УДК: 004.89 DOI 10.33111/mise.103.5

Гладка Ю. А.,
кандидат фізико-математичних наук, доцент
кафедри системного аналізу та кібербезпеки
КНЕУ імені Вадима Гетьмана,
Остапенко В. О.,
студентка 4 курсу за освітньо-професійною
програмою «Системний аналіз»,
КНЕУ імені Вадима Гетьмана
Чугаєва О.В.,
старший викладач кафедри системного аналізу та кібербезпеки
КНЕУ імені Вадима Гетьмана

Gladka Yu. A.,
Candidate of Physical and Mathematical Sciences, Associate Professor
of System Analysis and Cybersecurity Department KNEU
named after Vadym Hetman
Ostapenko V. O.,
Bachelor's degree in «System Analysis», KNEU named after Vadym Hetman
Chuhaieva O.V.,
Senior lecturer of System Analysis and Cybersecurity Department
KNEU named after Vadym Hetman

ОПТИМІЗАЦІЯ ТРАНСПОРТНИХ ЗАДАЧ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

OPTIMISATION OF TRANSPORTATION PROBLEMS USING NEURAL NETWORKS

Анотація. *В сфері логістичних систем дуже мінливий характер навколишнього середовища вносить невизначеність щодо його стану. Як наслідок, виникає необхідність вдосконалення традиційних математичних інструментів. Стаття присвячена застосуванню нейронних мереж у вирішенні логістичних задач та підбору поправкових коефіцієнтів у класичних транспортних задачах. Розглянуто вплив зовнішніх факторів на логістичні процеси. Описано використання багатошарової нейронної мережі з використанням бібліотеки Keras мови програмування Python для автоматизованого підбору необхідних параметрів на основі історичних даних та стандартних розрахунків.*

У статті представлено структуру та параметри нейронної мережі, досліджено її ефективність. Описується процес тренування та використання мережі для прогнозування поправкових коефіцієнтів.

Ключові слова: *нейронна мережа, логістична задача, транспортна задача, аналіз, логістичні виклики, ефективність методів оптимізації, багатошарова нейронна мережа, тренування нейронної мережі, поправкові коефіцієнти*

Abstract. *In the field of logistics systems, the very minimal nature of the excessive middle ground will introduce insignificance to the point where I will*

become. As a result, there is a need to thoroughly modernize traditional mathematical tools. The article is devoted to the establishment of neural measures in major logistics problems and the selection of correction coefficients in classical transport problems. The influx of external factors on logistics processes is examined. We describe the use of a multi-ball neural network using the Keras library and Python programming for the automated selection of necessary parameters based on historical data and standard layouts. The article presents the structure and parameters of the neural network and monitors its effectiveness. The process of training and calculating measures for predicting correction coefficients is described.
Keywords: *neural network, logistics problem, transport task, analysis, logistics calls, effectiveness of optimization methods, rich-ball neural network, neural network training, correction coefficients*

Актуальність теми дослідження. На сьогодняшній день розв'язання задач логістики є достатньо актуальним питанням. Існуючі методи оптимізації та управління, хоча й ефективні, але залишають деякі проблеми, тому потребують подальшого вдосконалення. Сучасні логістичні виклики передбачають управління складними системами виробництва, постачання та збуту. У цьому контексті важливого значення набуває вирішення різних специфічних проблем, таких як оцінка ефективності складних логістичних систем, координація і полегшення транспортування готової продукції до кінцевих споживачів.

Постановка проблеми. В сфері логістичних систем дуже мінливий характер навколишнього середовища вносить невизначеність щодо його стану. Як наслідок, виникає необхідність вдосконалення традиційних математичних інструментів. Крім того, завдання, пов'язані з логістичними системами, мають високий рівень складності через свою багатовимірність.

Враховуючи вищезгадане, запропонований метод застосування багатошарових нейронних мереж для автоматизованого підбору поправкових коефіцієнтів у класичних транспортних задачах має потенціал вирішення проблем, пов'язаних із змінністю та комплексністю завдань у цій галузі.

