Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
https://nv.nltu.edu.ua
https://doi.org/10.36930/40310317
Article received 30.03.2021 p.
Article accepted 29.04.2021 p.
UDC 004.738.5:316.[728+772]

ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

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

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

ПРОГНОЗУВАННЯ ПОПИТУ НА ПАСАЖИРСЬКІ ПЕРЕВЕЗЕННЯ ТАКСІ МЕТОДАМИ НЕЙРОННОЇ МЕРЕЖІ

Розглянуто особливості прогнозування попиту на пасажирські перевезення таксі методами нейронної мережі за різних наборів вхідних даних, складу параметрів архітектури мережі, конфігурації апаратного забезпечення та його потужності. З'ясовано, що для зменшення тривалості очікування нових замовлень та відстані до клієнтів доцільно використовувати відповідні інформаційно-аналітичні системи, робота яких ґрунтується на штучному інтелекті. Це дасть змогу вирішити проблему попиту на перевезення таксі у відповідний період доби з врахуванням погодних умов, святкових, вихідних і робочих днів, а також пори року. Врахування ж наявних транспортних об'єктів – авіарейсів, потягів чи автобусів значно покращують роботу такої дорадчої системи. Використана в роботі гібридна архітектура нейро-фаззі мережі дає змогу одночасно вирішувати завдання короткотермінового прогнозування попиту на пасажирські перевезення таксі, а також проводити діагностику самої мережі, що полягає у виявленні різких змін властивостей обчислювального процесу. Для досягнення відповідної точності прогнозу в роботі опрацьовано набори вхідних даних у кількості 4,5 млн поїздок таксі. Для зменшення тривалості процедури навчання нейронної мережі організовано паралельні обчислення між різними вузлами мережі за допомогою графічних процесорів.

Проведено навчання нейронної мережі на центральному процесорі, одному та двох графічних процесорах відповідно. З'ясовано, що організація паралельних обчислень на декількох графічних процесорах не завжди зменшує тривалість процедури навчання мережі, оскільки витрати на синхронізацію градієнтів між активними процесами значно перевищують користь від паралельних розрахунків. Встановлено, що за умови великого обсягу даних для організації паралельних обчислень та відповідної архітектури нейронної мережі можна досягти деякого зменшення тривалості процедури її навчання. Визначено, що зменшення тривалості процедури навчання нейронної мережі залежить від таких чинників: її архітектури, кількості параметрів навчання, конфігурації апаратного забезпечення та організації паралельних розрахунків.

Ключові слова: інформаційно-аналітична система; конфігурація апаратного забезпечення; машинне навчання, навчання нейронної мережі, пришвидшення процедури навчання, паралелізація процедури навчання.

Вступ

Найважливішим завданням будь-якої компанії-перевізника та водія таксі є зведення до мінімуму тривалості очікування нових замовлень та відстані до клієнтів на момент їх надходження. Аби досягти цієї мети, потрібно розуміти транспортну логістику та вміння оцінити місцевий попит на перевезення пасажирів залежно від, наприклад, погоди, пори дня, святкових чи вихідних днів, культурних заходів і т.п. Знання місця перебування майбутніх пасажирів – це проблема, яку більшість водіїв таксі вирішують на свій розсуд залежно від кваліфікації та набутого досвіду. Багато ж інших перевізників використовують поради, наприклад, інноваційних програмних продуктів [52], робота яких, зазвичай, побудована на штучному інтелекті.

Існують різні підходи до розроблення систем штучного інтелекту [45]. Наприклад, структурний підхід [23] передбачає побудову таких систем шляхом моделювання роботи людського мозку за допомогою нейронних мереж [4]. Останнім часом вони стають особливо популярними, позаяк знайшли широке застосування для вирішення різних завдань з багатьох областей знань, наприклад, інтелектуального аналізу даних, прогнозування бізнес-діяльності, маркетингу продукції тощо. Однак, для того, щоб нейронна мережа давала хороші результати, необхідно обробити значний обсяг вхідних даних для навчання мережі [26, 27]. Машинне навчання мережі, яке стосується процесу самостійного отримання знань інтелектуальною системою під час її роботи, є найбільш ресурсномістким завданням. Для вирішення цієї проблеми застосовують розпаралелювання процедури навчання мережі з використанням графічних процесорів [21, 54].

Об'єкт дослідження – навчання нейронної мережі для прогнозування попиту на пасажирські перевезення таксі.

Предмет дослідження – методи і засоби, що дають змогу пришвидшити процедуру навчання нейронної мережі для прогнозування попиту на пасажирські перевезення таксі за різних умов зовнішнього середовища,

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. Прогнозування попиту на пасажирські перевезення таксі методами нейронної мережі. Науковий вісник НЛТУ України. 2021, т. 31, № 3. С. 109–119.
Citation APA: Zghoba, M. I., & Hrytsiuk, Yu. I. (2021). Forecasting the demand for passenger taxi transport by neural network methods. *Scientific Bulletin of UNFU, 31*(3), 109–119. https://doi.org/10.36930/40310317

конфігурації апаратного забезпечення та його потужності.

Мета роботи – визначити особливості прогнозування попиту на пасажирські перевезення таксі методами нейронної мережі, що має архітектуру багатошарового персептрону, та швидкість її навчання за допомогою графічного процесора порівняно з традиційним підходом.

Для досягнення зазначеної мети визначено такі *основні завдання дослідження*:

- проаналізувати останні дослідження та публікацій, в яких наведено особливості передбачення попиту на пасажирські перевезення таксі;
- визначити дані для навчання мережі, встановити її архітектуру, функцію втрат і метод оптимізації процедури навчання;
- розробити архітектуру нейронної мереж, проаналізувати особливості її навчання на звичайному графічному процесорах з використанням паралельних обчислень;
- здійснити порівняння тривалості навчання нейронної мережі на одному та двох графічних процесорах, а також навчання на центральному процесорі;
- обговорити результати дослідження та зробити висновки про ефективність проведеного дослідження.

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

Практична значущість результатів дослідження полягає у тому, що запропонована архітектура багатошарової нейронної мережі дає змогу значно точніше прогнозувати попит на пасажирські перевезення таксі за різних умов завантаженості транспортної мережі та її інфраструктури, погодних умов, святкових і вихідних днів, пори року та періоду доби.

Аналіз останніх досліджень та публікацій. Сучасні підходи до вирішення завдань прогнозування попиту на надання послуг знаходять широке застосування у повсякденній роботі диспетчерів різних компаній на пасажирські перевезення таксі [12, 43]. Передбачення місця перебування майбутніх пасажирів залежно від багатьох обставин, транспортна логістики різних населених пунктів – далеко не повний перелік питань, які доводиться враховувати диспетчеру перевезень таксі для прийняття рішень про їхнє чергове призначення за викликом [17, 51]. Однак, існують певні закономірності, які є складними для розуміння та вирішення людиною-диспетчером, але їх можна формалізувати, сформулювати у вигляді певних постановок задач і з достатньою точністю розв'язати за допомогою методів штучного інтелекту, наприклад, методами нейронних мереж [13] і їхнього машинного навчання [36].

У роботі [12] було запропоновано інформаційну систему, яка аналізує дані, зібрані з міської транспортної мережі. На підставі отриманих знань проводить прогнозування попиту на пасажирські перевезення таксі, використовуючи інформацію про тривалість їхнього перебування в дорозі та швидкість переміщення відповідно до стану дорожнього руху. Запропонована в роботі інформаційна модель збалансовує кількість поїздок таксі та пропонує водієві місця для паркування на підставі попередніх його платежів та наявних вільних місць. Робота системи спрямована на скорочення відстані до клієнтів і мінімізацію тривалості очікування водіями таксі замовлень, а потенційними клієнтами – їхнього прибуття. Результати оброблення статистичних даних показали, що за незмінної кількості таксі середнє співвідношення попиту на перевезення пасажирів і пропозиції таксі зменшилися на 31,7 %, а середній загальний їхній пробіг без пасажирів скоротився на 10,13 %.

