

ція методів до змінних умов і співпраця з географічними інформаційними системами. Загалом оптимізація маршрутизації у транспортних мережах є актуальною і важливою задачею з потенціалом покращити якість життя та зменшити витрати в різних галузях. Розглянуті методи можуть бути використані для вирішення реальних проблем у сферах транспорту і логістики.

Список використаних джерел

1. Гарін, Д. В. (2018). Оптимізація маршрутизації в логістичних системах. Київ: Видавництво «Логістика».
2. Петренко, О. І., Коваленко, В. М. (2019). Застосування методів оптимізації у транспортних системах. Дніпро: Дніпропетровський національний університет ім. О. Гончара.
3. Коцюба, В. П. (2020). Геоінформаційні системи у транспорті. Львів: Видавництво Львівської політехніки.
4. Степаненко, О. В., Мельник, І. М. (2017). Оптимізація маршрутизації в автомобільних транспортних системах. Харків: Видавництво Харківського національного університету ім. Каразіна.
5. Січко, Т. В., Смоктей, К. В., Ткачук, А. О. Прикладні аспекти розрахунку структурно-топологічних характеристик систем. Системи та технології. 2019. № 1(57), 2019. С. 141–153.

УДК 004.89

*Поліщук А. М., здобувач 3 курсу спеціальності 122 Комп'ютерні науки,
Хмелівський Ю. С., асистент кафедри інформаційних технологій*

ОПТИМІЗАЦІЯ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПОКРАЩЕННЯ ЇХНЬОЇ ПРОДУКТИВНОСТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Останніми роками глибокі нейронні мережі (ГНМ) завоювали значну популярність завдяки своїм вражаючим досягненням у багатьох сферах, зокрема комп'ютерне зорове сприйняття та обробка природних мов. Однак разом зі зростанням їх застосування виникають проблеми, пов'язані з високим рівнем складності та великою кількістю параметрів, що можуть призводити до значних обчислювальних витрат. Глибокі нейронні мережі, особливо глибокі згорткові та рекурентні мережі, мають велику кількість шарів, які потрібно оптимізувати під час тренування. Для цього необхідно на великих обсягах даних ефективно навчання моделі, що може призводити до перетренування на невеликих наборо-

рах даних. Також велика кількість параметрів ускладнює роботу з моделями і може вимагати великих обчислювальних ресурсів для їх тренування та використання.

Отже, виникає потреба в пошуку ефективних методів оптимізації для покращення продуктивності глибоких нейронних мереж. Деякі з напрямів досліджень передбачають оптимізацію функцій витрат, використання регуляризації для уникнення перетренування, а також розробку нових архітектур та методів оптимізації, які дають змогу ефективно використовувати обмежені обчислювальні ресурси. Одним із ключових завдань є зменшення кількості параметрів без втрати продуктивності моделі, що може передбачати застосування методів стиснення моделей, які дають змогу зберігати лише найважливіші параметри, або розробку нових методів регуляризації для автоматичного видалення зайвих параметрів. Усі ці напрями досліджень спрямовані на те, щоб зробити глибокі нейронні мережі більш доступними та ефективними у різних застосуваннях, зменшуючи обчислювальні витрати та покращуючи їх здатність до роботи з обмеженими обсягами даних.

Оптимізація глибоких нейронних мереж є актуальною проблемою в сучасній науці та технологіях. Вдосконалення продуктивності ГНМ може відкрити нові можливості в області штучного інтелекту, а також допоможе ефективно використовувати їх у реальних умовах [1].

Останні дослідження у сфері оптимізації ГНМ сфокусовані на різноманітних аспектах, як-от алгоритми оптимізації, регуляризація, архітектурна оптимізація та інші. Низка робіт вказує на ефективність методів оптимізації градієнтів, як-от Adam та RMSProp, водночас як інші дослідження розглядають використання еволюційних алгоритмів для оптимізації архітектур нейронних мереж.

Метою є вдосконалення продуктивності глибоких нейронних мереж шляхом аналізу та оптимізації різних аспектів їх структури та навчання.

Вплив різних методів оптимізації та регуляризації на результати нейронних мереж є критично важливим аспектом у галузі глибокого навчання.

1. Стандартний стохастичний градієнтний спуск (SGD). Це базовий метод оптимізації, який оновлює масу моделі в напрямі, протилежному градієнту функції втрати. Однак цей метод може застрягати в локальних мінімумах та мати повільний збіг.

2. Метод Адама. Цей метод використовує експоненційно зважене середнє градієнта та квадрата градієнта. Він може адаптувати швидкість навчання для кожного параметра окремо, що робить його ефективним для швидкого та стабільного тренування [2].

