

**Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики**

**Ministry of Education and Science of Ukraine
Ivan Franko National University of Lviv
Faculty of Applied Mathematics and Informatics**

**МІЖНАРОДНА СТУДЕНТСЬКА НАУКОВА
КОНФЕРЕНЦІЯ З ПРИКЛАДНОЇ МАТЕМАТИКИ ТА
КОМП'ЮТЕРНИХ НАУК МСНКПМКН – 2023**

**INTERNATIONAL STUDENT SCIENTIFIC CONFERENCE
ON APPLIED MATHEMATICS AND COMPUTER
SCIENCES ISSCAMCS – 2023**

**Львів, 4-5 травня 2023
Lviv, May 4-5, 2023**

Метою конференції є сприяння співпраці між студентами, що вивчають споріднені області прикладної математики та комп'ютерних наук.

Головні теми: інформаційні технології та системи, комп'ютерне моделювання, інженерія.

Організаційний комітет:

Дияк Іван – декан факультету прикладної математики та інформатики.

Горlach Віталій – заступник декана факультету прикладної математики та інформатики.

Шевчук Маріан-Северин – голова студентської ради факультету прикладної математики та інформатики.

Шевченко Ірина – студентка факультету прикладної математики та інформатики.

Вовк Михайло – студент факультету прикладної математики та інформатики.

Коропецький Роман – студент факультету прикладної математики та інформатики.

Дикий Назар – студент факультету прикладної математики та інформатики.

Дацюк Олексій – студент факультету прикладної математики та інформатики.

Кульгавець Яромир – студент факультету прикладної математики та інформатики.

Сідукова Софія – студентка факультету прикладної математики та інформатики.

Допко Анна – студентка факультету прикладної математики та інформатики.

Чіх Володимир – студент факультету прикладної математики та інформатики.

Авторський покажчик	Author's index
Bohdan Buhrii	118
Dmytro Omelian	167
Maksym Batiuk	78
Nazar Protsiv	167
Sofiia Shaposhnikova	167
Valerii Veseliak	157
Viktor Seredovych	122
Vitalii Mil	78
Vladyslav Kutsyk	4
Volodymyr Milchanovskyi	127
Yaroslav Mashtaliar	14
Yuriy Muzychuk	122
Анастасія Петришин	26
Анастасія Попович	51
Анатолій Музичук	34
Аркадій Марченко	100
Артем Ядельський	162
Богдан Малець	86
Богдан Петровський	30
Вікторія Чумакевич	76
Віталій Грициндишин	91
Віталій Лучко	131
Віталій Хомин	109
Володимир Самотій	151
Галина Квасниця	37
Дмитро Альянах	64
Ігор Мручко	55
Ірина Шевченко	134
Катерина Тимошук	41
Любов-Василина Вовк	141
Максим Каратаєв	37
Марія Гартованець	7
Марія Малашняк	96
Марія Пономаренко	104
Марія Станович	59
Маркіян Мандзак	82
Микола Баранов	162
Микола Притула	11
Оксана Паляниця	47
Олександр Фунтов	22
Олексій Дацюк	138
Ольга Баранова	71
Ольга Прядко	104
Орест-Теодор Кияк	71
Петро Тарнавський	68
Софія Шувар	37
Софія Яропуд	113
Станіслав Красічинський	147
Тетяна Муковоз	154
Юліана Юрченко	34
Юлія Панас	20
Юрій Віпшовський	11
Юрій Джас	16
Юрій Михайлишин	22

GAMBLER'S RUIN PROBLEM AND ITS APPLICATION

Vladyslav Kutsyk

Ivan Franko National University of Lviv

Faculty of Applied Mathematics and Informatics

vladkutsyk@gmail.com

Gambler's ruin problem

The problem is stated in this way: consider a gambler, who starts with an initial fortune of \$1 and then on each successive gamble either wins \$1 or loses \$1 independent of the past with probabilities p and $q = 1 - p$ respectively. The gambler's objective is to reach a total fortune of N \$, without first getting ruined (running out of money). If the gambler succeeds, then the gambler is said to win the game. In any case, the gambler stops playing after winning or getting ruined, whichever happens first. There is nothing special about starting with 1\$, more generally the gambler starts with i \$ where $0 < i < N$. Taking this into consideration, we can transform this game into a field of two players, A and B, who have a total of N coins between them. Player A starts with i coins, $1 \leq i \leq N-1$, and makes a series of independent 1 coin bets each having probability p of winning 1 coin and q of losing 1 coin. The game ends when Player A loses all of his coins or when his goal of winning N coins is reached. The objective is to determine Player A's ruin probability, that is, the chance of reaching state 0 assuming Player A begins with i coins. The probability that Player A will own all the N coins, $P(A_i)$, if Player A starts with i coins and Player B starts with $N-i$ coins is given by:

$$P(A_i) = \begin{cases} \frac{1 - \left(\frac{q}{p}\right)^i}{1 - \left(\frac{q}{p}\right)^N}, & p \neq 0,5 \\ \frac{i}{N}, & p = q = 0,5 \end{cases} \quad \forall i \in \{1, 2, \dots, N\} \quad (1)$$

and the probability that Player B will own all the N coins, $P(B_i)$, if Player A starts with i coins and Player B starts with $N - i$ coins is given by:

$$P(B_i) = \begin{cases} \frac{1 - \left(\frac{q}{p}\right)^{N-i}}{1 - \left(\frac{q}{p}\right)^N}, & q \neq 0,5 \\ \frac{N-i}{N}, & p = q = 0,5 \end{cases} \quad \forall i \in \{1, 2, \dots, N\} \quad (2)$$

Expanding our formulas into infinite number of general capital and a finite number of player's one:

$$\lim_{N \rightarrow \infty} P_i = 1 - \left(\frac{q}{p}\right)^i, \quad p > 0,5 \quad (3)$$

$$\lim_{N \rightarrow \infty} P_i = 0, \quad p \leq 0,5 \quad (4)$$

To interpret the meaning of (3) and (4), suppose that the gambler, starting with budget of i coins, wishes to continue gambling forever until (if at all) ruined, with the intention of earning as much money as possible. So there is no winning value N , the gambler will only stop, if ruined. What will happen? Formula (3) says, that if $p > 0,5$ (each gamble is in his favor), then there is a positive probability that the gambler will never get ruined but instead will become infinitely rich. Formula (4) says that if $p \leq 0,5$ (each gamble is not in his favor), then with probability one the gambler will get ruined.

