УДК 519.95

В.І. Норкін, А.Ю. Козирєв, Б.В. Норкін

СУЧАСНІ СТОХАСТИЧНІ КВАЗІГРАДІЄНТНІ АЛГОРИТМИ ОПТИМІЗАЦІЇ*

Норкін Володимир Іванович

Інститут кібернетики імені В.М. Глушкова НАН України,м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», orcid: 0000-0003-3255-0405

vladimir.norkin@gmail.com

Козирєв Антон Юрійович

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
orcid: 0009-0007-6692-2162

a.kozyriev@kpi.ua

Норкін Богдан Володимирович

Інститут кібернетики імені В.М. Глушкова НАН України, м. Київ,

bogdan.norkin@gmail.com

Стохастична оптимізація стала провідним методом у різних галузях, таких як машинне навчання, нейронні мережі та обробка сигналів. Ці задачі спрямовані на мінімізацію цільової функції із зашумленими та невизначеними даними. Всебічно порівнюються сучасні квазіградієнтні методи стохастичної оптимізації, ілюструються їхні основні принципи, властивості збіжності та практичні застосування. Вводяться основні поняття градієнтного спуску, стохастичної апроксимації та оптимізації, після чого детально пояснюються методи оптимізації. Поглиблено аналізуються адаптивні стохастичні градієнтні методи, акцентується увага на їхній здатності динамічно змінювати швидкість навчання залежно від структури задачі. Досліджуються узагальнення цих методів на негладкі випадки, описуються проблеми, що виникають при негладких оптимізаційних ландшафтах. Ілюструється застосування вдосконалених методів у контексті задач безумовної оптимізації та демонструється їхня ефективність у прискоренні збіжності та підвищенні точності. Цей порівняльний аналіз має на меті дати дослідникам і практикам глибше розуміння останніх досягнень у стохастичній оптимізації та окреслити шлях для майбутніх інновацій.

Ключові слова: стохастична оптимізація, стохастична апроксимація, стохастична негладка оптимізація, стохастичний градієнтний спуск, стохастичний квазіградієнтний метод, адаптивний метод стохастичного градієнта, кінцево-різницевий стохастичний градієнт.

* Робота авторів підтримана Національним фондом досліджень України (проєкт 2020.02/0121).

Вступ

В останні роки зростаюча складність проблем, керованих даними, і збільшення масштабів наборів даних підкреслили важливість ефективних методів оптимізації в різних галузях, таких як машинне навчання, комп'ютерний зір (CV — Computer Vision) і обробка природної мови (NLP — Natural Language Processing). Методи стохастичної оптимізації, які спрямовані на мінімізацію цільових функцій в умовах невизначеності та шуму, привертають значну увагу як метод вирішення цих задач. Основна ідея таких методів полягає у використанні градієнтної інформації з випадково вибраної підмножини даних, що зменшує обчислювальну складність при збереженні загального напрямку до оптимуму. У цій статті проведемо порівняльний аналіз сучасних квазіградієнтних методів стохастичної оптимізації, розглянемо їхні теоретичні основи, властивості збіжності та практичні застосування. Введемо фундаментальні поняття градієнта, градієнтного спуску та стохастичної оптимізації, перш ніж досліджувати тонкощі класичних та адаптивних стохастичних градієнтних методів.

Крім того, досліджуються узагальнення цих методів на негладкі випадки та їхню застосовність у нейронних мережах. Оптимізація відіграє вирішальну роль у навчанні моделей та їхній продуктивності в цій галузі. Запропоновані у статті всебічний огляд і порівняння сучасних методів стохастичної оптимізації мають на меті заохочувати дослідників і практиків до більш усвідомленого вибору, що в підсумку сприятиме подальшому прогресу.

Історично еволюція методів оптимізації характеризується появою та розвитком багатьох стохастичних квазіградієнтних (SQG — Stochastic Quasi-Gradient) методів. Першими стохастичними методами оптимізації вважаються методи апроксимації Роббінса–Монро [1] та Кіфера–Вольфовіца [2] у 1950-х роках. Після цих новаторських робіт в 1960-х роках з'явився алгоритм стохастичного квазіградієнтного спуску Ю.М. Єрмол'єва [3] — метод управління невизначеностями, притаманними негладким опуклим стохастичним задачам.

У 1960-ті роки Б.Т. Поляк запропонував «метод важкої кулі» (Momentum — метод імпульсу) [4, 5], який включав множник швидкості, що породжувало більш плавну траєкторію збліження та надавало вагову складову попереднім градієнтам. Не менш важливим внеском цієї епохи був прискорений метод Нестерова (NAG — Nesterov's Average Gradient Method) [6] — особливо впливовий метод, який запровадив механізм попередньої оцінки градієнта і в такий спосіб покращив збіжність і зменшив осциляції траєкторії збіжності. Крім того, в цьому десятилітті також з'явився метод дзеркального спуску Немировського–Юдіна [7] — метод оптимізації з усередненням траєкторії спуску, що забезпечує більшу гнучкість на всьому оптимізаційному ландшафті.

З 2010-х років спільнота оптимізаторів переорієнтувалася на адаптивні методи, що демонструвало тенденцію до коригування швидкості навчання на основі історичної інформації про градієнт. AdaGrad (Adaptive Gradient) [8] став лідером у цій галузі завдяки унікальному калібруванню швидкості навчання щодо кожного параметра та демонструванню своєї ефективності, особливо на розріджених наборах даних [9]. RMSProp (Running Mean Square Propagation), що наслідував його, застосував ковзну середню попередніх квадратичних градієнтів для стабілізації змін швидкості навчання [10]. Алгоритм ADAM (Adaptive Momentum), що синтезував принципи «метода важкої кулі» і RMSProp, зберіг оцінки першого моменту градієнта, а також його незміщений другий момент і у такий спосіб позиціонував себе як передовий алгоритм у величезному просторі методів глибокого навчання [11].

