

Список використаних джерел

1. Ullman J. D., Widom J. A First Course in Database Systems. Pearson, 2015. 524 p.
2. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall, 2004. 624 p.
3. Литвиненко І. І. Технології програмування для сучасних вебплатформ. Київ: ІнфоПринт, 2021. 256 с.
4. Nardi B. A. A Study of Human Interaction with Technology. MIT Press, 2017. 300 p.
5. Гончарук О. Ю. Інформаційні системи підтримки прийняття рішень. Одеса: ОНУ, 2019. 272 с.
6. Microsoft. Building Cloud-Based Web Applications. Redmond: Microsoft Press, 2020. 384 p.

УДК 519.8:004.03

*Яценко В. В., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій*

ЗАСТОСУВАННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ДЛЯ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ ІЗ ВЕЛИКИМ ОБСЯГОМ ПАРАМЕТРІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Технології обробки даних критично важливі для галузей наукових досліджень та штучного інтелекту. Розвиток нейронних мереж, зокрема глибинного навчання, відкрив принципово нові можливості для розв'язання складних завдань розпізнавання патернів та прогнозування [1]. З кожною новою ітерацією розробки нейронні мережі переходили від простої до більш складної архітектури. Поступово штучний інтелект переходив від розгорнутої архітектури до згорткової, рекурентної, і наразі перебуває на етапі архітектури-трансформера [2]. Зі зміною архітектури експоненційно зростала і кількість їх параметрів. Сучасні моделі налічують сотні мільярдів параметрів, демонструючи стрімку тенденцію до ускладнення архітектур та збільшення обчислювальної потужності. Відтак створення сучасної нейронної мережі висуває особливі вимоги до процесів її навчання. Принципового значення набуває вибір стратегії оптимізації, яка безпосередньо впливає на швидкість збіжності, стабільність навчання та здатність мережі до узагальнення. Різні методи оптимізації демонструють різні підходи до цієї задачі, мають свої переваги та недоліки, які варто враховувати під час вибору методу для конкретного завдання. Фундаментальним елементом такого налаштування є функція втрат, яка кількісно оцінює розбіжність між передбаченнями моделі та реальними даними [3]. Правильно підібрана функція втрат дає змогу не лише оцінити точність моделі, але й забезпечити ефективне налаштування її параметрів під час навчання.

Метою дослідження стане аналіз алгоритмів оптимізації. Під час дослідження необхідно розглянути методи оптимізації, їх особливості, переваги та обмеження, які впливають на навчання великих нейронних мереж, надати ідею комбінованого методу і підсумувати відповідну інформацію.

У статті *An optimization Strategy for Deep Neural Networks Training* представлений стратегічний підхід до оптимізації навчання нейронних мереж із використанням сучасних адаптивних алгоритмів і методів регуляризації [4]. Автори акцентують на комбінованих алгоритмах, які поєднують адаптивне масштабування градієнтів і зменшення складності обчислень через ефективні стратегії розподілу параметрів. Результати досліджень показують, що ці інновації допомагають скоротити час навчання моделей і забезпечити стабільну збіжність.

Класичний градієнтний спуск є одним із найперших і базових підходів. У цьому методі оновлення параметрів моделі виконується на основі градієнта функції втрат щодо кожного параметра. Головна ідея полягає в тому, щоб рухатися в напрямку, протилежному градієнту, з метою зменшення значення функції втрат. Градієнтний спуск на основі всієї вибірки даних гарантує стабільність, але має значну обчислювальну складність, особливо для великих наборів даних. Це робить метод повільним і малопридатним для сучасних нейронних мереж.

Стохастичний градієнтний спуск (SGD) спрощує задачу, використовуючи випадковий зразок із набору даних на кожному етапі оновлення. Це значно прискорює процес навчання, але додає коливань у траєкторії збіжності, через що процес може бути менш стабільним. Міні-батч градієнтний спуск усуває цю проблему, даючи змогу використовувати невеликі групи даних для кожного кроку. Це знижує випадковість, характерну для SGD, і робить метод більш придатним для масштабних задач.