Risk insurance business application

Looking through the operational activity of insurance company, we can make assumptions, that this company constantly earns from interest 1\$ per day, but with option to suffer each day a claim against it for the amount \$2 with probability $q = 1 - p$ and independency from previous conditions. Whenever such a claim is suffered, \$2 is removed from the reserve of money. Thus on the n^{th} day, the net income for that day is exactly as in the gamblers' ruin problem: 1 with probability p , -1 with probability q . If the insurance company starts off initially with a reserve of $i\$ \geq 1$, then what is the probability it will eventually get ruined (run out of money)? As we don't have time limits in this case, to find the proper answer, we need to use formulas (3) and (4): if $p > 0,5$ then the probability is given by $\left(\frac{p}{q}\right)^i > 0$, whereas if $p \leq 0,5$ ruin will always occur. This makes intuitive sense because if $p > 0,5$, then the average net income per day is $E(\Delta) = p - q > 0$, whereas if $p \leq 0,5$, then the average net income per day is $E(\Delta) = p - q \leq 0$. So that, insurance company, having average earnings per day and possibilities for the claim calculated, can get the answer to the question: Is its activity still profitable?

References:

1. Lefebvre, M. The Gambler's Ruin Problem for a Markov Chain Related to the Bessel Process. / Lefebvre, M. // Statistics & Probability Letters. - 2008. - № 78(15). – pp. 2314-2320

2. Lengyel, T. Gambler's Ruin and Winning a Series by mGames. / Lengyel, T. // Annals of the Institute of Statistical Mathematics. - 2009a. - № 63(1). – pp. 181-195.
3. Mitzenmacher, M. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. / Mitzenmacher, M., Upfal, E. //Cambridge University Press. – 2005. – p. 298.

МЕТОД ФУНДАМЕНТАЛЬНИХ РОЗВ'ЯЗКІВ ДЛЯ ЧИСЕЛЬНОГО РОЗВ'ЯЗУВАННЯ ТРИВИМІРНОЇ МІШАНОЇ ЗАДАЧІ ДЛЯ РІВНЯННЯ КЛЕЙНА-ГОРДОНА

Марія Гартованець

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

mariia.hartovanets@lnu.edu.ua

Розглядаємо чисельне розв'язування мішаної задачі для рівняння Клейна-Гордона у тривимірній двозв'язній області, використовуючи метод фундаментальних розв'язків (МФР). Невідому функцію апроксимуємо лінійною комбінацією звужень фундаментальних розв'язків, а невідомі коефіцієнти знаходимо методом колокації з врахуванням граничних умов.

Постановка задачі

Нехай задано поверхні Γ_1 і Γ_2 , що обмежують тривимірну область $D \subset \mathbb{R}^3$. Γ_1 - внутрішня поверхня, а Γ_2 - зовнішня. Припускаємо, що для поверхонь задані наступні параметричні подання:

$$\Gamma_i = \{x_i(\theta, \eta) = x_{i1}(\theta, \eta), x_{i2}(\theta, \eta), x_{i3}(\theta, \eta), \theta \in [0, 2\pi], \eta \in [0, \pi]\}, \quad i = \{1, 2\}.$$

Потрібно знайти функцію $u \in C^2(D) \cap C^1(\bar{D})$, яка задовільнятиме рівняння Клейна-Гордона і граничні умови Діріхле та Неймана:

$$\begin{aligned} \Delta u - ku &= 0 \text{ в } D, \\ u|_{\Gamma_1} &= f_1, \quad \frac{\partial u}{\partial \nu}|_{\Gamma_2} = f_2, \end{aligned} \quad (1)$$

де Δ - оператор Лапласа, $k > 0$ - деяка задана константа, f_1, f_2 - достатньо гладкі функції, ν - одиничний вектор зовнішньої нормалі до поверхонь Γ_1, Γ_2 .

Метод фундаментальних розв'язків

Фундаментальний розв'язок рівняння Клейна-Гордона матиме вигляд:

$$\Phi(x, y) = \frac{1}{4\pi a} e^{-ka}, \text{ де } a = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} - \text{Евклідова відстань в } \mathbb{R}^3.$$

Нормальна похідна від фундаментального розв'язку має вигляд:

$$\frac{\partial \Phi(x, y)}{\partial \nu(x)} = \frac{((x-y)^T, \nu(x))}{4\pi a^3}, \quad x \neq y, \quad x, y \in \mathbb{R}^3.$$

Апроксимація розв'язку рівняння (1) є функція вигляду:

$$u_N(x) = \sum_{i=0}^N \lambda_i \Phi(x, y_i),$$

де λ_i – невідомі коефіцієнти, $x \in D, y_j \notin \Gamma_1 \cup \Gamma_2$, тобто вибрані поза областю відомі точки джерела.