Останні тенденції ілюструють зростання кількості оптимізаційних задач, що характеризуються негладкими цільовими функціями, наприклад нейронні мережі

 ISSN 2786-6491

з розривними функціями активації [12–14]. Класичні алгоритми градієнтного спуску, які детально описані вище, призначені виключно для розв'язання задач з гладкими цільовими функціями. Отже, можуть виникати проблеми в контекстах з негладкими цільовими функціями через розриви градієнта, які є основним джерелом навігаційної інформації. Поширеним рішенням проблеми є застосування методу згладжування Стеклова [15–18], тобто апроксимація значення градієнта в певному околі поточної точки використання згладженої цільової функції в градієнтній рекурентній послідовності. Інше рішення передбачає апроксимацію градієнта за допомогою відповідних скінченних різниць [16].

1. Задача стохастичної оптимізації

Математичне формулювання, яке має на меті знайти оптимальний розв'язок цільової функції в умовах невизначеності та шуму, притаманних базовим даним або системі, — це задача стохастичної оптимізації. На відміну від детермінованої оптимізації, де цільова функція і обмеження точно визначені, стохастична оптимізація має справу з проблемами, де деякі компоненти мають випадковий характер, що робить оптимізаційний ландшафт більш складним.

Нехай $F : W \to \mathbb{R}^1$ — цільова функція з областю визначення $W \subset \mathbb{R}^n$, $f : W \times \Xi \to \mathbb{R}^1$ — опукла та диференційована функція, яка залежить від детермінованої змінної $w \in W$ та стохастичної змінної $\xi \in \Xi$, визначеної на просторі (Ξ, Σ, P), $\mathbb{R}^n$ — стандартний n-вимірний евклідів простір. Тоді задачу стохастичної оптимізації можна подати у вигляді [3, 19]

$$\min_{w \in W} \left[F(w) = \mathbb{E}\, f(w, \xi) = \int_{\xi \in \Xi} f(w, \xi) P(d\xi) \right], \tag{1}$$

де $\mathbb{E}$ — оператор математичного сподівання. Основною проблемою задачі (1) є неможливість явно обчислити значення інтеграла (математичне сподівання) та значення градієнта інтеграла. Рішення полягає у використанні алгоритмів стохастичного градієнтного спуску, які використовують градієнти $\nabla_w f(w, \xi)$ стохастичної функції $f(\cdot, \xi)$ або їхні кінцево-різницеві заміни на кожній ітерації.

1.1. Стохастична апроксимація. Перші кроки в розробці стохастичних алгоритмів зроблено в роботі Роббінса та Монро [1], де автори розглянули задачу пошуку кореня функції $F : \mathbb{R} = W \to \mathbb{R}$ з відповідним шумовим доданком за припущенням, що розподіл його має нульове математичне сподівання. Припустимо, що функцію $F(w)$ не можна безпосередньо спостерігати, тоді маємо змогу наблизити її за допомогою вимірювання випадкової величини $f(w^k, \xi^k)$ у точках w^k і такої, що для умовного стохастичного сподівання виконано $\mathbb{E}_{\xi^k}[\, f(w^k, \xi^k) \mid w^k \,] = F(w)$. Відповідно, запропонований алгоритм являє собою наступну рекурентну послідовність:

$$w^{k+1} = w^k - \rho_k f(w^k, \xi^k), \quad k = 0, 1, \ldots, \tag{2}$$

де ρ_k — послідовність додатних покрокових множників. Кіфер та Вольфовіц [2] запропонували розширення алгоритму (2) на пошук екстремуму функції $F : \mathbb{R} \to \mathbb{R}$ за припущенням, що є випадкові спостереження $g(w^k, \xi^k)$ градієнта $\nabla_w F(w)$ функції вартості $F(w)$ в точці w^k на ітерації k. Як $g(w^k, \xi^k)$

автори пропонують брати кінцево-різницеві оцінки градієнтів функції $f(\cdot, \xi^k)$ в точці w^k. Запропонований алгоритм являє собою кінцево-різницеву апроксимацію за збуреними значеннями параметрів $w^k + \delta_k$ та $w^k - \delta_k$ відповідно:

$$w^{k+1} = w^k - \rho_k \frac{f(w^k + \delta_k, \xi_k) - f(w^k - \delta_k, \xi_k)}{2\delta_k}, \quad k = 0, 1, \ldots . \tag{3}$$

1.2. Застосування стохастичних квазіградієнтних алгоритмів у машинному навчанні. Проблеми стохастичної оптимізації виникають у різних сферах, таких як фінанси, управління ланцюгами поставок, машинне навчання та дослідження операцій, де рішення повинні прийматися на основі неповної або невизначеної інформації [3, 19].

Штучна нейронна мережа — математична модель, що є деяким представленням нейронних мереж живих тварин та діє схожим чином. Ці моделі складаються з нейронів та синаптичних зв'язків між ними. Нейрони поділяють на окремі групи, які називають шарами. Загальний принцип роботи полягає у тому, що вхідні дані поступово передаються між нейронами, які утворюють між собою синапс. Синаптичні зв'язки мають деякий числовий параметр, що зветься «вага» та відповідає за силу зв'язку між нейронами або, інакше кажучи, відображає міру впливу інформації з попереднього нейрона на наступний.

На цей час абсолютна більшість нейронних мереж належить до класу feed-forward neural network (дослівно: пряма нейронна мережа). Задачею такої мережі є знаходження деякої функції $\Phi(x)$, яка для певної вибірки пар $\{x; y\}$ наближатиме функцію $y = \Phi(x)$ для усіх пар вибірки. Для пошуку такої апроксимації задається також цільова функція похибки $f(y', y)$, де y' — згенерований нейромережею результат, а y — очікуваний результат, а також наступна модель нейронної мережі. Нехай $w = (w_1, ..., w_j, ..., w_n)$ — ваги нейронної мережі, тоді вихідний результат мережі залежить від вхідних даних x і ваг w моделі: $y' = \varphi(w, x)$.

