Линейная регрессия и градиентный спуск

Гончаров Павел kaliostrogoblin3@gmail.com
Нестереня Игорь nesterione@gmail.com

Задача регрессии

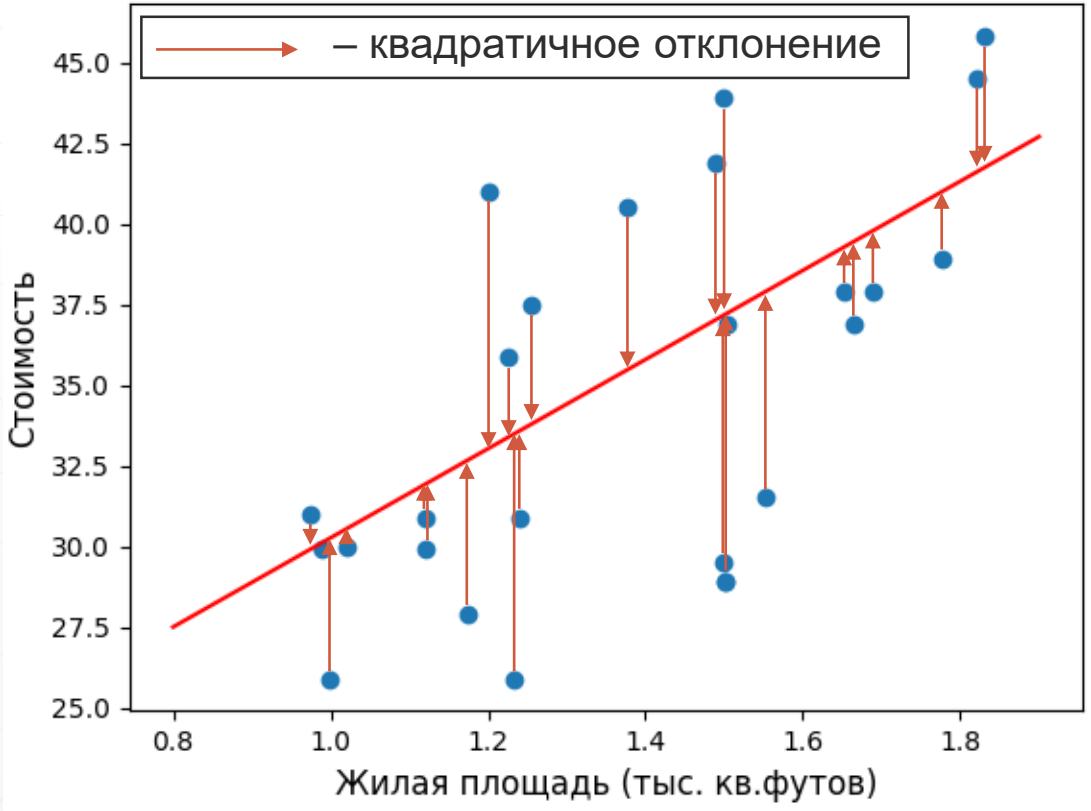
Регрессия (лат. *regressio* «обратное движение, возвращение») в математике и теории вероятности – это математическое выражение, отражающее зависимость целевой переменной *y* от независимых переменных *x* при условии статистической значимости.

История (из вики):
«Этот термин в статистике впервые был использован Френсисом Гальтоном (1886) в связи с исследованием вопросов наследования физических характеристик человека. В качестве одной из характеристик был взят рост человека; при этом было обнаружено, что в целом сыновья высоких отцов, что не удивительно, оказались более высокими, чем сыновья отцов с низким ростом. Более интересным было то, что разброс в росте сыновей был меньшим, чем разброс в росте отцов. Так проявлялась тенденция возвращения роста сыновей к среднему (regression to mediocrity), то есть «регресс». Этот факт был продемонстрирован вычислением среднего роста сыновей отцов, рост которых равен 56 дюймам, вычислением среднего роста сыновей отцов, рост которых равен 58 дюймам, и т. д. После этого результаты были изображены на плоскости, по оси ординат которой откладывались значения среднего роста сыновей, а по оси абсцисс — значения среднего роста отцов. Точки (приближённо) легли на прямую с положительным углом наклона меньше 45°; важно, что регрессия была линейной».

Линейная регрессия

Линейная регрессия – модель зависимости одной целевой переменной от другой или нескольких независимых переменных (регрессоров) с линейной функцией зависимости.