3. RMSprop. Цей метод також адаптує швидкість навчання для кожного параметра, але використовує експоненційно згладжений середній квадрат градієнта. Це допомагає вирішити проблему швидкого величезного зростання градієнта.

Використання різних методів оптимізації допомагає досягти оптимального балансу між швидкістю збігу та уникненням проблем, як-от розходження чи застрягання в локальних мінімумах.

Методи регуляризації:

1. Dropout. Цей метод регуляризації полягає у випадковому вимиканні деяких нейронів під час тренування. Це допомагає уникнути перенавчання;

2. L1- та L2-регуляризація. Ці методи вводять штраф на величину мас, змушуючи модель використовувати менше параметрів. Це допомагає уникнути перенавчання;

3. Batch Normalization. Цей метод вставляє шар нормалізації між шарами мережі. Він допомагає стабілізувати процес тренування та прискорює збіг моделі.

Експерименти показали, що комбінація різних методів регуляризації може призводити до ще кращих результатів. Важливо враховувати специфіку завдання та розмір набору даних під час вибору підходів.

Вплив комбінації: ефективність нейронної мережі визначається не тільки окремими методами, але й їх комбінацією. Наприклад, використання алгоритму Адама разом із dropout може покращити здатність моделі уникати перенавчання та підвищити її робастність.

Також важливо враховувати взаємодію параметрів, як-от швидкість навчання та розмір пакету для навчання. Наприклад, занадто велика швидкість навчання може призвести до розходження, тоді як занадто мала може призвести до повільного збігу.

Архітектура мережі:

1. Глибина мережі: використання глибоких архітектур може допомогти у вирішенні складних завдань та отриманні високої точності. Проте важливо уникати перенавчання, особливо за обмежених ресурсів для тренування.

2. Кількість нейронів в шарах: оптимальний розмір шарів може залежати від специфіки завдання. Велика кількість нейронів може призвести до перенавчання, тоді як занадто мала – до недонавчання.

3. Функції активації: вибір функцій активації, як-от ReLU, Sigmoid або Tanh, впливає на здатність моделі до навчання та уникнення проблем, як-от зникаючий градієнт.

Параметри навчання:

1. Швидкість навчання: оптимальна швидкість навчання допомагає досягти збіжності моделі. Проте важливо уникати занадто великих значень, які можуть призвести до розходження.

2. Розмір пакету для навчання: великий розмір пакету може прискорити тренування, але під час цього може виникнути проблема втрати різноманітності даних. Малий розмір пакету може призвести до непостійності оновлень ваг.

Функція втрати:

Вибір оптимальної функції втрати: залежно від завдання вибір відповідної функції втрати може бути критичним. Наприклад, для класифікації може використовуватися категоріальна крос-ентропія, а для регресії – середньоквадратична помилка.

Регуляризація:

Використання dropout та L1 / L2 регуляризації: ці методи можуть допомогти уникнути перенавчання та покращити загальну робастність моделі.

Експерименти показали, що оптимальна комбінація параметрів залежить від конкретного завдання та набору даних. Постійне налаштування та вдосконалення цих параметрів під час тренування допомагає досягти кращих результатів та покращити ефективність нейронних мереж.

Використання функцій активації, як-от ReLU або Leaky ReLU, може сприяти швидшому збігу під час тренування. Також треба зазначити, що велике значення має правильний вибір функції втрат для кожного конкретного завдання. Використання відповідної функції втрати може значно поліпшити результати моделі та допомогти уникнути проблем перенавчання.

Загальним висновком до виконаних досліджень є те, що успіх у навчанні нейронних мереж залежить від балансу між різними методами оптимізації та регуляризації, вибором правильних параметрів і архітектурою мережі. Також важливо постійно адаптувати ці параметри під час тренування, особливо під час роботи з великими та складними даними. Ці дослідження роблять важливий внесок у сферу штучного інтелекту, допомагаючи розуміти, як різні методи можуть взаємодіяти та впливати на результати моделей.

Список використаних джерел

1. Гудфелоу, Я., Бенжіо, І., Курвіль, А. Глибоке навчання. MIT Press, (2016). 652 с.
2. Кінгма, Д. Адам: Метод стохастичної оптимізації. Препринт arXiv arXiv. Вип. 1412, п. 6980. (2014). С. 1–15.
3. Срівастава, Н., Гінтон, Г., Крижевський, А., Суцкевер, Я., Салахутдінов, Р. (2014). Dropout: Простий спосіб запобігти перенавчанню нейронних мереж. Journal of Machine Learning Research, 15, С. 1929–1958.