

# Влияние гиперпараметров оптимизатора Lion на точность предсказания в классических датасетах

Баранов А.А., СПбГУ, Санкт-Петербург [st107591@spbu.ru](mailto:st107591@spbu.ru),  
Михайлов Д.А., СПб ФИЦ РАН, Санкт-Петербург [dam@dsccs.pro](mailto:dam@dsccs.pro)

## Аннотация

Оптимизатор Lion, разработанный в 2023 году, обладает уникальными механизмами обновления параметров, отличающимися от традиционных оптимизаторов SGD, Adam, что обуславливает актуальность его подробного изучения. В данной работе проводится исследование влияния гиперпараметров оптимизатора Lion на обучение нейронных сетей. Эксперимент ставится с использованием сеток гиперпараметров на датасетах Titanic, Mushroom, Spambase и модели двухслойной нейронной сети с ReLU и Dropout. Результаты эксперимента показали существенное влияние ряда гиперпараметров на итоговую модель. Полученные данные из исследования могут быть полезны при практическом применении оптимизатора Lion для глубокого обучения нейронных сетей.

## Введение

Спрос на нейронные сети с каждым годом только растёт [1], и вместе с ним возрастают требования к их скорости обучения и эффективности [2] [5]. В свою очередь, алгоритмы оптимизации нейронных сетей позволяют достичь более высоких результатов в качестве ускорения обучения и повышения качества итоговой модели [3] [6].

Одним из таких алгоритмов оптимизации является Lion – оптимизатор, разработанный в 2023 году командой Google Brain с использованием автоматизированного поиска алгоритмов. Ключевое отличие Lion от более традиционных алгоритмов оптимизации, таких как Adam и SGD, заключается в использовании знака текущего градиента и момента в качестве обновления параметров, что позволяет достичь меньших затрат по памяти [4].

В рамках данной работы проводится исследование влияния гиперпараметров оптимизатора Lion на точность предсказания, а также сравнение Lion с популярным алгоритмом оптимизации AdamW, являющимся модификацией алгоритма Adam.

## Сравнение Lion с AdamW

AdamW и Lion – это два современных оптимизатора, которые представляют разные подходы к обновлению параметров нейросети.

Оптимизатор AdamW является адаптивным методом обучения. Его ключевая идея лежит в вычислении первого момента  $m_t$  и второго момента  $v_t$  с целью последующего обновления направления и величины параметров алгоритма. В частности, вычисление параметра модели  $\theta$  выглядит следующим образом:

$$\theta_t = \theta_{t-1} - \eta_t \left( \frac{m_t}{\sqrt{v_t} + \epsilon} + \lambda \theta_{t-1} \right),$$

где  $\theta_{t-1}$  – параметр модели на предыдущем шаге;  $\theta_t$  – параметр модели на текущем шаге;  $\eta_t$  – скорость обучения;  $m_t$  и  $v_t$  — соответственно первый и второй моменты градиента;  $\epsilon$  – небольшое число для избежания деления на ноль;  $\lambda$  – скорость затухания весов.

В отличие от AdamW подход Lion является более эффективным и менее трудоёмким. Он заключается в отслеживании направления обновления. Его ключевое различие кроется в том, что он заменяет масштабированные значения градиента на их знак, следующим образом:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} f(\theta_t), \quad (1)$$

$$\theta_{t+1} = \theta_t - \eta_t (\text{sign}(m_t) - \lambda \theta_{t-1}), \quad (2)$$

где  $\beta_1$  – коэффициент экспоненциального сглаживания градиента;  $\text{sign}(m_t)$  — направление обновления (по знаку);  $f$  – функция потерь.

Характеристика	Lion	AdamW
Адаптивность	Нет(фиксированный шаг)	Да(используется второй момент)
Сглаживание	$\beta_1$ , только знак	$\beta_1, \beta_2$
Затраты по памяти	Хранение $\beta_1$	Хранение $\beta_1, \beta_2$
Вычислительные нагрузки	Знак	Корень, деление

Таблица 1: Сравнение AdamW и Lion

Таким образом, алгоритм оптимизации Lion представляет собой эффективную альтернативу популярным алгоритмам оптимизации, таким как

AdamW и другие. Особый интерес вызывают его гиперпараметры: коэффициент экспоненциального сглаживания градиента( $\beta_1$ ), скорость обучения( $\eta$ ), скорость затухания весов( $\lambda$ ).