Пример: предсказание цен на жилье в зависимости от размера жилой площади

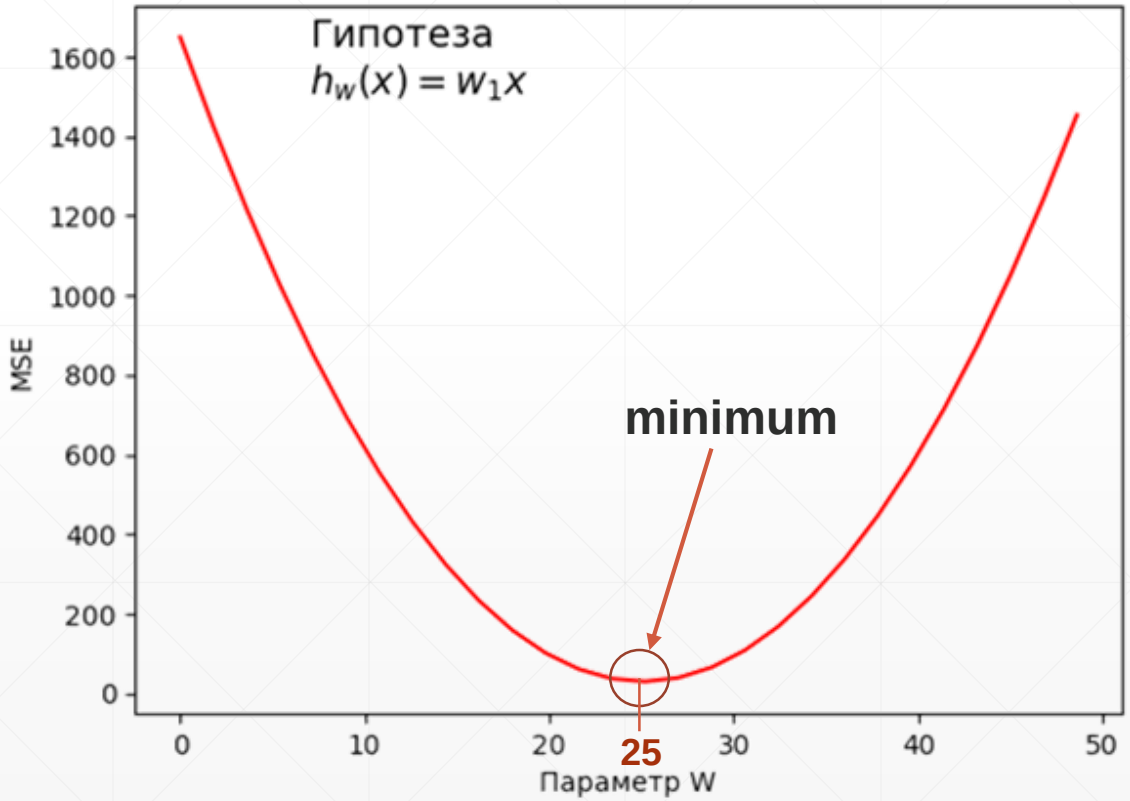


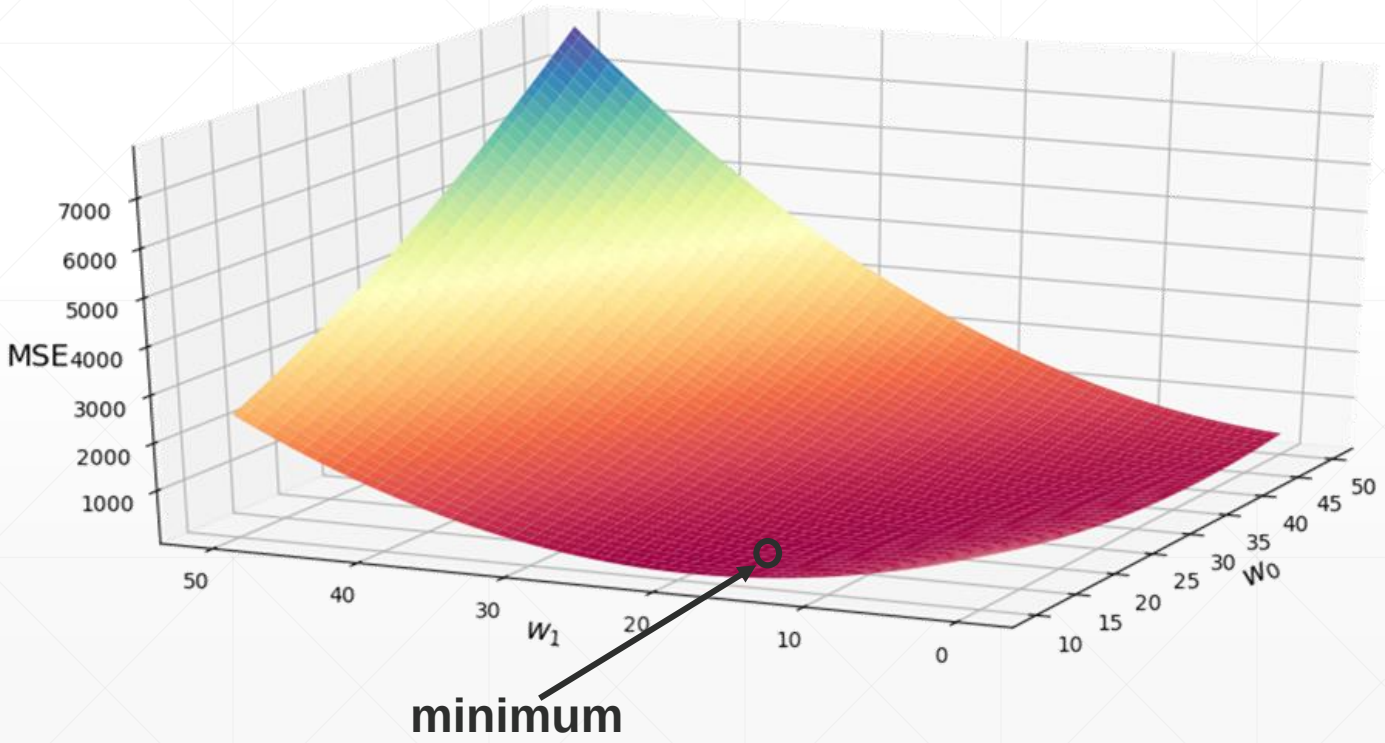
Цель: найти такие параметры w гипотезы $h_w(x)$, при которых среднеквадратичное отклонение $\frac{1}{m}\sum_i \left(h_w(x^{(i)}) - y^{(i)}\right)^2$ будет минимальным, где $(x^{(i)}, y^{(i)})$ – это обучающий пример.