У роботі [40] було визначено проблему ідентифікації сусідства таксі за великої кількості різних транспортних об'єктів. Автори досліджували тривалість очікування водіями таксі пасажирів у аеропортах залежно від періоду доби, погоди, прибуття авіарейсів, потребою пасажирів на перевезення. Традиційно, раніше диспетчери таксі постійно контролювали авіарейси і давали вказівки водіям таксі забрати потенційних клієнтів з відповідних терміналів. Тепер, намагаючись підтримувати черговість попиту на таксі, диспетчери почали застосовувати людино-машинні системи, внаслідок чого ефективність обслуговування пасажирів покращилась на 16,8 %.

У роботі [51] було запропоновано інформаційну систему, в якій використано нейронні мережі для прогнозування попиту на пасажирські перевезення таксі на підставі історичних даних, отриманих за допомогою GPS. Вони використовували мережу змішаної щільності (MDN) разом з довготривалою пам'яттю (LSTM) для зберігання попередніх значень місця перебування таксі. Така інтелектуальна система дає змогу досягти 83 % точності прогнозування попиту на таксі.

У роботі [53] було запропоновано інформаційну систему, у якій 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/паркування) та запровадили систему міських рекомендацій як для пасажирів, так і для водіїв таксі стосовно місця їх перебування. Також було вдосконалено рекомендації водіям таксі щодо часу та величини черги на місцях паркування, а також день тижня та наявні погодні умови.

У роботі [34] було розроблено метод прогнозування попиту потенційних пасажирів на таксі на найближчий 30-хвилинний період. Метод враховує транспортні подій за GPS-навігатором у реальному часі, передані в систему з 4,5 сотні таксі, яка визначає найбільший їхній попит. Автори порівняли прогнозовані дані з фактичним попитом на 63 таксі в місті Порто, внаслідок чого випробовувана модель досягла точності понад 74 %.

У роботі [35] було реалізовано подібний підхід до прогнозування попиту на пасажирські перевезення таксі для різних районів міста, враховуючи методику часових рядів. Вони згрупували наявні дані про 25 основних районів Токіо через кожні чотири години та передбачили попит на таксі у кожному районі міста на найближчий період часу. Науковцям вдалося прогнозувати попит на таксі з похибкою в межах 6-24 % для різних періодів доби та районів міста [18]. Вони також врахували кількість опадів як булевий вхідний параметр – наявність чи відсутність дощу. Проте, цей параметр не мав статистичного значення, позаяк не враховував наявність постійних опадів або той факт, коли кількість опадів досягає певного порогу.

У роботі [17] було зроблено подібне прогнозування попиту на пасажирські перевезення таксі для Нью-Йор-

ка. Поділивши місто на багато квадратів, автори намагались передбачити кількість поїздок таксі цими квадратами за годину. Вони використали 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 втрат, що враховують райони міста, період доби, день тижня та погодинну кількість опадів. Як і в роботі [35], вони вважають, що кількість опадів не є статистично значущим параметром.

У роботі [43] було досліджено методи прогнозування попиту на пасажирські перевезення таксі в містах США, використовуючи для цього різноманітні параметри: кількість населення та рівень його зайнятості, громадський транспорт, рівень туризму та заходи, пов'язані з великими подіями, кількість ділових відвідувачів, частку населення з низьким рівнем доходу та володіння транспортними засобами. Внаслідок проведеного дослідження було встановлено, що існують надзвичайно важливі параметри, здатні передбачати попит: кількість власних автомобілів, активність в аеропорту та поїздки в метро.

У роботі [37] було досліджено метод гібридного паралелізму, що є більш ефективним засобом зменшення загальної тривалості процедури навчання нейронної мережі порівняно з послідовним надходженням даних чи обробленням їх у моделі системи. Водночас, у роботі [29] було запроваджено гібридний підхід до навчання згорткових нейронних мереж, який поєднує паралельне надходження даних у обчислювальні частини моделі (згорткові шари) разом із паралельним обробленням даних у моделі для шарів з великою кількістю параметрів (повністю зв'язані шари). Цей підхід дає змогу масштабувати нейронні мережі значно краще, ніж усі альтернативи сучасних конволюційних мереж [30].

У роботі [1] запропоновано нейро-фаззі мережу, призначену для вирішення завдань екстраполяції багатовимірних нестаціонарних стохастичних і хаотичних часових рядів за умов короткої навчальної вибірки. В основу мережі закладено багатовимірний нео-фаззі-нейрон із спеціально організованим вхідним шаром і сплайн-функціями належності. Розроблена інформаційна система забезпечує високу точність апроксимації щодо середньоквадратичної похибки та значну швидкість збіжності за рахунок використання процедури навчання другого порядку. Розроблене ПЗ, що реалізує запропоновану архітектуру нейро-фаззі мережі, дає змогу провести експерименти з дослідження її властивостей. Результати проведених експериментів підтвердили придатність такої архітектури мережі до розв'язання задач Data Mining та більш високу точність прогнозування порівняно з традиційними нейро-фаззі системами.

Необхідно зазначити, що під час моделювання часових рядів часто використовують нелінійну авторегресійну екзогенну модель NARX (англ. *Nonlinear Autoregressive Exogenous Model*), яка має екзогенні входи [3]. Це означає, що така модель ставить поточне значення часового ряду у відповідність до минулих значень того самого ряду, та поточних і минулих значень приводного (екзогенного) ряду – тобто, зовнішньо визначеного ряду, який впливає на цільовий ряд. Також ця модель містить член "похибки", який відповідає тому фактові, що знання інших членів не дає можливості передбачувати поточне значення часового ряду точно.

Отже, серед безлічі підходів, що використовують для вирішення завдань прогнозування попиту на пасажирські перевезення таксі, особливо ефективними показали себе штучні нейронні мережі та нейро-фаззі мережі [48], завдяки своїм універсальним апроксимаційним і екстраполяційним можливостям і здатності навчатися в умовах істотної структурної та параметричної невизначеності про характеристики прогнозованих процесів. При цьому, основу таких інформаційно-аналітичних систем становлять мережі з прямою передачею інформації та елементами затримки у вхідних шарах. Реалізують їх нелінійні моделі авторегресії з екзогенними входами (NARX-моделі), які є окремим випадком більш загальних структур, що містять компоненти ковзкого середнього, мають значну гнучкість та більш високу точністю прогнозування. NARMAX-моделі достатньо прості в реалізації на підставі рекурентних нейронних мереж [33], які з обчислювальної точки зору набагато ефективніші, ніж мережі з прямою передачею інформації [15, 20, 49].

Часовий ряд (англ. *Time Series*) – це ряд точок даних, проіндексованих (або перелічених, або відкладених на графіку) в хронологічному порядку. Найчастіше часовий ряд є послідовністю, взятою на рівновіддалених точках у часі, які йдуть одна за іншою. Отже, він є послідовністю даних дискретного часу. Аналіз часових рядів (англ. *Time Series Analysis*) містить відповідні методи аналізу даних, призначених для видобування значущих статистик та інших характеристик даних. Методи аналізу часових рядів розділено на лінійні й нелінійні, одновимірні й багатовимірні. Прогнозування часових рядів (англ. *Time Series Forecasting*) – це застосування моделі для передбачення майбутніх значень на підставі значень попередньо спостережених. Зібрані дані щодо місця перебування потенційних пасажирів на таксі, а також їхнього місцезнаходження – одне з наглядних прикладів часових рядів, вирішення яких дасть змогу ефективно і точно прогнозувати попит на пасажирські перевезення таксі.

В задачах оброблення нелінійних часових рядів найбільшого поширення набули три типи зворотних нейронних мереж: мережі Вільямса-Зіпсера [50], Елмана [11] і Джордана [24]. Для вирішення завдань аналізу та прогнозування часових рядів, а особливо – виявлення змін їх властивостей ці мережі вимагають деякої модифікації, що стосується, насамперед, алгоритмів машинного навчання. Причина в тому, що всі зазначені нейронні мережі можна навчити в пакетному режимі, позаяк вони не передбачають ситуацію, коли дані спостережень часового ряду надходять на оброблення послідовно одне за іншим. Водночас, в завданні прогнозування попиту на пасажирські перевезення таксі в реальному часі дані в інформаційну систему надходять послідовно і безперервно протягом доби.

