

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 51-74

МЕТОД ГРАДИЕНТНОГО СПУСКА В МАШИННОМ ОБУЧЕНИИ

Мартынов Андрей Витальевич, студент, направление подготовки 09.03.04 Программная инженерия, Оренбургский государственный университет, Оренбург
e-mail: andry.martynov16.08.2003@gmail.com

Кандыба Владимир Евгеньевич, студент, направление подготовки 09.03.04 Программная инженерия, Оренбургский государственный университет, Оренбург
e-mail: nemmen222@bk.ru

Научный руководитель: **Крючкова Ирина Викторовна**, кандидат технических наук, доцент, доцент кафедры прикладной математики, Оренбургский государственный университет, Оренбург
e-mail: kachmf@yandex.ru

***Аннотация.** Метод градиентного спуска часто применяется в машинном обучении, в частности в нейронных сетях. В современном мире множество компаний для работы применяют искусственный интеллект, в котором используются нейронные сети, способные обучаться за счет метода градиентного спуска, в основе которого лежит нахождение минимального значения функции. Актуальность исследования обусловлена необходимостью нахождения такого способа оптимизации, который был бы способен обработать большие объемы данных за относительно короткий промежуток времени, именно этими свойствами и обладает метод градиентного спуска. Целью же исследования выступает определение преимуществ метода градиентного спуска в машинном обучении. В процессе исследования использовались общенаучные методы и подходы. В результате появилась программа для создания функции по аппроксимирующим данным.*

***Ключевые слова:** машинное обучение, градиент, градиентный спуск, функция потерь, алгоритм.*

***Благодарности:** статья подготовлена в рамках исследования, проводимого в ходе реализации стратегического проекта «Технологии и кадры для ОПК», выполняемого по программе стратегического академического лидерства «Приоритет-2030».*

***Для цитирования:** Мартынов А. В., Кандыба В. Е. Метод градиентного спуска в машинном обучении // Шаг в науку. – 2022. – № 4. – С. 4–8.*

GRADIENT DESCENT METHOD IN MACHINE LEARNING

Martynov Andrey Vitalievich, student, training program 09.03.04 Software engineering, Orenburg State University, Orenburg
e-mail: andry.martynov16.08.2003@gmail.com

Kandyba Vladimir Evgenievich, student, training program 09.03.04 Software engineering, Orenburg State University, Orenburg
e-mail: nemmen222@bk.ru

Research advisor: **Kryuchkova Irina Viktorovna**, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Applied Mathematics, Orenburg State University, Orenburg
e-mail: kachmf@yandex.ru

***Abstract.** The gradient descent method is often used in machine learning, in particular in neural networks. In today's world, many companies use artificial intelligence to work, which uses neural networks that can be trained using the gradient descent method, which is based on finding the minimum value of a function. The relevance of the study is due to the need to find such an optimization method that would be able to process large amounts of data in a relatively short period of time, these are the properties that the gradient descent method has. The purpose*

of the study is to determine the advantages of the gradient descent method in machine learning. In the process of research, general scientific methods and approaches were used. As a result, a program appeared for creating a function from approximating data.

Key words: machine learning, gradient, gradient descent, loss function, algorithm.

Acknowledgments: the article was prepared as part of a study conducted during the implementation of the strategic project «Technologies and personnel for the defense industry», carried out under the program of strategic academic leadership «Priority 2030».

Cite as: Martynov, A. V., Kandyba, V. E. (2022) [Gradient descent method in machine learning]. *Shag v nauku* [Step into science]. Vol. 4, pp. 4–8.

Актуальность изучения нейронных сетей подтверждается их многочисленными применениями в различных областях человеческой деятельности. К ним относятся: автоматизация процессов распознавания образов, прогнозирование, адаптивное управление, создание экспертных систем и организация ассоциативной памяти [5].

Возможности быстрого обучения и дообучения нейросетевых экспертных систем позволяют им отражать характеристики быстро меняющегося внешнего мира и оперировать практическими знаниями способом, который требует меньше времени и труда, чем традиционная формализация знаний человеческими экспертами [4]. Актуальность исследований, связанных с нейронными сетями, обусловлена тем, что человеческий мозг обрабатывает поступающую информацию иначе, чем цифровые методы обработки.

Целью же исследования выступает определение преимуществ метода градиентного спуска в машинном обучении. В процессе исследования были изучены различные теоретические подходы применения градиентного спуска, была создана программа: для подбора параметров аппроксимирующей функции, для демонстрации работы градиентного спуска на практике.