$$\frac{1}{m}\sum_i \left(h_w(x^{(i)}) - y^{(i)}\right)^2 \to min$$

Одномерная и многомерная оптимизация

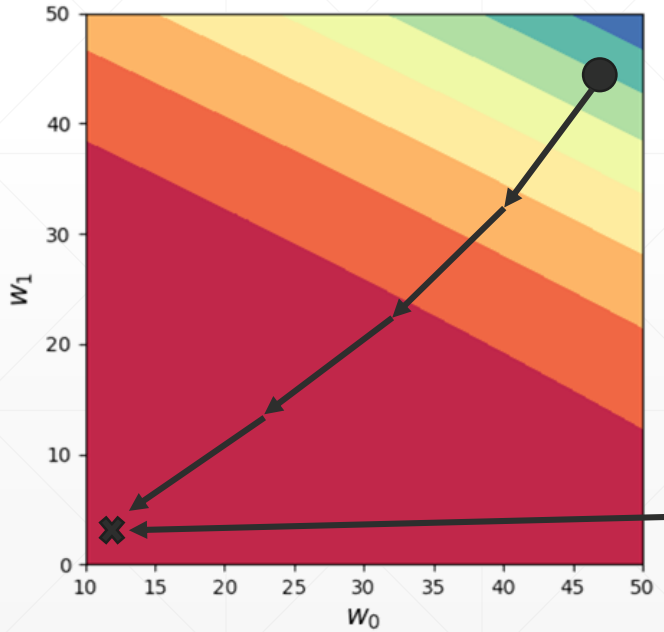
Задача оптимизации сводится к задаче поиска экстремума (максимума или минимума) функции ошибки (*cost function*). Cost для регрессии – это, как правило, среднеквадратичное отклонение (MSE).

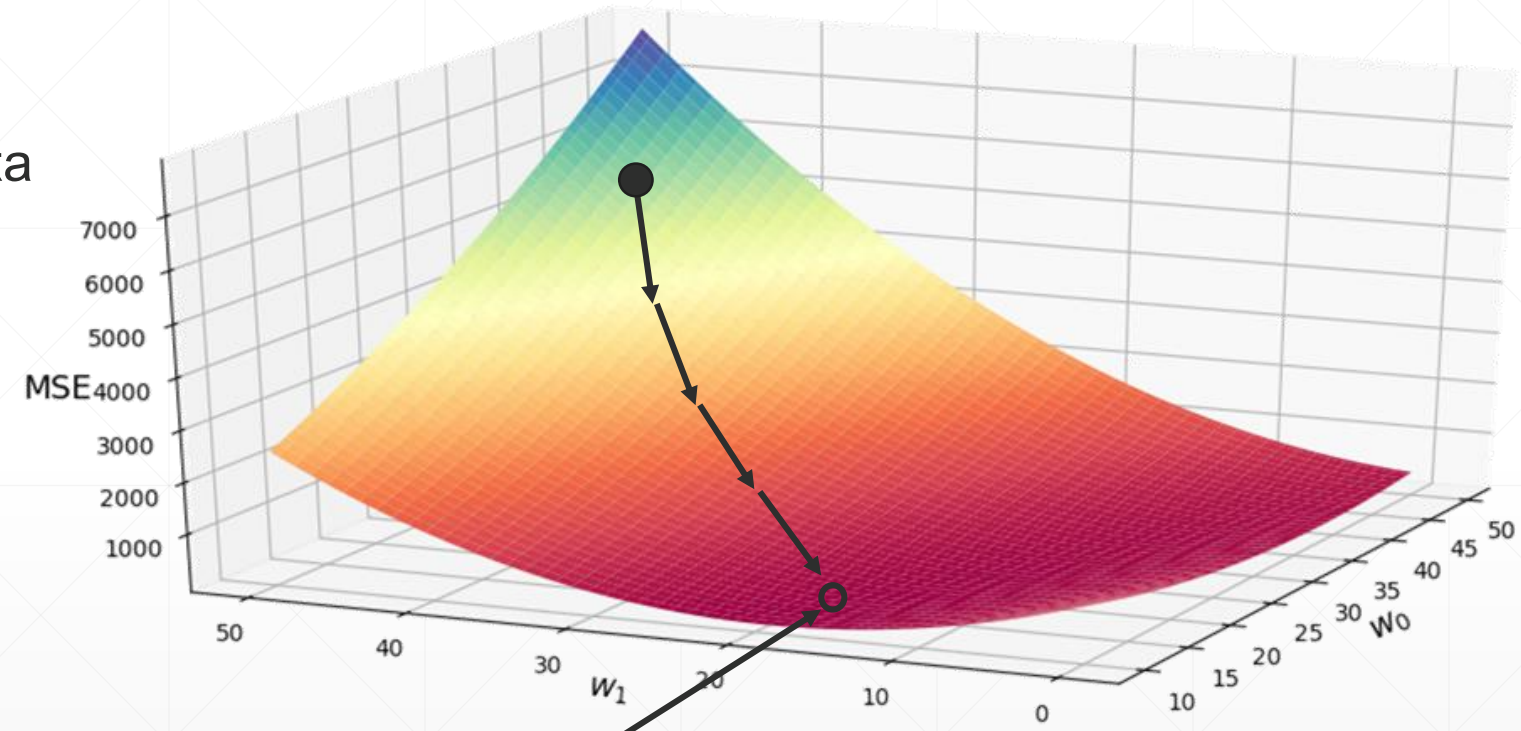




Методы многомерной оптимизации. Градиентный спуск

- Хука – Дживса
- Нелдера-Мида
- метод полного перебора
- метод покоординатного спуска
- **градиентный спуск**





Локальный минимум

Градиент – вектор, указывающий направление роста функции.

