ISSN 2707-1898 (print)
Український журнал інформаційних технологій
Ukrainian Journal of Information Technology
http://science.lpnu.ua/uk/ujit
https://doi.org/10.23939/ujit2020.02.029
Article received 15.10.2020 p.
Article accepted 25.10.2020 p.
UDK 004.738.5:316.[728+772]

Correspondence author
M. I. Zghoba
Mariia.Zghoba@gmail.com

М. І. Згоба, Ю. І. Грицюк
Національний університет "Львівська політехніка", м. Львів, Україна

ТРЕНУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ ПОПИТУ НА ПАСАЖИРСЬКІ ПЕРЕВЕЗЕННЯ ТАКСІ ЗА ДОПОМОГОЮ ГРАФІЧНИХ ПРОЦЕСОРІВ

Розглянуто особливості тренування нейронної мережі для прогнозування попиту на пасажирські перевезення таксі за допомогою графічних процесорів, що дало змогу пришвидшити процедуру навчання за різних наборів вхідних даних і конфігурацій апаратного забезпечення та його потужності. З'ясовано, що послуги таксі стають доступнішими для більшої кількості людей. Найважливішим завданням будь-якої компанії та водія таксі є мінімізація тривалості очікування нових замовлень та відстані до клієнтів на момент їх замовлення. Аби досягти цієї мети, потрібно мати розуміння транспортної логістики та вміння оцінити географічний попит на перевезення залежно від багатьох чинників. Розглянуто приклад тренування нейронної мережі для передбачення попиту на пасажирські перевезення таксі. Встановлено, щоб нейронна мережа давала хороші прогнози, необхідно обробити великий набір вхідних даних. Оскільки навчання нейронної мережі – це довготривалий процес, то для вирішення цієї проблеми було застосовано розпаралелювання процедури навчання мережі з використанням графічних процесорів.

Проведено навчання нейронної мережі на центральному процесорі, одному та двох графічних процесорах відповідно, виконано порівняння тривалості процедури навчання мережі для однієї епохи. Оцінено вплив кількості використаних графічних процесорів на тривалість тренування нейронної мережі у двох різних конфігураціях апаратного забезпечення та його потужності. Тренування мережі здійснено за допомогою набору даних, який містить 4.5 млн поїздок у межах одного міста. Результати дослідження показують, що пришвидшення процедури навчання за допомогою графічних процесорів не завжди дає позитивний результат, позаяк залежить від багатьох чинників – розміру вибірки вхідних даних, правильного поділу вибірки даних на менші підвибірки, а також характеристик апаратного забезпечення та його потужності.

Ключові слова: машинне навчання, прогнозування попиту, тренування нейронної мережі, пришвидшення процедури навчання, паралелізація процедури тренування.

Вступ

Послуги таксі є зручним способом перевезення пасажирів і їхнього багажу не тільки у великих і малих містах, але й малих населених пунктах. Як і багато інших сфер надання послуг, логістика пасажирських перевезень таксі переживають сьогодні швидку цифрову трансформацію. Нові гравці, такі як, наприклад, Uber, збільшують свою частку ринку за допомогою інноваційних цифрових продуктів [31]. З кожним роком послуги таксі стають доступнішими для значної кількості потенційних пасажирів.

Найважливішим завданням будь-якої компанії та водія таксі є мінімізація тривалості очікування нових замовлень та відстані до клієнтів на момент їх надходження. Аби досягти цієї мети, потрібно мати розуміння транспортної логістики та вміння оцінити географічний попит на перевезення пасажирів залежно від багатьох чинників, наприклад, пори дня, погоди, святкових чи вихідних днів, культурних заходів і т.п. Прогнозування місця перебування майбутніх пасажирів – це проблема, яку можна вирішити за допомогою нейронних мереж і їхнього машинного навчання.

Останнім часом нейронні мережі набирають неабиякої популярності, позаяк їх широко застосовують для розв'язання різних задач з багатьох предметних областей знань, наприклад, туризму, комерції, маркетингу та інші. Проте, для вирішення завдань прогнозування попиту на товари чи послуги за допомогою машинного навчання не знайшли широкого застосування у повсякденній роботі, наприклад, відповідних сервісів пасажирських перевезень.

Для того, щоб нейронна мережа давала хороші прогнози, необхідно обробити значний набір вхідних даних [16]. Тренування мережі по суті означає вибір однієї моделі з множини дозволених моделей (або, в баєсовій системі – визначення розподілу над множиною дозволених моделей), що зводить витрати на її навчання до мінімуму [18]. Отже, тренування мережі є найбільш ресурсомістким завданням. Щоб вирішити цю проблему якнайшвидше, застосовують розпаралелювання процедури навчання мережі з використанням графічних процесорів [32].

Для реалізації процедури передбачення попиту на надання послуг часто використовують багатошаровий персептрон (англ. *Multilayer Perceptron*). Для обчислення та оновлення ваг елементів мережі застосовують техніку стохастичного градієнта спуску [20]. Функцією втрати для вирішення проблеми прогнозування часто вибирають середньоквадратичну помилку.

Об'єкт дослідження – тренування нейронної мережі для передбачення попиту на перевезення таксі.

Предмет дослідження – методи і засоби, які дадуть змогу пришвидшити процедуру навчання нейронної мережі для передбачення попиту на пасажирські перевезення таксі за різних умов зовнішнього середовища, конфігурації апаратного забезпечення та його потужності.

Мета роботи – дослідити вплив паралелізації нейронної мереж, що має архітектуру багатошарового персептрону, на швидкість її навчання порівняно зі звичайним тренуванням за допомогою центрального процесора.

Для досягнення зазначеної мети визначено такі *основні завдання дослідження*: здійснити порівняння тривалості процедури навчання розпаралеленої нейронної мережі на одному та двох графічних процесорах, а також тренування на центральному процесорі.

Наукова новизна отриманих результатів дослідження – розроблено підхід, який дає можливість провести тренування нейронної мережі для прогнозування попиту на пасажирські перевезення таксі за допомогою графічних процесорів, що дало змогу пришвидшити її навчання за різних наборів вхідних даних і конфігурації апаратного забезпечення та його потужності.