Метод градиентного спуска – это общепринятая стратегия для оптимизации, используемая в глубоком обучении и машинном обучении.

Градиент – это вектор, который по своему направлению представляет наибольшее увеличение функции. В математическом анализе градиентом называют вектор, координаты которого являются частными производными функции, то есть, если задана функция $f(x, y, z)$, то градиент рассчитывается по следующей формуле [2]:

$$\text{grad } f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right). \quad (1)$$

Открытие метода градиентного спуска принадлежит Огюстену Луи Коши, впервые предложившему данный метод в 1847 г., однако свойства сходимости для задач нелинейной оптимизации были впервые изучены Хаскеллом Карри только в 1944 году [6]. После чего метод стал популярным и часто используемым.

Идея градиентного подхода заключается в выполнении повторяющихся шагов в направлении, противоположном градиенту функции в данной точке. Фактически с каждой итерацией алгоритма мы все ближе и ближе подходим к минимуму функции [7]. Поэтому этот метод можно использовать для получения наименьшего значения ошибки или для нахождения весов при обучении нейронных сетей. Веса – это значения, которые указывают на важную информацию, при обучении нейронных сетей с использованием «учителя».

Существует три типа градиентного спуска, рассмотрим каждый из них:

Пакетный градиентный спуск, также известный как ванильный градиентный спуск. Пакетный градиентный спуск вычисляет ошибку для каждого набора данных в обучающей выборке. Суть алгоритма

в том, что мы шаг за шагом приближаемся к минимуму функции среднеквадратичной ошибки. Среднеквадратичная ошибка – это функция, определяющая квадрат разницы между значением, полученным алгоритмом, и идеальным значением. Следовательно, пакетный градиентный спуск имеет несколько недостатков, так как он считает ошибку, усредняя ее на всей выборке. Во-первых, вычисление ошибки для каждого набора занимает очень много времени, потому что нужно вычислять градиент для каждого параметра. Во-вторых, и это главное, если вычисления привели к локальному минимуму, то выйти из него невозможно, а истинное значение соответствует глобальному минимуму [3].

Есть два способа устранить недостатки пакетного спуска по градиенту:

- первый шаг заключается в том, чтобы взять часть всей выборки, а не всю ее целиком;
- второй способ – менять шаг по мере приближения к минимуму.

Стохастический градиентный спуск – еще один тип градиентного спуска, в основе которого лежат принципы взятия не всей выборки, а некоторых случайно выбранных из нее объектов. Во-первых,

если расчеты привели к локальному минимуму, алгоритм легко выйдет из него. Во-вторых, можно легко добавить новый объект, и не нужно будет заново все пересчитывать, ведь есть вероятность, что он попадет в расчете как случайный объект.

Существует еще один вид градиентного спуска, так называемый мини-пакетный градиентный спуск, который использует ту же идею, что и стохастический, но берет не один случайный объект, а небольшую группу объектов из выборочных дан-

ных. Это обеспечивает баланс между эффективностью пакетного градиентного спуска и надежностью стохастического градиентного спуска.

Сходимость пакетного, стохастического и мини-пакетного градиентного спуска показана на рисунках 1, 2, 3 соответственно. Из изображений интеграции можно сделать вывод, что выбор типа градиентного спуска зависит от скорости и точности, которую мы хотим увидеть [6].