В роботі [22] була наведена архітектура рекурентної штучної нейронної мережі, що є своєрідним гібридом багатошарового персептрона і рекурентної мережі Вільямса-Зіпсера, призначеної для оброблення нелінійних часових рядів у режимі послідовного оброблення даних реального часу. Ця мережа має специфічну архітектуру з одним лінійним вихідним нейроном і одним додатковим лінійним нейроном (адаптивним лінійним

асоціатором) в першому прихованому шарі. Головною ж особливістю цієї мережі є відсутність контекстного шару зворотного зв'язку, замість якого між прихованим і вихідним шарами встановлені елементи чистого запізнювання. Дана мережа реалізує прогнозну NARMAX-модель з числовими екзогенними входами, її можна навчити за допомогою алгоритму градієнтного спуску з постійним кроком, налаштовуючи свої ваги нейронів мережі в міру надходження нових даних.

До основних недоліків цієї мережі в завданні прогнозування попиту на пасажирські перевезення таксі належить неможливість оброблення інформації, заданої в нечисловій формі, яка надходить на оброблення в реальному часі. Тому в цій роботі спробуємо синтезувати гібридну нейро-фаззі систему, яка має об'єднати рекурентну нейронну мережу [22] з її необхідною гнучкістю та простотою реалізації, а також багатошарову нейро-фаззі мережу [4] зі значною швидкістю навчання та можливістю оброблення інформації, заданої як у числовій формі, так і в інших шкалах, наприклад, порядковій чи номінальних.

Результати дослідження та їх обговорення

Передбачення попиту на пасажирські перевезення таксі. Як і будь-яка компанія-перевізник, так і водії таксі прагнуть зменшити тривалість очікування нових замовлень та відстань до клієнтів на момент їх надходження 2. Зазвичай, водії самостійно вирішують, де чекати пасажирів так, щоб вони могли їх швидко забрати. Природно, що багато з них не завжди знають достовірно, де будуть знаходитися потенційні пасажири, однак досвідчені водії можуть здогадуватися та прогнозувати такі місця на підставі свого попереднього досвіду. Диспетчерська система відправки таксі ефективно допомагає як клієнтам, так і водіям значно скоротити тривалість очікування таксі й замовлень їхнім водіям [17, 43]. Проте, ця служба не зажди володіє достовірною інформацією про місця скупчення потенційних клієнтів і місця перебування таксі – на стоянці чи в дорозі. Тому диспетчер таксі-центру не в стані оперативно організувати та надіслати необхідну кількість таксі до місць перебування пасажирів для їх перевезення [2, 12, 51].

Відомо [38], що попит на пасажирські перевезення таксі може мінятися під впливом різних чинників, а часто проблеми виникають й через відсутність місць паркування. Нерідко у певних районах, особливо ближче до центральної частини міста, не вистачає вільних місць на парковці. А якщо вони й трапляються, то там стягують плату за стоянку, тобто очікування наступного замовлення. Також бувають випадки, коли нові замовлення клієнтів знаходяться дуже далеко від місця стоянки таксі.

Отже, аби мінімізувати тривалість пасажирських перевезень таксі, потрібно мати розуміння транспортної логістики певної місцевості та вміння оцінити можливий попит таксі залежно від погоди, району міста, пори дня, святкових чи вихідних днів, культурних заходів, наявності транспортних об'єктів (аеропорти, вокзали) і т.п.

Дані для навчання мережі. Для навчання нейронної мережі у нашому дослідженні було використано набір вхідних даних, що містить 4,5 млн поїздок таксі компанії Uber в межах Нью-Йорка за період з квітня по жовтень 2019 року [14]. Для більшої точності передбачень зібрано та додано до набору даних погодинні метеорологічні спостереження за цей самий період, а саме – температуру повітря та наявність семи видів атмосферних явищ [6].

Сформований набір вхідних даних 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 [46]. Широта та довгота точки – це атрибути, які нейронна мережа повинна передбачати для визначення місцезнаходження пасажира у заданий час та наявну погоду [16] і [41].

Для навчання нейронної мережі нами використано оптимізовану тензорну бібліотеку з відкритим кодом для глибинного навчання з використанням графічних процесорів і процесорів PyTorch [39]. Вона забезпечує максимальну гнучкість та швидкість виконання обчислень, що дає змогу швидко розробити потрібну кодову базу. Хороша документація та простий синтаксис дає можливість розпаралелити процедуру навчання, тобто можна розподіляти паралельні розрахунки між декількома ядрами звичайного або графічного процесора [8].

Багатошаровий перцептрон. Для прогнозування попиту на пасажирські перевезення таксі нами використано *багатошаровий персептрон* (англ. *Multilayer Perceptron*) як один з видів нейронних мереж (рис. 1). Це різновид мережі прямого зв'язку з одним або декількома рівнями внутрішніх прихованих шарів між вхідним і вихідним шарами. Вхідний сигнал у такій мережі поширюється від шару до шару. Вихідні одиниці представляють гіперплощину в просторі шаблонів введення даних. Мережа містить M шарів, кожен з яких складається з $J_m, m=\overline{1,M}$ вузлів. Ваги нейронів мережі від (m–1)-го до m-го шару позначають w^{m-1}. Функції зміщення, виходу та активації i-го нейрона в m-му шарі, відповідно, позначають як $q_i^{(m)}$, $q_i^{(m)}$ та $\psi_i^{(m)}(\cdot)$ [10]. Кожна нейронна мережа має нелінійну функцію активації.

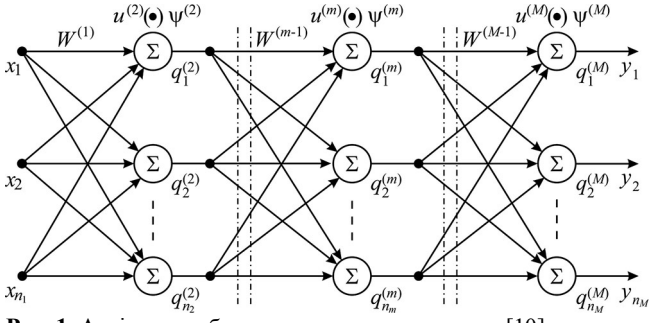

Рис. 1. Архітектура багатошарового персептрону [10]

Стохастичний градієнтний спуск. Найпоширенішою технікою для обчислення та оновлення ваг нейронів мережі є *стохастичний градієнтний спуск* SGD (англ. *Stochastic gradient descent*), яка виконує ітеративно такі кроки [31]. Спочатку обчислюють крок передачі вперед, де вхідні вибірки даних обробляють шар за шаром, поки не буде отримано прогноз після останнього шару. На наступному кроці (зворотному розповсюдженні) ваги нейронів мережі оновлюють на підставі обчисленої різниці (градієнтів) між прогнозованими та позначеними результатами. Ці кроки виконують ітеративно для всіх міні-пакетів (англ. *Mini-batch*) у наборі даних. Як тільки всі міні-пакети опрацьовано, то одну епоху розрахунку вважають завершеною. Це означає, що весь набір вхідних даних передавався вперед і назад

через неї тільки один раз. Процес розрахунку продовжують протягом декількох наступних епох. Оскільки метод SGD прагне мінімізувати функцію втрати, знаходячи ваги нейронів мережі, які мають відповідати цілому набору даних, то процедура розрахунку на кожному кроці апроксимує дані для всієї навчальної вибірки [32].