Практична значущість результатів дослідження полягає у тому, що вони показали незначну користь від використання графічних процесорів для паралельного тренування обраної нейронної мережі, що має архітектуру багатошарового персептрону. Якщо вибірка даних недостатньо велика для паралелізації процедури навчання на декількох графічних процесорах, то таке тренування є значно тривалішим, ніж звичайне на одному графічному або центральному процесорі.

Аналіз останніх досліджень та публікацій. Сьогодні методи для вирішення завдань прогнозування попиту на надання послуг не знайшли широкого застосування у повсякденній роботі сервісів пасажирських перевезень таксі. Проте, значна кількість сучасних досліджень свідчать про актуальність цієї проблеми. Прогнозування місця перебування майбутніх пасажирів – це проблема, що ідеально підходить для її вирішення за допомогою машинного навчання нейронної мережі. Існують закономірності, які дуже складні для розуміння транспортної логістики та вирішення оператором таксі, але їх можна формалізувати, сформулювати у вигляді постановки задачі й розв'язати за допомогою машинного навчання [26]. Це є основною характеристикою проблеми, що стосуються штучних нейронних мереж і їх машинного навчання [7].

З іншого боку, для того, щоб мережа давала хороші прогнози, необхідно обробити великий набір вхідних даних [15]. Тому навчання нейронної мережі є найбільш ресурсомістким завданням. Іноді ця процедура може виконуватись за лічені хвилини, проте бувають випадки, коли традиційний підхід до тренування мережі не може впоратись з таким обсягом вхідних даних. Щоб вирішити цю проблему, науковці все більше схиляються до розпаралелювання процедури навчання мережі з використанням графічних процесорів [10].

Fei Miao та інші [13] запропонували сучасну надійну інформаційну систему, яка вивчає дані, зібрані з наявної транспортної мережі. На підставі отриманих знань відбувається аналіз попиту на пасажирські перевезення таксі. В роботі наведено методи прогнозування попиту на надання послуг таксі, використовуючи інформацію про тривалість їхнього перебування в дорозі та швидкість пересування відповідно до стану дорожнього руху. Запропонована ними робототехнічна модель збалансовує кількість поїздок таксі та пропонує найкращу систему паркування на підставі попередніх платежів водія. Розроблені алгоритми спрямовані на скорочення відстані до клієнтів і мінімізації тривалості очікування як замовлення, так і потенційних клієнтів. Хоча оптимізація в даному підході має на меті мінімізувати найбільші витрати зі всіх наявних, використовуючи випадкові параметри, однак це призводить до середньої продуктивності системи загалом. Проте, для системи відправлення таксі важливо вирішити питання компенсації між найгіршим випадком і середніми витратами на відправлення при невизначеному попиті. Оцінки експертів показують, що відповідно до розробленої авторами надійної системи відправки таксі, середньостатистичний дисбаланс співвідношення попиту та пропозиції зменшується на 31,7 %, а середній загальний пробіг без пасажирів скорочується на 10,13 %, або приблизно на 32 млн км за один рік.

Mohammad Saiedur Rahaman та інші [6] визначили проблему ідентифікації сусідства таксі за наявності великої кількості різнорідних контекстуальних модулів. У своїй роботі автори досліджували проблему прогнозування тривалості очікування водіями таксі пасажирів в аеропортах, а також різнорідні елементи, пов'язані з періодом доби, погодою, прибуттями авіарейсів і поїздками пасажирів на таксі. Авторами було встановлено, що таксі вважають найпростішим видом транспорту для трансферу між аеропортом і містом. Менеджери перевезень пасажирів на таксі постійно контролюють кількість авіарейсів і дають вказівки водіям забрати потенційних клієнтів з терміналів. Щоб забезпечити безперебійність цього процесу, менеджери оцінюють попит на таксі з прийдешнього авіарейсу. Рейтинг задоволеності водіїв таксі від кількості замовлень в аеропортах залежить від належного управління чергами пасажирів і таксі. Намагаючись підтримувати симетричність попиту на таксі, менеджери аеропортових перевезень застосовують підхід, який вимагає розширеного втручання людино-машинної системи. Проведені авторами наукові дослідження стверджують, що якість результатів для окресленого ними мікрорайону значно покращується за рахунок врахування відповідних неоднорідних контекстуальних чинників, тому ефективність збільшується.

Jun Xu та інші [25] запропонували систему, в якій вони використали періодичні нейронні мережі для прогнозування попиту на таксі у майбутньому на підставі історичних даних, які мають GPS. Вони використовували довготривалу пам'ять (LSTM) для прогнозування майбутнього попиту на пасажирські перевезення таксі. Зазвичай, LSTM використовують для розпізнавання рукописного тексту, оброблення природних мов тощо. Однак, у своїй моделі автори використовують деякий механізм для зберігання попереднього значення, тобто мережу змішаної щільності (MDN) разом із довготривалою пам'яттю (LSTM). Цей алгоритм дає змогу досягти 83 % точності при прогнозуванні майбутнього попиту на пасажирські перевезення таксі.

Nicholas Jing Yuan та інші [1] запропонували систему рекомендацій як для водіїв таксі, так і для пасажирів, які очікують прибуття таксі. У своєму методі автори застосовують знання про ймовірність прибуття таксі за траєкторією його руху по GPS. Вони розробили ймовірнісну модель для формулювання залежної від часу поведінки таксі (підбирання/висадки/подорожі/парку-

вання) та запровадили систему міських рекомендацій як для пасажирів, так і для водіїв таксі. Також автори вдосконалили рекомендації водіям таксі, враховуючи час та довжину черги на місцях паркування. Як для водіїв таксі, так і для пасажирів автори побудували відповідні моделі, що враховують день тижня та історичні погодні умови.