Градиентный спуск — метод нахождения *локального* экстремума (минимума или максимума) функции с помощью движения вдоль градиента (для минимума – антиградиента).

Алгоритм градиентного спуска

Цель: изменять значение параметров w модели $h_w(x)$ «шагая» в направлении к локальному минимуму функции ошибки ($J(w)$). В случае регрессии $J(w)$ = MSE.

Алгоритм:

- задать начальное значение параметров w, например $w_0 = w_1 = \cdots = w_n = 0$
- определить точность ε, например $\varepsilon = 0{,}001$
- задать ***скорость обучения*** α
- повторять до сходимости $J(w)_i - J(w)_{i-1} < \varepsilon$:
  - для всех параметров w найти смещение:

$$temp0 = w_0 - \alpha \frac{\partial}{\partial w_0} J(w_0, w_1, \dots, w_n);$$

$$\dots$$

$$tempN = w_n - \alpha \frac{\partial}{\partial w_n} J(w_0, w_1, \dots, w_n).$$

  - обновить все веса w_i, где $i = 0{,}1, \dots, n$:

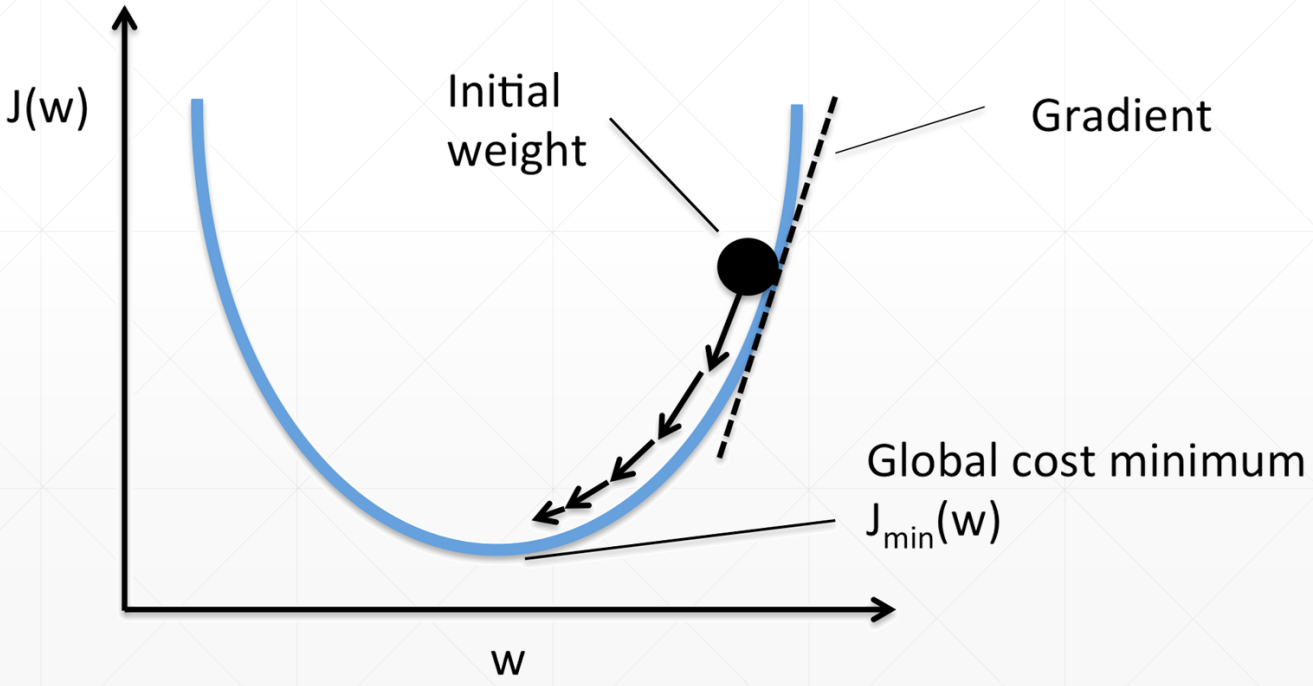
$$w_n = tempN$$

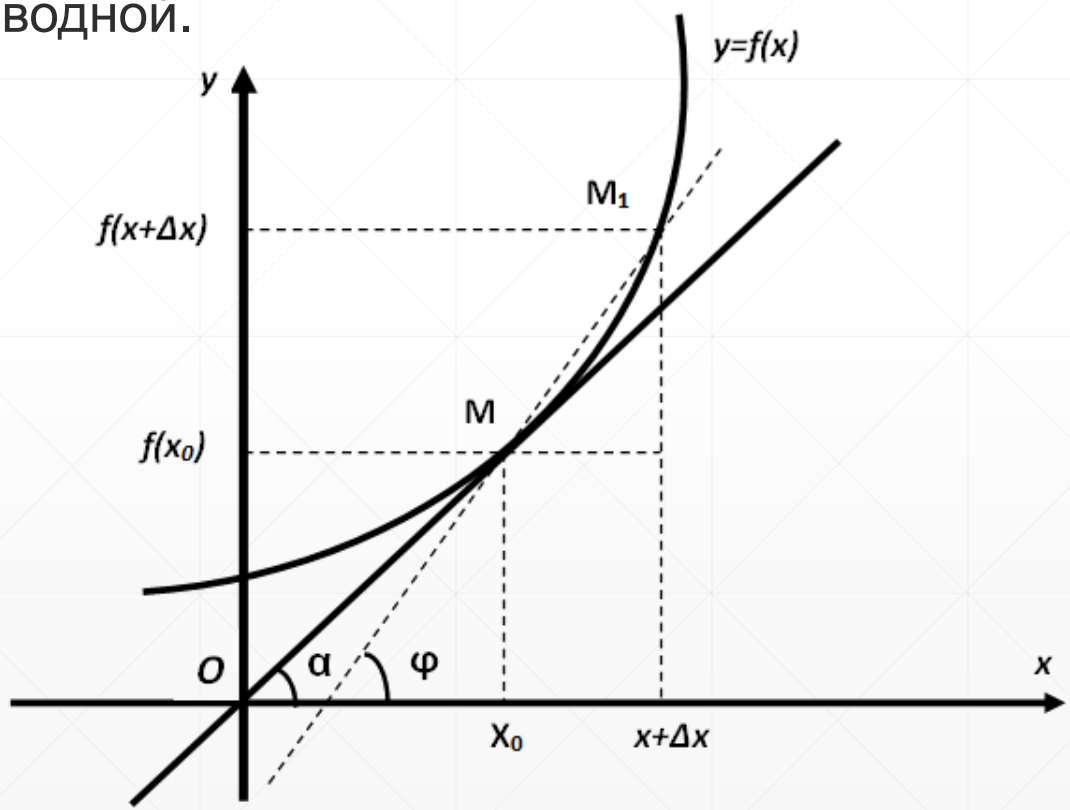
До тех пор, пока ошибка уменьшается больше, чем на epsilon

Частная производная cost function по параметру w_n

Частная производная

Частная производная – это отношение приращение функции по выбранной переменной к приращению этой переменной, скорость изменения функции в данной точке. В геометрическом смысле производная представляет собой угловой коэффициент касательной к функции в точке производной.





Обновление параметров модели

ОСТОРОЖНО МНОГО ФОРМУЛ!

