Застосування CNN для моделювання автономних транспортних засобів широко використовується та визнано дуже ефективним [2, 5, 10], тому для даної задачі було обрано саме цю модель.

Проведені дослідження показали, що алгоритми та підходи які використовувались у системі другого рівня автоматизації керування є більш ефективними ніж у системі нульового рівня.

Література

1. Self-drivingcar. URL : https://en.wikipedia.org/wiki/Self-driving_car. Заголовок з екрана.
2. MIT 6.S094: ConvolutionalNeuralNetworksforEnd-to-EndLearningoftheDrivingTask. URL : https://www.youtube.com/watch?v=U1toUkZw6VI&t. Заголовок з екрана.
3. How Safe Is Safe Enough for Self□Driving Vehicles? URL : https://onlinelibrary.wiley.com/doi/full/10.1111/risa.13116. — Заголовок з екрана.
4. Machinelearning. URL : https://en.wikipedia.org/wiki/Machine_learning. Заголовок з екрана.
5. Хайкин, Саймон. X15 Нейронные сети: полный курс, 2-е издание : Пер. с англ. М. : Издательский дом «Вильямс», 2006. 1104 с. : ил. Парал. тит. Англ.
6. A comparative study between LBP and Haar-like features for Face Detection using OpenCV. URL : https://ieeexplore.ieee.org/document/8500159. Заголовок з екрана.
7. Histogram of oriented gradients. URL : https://en.wikipedia.org/wiki/ Histogram_of_oriented_gradients. — Заголовок з екрана.
8. LocalBinaryPatterns. URL : https://en.wikipedia.org/wiki/Local_binary_patterns. Заголовок з екрана.
9. Haar-likefeature. URL : https://en.wikipedia.org/wiki/Haar-like_feature. Заголовок з екрана.
10. A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way. URL : https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Заголовок з екрана.
11. Convolutional neural network. URL : https://en.wikipedia.org/wiki/ Convolutional_neural_network. Заголовок з екрана.

Підсекція фізико-математичних наук (прикладна математика)

УДК 519.688:004.02:004.9

ЕФЕКТИВНІСТЬ АЛГОРИТМУ ПОБУДОВИ ВИПАДКОВОГО ЛАТИНСЬКОГО КВАДРАТУ

А. С. Акопян, О. С. Вєтров, К. М. Довбня

Латинський квадрат порядку n – це квадратна таблиця $n \times n$, складена з n будь-яких елементів таким чином, що кожний елемент повторюється в кожному рядку і кожному стовпці лише один раз. Елементами латинського квадрату можуть бути числа (букви, картинки тощо), але в алгоритмічному сенсі варто розглядати лише випадок числового заповнення таблиці, розуміючи латинський квадрат як квадратну матрицю із вказаною властивістю.

На перший погляд може видатись, що латинський квадрат – це просто математична забавка, але насправді латинські квадрати (як і похідні від них математичні об'єкти) знаходять широке застосування у прикладних задачах плануванні експериментів [1] (наприклад, при у фармакології [2–3], сільськогосподарській справі [4] тощо), при моделюванні економічних процесів [5], в задачах кодування [6], etc. З математичної точки зору, латинські квадрати є комбінаторним аналогом скінчених квазігруп [6], і саме вивчення їх алгебраїчних властивостей може дати підказки для побудови більш ефективних прикладних алгоритмів.

Очевидно, що кожен рядок (стовпчик) латинського квадрату є перестановкою чисел $1..n$. Точна кількість латинських квадратів $L(\mathrm{n})$ визначена конструктивно лише для невеликих значень n. Для довільного n існує відома оцінка [7]

$$\frac{(n!)^{2n}}{n^{n^2}} \le L(\mathrm{n}) \le \prod_{m=1}^{n} (m!)^{\frac{n}{m}}.$$

Виходячи із факторіального порядку зазначеної формули, конкретний алгоритм побудови латинського квадрату треба будувати якомога простішим з точки зору часової складності [8].

Побудова довільного латинського квадрату є досить простою алгоритмічною задачею, способів вирішення якої існує декілька способів, які відрізняються один від одного складністю, залежністю від парності-непарності порядку квадрата тощо.

Найбільш простим способом побудови латинського квадрату є, мабуть, однокрокова циклічна перестановка в кінець стрічки. Суть її у тому, що кожний наступний рядок латинського квадрата виходить з попереднього рядка циклічним зрушенням елементів на деяке постійне число m. При цьому вочевидь m та n мають бути взаємно простими числами.

Як приклад, розглянемо задачу побудови випадкового латинського квадрату довільного порядку. У роботі [9] описаний елементарний спосіб вирішення цієї проблеми. В кожному рядку покроково генерується псевдовипадкове число з діапазону $1..n$. Якщо отримане число вже наявне у відповідному рядку та стовбці матриці, воно генерується знову, поки спроба не виявиться успішною. Пропонований підхід гарантує побудову дійсно випадкового квадрату, але суттєвим недоліком зазначено алгоритму є його часова неефективність. Як зазначають автори [9], при $n \ge 10$ успішного закінчення програми можна і не дочекатись.