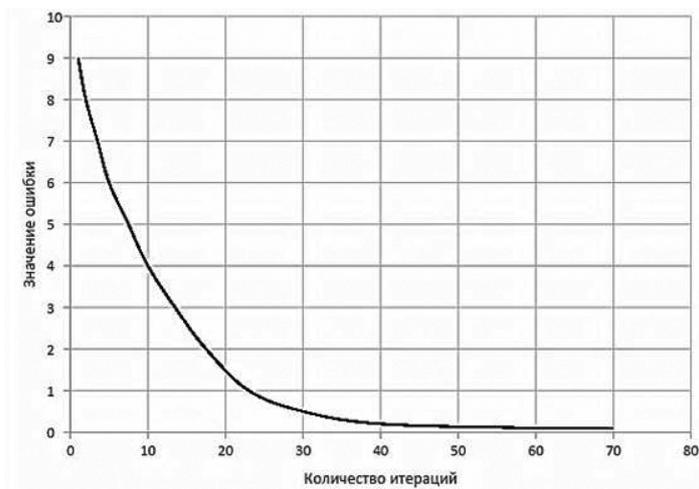


Рисунок 1. Сходимость пакетного градиентного спуска

Источник: разработано авторами

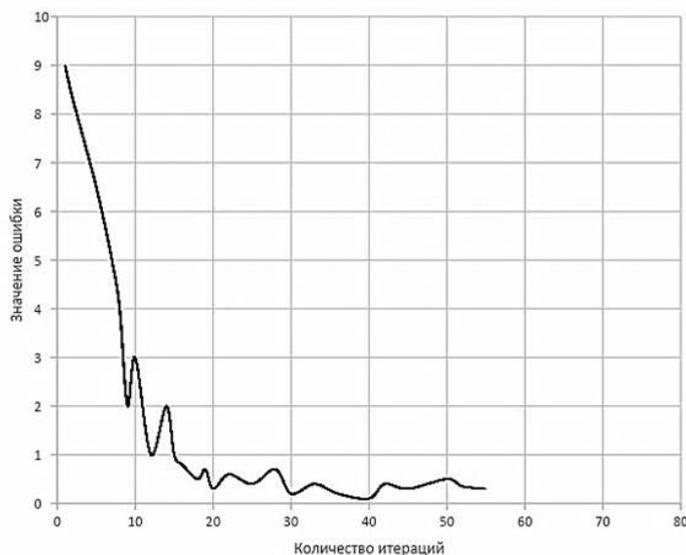


Рисунок 2. Сходимость стохастического градиентного спуска

Источник: разработано авторами

В процессе исследования нами разработана программа на языке программирования Python в интегрированной среде PyCharm, которая с помощью метода градиентного спуска подбирает оптимальные значения параметров a и b функции $f = ax + b$,

при аппроксимации экспериментальных данных y_i . Данные, поступающие на вход программы, можно наглядно увидеть на рисунке 4.

Критерием качества выполнения алгоритма выступает минимум суммы квадратов отклонений. То

есть мы берем расстояние от экспериментальных данных y_i до теоретической прямой, все возводим в квадрат и суммируем.

И аппроксимация будет тем лучше, чем меньше значение E . То есть E – это такой критерий качества, который должен быть минимален [1].

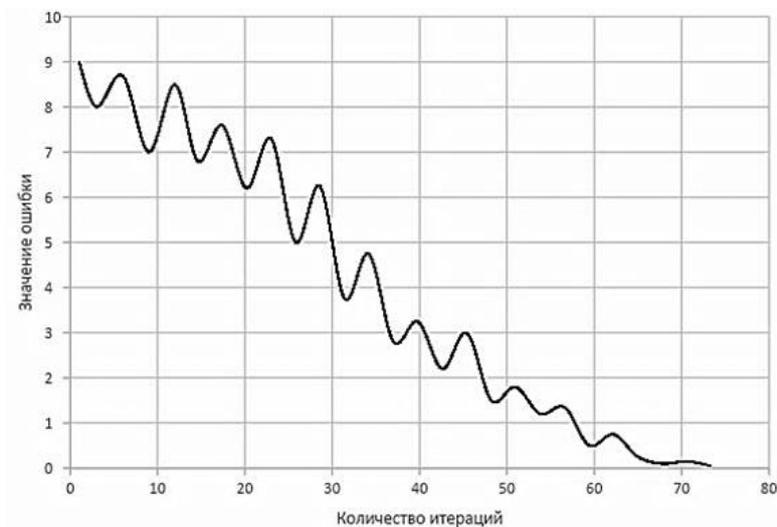


Рисунок 3. Сходимость мини-пакетного градиентного спуска
Источник: разработано авторами

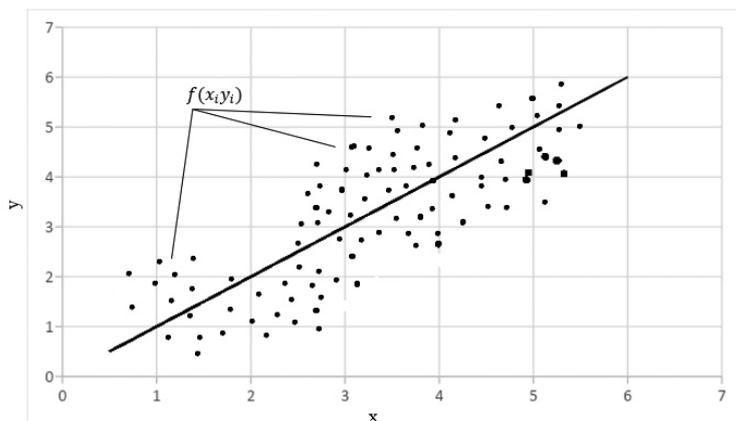


Рисунок 4. Теоретическая прямая и экспериментальные данные
Источник: разработано авторами

$$E(a, b) = \sum_{i=1}^N (y_i - f(x_i))^2 = \sum_{i=1}^N (y_i - ax_i - b)^2. \quad (2)$$