Аналіз останніх досліджень і публікацій. Останнім часом багато вчених та дослідників досліджували застосування новітніх технологій в логістиці. Наприклад, автори Шенхао Ван, Байчуань Мо, Цзіньхуа Чжао в статті «Прогнозування вибору виду транспорту за допомогою 86 класифікаторів машинного навчання: Емпіричне порівняльне дослідження» застосовують велику кількість класифікаторів машинного навчання для прогнозування поведінки під час подорожей. Для отримання узагальнюючих результатів у цій статті представлено емпіричний зразок з використанням 86 класифікаторів з 14 модельних сімейств для прогнозування вибо-

ру способу пересування на основі набору даних Національного обстеження подорожей домогосподарств (NHTS) 2017 року. Олена Головіна (м. Кременчук, Класичний приватний університет), в статті «Сучасні технології в управлінні транспортною логістикою» досліджувала роль та можливості використання сучасних технологій, а саме штучного інтелекту, в управлінні транспортною логістикою. В статті розглядаються різні аспекти використання штучного інтелекту, такі як машинне навчання, нейронні мережі, системи контроля трафіка, системи голосового сповіщення, системи контроля парку транспортних засобів, системи геолокації в контексті транспортної логістики.

Постановка завдання. Мета статті полягає в дослідженні та розробці методології застосування нейронних мереж для покращення вирішення класичних транспортних задач. Основним завданням є розробка ефективної моделі, здатної автоматично підбирати поправкові коефіцієнти на основі історичних даних.

Виклад основного матеріалу. Однією з проблем, з якою стикаються різні організації, є вирішення транспортної задачі, а саме транспортування/відправлення товарів з виробництва до місця призначення з найменшими можливими витратами при дотриманні обмежень на попит і пропозицію. Для розв'язання цієї задачі використовують особливий клас методів лінійного програмування, який був розроблений для моделей з лінійними цільовими функціями та обмеженнями. Транспортна задача може бути збалансованою — це той випадок, коли загальна пропозиція дорівнює загальному попиту, або незбалансованою — попит перевищує пропозицію, або навпаки.

Транспортну задачу можна розділити на два типи: транспортна задача, що ґрунтується на вартості — зосереджена на визначенні транспортного плану, який мінімізує витрати на транспортування товарів, і транспортна задача, що ґрунтується на часі — ставить на перше місце максимальну ефективність використання часу.

При створенні системи постачання готової продукції організація чи підприємство повинні вирішити низку питань, пов'язаних з транспортуванням. Насамперед, це стосується вибору відповідного способу перевезення, способів організації перевезення та визначення найбільш відповідних транспортних засобів, які слід задіяти.

При виборі найбільш ефективного способу транспортування, в першу чергу, необхідно враховувати відповідність типу транспортного засобу специфічним характеристикам товарів, що достав-

ляються. Наприклад, такі фактори, як безпека товару, оптимальне використання місткості та вантажопідйомності транспортного засобу, а також зниження транспортних витрат.

Важливо підкреслити, що одним із суттєвих обмежень використання критерію мінімальної відстані без урахування часу є нехтування безпекою вантажу під час перевезення. Безпека вантажу відіграє вирішальну роль у транспортуванні, оскільки є першочерговим завданням. Коли водії здійснюють фактичну доставку без попередньо затвердженого маршруту, вони стикаються з проблемою вибору маршруту. У таких ситуаціях зазвичай обирають найкоротший маршрут, припускаючи, що це найефективніший варіант. Але варто зазначити, що умови руху на різних дорогах можуть суттєво відрізнятися. Тому не завжди доцільно обирати найкоротший маршрут, якщо він передбачає затори або рух дорогами з пошкодженим покриттям. У цих випадках найкоротший маршрут може бути не найкращим рішенням.

Для покращення оцінки дорожньої мережі пропонується включити оцінюючий показник для визначення відстані між пунктами, що охоплює як якісні, так і кількісні параметри. Наприклад, при перевезенні крихких товарів, збереження яких є пріоритетнішим за своєчасну доставку, важливо встановити додаткові вимоги до транспортування. Отже, при організації транспортної логістики відстані слід розглядати не лише у фізичних одиницях, а й в умовних одиницях, які враховують поправкові коефіцієнти.