Moreira-Matias та інші [23] розробили альтернативний до машинного навчання метод прогнозування оцінки попиту пасажирів на найближчий 30-хвилинний період. А саме, методи прогнозування подій GPS в реальному часі, переданих між 441 автомобілями в мережі таксі. Вони порівняли свою оцінку в реальному часі з фактичним попитом на 63 таксі в місті Порто. Випробувана модель досягла точності понад 74 %.

Naoto Mukai та Naoto Yoden [24] спробували реалізувати подібний спосіб прогнозування попиту на пасажирські перевезення таксі на підставі географічних районів та часових відрізків. Вони згрупували наявні дані про 25 основних районів Токіо за часовими відрізками по чотири години та передбачили попит у кожному районі міста на майбутній проміжок часу. Цей підхід дещо відрізняється від підходу у роботі [23], оскільки вони не передбачають використання доступних нових даних для прогнозування наступного набору попиту. Їхній підхід також можна використовувати для значно меншої кількості вхідних параметрів передбачення попиту на таксі.

Науковцям вдалося передбачити попит на таксі для районів Токіо з похибкою в межах 6-24 % для різних періодів доби та зон міста [9]. Вони намагались врахувати у своїх дослідженнях кількість опадів як булевий вхідний параметр – йде дощ чи не дощить. Проте, вони зробили висновок, що це немає статистичного значення. Однак, автори стверджують, що більш обширні параметри погоди можуть, можливо, дати кращі прогнози. Наприклад, наявність кількості опадів як безперервний параметр або врахування його, якщо кількість опадів досягає певного порогу.

Грінберг та інші [12] зробили подібне прогнозування попиту на таксі для Нью-Йорка. Розбивши місто на багато квадратів, автори намагались передбачити кількість поїздок сітками квадратів за дану годину. Вони спробували три підходи до машинного навчання: лінійну регресію найменших квадратів, регресію вектора підтримки (англ. *Support Vector Machines*) та регресію дерева рішень. У дослідженні було використано різні набори функцій, що містять зону міста, період доби, день тижня та погодинну кількість опадів. Як і Naoto Mukai та Naoto Yoden, вони не вважають, що кількість опадів є статистично значущим параметром.

Schaller [2] досліджував метод прогнозування попиту пасажирських перевезень таксі на більш високому рівні, щоб запропонувати алгоритм регулювання автомобілів таксі в містах США. Він використовує різноманітні параметри для прогнозування попиту на таксі: чисельність населення та рівень його зайнятості, громадський транспорт, рівень туризму та заходи з великими подіями, кількість ділових відвідувачів, частку населення з низьким рівнем доходу та володіння транспортними засобами. Внаслідок проведеного дослідження науковець виявив, що існує три параметри надзвичайно важливі у своїй здатності передбачати попит: кількість власних автомобілів, активність в аеропорту та поїздки в метро.

Pal та інші [27] досліджували метод гібридного паралелізму, що є більш ефективним засобом у зменшені загальної тривалості процедури навчання нейронної мережі ніж паралелізм даних чи моделі. Krizhevsky [19] запровадив один гібридний підхід до навчання згорткових нейронних мереж, який поєднує паралелізм даних для обчислювальних частин моделі (згорткові шари) разом із паралелізмом моделі для шарів з великою кількістю параметрів (повністю зв'язані шари). Цей підхід масштабується значно краще, ніж усі альтернативи сучасних конволюційних мереж.

Отже, для навчання нейронної мережі нам варто використовувати значну кількість наборів вхідних даних. В нашому дослідженні було використано 4,5 млн поїздок таксі компанії Uber в межах Нью-Йорка за період з квітня по жовтень 2014 року. Для більшої точності передбачень нами було зібрано та додано до набору даних погодинні метеорологічні дані за цей період, а саме, температуру повітря та наявність семи видів атмосферних явищ. Тренування мережі проводили на одному центральному процесорі, одному та двох графічних процесорах NVIDIA TESLA K80.

Для проведення дослідження у цій роботі взято за основу час, за який нейронна мережа проходить одну епоху (англ. *epoch*) тренування. Швидкість процедури навчання нейронної мережі залежить, насамперед, від обраного алгоритму її тренування та архітектури програмної системи, кількості наборів даних для її тренування та їхнього обсягу, комплектації апаратного забезпечення та його потужності. Дослідження виконано для двох різних конфігурацій апаратного забезпечення, аби показати користь від паралелізації нейронної мережі залежно від розміру вхідних даних у наборі та їхньої кількості.

Результати дослідження та їх обговорення

Передбачення попиту на пасажирські перевезення таксі. Найважливішим завданням будь-якої компанії та водія таксі є мінімізація тривалості очікування нових замовлень та відстані до клієнтів на момент їх надходження. Зараз водіям необхідно самостійно вирішувати, де чекати пасажирів, так щоб вони могли їх швидко забрати. Природно, що вони ніколи не можуть достовірно знати, у якому місці будуть майбутні пасажири, однак досвідчені водії можуть робити здогадки та прогнози на підставі своїх знань. Система відправки таксі ефективно допомагає клієнтам та водіям. Ефективна робота цієї системи також допомагає скоротити тривалість очікування як замовлення, так і самого таксі. Проте водії однаково не мають достатньої інформації про те, де чекати, щоб швидко приїхати до нового пасажира. А оператор таксі-центру може організувати та надіслати необхідну кількість водіїв до району, виходячи з історичних даних.

Відомо [22], що попит на пасажирські перевезення таксі змінюється під впливом різних чинників. Окремі проблеми виникають через паркування у місті. Нерідко у певних районах, особливо ближче до центральної частини міста, не вистачає вільних місць на парковці. Якщо вдалось знайти паркомісце, то може стягуватись

плата за час очікування наступного замовлення у цьому місці. Також необхідно взяти до уваги й те, що бувають випадки, коли нові замовлення клієнтів з'являються дуже далеко від місця очікування водія.

Отже, аби досягти цієї мети, потрібно мати розуміння транспортної логістики та вміння оцінити можливий географічний попит на пасажирські перевезення таксі залежно від багатьох чинників, наприклад, пори дня, погоди, святкових днів, культурних заходів і т.п.

