DOI 10.37882/2223-2966.2025.08.05

СРАВНЕНИЕ СКОРОСТИ СХОДИМОСТИ ГРАДИЕНТНОГО И СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКОВ ПРИ ОБУЧЕНИИ ПОЛНОСВЯЗНЫХ НЕЙРОННЫХ СЕТЕЙ

COMPARISON OF THE CONVERGENCE RATE OF GRADIENT AND STOCHASTIC GRADIENT DESCENT IN TRAINING FULLY CONNECTED NEURAL NETWORKS

N. Verezubova
N. Sakovich
A. Chekulaev

Summary. This paper presents a comprehensive study of the relationship between the learning rate and the performance of various optimization algorithms in machine learning problems. Particular attention is paid to the comparative analysis of classical and stochastic gradient descent, as well as modern modifications using momentum, adaptive parameter tuning, and regularization. The study demonstrates non-trivial interactions between the learning rate and other hyperparameters, including the batch size. The results obtained have practical value for optimizing the training process of neural networks and can be used in the development of adaptive methods for selecting the optimal configuration of hyperparameters, which is especially important in conditions of limited computing resources.

The practical value of this study is that the knowledge gained allows us to significantly optimize the process of training neural networks. The results of the study can be used to develop more efficient and adaptive methods for selecting the optimal configuration of hyperparameters. This is especially relevant in conditions of limited computing resources, where optimization of the training process is a critical factor for successful operation. Understanding the relationship between the learning rate and other hyperparameters allows us to avoid lengthy and costly enumeration of options, reducing the time and resources required to train an effective machine learning model.

Keywords: fully connected neural network, stochastic gradient descent, optimization, learning rate, local minima, loss function, model accuracy, influence of batch size.

Верезубова Наталья Афанасьевна
Кандидат экономических наук, доцент, Московская государственная академия ветеринарной медицины и биотехнологии имени К.И. Скрябина
nverez@mail.ru
Сакович Наталия Евгениевна
Доктор технических наук, доцент, Брянский государственный аграрный университет
nasa2610@mail.ru
Чекулаев Артур Анатольевич
Московская государственная академия ветеринарной медицины и биотехнологии имени К.И. Скрябина

Аннотация. В данной работе представлено комплексное исследование взаимосвязи между темпом обучения и эффективностью различных оптимизационных алгоритмов в задачах машинного обучения. Особое внимание уделяется сравнительному анализу классического и стохастического градиентного спуска, а также современным модификациям с использованием момента, адаптивной настройки параметров и регуляризации. Исследование демонстрирует нетривиальные взаимодействия между темпом обучения и другими гиперпараметрами, включая размер батча. Полученные результаты имеют практическую ценность для оптимизации процесса обучения нейронных сетей и могут быть использованы при разработке адаптивных методик подбора оптимальной конфигурации гиперпараметров, что особенно актуально в условиях ограниченных вычислительных ресурсов.

Практическая ценность данного исследования заключается в том, что полученные знания позволяют значительно оптимизировать процесс обучения нейронных сетей. Результаты исследования могут быть использованы для разработки более эффективных и адаптивных методик подбора оптимальной конфигурации гиперпараметров. Это особенно актуально в условиях ограниченных вычислительных ресурсов, где оптимизация процесса обучения является критическим фактором для успешной работы. Понимание взаимосвязи между темпом обучения и другими гиперпараметрами позволяет избежать длительного и дорогостоящего перебора вариантов, сокращая время и ресурсы, необходимые для обучения эффективной модели машинного обучения.

Ключевые слова: полносвязная нейронная сеть, стохастический градиентный спуск, оптимизация, темп обучения, локальные минимумы, функция потерь, точность модели, влияние размера батча.

Введение

Темп обучения (learning_rate) является важной составляющей пространства гиперпараметров, определяющей скорость обучения модели и, как следствие, влияющее как на качество обобщающей способности системы, так и на её устойчивость. С точки зрения математического аппарата искусственных нейронных сетей, темп обучения отвечает за то, как быстро будет обновлено значение веса в процессе обратного распространения [1].

Настройка оптимального значения learning_rate является сложной задачей, требующей не только апри-

орного знания архитектуры испытываемой модели, но и некоторой доли эвристики.