Невідомі коефіцієнти шукатимемо за методом колокації. Точки колокації рівномірно розподілюємо на поверхнях Γ_1, Γ_2 , а $\{y_j\}_{j=1}^N$ вибираємо поза областю \bar{D}

З подання базисних функцій бачимо, що вони не мають сингулярних точок в області D , тому апроксимація u_N задовільняє рівняння Клейна-Гордона. Щоб отримати СЛАР для знаходження невідомих коефіцієнтів λ_j , підставляємо u_N у граничні умови Діріхле та Неймана в точках колокації. Система рівнянь матиме наступний вигляд:

$$\begin{cases} \sum_{j=1}^N \lambda_j \Phi(x_{1i}, y_j) = f_1(x_{1i}), \\ \sum_{j=1}^N \lambda_j \frac{\Phi(x_{2i}, y_j)}{\partial v(x)} = f_2(x_{2i}). \end{cases} \quad i = \overline{1, N/2} \quad (2)$$

СЛАР можна подати у матричному вигляді: $A\lambda=b$, де матриця $A=(a_{ij}) \in \mathbf{R}^{N \times N}$ і вектори $\lambda=(\lambda_j) \in \mathbf{R}^N$, $b=(b_j) \in \mathbf{R}^N$ відповідно мають вигляд:

$$\begin{aligned} \lambda &= (\lambda_1, \lambda_2, \dots, \lambda_N)^T, \\ b &= (f_1(x_{11}), \dots, f_1(x_{1m}), f_2(x_{21}), \dots, f_2(x_{2m}))^T \\ A &= \begin{pmatrix} \Phi(x_{11}, y_1) & \dots & \Phi(x_{11}, y_N) \\ \vdots & \ddots & \vdots \\ \Phi(x_{1m}, y_1) & \dots & \Phi(x_{1m}, y_N) \\ \vdots & \ddots & \vdots \\ \frac{\Phi(x_{21}, y_1)}{\partial v} & \dots & \frac{\Phi(x_{21}, y_N)}{\partial v} \\ \vdots & \ddots & \vdots \\ \frac{\Phi(x_{2m}, y_1)}{\partial v} & \dots & \frac{\Phi(x_{2m}, y_N)}{\partial v} \end{pmatrix}, \text{ де } m = \frac{N}{2}. \end{aligned}$$

Дану СЛАР розв’язуємо методом Гауса. Оскільки вихідна задача коректна, то й одержана система рівнянь буде коректною.

Чисельний експеримент

Приклад №1

Нехай поверхні Γ_1 та Γ_2 мають таке параметричне подання (див Рис.1):

$$\Gamma_1 = \{x_1(\theta, \eta) = 0.5(\sin\theta \cos \eta, \sin\theta \sin \eta, \cos\theta), \theta \in [0, 2\pi], \eta \in [0, \pi]\},$$

$\Gamma_2 = \{x_2(\theta, \eta) = 2(\sin\theta\cos \eta, \sin\theta\sin \eta, \cos\theta), \theta \in [0,2\pi], \eta \in [0, \pi]\}.$

$k = 1$ і в ролі точного розв'язку використаємо звуження фундаментального розв'язку:

$u(x) = \Phi(x, y), y = (0,0,0), f_1(x) = \Phi(x, y), x \in \Gamma_1, f_2(x) = \frac{\partial \Phi(x,y)}{\partial v(x)}, x \in \Gamma_2.$

На рис. 2 наведено розподіл точок колокації і джерела.

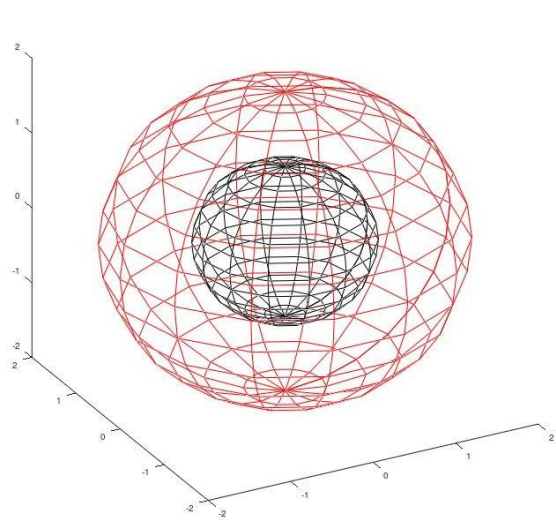


Рис.1 Вигляд області

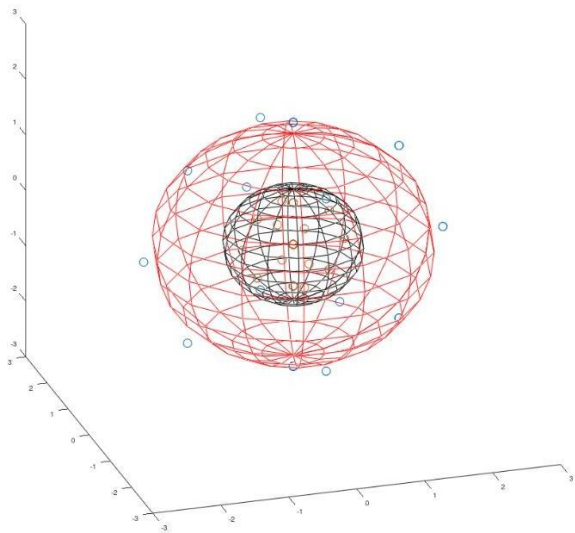


Рис.2 Розподіл точок колокації і джерела

Значення похибки розв'язку задачі у деяких точках із області D для прикладу №1 наведено у таблиці.

N	x=(0.8; 0; 0)	x=(0; 0.7; 0)	x=(0; 0.5; -0.7)
12	0.035716	0.045524	0.027723
24	0.0066158	0.0079360	0.0053581
36	1.3535e-03	1.7094e-03	9.4941e-04
48	3.3792e-04	4.4153e-04	2.5216e-04
60	9.8050e-05	1.2538e-04	7.3515e-05
120	4.6257e-07	7.4590e-07	3.3514e-07

Табл.№1

У роботі наведено алгоритм розв’язування тривимірної мішаної задачі для рівняння Клейна-Гордона, використовуючи МФР. А також продемонстровано результати чисельного експерименту, які підтверджують застосовність даного підходу.

Список літератури

- [1] Bogomolny A., Fundamental solutions method for elliptic boundary value problems. / A. Bogomolny // SIAM Journal on Numerical Analysis// - 1985.
- [2] Stenger F., Numerical Methods Based on sinc and Analytic Functions / F. Stenger // - Vol. 20, Springer Series in Computation Mathematics, Springer-Verlag, New York, 1993 .
- [3] Kress R., Linear Integral Equations, 2nd. ed. / R. Kress // - New-York: SpringerVerlag, - 1989.
- [4] Wei T., Method of fundamental solutions with regularization techniques for Cauchy problems of elliptic operators. / T. Wei, L. Ling YC Hon // Engineering Analysis with Boundary Elements // - 2007.
- [5] Абрамович М., Справочник по специальным функциям /М. Абрамович, И. Стиган.// - М.: Наука, - 1979.

