УДК 681.3.06
В. П. Марценюк, докт. техн. наук, проф., Н. В. Мілян
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ОГЛЯД МЕТОДІВ ОПТИМІЗАЦІЇ В МАШИННОМУ НАВЧАННІ: ГРАДІЄНТНИЙ СПУСК ТА СТОХАСТИЧНИЙ ГРАДІЄНТНИЙ СПУСК

V. P. Martsenyuk, Dr., Prof., N. V. Milian
REVIEW OF OPTIMIZATION METHODS IN MACHINE LEARNING: GRADIENT DESCENT AND STOCHASTIC GRADIENT DESCENT

У галузі машинного навчання найбільш часто використовуються методи оптимізації першого порядку, які в основному базуються на градієнтному спуску.

Градієнтний спуск: метод градієнтного спуску є найбільш раннім і найпоширенішим методом оптимізації. Ідея методу градієнтного спуску полягає в тому, що змінні ітеративно оновлюються у (протилежному) напрямку градієнтів цільової функції. Оновлення виконується для поступового зближення до оптимального значення цільової функції. Швидкість навчання η визначає розмір кроку в кожній ітерації, таким чином, впливає на кількість ітерацій, щоб досягти оптимального значення [6].

Алгоритм найшвидшого спуску – широко відомий алгоритм. Ідея полягає у тому, щоб вибрати відповідний напрямок пошуку на кожній ітерації, щоб значення цільової функції мінімізувалося якнайшвидше. Градієнтний спуск і найшвидший спуск – це не одне і те ж, оскільки напрямок негативного градієнта не завжди спускається найшвидше. Градієнтний спуск – приклад використання норми Евкліда при крутому спуску [5].

Для моделі лінійної регресії вважається, що $f_\theta(x)$ – це функція, яку слід вивчити, $L(\theta)$ – функція втрат, а θ – параметр, який слід оптимізувати. Мета – мінімізувати функцію втрат за допомогою

$$L(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(y^i - f_\theta(x^i))^2 \tag{1}$$

$$f_\theta(x) = \sum_{j=1}^{D}\theta_j x_j \tag{2}$$

де N – кількість навчальних зразків, D – кількість вхідних характеристик, x^i – незалежна змінна з $x^i = (x_1^i,..., x_D^i)$ для $i = 1,..., N$ і y^i – цільовий результат. Градієнтний спуск чергує наступні два кроки, поки не сходиться:

1) Вивести $\mathrm{L}(\theta)$ для θ_j, щоб отримати градієнт, відповідний кожному θ_j:

$$\frac{\partial L(\theta)}{\partial \theta_j} = -\frac{1}{N}\sum_{i=1}^{N}(y^i - f_\theta(x^i))x_j^i \tag{3}$$

2) Оновити кожну θ_j у негативному напрямку градієнта, щоб мінімізувати функцію втрат:

$$\theta_j^{'} = \theta_j + \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (y^i - f_\theta(x^i)) x_j^i \qquad (4)$$

Метод градієнтного спуску простий у реалізації. Рішення є глобально оптимальним, коли цільова функція опукла. Часто вона сходиться з меншою швидкістю, якщо змінна ближче до оптимального рішення і потрібно проводити більш ретельну ітерацію.

У наведеному вище прикладі лінійної регресії потрібно звернути увагу, що всі навчальні дані використовуються на кожному етапі ітерації, тому метод градієнтного спуску також називають пакетним градієнтним спуском. Якщо кількість зразків дорівнює N, а розмірність x дорівнює D, складність обчислень для кожної ітерації буде $O(ND)$. Для зменшення витрат на обчислення були запропоновані деякі методи розпаралелювання [2], [3]. Однак вартість все ще важко прийняти при роботі з великомасштабними даними. Таким чином, з'являється метод стохастичного градієнтного спуску.

Оскільки пакетний градієнтний спуск має високу обчислювальну складність у кожній ітерації для великомасштабних даних і не дозволяє оновлення через Інтернет, було запропоновано стохастичний градієнтний спуск (SGD) [1]. Ідея стохастичного градієнтного спуску полягає у використанні однієї вибірки випадковим чином для оновлення градієнта за ітерацію, а не безпосереднього обчислення точного значення градієнта. Стохастичний градієнт є неупередженою оцінкою реального градієнта [1]. Вартість алгоритму стохастичного градієнтного спуску не залежить від чисел вибірки і може досягти сублінійної швидкості збіжності [4]. SGD скорочує час оновлення для роботи з великою кількістю вибірок і видаляє певну кількість обчислювальної надмірності, що значно прискорює обчислення. У сильній опуклій задачі SGD може досягти оптимальної швидкості збіжності.

Література

1. H. Robbins and S. Monro, "A stochastic approximation method," The Annals of Mathematical Statistics, pp. 400–407, 1951.

2. J. Alspector, R. Meir, B. Yuhas, A. Jayakumar, and D. Lippe, "A parallel gradient descent method for learning in analog VLSI neural networks," in Advances in Neural Information Processing Systems, 1993, pp. 836–844.

3. J. Nocedal and S. J. Wright, Numerical Optimization. Springer, 2006.

4. R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in Advances in Neural Information Processing Systems, 2013, pp. 315–323.

5. S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.

6. S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.