У роботі [28] зазначено, що для оптимізації процедури навчання нейронної мережі набуває популярності метод SGD Adam, який використовує квадратичні градієнти для масштабування швидкості навчання та його імпульсу, застосовуючи для цього ковзну середню градієнта замість самого градієнта. Також цей метод використовує оцінки першого та другого моментів градієнта, щоб адаптувати швидкість навчання для кожної ваги нейронів мережі. Метод SGD Adam добре підходить для машинного навчання багатошарових нейронних мереж з великими обсягами даних, насамперед, завдяки організації паралельних розрахунків.

Функції втрати. Нейронну мережу можна навчати, використовуючи для цього попередні значення вхідних і результуючих даних для оновлення ваг нейронів мережі, отримуючи при цьому правильний вихід. Така процедура навчання пов'язана з потребою її оптимізації, де помилка навчання має бути мінімальною. Для мінімізації помилки та правильного навчання мережі цю помилку (її значення) потрібно встановити і визначити, чи відповідає вона функції втрат.

Функція втрати (англ. *Loss function*) описує, наскільки далеко знаходиться модель мережі від здійснення ідеальних прогнозів для заданих даних [42]. Результатом роботи функції втрати є відносне значення помилки навчання мережі. Якщо прогнози стають кращими, 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 як суму квадратів відстаней між цільовою змінною та передбаченими значеннями, поділених на їх кількість, а саме

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

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

Архітектура нейронної мережі. Для прогнозування попиту на пасажирські перевезення таксі нами вибрано рекурентну нейро-фаззі мережу, запропоновану і описану в роботі [1]. Мережа має три шари – перший прихований шар фазифікації, другий прихований шар, вихідний шар (рис. 2). Її складовими є стандартні нейрони – елементарні персептрони Розенблата з сигмоїдально активаційними функціями втрат, адаптивним лінійним асоціатором, елементом затримки z^{-1} і блоками фазифікації, призначеними для перетворення вхідних змінних, що характеризують вплив навколишнього середовища, в кількісну форму рівнів належності, розміщеними в інтервалі [0, 1]. Блоки фазифікації, що використовують трикутні функції належності й сінглтон, формують перший прихований шар мережі, який повністю співпадає за структурою з першим прихованим шаром прогнозної нейро-фаззі мережі [4]. Нагадаємо, сінглтон (англ. *Singleton* – одинак) – породжувальний шаблон проектування, який гарантує, що об'єкт класу матиме тільки один екземпляр та надає глобальну точку доступу до нього.

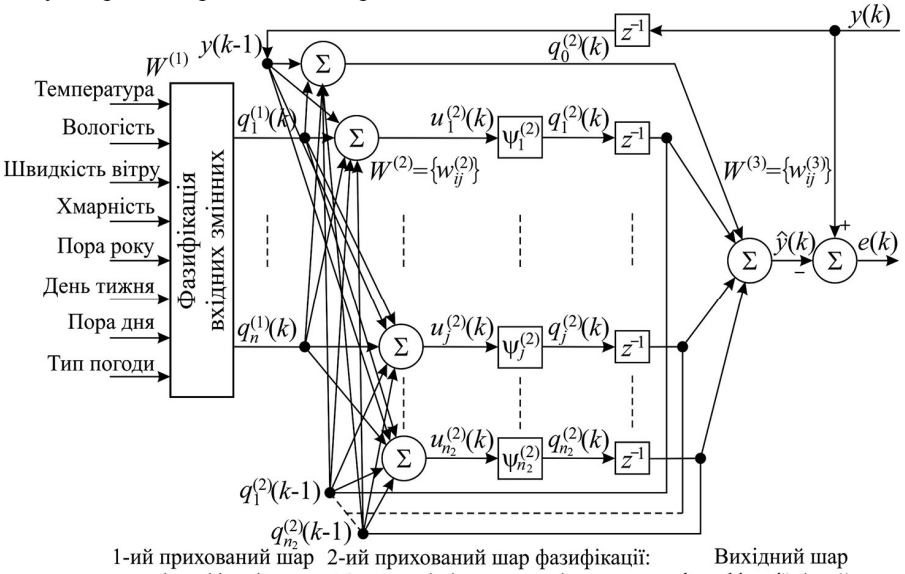

Рис. 2. Прогнозна рекурентна нейро-фаззі мережа

Сигнали першого прихованого шару у вигляді $(n\times 1)$-вектора

$$q^{(1)}(k) = \left[q_i^{(1)},\ i = \overline{1,n} \right]^{\mathrm{T}},\ k = 0,1,2,...,K\ , \qquad (2)$$

з компонентами, що описують постійні та змінні характеристики в числовій формі рівнів належності, надходять на другий прихований шар, утворений одним адаптивним лінійним асоціатором з додатковим входом затриманого значення прогнозованого сигналу $y(k–1)$ (де k – поточний дискретний час), в кількості n_2 ідентичними нейронами з сигмоїдальними активаційними функціями втрат $\psi_j^{(2)},\ j = \overline{1,n}$ і синаптичними вагами $w_{ij}^{(2)}$ нейронів мережі. Виходи другого прихованого шару об'єднують в $((n_2 + 1)\times 1)$-вектор, а саме

$$q^{(2)}(k) = \left[q_j^{(2)}(k),\ j = \overline{0, n+n_2} \right]^{\mathrm{T}} = \left[q_0^{(2)}(k),\ q_\psi^{(2)\mathrm{T}}(k) \right]^{\mathrm{T}} . \qquad (3)$$

Між виходами елементарних персептронів Розенблата другого прихованого шару і входами вихідного шару, утвореного єдиним адаптивним лінійним асоціатором з синаптичними вагами $w_{ij}^{(3)}$ нейронів мережі, встановлено n_2 елементів затримки z^{-1} так, що вектор затриманих сигналів $q_{\psi}^{(2)}(k-1)$ подають одночасно і на входи вихідного шару, і каналами зворотного зв'язку на входи другого прихованого шару. Отже, кожен нейрон другого прихованого шару має $n + n_2 + 1$ входів (і відповідно $n + n_2 + 1$ синаптичних ваг $w_{ij}^{(2)}$, а нейрон вихідного шару має $n_2 + 1$ входів і стільки ж синаптичних ваг нейронів мережі.

З огляду на зазначене вище, відповідні перетворення, що реалізує архітектура нейро-фаззі мережі (див. рис. 2), можна записати у такому вигляді:

$$\hat{y}(k) = w_0^{(3)}(k) q_0^{(2)}(k) + \sum_{i=1}^{n_2} w_i^{(3)}(k) q_i^{(2)}(k-1); \quad (4)$$

$$q_0^{(2)}(k) = u_0^{(2)}(k); \quad (5)$$

$$q_j^{(2)}(k) = \psi_j^{(2)}\left(u_j^{(2)}(k)\right),\ j = \overline{1, n_2}; \quad (6)$$

$$u_j^{(2)}(k) = \sum_{i=0}^{n+n_2} w_{ij}^{(2)}(k) \tilde{q}_i^{(1)}(k),\ j = \overline{1, n_2}; \quad (7)$$

$$\tilde{q}_i^{(1)}(k) = \begin{cases} y(k-1), \text{ якщо } i = 0; \\ q_i^{(1)}(k), \text{ якщо } 1 \le i \le n; \\ q_{i-n}^{(2)}(k-1), \text{ якщо } n+1 \le i \le n+n_2, \end{cases} \quad i = \overline{0, n+n_2}. \quad (8)$$

Увівши далі два вектори:

$$\tilde{q}^{(2)}(k) = \left[q_j^{(2)}(k),\ j = \overline{0, n+n_2} \right]^{\mathrm{T}};\ w^{(3)}(k) = \left[w_j^{(3)}(k),\ j = \overline{0, n_2} \right]^{\mathrm{T}} \quad (9)$$

розміром $(n_2 + 1)\times 1$;

$$\tilde{q}^{(1)}(k) = \left[y(k-1);\ q_j^{(1)}(k),\ j = \overline{1, n};\ q_j^{(2)}(k-1),\ j = \overline{1, n_2} \right]^{\mathrm{T}} \quad (10)$$