Дані для тренування. Для тренування нейронної мережі у роботі використовується набір вхідних даних, що містить 4,5 млн поїздок таксі компанії Uber в межах Нью-Йорка за період з квітня по жовтень 2014 року. Набір вхідних даних взято з відкритих джерел [22]. Для більшої точності передбачень зібрано та додано до набору даних погодинні метеорологічні дані за цей самий період, а саме температуру повітря та наявність семи видів атмосферних явищ.

Сформований набір вхідних даних для тренування нейронної мережі має такі атрибути: день, місяць і рік поїздки, широта та довгота початкової точки поїздки, температура повітря, наявність легкого, середнього та сильного дощу, наявність туману, серпанку, легкого снігу [30]. Широта та довгота початкової точки поїздки – це атрибути, які нейронна мережа повинна передбачати для визначення місцезнаходження пасажира у заданий час та наявну погоду [8], [29].

Багатошаровий перцептрон. Для передбачення місцезнаходження наступного пасажира використано *багатошаровий персептрон* (англ. *Multilayer Perceptron*) як один з видів нейронних мереж. *Багатошаровий персептрон* – це різновид мережі прямого зв'язку з одним або декількома рівнями внутрішніх шарів між вхідним і вихідним шарами. Вхідний сигнал у такій мережі поширюється в одному напрямку, від шару до шару. Вихідні одиниці представляють гіперплощину в просторі шаблонів введення. Мережа складається з M шарів, кожен з яких містить $J_m, m = \overline{1,M}$ вузлів. Ваги від (m–1)-го шару до m-го шару позначаються $w^{m\text{-}1}$. Функція зміщення, виходу та активації i-го нейрона в m-му шарі, відповідно, позначається як $\theta_i^{(m)}$, $\theta_i^{(m)}$ та $\phi_i^{(m)}(\cdot)$ [5].

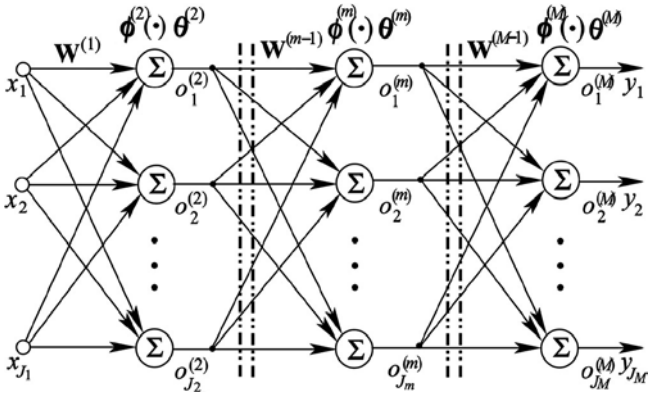

Рис. 1. Архітектура багатошарового персептрону [5]

Багатошарові персептрони успішно застосовують для вирішення різноманітних завдань та мають такі особливості визнання. Кожна нейронна мережа має нелінійну функцію активації. Окрім шарів входу та виходу, такого типу нейронна мережа має декілька прихованих шарів.

Засіб для тренування нейронної мережі. Для тренування нейронної мережі нами використано оптимізовану тензорну бібліотеку з відкритим кодом для глибокого навчання з використанням графічних процесорів та процесорів PyTorch [28]. Вона забезпечує максимальну гнучкість та швидкість наукових обчислень для глибокого навчання.

Використання бібліотеки PyTorch для тренування мережі має деякі переваги. Хороша документація та простий синтаксис дає змогу швидко розробити потрібну кодову базу. Бібліотека дає можливість розпаралелити процедуру тренування. З її використанням можна розподіляти обчислювальну роботу між декількома ядрами звичайного або графічного процесора [3].

Стохастичний градієнтний спуск. Найпоширенішою технікою для обчислення та оновлення ваг мережі є *стохастичний градієнтний спуск* SGD (англ. *Stochastic gradient descent*). SGD виконує такі кроки ітеративно. Спочатку обчислюється крок передачі вперед. Вхідні вибірки обробляються шар за шаром, поки не буде отримано прогноз після останнього шару. На наступному кроці, зворотному розповсюдженні, ваги оновлюються на підставі обчисленої різниці (градієнтів) між прогнозованими та позначеними результатами. Ці кроки виконуються ітеративно для всіх міні-пакетів (англ. *mini-batch*) у наборі даних. Як тільки всі міні-пакети опрацьовано, то одна епоха завершується. Це означає, що весь набір вхідних даних передавався вперед і назад через DNN тільки один раз. Процес можна продовжувати протягом наступних епох. SGD прагне мінімізувати функцію втрати, знаходячи ваги, які б у кращому випадку відповідали цілому набору даних. Отже, SGD виконує функцію апроксимації для всієї навчальної процедури [21].

Останнім часом набуває популярності використання оптимізатора типу SGD Adam. Він використовує квадратичні градієнти для масштабування швидкості процедури навчання, та імпульс, використовуючи ковзну середню градієнта замість самого градієнта. Метод використовує оцінки першого та другого моментів градієнта, щоб адаптувати швидкість процедури навчання для кожної ваги нейронної мережі [17]. Adam добре підходить для вирішення проблем машинного навчання з великими наборами даних, а також з глибокими багатошаровими нейронними мережами, частково завдяки хорошому коефіцієнту пам'яті.

Функції втрати. Нейронну мережу можна навчати, використовуючи для цього попередні значення вхідних і результуючих даних для оновлення ваг, отримуючи при цьому правильний вихід. Така процедура навчання мережі межує з проблемою її оптимізації, де помилка навчання мережі має зводитися до мінімуму. Щоб мінімізувати цю помилку та правильно навчити мережу, насамперед помилку потрібно визначити.

