

УДК 004.89:004.032.26

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ АЛГОРИТМУ ГРАДІЄНТНОГО СПУСКУ

Ткаченко М.А.

Науковий керівник – к.т.н., доц. Валенда Н.А.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

тел.: +38 (057) 702-10-13, email: maksym.tkachenko2@nure.ua

This paper explains the concept of gradient descent, a popular optimization algorithm used in machine learning to find the optimal parameters of a model. The paper introduces the concept of a cost function, which is used to measure the difference between the predicted and actual values in a machine learning model. The aim of gradient descent is to minimize this cost function by finding the optimal set of model parameters. To illustrate the concept of gradient descent, the paper uses a simple example of a linear regression model. Overall, the paper provides a clear and concise explanation of the key concepts and steps involved in the gradient descent algorithm, making it a useful resource for anyone interested in understanding machine learning optimization techniques.

Градiєнтний спуск – це фундаментальний алгоритм навчання нейронних мереж. У машинному навчанні ми багато чого оптимізуємо. Коли ми підганяємо пряму лінію за допомогою лінійної регресії, ми оптимізуємо перехват і нахил. Якщо ми навчимося оптимізувати пряму лінію за допомогою градiєнтного спуску, то ми вивчимо стратегію, яка оптимізує багато інших проблем оптимізації, з якими ми стикаємося в машинному навчанні [1].

Лінійна регресія – це підгонка лінії до точок даних за допомогою градiєнтного спуску. Обрана функція витрат – це сума квадратичних помилок. Взнявши похідні від нахилу і перетинів, ми дізнаємося, де знаходяться оптимальні значення для найкращої підгонки. І ніхто ніколи не вирішує це завдання вручну. Це робиться за допомогою комп'ютера. Важливо зрозуміти концепти. Великий важливий концепт полягає в тому, що ми хочемо мінімізувати квадрат відстані між фактичними значеннями даних і підігнаною лінією для прогнозів. Фінальна підігнана пряма лінія дає найменші квадрати помилок між собою і реальними даними. Градiєнтний спуск знаходить мінімальне значення, роблячи кроки від початкового випадкового припущення до досягнення найкращих значень нахилу і перетину з найменшою помилкою підгонки. Наведемо алгоритм градiєнтного спуску крок за кроком [1]:

Крок 1. Візьміть похідну функції витрат для кожного параметра в ній. На чудернацькому жаргоні машинного навчання, візьміть градiєнт функції витрат.

Крок 2. Виберіть випадкові значення для параметрів.

Крок 3. Підставте значення параметрів у похідні.

Крок 4. Обчисліть розміри кроків.

$$\text{Розмір кроку} = \text{Нахил} * \text{Швидкість навчання}$$

Крок 5. Обчисліть нові параметри.

$$\text{Новий параметр} = \text{Старий параметр} - \text{Розмір кроку}$$

Крок 6. Поверніться до кроку 3 і повторюйте доти, доки розмір кроку не наблизиться до 0 або доки ви не досягнете максимальної кількості кроків.

Можливості застосування алгоритму градієнтного спуску наступні: алгоритм градієнтного спуску є потужним інструментом для навчання нейронних мереж, оскільки він дозволяє зменшити функцію витрат шляхом коригування ваг мережі. Застосування алгоритму градієнтного спуску може забезпечити швидку збіжність процесу навчання, зменшуючи час, необхідний для досягнення заданих витрат. Основною перевагою алгоритму градієнтного спуску є його універсальність – він може бути використаний для навчання будь-якої нейронної мережі.

Особливості алгоритму градієнтного спуску наступні: використання алгоритму градієнтного спуску може призвести до проблеми "застрягання" в локальному мінімумі чи сідловій точці функції витрат, що може перешкодити досягненню глобального мінімуму [2]. Використання алгоритму градієнтного спуску, зокрема в рекурентних нейронних мережах, може призвести до проблеми зникаючих та вибухаючих градієнтів, в результаті чого алгоритм перестає вчитися [2]. При використанні алгоритму градієнтного спуску, необхідно враховувати його гіперпараметр, такий як швидкість навчання, щоб забезпечити отримання найменших витрат на тестових даних.

Узагальнюючи, дослідження спрямоване на вивчення внутрішньої роботи алгоритму градієнтного спуску та його покращення в контексті навчання нейронних мереж. Отримані в результаті дослідження роботи алгоритму висновки можуть допомогти розробити нові алгоритми для навчання нейронних мереж та покращити існуючі шляхом модифікації алгоритму градієнтного спуску для подолання проблеми "застрягання" в локальному мінімумі, а також створення нових архітектур нейронних мереж для подолання проблеми зникаючих і вибухаючих градієнтів. В цьому нам допоможе вивчена стратегія як працює градієнтний спуск, що є фундаментальним алгоритмом для машинного навчання.

Список використаних джерел:

1. StatQuest with Josh Starmer. Gradient Descent, Step-by-Step, 2019. YouTube. URL: <https://www.youtube.com/watch?v=sDv4f4s2SB8> (дата звернення: 05.04.2023).

2. What is Gradient Descent? IBM. URL: <https://www.ibm.com/topics/gradient-descent> (дата звернення: 09.04.2023).