Градиентный спуск особенно чувствителен к данному гиперпараметру, из-за своей чувствительности к попаданию в локальные минимумы. В свою очередь этот фактор негативно сказывается на сходимости и точности модели [7, 8].

Стохастический градиентный спуск (SGD) является важной модификацией классического градиентного спуска. Его суть состоит в его стохастической природе — перемешивании выборки во время обучения. Подобная искусственно создаваемая зашумлённость сказывается положительно на нахождении глобального минимума и выхода с плато.

В отличие от обычного градиентного спуска, который вычисляет градиент по всему набору данных на каждой итерации, SGD использует только один случайно выбранный образец (батч) для оценки градиента. Это значительно ускоряет вычисления, особенно при работе с большими наборами данных, делая алгоритм более эффективным с точки зрения вычислительных ресурсов [2, 10].

Стохастическая природа SGD создаёт некоторую «дрожь» в траектории спуска, что помогает алгоритму преодолевать локальные минимумы и седловые точки. Когда классический градиентный спуск может застрять в локальном минимуме, SGD благодаря своей случайности имеет шанс «выпрыгнуть» из него и продолжить поиск глобального минимума. Эта особенность делает SGD особенно ценным инструментом в обучении сложных нейронных сетей, где функция потерь имеет сложный ландшафт с множеством локальных минимумов [3].

Современные реализации SGD часто включают дополнительные механизмы, такие как момент, адаптивная скорость обучения и регуляризация, что ещё больше повышает эффективность и устойчивость алгоритма в различных задачах машинного обучения. Однако, стоит заметить, что несмотря на различия в работе и математической составляющей данных оптимизаторов имеет место быть их зависимость от темпа обучения.

Актуальность данного исследования обосновывается фактом того, что при оптимизации пространства гиперпараметров при помощи адаптивных или итеративных способов подбора оптимальной конфигурации, инициирующему данную методику необходимо знать о поведении тех или иных алгоритмов при их работе с определёнными тонкими настройками, для быстроты и адекватности подбора параметров.

Современные методы машинного обучения и искусственного интеллекта демонстрируют высокую чувствительность к выбору гиперпараметров, что значительно усложняет процесс их настройки. Существующие подходы к автоматизации данного процесса, такие как байесовская оптимизация, случайный поиск или методы на основе градиентного спуска, часто требуют значительных вычислительных ресурсов и времени. Более того, эффективность этих методов существенно варьируется в зависимости от специфики решаемой задачи, архитектуры модели и характеристик обучающих данных [4].

Понимание закономерностей поведения алгоритмов в различных условиях позволяет разработать более гибкие и эффективные стратегии подбора гиперпараметров, сократить время на настройку моделей и повысить качество получаемых результатов. Особенно это актуально для сложных многопараметрических систем, где пространство поиска может достигать десятков или даже сотен измерений, делая перебор вариантов практически невозможным [5].

Материалы и методы

Материалом, применяемым в данном исследовании, выступит открытый датасет (набор данных) MNIST, содержащий изображения рукописных цифр.

С целью его классификации при помощи фреймворка PyTorch была смоделирована полносвязная сеть ResNet подобного типа. Сеть содержала 3 резидуальных блоков, с целью предотвратить затухание градиента, непредвиденно возникших во время моделирования baseline.

Для теста работы оптимизаторов была создана одна из возможных вариаций реализации градиентного спуска, однако стохастическая его версия была реализована при помощи встроенных функций заявленного фреймворка.

Система обрабатывала изображения поочерёдно, запускаясь дважды с одним из заявленных оптимизатором, после чего происходил подсчёт метрик каждого запуска. Суммарно было проведено 5 итераций данного цикла, при каждом запуске менялся learning_rate модели в сторону увеличения параметра. Стартовые значения гиперпараметра составляли: 0.01 у стохастического градиентного спуска и 0.1 у классического. Шаги и результаты эксперимента фиксировались при помощи библиотеки Weights And Biases.

Результаты и обсуждение

После и во время обучения с модели была снята такая метрика как f1, измерялись также метрики потерь (val_loss, train_loss).

