

# Отчёт по лабораторной работе №3

## Метод стохастического градиентного спуска (SGD) и его модификации

### Задача:

Цель лабораторной работы — реализовать и исследовать на эффективность метод стохастического градиентного спуска (SGD) для решения задачи линейной регрессии. Исследование проводится с различными размерами батча, функциями изменения шага обучения, а также с использованием различных модификаций SGD.

### Используемые методы

#### Линейная регрессия:

Линейная регрессия используется для моделирования отношений между зависимой переменной (target) и одной или несколькими независимыми переменными (features). Модель имеет вид:

$$y = \text{beta\_0} + \text{beta\_1} * x\_1 + \text{beta\_2} * x\_2 + \dots + \text{beta\_n} * x\_n + \text{eps}$$

#### Стохастический градиентный спуск (SGD):

SGD — это метод оптимизации, используемый для нахождения минимума функции потерь.

Он обновляет веса модели по формуле:

$$\mathbf{w} = \mathbf{w} - \text{eta} * \text{delta\_w} * J(\mathbf{w})$$

где  $\text{eta}$  — шаг обучения,  $J(\mathbf{w})$  — функция потерь.

### Модификации SGD

- **Momentum**: ускоряет SGD, добавляя накопленную предыдущую скорость к текущему градиенту.
- **Nesterov**: модификация Momentum, где градиент вычисляется с учётом обновления.
- **AdaGrad**: адаптирует шаг обучения для каждого параметра, уменьшая его для часто встречающихся признаков.
- **RMSProp**: исправляет недостатки AdaGrad, изменяя масштабирование шага обучения.
- **Adam**: объединяет идеи Momentum и RMSProp.

### Реализация

Реализация алгоритмов SGD и его модификаций была выполнена на языке Python с использованием библиотеки `numpy`.