## Экспериментальное исследование влияния гиперпараметров оптимизатора Lion на точность

В рамках работы был проведён эксперимент с целью оценки точности фиксированной модели, использующей оптимизатор Lion как с параметрами по умолчанию, так и с гиперпараметрами, подобранными с помощью перебора по сетке гиперпараметров.

В эксперименте была реализована нейросетевая модель на основе библиотеки PyTorch. Архитектура модели представляет собой простую полносвязную нейросеть, включающую следующие компоненты: входной слой; функция активации ReLU; слой Dropout; выходной слой.

Гиперпараметры модели были заданы заранее и оставались фиксированными на протяжении всех экспериментов. Единственными варьируемыми величинами были гиперпараметры оптимизатора Lion:  $\beta_1$  – коэффициент экспоненциального сглаживания градиента;  $\eta$  – скорость обучения;  $\lambda$  – скорость затухания весов.

В исследовании использовались датасеты: Titanic, Spambase, Mushroom.

Titanic датасет состоит из 891 записи и 12 признаков. Ключевая цель модели, использующей данный датасет, определить, выжил ли конкретный человек или нет. Основные признаки датасета: пол человека; класс билета – значение данного поля один из вариантов: первый класс, второй класс, третий класс; возраст; стоимость билета; порт отправления – значение данного поля один из вариантов: Шербур, Квинстаун, Саутгемптон.

Spambase датасет состоит из 4601 записи и 57 признаков. Ключевая цель модели, использующей данный датасет, определить, является ли письмо спамом или нет. Основные признаки датасета делятся на несколько групп: частоты слов — частота возникновения наиболее характерных для спама слов в письме; частоты символов — содержание специальных символов в тексте; статистика по длине слов — различные длины последовательностей заглавных букв.

Mushroom датасет состоит из 8124 записей и 22 признаков. Ключевая цель модели, использующей данный датасет, определить, является ли гриб ядовитым. Все признаки датасета являются категориальными, они состоят из списка значений. Основные признаки: запах; цвет спор; цвет пластинок;

размер пластинок; образование синяков; тип ножки; цвет шляпки; среда обитания.

Экспериментальное исследование проводилось с использованием двух сеток гиперпараметров: одна – большего размера и предназначена для малого датасета, чтобы время обучения соответствовало обучению на больших датасетах с меньшими по размеру сетками.

Первая сетка использовалась для датасета Titanic, она состояла из 20 значений по каждому гиперпараметру оптимизатора Lion, что в сумме составляет 8000 комбинаций. Её значения варьировались следующим образом:

- коэффициент экспоненциального сглаживания градиента  $\beta_1$ : от 0.8 до 0.98;
- скорость обучения  $\eta_t$ : от  $5 \cdot 10^{-5}$  до  $10^{-2}$ ;
- скорость затухания весов  $\lambda$ : от 0 до 0.1.

Вторая сетка использовалась для датасетов Mushroom, Spambase, она состояла из 10 значений по каждому гиперпараметру оптимизатора Lion, что в сумме составляет 1000 комбинаций. Её значения варьировались следующим образом:

- коэффициент экспоненциального сглаживания градиента  $\beta_1$ : от 0.8 до 0.99;
- скорость обучения  $\eta_t$ : от  $10^{-5}$  до  $10^{-2}$ ;
- скорость затухания весов  $\lambda$ : от 0 до 0.1.

В рамках экспериментального исследования использовалась конфигурация оптимизатора Lion с гиперпараметрами, установленными по умолчанию:

- коэффициент экспоненциального сглаживания градиента  $\beta_1$ : 0.9;
- скорость обучения  $\eta_t$ :  $10^{-4}$ ;
- скорость затухания весов  $\lambda$ : 0.