Стан дорожнього покриття має величезне значення для безпечного транспортування вантажів.

При розробці планів перевезень доцільно враховувати наступні фактори і відповідно вводити поправкові коефіцієнти:

- Дефекти дорожнього покриття, такі як різні види пошкоджень, можуть мати негативний вплив як на технічний стан транспортного засобу, так і на безпеку вантажу.
- У густонаселених регіонах набагато більше перехресть, а отже, набагато більше коливань швидкості під час руху по об'їзних маршрутах.
- Рівень руху на конкретній ділянці дороги відіграє вирішальну роль у визначенні найефективнішого способу дій, оскільки у випадках високої інтенсивності руху може бути доцільно обрати альтернативний маршрут.
- Ступінь завантаженості проїжджої частини підвищує ймовірність виникнення заторів та аварій.

- Підйоми та спуски суттєво впливають на процес транспортування

- Для забезпечення безпеки та запобігання аваріям дуже важливо мінімізувати навантаження на шлях, під яким мається на увазі сумарна вага вантажу і транспортних засобів, що проїжджають через певну ділянку за певний проміжок часу.

- Дорога повинна володіти архітектурними якостями, які відповідають необхідним вимогам з точки зору зручності та безпеки руху. Ці елементи охоплюють проїжджу частину, мости та будівлі.

За допомогою цих поправкових коефіцієнтів можна розширити класичну математичну модель транспортної задачі. В даній статті мається на увазі наступний вигляд моделі:

$$\min \sum_{i\epsilon I} \sum_{j\epsilon J} c_{ij} x_{ij}, \tag{1}$$

за умови, що:

$$\sum_{j\epsilon J} x_{ij} \leq S_i \ \ \forall i \in I; \tag{2}$$

$$\sum_{i\epsilon I} x_{ij} \geq D_j \ \ \forall j \in J; \tag{3}$$

$$x_{ij} \geq 0 \ \ \forall i \in I \ \ \forall j \in J. \tag{4}$$

де x_{ij} — кількість вантажу, що відправляється з $i \in I$ в $j \in J$,

c_{ij} — вартість перевезення одиниці вантажу між усіма парами i, j,

S_i — обсяг пропозиції у вузлі і для $i \in I$,

D_j — потреба у вузлі j для $j \in J$,

I — множина вузлів, звідки відправляється вантаж,

J — множина вузлів, куди прибуває вантаж.

Описана транспортна задача є типовою задачею лінійного програмування, яка просто вирішується за допомогою класичних методів оптимізації. Використання нейронних мереж для розв'язання цієї конкретної задачі є неефективним.

Нейронні мережі, зазвичай, використовуються для розв'язання задач, де існує складна нелінійність чи невизначеність в даних, і коли традиційні методи можуть бути неефективними. Наприклад, для розрахунку поправкових коефіцієнтів. В реальності транспортування продукції вимагає врахування безлічі факторів, які в свою чергу вимагають багато розрахунків на основі великих статистичних даних. Мета цієї статті — висвітлити перспективи застосування нейронних мереж, тому буде використовуватися лише простий приклад цих даних, отриманих за простим розрахунком.

Розширимо математичну модель задачі шляхом модифікації вартості перевезення для кожної пари пунктів на основі відповідних оцінок поправкових коефіцієнтів.

Математично:

$$c'_{ij} = c_{ij} * K_{ij}, \quad (5)$$

де c_{ij} — вартість перевезення одиниці вантажу між усіма парами i, j,

c'_{ij} — оновлена вартість перевезення одиниці вантажу,

K_{ij} — поправковий коефіцієнт.

Тепер вартість перевезення залежить від поправкових коефіцієнтів.

Нейронні мережі — це складні мережі взаємопов'язаних вузлів, або нейронів, які співпрацюють для вирішення складних проблем.