Функція втрати (англ. *Loss function*) описує, наскільки далеко знаходиться мережа від здійснення ідеальних прогнозів для заданих даних [4]. Результатом роботи функції втрати є відносне значення. Якщо прогнози покращуються, то значення функції втрати зменшується. Вибір типу функції для нейронної мережі значною мірою залежить від проблеми, яку потрібно вирішити. Для прогнозування попиту на пасажирські

перевезення таксі як функцію втрати найчастіше використовують *середньоквадратичну помилку* MSE (англ. *Mean Square Error*). Визначають яку суму квадратів відстаней між цільовою змінною та передбаченими значеннями, поділених на їх кількість, а саме

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(v_i y_i - y^p)^2 , \quad (1)$$

де: n – кількість нейронів у схованому шарі; v_i – вага синапсу j-го нейрона схованого шару до вихідного нейрона; y_i – вихідне значення j-го нейрона схованого шару (цільова змінна); y^p – поріг вихідного нейрона (передбачене значення).

Тренування нейронної мережі на графічному процесорі. Для будь-якої нейронної мережі стадія її навчання є найбільш ресурсомістким завданням. Якщо нейронна мережа має приблизно 10, 100 або навіть 100 000 параметрів, то звичайний комп'ютер однаково зможе впоратися з цим за лічені хвилини, чи десятки хвилин. Проте, якщо нейронна мережа має понад 10 мільярдів параметрів, тоді для її навчання, використовуючи традиційний підхід, були б потрібні місяці.

Глибока нейронна мережа DNN (англ. *Deep Neu Ral Network*) – це штучна нейронна мережа з декількома прихованими шарами. Подібно до звичайних нейронних мереж, глибокі нейронні мережі можуть моделювати складні нелінійні відносини між елементами. Під час навчання глибокої нейронної мережі отримувана модель намагається подати об'єкт у вигляді комбінації простих примітивів (наприклад, у задачі розпізнавання осіб такими примітивами можуть бути частини обличчя: ніс, очі, рот і т.д.). Додаткові шари дають змогу будувати абстракції все більш високих рівнів, що і дає можливість будувати моделі для розпізнавання складних об'єктів реального світу.

Як правило, глибинні мережі будуються як мережі прямого поширення. Однак останні дослідження показали [17], як можна застосувати техніку глибинного навчання для рекурентних нейронних мереж. Згорткові нейронні мережі використовують в області машинного зору, де цей підхід показав себе як ефективний. Також згорткові нейронні мережі були застосовані для розпізнавання мови.

Навчання глибинних нейронних мереж можна здійснити за допомогою звичайного алгоритму зворотного поширення помилки. Існує велика кількість модифікацій такого алгоритму, в яких використовують декілька правил налаштування ваг. Наприклад, для навчання вагових коефіцієнтів $\omega_{ij}(t)$ можна використати алгоритм стохастичного градієнтного спуску:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta\frac{\partial C}{\partial \omega_{ij}}, j = \overline{1,m};\ i = \overline{1,n} , \quad (2)$$

де: η – стала для регулювання величини поточного кроку; C – функція втрат. Вибір функції втрат може бути обумовлений класом завдання машинного навчання (з учителем, без учителя, з підкріпленням) і функцією активації. До двох головних проблем глибоких нейронних мереж належать ті ж проблеми, що виникають і при навчанні звичайних нейронних мереж: тривалість навчання та перенавчання.

Глибокі структури нейронних мереж більше схильні до перенавчання, оскільки, маючи значну кількість шарів, що дають змогу моделювати високорівневі абстракції, нейронна мережа може "вивчити" рідкісні ситуації. У цьому випадку можуть допомогти різні види регуляризації процедури навчання мережі. Один з можливих методів регуляризації (англ. *Dropout*) припускає випадково вилучені вузли нейронної мережі під час навчання. У деяких випадках це допомагає значно менше запам'ятовувати рідкісні залежності в тренувальних наборах даних.

Через простоту реалізації та хорошу збіжність для реалізації процедури навчання глибоких нейронних мереж часто використовують метод зворотного поширення помилки і градієнтний спуск. Однак, при навчанні глибоких структур мережі виникає декілька проблем, які особливо важливі при оптимізації функцій втрат у просторі великої розмірності: кількість обчислювальних елементів, початкові умови для ваг мережі, а також константа регулювання величини кроку.

Варто зазначити, що алгоритм стохастичного градієнтного спуску відомий своєю проблемою зникаючого градієнта (англ. *Vanishing Gradient*), яка полягає в ослабленні градієнта, а значить і зменшення швидкості процесу навчання мережі в міру поглиблення від останніх її шарів до початку мережі. Через це глибокі шари нейронної мережі дуже погано навчаються. Проте, останнім часом деякі науковці [23], [27] замість функції активації вузла мережі виду сигмоїда в глибоких нейронних мережах почали використовувати нелінійність виду ReLU (англ. *Rectified Linear Unit*), функцію якої можна описати як $\max(0, x)$. За такого підходу глибока нейронна мережа з таким видом функції активації немає проблеми ослаблення градієнта і добре навчається методом градієнтного спуску. За умов великих розмірностей повний перебір всіх комбінацій значень параметрів непрактичний.

Глибинні нейронні мережі можна навчати значно швидше, виконуючи для цього всі операції одночасно, а не одну за одною. Цього можна досягти, застосовуючи графічний процесор, наприклад, GPU (англ. *Graphics Processing Unit*) – спеціалізований процесор із виділеною пам'яттю, який, зазвичай, виконує операції з плаваючою комою, необхідні для рендерингу графіки [11]. Сучасні графічні процесори дуже ефективно обробляють та відображують комп'ютерну графіку завдяки спеціалізованій конвеєрній архітектурі, вони набагато ефективніші під час оброблення графічної інформації, ніж типовий центральний процесор. Також графічні процесори оптимізовані для навчання моделей штучного інтелекту та проведення глибокого навчання, оскільки вони можуть обробляти декілька обчислень одночасно.

Паралелізація даних в нейронних мережах. Ефективне навчання нейронних мереж є важливою частиною глибокого їх навчання. Зі збільшенням обсягу навчальних наборів даних й складності моделі мережі пропорційно зростає обчислювальна потужність процесора та потреба в пам'яті комп'ютера. Пришвидшення процедури навчання сприяє підвищенню якості моделі, даючи можливість здійснити навчання нейронних мереж на значно більших наборах даних.