Адаптивні методи оптимізації значно розширили можливості навчання нейронних мереж. Один із найпопулярніших таких методів – Adam (Adaptive Moment Estimation). Він поєднує дві важливі ідеї: момент і адаптивне масштабування градієнтів. У цьому методі обчислюється експоненціальне згладжування середнього значення градієнтів і їх квадратів. Це допомагає Adam бути швидким і ефективним навіть для задач зі складними ландшафтами функції втрат. Проте Adam має тенденцію до нестійкості на пізніх етапах навчання, оскільки адаптивне масштабування може знижувати його здатність знаходити глобальний мінімум.

RMSProp – ще один адаптивний метод, який працює за принципом поділу градієнта на квадрат кореня середнього значення попередніх градієнтів [5]. Це дає змогу адаптивно змінювати швидкість навчання для кожного параметра, залежно від його важливості. RMSProp ефективний для задач із нерівномірно розподіленими градієнтами, але іноді може демонструвати повільнішу збіжність, порівняно з Adam.

Adagrad адаптує швидкість навчання для кожного параметра, зменшуючи її для тих параметрів, які часто оновлюються, і збільшуючи для рідко оновлюваних. Цей підхід добре працює для задач із розрідженими даними, наприклад, у роботі з текстом. Однак основний недолік Adagrad полягає у поступовому зменшенні швидкості навчання до дуже низьких значень, через що модель може перестати навчатися до досягнення мінімуму функції втрат.

Існують також модифікації градієнтного спуску, які додають до базового алгоритму додаткові компоненти для покращення процесу оптимізації. Момен-

тум є технікою, що використовує попередні значення градієнтів для визначення напрямку руху, додаючи інерцію до оновлення параметрів. Це допомагає швидше долати області з плоскими ландшафтами, прискорюючи збіжність. Nesterov Momentum покращує цей підхід, обчислюючи градієнт у точці, зрушений на крок вперед за інерцією. Це дає змогу більш точно передбачати напрямок і знижує ризик надмірних коливань.

AdamW є вдосконаленням Adam, яке додає регуляризацію за допомогою зменшення ваг параметрів. Це дає змогу уникати перенавчання і зберігати стійкість процесу навчання навіть на великих наборах даних. Така модифікація робить AdamW одним із найкращих виборів для задач, пов'язаних із сучасними нейронними мережами.

Хоча кожен із цих методів має свої переваги, жоден із них не є ідеальним для всіх ситуацій. Об'єднання їхніх сильних сторін у новому підході може бути перспективним рішенням. Наприклад, метод, який комбінує адаптивність Adam, інерцію моментуму і регуляризацію AdamW, здатен забезпечити стабільність, швидкість та ефективність збіжності. Це особливо актуально для задач із великими обсягами даних і моделей із великою кількістю параметрів. Подібні комбіновані методи оптимізації можуть стати основою для навчання сучасних нейронних мереж із великими обсягами параметрів, даючи змогу покращити якість моделей і скоротити час їх навчання.

Отже, різноманітність методів оптимізації надає широкий спектр інструментів для навчання нейронних мереж. Кожен метод має свої переваги й обмеження, тому вибір залежить від специфіки задачі. Однак комбіновані підходи, які інтегрують сильні сторони кількох методів, відкривають нові можливості для ефективного навчання сучасних моделей. Впровадження таких алгоритмів може стати важливим кроком у розвитку оптимізації нейронних мереж, сприяючи підвищенню якості моделей і зменшенню витрат часу на їх навчання.

Список використаних джерел

1. What is Deep Learning? URL: <https://cloud.google.com/discover/what-is-deep-learning> (дата звернення: 28.11.2024).
2. What Is a Transformer Model? URL: <https://blogs.nvidia.com/blog/what-is-a-transformer-model/> (дата звернення: 28.11.2024).
3. What's a loss function? URL: <https://www.datarobot.com/blog/introduction-to-loss-functions/> (дата звернення: 28.11.2024).
4. Wu T., Zeng P., Song An C. optimization Strategy for Deep Neural Networks Training. URL: <https://ieeexplore.ieee.org/document/10009665> (дата звернення: 28.11.2024).
5. Understanding RMSprop – faster neural network learning. URL: <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a> (дата звернення: 28.11.2024).