РОЗПІЗНАВАННЯ ТЕКСТУ З КАРТИНКИ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

Віпшовський Ю.А.¹, Притула М.М.²

Львівський національний університет імені Івана Франка, м. Львів,

Факультет прикладної математики та інформатики,

4 рік навчання, 122: Комп'ютерні науки¹

Кафедра дискретного аналізу та інтелектуальних систем²

`vip.cool.ua@gmail.com`

Вступ

За останні роки, завдяки зростанню потужності комп'ютерів та доступності великих даних, нейронні мережі стали дуже популярним інструментом в області комп'ютерного зору та обробки зображень. Зараз, завдяки нейронним мережам, ми можемо розпізнавати об'єкти на зображеннях, аналізувати зміни у відео та багато іншого.

Одним з застосувань нейронних мереж є розпізнавання слів на зображеннях. Ця задача не є простою через те, що зображення – це великий масив пікселів, кожен із яких ніяк не зв'язаний із наступним, тому аналізувати їх дуже складно з допомогою комп'ютера. Також зображення пікселями можуть сильно відрізнятись мати різну якість, гамму та освітлення, але в той самий час мати однакове смислове навантаження(фотографія одного і того ж слова може сильно відрізнятись).

Метою роботи є створення нейронної мережі для виконання задачі розпізнавання слів із зображень, та розробка веб-аплікації для взаємодії із мережею.

Результати дослідження

Для реалізації нейронної мережі для розпізнавання слів англійською мовою обрано мову Python та бібліотеку Keras. Для вирішення цієї проблеми необхідно вирішити такі проблеми:

1. Обробка вхідного зображення

Дана задача полягає у тому щоб маючи звичайну фотографію перетворити її до формату який звичний для нейронної мережі.

Дослідивши датасет із словами я дійшов висновку, що усі вхідні зображення є у відтінках сірого, отже першим кроком обробки зображення буде перетворення його із кольорового до зображення у відтінках сірого. Також у результаті дослідження

було виявлено що у середньому розмір усіх картинок 170x32px, тому вхідне зображення також варто звести до такого розширення. Та зменшити додатково кількість шумів.

Для реалізації цього було використано бібліотеку OpenCV.

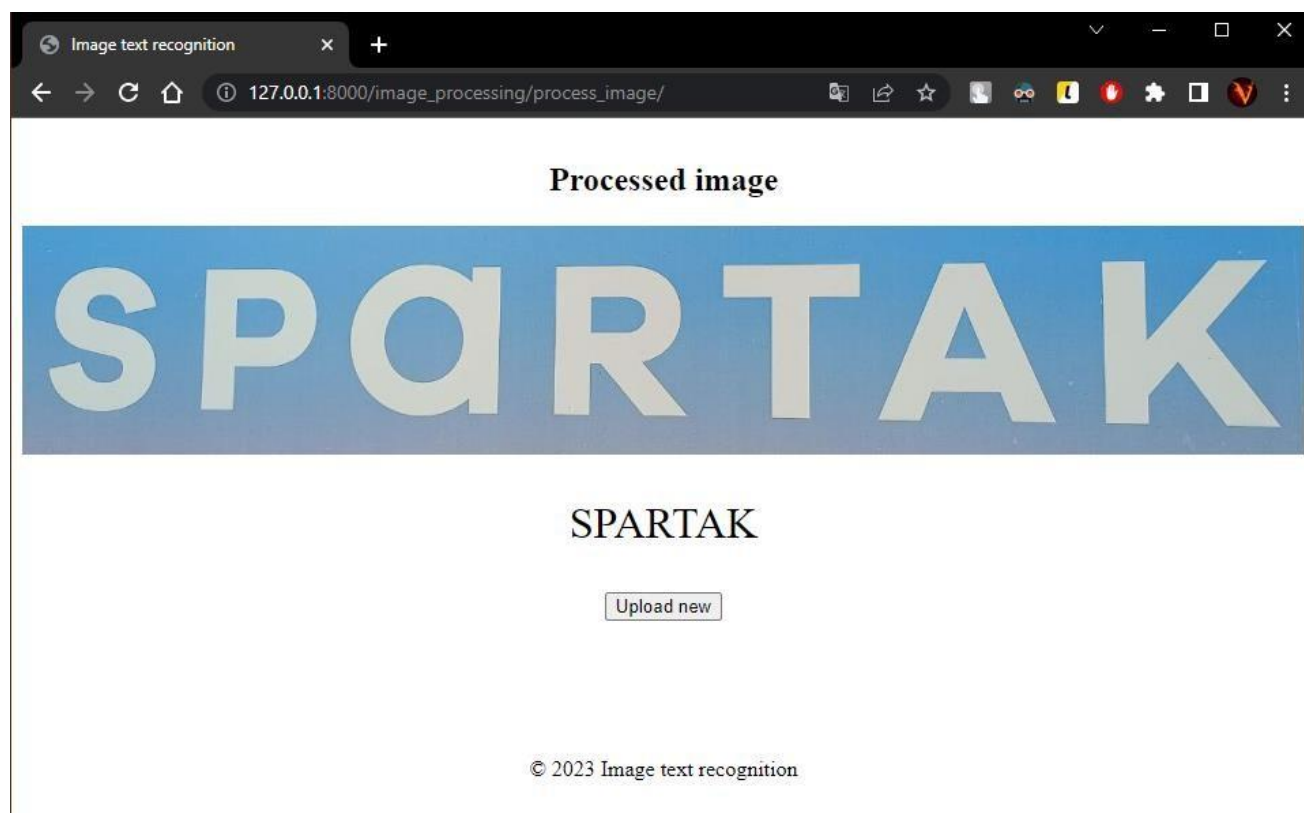
2. Розпізнавання слова

Дана задача полягає у тому щоб маючи фотографію слова повернути у результаті словом у вигляді текстової інформації.