Для того, чтобы обновить параметр, нужно от текущего значения параметра w_i отнять производную функции ошибки $J(w)$ по параметру w_i – $\frac{\partial J(w)}{\partial w_i}$, умноженную на скорость обучения α:

$$w_i = w_i - \alpha \frac{\partial}{\partial w_i} J(w)$$

Например:

$$h_w(x) = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n = w^T x, \text{где } x_0 = 1 \quad (1)$$

1) **Модель линейной регрессии**
2) **Функция ошибки, *cost function***
3) **Частная производная функции ошибки по параметру w_i**

$$J(w) = \frac{1}{2m} \sum_{j=1}^{m} \left(h_w(x^{(j)}) - y^{(j)}\right)^2 \quad (2)$$

$$\frac{\partial}{\partial w_i} J(w) = \frac{1}{m} \sum_{j=1}^{m} \left(h_w\left(x^{(j)}\right) - y^{(j)}\right) x_i^{(j)} \quad (3)$$

Алгоритм градиентного спуска

Алгоритм:

- задать начальное значение параметров w, например $w_0 = w_1 = \cdots = w_n = 0$
- определить точность ε, например $\varepsilon = 0{,}001$
- задать ***скорость обучения*** $\alpha \in [0,1]$
- повторять до сходимости $J(w)_i - J(w)_{i-1} < \varepsilon$:
  - для всех параметров w найти смещение:

$$temp0 = w_0 - \alpha \frac{1}{m} \sum_{j=1}^{m} \left(h_w\left(x^{(j)}\right) - y^{(j)}\right)^2 x_0^{(j)};$$

$$temp1 = w_1 - \alpha \frac{1}{m} \sum_{j=1}^{m} \left(h_w\left(x^{(j)}\right) - y^{(j)}\right)^2 x_1^{(j)};$$

$$\cdots$$

$$tempN = w_n - \alpha \frac{1}{m} \sum_{j=1}^{m} \left(h_w\left(x^{(j)}\right) - y^{(j)}\right)^2 x_n^{(j)}.$$

  - обновить все веса w_i, где $i = 0,1,\ldots,n$:

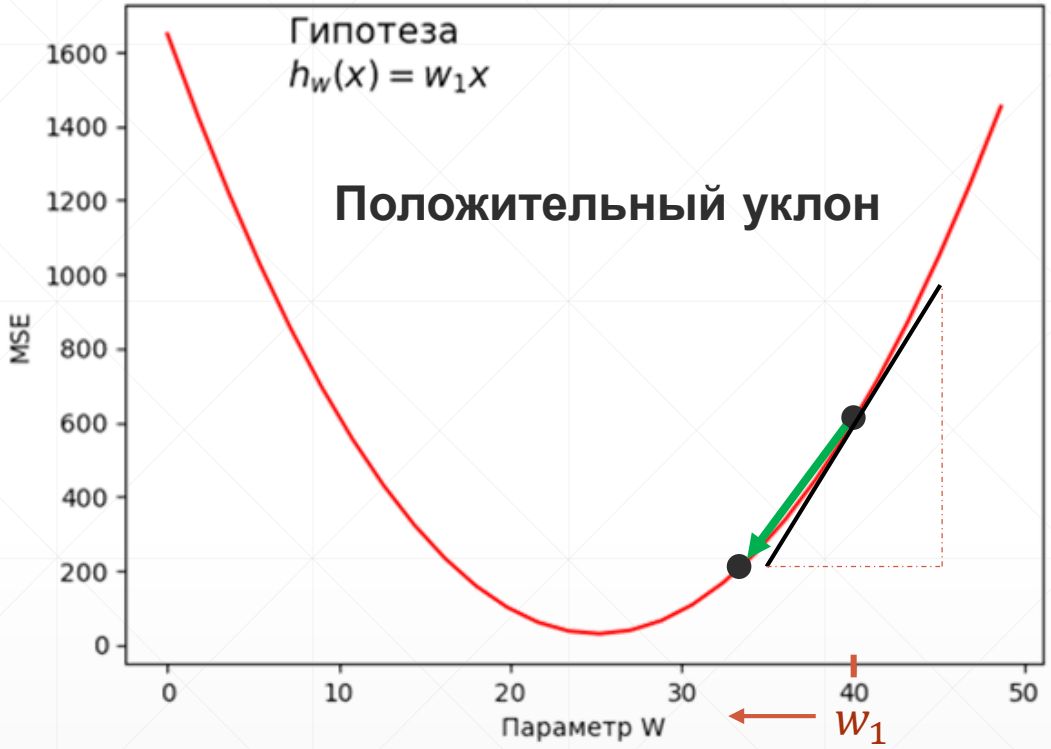
$$w_0 = temp0;$$

$$\ldots$$

$$w_n = temp1;$$

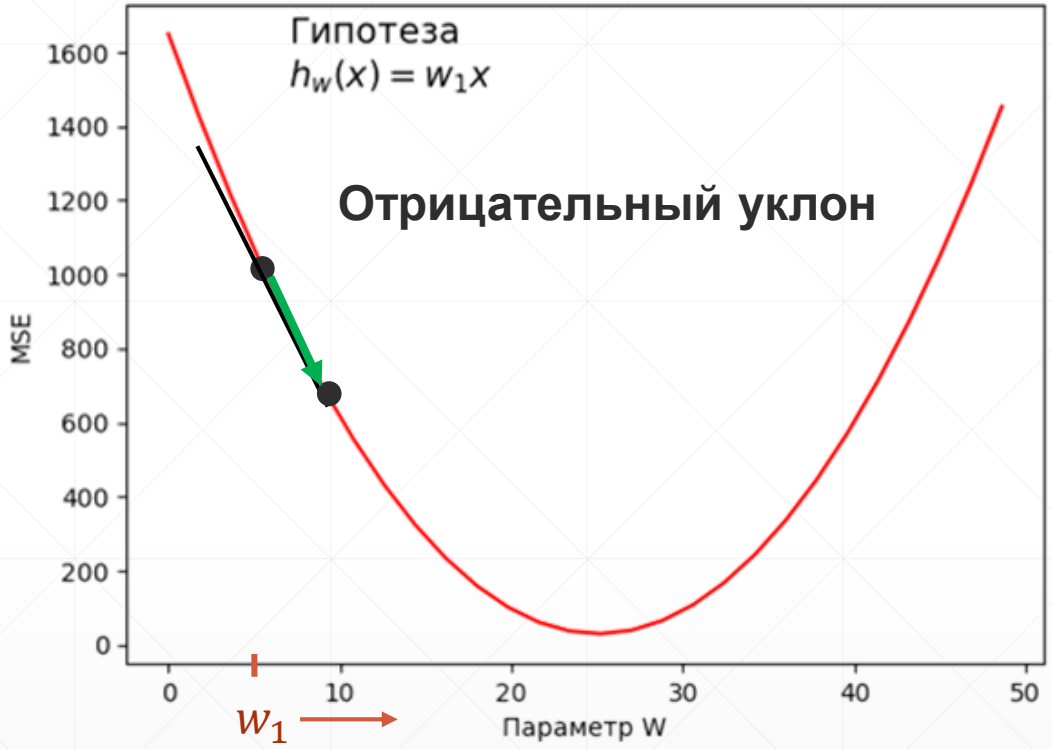
Веса нужно обновлять все **одновременно**, для этого сначала подсчитывается смещение, а потом уже обновляются все веса.

Как это работает?



