**Тема 3. Методы многомерной оптимизации (Градиентный спуск, Ньютна-сопряженного градиента)**

**Задание 1.** Написать (добавить в собственный класс/библиотеку) следующие функции:

1. Поиск экстремума функции многих переменной методом **градиентного спуска с постоянным шагом**;

*\* Возможна самостоятельная декомпозиция задачи на нескольких функций, решающих конкретные подзадачи. В таком случае, для удобства вызова пользователем, необходимо предусмотреть оберточную функцию.*

*Формат входных данных:*

Обязательные параметры:

а) Минимизируемая функция в аналитическом виде;

б) Функция градиента в аналитическом виде;

*\* оптимально определять автоматически с помощью дополнительного метода.*

в) Константа шага (λ);

Необязательные параметры:

г) Максимальное число итераций (по умолчанию: 500 );

д) Точность оптимизации по аргументу для критерия Останова (по умолчанию: 10^-5);

е) Флаг «вывод промежуточных результатов» (по умолчанию: False) – при установке TRUE выводит полученные значения результатов на каждой итерации);

ж) Флаг «запись промежуточных результатов в датасет» (по умолчанию: False) – при установке TRUE записывает номер итерации и полученные значения результатов на каждой итерации в pandas dataset).

*\* возможно добавление других обязательны или необязательных параметров*

*Формат выходных данных:*

а) Найденное значение координаты точки экстремума;

б) Значение функции в точке экстремума;

г) Отчет о работе алгоритма (например флаг: 0 – найдено значение с заданной точностью; 1 – достигнуто максимальное количество итераций; 2 – выполнено с ошибкой).

1. Поиск экстремума функции многих переменной методом **градиентного спуска с дроблением шага**;

*\* Возможна самостоятельная декомпозиция задачи на нескольких функций, решающих конкретные подзадачи. В таком случае, для удобства вызова пользователем, необходимо предусмотреть оберточную функцию.*

*Формат входных данных:*

Обязательные параметры:

а) Минимизируемая функция в аналитическом виде;

б) Функция градиента в аналитическом виде;

*\* оптимально определять автоматически с помощью дополнительного метода.*

в) Начальный шаг (λ0);

г) Значение параметра оценки (ε);

д) Значение параметра дробления (δ);

Необязательные параметры:

е) Максимальное число итераций (по умолчанию: 500 );

ж) Точность оптимизации по аргументу для критерия Останова (по умолчанию: 10^-5);

з) Флаг «вывод промежуточных результатов» (по умолчанию: False) – при установке TRUE выводит полученные значения результатов на каждой итерации);

и) Флаг «запись промежуточных результатов в датасет» (по умолчанию: False) – при установке TRUE записывает номер итерации и полученные значения результатов на каждой итерации в pandas dataset).

*\* возможно добавление других обязательны или необязательных параметров*

*Формат выходных данных:*

а) Найденное значение координаты точки экстремума;

б) Значение функции в точке экстремума;

г) Отчет о работе алгоритма (например флаг: 0 – найдено значение с заданной точностью; 1 – достигнуто максимальное количество итераций; 2 – выполнено с ошибкой).

1. Поиск экстремума функции многих переменной методом **наискорейшего** **градиентного спуска**;

*\* Возможна самостоятельная декомпозиция задачи на нескольких функций, решающих конкретные подзадачи. В таком случае, для удобства вызова пользователем, необходимо предусмотреть оберточную функцию.*

*Формат входных данных:*

Обязательные параметры:

а) Минимизируемая функция в аналитическом виде;

б) Функция градиента в аналитическом виде;

*\* оптимально определять автоматически с помощью дополнительного метода.*

Необязательные параметры:

в) Максимальное число итераций (по умолчанию: 500 );

г) Точность оптимизации по аргументу для критерия Останова (по умолчанию: 10^-5);

д) Флаг «вывод промежуточных результатов» (по умолчанию: False) – при установке TRUE выводит полученные значения результатов на каждой итерации);

е) Флаг «запись промежуточных результатов в датасет» (по умолчанию: False) – при установке TRUE записывает номер итерации и полученные значения результатов на каждой итерации в pandas dataset).