Сформулюємо задачу навчання нейронної мережі наступним чином [12]:

$$F(w) = \mathbb{E}_{x,y} f(\varphi(w, x), y) \to \min_{w \in \mathbb{R}^n},$$

де математичне сподівання береться за випадковими парами $\xi = (x, y)$.

У прямих нейронних мережах для цього використовується чисельний метод стохастичного градієнтного спуску. Загальна формула алгоритму є доволі простою:

$$w^{k+1} = w^k - \rho \cdot \nabla_w f(\varphi(w^k, x^k), y^k), \quad w^0 \in \mathbb{R}^n, \quad k = 0, 1, \ldots,$$

де $F(w)$ — цільова функція для мінімізації, ρ — параметр швидкості градієнтного спуску, $\xi^k = (x^k, y^k)$ — приклад тренувальної вибірки, що використовується на ітерації k. Ініціалізація параметрів w^0 моделі відбувається випадковим чином, тому їх треба адаптувати для мінімізації цільової функції.

У разі штучних нейронних мереж розрахунок градієнта відбувається для кожної компоненти вектора параметрів w. Для розрахунку значень похідної активно використовуються правила диференціювання складної функції (backpropagation, автоматичне диференціювання).

Однією з головних проблем цього методу вважається його нестабільність та потреба у налаштуванні параметра ρ.

 ISSN 2786-6491

2. Стохастичні градієнтні методи

У сценаріях, де можемо обчислити значення функції $F(w)$ та її відповідні градієнти $\nabla F(w)$, задача оптимізації (1) розв'язується за допомогою детермінованих методів нелінійного програмування. Найпростіший градієнтний алгоритм сформулюємо наступним чином [3]:

$$w^{k+1} = \Pi_W(w^k - \rho_k \nabla F(w^k)), \Pi_W(v) = \arg\min_{w \in W} \|-w\|, w^0 \in W, k \in \mathbb{N}, \quad (4)$$

де k позначає номер ітерації для даного методу, $\nabla F(w^k)$ — градієнт цільової функції $F(w^k)$ у точці $w = w^k$, а Π_W є ортогональним оператором проєктування на компактну опуклу множину W.

2.1. Стохастичний градієнтний спуск.

Визначення [3]. Випадковий вектор u^k називається стохастичним градієнтом функції $F(w)$ у точці $w = w^k$, якщо виконується умова $\mathbb{E}\{u^k \mid w^k\} = \nabla F(w^k)$, де $\mathbb{E}\{u^k \mid w^k\}$ позначає умовне математичне сподівання.

Отже, якщо $\nabla F(w) = \mathbb{E}[\nabla_w f(w, \xi)]$, то вектор $u^k = \nabla_w f(w^k, \xi^k)$, градієнт вздовж змінної w для функції $f(\cdot, \xi^k)$, що визнає значення параметра $\xi = \xi^k$ сталим, дійсно є стохастичним градієнтом функції $F(w)$ в точці $w = w^k$.

Нехай ρ_k позначає невід'ємні множники кроку для градієнта, а $\{\xi^k\}$ — незалежні спостереження (або статистику) випадкової величини ξ. Відповідний метод стохастичного градієнтного спуску (SGD — Stochastic Gradient Decent) тоді виражається наступним рекурентним рівнянням [3]:

$$w^{k+1} = \Pi_X(w^k - \rho_k u^k), w^0 \in X, k \in \mathbb{N}. \quad (5)$$

Для забезпечення збіжності методу, заданого рівнянням (5), потрібно, щоб множники кроку $\rho_k \geq 0$ задовольняли умови, окреслені нижче:

$$\sum_{k=0}^{\infty} \rho_k = +\infty, \sum_{k=0}^{\infty} \rho_k^2 < +\infty. \quad (6)$$

Ці умови свідчать про те, що в той час, як сума множників кроків повинна прямувати до нескінченності, сума їхніх квадратів має залишатися скінченною. Зазначені вимоги є важливими для підтримки балансу між достатнім прогресом та уникненням занадто великих кроків, які можуть порушити збіжність алгоритму.

У пакетному варіанті градієнтного спуску (5), дослідженому в роботах [12, 20], використовується наступний варіант градієнта $u^k = \frac{1}{N_k} \sum_{i=1}^{N_k} \nabla_w f(w^k, \xi_i^k)$. Тут $\{\xi_i^k, i = 1, ..., N_k\}$ репрезентує незалежні спостереження випадкової величини ξ на ітерації k. Однак варто зазначити, що з погляду споживання пам'яті пакетна версія потрапляє до категорії «жадібних алгоритмів», оскільки вона потребує завантаження до пам'яті всього набору вхідних ознак. І навпаки, SGD завантажує лише один елемент з набору ознак, наприклад один запис з великого масиву даних. Це робить SGD більш придатним для вирішення таких завдань, як глибоке навчання нейронних мереж.

У мініпакетній версії [9, 21] стохастичний градієнт обчислюється з випадково вибраної підмножини стохастичних змінних $M(\xi_i^k) \subset \{\xi_i^k\}_{N_k}$ з фіксованим розміром $\forall i : \left| M(\xi_i^k) \right| = m.$ Цей варіант має вигляд $u^k = \frac{1}{m} \sum_{i=1}^{m} \nabla_w f(w^k, \xi_i^k)$, $\{\xi_i^k,\ i = 1, ..., m\}$, $m < N_k$, і часто розглядається як «золота середина» між стабільністю збіжності, яку забезпечує пакетний градієнтний спуск, та оптимальним використанням пам'яті стохастичного градієнтного спуску.