Паралельне навчання глибинних нейронних мереж має істотні переваги перед серійним навчанням. Нерідкі випадки, коли великі моделі нейронних мереж не можуть поміститися в пам'яті одного графічного процесо-

ра. Навчання різних частин моделі на різних графічних процесорах – це простий спосіб подолати обмеження пам'яті графічних процесорів. Водночас, застосування паралельних навчальних стратегій приводять до дещо швидшого навчання, розподіляючи перекриваючі обчислення між різними вузлами. Використання обчислень, зв'язку та перекриття введення/виведення даних сприяє кращому використанню обчислювальних ресурсів і, як наслідок, значно швидшій конвергенції при навчанні моделі мережі.

Варто зазначити, що під час навчання нейронної мережі за допомогою паралелізації даних нами було враховано такі дві характеристики: витрати на зв'язок між різними вузлами для синхронізації градієнтів під тривалість тренування та витрати часу для обчислень [14].

Обговорення результатів дослідження. Дослідження здійснено на центральному процесорі, одному графічному процесорі NVIDIA Tesla K80 та на двох таких процесорах відповідно. Тренування мережі проведено на двох різних розмірах вхідних даних batch_size = 16 та batch_size = 2048. Результатом дослідження є порівняння часу для однієї епохи тренування.

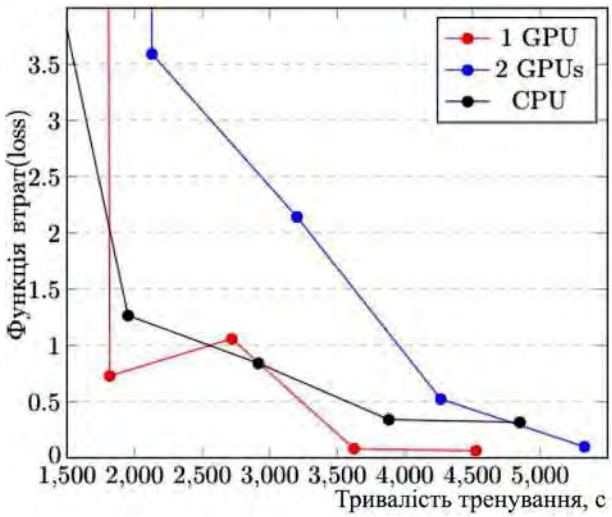

Рис. 2. Залежність тривалості процедури навчання від кількості використаних ресурсів (mini-batch = 16)

Дані, наведені на рис. 2, показують порівняння тривалості процедури навчання нейронної мережі для однієї епохи на одному центральному процесорі, одному та двох графічних процесорах, використовуючи для цього mini-batch розміром 16 проб. Це порівняння представлене на осі *X*. Вісь *Y* представляє функцію втрати – середньоквадратична помилка.

Дослідження на одному графічному процесорі (червона лінія) пришвидшує тренування на центральному процесорі (чорна лінія) у 1.0715 разів. Таке невелике пришвидшення процедури навчання пояснюється малим розміром набору даних, що розпаралелюються (розмір mini-batch) та значними витратами часу для передачі даних між графічним і центральним процесором, який керує процедурою тренуванням.

Тренування на двох графічних процесорах (синя лінія) погіршує тривалість процедури навчання мережі порівняно з тренуванням на одному графічному або центральному процесорі. Такий результат пояснюється високими витратами часу для синхронізації градієнтів спуску між двома графічними процесорами перед кожним оновленням параметрів мережі (після кожної batch-ітерації). Ці витрати для синхронізації не перекриваються ефектом від паралелізації обчислень, оскільки розмір вхідних даних, що розпаралелюються (розмір mini-batch), є занадто малим.

Отже, при використанні mini-batch розміру 16 проб найоптимальнішим є тренування обраної конфігурації нейронної мережі для передбачення попиту на пасажирські перевезення таксі тільки на одному графічному процесорі.

Дані, наведені на рис. 3, показують результат порівняння тривалості процедури навчання нейронної мережі для однієї епохи на одному центральному процесорі, одному та двох графічних процесорах, використовуючи mini-batch розміром 2048 проб. Розмір вхідних даних, що розпаралелюються, mini-batch розмір, є у 128 разів більший за розмір вхідних даних, використаний для отримання результатів з рис. 2.

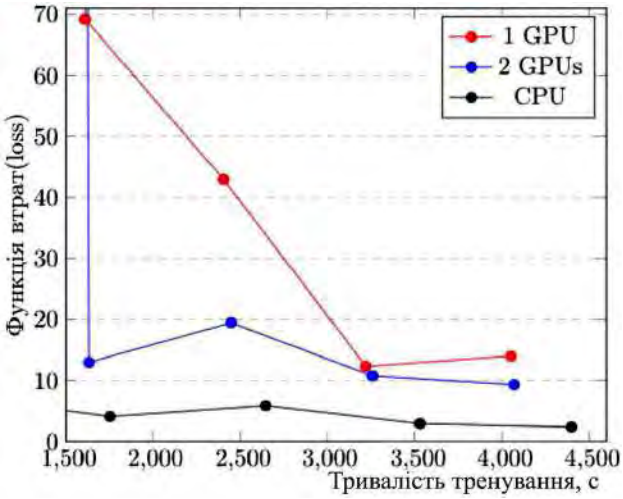

Рис. 3. Залежність тривалості процедури навчання від кількості використаних ресурсів (mini-batch = 2048)

Тренування нейронної мережі на одному (червона лінія) та двох (синя лінія) графічних процесорах є у 1.0861 та 1.0811 відповідно разів швидше, ніж тренування на одному центральному процесорі (чорна лінія). Порівняно з результатами для mini-batch розміру 16 проб, тренування на двох графічних процесорах відбувається значно швидше, ніж на одному центральному процесорі. Проте, немає пришвидшення між тренуванням мережі на одному та двох графічних процесорах, оскільки користь від паралелізації обчислень на двох графічних процесорах компенсується витратами на синхронізацію між цими процесорами під тривалість тренування.