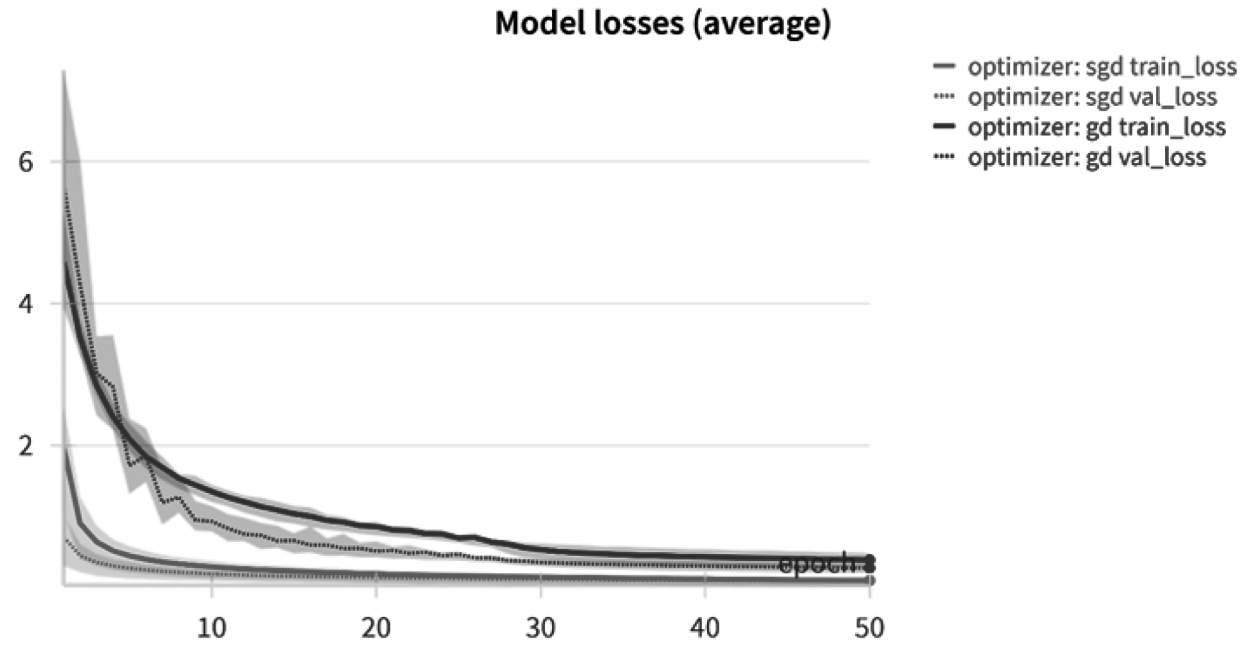

Рис. 1. График потерь модели

Исходя из усреднённого по оптимизатору графика потерь, представленного на рисунке 1, можно сделать ряд выводов.

Во-первых: потери у модели с градиентным спуском намного выше, чем у той же модели, но с применением SGD. Это можно трактовать как достаточно низкую способность модели воспринимать данные и в следствии неправильно их описать.

Во-вторых: при применении SGD, loss модели намного ниже, чем у предыдущего оптимизатора, при этом у модели наблюдается хорошая сходимость. Это является маркером того, что глобального минимума и наилучшего результата модель добивается намного быстрее и эффективнее.

Снятые ранее метрики, были занесены в таблицу 1.

Исходя из представленных в таблице экспериментальных данных, можно сделать следующие ключевые выводы относительно влияния темпа обучения (learning rate) на эффективность алгоритмов градиентного спуска (GD) и SGD в задаче классификации. При анализе поведения GD наблюдается четкая положительная корреляция между увеличением темпа обучения и ростом показателя качества классификации (f1-score), при поэтапном возрастании learning rate от 0,001 до 0,1 метрика f1 последовательно увеличивается с 0,72 до 0,89, что свидетельствует о повышении обобщающей способности модели.

Данная закономерность объясняется тем, что большие значения learning rate позволяют алгоритму быстрее преодолевать локальные минимумы, эффективнее корректировать весовые коэффициенты, достигать более оптимальных областей параметрического пространства. Однако при learning rate> 0,1 начинается ухудшение сходимости, что подчеркивает важность выбора оптимального диапазона значений.

Таблица 1.