$$w_i = w_i - \alpha\left(\frac{\partial}{\partial w_i} J(w) \geq 0\right)$$

Вес уменьшается

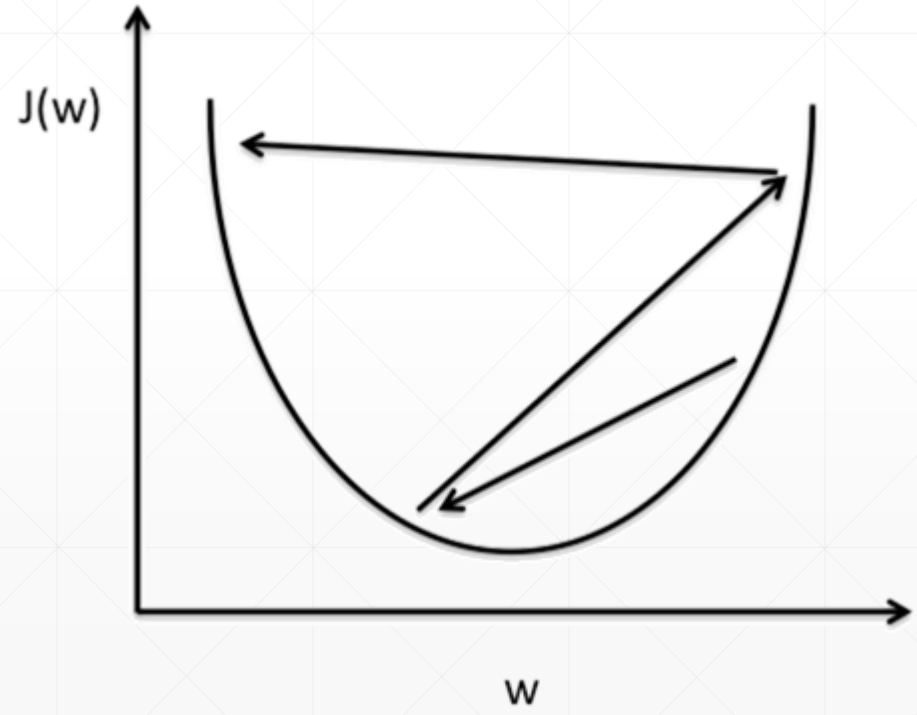


$$w_i = w_i - \alpha\left(\frac{\partial}{\partial w_i} J(w) \leq 0\right)$$

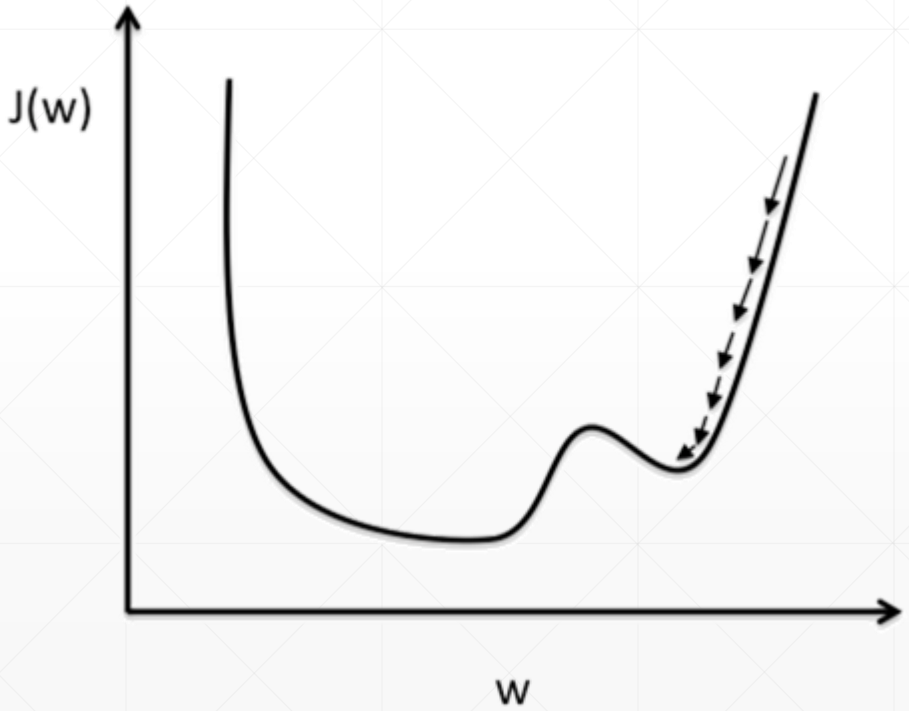
Вес увеличивается

Скорость обучения

Скорость обучения – это параметр обучения $\alpha \in [0,1]$, контролирующий величину коррекции весов-коэффициентов модели. Если $\alpha = 0$, то параметры w будут оставаться неизменными.



Большая α: промах (overshooting)

