Метод стохастического градиентного спуска (SGD) и его модификации

Лабораторная работа №3

Иванов Владимир, М3235
Шепелев Матвей, М3235
Зубарев Денис, М3235
Команда ООО “АДВЧИПСГРУПП”

1 Постановка задачи

В данной работе исследуется метод стохастического градиентного спуска и его модификации.

1.1 SGD

Стохастический градиентный спуск изменяет параметры w следующим образом:

$$\omega := \omega - \alpha \nabla_{\omega} L(\omega, x_i, y_i)$$

где i выбирается случайно.

1.2 SGD с батчем

Также есть модификация SGD, которая вместо одного элемента суммы использует ее некоторый набор, т.е:

$$\omega := \omega - \alpha \sum_{i \in \{i_j\}} \nabla_{\omega} L(\omega, x_i, y_i)$$

где набор $\{i_j\}$ выбирается случайно.

2 Реализованные методы (модификации)

Реализован SGD с:

1. Выбором размера батча
2. Возможностью выбора регуляризации (реализованы L1, L2, Elastic)
3. Модификацией Momentum.
4. Различными `LearningRateFunc` для Learning Rate Scheduling.

На каждой эпохе батч выбирается вероятностно.

Количество эпох, размер батча, регуляризация и функция выбора шага вынесены в сигнатуру метода.

Также для сравнения использован SGD из библиотеки torch (`torch.optima`).

3 Результаты исследования

Для исследования использовались точки полинома $f(x) = 4 + 3x + 2x^2 + x^3$ (плюс зашумление).

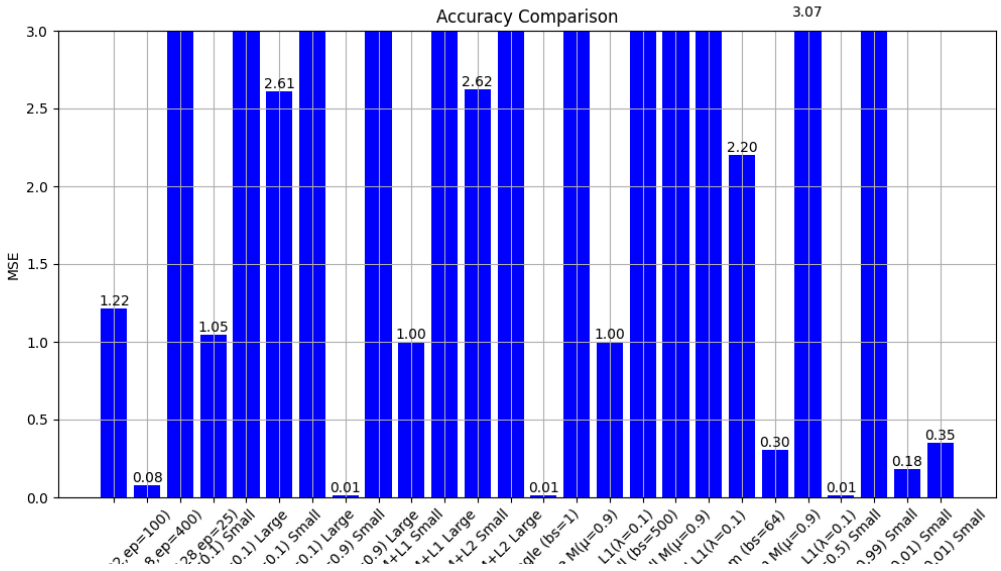

Рис. 1: сравнение погрешности

Наименьшую погрешность (== значение функции среднеквадратической ошибки) имеют SGD с инерцией и маленьким размером батча.

Рис. 2: сравнение времени исполнения

Чтобы сравнение было (примерно) справедливым, у каждого SGD количество эпох подобрано таким образом, чтобы время исполнения было +- одинаковым.

