1. Метод градиентного спуска

Градиентные методы довольно часто используются для решения задач многомерной безусловной оптимизации. Пусть стоит задача найти минимум функции:

$$f(\theta) \to \min_{\theta}, \tag{1}$$

где θ – **вектор** переменных. Алгоритм метода градиентного спуска состоит в последовательном движении в направлении наискорейшего спуска, то есть в направлении антиградиента $-\nabla_\theta f(\theta)$.

<u>*Метод градиентного спуска*</u>

Вход: функция $f(\theta)$, начальное приближение θ^0, шаг градиентного спуска η, число итераций n, ϵ.

1. Определить $\nabla_\theta f$.
2. На каждом шаге $t = 1, 2, \ \ldots, n$:

   $\theta^{(t+1)} = \theta^{(t)} - \eta \, \nabla_\theta f\left(\theta^{(t)}\right)$.

Возможные критерии останова:

a. $||\,\theta^{(t+1)} - \theta^{(t)}\,|| \ \le \epsilon$,

b. $||\, f\left(\theta^{(t+1)}\right) - f\left(\theta^{(t)}\right) ||\ \le \epsilon$.

Если взять чересчур большим шаг градиентного спуска, то есть риск «перескочить» минимум. Чтобы этого избежать на каждой итерации алгоритма шаг градиентного спуска можно менять:

1. Обратно пропорционально номеру итерации:

   $\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)} \, \nabla_\theta f\left(\theta^{(t)}\right)$

   $\eta^{(t+1)} = \frac{\eta^{(0)}}{t}$.

2. Недостатком первого способа является то, что с увеличением номера итерации дальнейшие шаги будут чересчур маленькими, и есть риск не дойти до минимума. Чтобы этого избежать, можно использовать экспоненциально затухающий шаг:

   $\eta^{(t+1)} = \eta^{(0)} \, e^{\frac{1-t}{t}}$.

Выбирать шаг градиентного спуска можно таким образом, чтобы значение функции $f\left(\theta^{(t+1)} - \eta^{(t+1)} \, \nabla_\theta f\left(\theta^{(t+1)}\right)\right)$ было наименьшим. Для нахождения оптимального шага $\eta^{(t+1)}$ используются любые методы одномерной оптимизации. Полученный алгоритм имеет название **метод наискорейшего спуска**.

Задание 1

1. Сгенерировать 1000 точек по следующему правилу:

   $y = \cos(1.5\,\pi x) + \mathcal{N}(0, \ 1)$.

2. Методом градиентного спуска определить параметры θ в модели полиномиальной регрессии.
3. Отобразить на графике изменения значения функционала качества с номером итерации.
4. Реализовать метод кросс-валидации для оценки качества полученной модели.
5. Определить степень полинома, при которой достигается наилучшее качество модели на кросс-валидации.

6. Полученную функцию отобразить на графике.
7. Дополнительно. Реализовать метод наискорейшего спуска. Для этого необходимо реализовать один из методов одномерной оптимизации, например, метод золотого сечения.