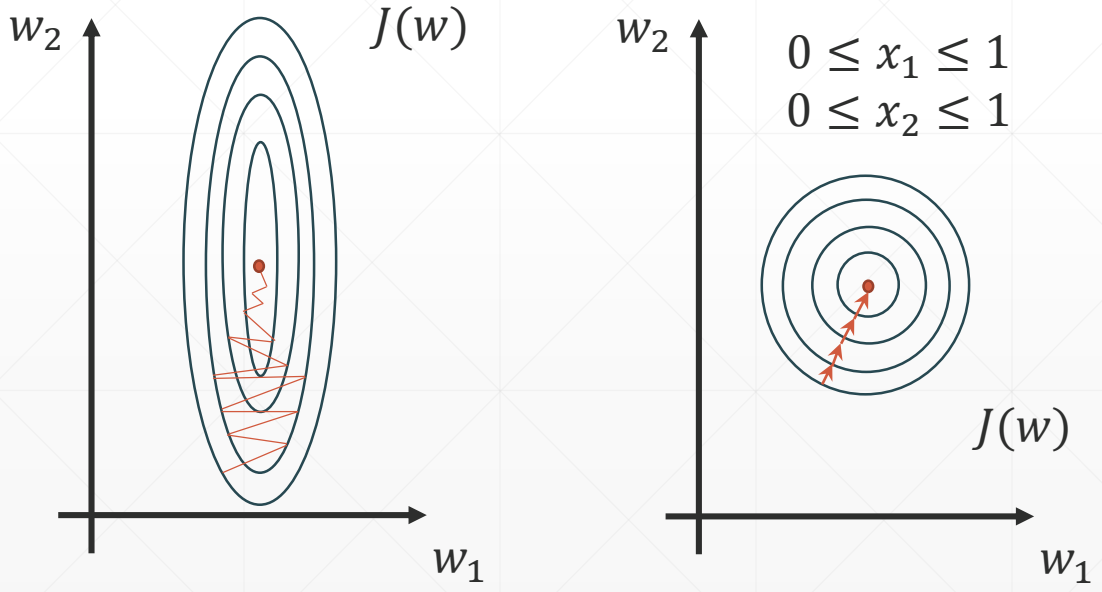


Слишком маленькая α: очень много итераций, попадание в локальный минимум

Нормализация и стандартизация данных

Как правило, каждый признак имеет свой диапазон значений. Например, если мы говорим об оценке стоимости жилья по каким-то критериям, то параметр «число комнат» будет иметь значение от 1-10, а параметр «размер жилой площади» может измеряться сотнями квадратных метров. Важно привести все параметры к виду:

$$-1 \leq x \leq 1$$



1) MinMax масштабирование:

$$\frac{x - x_{min}}{x_{max} - x_{min}} \Rightarrow 0 \leq x \leq 1$$

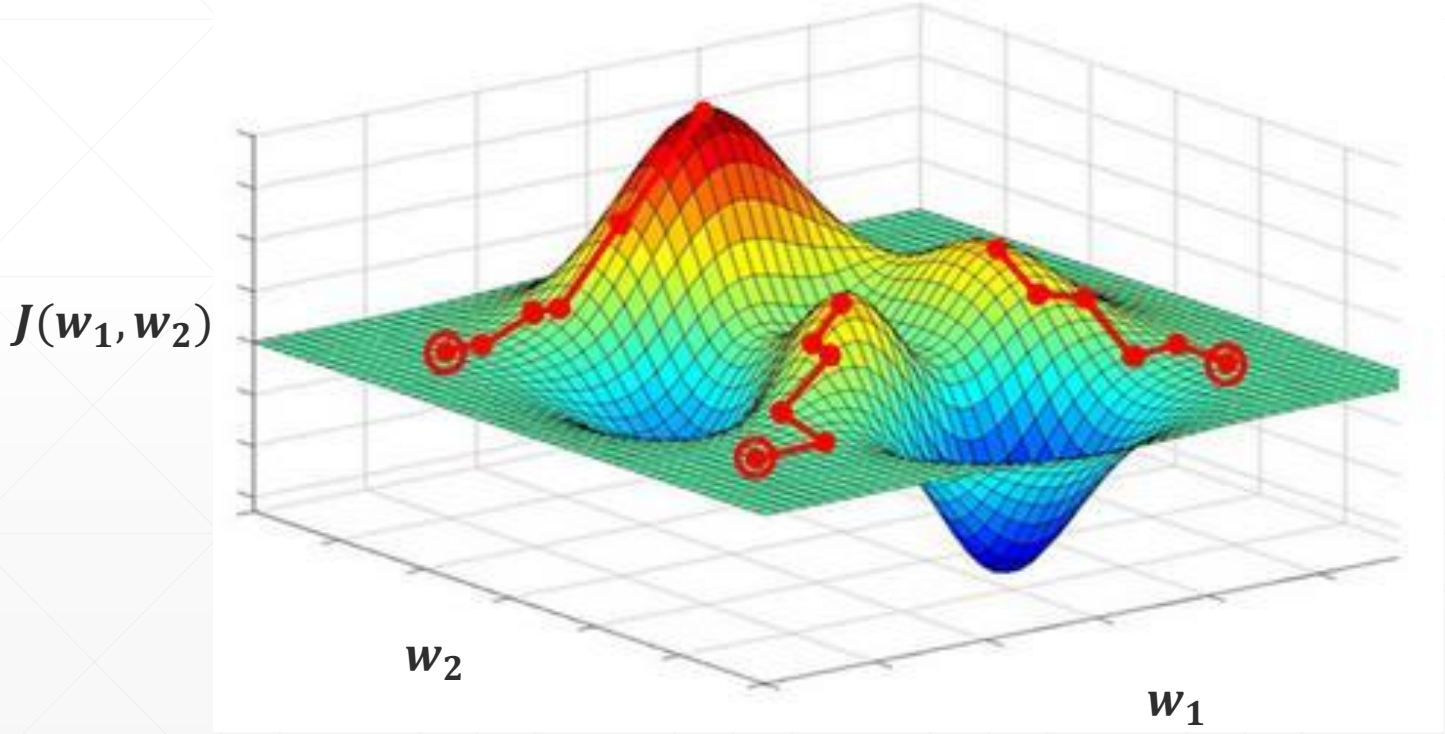
2) Z-score стандартизация:

$$\frac{x - \mu}{\sigma} \Rightarrow -1 \leq x \leq 1$$

$$\mu - \text{среднее}, \sigma = x_{max} - x_{min}$$

Проблема градиентного спуска. Локальный минимум

Одной из главных проблем градиентного спуска является то, что при разном начальном значении можно попасть в **разные локальные минимумы**. Бороться с этим можно: начинать спуск из разных начальных точек, а потом сравнивать результаты величины ошибки и выбирать наилучший вариант.



Использование всей выборки данных для каждого шага спуска называется **batch training**.

Сделаем свою линейную регрессию с параметрами и градиентным спуском!