Інформація, наведена на рис. 4, показує залежність тривалості процедури навчання нейронної мережі для однієї епохи між усіма використаними у цій роботі апаратними конфігураціями комп'ютерної техніки. Тренування на одному графічному процесорі з використанням mini-batch розміру 2048 проб було найменшим, водночас як аналогічне тренування на двох графічних процесорах розміру 16 проб було найбільшим.

Різниця між тривалістю тренування нейронної мережі для однієї епохи знаходиться в межах сотень секунд для обраних апаратних конфігурацій та вибраної архітектури нейронної мережі. Варто зазначити, що архітектура розглянутої нейронної мережі містить велику

кількість параметрів, та відносно малу кількість обчислень, порівняно з іншими архітектурами.

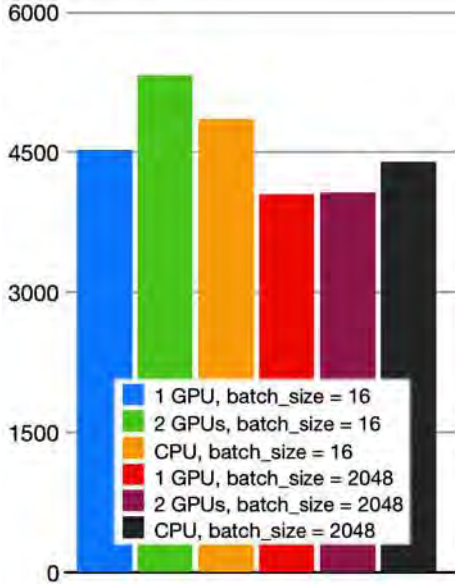

Рис. 4. Зведена залежність тривалості процедури навчання від кількості використаних ресурсів

Так, наприклад, для згорткових нейронних мереж, де кількість обчислень більша, ефект від паралелізації на декількох графічних процесорах буде більший. Проте, витрати для синхронізації між графічними процесорами будуть сталими. Отже, за допомогою розглянутих методів паралелізації процедури тренування нейронних мереж, можна досягти значно кращих результатів за умови обрання відповідної архітектури програмної системи, основою якої є нейронна мережа.

Висновки

1. З'ясовано, що для мінімізації тривалості очікування нових замовлень та відстані до клієнтів на момент їх надходження, а саме для вирішення проблеми прогнозування попиту на пасажирські перевезення таксі, доцільно використовувати засоби машинного навчання.

2. Встановлено, щоб досягти великої точності у прогнозуванні попиту на пасажирські перевезення таксі, нами опрацьовано великий набір вхідних даних, а саме 4,5 млн поїздок таксі. Для зменшення тривалості процесу тренування нейронної мережі застосовано розпаралелювання процедури її навчання з використанням графічних процесорів.

3. Проведено тренування нейронної мережі на центральному процесорі, одному та двох графічних процесорах відповідно. Паралелізація на декількох графічних процесорах не завжди означає пришвидшення процедури навчання, оскільки витрати на комунікацію між активними процесами на різних графічних процесорах перевищують користь від паралелізації процедури тренування. За умови обрання достатньо великого обсягу даних для паралелізації виконання розрахунків (великий mini-batch розмір) можна досягти шляхом пришвидшення процедури навчання мережі.

4. Визначено, що пришвидшення процедури навчання нейронної мережі залежить від багатьох чинників: її архітектури, гіперпараметрів тренування, конфігурації апаратного забезпечення та методу паралелізації виконання розрахунків.

References

[1] Biao Leng, Heng Du, Jianyuan Wang, Li Li, & Zhang Xiong. (2016). Analysis of Taxi Drivers Behaviors Within a Battle Between Two Taxi Apps. *IEEE Transactions on Intelligent Transportation Systems, 17*(1), 296–300. https://doi.org/10.1109/TITS.2015.2461000

[2] Bruce Schaller. (2005). A regression model of the number of taxicabs in US cities. *Journal of Public Transportation, 8*(5), 4–11. http://doi.org/10.5038/2375-0901.8.5.4

[3] Dhiraj, K. (2019). *10 reasons why PyTorch is the deep learning framework of the future*. Retrieved from: https://heartbeat.fritz.ai/10-reasons-why-pytorch-is-the-deep-learning-framework-of-future-6788bd6b5cc2

[4] Dipanjan Sarkar, Raghav Bali, & Tushar Sharma. (2018). *Practical Machine Learning with Python*. Springer Science+Business Media. New York.

[5] Du, K.-L., & Swamy, M.N.s. (2014). Multilayer Perceptrons: Architecture and Error Backpropagation. *Neural Networks and Statistical Learning*, pp. 83–126. https://doi.org/10.1007/978-1-4471-5571-3_4

[6] Fei Miao, Shuo Han, Shan Lin, Qian Wang, John A. Stankovic, Abdeltawab Hendawi, Desheng Zhang, Tain He, & George J. Pappas. (2019). Data-Driven Robust Taxi Dispatch Under Demand Uncertainties. *IEEE Transactions on Control Systems Technology, 17*(1), 175–191. https://doi.org/10.1109/TCST.2017.2766042

[7] Firmino, P., de Mattos, Neto P., & Ferreira, T. (2014). Correcting and combining time series forecasters. *Neural Networks, 50*, 1–11.

[8] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(1), 142–158. https://doi.org/10.1109/TPAMI.2015.2437384

[9] Grossberg, S. Z. (2010). *Neural Networks and Natural Intelligence*. Cambridge, MA: MIT Press, 651 p.

[10] Haykin, S. (2008). *Neural Networks and Learning Machines*. New Jersey: Prentice Hall, 936 p.

[11] Jason Dsouza. (2020). *What is a GPU and do you need one in Deep Learning*? Retrieved from: https://towardsdatascience.com/what-is-a-gpu-and-do-you-need-one-in-deep-learning-718b9597aa0d

[12] John Grinberg, Arzav Jain, & Arzav Vivek (2014). Predicting Taxi Pickups in New York City. Retrieved from: http://robots.stanford.edu/cs221/2016/restricted/projects/vhchoksi/final.pdf.