розміром $(n + n_2 + 1)\times 1$; матриця синаптичних ваг нейронів другого прихованого шару мережі $W^{(2)}$ розміром $(n_2 + 1)\times(n_2 + 1)$ і матриця активаційних функцій

$$W^{(2)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & w_{11}^{(2)} & \cdots & w_{1,n_2}^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & w_{n_2,1}^{(2)} & \cdots & w_{n_2,n_2}^{(2)} \end{pmatrix},\ \Psi^{(1)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \psi_1^{(2)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \psi_{n_2}^{(2)} \end{pmatrix} \quad (11)$$

розміром $(n_2 + 1)\times(n_2 + 1)$, можна переписати співвідношення (4)–(8) в такому компактному вигляді

$$\hat{y}(k) = w^{(3)\mathrm{T}}(k)\tilde{q}^{(2)}(k) = w^{(3)\mathrm{T}}(k)\Psi^{(2)}\left(W^{(2)}(k)\tilde{q}^{(1)}(k)\right). \quad (12)$$

У процесі навчання нейро-фаззі мережі необхідно визначити оптимальні значення параметрів $w_{ij}^{(3)}$ і $W^{(2)}$, а також моменти різкої зміни властивостей прогнозованого процесу, які можуть проявитися в несподіваних "скачках" синаптичних ваг нейронів мережі.

Навчання рекурентної нейро-фаззі мережі. Навчання мережі (див. рис. 2) будемо проводити шляхом покрокової мінімізації значення стандартного середньоквадратичного критерію (1) в такому вигляді

$$E(k) = \frac{1}{2}e^2(k) = \frac{1}{2}\left(y(k) - \hat{y}(k)\right)^2 = \\ = \frac{1}{2}\left(y(k) - w_0^{(3)}(k) q_0^{(2)}(k) - \sum_{i=1}^{n_2} w_i^{(3)}(k) q_i^{(2)}(k-1) \right)^2 \quad (13)$$

за допомогою градієнтної процедури оптимізації з урахуванням таких співвідношень:

$$\frac{\partial E(k)}{\partial w_{ij}^{(s)}(k)} = \frac{\partial}{\partial w_{ij}^{(s)}(k)} \frac{1}{2} e^2(k) = -e(k) \frac{\partial \hat{y}(k)}{\partial w_{ij}^{(s)}(k)},\ s = 2,3; \quad (14)$$

$$w_{ij}^{(s)}(k+1) = w_{ij}^{(s)}(k) - \eta \frac{\partial E(k)}{\partial w_{ij}^{(s)}(k)} = w_{ij}^{(s)}(k) + \eta(k) e(k) \frac{\partial \hat{y}(k)}{\partial w_{ij}^{(s)}(k)},\ s = 2,3, \quad (15)$$

де $\eta(k)$ – параметр кроку пошуку оптимального значення, який обирають з емпіричних міркувань і досить часто вважають, що він є постійною величиною.

Спочатку розглянемо процес налаштування ваг вихідного нейрона мережі. З рівнянь (14) і (15) отримаємо таке співвідношення

$$w_i^{(3)}(k+1) = w_i^{(3)}(k) + \eta(k) e(k) \frac{\partial \hat{y}(k)}{\partial w_i^{(3)}(k)},\ i = \overline{0, n_2}, \quad (16)$$

де $\dfrac{\partial \hat{y}(k)}{\partial w_i^{(3)}(k)} = \dfrac{\partial}{\partial w_i^{(3)}(k)} \left(w_0^{(3)}(k) q_0^{(2)}(k) + \sum_{l=1}^{n_2} w_l^{(3)}(k) q_l^{(2)}(k-1) \right),\ i = \overline{0, n_2}.$ (17)

Для налаштування синаптичних ваг другого прихованого шару мережі використаємо стандартну процедуру зворотного поширення помилки розрахунку в такій рекурентній формі [5]:

$$w_i^{(2)}(k+1) = w_i^{(2)}(k) + \eta(k) \delta_j^{(2)}(k) \tilde{q}_i^{(1)}(k),\ j = \overline{0, n_2};\ i = \overline{0, n+n_2}, \quad (18)$$

де $\delta_j^{(2)}(k)$ – локальна помилка розрахунку другого прихованого шару, яку опишемо таким виразом

$$\delta_j^{(2)}(k) = \frac{\partial \psi_j^{(2)}\left(u_j^{(2)}(k)\right)}{\partial u_j^{(2)}(k)} e(k) w_j^{(3)}(k),\ j = \overline{0, n_2}. \quad (19)$$

Отже, з врахуванням (19) остаточно процес налаштування синаптичних ваг нейронів другого прихованого шару мережі (18) опишемо таким рекурентним співвідношенням

$$w_{ij}^{(2)}(k+1) = w_{ij}^{(2)}(k) + \eta(k) \frac{\partial \psi_j^{(2)}\left(u_j^{(2)}(k)\right)}{\partial u_j^{(2)}(k)} e(k) w_j^{(3)}(k) \tilde{q}_i^{(1)}(k), \\ j = \overline{0, n_2};\ i = \overline{0, n+n_2}. \quad (20)$$

Швидкість збіжності процесів (16) і (20) повністю встановлює вибране значення кроку пошуку $\eta(k)$ і цю швидкість можна істотно збільшити шляхом спеціального підбору цього кроку [4]. Завдяки цьому можна розглядати спільне завдання прогнозування та діагностики рекурентної нейро-фаззі мережі, що полягає у виявленні різких змін властивостей обчислювального процесу.

У зв'язку з цим для навчання ваг вихідного шару мережі доцільно використовувати процедуру, що володіє як фільтруючими (згладжування збурень і перешкод), так і відстежувальними (виявлення стрибків) властивостями рекурентної нейро-фаззі мережі. Оскільки вихідний сигнал мережі лінійно залежить від синаптичних ваг вихідного нейрона, то для їх налаштування можна використати експоненціально зважений рекурентний метод найменших квадратів у такому вигляді

$$\begin{cases} w_i^{(3)}(k+1) = w_i^{(3)}(k) + \dfrac{P(k) \cdot e(k) \cdot \tilde{q}^{(2)}(k)}{\alpha + \tilde{q}^{(2)\mathrm{T}}(k) \cdot P(k) \cdot \tilde{q}^{(2)}(k)}; \\ P(k+1) = \dfrac{1}{\alpha}\left(P(k) - \dfrac{P(k) \cdot \tilde{q}^{(2)}(k) \cdot \tilde{q}^{(2)\mathrm{T}}(k) \cdot P(k)}{\alpha + \tilde{q}^{(2)\mathrm{T}}(k) \cdot P(k) \cdot \tilde{q}^{(2)}(k)} \right), \end{cases} \quad i = \overline{0, n_2}. \quad (21)$$

де $0 < \alpha < 1$ – параметр "забування" застарілої інформації, що визначає компроміс між фільтруючими і відстежувальними властивостями мережі. Чим більше значен-

ня α, тим інерційніший процес навчання, а його менше значення вказує на швидке виникнення реакції на можливі зміни.

Для виявлення цих змін часто використовують алгоритм Хегглунда [19] в такій формі

$$\begin{cases} \theta(k+1) = \eta_\theta \theta(k) + w^{(3)}(k+1) - w^{(3)}(k); \\ \mu(k+1) = sign\left(\theta^{\mathrm{T}}(k+1)\left(w^{(3)}(k+1) - w^{(3)}(k)\right)\right), \end{cases} \quad (22)$$

де $0 \le \eta_\theta < 1$. Якщо діагностуючий сигнал $\mu(k+1)$ протягом декількох підряд кроків пошуку набуває значення +1, то в контрольованому сигналі $y(k)$ виникли різкі зміни.

Отже, запропонована нейро-фаззі мережа забезпечує не тільки короткотермінове прогнозування попиту на пасажирські перевезення таксі, але й здійснює контроль за різкими його змінами, що надзвичайно важливо для ефективної роботи міських диспетчерських систем.