Также в качестве метрики качества была использована метрика Accuracy (точности) предсказания ключевой переменной в каждом из датасетов. Она подсчитывается с помощью следующей формулы:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

где  $TP$  — истинно положительные: модель правильно предсказала положительный класс,  $TN$  — истинно отрицательные: модель правильно предсказала отрицательный класс,  $FP$  — ложно положительные: модель ошибочно предсказала положительный класс,  $FN$  — ложно отрицательные: модель ошибочно предсказала отрицательный класс.

Эксперимент показал, что для средних и больших датасетов результаты, полученные с использованием гиперпараметров по умолчанию, отличаются от наилучшей комбинации, найденной в процессе поиска, не более чем на 0.5 процента. Так точность модели для датасета Spambase оказалась следующей:

- Lion с гиперпараметрами по умолчанию: 0.93268;
- Lion с лучшей комбинацией из сетки гиперпараметров: 0.93702;
- прирост точности: 0.00434.

Точность модели для датасета Mushroom, в отличие от датасета Spambase, снизилась, что, вероятно, обусловлено ограниченным размером используемой сетки гиперпараметров:

- Lion с гиперпараметрами по умолчанию: 0.98708;
- Lion с лучшей комбинацией из сетки гиперпараметров: 0.98154;
- прирост точности:  $-0.00554$ .

Однако в ходе эксперимента с использованием датасета Titanic, результаты улучшились более чем на 22.5 процента. Так точность модели на данном датасете оказалась следующей:

- Lion с гиперпараметрами по умолчанию: 0.67832;
- Lion с лучшей комбинацией из сетки гиперпараметров: 0.83217;
- прирост точности: 0.15385.

## Заключение

Основной результат работы заключается в комплексном экспериментальном исследовании влияния ключевых гиперпараметров оптимизатора Lion на качество обучения нейронных сетей на прикладных задачах, что позволяет расширить понимание его практического потенциала в свете недавней публикации алгоритма.

Улучшение результатов в первую очередь обусловлено увеличением размера сетки гиперпараметров, и как следствие, увеличением количества возможных комбинаций гиперпараметров. На основании результатов эксперимента можно утверждать, что гиперпараметры оптимизатора Lion существенно влияют на точность модели, особенно в условиях ограниченного объёма данных. Это показывает важность настройки гиперпараметров оптимизатора Lion при работе с малыми датасетами.

### Список литературы

- [1] Stefan H., Conrad S., Sarvnaz K., Alexandra B., Claire N.: Artificial intelligence adoption in the physical sciences, natural sciences, life sciences, social sciences and the arts and humanities: A bibliometric analysis of research publications from 1960-2021 // *Technology in Society*. – 2023. – Т. 74. – С. 102260.
- [2] Alen J., Dubravko F., Juraj H., Hrvoje B.: Short-Term Photovoltaic Power Plant Output Forecasting Using Sky Images and Deep Learning // *Energies*. – 2023. – Т. 16. – No. 14. – Pp. 5428.
- [3] Pamadi V. N., Rastogi D. Optimizing Neural Network Performance Through Adaptive Learning Algorithms // *International Journal of Research in all Subjects in Multi Languages [Subject: Computer Science]*. 2025. Vol. 13. – Pp. 2321-2853.
- [4] Xiangning C., Chen L., Da H. Symbolic Discovery of Optimization Algorithms. // 37th Conference on Neural Information Processing Systems (NeurIPS 2023). – 2023. – Vol. 36. – Pp. 49205-49233.
- [5] Alexander D. G., Chris L., Matthew T.J.: Can Learning Optimization Make Reinforcement Learning Less Difficult? // 38th Conference on Neural Information Processing Systems (NeurIPS 2024). – 2024. – Vol. 37. – Pp. 5454-5497.
- [6] Carlo A., Silvia S., Jakob N. F., Patrick R0., Yee W. T.: Meta-Learning Objectives for Preference Optimization // *arXiv preprint arXiv:2411.06568*. - 2024.