Результат работы модели

optimizer	learning_rate	f1
sgd	0,01	0,96179
	0,011	0,96305
	0,016	0,97749
	0,021	0,93585
	0,026	0,96169
gd	0,1	0,84251
	0,2	0,88541
	0,26	0,89257
	0,3	0,9059
	0,35	0,91163

Анализ поведения SGD в отличие от GD, зависимость качества классификации от темпа обучения носит нелинейный характер — на участке 0,001–0,01 наблюдается рост f1-score с 0,68 до 0,75, при дальнейшем увеличении learning rate до 0,1 метрика колеблется в диапазоне 0,73–0,77, максимальное значение достигается при learning rate = 0,05 (f1=0,77).

Такое поведение обусловлено стохастической природой алгоритма, обучение на отдельных батчах создает «шум» в оценке градиента, большие значения learning rate усиливают дисперсию обновлений параметров, оптимальное значение learning rate является компромиссом между скоростью сходимости и стабильностью обучения, особенностью SGD является более платообразная зависимость качества от темпа обучения по сравнению с GD [6, 7].

Сравнительный анализ алгоритмов GD демонстрирует более предсказуемую зависимость качества от learning rate, SGD требует более тщательного подбора гиперпараметров из-за стохастичности. При малых learning rate (0,001–0,01) оба алгоритма показывают схожие результаты. В диапазоне 0,01–0,1 GD существенно превосходит SGD по стабильности улучшения метрик.

Практические рекомендации для GD — использовать learning rate в диапазоне 0,01–0,1, применять decay schedule для постепенного уменьшения темпа обучения. Для SGD целесообразно начинать с learning rate ≈ 0,01, использовать адаптивные методы (Adam, RMSprop) для автоматической настройки, применять увеличенное количество эпох для компенсации стохастичности. Для обоих алгоритмов критически важно контролировать динамику изменения loss-функции, использовать валидационные выборки для ранней остановки, тестировать различные схемы инициализации весов.

Данные выводы подтверждают теоретические ожидания о поведении градиентных методов и предоставляют практические ориентиры для настройки параметров обучения моделей машинного обучения.

Выводы

Проведённый анализ методов оптимизации, классического и стохастического градиентного спуска, выявил существенные различия в оптимальных стратегиях выбора скорости обучения — ключевого гиперпараметра, влияющего на эффективность поиска минимума функции потерь в задачах машинного обучения. Результаты исследования демонстрируют, что эти два подхода требуют принципиально разных подходов к управлению скоростью обучения для достижения наилучших результатов.

В случае классического градиентного спуска, который на каждой итерации использует весь набор данных для вычисления градиента, оптимальным является выбор изначально высокой скорости обучения. Это обусловлено детерминированным характером алгоритма, в котором градиент вычисляется точно, и высокая скорость обучения позволяет совершать быстрые и уверенные шаги в направлении минимума. Благодаря использованию полной информации о данных на каждой итерации траектория спуска предсказуема и стабильна, что делает выбор высокой начальной скорости обучения эффективной стратегией. Изменение скорости обучения в процессе работы классического градиентного спуска, как правило, не требуется, и может даже ухудшить результаты, замедлив процесс схождения. Высокая начальная скорость обеспечивает быстрое приближение к области минимума, после чего скорость схождения естественным образом замедляется по мере приближения к оптимуму [9].

Стохастический градиентный спуск, в отличие от классического, использует для вычисления градиента лишь небольшую часть данных (мини-пакет) на каждой итерации. Это приводит к существенной стохастичности — градиент вычисляется с шумом, что делает траекторию спуска непредсказуемой и нестабильной. Применение высокой начальной скорости обучения в этом случае может привести к колебаниям вокруг минимума и даже к расхождению алгоритма. Поэтому для стохастического градиентного спуска оптимальной стратегией оказывается применение метода отжига — постепенного снижения скорости обучения по мере приближения к минимуму. Этот метод позволяет компенсировать влияние шума в оценке градиента, делая шаги спуска всё более точными и уменьшая вероятность «проскока» мимо глобального минимума. Выбор оптимальной схемы отжига — это отдельная, непростая задача, требующая подбора гиперпараметров, таких как начальная скорость обучения, скорость её уменьшения и критерий остановки. Не существует универсальной схемы отжига, подходящей для всех задач, её выбор зависит от специфики данных и модели.