Із попереднього досвіду у розробці нейронних мереж для розпізнавання цифр, я дійшов висновку що на даний момент найкращим типом нейронної мережі для цієї задачі є згорткова нейронна мережа. Така мережа шукає якісь певні риси для кожного класу класифікації. Якщо потім рекурентно по усьому зображенню переглянути ці риси ми можемо оприділити які букви є зображені на фотографії а потім уже об'єднати ці букви у слово. Тоді ми отримаємо CRNN(Convolutional Recurrent Neural Network) Згорткову Рекурентну Нейронну Мережу.

Проблема цієї мережі, у великій кількості параметрів, тому для навчання необхідний великий набір даних, та багато часу.

Користувач може використати веб інтерфейс для доступу до мережі, відправивши фото слова через форму на сайті. Для створення такого інтерфейсу я скористався також мовою Python та бібліотекою Django. З допомогою цієї аплікації користувач може завантажити довільну фотографію і мережа обробивши її поверне текст який написано на ній (Малюнок 1)



Малюнок 1

Висновок

У даній роботі розроблено веб додаток для розпізнавання англійських слів із фотографій за допомогою бібліотек Keras та Django. Був реалізований такий функціонал:

- Клієнт-серверний застосунок: веб аплікація яка при завантаженні сервера завантажує у пам'ять натреновану нейронну мережу і очікує від користувача картинку. Коли картинка приходить застосунок обробляє її, відправляє моделі для передбачення тексту і повертає після обробки користувачу картинку та текст із неї.
- Обробка зображення: зображення зменшується до розширення 170x32px перетворюється із різнокольорового на зображення у відтінках сірого, та обробляється на видалення шумів.
- Нейронна мережа: згорткова рекурентна нейронна мережа, яка створює із зображення масив ознак і рекурентно їх оглядає. У результаті вона отримує на виході зображене слово у вигляді тексту.

Список використаних джерел

1. Min-Ling Zhang, Zhi-Hua Zhou «Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization» // IEEE Transactions on Knowledge and Data Engineering - 2006 – Volume 18 - P. 1338 – 1351
2. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. // MIT Press - 2016 –
3. Towardsdatascience. Harald Scheidl. An Intuitive Explanation of Connectionist Temporal Classification – 2018 - Режим доступу: <https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c>

STREAMLINING GRAPHQL WEB SERVER DEVELOPMENT WITH A SCHEMA-DRIVEN APPROACH

Yaroslav Mashtaliar¹

¹ *Ivan Franko National University of Lviv (Department of Applied Mathematics)*

Ukraine, Lviv

yaroslav.mashtaliar@lnu.edu.ua

Abstract

GraphQL is a popular alternative to REST for building web APIs due to its flexibility, efficiency, and type safety. However, developing a GraphQL web server from scratch can be time-consuming and error-prone. This paper discusses the challenges of automatic code generation for a GraphQL web server based on a user-defined schema in Python.

Keywords: Code generation, GraphQL

Introduction

GraphQL starts with the GraphQL schema, which resembles JSON and contains type definitions. The object type consists of fields, each with its own name, type, and parameters. This creates a graph of relationships between object types.

Two special object types are used to interact with the object types: Query and Mutation. They form the entrypoint to the GraphQL API of the schema, which can be interacted with using the GraphQL query language. For simplicity, a field of the Query type is called "query" and a field of the Mutation type is called "mutation".

Generation of Query and Mutation types

Commonly, for each object type, there is a query for reading all instances and a query for reading a specific instance. Similar to mutations: create a new instance, update an existing instance, or delete an existing instance. It is possible to generate such queries and mutations for a given object type. Those transform type fields to parameters and/or use type primary key as reference to an existing instance. The primary key of a type is defined by annotating fields of the type with the custom @pk directive.

Generation of ORM models

Relational databases are used for persistence on servers, and ORM(Object-Relational Mapping) solutions are adopted to simplify integration. SQLAlchemy ORM is used in this paper.

The main challenge of generating an ORM model from a GraphQL type is the relationships between the types. A many-to-one or one-to-many relationship involves

adding a foreign key and a "sqlalchemy.orm.relationship" attribute to the type on the "many" side. In one-to-one case the foreign key is added to the model of the type that appears "earlier" in the schema. A many-to-many relationship requires an extra association table that links two models.

Generation of Ariadne Components

The Ariadne Python library is used in the paper as the GraphQL API implementation.

A resolver is a function that resolves a field of a type, including queries and mutations. For queries and mutations described earlier, the appropriate resolvers could also be generated. Get all uses `query.all()` and get specific uses `query.get()` of the model. Delete uses `db.session.delete()` on the result of `query.get()`. Create passes the field arguments to the model constructor and persists with `db.session.add()` and `db.session.commit()`. Update assigns the field arguments to the instance obtained via `query.get()`. If the object being mutated has fields referencing other objects, the respective parameters operate with the primary keys of those objects.

Conclusions

A set of principles and rules was developed for generating a GraphQL server based on user-defined GraphQL schemas. Future work will focus on covering more sophisticated and advanced aspects and requirements.

References

1. Eve Porcello, Alex Banks. Learning GraphQL. O'Reilly Media - 2018 — 196p
2. Jason Myers, Rick Copeland. Essential SQLAlchemy. O'Reilly Media — 2015 — 208
3. GraphQL Documentation [Electronic resource]. – Available from: <https://graphql.org>
4. Ariadne Documentation [Electronic resource]. – Available from: <https://ariadnegraphql.org>
5. SQLAlchemy Documentation [Electronic resource]. – Available from: <https://www.sqlalchemy.org>

МОДЕЛЬ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТУ З АУДІО-ФАЙЛІВ

Юрій Джас

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

yriydzhas@gmail.com

Вступ

У зв'язку з масштабним розвитком різних інтернет-сервісів та застосунків, актуальність проблеми розпізнавання тексту з аудіо-файлів є досить значною. Розвиток моделей, подібних цій, може прибрати потребу у ручному вводі тексту, а також добитись зручного використання голосових асистентів. Для таких задач найчастіше використовують механізми машинного навчання. Одним з основних шляхів вирішення цієї проблеми є модель машинного навчання, яка базується на бібліотеці TensorFlow.