Нейронні мережі складаються з паралельних процесорів, розташованих шарами. Перший шар отримує вхідну інформацію, а наступні обробляють вихідні дані попереднього. Кожен вузол має обмежену область знань і пов'язаний з вузлами інших шарів. Мережі адаптуються з часом, навчаючись на основі оцінки важливості вхідних даних, які сприяють отриманню вірогідних результатів.

Застосування нейронних мереж для отримання коефіцієнтів у даному випадку може бути обґрунтоване кількома причинами:

1. Складність взаємозв'язків: Між різними факторами, такими як дефекти дорожнього покриття, населені пункти, інтенсивність руху та інші, може існувати складна та нелінійна взаємодія. Нейронні мережі здатні автоматично виявляти та враховувати такі складні взаємозв'язки, що може поліпшити точність моделі порівняно зі стандартними математичними методами.

2. Гнучкість моделі: Нейронні мережі дозволяють створювати гнучкі моделі, які можуть адаптуватися до різних умов та вхідних даних. Це особливо важливо в сфері транспортного планування, де умови можуть значно змінюватися.

3. Можливість враховувати неявні зв'язки: Нейронні мережі можуть виявляти неявні зв'язки та закономірності в даних, які можуть бути важко визначити або формалізувати математично.

4. Автоматичне навчання: Нейронні мережі можуть автоматично навчатися з великої кількості даних, що робить їх ефективними в ситуаціях, де важко або неможливо визначити точні математичні залежності.

В даному випадку, нейронна мережа може допомогти покращити прогнозування коефіцієнтів, які враховують вплив різних факторів на перевезення вантажу.

Отже, для створення моделі нейронної мережі було розраховано тестовий набір вхідних даних, а саме розрахунок впливу:

- дефектів дорожнього покриття: a * 0.1, де a — кількість дефектів;
- населених пунктів: b * 0.05, де b — кількість пунктів;
- інтенсивності руху: c * 0.02, де c — певна інтенсивність;
- рівня завантаження дороги: d * 0.1, де d — рівень;
- підйомів і спусків: e * 0.05, де e — їх кількість;
- якості дороги: f * 0.1, де f — певна оцінка якості;
- архітектурних якостей дороги: g * 0.05, де g — певна оцінка якості.

Для реалізації ідеї використовується мова програмування Python. Вона визнана широкою аудиторією як одна з найбільш зручних та популярних мов програмування для роботи з нейронними мережами і глибоким навчанням.

Щоб мати можливість навчити нейронну мережу, було використано випадкові дані та, вручну розраховані по цим даним, поправкові коефіцієнти. А саме 42 набори значень [a, b, c, d, e, f, g] та 42 набори значень коефіцієнтів [$k_a, k_b, k_c, k_d, k_e, k_f, k_g$] :

Таблиця 1

ДАНІ ДЛЯ НАВЧАННЯ

X_train = np.array([	y_train = np.array([
[3, 2, 4, 3, 2, 4, 1],	[0.3, 0.1, 0.08, 0.3, 0.1, 0.4, 0.05],
[2, 1, 3, 2, 1, 3, 1],	[0.2, 0.05, 0.06, 0.2, 0.05, 0.3, 0.05],
[12, 1, 2, 2, 0, 5, 1],	[1.2, 0.05, 0.04, 0.2, 0.0, 0.5, 0.05],
[8, 2, 3, 4, 1, 3, 2],	[0.8, 0.1, 0.06, 0.4, 0.05, 0.3, 0.1],
[15, 3, 1, 2, 2, 4, 4],	[1.5, 0.15, 0.02, 0.2, 0.1, 0.4, 0.2],
[10, 1, 5, 3, 0, 2, 3],	[1.0, 0.05, 0.1, 0.3, 0.0, 0.2, 0.15],
[5, 4, 2, 1, 3, 1, 5],	[0.5, 0.2, 0.04, 0.1, 0.15, 0.1, 0.25],
[12, 2, 4, 5, 1, 2, 1],	[1.2, 0.1, 0.08, 0.5, 0.05, 0.2, 0.05],
[8, 1, 3, 2, 0, 4, 4],	[0.8, 0.05, 0.06, 0.2, 0.0, 0.4, 0.2],
[15, 3, 1, 3, 2, 3, 2],	[1.5, 0.15, 0.02, 0.3, 0.1, 0.3, 0.1],
[10, 2, 5, 4, 1, 1, 3],	[1.0, 0.1, 0.1, 0.4, 0.05, 0.1, 0.15],
[5, 4, 2, 2, 3, 5, 5],	[0.5, 0.2, 0.04, 0.2, 0.15, 0.5, 0.25],
[12, 1, 4, 1, 0, 2, 1],	[1.2, 0.05, 0.08, 0.1, 0.0, 0.2, 0.05],
[8, 2, 3, 3, 1, 4, 3],	[0.8, 0.1, 0.06, 0.3, 0.05, 0.4, 0.15],