Навчання нейронної мережі на графічному процесорі. Для будь-якої нейронної мережі стадія її навчання є найбільш ресурсномістким завданням. Якщо архітектура мережі має приблизно 10, 100 або навіть 100 000 параметрів, то звичайний комп'ютер зможе впоратися з цим завданням за лічені хвилини. Проте, наявність в нейронній мережі декількох мільйонів параметрів призведе до того, що її традиційне навчання буде проходити тижні чи навіть місяці.

Глибинне машинне навчання (англ. *Deep Machine Learning*) – техніка оброблення даних різними алгоритмами, які дають змогу моделювати високорівневі абстракції в даних, застосовуючи для цього глибинний граф із декількома обробними шарами, а також лінійні чи нелінійні перетворення. Особливістю глибинного навчання є заміна ознак ручної роботи дієвими алгоритмами автоматичного або напівавтоматичного навчання ознак та ієрархічного їх виділення [47].

Деякі дослідження [28] показали, як можна застосувати техніку глибинного навчання для рекурентних нейронних мереж, передусім за допомогою звичайного алгоритму зворотного поширення помилки. Існує велика кількість модифікацій такого алгоритму, в яких використовують декілька правил налаштування ваг нейронів мережі. Наприклад, для навчання вагових коефіцієнтів $\omega_{ij}(k)$ використовують алгоритм стохастичного градієнтного спуску:

$$\omega_{ij}(k+1) = \omega_{ij}(k) + \eta \frac{\partial C}{\partial \omega_{ij}}, j = \overline{1,m}; i = \overline{1,n}, \quad (23)$$

де: η – стала для регулювання величини поточного кроку пошуку; C – функція втрат. Вибір функції втрат обумовлений класом завдання машинного навчання (з учителем, без учителя, з підкріпленням) і функцією активації. До основної проблеми глибинного навчання рекурентних нейронних мереж належить тривалість процедури навчання та перенавчання.

Багатошарові структури нейронних мереж більше схильні до перенавчання, оскільки значна кількість шарів дає змогу моделювати високорівневі абстракції в даних, тому модель мережі може "вивчити" рідкісні ситуації, зайві у її подальшій роботі. У цьому випадку застосовують різні методи регулювання процедури навчання мережі. Один з методів регулювання припускає вилучення випадкових вузлів нейронної мережі під час навчання. У деяких випадках це допомагає значно менше запам'ятовувати рідкісні ситуації в тренувальних наборах даних.

Через простоту реалізації та хорошу збіжність процедури глибинного навчання рекурентних нейронних мереж для "правильного" їх навчання часто використовують метод зворотного поширення помилки і градієнтний спуск. Однак, при навчанні прихованих шарів нейронів мережі виникає декілька проблем, які особливо важливі при оптимізації функції втрат у просторі великої розмірності: кількість обчислювальних елементів, початкові умови для ваг нейронів мережі, а також константа регулювання величини кроку пошуку оптимальних значень.

Окрім цього, алгоритм стохастичного градієнтного спуску відомий проблемою зникаючого градієнта (англ. *Vanishing Gradient*), яка полягає в ослабленні градієнта, а значить і зменшенні швидкості процедури навчання мережі в міру поглиблення від останніх її шарів до початку мережі. Через це приховані шари нейронної мережі погано навчаються. Проте, деякі науковці [34, 37] замість сигмоїдальної активаційної функції вузла мережі в багатьох нейронних мережах пропонують використовувати різні види нелінійності ReLU (англ. *Rectified Linear Unit*), функція активації якої має вигляд $\max(0, x)$. За такого підходу в рекурентних нейронних мережах практично відсутні проблеми ослаблення градієнта, тому вони добре навчаються методом градієнтного спуску.

Глибинне навчання рекурентних нейронних мереж може проходити значно швидше, якщо всі операції виконувати одночасно, а не одну за іншою. Застосування спеціалізованих графічних процесорів із виділеною пам'яттю, наприклад, GPU (англ. *Graphics Processing Unit*), дає змогу виконувати всі операції навчання одночасно, як це виконують для рендерингу графіки [9]. Такі процесори ефективно обробляють та відображують комп'ютерну графіку завдяки спеціалізованій конвеєрній архітектурі. Також вони оптимізовані для глибинного навчання моделей штучного інтелекту, оскільки здатні виконувати паралельні розрахунки.

Організація паралельних обчислень у нейронних мережах. Глибинне навчання нейронних мереж є важливою частиною їх підготовки до ефективної роботи. Зі збільшенням обсягу навчальних даних й складності архітектури мережі пропорційно зростає потреба в обчислювальній потужності процесора та пам'яті комп'ютера. Зменшення тривалості процедури навчання мережі сприяє підвищенню якості роботи моделі, даючи їй можливість здійснити подальші розрахунки вже на реальних наборах даних різної величини.

Нагадаємо, що алгоритми глибинного навчання перетворюють свої входи крізь більшу кількість шарів, ніж алгоритми поверхневого навчання. На кожному шарі сигнал перетворюють блоком оброблення (штучним нейроном), параметри якого "навчаються" шляхом тренування [44]. Ланцюг перетворень від входу до виходу є *шляхом передачі довіри* (ШПД, англ. *Credit Assignment Path*, CAP). ШПД описують потенційно причинні зв'язки між входом і виходом, і можуть мати змінну довжину. Для нейронної мережі прямого поширення довжина шляхів передачі довіри і, відтак глибина цієї мережі, є числом прихованих шарів плюс один (вихідний шар також параметризовано). Для рекурентних нейронних мереж, в яких сигнал може поширюватися через якийсь шар більше одного разу, ШПД має потенційно необмежену довжину. Універсально узгодженого порогу гли-

бини, що відділяв би поверхневе навчання від глибинного, не існує, але більшість дослідників у цій галузі погоджуються з тим, що глибинне навчання має декілька нелінійних шарів (ШПД > 2), а Шмідгубер розглядає ШПД > 10 як дуже глибинне навчання [44].

Паралельне глибинне навчання нейронних мереж має істотні переваги перед традиційним навчанням [7]. Часто буває, коли великі моделі штучного інтелекту не поміщаються в пам'яті одного звичайного процесора. Навчання різних частин моделі на графічних процесорах GPU – простий спосіб подолати обмеження пам'яті будь-якого процесора. Водночас, застосування паралельних навчальних стратегій приводять до дещо швидшого навчання нейронної мережі, розподіляючи перекриваючі обчислення між різними її вузлами.

Під час навчання нейронної мережі (див. рис. 2) шляхом організації паралельних обчислень нами було враховано такі дві її характеристики: витрати на зв'язок між різними вузлами мережі для синхронізації градієнтів і витрати часу для виконання розрахунків [25].

Обговорення результатів дослідження. Дослідження здійснено на центральному процесорі, одному графічному процесорі NVIDIA Tesla K80 та на двох таких процесорах відповідно. Навчання нейронної мережі проведено на двох розмірах вхідних даних batch_size = 16 та 2048 проб. Результати дослідження передбачають порівняння тривалості процедури навчання мережі для однієї епохи.

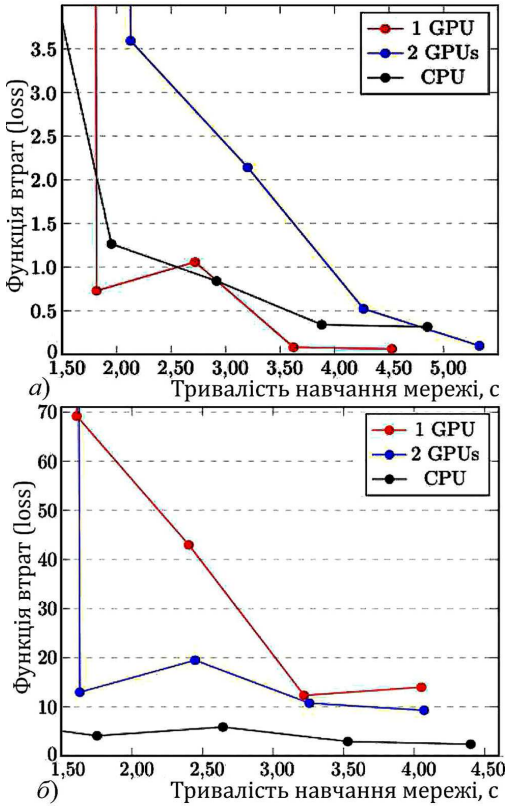

Рис. 3. Залежність тривалості процедури навчання нейронної мережі від кількості використаних ресурсів: *а*) mini-batch = 16 проб; *б*) mini-batch = 2048 проб

Дані, наведені на рис. 3,*а*, 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. За функцію втрати взято середньоквадратичну помилку (1).