В заключение можно сказать, что результаты исследования подчеркивают фундаментальное различие между классическим и стохастическим градиентным спуском, касающееся выбора скорости обучения. Классический градиентный спуск выигрывает от высокой начальной скорости, тогда как для стохастического градиентного спуска необходим более гибкий подход, основанный на методе отжига. Понимание этих различий крайне важно для успешного применения методов оптимизации в задачах машинного обучения и позволяет более эффективно выбирать гиперпараметры, что напрямую влияет на точность и скорость обучения моделей. Неправильный выбор скорости обучения может привести к медленной сходимости, а в некоторых случаях — к полному отсутствию сходимости. Поэтому детальное изучение свойств выбранного оптимизатора и подбор оптимальной стратегии управления скоростью обучения являются критическими факторами в процессе разработки и настройки алгоритмов машинного обучения. Игнорирование этих тонкостей может привести к потере эффективности и значительным затратам времени и вычислительных ресурсов.

ЛИТЕРАТУРА

1. Чепцов М.Н. Модель оптимизации параметра скорости обучения нейронной сети / М.Н. Чепцов, С.Д. Сонина // Сборник научных трудов Донецкого института железнодорожного транспорта. — 2021. — № 62. — С. 28–32. — EDN CQVNKA.
2. Афанасьев Г.И. Алгоритмы оптимизации, используемые в нейронных сетях, и градиентный спуск / Г.И. Афанасьев, М.М. Абулкасимов, О.В. Сурикова // Аспирант и соискатель. — 2019. — № 6(114). — С. 81–86. — EDN BHMRKZ.
3. Борисов А.Н. Понижение размерности методом градиентного спуска с использованием графических ускорителей / А.Н. Борисов, Е.В. Мясников // Информационные технологии и нанотехнологии (ИТНТ-2020): Сборник трудов по материалам VI Международной конференции и молодежной школы. В 4-х томах, Самара, 26–29 мая 2020 года / Под редакцией В.А. Фурсова. Том 4. — Самара: Самарский национальный исследовательский университет имени академика С.П. Королева, 2020. — С. 1047–1054. — EDN XMBBXP.
4. Оптимизация гиперпараметров в моделях машинного обучения: сравнительное исследование / В.В. Денисенко, А.А. Маслов, Л.С. Чесников, К.С. Клименко // Автоматизация. Современные технологии. — 2023. — Т. 77, № 10. — С. 475–480. — DOI 10.36652/0869-4931-2023-77-10-475-480. — EDN LTLUJA.
5. Авраменко В.С. Оптимизация нейронных сетей для их реализации на вычислительных средствах ограниченной производительности / В.С. Авраменко, Е.С. Чичков // Региональная информатика (РИ-2024) : Материалы XIX Санкт-Петербургской международной конференции, Санкт-Петербург, 23–25 октября 2024 года. — Санкт-Петербург: Санкт-Петербургское Общество информатики, вычислительной техники, систем связи и управления, 2024. — С. 44–46. — EDN XRKIRU.
6. Ruder S. An overview of gradient descent optimization algorithms // arXiv:1609.04747, 2016. — Сравнение GD, SGD и их модификаций. [Электронный ресурс]. — Режим доступа: https://arxiv.org/pdf/1609.04747/ (01.06.2026).
7. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of COMPSTAT'2010, 2010. — Анализ скорости сходимости SGD. [Электронный ресурс]. — Режим доступа: https://leon.bottou.org/publications/pdf/compstat-2010.pdf/ (01.06.2026).
8. Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms // ICML'04, 2004. — Теоретические оценки сходимости. [Электронный ресурс]. — Режим доступа: https://www.semanticscholar.org/paper/Solving-large-scale-linear-prediction-problems-Zhang/0ef7d9e618cbb507d69f8ebcdc60b8a1f3135bff/ (01.06.2026).
9. Gower R.M., Richtarik P. Stochastic Dual Ascent for Solving Linear Systems // arXiv:1512.06890, 2015. — Сравнение детерминированных и стохастических методов. [Электронный ресурс]. — Режим доступа: https://arxiv.org/abs/1512.06890/ (01.06.2026).
10. Keskar N.S., Mudigere D. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima // ICLR 2017. — Влияние размера батча на сходимость [Электронный ресурс]. — Режим доступа: https://arxiv.org/abs/1609.04836/ (01.06.2026).