2.2. Прискорені методи. Базовий мініпакетний градієнтний спуск має певні труднощі, пов'язані зі збіжністю. Основна проблема полягає у визначенні оптимальної швидкості навчання, оскільки занадто мала швидкість призводить до повільної збіжності, тоді як велика може спричинити нестабільність і потенційно призвести до розбіжності. Програмні стратегії швидкості навчання пропонують певний рівень контролю, але вони за своєю суттю обмежені в адаптивності через власні фіксовані параметри, які можуть не повністю відповідати специфічним особливостям визначеного набору даних. Зазначимо, що застосування єдиної швидкості навчання для всіх оновлень параметрів може бути неефективним для розріджених даних або ознак, що змінюються з різною частотою. Більше того, мінімізація неопуклих функцій помилок у нейронних мережах може бути ускладнена через ризик потрапляння в пастку локальних мінімумів, які є неоптимальними, або сідлових точок [22] через близькі до нуля градієнти.

Для вирішення цих проблем можна застосувати нормалізацію стохастичних градієнтів [12] у рекурентному рівнянні (5):

$$w^{k+1} = \Pi_X (w^k - \rho_k H^k u^k),\ w^0 \in X,\ k \in \mathbb{N}, \tag{7}$$

де H^k — квадратна діагональна матриця з невід'ємними коефіцієнтами на діагоналі, які визначаються поточною ітерацією w^k або всією траєкторією $\{w^0, w^1, ..., w^k\}$. Як приклад, елементами діагоналі можуть бути $h_{ii}^k = \frac{1}{(\| u^k \| + \varepsilon)}$, де $\varepsilon > 0.$

Метод дзеркального спуску Немировського–Юдіна [7] ще більше покращує процес збіжності шляхом додавання усереднення до ітераційної послідовності (5), як показано нижче:

$$\overline{w}^{k+1} = \sum_{i=1}^{k+1} \rho_i w^i \Big/ \sum_{i=1}^{k+1} \rho_i = (1 - \sigma_{k+1}) \overline{w}^k + \sigma_{k+1} w^{k+1},$$

$$\sigma_{k+1} = \rho_{k+1} \Big/ \sum_{i=0}^{k+1} \rho_i,\ k \in \mathbb{N}.$$

Б.Т. Поляк [4, 5] ввів множник прискорення γ разом з множником швидкості збліження ρ в «метод важкої кулі» (також відомий як градієнтний спуск по імпульсу), який відповідає величині градієнта спуску u^k. Це дозволяє керувати прискоренням швидкості збліження за величиною градієнта. Для опису поведінки градієнтного спуску автор використав фізичну аналогію руху тіла в потенційному полі під дією сили тертя, що описується ньютонівським диференціальним рівнянням другого порядку:

$$a\frac{\partial^2 w}{\partial t^2}+v\frac{\partial w}{\partial t}=-\nabla_w F(w),$$

де a — множник маси, v — множник тертя, а градієнт $\nabla_w F(w)$ — консервативне силове поле з потенціальною енергією. Можемо використати кінцево-різницеві оцінки градієнта для перетворення множників прискорення та швидкості у дискретні значення:

$$a\frac{w^{k+1}-2w^k+w^{k-1}}{\Delta t^2}+v\frac{w^{k+1}-w^k}{\Delta t}=-\nabla_w F(w).$$

Далі впорядковуємо схожі терміни і розташовуємо незалежні параметри з одного боку, а решту параметрів — з іншого:

$$w^{k+1}-w^k=-\frac{\Delta t^2}{a+v\Delta t}\nabla_w F(w^k)+\frac{a}{a+v\Delta t}(w^k-w^{k-1}).$$

З огляду на умову фіксованого значення z для кожного елемента x незалежного набору параметрів можемо спростити наведене вище рівняння через вираження кроку градієнтного спуску як $\rho=\frac{\Delta t^2}{(a+v\Delta t)}$ і множника імпульсу — як $\gamma=\frac{a}{(a+v\Delta t)}$ і отримати рекурентну послідовність «методу важкої кулі» [4, 5, 12, 23, 24] для розв'язання оптимізаційної задачі (1) на просторі $W=\mathbb{R}^n$:

$$w^{k+1}=w^k+\gamma_k(w^k-w^{k-1})-\rho_k\nabla_x F(w^k),\ w^0\in\mathbb{R}^n,\ k\in\mathbb{N}.$$

Множники $\gamma_k>0, \rho_k>0$ можуть залежати від w^k або $\{w^0,...,w^k\}$, а замість визначених градієнтів $\nabla F(w^k)$ дозволено використовувати стохастичні градієнти u^k функції $F(w), w=w^k$. Імпульсний множник може прискорити швидкість збіжності алгоритму градієнтного спуску і зробити його швидшим в областях плато і сідлових точках цільової функції. Однак накопичення імпульсу градієнтного спуску ускладнює контроль швидкості збіжності в околі мінімуму. Алгоритм може «пропускати» мінімуми і вимагати додаткових ітерацій для зменшення множника імпульсу.

Метод Нестерова (або метод яружного кроку) [6, 25] визначається наступною рекурентною послідовністю:

$$w^k=v^k-\rho_k\nabla F(v^k),$$

$$v^{k+1}=w^k+\gamma_k(w^k-w^{k-1}),\ w^0=v^0\in\mathbb{R}^n,\ k\in\mathbb{N},$$

де $\gamma_k>0$ — крок вздовж яру. Замість детермінованих градієнтів $\nabla F(w^k)$ можна використовувати стохастичні градієнти функції $F(w)$. Перше рівняння методу означає спуск з точки v^k до низини w^k яру функції $F(w)$, а друге рівняння — крок вздовж яру з точки w^k у напрямку (w^k-w^{k-1}).

