

Список використаних джерел

1. Градієнтний спуск: алгоритм та приклад на Python. *Robotdreams*. URL: <https://robotdreams.cc/uk/blog/331-gradiyentniy-spusk-algorithm-ta-priklad-na-python> (дата звернення 28.04.2024).
2. Нельсон Д. Що таке градієнтний спуск? *UniteAI*. Серпень 23, 2020. URL: <https://www.unite.ai/uk/what-is-gradient-descent/> (дата звернення 28.04.2024).
3. Лекції з дисципліни «Обчислювальна математика та програмування» для студентів денної та заочної форм навчання напряму підготовки 6.051301 «Хімічна технологія» / укл. Г. М. Дмитрієнко. Східноукраїнський національний університет імені Володимира Даля, 2010. 88 с.

УДК 004.9

*Ілик В. В., здобувач 2 курсу
спеціальності 122 Комп'ютерні науки,
науковий керівник:
Якубич К. О., асистент кафедри
інформаційних технологій*

ТЕОРІЇ АЛГОРИТМІВ У ЗАДАЧАХ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Штучний інтелект (ШІ) є однією з найбільш динамічних галузей сучасної науки та техніки, яка швидко розвивається. В основі багатьох досягнень ШІ лежать теорії алгоритмів, які забезпечують ефективне розв'язання складних задач, аналіз великих обсягів даних та прийняття рішень. Розвиток алгоритмів для ШІ є критичним для прогресу у сферах обробки природної мови, розпізнавання образів, автоматизації та робототехніки.

Розробка ефективних алгоритмів для ШІ має велике значення для багатьох галузей економіки та суспільства. Зокрема, вони дають змогу автоматизувати рутинні процеси, підвищувати продуктивність, створювати нові можливості для аналізу даних та прогнозування. В умовах стрімкого зростання обсягів інформації, що потребує обробки, та складності задач, які потрібно розв'язувати, вдосконалення алгоритмів стає ключовим фактором успішного розвитку ШІ [1].

Аналіз останніх досліджень і публікацій. Останні дослідження в галузі теорії алгоритмів для ШІ зосереджені на таких напрямках:

1. Машинне навчання та глибоке навчання (алгоритми навчання на основі великих наборів даних, як-от нейронні мережі та методи підсилення).
2. Алгоритми оптимізації (методи оптимізації, включно з градієнтними методами, еволюційними алгоритмами та стохастичною оптимізацією, використовуються для покращення продуктивності моделей ШІ).
3. Алгоритми обробки природної мови (NLP) (розробка моделей для розуміння та генерації людської мови, зокрема трансформерів і моделей на основі BERT).
4. Алгоритми для автономних систем (включають плани дій і системи прийняття рішень для автономних транспортних засобів та роботів).

Метою цієї роботи є аналіз сучасних теорій алгоритмів, що використовуються в задачах штучного інтелекту, зокрема розгляд методів машинного навчан-

ня, оптимізації та обробки природної мови, а також визначення напрямів подальшого розвитку та вдосконалення алгоритмів для підвищення їх ефективності.

Виклад основного матеріалу. Основні задачі дослідження зазвичай такі:

- 1) визначення основних типів алгоритмів, що застосовуються в ШІ;
- 2) аналіз ефективності та застосовності різних алгоритмів у конкретних задачах ШІ;
- 3) визначення напрямів для покращення наявних алгоритмів та розробки нових методів [2, 3].

Машинне навчання (ML) є одним із основних підходів у ШІ, що базується на розробці алгоритмів, які можуть навчатися з даних. Глибоке навчання (DL) є підмножиною ML, що використовує багатопшарові нейронні мережі для автоматичного вилучення характеристик з даних. Основні алгоритми включають:

- нейронні мережі (використовуються для обробки складних вхідних даних, як-от зображення та звук).
- методи підсилення (адаптивні алгоритми, що комбінують прості моделі для підвищення загальної точності).

Оптимізація є критичним компонентом для навчання моделей ШІ. Основні методи включають:

- градієнтний спуск (використовується для мінімізації функцій помилки в нейронних мережах);
- еволюційні алгоритми (застосовуються для розв'язання задач з великою кількістю локальних мінімумів).

