Список використаних джерел
1. Kotlin Coroutines Guide. Kotlin Documentation. URL: https://kotlinlang.org/docs/coroutines-overview.html (Last accessed: 10.11.2024).
2. Ktor: Build Asynchronous Servers and Clients in Kotlin. URL: https://ktor.io/ (Last accessed: 10.11.2024).
3. Newman, S. Building Microservices: Designing Fine-Grained Systems. 1st Edition. Sebastopol, CA, USA : O'Reilly Media, 2015. 278 p.
4. Josh Skeen, David Greenhalgh. Kotlin Programming. Atlanta, GA, USA : Big Nerd Ranch Guides, 2018. 480 p.
5. Pierre-Olivier Laurence, Amanda Hinchman-Dominguez, Mike Dunn, G. Blake Meike. Programming Android with Kotlin: Achieving Structured Concurrency with Coroutines 1st Edition. Sebastopol, CA, USA : O'Reilly Media, 2021. 355 p.

ЛІЩИНСЬКА Л.Б.,
ВНТУ

ОСНОВНІ ПІДХОДИ ДО ПОБУДОВИ СИСТЕМИ АДАПТИВНОГО ТЕСТУВАННЯ ЗНАНЬ

Анотація. Проаналізовано основні підходи для побудови системи адаптивного тестування знань.
Ключові слова: адаптивне тестування, машинне навчання, класифікація.

Традиційне тестування у вигляді стандартизованих тестів фіксованою довжин сьогодні, переростає у нові ефективні форми адаптивного тестування, що базується на відмінних від традиційних теоретико-методологічних підходах [1]. У зв'язку з цим, розробка методів і програмних засобів для систем адаптивного тестування знань є актуальними задачами різноманітних сфер діяльності, які потрібують об'єктивної та ефективної оцінки знань.

В загальному алгоритм адаптивного тестування знань складається з таких кроків:

1. З набору завдань вибирається найбільш підходяще (за певними параметрами) для користувача завдання.
2. Користувач вирішує завдання правильно чи неправильно.
3. Оцінка користувача оновлюється на підставі цієї відповіді.
4. Дані кроки повторюються до тих пір, поки згідно певна умова виконується. Така умова називається критерієм зупинки тестування. Як тільки вона задовольняється тестування вважається завершеним.

Оскільки на початку процесу тестування системі невідомо про рівень знань користувача поки він не відповість щонайменше на перше запитання, тестування починається з середнього рівня складності, вважаючи рівень підготовки користувача як «середній». Наступне завдання системи – якомога швидше пристосуватись (адаптуватись) до рівня користувача, для найефективнішої оцінки його знань.

В загальному випадку для розробки системи адаптивного тестування знань необхідні такі компоненти.

Набір тестових завдань, які відкалібровані за складністю. Банк завдань повинен калібруватись відповідно до певної психометричної моделі.

Точка входу у тест. В основному всі системи адаптивного тестування знань припускають, що кожен користувач, який починає тестування має «середній» рівень знань, але якщо користувач використовує систему повторно, є можливість починати тестування з іншого рівня.

Логіка вибору завдання з тестового набору. Кожна система тестування знань імплементує власну логіку для найбільш точного підбору завдань під рівень знань користувача.

Алгоритм підрахунку результатів. Після кожного вирішеного завдання, не важливо чи успішно, оцінка користувача змінюється в ту, чи іншу сторону. В результаті проходження N завдань система складає результуючу оцінку, яка, за потреби, може бути приведена до будь-якої системи оцінювання [2].

Критерій визначення завершення тестування. Система може пропонувати користувачеві завдання до тих пір, доки вона не зможе адекватно оцінити рівень його знань. Саме момент, коли оцінка стає «адекватною» і є таким критерієм. В кожні системі тестування даний критеріїв може кардинально відрізнятись. В класичному тестуванні, зазвичай, процес тестовая зупиняється коли

досягнуто ліміту дозволених помилок. В адаптивному ж тестуванні це може бути при досягненні певної межі «рівня знань» в обидва боки. В цьому полягає одна з ключових переваг адаптивного тестування, а саме – точність оцінювання знань користувача.