TensorFlow - відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови та тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття.

Постановка задачі

Враховуючи особливості даної теми, можна виділити основні завдання:

- Підготовка вибірки даних.
- Опрацювання вибірки даних (тренування).
- Нормалізація отриманих результатів.
- Перевірка моделі на тестовій вибірці даних.

Хід виконання

Першим кроком потрібно вибрати достатньо великий набір даних, щоб отримати точні результати. Для цього була використана відкрита вибірка даних від компанії Google, що має назву Google Speech Commands Dataset [1]. Набір даних містить 65 000 односекундних вимов 30 коротких слів тисячами різних людей. Даний набір призначений для створення простих, але корисних голосових інтерфейсів для додатків, що включають такі поширені слова, як "Так", "Ні", цифри та вказівки.

Для коректної підготовки даних була використана бібліотека для зручної роботи для аудіо та музики Librosa [2]. Вона допомагає виконувати як базові, так і складні завдання, пов'язані з обробкою аудіо, візуалізуючи дискретизований сигнал, амплітуду та звукову спектрограму.

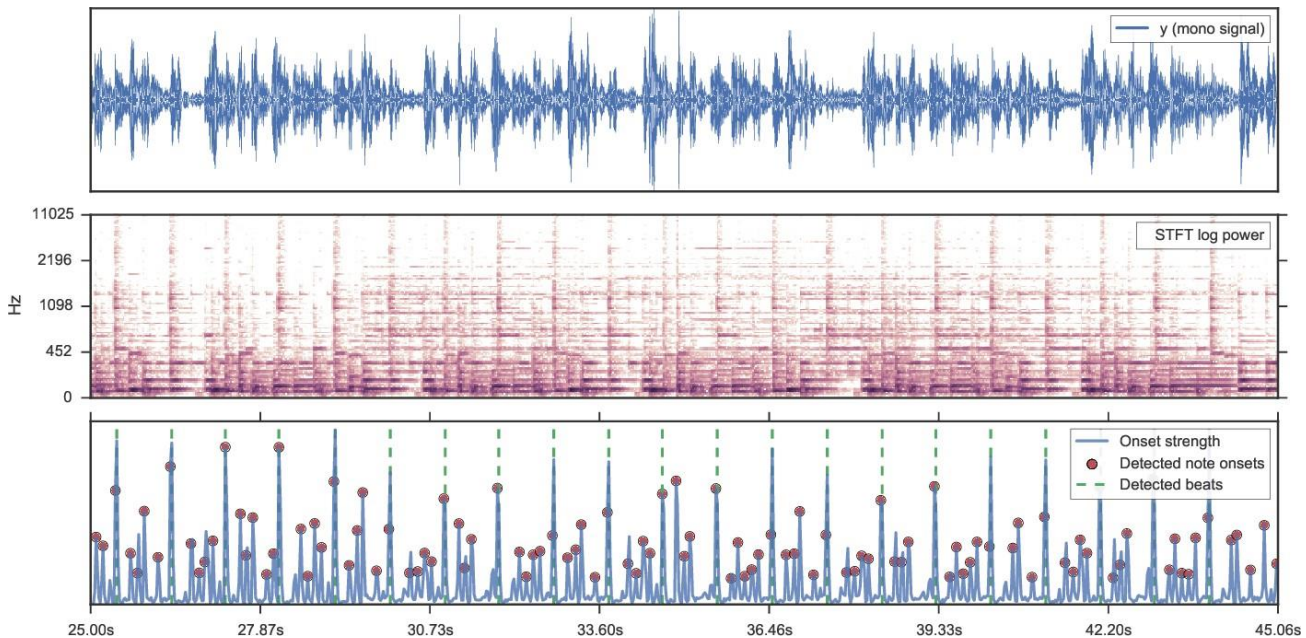


Рис. 1. Аналіз звукового сигналу

Для переведення аудіо-файлу в машинний код було використано MFC (Mel-frequency cepstrum) [3] представлення спектру потужності звуку, яке базується на лінійному косинус-перетворенні логарифмічного спектру потужності.

Для обчислення спектру потужності звуку були використані коефіцієнти MFC (MFCCs). Ці коефіцієнти в сукупності складають MFC. Якщо кепстральний коефіцієнт має додатне значення, то більша частина спектральної енергії зосереджена в низькочастотній області. З іншого боку, якщо кепстральний коефіцієнт має від'ємне значення, це означає, що більша частина спектральної енергії зосереджена на високих частотах. Таке викривлення частоти може забезпечити краще представлення звуку, наприклад, при стисненні звуку, що потенційно може зменшити смугу пропускання і вимоги до зберігання аудіосигналів.

Таким чином, опрацьовані аудіо-записи зберігаються у JSON форматі, де перераховані слова, з набору даних, та обчислені коефіцієнти MFC.

Для побудови моделі була використана згорткова нейронна мережа (CNN). Вона застосовується для класифікації кожного слова з нашого набору, а також для навчання та тестування даних. Ця нейронна мережа відрізняється від інших тим, що інші часто використовують готові дані типу ізольованого слова. Натомість CNN дозволяє обробляти дані з різних шумних середовищ, за різних умов і типів мовлення.