А для того, чтобы найти параметры, от которых зависит этот критерий качества, мы должны продифференцировать этот критерий качества по этим

двум параметрам a и b . В итоге получаем следующую систему частных производных функции:

$$\begin{cases} \frac{\partial E(a, b)}{\partial b} = -2 \sum_{i=1}^N (y_i - ax_i - b) \\ \frac{\partial E(a, b)}{\partial a} = -2 \sum_{i=1}^N (y_i - ax_i - b)x_i \end{cases} \quad (3)$$

Далее на основе этих производных получаем алгоритм подбора параметров a и b :

$$\begin{cases} a_{n+1} = a_n - \lambda_n^1 \frac{\partial E(a,b)}{\partial a} \\ b_{n+1} = b_n - \lambda_n^2 \frac{\partial E(a,b)}{\partial b} \end{cases} \quad (4)$$

В роли шагов здесь выступают независимые друг от друга λ_n^1 для a и λ_n^2 для b . Как мы можем посмотреть на рисунке 5, экспериментально подобранная

прямая и теоретическая прямая совпадают, значит алгоритм справился со своей задачей. Следовательно, метод градиентного спуска сработал отлично.

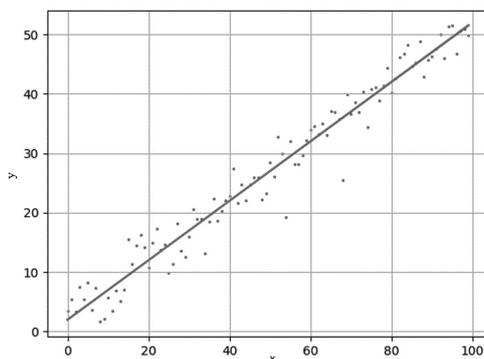


Рисунок 5. Результат работы алгоритма
Источник: разработано авторами

Стоит заметить, что мы разбили задачу для подбора параметров фактически на независимый поиск по каждому параметру, их связка сохранялась только в частных производных, но сам алгоритм сработал по каждому параметру независимо. В этом

и есть его преимущество, он практически линейно увеличивает вычислительную сложность при увеличении размерности пространства поиска, именно поэтому градиентный спуск получил такую широкую популярность в машинном обучении.

Литература

1. Поляк Б. Т. О некоторых способах ускорения сходимости итерационных методов // Ж. вычисл. матем. и матем. физ. – 1964. – Т. 4 – № 5. – С. 791–803.
2. Траск Э. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352 с. – URL: https://www.rulit.me/data/programs/resources/pdf/Trask_Grokaem-glubokoe-obuchenie_RuLit_Me_603639.pdf?ysclid=174f5f4i6650167401 (дата обращения: 08.05.2022).
3. Cherkassky V., Ma Y. (2009) Another look at statistical learning theory and regularization. *Neural Networks*. Vol. 22, No. 7, pp. 958–969. <https://doi.org/10.1016/j.neunet.2009.04.005> (In Eng.).
4. Kingma D. P., Ba J. L. (2015) Adam: A Method for Stochastic Optimization. Available at: https://arxiv.org/help/arxiv_identifier (accessed: 10.05.2022). (In Eng.).
5. Lemaréchal C. (2012). Cauchy and the gradient method. *Documenta Mathematica*. Extra Vol. , pp. 251–254. (In Eng.).
6. Vapnik V. N. (1999) An Overview of Statistical Learning Theory. *IEEE Transactions on neural networks*. Vol. 10, No. 5, pp. 988–999. <https://doi.org/10.1109/72.788640> (In Eng.).
7. Xuhui S., Cherkassky V., Li W. (2000) Measuring the VC-Dimension Using Optimized Experimental Design. *Neural Computation*. Vol. 12, No. 8. pp. 1969–1986. <https://doi.org/10.1162/089976600300015222> (In Eng.).

Статья поступила в редакцию: 12.05.2022; принята в печать: 25.10.2022.
Авторы прочитали и одобрили окончательный вариант рукописи.