2.3. Адаптивні методи. Для наведених вище методів актуальною є проблема збіжності на розріджених даних (проблема зникаючих градієнтів). Рішення полягає в адаптації кроку градієнтного спуску ρ_k до значень зі статистичної вибірки на певній

ітерації при виконанні більших оновлень параметрів w^k для розріджених значень вхідних параметрів ξ^k і малих оновлень для частих значень вхідних параметрів [12].

AdaGrad [8]. Даний алгоритм адаптує крок навчання кожного з параметрів при виконанні великих оновлень для нерегулярних параметрів та незначних для параметрів, які оновлюються занадто часто. Крок навчання ρ регуляризовано додатковим параметром G^k, який акумулює значення градієнтів на минулих кроках:

$$w_j^{k+1} = w_j^k - \frac{\rho}{\sqrt{G_j^k + \varepsilon}} g_j^k,\ j = 1, ..., n;\ k = 0, 1, ...,$$

де g_j^k — j-та компонента стохастичного градієнта функції F в точці $w^k = (w_1^k, ..., w_j^k, ..., w_n^k)$, G_j^k — комбінація попередньо акумульованих градієнтів $G_j^k = G_j^{k-1} + (g_j^k)^2$, $j = 1, ..., n$. У разі навчання нейронної мережі параметри $w = (w_1, ..., w_n)$ є вагами мережі, k означає крок оптимізації, а не шар мережі.

За цих обставин в околі мінімуму значення градієнтного множника наближається до нуля, а значення кроку — до нескінченності, що призведе до того, що послідовність градієнтного спуску вийде за межі простору визначення функції. Щоб вирішити цю проблему, потрібно додати до градієнтної складової G_j^k у знаменнику згладжуючий доданок $\varepsilon << 10^{-8}$.

Хоча метод і вирішує проблему налаштування величини кроку оптимізації, проте виникає нова — акумуляція градієнтів у знаменнику призводить до поступового наближення кроку до нуля. Метод має кращу збіжність на розрідженій вибірці, але сума квадратів значень градієнта призводить до поступового зменшення значення кроку градієнта.

RMSProp [8, 10]. Серйозним недоліком AdaGrad є експоненційне наближення кроку оптимізації до нуля через неконтрольований ріст значення нормування. З одного боку, це стабілізує алгоритм оптимізації, з іншого — крок оптимізації стає настільки незначним, що модель ніколи не наблизиться до точки оптимуму. Вирішенням недоліку неконтрольованого зменшення кроку є використання стохастичної апроксимації значення компоненти градієнта G_k величиною $\mathbb{E}[G_k]$. В алгоритмі RMSProp (середньоквадратичного адаптивного кроку) ця проблема вирішується за допомогою ковзного середнього:

$$G_j^k = \beta G_j^{k-1} + (1-\beta)(g_j^k)^2,\ \ w_j^{k+1} = w_j^k - \frac{\rho}{\sqrt{G_j^k + \varepsilon}} g_j^k,\ j = 1, ..., n;\ k = 0, 1, ...,$$

де β — параметр усереднення, зазвичай 0,9.

ADAM [11]. Даний алгоритм, на відміну від AdaGrad та RMSProp, адаптує не лише величину кроку оптимізації, але й зберігає напрямок руху минулих оновлень подібно до моменту:

$$m_j^k = \beta_1 m_j^{k-1} + (1-\beta_1) g_j^k,\ \ v_j^k = \beta_2 v_j^{k-1} + (1-\beta_2)(g_j^k)^2,$$

$$w_j^{k+1} = w_j^k - \frac{\rho}{\sqrt{v_j^k + \epsilon}} m_j^k,\ \ j = 1, ..., n;\ k = 0, 1, ...,$$

де β_1, β_2 — параметри, типові значення 0,9 та 0,999 відповідно.

Додано примітку [H1]: нормування? унормування?

ISSN 2786-6491

Важливо зазначити, що значення m^k і v^k можуть бути зміщеними (тобто очікуване значення параметра не дорівнює самому значенню), що призводить до неочікуваної поведінки дисперсії коливань. Виправлені оцінки моментів першого та другого порядку мають вигляд [11]:

$$\bar{m}^k = \frac{m^k}{1-\beta_1}, \bar{v}^k = \frac{v^k}{1-\beta_2}.$$

Величини $\bar{m}^k$ та $\bar{v}^k$ є поточними статистичними оцінками градієнтів $\nabla F(w^k)$ та норм $\left\|\nabla F(w^k)\right\|^2$ для цільової функції задачі (1). Таким чином, отримуємо ітераційну послідовність методу адаптивного оцінювання моментів (ADAM) [11]:

$$w_j^{k+1} = w_j^k - \frac{\rho}{\sqrt{\bar{v}_j^k + \varepsilon}} \bar{m}_j^k,\ j = 1,..., n,\ w^0 \in \mathbb{R}^n,\ k \in \mathrm{N}.$$

2.4. Узагальнення для негладких функцій. У класичних градієнтних алгоритмах обчислення значень градієнта $\nabla_w F(w)$ є обов'язковою вимогою, що створює значні труднощі в контексті негладких функцій. Такі функції, що характеризуються відсутністю диференційованості в певних точках області визначення, призводять до значних труднощів при використанні традиційних градієнтних методів оптимізації. Проявами цих труднощів є низька продуктивність алгоритмів через потенційні проблеми зі збіжністю, непередбачувані напрямки пошуку та загальну нездатність встановити навіть локальний оптимум. Узагальнення прискорених методів (важкої кулі [3], яружного кроку [6], дзеркального спуску [7]) на клас так званих узагальнено диференційованих функцій здійснено у роботі [26].