[15, 3, 1, 4, 2, 3, 2], [10, 1, 5, 5, 0, 1, 5], [5, 4, 2, 2, 3, 4, 4], [12, 2, 4, 3, 1, 2, 1], [8, 1, 3, 2, 0, 4, 3], [10, 2, 5, 4, 1, 1, 5], [5, 4, 2, 2, 3, 5, 4], [12, 1, 4, 5, 1, 2, 1], [8, 14, 2, 7, 2, 4, 2], [8, 0, 15, 3, 15, 0, 8], [15, 15, 6, 7, 17, 10, 4], [16, 3, 11, 16, 3, 4, 1], [11, 10, 17, 4, 4, 13, 5], [8, 8, 11, 11, 1, 9, 1], [16, 9, 4, 9, 4, 17, 11], [7, 15, 2, 1, 2, 15, 8], [15, 2, 2, 14, 5, 14, 12], [5, 4, 3, 13, 9, 12, 12], [15, 3, 10, 13, 17, 10, 4], [0, 15, 6, 4, 0, 17, 4], [2, 1, 16, 9, 9, 3, 15], [15, 2, 7, 1, 13, 3, 2], [8, 0, 8, 15, 16, 6, 12], [12, 5, 8, 9, 10, 17, 16], [16, 6, 9, 6, 9, 15, 10], [1, 14, 7, 9, 8, 8, 0], [5, 2, 7, 3, 5, 8, 15], [8, 4, 0, 12, 4, 16, 9]])	[1.5, 0.15, 0.02, 0.4, 0.1, 0.3, 0.1], [1.0, 0.05, 0.1, 0.5, 0.0, 0.1, 0.25], [0.5, 0.2, 0.04, 0.2, 0.15, 0.4, 0.2], [1.2, 0.1, 0.08, 0.3, 0.05, 0.2, 0.05], [0.8, 0.05, 0.06, 0.2, 0.0, 0.4, 0.15], [1.0, 0.1, 0.1, 0.4, 0.05, 0.1, 0.25], [0.5, 0.2, 0.04, 0.2, 0.15, 0.5, 0.2], [1.2, 0.05, 0.08, 0.5, 0.05, 0.2, 0.05], [0.8, 0.7000000000000001, 0.04, 0.7000000000000001, 0.1, 0.4, 0.1], [0.8, 0.0, 0.3, 0.30000000000000004, 0.75, 0.0, 0.4], [1.5, 0.75, 0.12, 0.7000000000000001, 0.8500000000000001, 1.0, 0.2], [1.6, 0.15000000000000002, 0.22, 1.6, 0.15000000000000002, 0.4, 0.05], [1.1, 0.5, 0.34, 0.4, 0.2, 1.3, 0.25], [0.8, 0.4, 0.22, 1.1, 0.05, 0.9, 0.05], [1.6, 0.45, 0.08, 0.9, 0.2, 1.7000000000000002, 0.55], [0.7000000000000001, 0.75, 0.04, 0.1, 0.1, 1.5, 0.4], [1.5, 0.1, 0.04, 1.4000000000000001, 0.25, 1.4000000000000001, 0.6000000000000001], [0.5, 0.2, 0.06, 1.3, 0.45, 1.2000000000000002, 0.6000000000000001], [1.5, 0.15000000000000002, 0.2, 1.3, 0.8500000000000001, 1.0, 0.2], [0.0, 0.75, 0.12, 0.4, 0.0, 1.7000000000000002, 0.2], [0.2, 0.05, 0.32, 0.9, 0.45, 0.30000000000000004, 0.75], [1.5, 0.1, 0.14, 0.1, 0.65, 0.30000000000000004, 0.1], [0.8, 0.0, 0.16, 1.5, 0.8, 0.6000000000000001, 0.6000000000000001], [1.2000000000000002, 0.25, 0.16, 0.9, 0.5, 1.7000000000000002, 0.8], [1.6, 0.30000000000000004, 0.18, 0.6000000000000001, 0.45, 1.5, 0.5], [0.1, 0.7000000000000001, 0.14, 0.9, 0.4, 0.8, 0.0], [0.5, 0.1, 0.14, 0.30000000000000004, 0.25, 0.8, 0.75], [0.8, 0.2, 0.0, 1.2000000000000002, 0.2, 1.6, 0.45]])