В даній роботі пропонується алгоритм генерації випадкового латинського квадрату зі складністю $O(n^2)$, тобто просто заповнення кожного елемента матриці за константу кількість арифметичних операцій. Нижче наведений код реалізації пропонованого авторами алгоритму (Python 3.6.5).

```
1.  from random import randint
2.  def gcd(a, b):
3.      if(b):
4.          return gcd(b, a%b)
5.      return a
6.  def RandPerm(n):
7.      RandomSequence = list()
8.      BasicSequence = [index for index in range(1,n+1)]
9.      N = n - 1
10.     for step in range(n):
11.         index = randint(0, N)
12.         RandomSequence.append(BasicSequence[index])
13.         del BasicSequence[index]
14.         N -= 1
15.     return RandomSequence
16. n = int (input (“Latin square order “))
17. LatinSquare = [[0]*n for index in range(n)]
18. shifts = [1]
19. for number in range(2,n):
20.     if (gcd(number, n) == 1):
21.         shifts.append(number)
```

```
22.  LatinSquare [0] = [index for index in range(1,n+1)]
23.  shiftRandom = shifts[randint(1,len(shifts)-1)]
24.  for rI in range(1,n):
25.     for cI in range(n):
26.        LatinSquare [rI][cI] = LatinSquare [0][(shiftRandom*rI+cI) % n]
```

Тут латинський квадрат – матриця LatinSquare [n][n], список shifts – список допустимих зсувів для конкретного n, shiftRandom – випадково обраний із списку shifts конкретний зсув, функція gcd() (рядки коду 2–5) – дещо спрощена реалізація алгоритму Евкліда.

Особливістю побудованого латинського квадрату буде те, що перший його рядок – послідовність чисел $1, 2..n$. Для того, щоб першим рядком була випадкова перестановка $1..n$ необхідно 22 рядок коду змінити на рядок LatinSquare [0] = RandPerm(n). У даному випадку функція RandPerm() – один із можливих способів генерації довільної перестановки.

Література

1. Табакова І. С. Складання латинських квадратів для застосування у плануванні експериментів. *Системи обробки інформації.* 2017. № 4 (150). С. 52–54.

2. Барчук О. З., Грошовий Т. А., Заліська О. М., Шалата В. Я. Вивчення впливу допоміжних речовин на фармако-технологічні властивості таблеток екстракту чорниці листя, екстракту козлятника трави та таурину, отриманих методом прямого пресування. *Фармацевтичний часопис.* 2018. № 1. С. 47–56.

3. Тригубчак О. В., Грошовий Т. А., Гуреєва С. М. Дослідження впливу природи допоміжних речовин на показники якості шипучих таблеток ацетилсаліцилової кислоти, парацетамолу та аскорбінової кислоти. *Актуальні питання фармацевтичної і медичної науки та практики.* 2018. Т. 11, № 1(26). С. 64–68.

4. Lakić N. The application of Latin square in agronomic research. *Journal of Agricultural Sciences*. 2001. Vol. 46 (1). P. 71–77.

5. Dubnitskiy V. Yu., Kobylin A. M., Kobylin O. A. Застосування латинського квадрату для визначення характеристик обчислювального процесу, що істотно впливають на невизначеність результату обчислень основних типів економічних індексів. *Системи управління, навігації та зв'язку. Збірник наукових праць.* Полтава: ПНТУ, 2017. Т. 2 (42). С. 76–80.

6. Shcherbacov V., Elements of Quasigroup Theory and Applications. *Chapman & Hall/CRC Monographs and Research Notes in Mathematics*, 2017. 598 p.

7. van Lint J. H., Wilson R. M. A Course in Combinatorics. Cambridge University Press, 2001. 604 p.

8. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. Алгоритмы. Построение и анализ. М. : Вильямс, 2013. 1328 с.

9. Провков В. С., Дорохов Д. С. Латинский квадрат и его применение. *Безопасность информационного пространства:* материалы XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Екатеринбург, 2–4 декабря 2013 г. Екатеринбург : Изд-во Урал. ун-та, 2014. С. 239–242.

УДК 519.6:004.02:004.9

ДЕЯКІ ОСОБЛИВОСТІ АНАЛІТИЧНИХ ОБЧИСЛЕНЬ ЗА ДОПОМОГОЮ СИСТЕМ КОМП'ЮТЕРНОЇ АЛГЕБРИ

В. Ю. Василенко, О. С. Вєтров, В. П. Шевченко

При вирішенні прикладних математичних задач, сучасній обсяг даних вимагає застосування комп'ютерних обчислень, як числових, так і аналітичних. Такі системи комп'ютерної алгебри, як Maple та Wolfram Mathematica, допомагають дослідникам не витрачати зайвий час на процес технічних символьних розрахунків, а зосередитись на вирішенні самої проблеми. Окрім основних алгебраїчних перетворень, велику частину символьних обчислень припадає на аналітичне інтегрування та розв'язок диференційних