Результати дослідження показали, що процедура навчання нейронної мережі на одному графічному процесорі (червона лінія) проходить дещо швидше від аналогічного навчання на центральному процесорі (чорна лінія) у 1.0715 разів. Таке незначне покращення результату пов'язане з малим розміром набору розпаралелених даних (mini-batch = 16 проб) та значними витратами часу для їх передачі між графічним і центральним процесором, який керує процедурою навчання.

Навчання нейронної мережі за допомогою двох графічних процесорів (синя лінія) погіршує тривалість цієї процедури порівняно з навчанням на одному графічному або центральному процесорі. Причина – значні витратами часу для синхронізації градієнтів між двома графічними процесорами перед кожним оновленням параметрів мережі (після кожної batch-ітерації). Витрати часу на синхронізацію є більшими від організації паралельних обчислень, оскільки розмір розпаралелених вхідних даних є занадто малим.

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

Дані, наведені на рис. 3,*б*, показують 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 розпаралелених вхідних даних у 128 разів більший за розмір вхідних даних, використаних для отримання результатів на рис. 3,*а*.

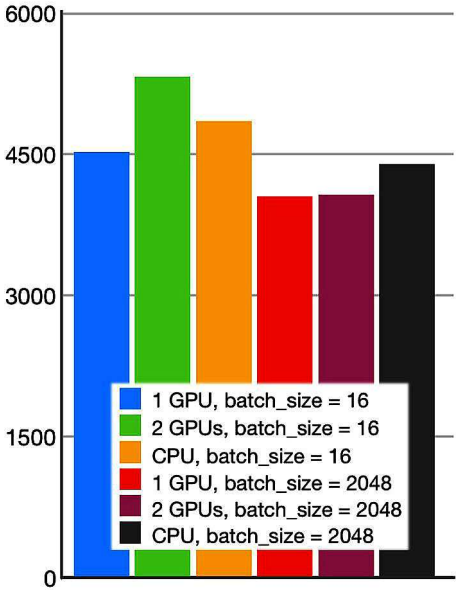

Рис. 4. Зведена залежність тривалості процедури навчання від кількості використаних ресурсів

Навчання нейронної мережі на одному (червона лінія) та двох (синя лінія) графічних процесорах проходить швидше у 1.0861 та 1.0811 рази відповідно, ніж навчання на одному центральному процесорі (чорна лінія). Порівняно з результатами для mini-batch розміру 16 проб, навчання на двох графічних процесорах проходить дещо швидше, ніж на одному центральному процесорі. Проте, немає значних успіхів між навчанням мережі на одному та двох графічних процесорах, оскільки користь від організації паралельних обчислень на двох процесорах втрачають для синхронізації градієнтів між

цими процесорами перед кожним оновленням нейронів мережі.

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

Незначна різниця між тривалістю навчання нейронної мережі для однієї епохи не залежить від обраних апаратних складових комп'ютерної техніки та архітектури мережі (див. рис. 2), яка містить велику кількість вузлів і відносно малу кількість обчислень, порівняно з іншими архітектурами. Наприклад, для згорткових нейронних мереж, де кількість обчислень відчутно більша, ефект від організації паралельних розрахунків на декількох графічних процесорах буде значно відчутнішим. Проте, витрати на синхронізацію градієнтів між графічними процесорами практично сталі. Отже, використання організації паралельних обчислень для реалізації процедури навчання нейронної мережі може дати значно кращі результати за умови обрання відповідної її архітектури.

Висновки

З'ясовано, що для мінімізації тривалості очікування нових замовлень та відстані до клієнтів на момент їх надходження доцільно використовувати відповідні інформаційно-аналітичні системи, робота яких ґрунтується на штучному інтелекті. Це дасть змогу вирішити проблему прогнозування попиту на пасажирські перевезення таксі як для їхньої кількості у відповідний період доби з врахуванням погодних умов, протягом тижня з врахуванням святкових, вихідних і робочих днів, так і протягом року залежно від його пори. Врахування ж наявних транспортних об'єктів залежно від прибуття авіарейсів, потягів чи рейсових автобусів значно покращують роботу інформаційної системи.

Використана в роботі архітектура нейро-фаззі мережі є узагальненням багатошарового персептрона, рекурентної мережі Вільямса-Зіпсера та прогнозної багатошарової нейро-фаззі мережі. Така гібридна архітектура мережі дає змогу одночасно вирішувати завдання короткотермінового прогнозування попиту на пасажирські перевезення таксі, а також проводити діагностику самої мережі, що полягає у виявленні різких змін властивостей обчислювального процесу.

Для досягнення значної точності у прогнозуванні попиту на пасажирські перевезення таксі в роботі опрацьовано великий набір вхідних даних у кількості 4,5 млн поїздок таксі. Для зменшення тривалості процедури навчання нейронної мережі організовано паралельні обчислення між різними вузлами мережі за допомогою графічних процесорів.

Проведено навчання нейронної мережі на центральному процесорі, одному та двох графічних процесорах відповідно. З'ясовано, що організація паралельних обчислень на декількох графічних процесорах не означає зменшення тривалості процедури навчання мережі, оскільки витрати на синхронізацію градієнтів між активними процесами значно перевищують користь від паралельних розрахунків. За умови обрання достатньо великого обсягу даних (mini-batch розмір) для організації паралельних обчислень та відповідної архітектури нейронної мережі можна досягти деякого зменшення тривалості процедури її навчання.

Визначено, що зменшення тривалості процедури навчання нейронної мережі залежить від багатьох чинників: її архітектури, кількості параметрів навчання, конфігурації апаратного забезпечення та організації паралельних розрахунків.

References

1. Babenko, A. V., Bodyansky, E. V., Popov, S. V., Slipchenko, E. V. (2009). Predictive-diagnostic recurrent neuro-fuzzy network in the problem of controlling electricity consumption. *Information processing systems, 3*(77), 2–5. Retrieved from: http://nbuv.gov.ua/UJRN/soi_2009_3_3. [In Russian].
2. 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
3. Billings, S. A. (2013). Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley, 574 p. Retrieved from: https://www.amazon.com/Nonlinear-System-Identification-Frequency-Spatio-Temporal/dp/1119943590
4. Bodyanskiy, Y., Popov, S., & Rybalchenko, T. (2008). Multilayer neuro-fuzzy network for short term electric load forecasting. *Lecture Notes in Computer Science – Berlin, Heidelberg: Springer-Verlag, 5010*, 339–348. https://doi.org/10.1007/978-3-540-79709-8_34
5. Cichocki, A., & Unbehauen, R. (1993). *Neural Networks for Optimization and Signal Processing*. Stuttgart: Teubner, 526 p. Retrieved from: https://www.amazon.com/Neural-Networks-Optimization-Signal-Processing/dp/0471930105
6. Climate Data Online Search. (2020). *National centers for environmental information*. Retrieved from: https://www.ncdc.noaa.gov/cdo-web/search
7. Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing, 7*(3–4), 1–199. https://doi.org/10.1561/2000000039
8. 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
9. Dsouza, J. (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
10. 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
11. Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*, 179–211. https://doi.org/10.1016/0364-0213(90)90002-E
12. 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, 27*(1), 175–191. https://doi.org/10.1109/TCST.2017.2766042
13. Firmino, P., de Mattos, Neto P., & Ferreira, T. (2014). Correcting and combining time series forecasters. *Neural Networks, 50*, 1–11. https://doi.org/10.1016/j.neunet.2013.10.008
14. FiveThirtyEight. (2019). Uber Pickups in New York City. Retrieved from: https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city
15. Geqay, R., & Liu, T. (1997). Nonlinear modeling and prediction with feedforward and recurrent networks. *Physica D, 108*, 119–134. https://doi.org/10.1016/S0167-2789(97)82009-X