[13] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni, & Damla Turgut. (2018). Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks. *IEEE Transaction on Intelligent transport system, 19*(8), 2572–2581. https://doi.org/10.1109/TITS.2017.2755684

[14] Kennedy, R. K., Khoshgoftaar, T. M., Villanustre, F., & Humphrey, T. (2019). A parallel and distributed stochastic gradient descent implementation using commodity clusters. *Journal of Big Data, 6*(1), 16. https://doi.org/10.1186/s40537-019-0179-2

[15] Kiani, K. (2005). Detecting business cycle asymmetries using artificial neural networks and time series models. *Computational Economics, 26*(1), 65–89.

[16] Kim, Yoon. (2014). Convolutional neural networks for sentence classification. *IEMNLP*, 1746–1751.

[17] Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv – preprint arXiv: 1412.6980.

[18] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*, 1106–1114.

[19] Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *CoRR, abs/1404.5997*.

[20] Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567–581.
[21] Li, J., Nicolae, B., Wozniak, J., & Bosilca, G. (2019). Understanding scalability and fine-grain parallelism of synchronous data parallel training. *IEEE/ACM Workshop – Machine Learning in High Performance Computing Environments (MLHPC) IEEE,* pp. 1–8. https://doi.org/10.1109/MLHPC49564.2019.00006
[22] Lopatko, O., & Mykytyn, I. (2016). Neural networks as the means of forecasting the temperature value of a transient process. *Measuring Equipment and Metrology, 77*, 65–69.
[23] Luis Moreira-Matias, et al. (2012). A predictive model for the passenger demand on a taxi network. *International IEEE Conference on. IEEE, 15*, 1014–1019. https://doi.org/10.1109/ITSC.2012.6338680
[24] Naoto Mukai, & Naoto Yoden. (2012). Taxi Demand Forecasting Based on Taxi Probe Data by Neural Network. Intelligent Interactive Multimedia: Systems and Services. Ed. by Toyohide Watanabe et al. *Smart Innovation, Systems and Technologies 14*. Springer Berlin Heidelberg, pp. 589–597. https://doi.org/10.1007/978-3-642-29934-6_57
[25] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, & Xing Xie. (2013). T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering, 25*(10), 2390–2403. https://doi.org/10.1109/TKDE.2012.153
[26] Önder, E., Fırat, B., & Hepsen, A. (2013). Forecasting Macroeconomic Variables using Artificial Neural Network and Traditional Smoothing Techniques. *Journal of Applied Finance & Banking, 3*(4), 73–104.
[27] Pal, S., Ebrahimi, E., Zulfiqar, A., Fu, Y., Zhang, V., Migacz, S., Nellans, D., & Gupta, P. (2019). Optimizing multi-gpu parallelization strategies for deep learning training. *EEE Micro*, *39*(5), 91–101. https://doi.org/10.1109/MM.2019.2935967
[28] PyTorch. (2020). *PyTorch documentation*. Retrieved from: https://pytorch.org/docs/stable/index.html
[29] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
[30] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large–Scale Image Recognition. *CoRR, abs/1409.1556*. https://doi.org/10.1.1.740.6937
[31] YouTube. (2020). *Consumer assessment of taxi services in large cities*. Retrieved from: https://www.youtube.com/watch?v=RE2j1B7EdQM. [In Ukrainian].
[32] Zhang Xiang, Zhao Junbo, LeCun Yann. (2015). Character-level convolutional networks for text classification. In: *Advances in Neural Information Processing Systems*, pp. 649–657.

M. I. Zghoba, Yu. I. Hrytsiuk

Lviv Polytechnic National University, Lviv, Ukraine

TRAINING NEURAL NETWORK FOR TAXI PASSENGER DEMAND FORECASTING USING GRAPHICS PROCESSING UNITS

The peculiarities of neural network training for forecasting taxi passenger demand using graphics processing units are considered, which allowed to speed up the training procedure for different sets of input data, hardware configurations, and its power. It has been found that taxi services are becoming more accessible to a wide range of people. The most important task for any transportation company and taxi driver is to minimize the waiting time for new orders and to minimize the distance from drivers to passengers on order receiving. Understanding and assessing the geographical passenger demand that depends on many factors is crucial to achieve this goal. This paper describes an example of neural network training for predicting taxi passenger demand. It shows the importance of a large input dataset for the accuracy of the neural network. Since the training of a neural network is a lengthy process, parallel training was used to speed up the training.

The neural network for forecasting taxi passenger demand was trained using different hardware configurations, such as one CPU, one GPU, and two GPUs. The training times of one epoch were compared along with these configurations. The impact of different hardware configurations on training time was analyzed in this work. The network was trained using a dataset containing 4.5 million trips within one city. The results of this study show that the training with GPU accelerators doesn't necessarily improve the training time. The training time depends on many factors, such as input dataset size, splitting of the entire dataset into smaller subsets, as well as hardware and power characteristics.

Keywords: machine learning, demand forecasting, neural network training, training speedup, training parallelization.

Інформація про авторів:

Згоба Марія Іванівна, студентка, кафедра програмного забезпечення. **Email:** Mariia.Zghoba@gmail.com

Грицюк Юрій Іванович, д-р техн. наук, професор кафедри програмного забезпечення.
Email: yurii.i.hrytsiuk@lpnu.ua; https://orcid.org/0000-0001-8183-3466

Цитування за ДСТУ: Згоба М. І., Грицюк Ю. І. Тренування нейронної мережі для прогнозування попиту на пасажирські перевезення таксі за допомогою графічних процесорів. *Український журнал інформаційних технологій*. 2020, т. 2, № 1. С. 29–36.

Citation APA: Zghoba, M. I., & Hrytsiuk, Yu. I. (2020). Neural network training for forecasting the demand for passenger transportation by taxi using graphics processors. *Ukrainian Journal of Information Technology, 2*(1), 29–36. https://doi.org/10.23939/ujit2020.02.029