Вдосконалення системи тестування доцільно здійснювати за рахунок проектування та навчання моделі машинного навчання, яка має виконувати роль механізму оцінки користувача. Крім того, складність самих завдань має змінюватись в залежності від відповідей користувачів.

Список використаних джерел

1. Howard W. Computerized adaptive testing: A Primer (2nd Edition). Mahwah, NJ: ELawrence Erlbaum Associates, 2000, 361p.
2. Олійник М. М. Тест як інструмент кількісної діагностики рівня знань в сучасних технологіях навчання. Донецьк: Донецький національний університет, 2001. 83 с.

ЛУЦЕНКО Р. С.,
РОМАНЮК О. В.,
Вінницький національний технічний університет

ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ ГІБРИДНОГО АДАПТИВНОГО СКОРОЧЕННЯ РАНГУ ДЛЯ ОПТИМІЗАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ НА МОБІЛЬНИХ ПРИСТРОЯХ

Анотація: У роботі розглянуто метод гібридного адаптивного скорочення рангу (HARR) як новітній підхід до оптимізації великих мовних моделей (LLM) для мобільних пристроїв. Метод базується на поєднанні технік Low-Rank Adaptation (LoRA) та прунингу, що дозволяє ефективно адаптувати моделі під обмежені ресурси мобільних платформ, таких як обсяг пам'яті та потужність процесора. Впровадження HARR забезпечує зменшення обсягу моделей без втрати точності, сприяє підвищенню енергоефективності та швидкодії додатків на основі великих мовних моделей.

Ключові слова: мовні моделі, скорочення рангу, мобільні платформи, LoRA, прунинг, оптимізація.

Потреба в адаптації великих мовних моделей для мобільних платформ значно зросла останнім часом, що обумовлено широким впровадженням технологій штучного інтелекту в мобільні додатки. Однак мобільні пристрої, такі як смартфони, мають обмежені апаратні ресурси, що створює низку проблем для реалізації повномасштабних моделей. Скорочення розміру моделей є необхідною умовою для забезпечення ефективного виконання та економії енергії на таких пристроях. Метод HARR пропонує інноваційний підхід до зниження рангу матриць параметрів моделей, що дозволяє скоротити обчислювальне навантаження, зберігаючи при цьому високу точність.

Відсутність оптимізованих під мобільні платформи моделей призводить до необхідності розробки нових методів, здатних адаптувати LLM для пристроїв з обмеженими ресурсами. На сьогодні техніки Low-Rank Adaptation та прунингу застосовуються окремо, однак їх інтеграція може забезпечити більш гнучке та ефективне рішення, яке одночасно враховуватиме потреби зменшення розміру та забезпечення точності моделі.

Техніки LoRA та прунингу широко досліджуються у контексті оптимізації глибоких нейронних мереж. LoRA, як метод зниження рангу, дозволяє зменшити кількість параметрів шляхом використання низькорівневих матриць для основних компонентів моделі, що дає змогу значно зекономити пам'ять [1]. Прунинг же зосереджений на видаленні незначущих нейронів та зв'язків, що додатково скорочує модель та знижує її обчислювальну складність [2]. Проте, питання сумісного використання LoRA та прунингу в умовах обмежених ресурсів мобільних платформ досі залишається недостатньо дослідженим. Запропонований метод HARR заповнює цю прогалину, забезпечуючи ефективне поєднання цих технік та врахування апаратних обмежень.

Метод HARR складається з кількох основних етапів. Перший етап передбачає застосування техніки LoRA для створення низькорівневих матриць, що забезпечує початкове скорочення рангу. Далі застосовується адаптивний прунинг, що додатково видаляє елементи моделі, які мають низький вплив на загальну точність. Оскільки мобільні платформи мають індивідуальні обмеження щодо обсягу пам'яті та доступної потужності процесора, HARR включає етап збору інформації про апаратні характеристики пристрою. Ця інформація використовується для адаптивного налаштування