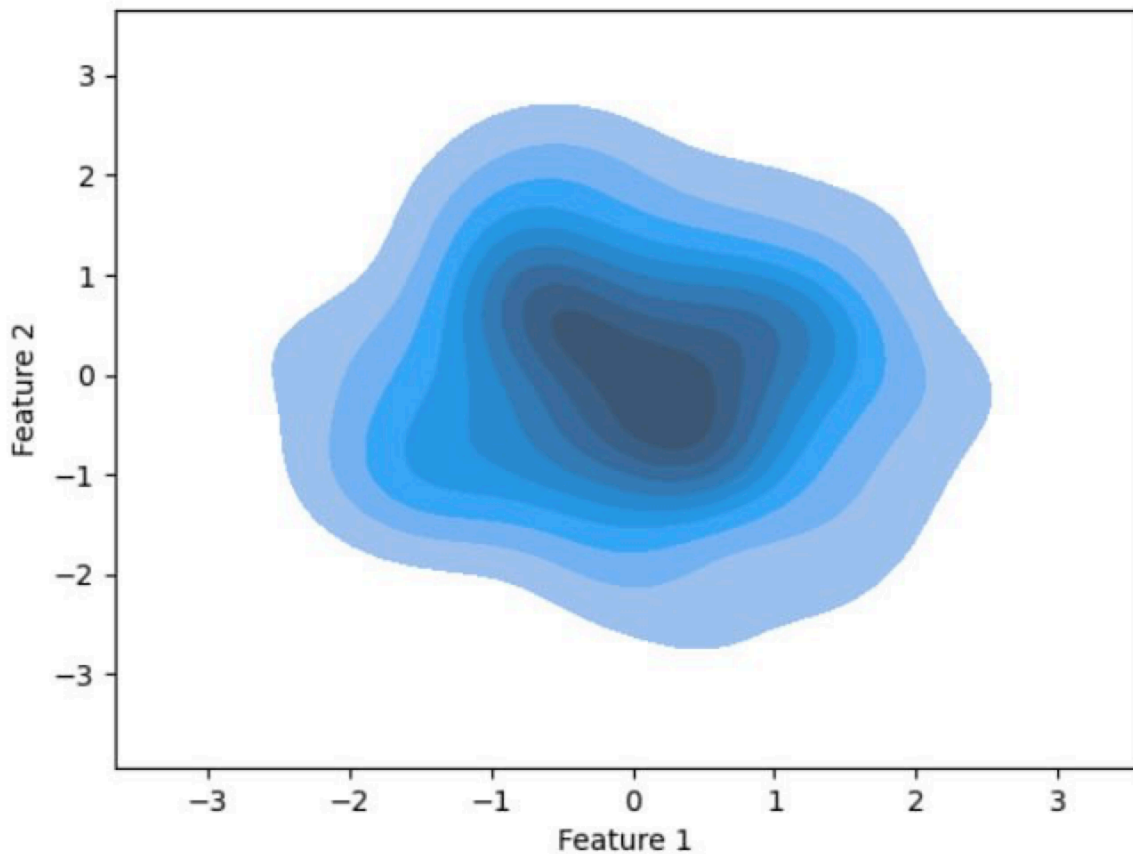
## Основной код

Основной код, отвечающий за реализацию и исследование различных методов SGD, находится в файлах `main.py`, `bonus_1.py`, и `plotting.py`. Функции и методы были реализованы с возможностью изменения гиперпараметров для дальнейшего анализа.

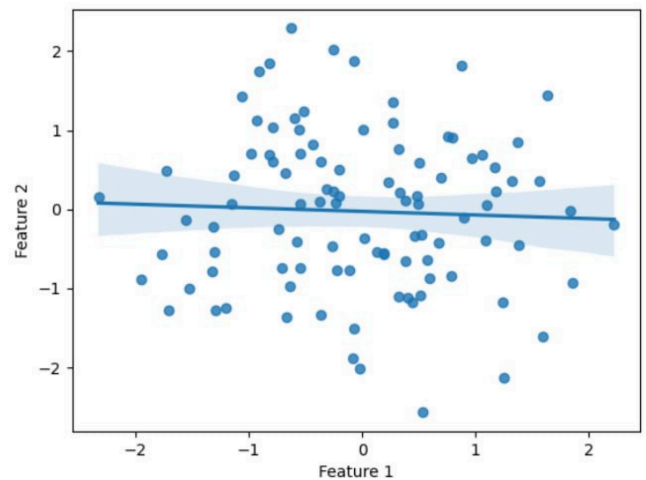
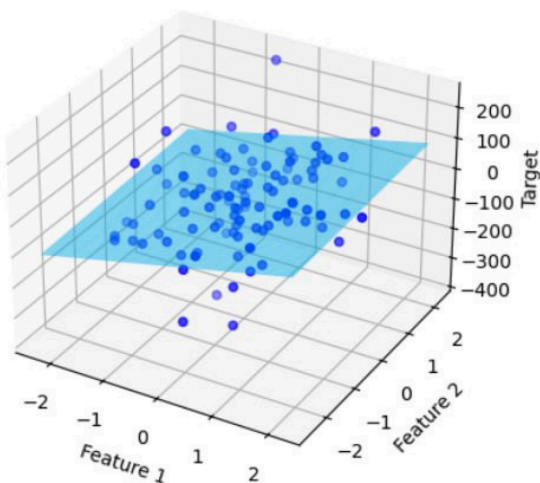
## Результаты исследования

Исследование проводилось на синтетических данных, сгенерированных для задачи линейной регрессии. Для сравнения эффективности различных методов были проведены замеры времени выполнения, потребляемой памяти и итоговых весов модели.

## Графики



На рисунке 1 представлен график распределения данных по двум признакам.



На рисунке 2 представлен 3D график зависимости целевой переменной от двух признаков.

## Анализ результатов

- **Потребление памяти:** Наименьшее потребление памяти показал метод custom SGD с batch\_size=2 и со step decay learning rate schedule, потребляя 0.0 MB.
- **Время выполнения:** Наименьшее время выполнения показал метод custom SGD с step decay learning rate schedule (0.028 секунд).
- **Итоговые веса:** Все методы показали схожие итоговые веса, что говорит о правильной реализации.

## Доп. задание 1

В данном бонусном задании необходимо реализовать и исследовать эффективность стохастического градиентного спуска (SGD) для полиномиальной регрессии с добавлением регуляризации различных методов (L1, L2, Elastic регуляризация).

Регуляризация применяется для предотвращения переобучения модели путем добавления штрафа за сложность модели

- L1 регуляризация (Lasso) добавляет штраф в виде абсолютной величины коэффициентов.
- L2 регуляризация (Ridge) добавляет штраф в виде квадрата величины коэффициентов.
- Elastic Net комбинирует обе техники, применяя как L1, так и L2 регуляризации.

## Анализ

- Без регуляризации: высокая точность, но переобучение.
- L1 регуляризация: улучшает интерпретируемость модели.
- L2 регуляризация: устойчива к шуму данных.
- Elastic Net: сочетает преимущества L1 и L2, но требует больше времени.