Логіка роботи нейронної мережі полягає в тому, щоб вивчити взаємозв'язки між вхідними умовами (факторами, такими як де-

фекти, тощо) і відповідними виходами (поправковими коефіцієнтами для планування перевезень).

Багатошаровий персептрон (MLP) це архітектура, що є різновидом нейронної мережі прямого поширення (FNN). Багатошаровий перцептрон — це тип штучної нейронної мережі, яка складається з трьох або більше шарів. Ці шари включають вхідний шар, один або більше прихованих шарів та вихідний шар. Кожен шар складається з нейронів, які з'єднані з нейронами наступного шару. Крім того, вона використовує нелінійні активаційні функції (ReLU) в прихованих шарах, що є ще однією характерною особливістю MLP. В процесі навчання мережа робить прогнози на наборі вхідних даних, порівнює їх з реальними значеннями, а потім налаштовує ваги у зв'язках, щоб зменшити похибку. Цей ітеративний процес поступово навчає мережу так, щоб входи відповідали бажаним результатам.

Алгоритм створення моделі та навчання нейронної мережі досить стандартний і виглядає наступним чином:

1. Імпорт бібліотек: Імпортується бібліотека numpy для роботи з матрицями та числовими операціями. З бібліотеки sklearn.preprocessing імпортується клас MinMaxScaler для нормалізації вхідних даних. Використовується Sequential, Dense і Dropout з бібліотеки Keras для побудови та конфігурації нейронної мережі.

2. Створення класу NeuralNetwork: Конструктор класу ініціалізує об'єкт класу зі змінною scaler для нормалізації та model для збереження архітектури нейронної мережі.

```
class NeuralNetwork:
 def __init__(self, input_size):
  self.scaler = MinMaxScaler()
  self.model = self.build_model(input_size)
```

3. Метод build_model: Створюється модель з трьома шарами нейронів: перший шар з 16 нейронами та активаційною функцією ReLU, другий шар Dropout для уникнення перенавчання, третій шар з 12 нейронами та активаційною функцією ReLU, і кінцевий шар з 7 нейронами та лінійною активаційною функцією. Модель компілюється з оптимізатором Adam і функцією втрат «mean_squared_error».

```
  def build_model(self, input_size):
   model = Sequential()
   model.add(Dense(16,       input_dim=input_size,
activation='relu'))
```

```
    model.add(Dropout(0.2))
    model.add(Dense(12, activation='relu'))
    model.add(Dense(7, activation='linear'))
    model.compile(optimizer='adam',
loss='mean_squared_error')
    return model
```

4. Метод train: Нормалізуються вхідні дані за допомогою MinMaxScaler. Модель тренується на нормалізованих даних з вказаною кількістю епох та розміром пакета.

```
   def train(self, X, y, epochs=3000,
batch_size=16):
    X_scaled = self.scaler.fit_transform(X)
    self.model.fit(X_scaled, y, epochs=epochs,
batch_size=batch_size)
```

5. Метод predict: Нормалізуються вхідні дані для прогнозування. Модель використовується для прогнозування значень.

```
   def predict(self, X):
    X_scaled = self.scaler.transform(X)
    return self.model.predict(X_scaled)
```