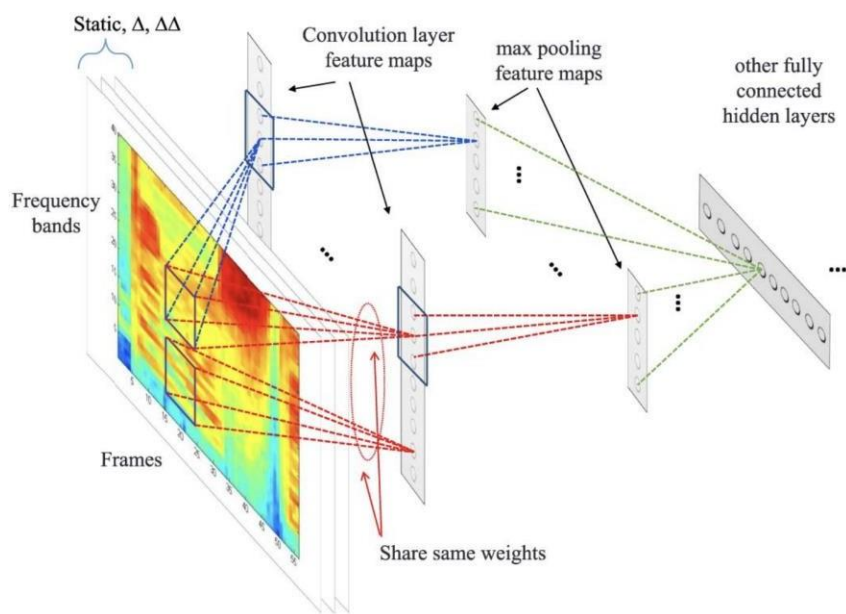


Рис. 2. Приклад роботи CNN

Для тренування моделі використовувалась високорівневий API бібліотеки TensorFlow під назвою Keras [4]. Вона створена з акцентом на розуміння методів глибокого навчання, таких як створення шарів для нейронних мереж, що підтримують концепції форм і математичних деталей. Після тренування було отримано графік того, як збільшувалась точність моделі, та як зменшувались результати функції втрат. Метою функцій втрат є обчислення величини, яку модель повинна намагатися мінімізувати під час навчання.

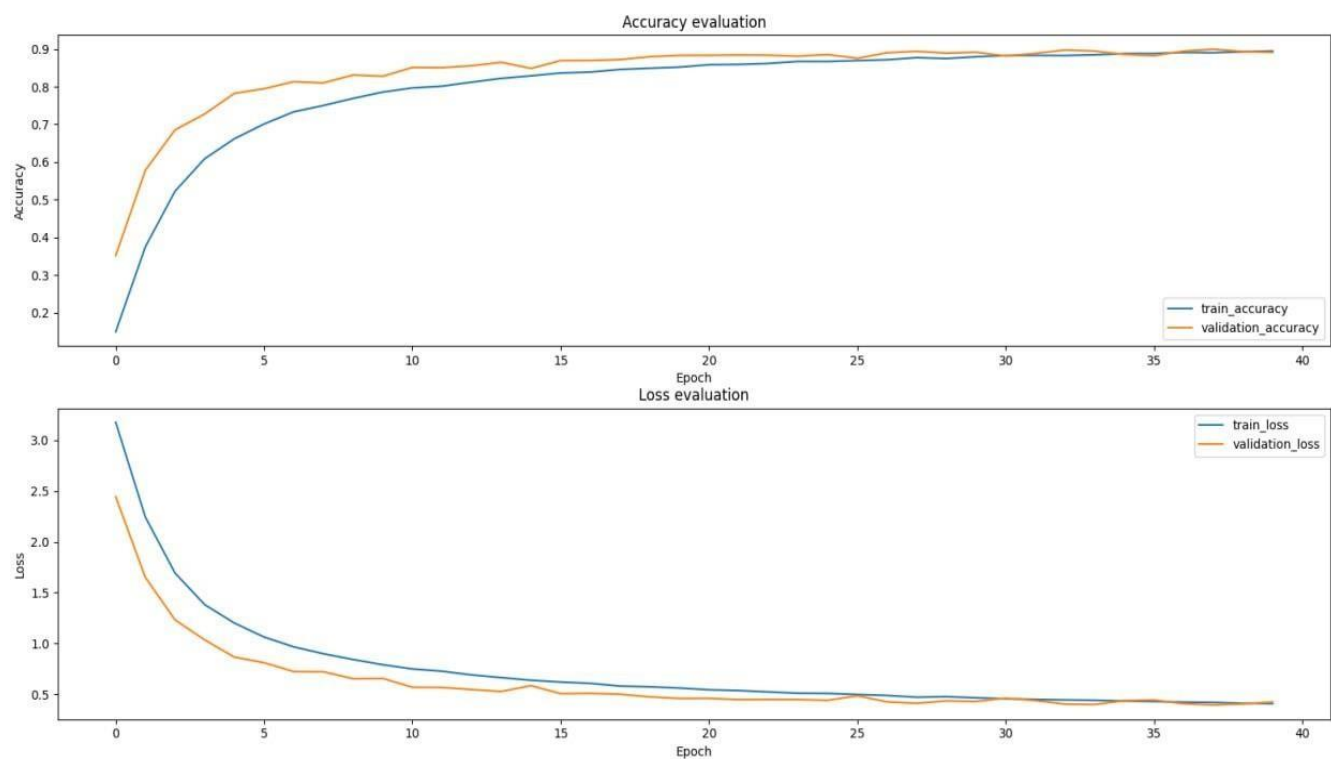


Рис. 3. Ілюстрація результатів тренування моделі

Висновок

У ході дослідження було написано програму на основі вище описаних технологій

та отримано успішні результати роботи моделі машинного навчання для розпізнавання тексту з одnoseкундних аудіо-файлів.

Список використаних джерел:

1. Google Speech Command Dataset [Електронний ресурс]. Режим доступу - <https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html>.
2. Librosa (documentation) [Електронний ресурс]. Режим доступу - <https://librosa.org/doc/latest/index.html>.
3. Librosa MFCCs (documentation) [Електронний ресурс]. Режим доступу - <https://librosa.org/doc/main/generated/librosa.feature.mfcc.html>.
4. TensorFlow Keras Sequential model (documentation) [Електронний ресурс]. Режим доступу - https://www.tensorflow.org/api_docs/python/tf/keras/Sequential.

