MetOpt3

Дроздов Олег, Широков Данил, Исламова Камиля

june 2025

1 Введение

В своей работе мы рассматриваем задачу многомерной линейной регрессии, применяемой для предсказания качества вина по его химическим характеристикам. В качестве метода обучения используется стохастический градиентный спуск (SGD), а также его модификация с моментом (SGD Momentum). Основная цель — минимизировать функцию ошибки, вычисляемую по методу наименьших квадратов, и исследовать влияние различных параметров оптимизации на качество модели.

2 Описание методов

2.1 Арифметические операции

Реализованы сложение, умножение, вычитание, деление, а также их векторные аналоги — для учета количества произведенных арифметических операций

2.2 sgd regressor

Метод многомерной линейной регрессии. В нем мы пытаемся обучить модель угадывать качество вина по 10-ти бальной системе оценки.

- здесь объектом служит экземпляр вина, его свойства (x-вектор) - различные химические показатели жидкости. Каждый объект относится к одному из 10-ти классов (оценка от 1 до 10).

- используем метод наименьших квадратов для обучения модели (пусть оценку вина она выдает дробным числом), тогда функция ошибки зависит от параметров модели $L(w, x, y)$ -> $(w * x_i - y_i)^2$.
- метод поддерживает различные стратегии выбора шага (константный, затухание со временем, экспоненциальное затухание).
- кроме того есть возможность использовать регуляризацию $(L1(Lasso), L2(Ridge), Elastic)$.
- гиперпараметры: $learning\_rate, batch\_size, decay\_rate, alpha, l1\_ratio$

2.3 SGD Momentum

(находится в отдельном файле)

- библиотечная реализация (torch.optim) модификации SDG с добавлением момента.
- момент - корректировочный вектор накапливающий в себе предыдущие градиенты. Такой подход помогает не застревать в локальных минимумах, как бы проскакивая их по инерции.
- зависит от двух параметров - momentum (0,9), отвечающий за степень влияния на шаг накопленных градиентов (инерция градиента), и learning rate (0,01), обозначающий сам шаг.

3 Объект исследования

Ссылка на датасет

Нами был выбран датасет, содержащий в себе данные о вине, его химический состав, значения сахаров, хлоридов, pH, процент алкоголя и т.п. с оценкой качества от 1 до 10.

Данные датасета были обработаны для более хорошей работы SGD, а именно отнормированы, т.е из каждого свойства объекта было вычтено среднее значение этого свойство и результат поделен на значение дисперсии по этому свойству. Например:

Так данные станут более однородными, и нам не нужно будет обращать внимание на разные единицы измерения в тех или иных показателях.

№	alcohol (X)	sulfur dioxide (X)	alcohol (scaled)	sulfur dioxide (scaled)
1	13.5	30	0.0	-1.0
2	14.0	50	1.0	1.0
3	13.0	40	-1.0	0.0

4 Функция

Для модели был выбран метод наименьших квадратов и итоговая функция средней ошибки выглядит следующим образом:

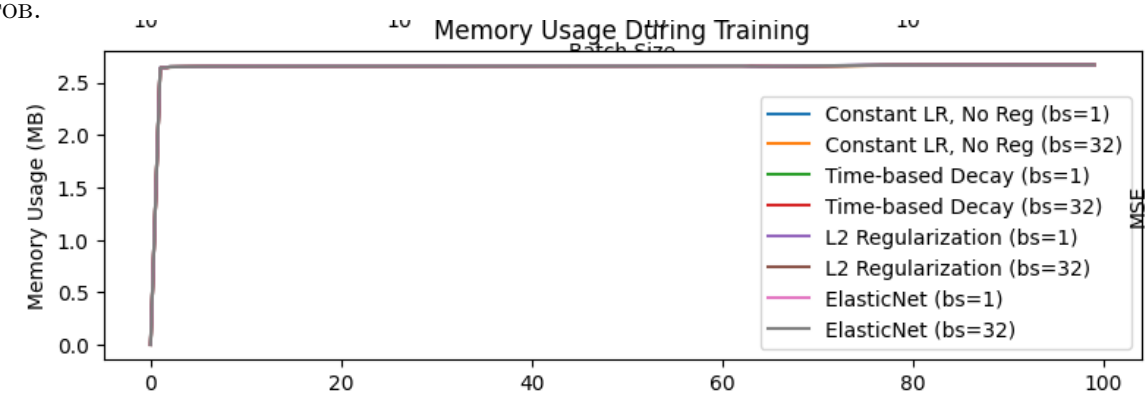
$$\frac{1}{N}\sum_{i=1}^{N}(wx_i - y_i)^2$$

где w - вектор параметров модели (размер 12), x - вектор свойств объекта и y - оценка объекта. Получается многомерная линейная регрессия.