Надалі будемо вважати, що функція $F(w), w \in W,$ є ліпшицевою, тобто існує додатна константа $L > 0,$ така, що $\left|F(v) - F(w)\right| \le L\left\|v - w\right\| \forall v, w \in W.$

Відомо, що ліпшицеві функції є неперервно диференційованими майже всюди, а в точках, де вона не є такою, визначають субдиференціал [27] $\partial F(w) = \{g = \lim_k \nabla F(w^k), w^k \to w\}.$

У дослідженнях [15, 16, 26, 28] запропоновано метод згладжування негладких функцій, а саме усереднення на гіперкубах (також відомий як згладжування Стеклова [18]), застосований для дослідження та оптимізації негладких ліпшицевих функцій з використанням стохастичних кінцево-різницевих методів. Щоб гарантувати локальну збіжність методу оптимізації до стаціонарних точок задачі, параметр згладжування прямує до нуля та узгоджується з ітераційними кроками методу. У роботах [29–34] знаходження оцінки градієнта ліпшицевої функції $F(w)$ пропонується вирішувати за допомогою згладжування на кулі:

$$F_\varepsilon(w) = (v_1)^{-1} \int_{V_1} F(w + \varepsilon v)\, dv,$$

де $V_1 = \{v \in \mathbb{R}^n : \left\|v\right\|_2 \le 1\}$ — одинична куля в $\mathbb{R}^n$ з об'ємом v_1.

Згладжена функція $F_\varepsilon(w)$ є неперервно диференційованою, її градієнт дорівнює очікуваному значенню субдиференціала по кулі [33]:

$$\nabla F_\varepsilon(w) = \frac{1}{\nu_n} \int_{V_1} \partial F(w + \varepsilon v)\, dv.$$

Альтернативно градієнт $\nabla F_\varepsilon(w)$ неперервної функції $F(w)$ обчислюється поверхневим інтегралом

$$\nabla F_\varepsilon(w) = \frac{n}{s_1}\int_{S_1}\frac{1}{2h}\big(F(w+\varepsilon v)-F(w-\varepsilon v)\big)v\,ds,$$

де $S_1=\{v\in\mathbb{R}^n:\|v\|_2=1\}$ — одинична сфера в $\mathbb{R}^n$ з площею s_1.

Позначимо $\tilde{v}$ випадковий вектор, рівномірно розподілений на одиничній сфері S_1, тоді градієнт $\nabla F_\varepsilon(w)$ може бути репрезентований у вигляді математичного сподівання

$$\nabla F_\varepsilon(x)=\frac{n}{h}\mathbb{E}_{\tilde{v}}[F(w+\varepsilon\tilde{v})-F(w-\varepsilon\tilde{v})]\tilde{v}.$$

При його використанні можемо мінімізувати згладжену функцію $F_\varepsilon(w)$ на множині $W\subseteq\mathbb{R}^n$ (і наближено мінімізувати вихідну функцію $F(w)$) стохастичним методом кінцево-різницевого градієнта:

$$w^{k+1}=\Pi_W(w^k-\rho_k\eta_k),\ \ w^0\in W,$$

$$\eta_t=m^{-1}\sum_{i=1}^m(2\varepsilon_k)^{-1}(F(w^k+\varepsilon_k\tilde{v}_i^k)-F(w^k-\varepsilon_k\tilde{v}_i^k))\,\tilde{v}_i^k,\ \ k=0,1,\ldots,$$

де $\{\tilde{v}_i^k\}_{i=1}^m$ — незалежні випадкові вектори, рівномірно розподілені на одиничній сфері, або

$$\eta_t=m^{-1}\sum_{i=1}^m\partial F(w^k+\varepsilon_k\bar{v}_i^k),$$

де $\{\bar{v}_i^k\}_{i=1}^m$ — незалежні випадкові вектори, рівномірно розподілені на одиничній кулі.

Швидкість збіжності усередненої траєкторії цього методу на опуклих негладких ліпшицевих функціях вивчена в [35].

3. Числові результати

Для порівняння швидкості збіжності методів проведено низку тестів на яружних функціях з різною кількістю кінцевих різниць

$$\eta_k=\sum_{i=1}^m\frac{1}{2\varepsilon}(F(w^k+\varepsilon\tilde{v}_i^k)-F(w^k+\varepsilon\tilde{v}_i^k))\,\tilde{v}_i^k.$$

Визначено, що невелика кількість m кінцевих різниць у стохастичному напрямку може суттєво зменшити кількість ітерацій. Реалізоване програмне забезпечення для тестування швидкості збіжності не використовує алгоритми автоматичного диференціювання для обчислення градієнта. Замість цього застосовували кінцеві різниці для оцінки градієнтів з розміром числової сітки ε, де неузагальнений градієнтний алгоритм має $m=1$ кінцевих різниць. За допомогою центральних кінцевих різниць другого порядку досягли точності $O(\varepsilon^2)$. Узагальнена форма градієнтного спуску здебільшого покращує швидкість збіжності неадаптивних методів, оскільки вони мають гірші навігаційні властивості через фіксований множник кроку ρ.

Порівняння швидкості збіжності алгоритмів на гладкій цільовій функції $f_1(w_1,w_2)=\log(1+w_1^2)^2+10w_2^2$ продемонстровано в табл. 1. Задані початкова точка $w^0=(0,1)$ та гіперпараметри $\rho=0,001$, $\rho_{adagrad}=0,1$, $\varepsilon=0,00001$, $\varepsilon_1=0,001$, $\varepsilon_2=0,01$, $\gamma=0,9$, $\beta_1=0,9$, $\beta_2=0,999$.

ISSN 2786-6491

Таблиця 1

m	SGD	Polyak	Nesterov	AdaGrad	RMSProp	ADAM
1	32331	3169	3175	1062	1055	3745
3	10528	1664	1435	450	1085	2076
5	17589	2151	2261	560	1073	2595
7	21349	2148	2389	610	1069	2726

В табл. 1 показано число ітерацій методів для досягнення точності по градієнту $\varepsilon_1 = 0,001$.