## Заключение

Регуляризация помогает предотвратить переобучение и улучшает обобщающую способность модели. Использование SGD ускоряет обучение на больших данных.

## Доп. задание 2

Разберем задачу оптимизации в методе опорных векторов (SVM). SVM используется для классификации. Цель - найти гиперплоскость, которая максимально разделяет два класса в данных. Пример задачи:

Пусть у нас есть набор данных с двумя классами в двумерном пространстве. Цель - найти такую линию (гиперплоскость), которая отделяет точки одного класса от точек другого класса с максимальным отступом.

Формальная формулировка задачи оптимизации:

$$\min_{w,b} (1/2 * ||w||^2)$$

при ограничениях:

$$y_i (w * x_i + b) \geq 1$$

где:

- **w** - вектор весов, определяющий направление гиперплоскости.
- **b** - смещение гиперплоскости относительно начала координат.
- **y<sub>i</sub>** - метка класса (1 или -1) для (i)-го объекта.
- **x<sub>i</sub>** - вектор признаков (i)-го объекта.

Решение:

- Преобразование данных: Если данные линейно неразделимы, используется ядровая функция для преобразования данных в более высокое пространство признаков, где они могут быть линейно разделимы.
- Алгоритм оптимизации: Метод последовательного минимального оптимизирования (SMO) эффективно решает задачу, разбивая её на подзадачи и обновляя веса итеративно.
- Решение задачи оптимизации: Метод находит вектор весов ( $\mathbf{w}$ ) и смещение  $b$ , которые минимизируют функцию потерь, удовлетворяя ограничениям.

Пример:

Пусть у нас есть двумерные данные с двумя классами. SVM находит гиперплоскость, которая разделяет эти точки с максимальным отступом, обеспечивая высокую обобщающую способность модели.

## Выводы

Методы SGD и его модификации продемонстрировали высокую эффективность в решении задачи линейной регрессии. Метод Adam показал наилучшее сочетание скорости сходимости и стабильности, в то время как методы Momentum и Nesterov продемонстрировали хорошие результаты по времени выполнения и потреблению памяти. Преимущества и ограничения методов:

1. SGD: Быстрота и малое потребление памяти, но высокая вероятность застревания в локальных минимумах.
2. Momentum: Ускоряет сходимость, снижая вероятность застревания.
3. Nesterov: Улучшает результаты Momentum за счёт предсказания обновления.
4. AdaGrad: Хорош для разреженных данных, но требует больших ресурсов памяти.
5. RMSProp: Исправляет недостатки AdaGrad, стабилизируя шаг обучения.
6. Adam: Комбинирует лучшие стороны Momentum и RMSProp, обеспечивая быструю и стабильную сходимость.

## Заключение

Лабораторная работа показала, что методы стохастического градиентного спуска и его модификации являются мощными инструментами для оптимизации линейных моделей. Их использование позволяет значительно улучшить качество и скорость обучения моделей машинного обучения.

# Таблица результатов

	Method	Memory Used	Time Taken
0	Custom SGD	0.046875	0.030722
1	Custom SGD with batch_size=2	-0.140625	0.025757
2	Custom SGD with step decay learning rate schedule	0.000000	0.032907
3	SGD	19.078125	6.378374
4	SGD	2.750000	6.365443
5	SGD	1.765625	6.454462
6	adagrad	0.296875	6.386738
7	rmsprop	2.296875	6.408670
8	adam	5.484375	6.521460

## Weights

[36.66862863381354, 6.30253057075465, 75.19033856002798, 59.97656563822274, 64.80486912495998]  
[36.68459537535625, 6.294228080099635, 75.18955106399918, 59.976390124722876, 64.80835710511667]  
[37.36411577395337, 5.627766310768766, 73.41174955986912, 57.41181803096993, 63.96101324384286]  
[36.668457, 6.303078, 75.18736, 59.972927, 64.80592]  
[36.6711, 6.3069715, 75.1811, 59.981583, 64.81481]  
[36.669235, 6.3005238, 75.18693, 59.95941, 64.809586]  
[36.666946, 6.305151, 75.19353, 59.97626, 64.81033]  
[36.68656, 6.28848, 75.183655, 59.998672, 64.801414]  
[36.66367, 6.298945, 75.18573, 59.97252, 64.80805]