6. Навчання нейронної мережі: Створюється екземпляр класу NeuralNetwork. Здійснюється навчання мережі на попередньо підготовлених даних X_train та y_train за допомогою методу train.

```
  input_size = X_train.shape[1]
neural_network = NeuralNetwork(input_size)
neural_network.train(X_train, y_train)
```

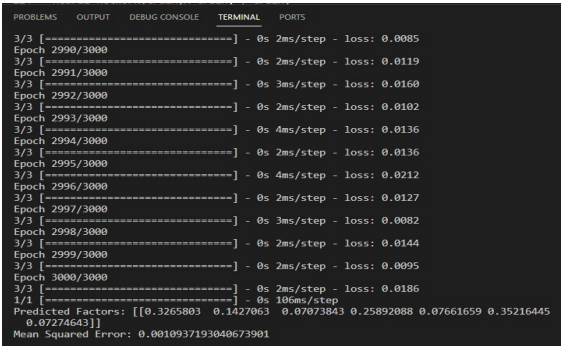
7. Прогнозування за допомогою навченої мережі: Створюється новий набір умов для прогнозування new_conditions. Викликається метод predict для прогнозування поправкових коефіцієнтів для цих умов.

```
new_conditions = np.array([[3, 2, 4, 3, 2, 4, 1]])
predicted_factors =
neural_network.predict(new_conditions)
```

8. Виведення результатів: Роздрукування прогнозованих коефіцієнтів для нових умов.

9. Оцінка роботи моделі: імпорт бібліотеки scikit-learn та matplotlib для порівняння реальних та передбачених значень за допомогою середньоквадратичної помилки (Mean Squared Error, MSE) та побудови графіка. Визначається масив реальних значень, які вже відомі. Застосовується функція MSE для порівняння реальних та передбачених значень. Результат зберігається у змінній mse. Додається лінія графіка для реальних значень з маркера-

ми ‘o’ та лінія графіка для передбачених значень з маркерами ‘x’. Відображається графік.



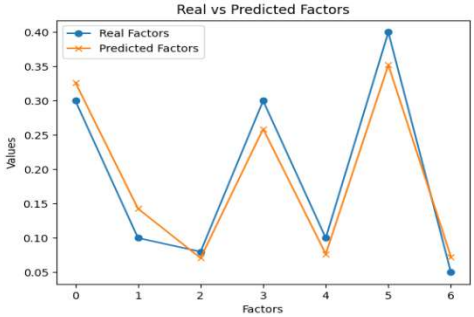


Рис. 1. Результат роботи моделі та його оцінка

```
  real_factors = np.array([[0.3, 0.1, 0.08, 0.3,
0.1, 0.4, 0.05]])
  mse = mean_squared_error(real_factors,
predicted_factors)
  print(«Mean Squared Error:», mse)
  plt.plot(real_factors.flatten(), label=‘Real
Factors’, marker=‘o’)
  plt.plot(predicted_factors.flatten(),
label=‘Predicted Factors’, marker=‘x’)
  plt.xlabel(‘Factors’)
  plt.ylabel(‘Values’)
  plt.legend()
```

```
plt.title('Real vs Predicted Factors')
plt.show()
```

Отже, маємо наступний результат: для факторів $a = 3$, $b = 2$, $c = 4$, $d = 3$, $e = 2$, $f = 4$, $g = 1$ ми мали наступні розраховані власноруч коефіцієнти: $k_{\mathrm{a}} = 0.3, k_{\mathrm{b}} = 0.1, k_{\mathrm{c}} = 0.08, k_{\mathrm{d}} = 0.3, k_{\mathrm{e}} = 0.1, k_{\mathrm{f}} = 0.4, k_{\mathrm{g}} = 0.05$. Навчена нейронна мережа для цих самих факторів розрахувала наступні коефіцієнти

$$k_{\mathrm{a}} = 0.3265803, k_{\mathrm{b}} = 0.1427063, k_{\mathrm{c}} = 0.07073843\ , k_{\mathrm{d}} = 0.25892088,\ k_{\mathrm{e}} == 0.07661659, k_{\mathrm{f}} = 0.35216445, k_{\mathrm{g}} = 0.07274643.$$

Значення mse = =0.0010937193040673901 є досить низьким і свідчить про добре приближення передбачених значень до реальних. Порівняно з реальними значеннями, передбаченні мають деякі відмінності, але загалом модель є достатньо точною у відтворенні вихідних даних, що видно з графіку.