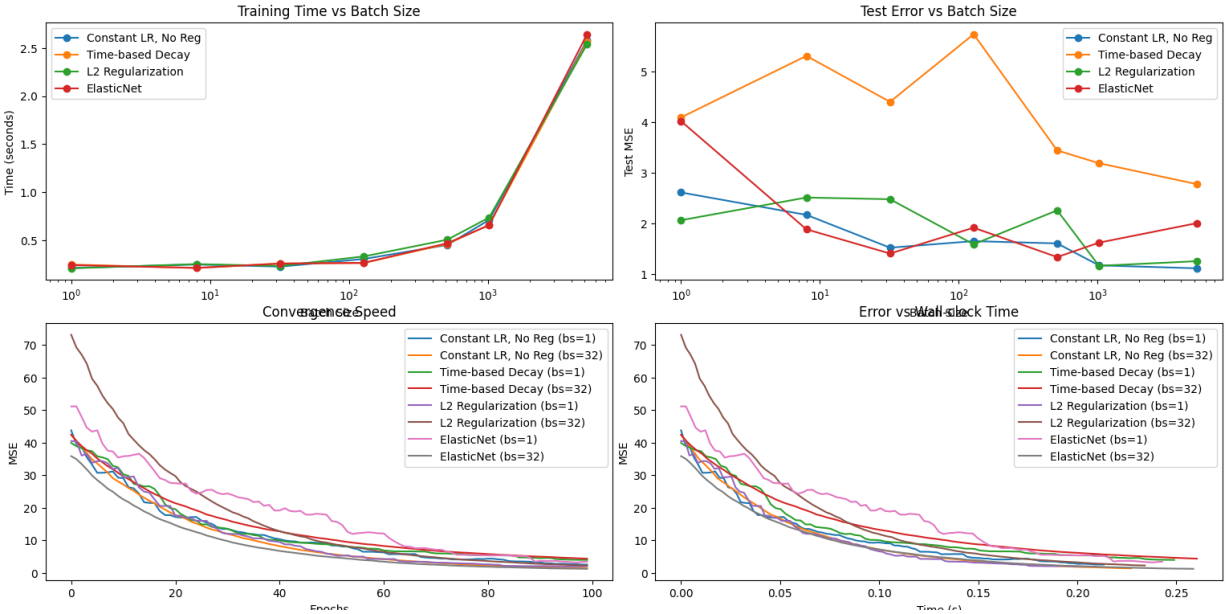
5 Результаты исследований

5.1 Затраты по памяти

Тут у нас получились практически одинаковые результаты для всех тестов.



5.2 Суммарная ошибка и общее время исполнения при разных значениях batch size



Как видно из графиков, на данном датасете (в нем 6700 объектов) лучше всего себя показывают запуски со средним размером батча (512 и 1024) с использованием Elastic регуляризации. Такой размер выборки не сильно усложняет вычисления, но показывает неплохую точность, а регуляризация не дает модели переобучиться, кроме того в нашем наборе данных у объектов много коррелирующих и связанных друг с другом свойств, а некоторые из них, не сильно влияют на выставленную оценку, а потому elastic особенно эффективен. Естественно самый точный способ - считать градиент сразу на всем датасете, но это очень дорого.

6 Таблица

Constant LR, No Reg		
Batch Size	Арифметические операции	Вычисления градиента
1	8700	100
8	44400	800
32	166800	3200
128	656400	12800
512	2614800	51200
1024	5226000	102400
5197	26508300	519700
Time-based Decay		
Batch Size	Arithmetic Operations	Gradient Calculations
1	9000	100
8	44700	800
32	167100	3200
128	656700	12800
512	2615100	51200
1024	5226300	102400
5197	26508600	519700

L2 Regularization		
Batch Size	Arithmetic Operations	Gradient Calculations
1	9800	100
8	45500	800
32	167900	3200
128	657500	12800
512	2615900	51200
1024	5227100	102400
5197	26509400	519700
ElasticNet		
Batch Size	Arithmetic Operations	Gradient Calculations
1	15300	100
8	51000	800
32	173400	3200
128	663000	12800
512	2621400	51200
1024	5232600	102400
5197	26514900	519700

7 Выводы

7.1 Эффективность методов оптимизации

- SGD с моментом (SGD-Momentum) показал более стабильную и быструю сходимость по сравнению с обычным SGD, особенно при небольших размерах батча (1, 32). Это связано с тем, что момент помогает преодолевать локальные минимумы и ускоряет обучение в направлениях с устойчивым градиентом.

- обычный SGD демонстрирует более высокую чувствительность к выбору размера батча и гиперпараметров, что может приводить к колебаниям в процессе обучения.

7.2 Влияние размера батча

- большие батчи обеспечивают более точные обновления параметров, но требуют больше времени на обучение из-за частых вычислений градиента.

- малые батчи (1, 8) ускоряют обучение, но могут снижать точность модели из-за менее точных обновлений. Однако на данном датасете (6700 объектов) оптимальным оказался средний размер батча (512), который сочетает приемлемую скорость и точность.

7.3 Регуляризация

- ElasticNet показал наилучшие результаты, так как комбинирует преимущества L1 и L2 регуляризации. Это особенно полезно для датасета с множеством коррелирующих признаков. Но естественно он не бесплатен и утяжеляет вычисления, делая градиент сложным.

7.4 Стратегии выбора шага обучения

- затухание шага (Time-based Decay) негативно повлияло на сходимость модели, судя по всему для нее был неудачно выбран начальный шаг.

- постоянный шаг (Constant LR) требовал более тщательного подбора learning rate, чтобы избежать расходимости или медленного обучения.

7.5 Вычислительные затраты

Количество арифметических операций и вычислений градиента линейно зависит от размера батча. Наибольшие затраты наблюдались при размере батча, равном всему датасету (5197 объектов), что подтверждает целесообразность использования средних батчей (512, 1024).

8 Конец

Спасибо за внимание!