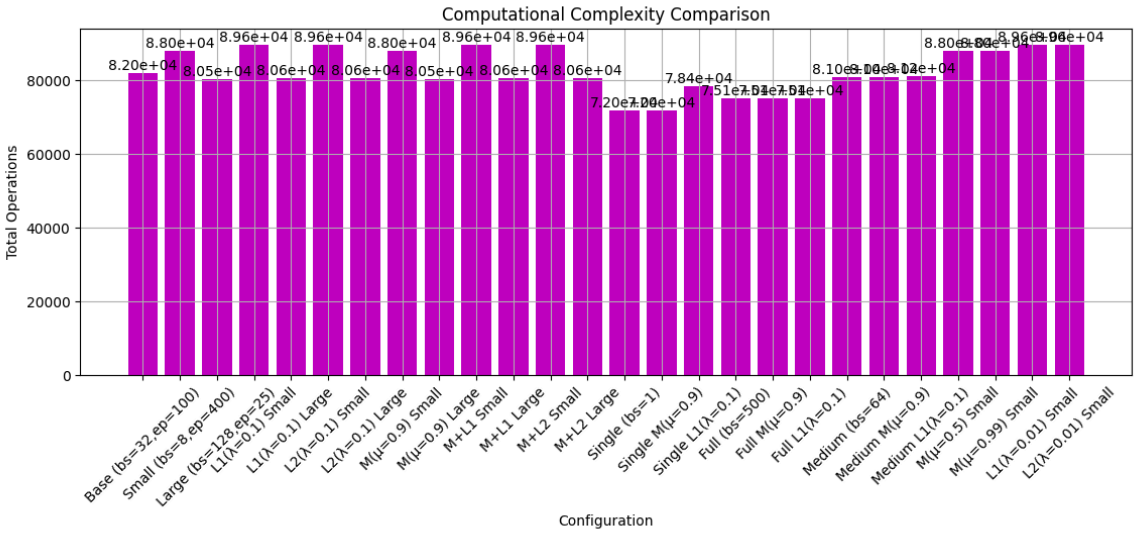

Рис. 3: сравнение количества операций

Наименьшее количество операций потребовалось стандартному SGD (с batch=1). Наибольшее - SGD с инерцией и регуляризацией L1/L2.

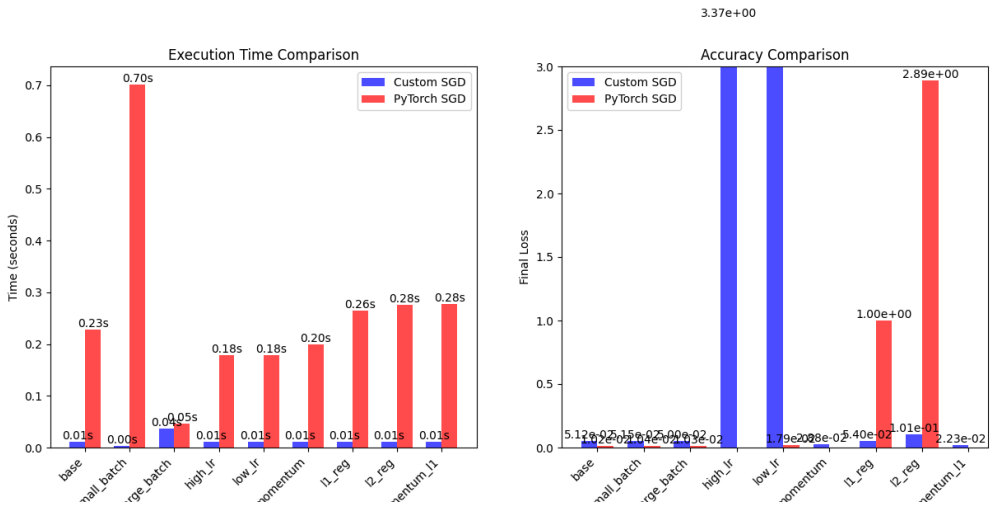

Рис. 4: сравнение с реализацией из `torch`

На удивление, собственная реализация SGD оказалась более эффективной, чем реализация `torch.optima` (за исключением экстремальных случаев с очень низким/высоким learning rate). Собственный ГС сходится заметно быстрее с более низкой погрешностью в случае использования регуляризации.

4 Активное обучение

В отличие от пассивного (того, которое изучалось в этой лабораторной) обучения, в активном обучении классифицирована только часть

датасета. Модель анализирует оставшуюся часть датасета и выбирает некоторые точки, классификацию которых спрашивает у оракула. Эти точки добавляются в размеченную область датасета и процесс повторяется до достижения нужной точности.

Если формализовывать:

$$L(f(x), y) \to \min$$

при этом $|\{x_i, y_i\}| < |\{x_j, y_j\}| + B$

где $\{x_i, y_i\}$ - набор, для которого точно известна классификация, $\{x_j, y_j\}$ - размеченный изначально набор, B - максимальное количество запросов к оракулу.

Одна из эвристик выбора точек, которые передаются оракулу - выбор точек с минимальным отступом (margin).

$$\Phi_x = P(y_1|x) - P(y_2|x)$$

где y_1 - класс, к которому наиболее вероятно по мнению модели относится точка, y_2 - второй по вероятности.

Если отступ велик, то велика и “уверенность” модели в классификации точки, поэтому имеет смысл запрашивать точки с минимальным отступом.

Примером задачи может стать модель, отличающая ручно написанные числа. Получать достаточно изображений и классифицировать их руками времязатратно, поэтому здесь отлично подойдет активное обучение.

С помощью выбора точек с минимальным отступом применение этой эвристики будет выглядеть так: например, модель имеет в неразмеченной части датасета точку x - неразборчиво написанную 4. Пусть $P(y_4|x) - P(y_9|x) = 0.01$ (т.е модель считает событие $x \in y_4$ примерно одинаково вероятным с тем, что $x \in y_9$). Тогда она отдаст эту точку человеку для классификации, и таким образом уменьшается неопределенность в критической области.

5 Вывод

Стохастичесикй градиентный спуск показал достаточно высокую эффективность относительно полного ГС.

Дополнительные оптимизации, такие как Momentum, заметно ускоряют сходимость алгоритма.

При этом мини батч не уменьшил количество эпох для сходимости на данном простеньком датасете.

Собственная реализация SGD оказалась более эффективной, чем библиотечная, так что, возможно, на более простых датасетах выгодно использовать более “легковесную” реализацию.