Висновок. У цій статті досліджено застосування нейронної мережі до пошуку поправкових коефіцієнтів, що розширюють стандартну математичну модель транспортної задачі. Ефективність та результативність запропонованого методу підтверджено за допомогою розрахунку середньоквадратичної помилки (MSE) та побудови графіка для порівняння реальних та передбачених значень розробленою моделлю. Наша модель є прикладом простої нейронної мережі, що, в свою чергу, обумовлює те, що результат роботи не ідеально збігається з очікуваним.

Подальші наукові дослідження доцільні в напрямках: оптимізація архітектури нейронних мереж — дослідження впливу різних архітектур нейронних мереж (кількість шарів, нейронів, функцій активації) на результати розв'язання транспортних задач та розгляд можливості використання глибоких нейронних мереж для моделювання складних логістичних сценаріїв. Інтеграція з іншими методами оптимізації — розгляд комбінованого підходу, який використовує нейронні мережі разом із традиційними методами оптимізації для знаходження більш ефективних рішень. Обробка невизначеності і стохастичність — вивчення та розробка методів для врахування невизначеності та стохастичності в транспортних задачах за допомогою нейронних мереж. Використання реальних даних — збільшення реалізму досліджень за рахунок використання реальних даних логістичних задач для навчання моделі.

Бібліографічні посилання

1. Transportation Problem Explained and how to solve it? [Електронний ресурс] — Режим доступу до ресурсу: https://www.mygreatlearning.com/blog/transportation-problem-explained/.

2. Стоколяс В.С. Ефективність транспортної логістики як складової логістичної системи [Електронний ресурс] — 2014. — Режим доступу до ресурсу: http://www.economy.nayka.com.ua/?op=1&z=3216.

3. Проблеми та перспективи розвитку транспортної логістики підприємств в сучасних умовах [Електронний ресурс] — 2023. — Режим доступу до ресурсу: http://confmanagement.kpi.ua/proc/article/view/279798.

4. What is a neural network? [Електронний ресурс] — Режим доступу до ресурсу: https://www.ibm.com/topics/neural-networks.

5. What are multilayer perceptrons? [Електронний ресурс] — Режим доступу до ресурсу: https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron.

6. Що таке багатошаровий персептрон (Multilayer Perceptron, MLP) у машинному навчанні? [Електронний ресурс] — Режим доступу до ресурсу: https://thetransmitted.com/adlucem/shho-take-mlp-u-mashynnomu-navchanni/.

7. Deep Learning Basics — Feed Forward Neural Networks (FFNN) [Електронний ресурс] — Режим доступу до ресурсу: https://medium.com/@sasirekharameshkumar/deep-learning-basics-part-10-feed-forward-neural-networks-ffnn-93a708f84a31.

8. Shenhao Wang, Baichuan Mo, Jinhua Zhao Predicting Travel Mode Choice with 86 Machine Learning Classifiers: An Empirical Benchmark Study [Електронний ресурс] — 2020. — Режим доступу до ресурсу: https://www.mit.edu/~baichuan/Baichuan/publication/Predicting%20Travel%20Mode%20Choice%20with%2086%20Machine%20Learning%20Classifiers%20An%20Empirical%20Benchmark%20Study.pdf.

9. Головіна О.В. Сучасні технології в управлінні транспортною логістикою [Електронний ресурс] — 2023. — Режим доступу до ресурсу: https://www.researchgate.net/publication/371219510_Sucasni_tehnologii_v_upravlinni_transportnou_logistikou/fulltext/6478f6a42cad460a1be942d7/Sucasni-tehnologii-v-upravlinni-transportnou-logistikou.pdf.

Статтю подано до редакції: 28.11.2023