Найсучасніші моделі NLP використовують:

- трансформери (моделі на основі архітектури трансформера, як-от BERT та GPT, які забезпечують високу точність у задачах розуміння тексту та генерації мови);
- рекурентні нейронні мережі (RNN) (використовуються для обробки послідовних даних, як-от текст або часові ряди).

Алгоритми для автономних систем включають методи планування дій та прийняття рішень:

- плани дій на основі пошуку (використовуються для визначення оптимальних маршрутів та стратегій);
- розподілені алгоритми (застосовуються в системах із багатьма агентами, наприклад, у робототехніці та автономних транспортних засобах) [4, 5].

Висновки. Алгоритми є невід'ємною частиною розвитку штучного інтелекту, забезпечуючи ефективне розв'язання складних задач. Сучасні дослідження спрямовані на вдосконалення наявних методів та розробку нових підходів для покращення продуктивності моделей ШІ. Основні напрями включають оптимізацію алгоритмів машинного навчання, розвиток нових методів обробки природної мови та впровадження ефективних алгоритмів для автономних систем.

Список використаних джерел

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. 2016. MIT Press.
2. Bishop C. M. Pattern Recognition and Machine Learning. 2006. Springer.
3. Vaswani A., et al. Attention is All You Need. In Advances in Neural Information Processing Systems. 2017.

4. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 2010. Prentice Hall.
5. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2018. MIT Press.

УДК 512.54:004.738:519.17

Коновалюк І. Л., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки, Вєтров О. С., старший викладач кафедри інформаційних технологій

НУМЕРАЦІЯ ПЕРЕСТАНОВОК

Донецький національний університет імені Василя Стуса, м. Вінниця

У світі математики перестановки є однією з ключових тем, яка відіграє важливу роль у різних математичних теоріях і практичних застосуваннях. Поняття перестановки виникає в контексті упорядкування скінченних множин, де кожен елемент займає своє визначене місце. Ця тема має глибокі корені і знаходиться у центрі досліджень комбінаторики, теорії ймовірностей, теорії графів та ін.

Перестановкою (the permutation) із n елементів називається будь-яка скінченна послідовність, яка одержується внаслідок упорядкування деякої скінченної множини, складеної з n елементів. Число всіх можливих перестановок із n елементів позначається як P_n [1]. Це число дорівнює добутку всіх цілих чисел від 1 до n . Число можливих перестановок із n елементів певного числового набору дорівнює добутку натуральних чисел від 1 до n .

Наприклад, якщо маємо множину з 3 елементів, кількість можливих перестановок буде:

$$P_3 = 3! = 3 \times 2 \times 1 = 6.$$

Розрахунок перестановок дає нам змогу з'ясувати різні варіанти розташування елементів набору, забезпечуючи водночас незмінну їх кількість. Кількість перестановок позначається як P_n , де n – кількість елементів множини. Перестановки обчислюються за формулою $P_n = n!$:

$$\begin{pmatrix} 1 & 4 & 3 & 2 \\ 3 & 1 & 2 & 4 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 4 & 5 \end{pmatrix}.$$

Рис. 1. Приклад перестановок двох наборів чисел

Перестановки – це аранжування елементів у певному порядку. Вони можуть бути корисні в різних областях математики, комп'ютерних наук, криптографії та багатьох інших сферах. Один із видів перестановок – нумерація перестановок. Це спосіб присвоєння унікального числа кожній можливій перестановці набору елементів.

Наприклад, розглянемо набір $\{1, 2, 3\}$. Його можна переставити у шість можливих способів: $\{1, 2, 3\}$, $\{1, 3, 2\}$, $\{2, 1, 3\}$, $\{2, 3, 1\}$, $\{3, 1, 2\}$, $\{3, 2, 1\}$. Нумерація перестановок дає змогу присвоїти кожній із цих перестановок унікальний номер або індекс, що дуже корисно під час обчислень та аналізу.

Формули для обчислення числа можливих перестановок та для визначення індексу конкретної перестановки можуть змінюватися залежно від конкретного