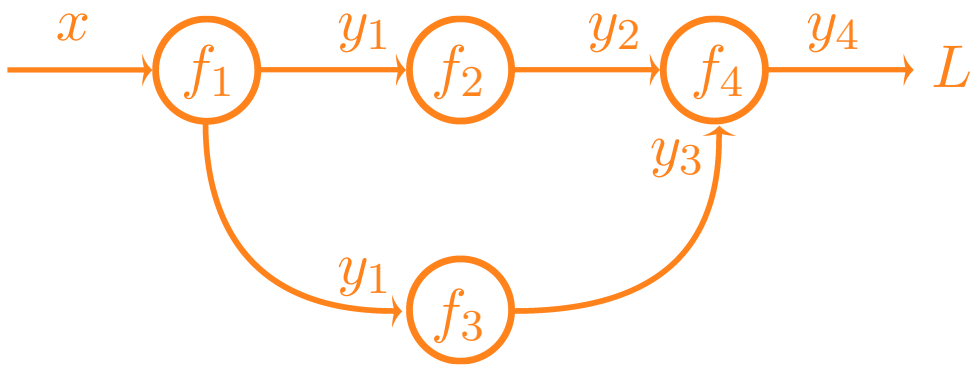
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y_4} \cdot \frac{\partial f_4}{\partial y_2} \cdot \frac{\partial f_2}{\partial w_2}$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

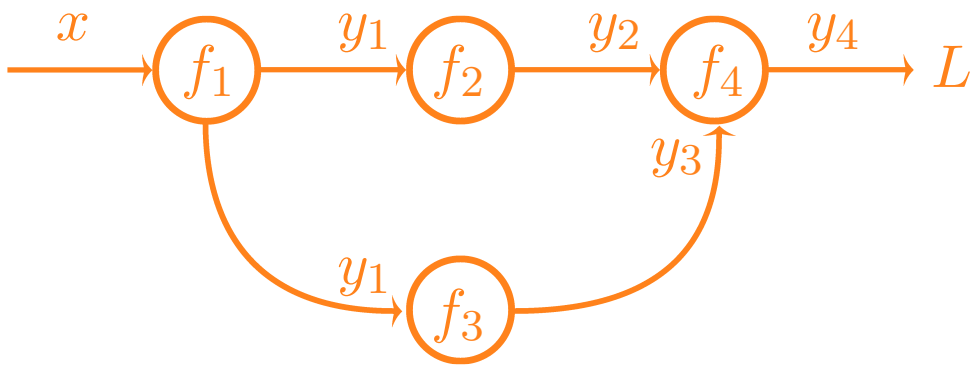
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_1} =$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

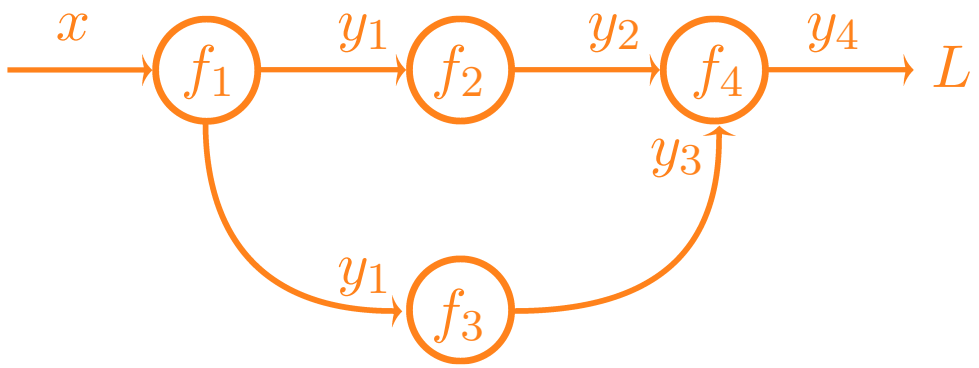
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\Big(f_2\big(f_1(x)\big),\ f_3\big(f_1(x)\big)\Big)\Big)$$

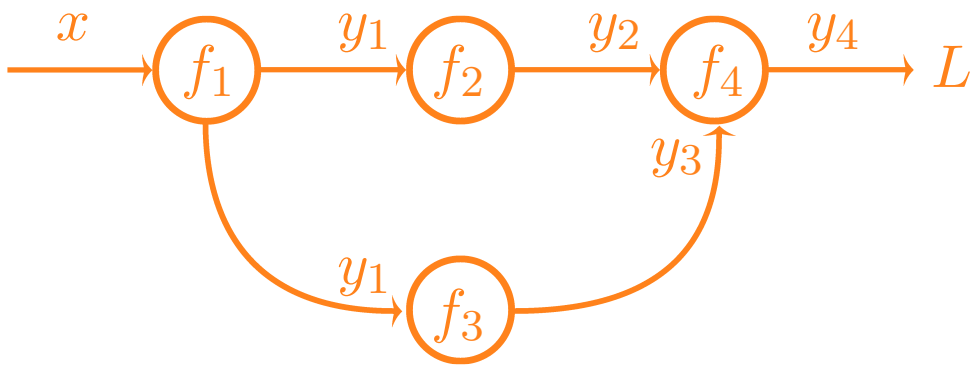
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}\left[\frac{\partial f_4}{\partial y_2}\right.$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\Big(f_2\big(f_1(x)\big),\ f_3\big(f_1(x)\big)\Big)\Big)$$