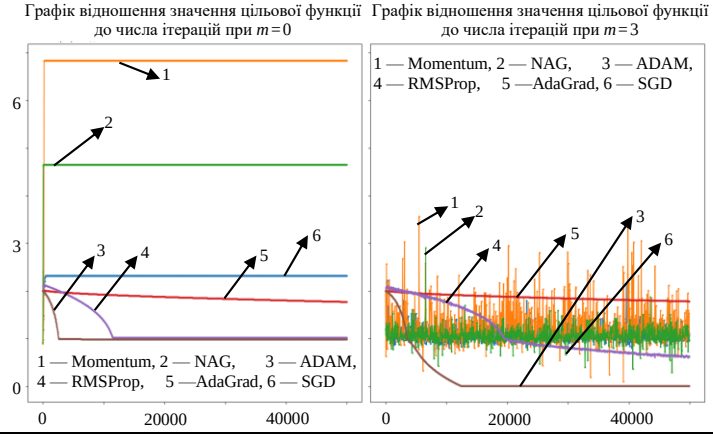
В табл. 2 репрезентується порівняння поведінки алгоритмів на негладкій неопуклій цільовій функції $f_2(w_1, w_2) = |1 - w_1| + 100 | w_2 - w_1^2 |$. Задані початкова точка $w^0 = (-1, 1)$ та гіперпараметри: кількість ітерацій = 50000, крок спуску $\rho = 0,001$, параметр згладжування $\varepsilon = 0,001$, $\gamma = 0,9$, $\beta_1 = 0,9$, $\beta_2 = 0,999$.

Таблиця 2

m	SGD	Polyak	Nesterov	AdaGrad	RMSProp	ADAM
0	2,321078	6,836364	4,648051	1,770273	1,021553	0,980610
3	1,127154	1,30318	1,017222	1,783618	0,597922	0,006344

В табл. 2 представлено значення цільової функції після 50000 ітерацій, значення $m = 0$ відповідає використанню детермінованого градієнта в методах, а $m = 3$ — трьох випадкових кінцевих різниць для оцінки градієнтів.

Хід ітерацій методів, залежність значень цільової функції $f_2(w_1, w_2)$ від числа ітерацій методу показано рисунку (ліворуч — $m = 0$, праворуч — $m = 3$).



Висновок

У статті репрезентовано комплексний огляд сучасних стохастичних градієнтних алгоритмів з їхнім узагальненням на негладкі цільові функції. Для узагальненого кінцево-різницевого алгоритму градієнтного спуску отримано оцінку швидкості збіжності. При експериментальному порівнянні швидкості збіжності на яружних функціях адаптивні градієнтні алгоритми в узагальненому вигляді показали кращу продуктивність, ніж їхнє класичне представлення. Проблемою неадаптивних градієнтних алгоритмів є їхній фіксований крок множника, який може бути вузьким місцем при розріджених вхідних даних.

MODERN STOCHASTIC QUASI-GRADIENT OPTIMIZATION ALGORITHMS

Vladimir Norkin

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»,

vladimir.norkin@gmail.com

Anton Kozyriev

National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»,

sonara4mus@gmail.com

Bogdan Norkin

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv,

bogdan.norkin@gmail.com

Stochastic optimization has become a leading method in various fields such as machine learning, neural networks training, and signal processing. These problems are aimed at minimizing the objective function with noisy and uncertain data. Such problems are attributed to stochastic programming. The article comprehensively compares modern quasi-gradient methods of stochastic optimization, illustrates their basic principles, convergence properties, and practical applications. First, basic concepts of gradient descent, stochastic approximation and optimization are introduced, and then optimization methods are explained in detail. Extensions of the basic gradient descent such as Nemirovski's mirror decent, Polyak's heavy ball (momentum) and Nesterov's valley step methods are reviewed. Beside these classical methods, adaptive stochastic gradient methods are analyzed in depth; attention is focused on their ability to dynamically change the learning rate and decent directions depending on the structure of the problem and a course of optimization. The nomenclature of adaptive stochastic gradient methods includes AdaGrad, RMSProp, ADAM. Generalizations of these methods to the case of non-smooth objective function are studied; problems arising in non-smooth optimization landscapes are described. These generalizations exploit the idea of smoothing coming back to Steklov (1907) and consist in approximation of the original objective function by a sequence of close smoothed functions. The latter admit approximation of their gradients in the form of finite differences in random directions. The application of these improved methods in the context of unconditional optimization problems is illustrated and their effectiveness in accelerating convergence and increasing accuracy is demonstrated. In particular, our experiments demonstrate a considerable positive effect of smoothing on the behavior of the methods in case of nonsmooth problems. This benchmarking study aims to provide researchers and practitioners with a deeper understanding of recent advances in stochastic optimization and outline a path for future innovation.

Keywords: stochastic optimization, stochastic approximation, stochastic non-smooth optimization, stochastic gradient descent, stochastic quasi-gradient method, adaptive stochastic gradient method, finite-difference stochastic gradient.

ПОСИЛАННЯ

1. Robbins H., Monro S. A stochastic approximation method. *The Annals of Mathematical Statistics.* 1951. Vol. 22(3). P. 400–407. DOI: https://doi.org/10.1214/aoms/1177729586
2. Kiefer J., Wolfowitz J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics.* 1952. Vol. 23, N 3. P. 462–466.
3. Ермольев Ю.М. Методы стохастического программирования. Москва : Наука, 1976.
4. Polyak B. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics.* 1964. Vol. 4(5). P. 1–17. DOI: https://doi.org/10.1016/0041-5553(64)90137-5
5. Поляк Б.Т. Введение в оптимизацию. Москва : Наука, 1983. 384 с.
6. Nesterov Y.E. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk. Russian Academy of Sciences.* 1983. Vol. 269. P. 543–547.

