

Оптимизационные задачи в машинном обучении. Проект №3: Методы многомерной оптимизации (Градиентный спуск, Ньютона-сопряженного градиента)

Документация

Проект Gradient_methods содержит следующий функционал:

Внутренние функции, не предназначены для прямого вызова пользователем

`_gradient(expr, point)` - Считает значение градиента функции в точке.

`_visualize(func, history)` - Строит 3d график функции и scatter plot движения градиента

`ask_input(ask_alpha=False, ask_alpha0=False, ask_delta=False, ask_gamma=False, ask_history=False, ask_visualizing=False)` - Запрашивает у пользователя входные данные

`constant_gradient_descent(func, x0, alpha=0.1, max_iter=500, epsilon=1e-5, show_history=False, visualize=False)` - Находит точку минимума функции методом градиентного спуска с константным шагом

`step_splitting_gd(func, x0, alpha0=0.1, delta=0.1, gamma=0.1, max_iter=500, epsilon=1e-5, show_history=False, visualize=False)` - Находит точку минимума функции методом градиентного спуска с дроблением шага

`fastest_gd(func, x0, max_iter=500, epsilon=1e-5, show_history=False, visualize=False)` - Находит точку минимума функции методом наискорейшего градиентного спуска

`conjugate_gradient_method(func, x0, max_iter=500, epsilon=1e-5, show_history=False, visualize=False)` - Находит точку минимума функции методом Ньютона-сопряжённого градиента.

Функции, предназначенные для прямого вызова пользователем

`compare()` - Запрашивает у пользователя входные данные через и выводит на экран таблицу с результатами работы всех алгоритмов

`gradient_descent()` - Предлагает пользователю выбрать алгоритм решения, запрашивает у него данные и находит точку минимума, выбранным алгоритмом

Более подробно о каждой из функций можно узнать с помощью `help(*Нужная функция*)`