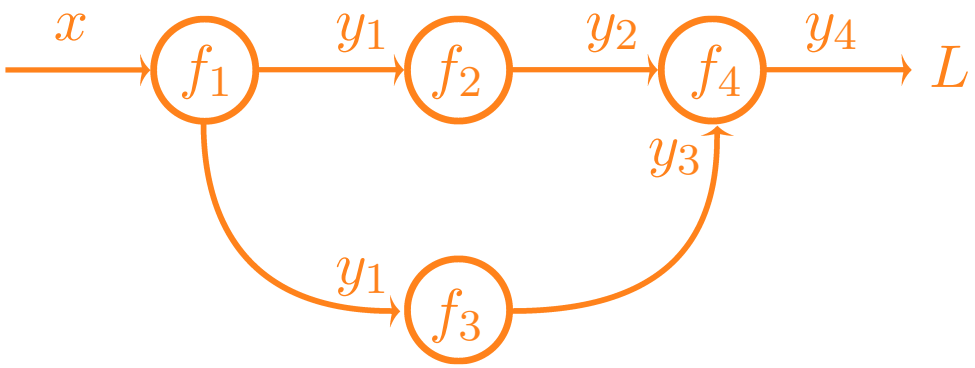
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}\left[\frac{\partial f_4}{\partial y_2} \cdot \frac{\partial f_2}{\partial y_1}\right.$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

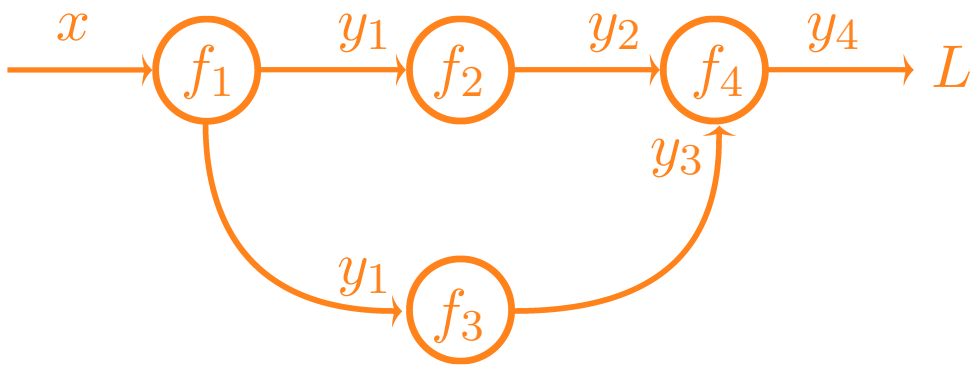
Правило дифференцирования:

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

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}\left[\frac{\partial f_4}{\partial y_2} \cdot \frac{\partial f_2}{\partial y_1} + \frac{\partial f_4}{\partial y_3}\right.$$

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

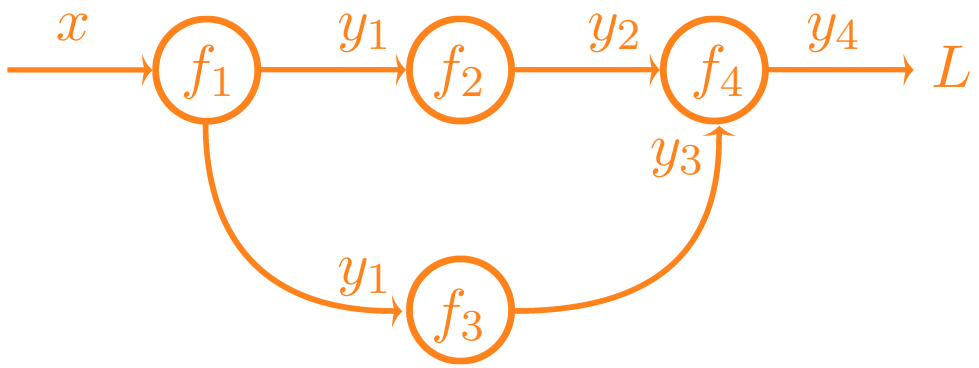
Правило дифференцирования:

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

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

Рассчитываем градиенты: правило цепочки

Мы знаем производную функции потерь. Но как вычислить производные по всем параметрам сети?



$$L\Big(f_4\big(f_2(f_1(x)),\ f_3(f_1(x))\big)\Big)$$

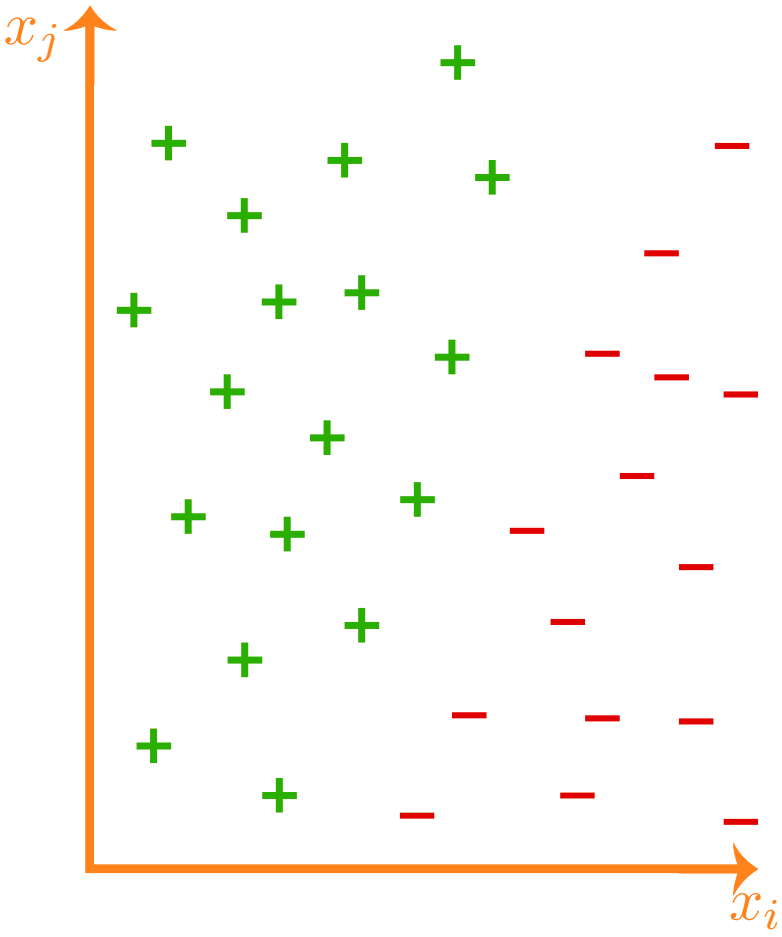
Правило дифференцирования:

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

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

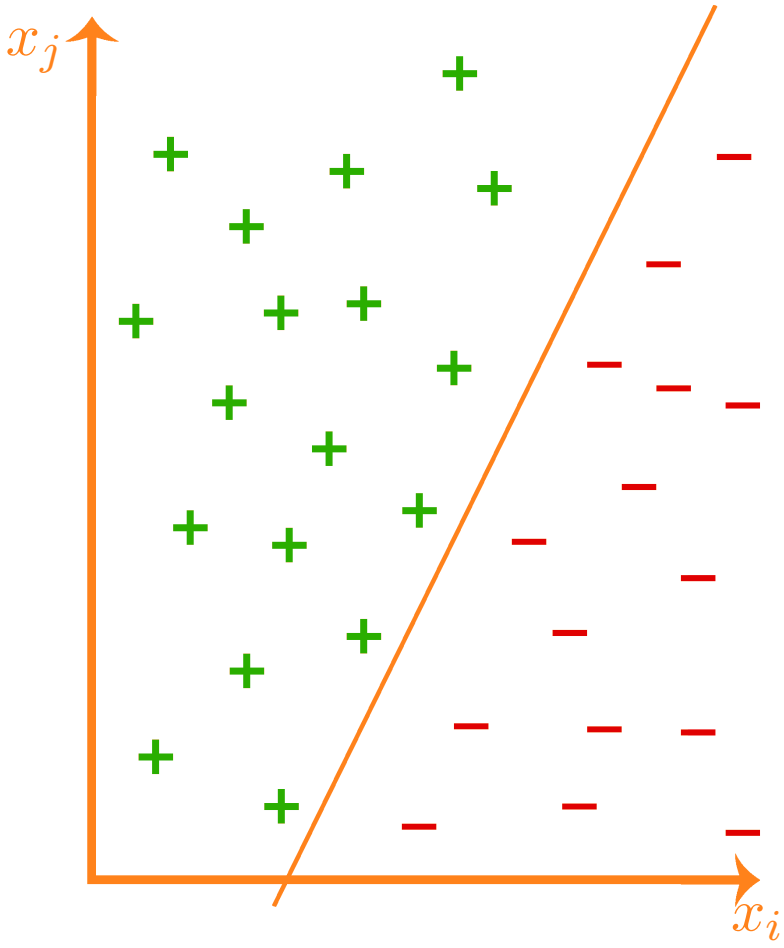
Пример: логистическая регрессия

Линейный алгоритм классификации:



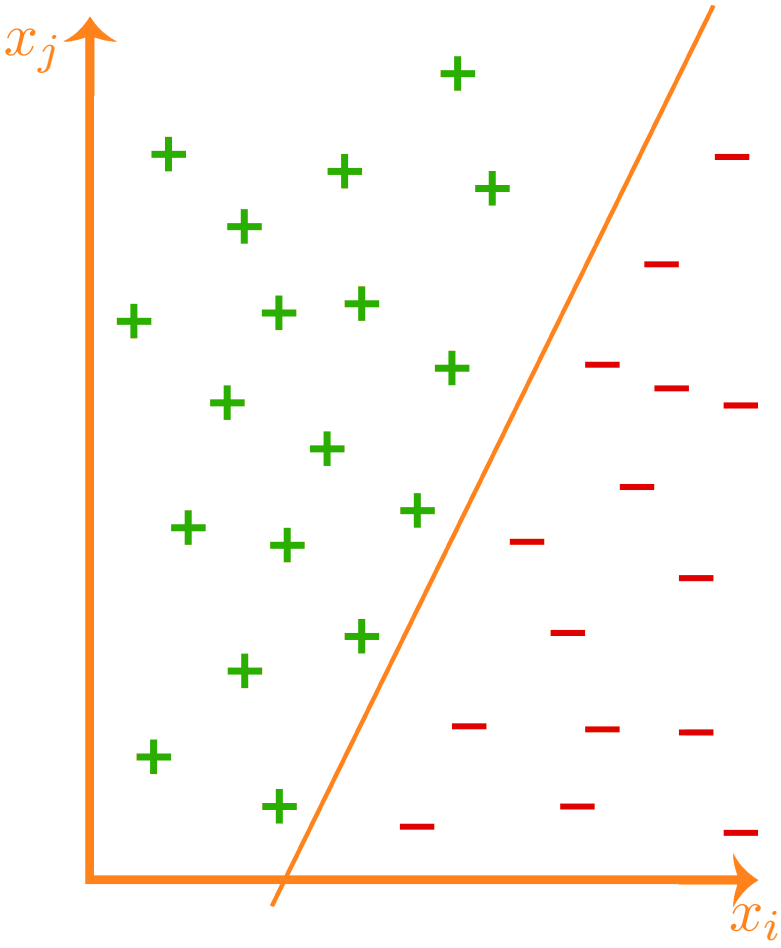
Пример: логистическая регрессия

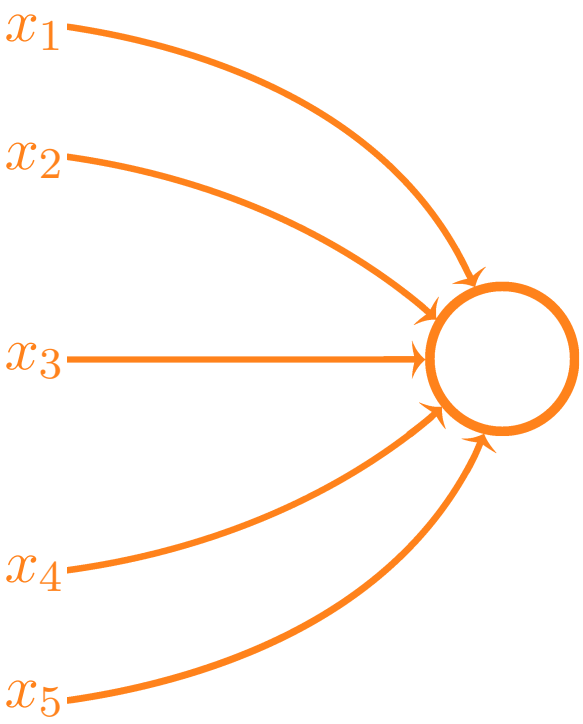
Линейный алгоритм классификации:



Пример: логистическая регрессия

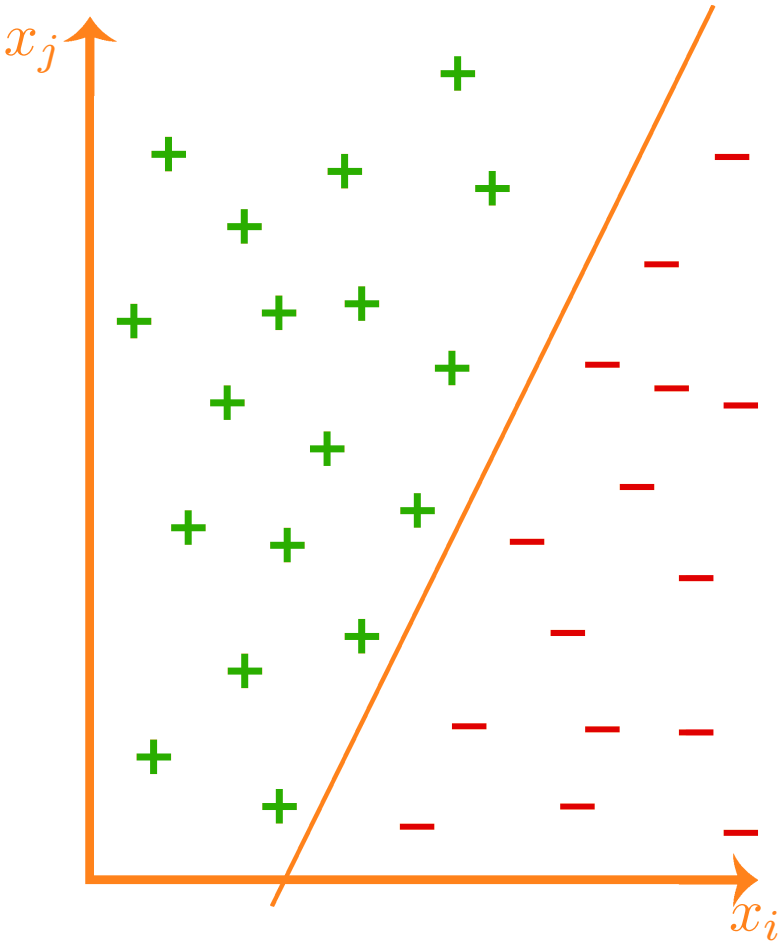
Линейный алгоритм классификации:

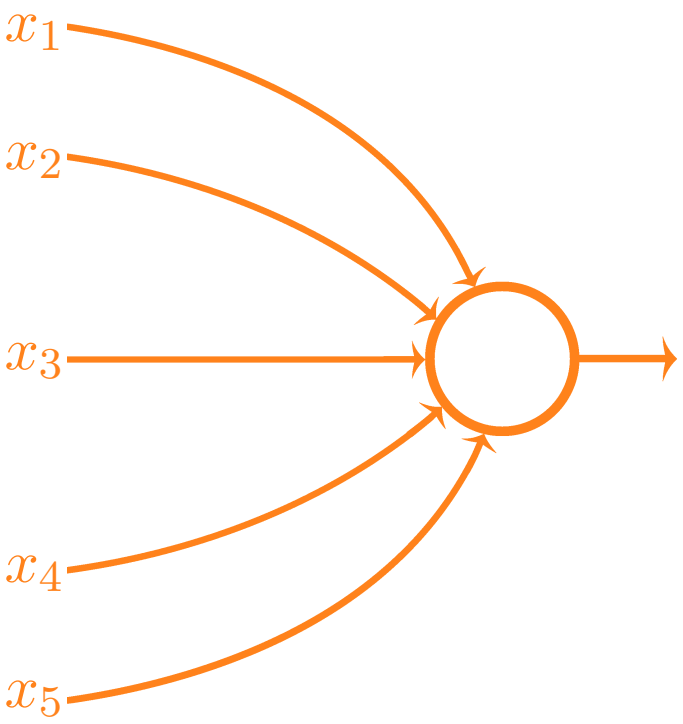




Пример: логистическая регрессия

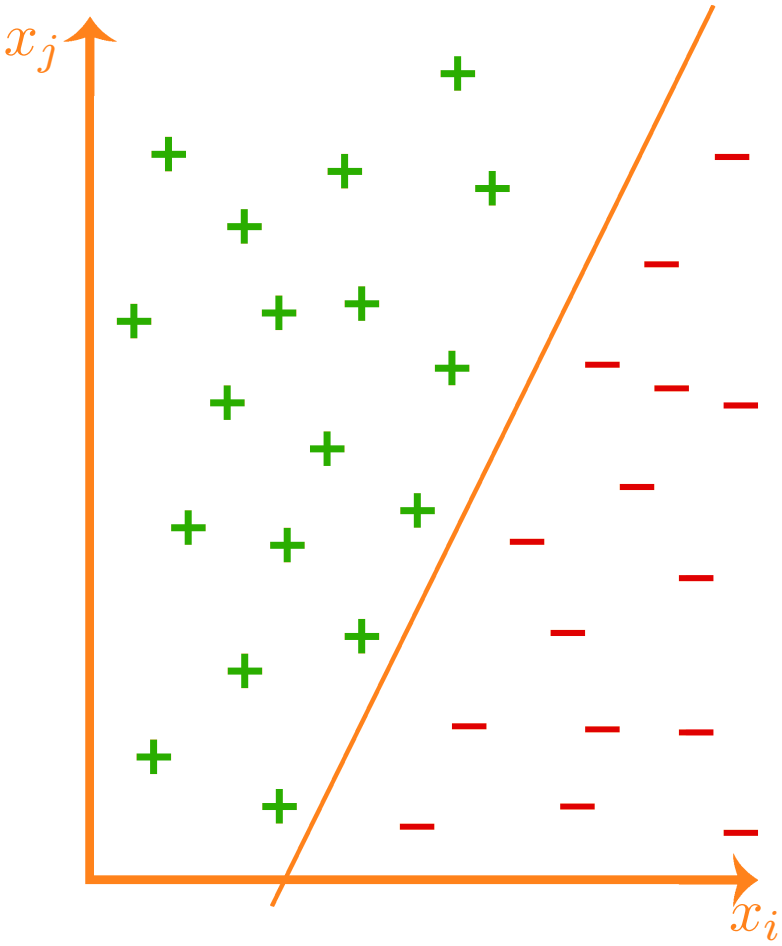
Линейный алгоритм классификации:



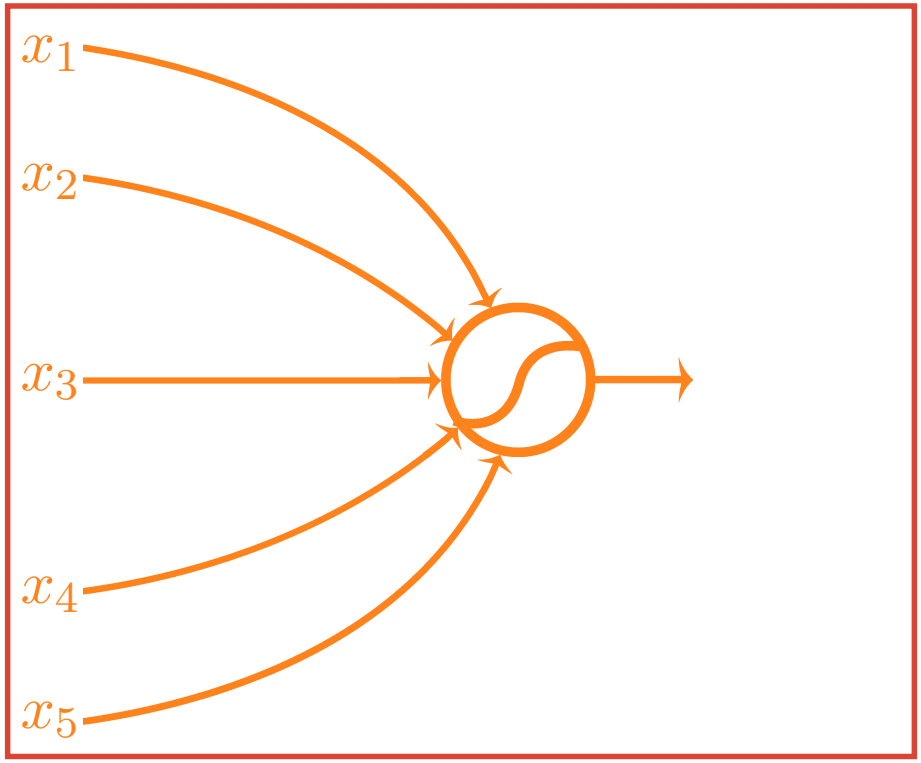


Пример: логистическая регрессия

Линейный алгоритм классификации:

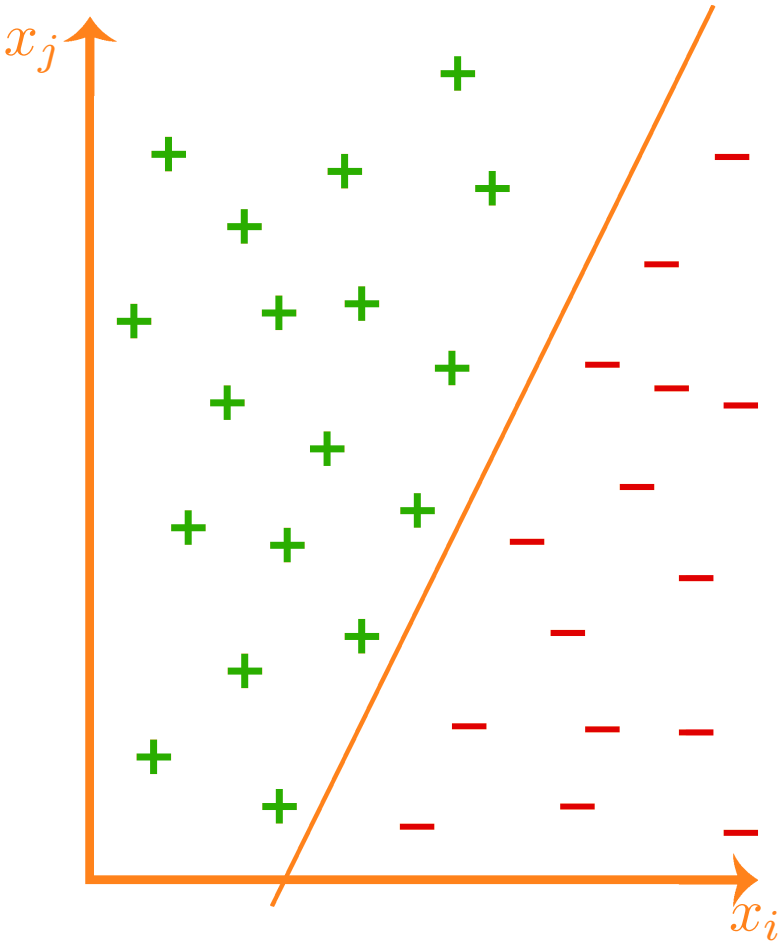


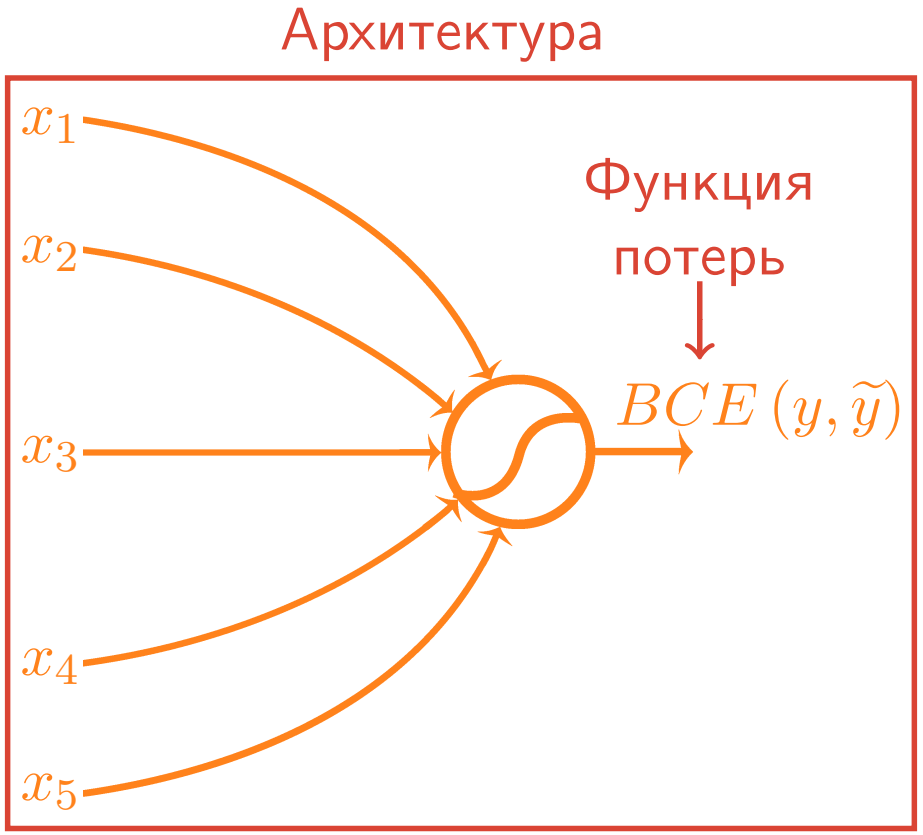
Архитектура



Пример: логистическая регрессия

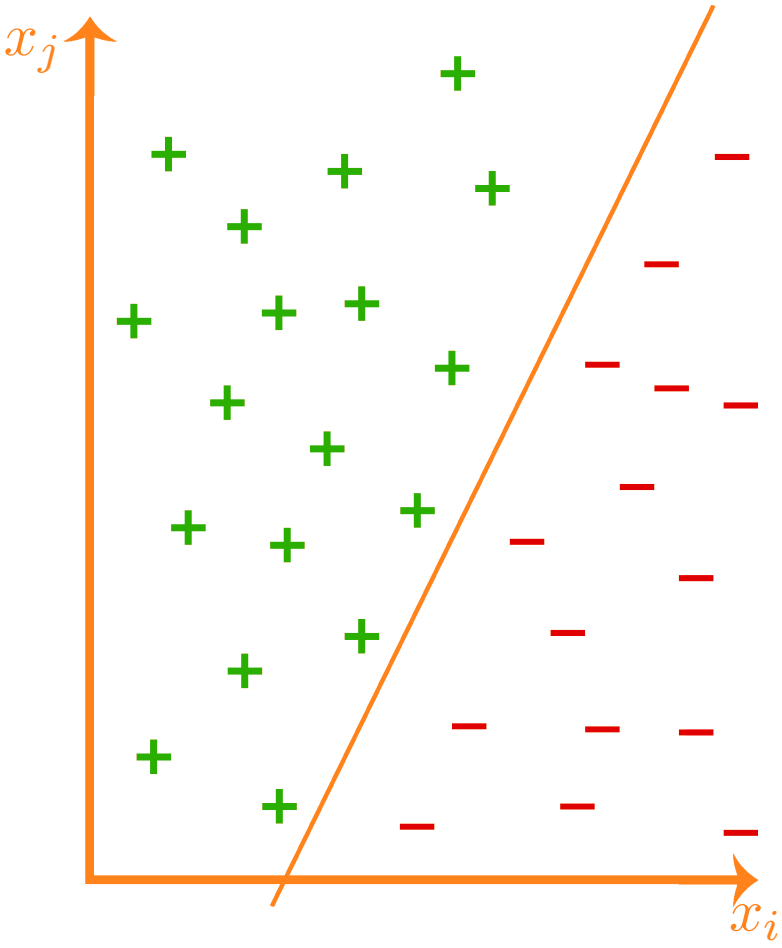
Линейный алгоритм классификации:



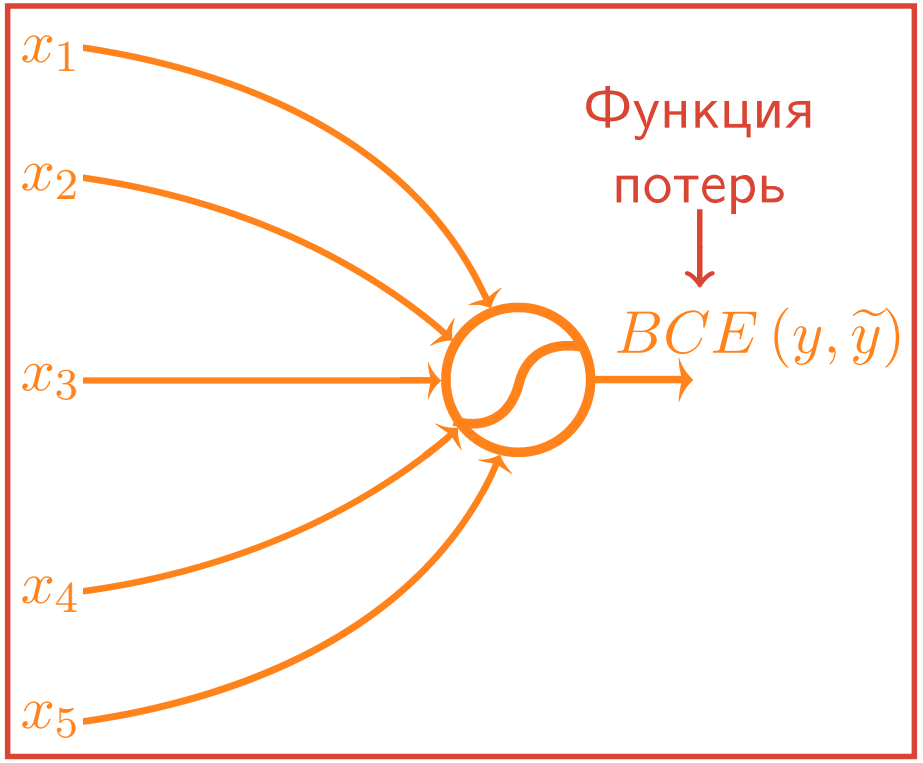


Пример: логистическая регрессия

Линейный алгоритм классификации:



Архитектура



Оптимизация – стохастический градиентный спуск

Итог

- Простой пример задачи для нейронной сети

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка
- Что такое функция потерь

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка
- Что такое функция потерь
- Что такое метрики

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка
- Что такое функция потерь
- Что такое метрики
- Что такое оптимизация сети

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка
- Что такое функция потерь
- Что такое метрики
- Что такое оптимизация сети
- Как считать производную по параметрам сети

Итог

- Простой пример задачи для нейронной сети
- Как построить простую нейронную сеть
- Что такое обучающая выборка
- Что такое функция потерь
- Что такое метрики
- Что такое оптимизация сети
- Как считать производную по параметрам сети
- Логистическая регрессия – тоже нейронная сеть