*\* возможно добавление других обязательны или необязательных параметров*

*Формат выходных данных:*

а) Найденное значение координаты точки экстремума;

б) Значение функции в точке экстремума;

г) Отчет о работе алгоритма (например флаг: 0 – найдено значение с заданной точностью; 1 – достигнуто максимальное количество итераций; 2 – выполнено с ошибкой).

***\* В качестве метода одномерной оптимизации рекомендуется использовать метод Брандта***

1. Поиск экстремума функции многих переменной при помощи **алгоритма Ньютона-сопряженного градиента**;

*\* Возможна самостоятельная декомпозиция задачи на нескольких функций, решающих конкретные подзадачи. В таком случае, для удобства вызова пользователем, необходимо предусмотреть оберточную функцию.*

*Формат входных данных:*

Обязательные параметры:

а) Функция в аналитическом виде;

*\* возможно добавление других обязательны или необязательных параметров*

*Формат выходных данных:*

а) Найденное значение координаты точки экстремума;

б) Значение функции в точке экстремума;

1. **Возможно добавление других функций (опционально )**

**Задание 2.** Для тестирования написанных функций а так же прототипирования различных методов проводится следующее приемо-сдаточное тестирование:

\* *Данные задания оформляются в отдельном пайплайне (в случае подключения собственной библиотеки), или ячейках, располагающихся ниже. Каждый тест пишется в своей ячейке и решает свой класс задач.*

1. Решение задачи на нахождение экстремума одномерной функции.

С клавиатуры вводится функция в аналитическом виде и начальные условия. Результатом работы программы является список точек экстремумов и значения исследуемой функции в данной точке, число итераций и другие параметры, описанные в выходных параметрах функций из п.1. Задание выполняется для каждого из разработанных алгоритмов (1-4).

**Дополнительные задания (опционально)**

1. Визуализировать работу одного из разработанных алгоритмов.

Предлагается построить график исходной функции и на нем отобразить способ построения приближенной аппроксимации для быстро сходящейся функции.

*\* Возможны другие варианты визуализации работы алгоритмов, на усмотрение студентов.*

1. Сравнить производительность 4 разработанных алгоритмов.

Предлагается для одной и той же функции провести поиск минимума всеми 4-я алгоритмами. При этом необходимо замерить время выполнения алгоритма и количество итераций.

Результаты предлагается оформить в виде следующей таблицы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Алгоритм1 | Алгоритм2 | Алгоритм3 | Алгоритм4 |
| Полученное решение |  |  |  |  |
| Время выполнения |  |  |  |  |
| Количество итераций |  |  |  |  |
|  |  |  |  |  |

1. Сравнить производительность приближенных алгоритмов и точных алгоритмов.

Предлагается для одной и той же функции провести поиск минимума оптимальным алгоритмом из п.5 и одним из алгоритмов 1 темы данного курса. При этом необходимо замерить время выполнения алгоритма и количество итераций.

Результаты предлагается оформить в виде следующей таблицы:

|  |  |  |
| --- | --- | --- |
| Параметр | Оптимальный итерационный алгоритм | Оптимальный точный алгоритм |
| Полученное решение |  |  |
| Время выполнения |  |  |
|  |  |  |

1. Необходимо оформить проектную документацию по проекту в формате файла «Описание структуры и разделов документации по проекту.docx»

Список литературы для подготовки:

1. <http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%81%D0%BF%D1%83%D1%81%D0%BA%D0%B0#.D0.9A.D1.80.D0.B8.D1.82.D0.B5.D1.80.D0.B8.D0.B9_.D0.BE.D1.81.D1.82.D0.B0.D0.BD.D0.BE.D0.B2.D0.B0>
2. <http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0._%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%A1%D1%82%D0%B5%D1%84%D1%84%D0%B5%D0%BD%D1%81%D0%B5%D0%BD%D0%B0>
3. <https://stackabuse.com/gradient-descent-in-python-implementation-and-theory/>