 ISSN 2786-6491

7. Nemirovskij A.S., Udin D.B., Dawson E.R. Problems of convex stochastic programming. John Wiley & Sons, 1983. P. 182–197.
8. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research.* 2011. Vol. 12. P. 2121–2159.
9. Ruder S. An overview of gradient descent optimization algorithms. 2016. arXiv preprint arXiv:1609.04747.
10. Tieleman T., Hinton G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *Coursera: Neural Networks for Machine Learning.* 2012. Vol. 4, N 2. P. 26–31.
11. Kingma D.P., Ba J. Adam: A method for stochastic optimization. 2014. arXiv preprint arXiv: 1412.6980.
12. Bottou L., Curtis F.E., Nocedal J. Optimization methods for large-scale machine learning. *SIAM Review.* 2018. Vol. 60(2). P. 223–311. DOI: https://doi.org/10.1137/16m1080173
13. Longo M., Opschoor J.A., Disch N., Schwab C., Zech J. De rham compatible deep neural network FEM. *Neural Networks.* 2023. Vol. 165. P. 721–739. DOI: https://doi.org/10.1016/j.neunet.2023.06.008
14. Ustimenko A., Prokhorenkova L. StochasticRank: global optimization of scale-free discrete functions. 2020. CoRR abs/2003.02122. DOI: https://doi.org/10.48550/arXiv.2003.02122
15. Gupal A.M. A method for the minimization of almost-differentiable functions. *Cybernetics*. 1977. Vol. 13(1). P. 115–117.
16. Гупал А.М. Стохастические методы решения негладких экстремальных задач. Киев : Наукова думка, 1979. 149 с.
17. Ermoliev Y.M., Norkin V.I., Wets R.J.B. The minimization of semicontinuous functions: Mollifier subgradients. *SIAM Journal on Control and Optimization*. 1995. Vol. 33(1). P. 149–167. DOI: https://doi.org/10.1137/s0363012992238369
18. Chagas J.Q., Diehl N.M.L., Guidolin P.L. Some properties for the Steklov averages. 2017. 33 p. DOI: https://doi.org/10.48550/arXiv.1707.06368
19. Shapiro A., Dentcheva D., Ruszczy´nski A. Lectures on stochastic programming: modeling and theory. *Society for Industrial Mathematics*. 2009. DOI: https://doi.org/10.1137/1.9780898718751, https://epubs.siam.org/doi/abs/10.1137/1.9780898718751
20. Qian X., Klabjan D. The impact of the mini-batch size on the variance of gradients in stochastic gradient descent. arXiv 2020. arXiv preprint arXiv:2004.13146.
21. Surono S., Thobirin A., Hsm Z.A., Astuti A.Y., Kp B.R., Oktavia M. Optimization of fuzzy system inference model on mini batch gradient descent. *Frontiers in Artificial Intelligence and Applications*. 2022. DOI: https://doi.org/10.3233/faia220387
22. Dauphin Y., Pascanu R., Gulcehre C., Cho K., Ganguli S., Bengio Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. 2014. arXiv preprint arXiv:1406.2572
23. Qian N. On the momentum term in gradient descent learning algorithms. *Neural Networks*. 1999. Vol. 12(1). P. 145–151. DOI: https://doi.org/10.1016/s0893-6080(98)00116-6
24. Liu W., Chen L., Chen Y., Zhang, W. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*. 2020. Vol. 31(8). P. 1754–176. DOI: https://doi.org/10.1109/tpds.2020.2975189
25. Walkington N.J. Nesterov's method for convex optimization. *SIAM Review*. 2023. Vol. 65(2). P. 539–562. DOI: https://doi.org/10.1137/21M1390037
26. Михалевич В.С., Гупал А.М., Норкин В.И. Методы невыпуклой оптимизации. М. : Наука, 1987. 280 с.
27. Clarke F.H. Optimization and nonsmooth analysis. SIAM, 1990. 305 p.
28. Mayne D.Q., Polak E. Nondifferential optimization via adaptive smoothing. *Journal of Optimization Theory and Applications.* 1984. Vol. 43(4). P. 601–613. DOI: https://doi.org/10.1007/bf00935008
29. Norkin V. Two random search algorithms for minimizing non-differentiable functions. In: Ermoliev Y.M., Kovalenko I.N. (eds.) Mathematical Methods of Operations Research and Reliability Theory. Kyiv : Institute of Cybernetics of the NAS of Ukraine, 1978. P. 36–40.
30. Nesterov Y. Smooth minimization of non-smooth functions. *Mathematical Programming*. 2004. Vol. 103(1). P. 127–152. DOI: https://doi.org/10.1007/s10107-004-0552-5
31. Nesterov Y., Spokoiny V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics.* 2015. Vol. 17(2). P. 527–566. DOI: https://doi.org/10.1007/s10208-015-9296-2
32. Duchi J.C., Jordan M.I., Wainwright M.J., Wibisono A. Optimal rates for zero-order convex optimization: the power of two function evaluations. *IEEE Transactions on Information Theory*. 2015. Vol. 61, N 5. P. 2788–2806. DOI: https://doi.org/10.1109/tit.2015.2409256
33. Shamir O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*. 2017. Vol. 18(1). P. 1703–1713.
34. Norkin V. A stochastic smoothing method for nonsmooth global optimization. *Cybernetics and Computer Technologies*. 2020. N 1. P. 5–14. DOI: https://doi.org/10.34229/2707-451x.20.1.1
35. Norkin V., Pichler A., Kozyriev A. Constrained global optimization by smoothing. arXiv Preprint, 2023. arXiv:2308.08422 [math.OC].

Отримано 26.03.2024