16. 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
17. Grinberg, J., Jain, A., & Vivek, A. (2014). Predicting Taxi Pickups in New York City. Retrieved from: http://robots.stanford.edu/cs221/2016/restricted/projects/vhchoksi/final.pdf.
18. Grossberg, S. Z. (2010). *Neural Networks and Natural Intelligence*. Cambridge, MA: MIT Press, 651 p. Retrieved from: https://mitpress.mit.edu/books/neural-networks-and-natural-intelligence
19. Haegglund, T. (1984). Adaptive control of systems subject to large parameter changes. *Proc. IFAC 9th Triennial World Congress*. Budapest, 993–998. https://doi.org/10.1016/S1474-6670(17)61102-9
20. Han, M., Xi, J., Xu, S., & Yin, F.-L. (2004). Prediction of chaotic time series based on the recurrent predictor neural network. *IEEE Trans. Signal Processing, 52*(12), 3409–3416. https://doi.org/10.1109/TSP.2004.837418
21. Haykin, S. (2008). *Neural Networks and Learning Machines*. New Jersey: Prentice Hall, 936 p. Retrieved from: https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf
22. Hewamalage, H., Bergmeir, C., & Bandara, K. (January–March 2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting, 37*(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008
23. Hlybovets, M. M., & Oletsky, O. V. (2002). Artificial Intelligence. Kyiv: Kyiv-Mohyla Academy Publishing House, 364 p. [In Ukrainian].
24. Jordan, M. (1992). Constrained supervised learning. *Journal of Mathematical Psychology*, *36*, 396-452. https://doi.org/10.1016/0022-2496(92)90029-7
25. 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
26. Kiani, K. (2005). Detecting business cycle asymmetries using artificial neural networks and time series models. *Computational Economics, 26*(1), 65–89. Retrieved from: https://link.springer.com/article/10.1007/s10614-005-7366-2
27. Kim, Yoon. (2014). Convolutional neural networks for sentence classification. *IEMNLP*, 1746–1751. Retrieved from: https://arxiv.org/abs/1408.5882
28. Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv – preprint arXiv: 1412.6980. Retrieved from: https://www.aminer.org/pub/5550415745ce0a409eb3a739/adam-a-method-for-stochastic-optimization
29. Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *CoRR, abs/1404.5997*. Retrieved from: https://arxiv.org/abs/1404.5997
30. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84–90. https://doi.org/10.1145/3065386
31. Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567–581. https://doi.org/10.1016/S0167-9236(03)00088-5
32. 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
33. Mandic, D. P., & Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Chichester: John Wiley&Sons, 285 p. https://doi.org/10.1002/047084535X
34. Moreira-Matias, Luis, 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
35. Mukai, N., & Yoden, N. (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
36. 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. https://doi.org/10.2139/ssrn.2264379
37. 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
38. Pelinska-Olko, E., & Lewkowicz, M. (2018). Numerical prediction of steady state temperature based on transient measurements. *MATEC Web of Conferences 240, 05024* (ICCHMT 2018). https://doi.org/10.1051/matecconf/201824005024
39. PyTorch. (2020). *PyTorch documentation*. Retrieved from: https://pytorch.org/docs/stable/index.html
40. Rahaman, M. S., Hamilton, M., & Salim, F. D. (2017). Queue Context Prediction Using Taxi Driver Knowledge. K-CAP 2017: Proceedings of the Knowledge Capture Conference, December 2017, Article No.: 35, 1–4. https://doi.org/10.1145/3148011.3154474
41. 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*(8), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
42. Sarkar, D., Bali, R., & Sharma, T. (2018). *Practical Machine Learning with Python.* Springer Science+Business Media. New York. Retrieved from: https://www.apress.com/gp/book/9781484232064
43. Schaller, B. (2005). A regression model of the number of taxicabs in US cities. *Journal of Public Transportation, 8*(5), 4. http://doi.org/10.5038/2375-0901.8.5.4
44. Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks 61*: 85–117. arXiv:1404.7828. https://doi.org/10.1016/j.neunet.2014.09.003
45. Shakhovskaya, N. B., Kaminsky, R. M., & Vovk, O. B. (2018). Artificial intelligence systems: textbook. way. Lviv: Lviv Polytechnic, 392 p. [In Ukrainian].
46. 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
47. Song, H. A., & Lee, S. Y. (2013). Hierarchical Representation Using NMF. Neural Information Processing. Lectures Notes in Computer Sciences 8226, (pp. 466–473). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-42054-2_58
48. Tzafestas, S., & Tzafestas, E. (2001). Computational intelligence techniques for short-term electric load forecasting. *Journal of Intelligent and Robotic Systems, 31*, 7–68. https://doi.org/10.1023/A:1012402930055
49. Wang, J., & Hu, S. (2002). Global asymptotic stability and global exponential stability of continuous-time recurrent neural networks. IEEE Trans. *Automatic Control, 47*, 802–807. https://doi.org/10.1109/TAC.2002.1000277
50. Williams, R. J., & Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation, 1*, 270–280. https://doi.org/10.1162/neco.1989.1.2.270
51. Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, D. (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
52. YouTube. (2020). *Consumer assessment of taxi services in large cities*. Retrieved from: https://www.youtube.com/watch?v=RE2j1B7EdQM. [In Ukrainian].

53. Yuan, N. J., Zheng, Yu., Zhang, L., & Xie, X. (2013). T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering, 25*(11), 2390–2403. https://doi.org/10.1109/TKDE.2012.153

54. Zhang Xiang, Zhao Junbo, LeCun Yann. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pp. 649–657. Retrieved from: https://arxiv.org/abs/1509.01626

M. I. Zghoba, Yu. I. Hrytsiuk

Lviv Polytechnic National University, Lviv, Ukraine

FORECASTING THE DEMAND FOR PASSENGER TAXI TRANSPORT BY NEURAL NETWORK METHODS

Peculiarities of forecasting the demand for passenger transportation by taxi by neural network methods for different sets of input data, composition of network architecture parameters, hardware configuration and its capacity are considered. It was found that to reduce the waiting time for new orders and the distance to customers, it is advisable to use appropriate information and analytical systems, the work of which is based on artificial intelligence. This will solve the problem of demand for taxi transportation in the relevant period of the day, taking into account weather conditions, holidays, weekends and working days, as well as the seasons. Taking into account the existing transport facilities - flights, trains or buses significantly improves the work of such an advisory system. The hybrid architecture of the neural-phase network used in the work allows to simultaneously solve the problem of short-term forecasting of demand for passenger taxis, as well as to diagnose the network itself, which is to detect abrupt changes in the properties of the computing process. To achieve the appropriate accuracy of the forecast, the work developed input sets in the amount of 4.5 million taxi trips. To reduce the duration of the neural network training procedure, parallel calculations are organized between different network nodes using graphics processors.

Neural network training on the CPU, one and two GPUs, respectively. It was found that the organization of parallel computing on several GPUs does not always reduce the duration of the network learning procedure, as the cost of synchronizing gradients between active processes far outweighs the benefits of parallel computing. It is established that with a large amount of data for the organization of parallel calculations and the corresponding architecture of the neural network, it is possible to achieve some reduction in the duration of the training procedure. It is determined that the reduction of the duration of the neural network learning procedure depends on the following factors: its architecture, the number of learning parameters, hardware configuration and organization of parallel calculations.

Keywords: information-analytical system; hardware configuration; machine learning, neural network learning, acceleration of learning process, parallelization of learning procedure.