УДОСКОНАЛЕННЯ РОБОТОТЕХНІКИ ЗА ДОПОМОГОЮ МІКРОСЕРВІСІВ І МЕРЕЖЕВИХ ПРОТОКОЛІВ

Панас Юлія

Львівський національний університет імені Івана Франка

Факультет прикладної математики і інформатики

yuliapanas5@gmail.com

1. **Використання мікросервісів для читання апаратного та програмного забезпечення роботів.** Робота з такими сервісами, як Django, Celery, Redis, React та інші, протоколами MQTT, SocketCAN для підтримки правильної комунікації з девайсами. Застосування Docker compose, інструменту, що дозволяє описати та запускати додатки, які складаються з багатьох контейнерів Docker. Використовуючи цю технологію, я створила мережу контейнерів для розгортання виробничого додатку, що включає базу даних, веб-сервер, та інші компоненти.
2. **Розшифровування бінарних повідомлень прочитаних з каналу шини.** Канал шини - це зв'язок між електронними пристроями, який дозволяє обмінюватись даними у вигляді бітів по певній фізичній мережі. Дані що розшифровуються представлені у вигляді бінарного коду. Встановивши протокол, за яким передаються дані, визначаються типи повідомлень, обробляю дані та дешифрую блоки бінарного коду. Аналізую отримані дані та визначаю їх значення та використання. В загальному, розшифрування бінарних повідомлень з каналу шини дозволяє зрозуміти, які дані передаються між електронними пристроями та що вони означають.
3. **Перепрошивка пристроїв за допомогою протоколу SocketCAN.** Щоб здійснити оновлення програмного забезпечення пристрою потрібно, поставити пристрій в режим завантажувача. Наступним кроком є збереження файлу нового програмного забезпечення. Як тільки пристрій переходить в стан очікування даних, починається запис нових даних. Останнє повідомлення це CRC-код, для перевірки помилок. Якщо ні то програмне забезпечення успішно перепрошилось.

Список літератури

1. SocketCAN - The official CAN API of the Linux kernel" by Oliver Hartkopp and Wolfgang Grandegger

2. "Linux SocketCAN Subsystem Documentation" by Marc Kleine-Budde and Wolfgang Grandegger
3. "Python Socket Programming: Build Socket-based Applications in Python" by Nitin Surya
4. "Python Microservices Development: Build, test, deploy, and monitor microservices using Python, Flask, and Docker" by Tarek Ziade

ОПТИМІЗАЦІЯ РЕЖИМІВ РУХУ ДВОЛАНКОВОГО МАНІПУЛЯТОРА З АКТИВНИМИ І ПАСИВНИМИ ПРИВОДАМИ

Юрій Михайлишин, Олександр Фунтов

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Yuriy.Mykhaylyshyn@lnu.edu.ua

oleksandr.funtov@lnu.edu.ua

Вступ

Робототехніка є однією з галузю що дуже активно розвивається, і дослідження в сфері оптимізації режимів руху роботів є дуже актуальними для розвитку цієї галузі. Особливо це стосується дволанкових маніпуляторів, які мають багато можливостей застосування в промисловості та робототехніці. У цій роботі ми досліджували оптимізацію режимів руху дволанкового маніпулятора з активними і пасивними приводами.

Проблема

Проблема, яку ми досліджуємо, полягає в покращенні продуктивності роботів, які працюють з дволанковим маніпулятором. Для досягнення цієї мети необхідно розробити ефективну стратегію оптимізації режимів руху маніпулятора з урахуванням різних типів приводів. Крім того, необхідно врахувати різні параметри, які можуть впливати на роботу маніпулятора, такі як маса, геометрія, тип матеріалу маніпулятора, маса вантажу. Вирішення цієї проблеми може допомогти покращити ефективність та точність робототехнічних пристроїв, що має велике значення для розвитку промисловості та автоматизації процесів.

Механічна модель та рівняння руху маніпулятора

Маніпулятор здійснює рух у вертикальній площині OXY , який відбувається внаслідок взаємодії сили тяжіння, моментів сил u_1 і u_2 , що генеруються активними приводами (наприклад, електродвигунами), а також моментів сил p_1 і p_2 , які генеруються пасивними приводами. Пасивний привід представимо у вигляді комбінації лінійної пружини і демпфера. Активний u_i та пасивний p_i моменти прикладені у шарнірі O_i , $i = 1, 2$. З метою спрощення математичної моделі маніпулятора вважаємо, що його шарніри є ідеальними, а маса захоплювача (разом із вантажем) зосереджена у точці В. Інерційністю рухомих частин активних і

пасивних приводів нехтуємо, а їх масу враховуємо в масоінерційних характеристиках відповідних ланок.

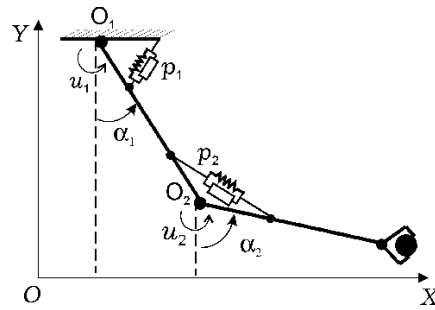


Рис. 1. Модель дволанкового маніпулятора

Введемо позначення: α_i – кут відхилення тіла G_i від вертикалі; m_i , r_i – маса тіла G_i і відстань від його центра маси до точки O_i ; J_i – момент інерції тіла G_i відносно осі шарніра O_i ; $a = |O_1O_2|$, $b = |O_2B|$ – довжини ланок маніпулятора; m – маса вантажу; g – прискорення сили тяжіння; $i = 1, 2$.

Тоді рух досліджуваного маніпулятора можна описати системою нелінійних диференціальних рівнянь

$$\begin{aligned} I_1 \ddot{\alpha}_1 + aK[\ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + \dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2)] + gD \sin \alpha_1 &= u_1 - u_2 - p_1 + p_2, \\ I_2 \ddot{\alpha}_2 + aK[\ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - \dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2)] + gK \sin \alpha_2 &= u_2 - p_2 \\ p_1 &= c_1(\alpha_1 - \theta_1) + k_1 \dot{\alpha}_1, p_2 = c_2(\alpha_2 - \alpha_1 - \theta_2) + k_2(\dot{\alpha}_2 - \dot{\alpha}_1) \end{aligned} \quad (2.1)$$

де c_i , θ_i – коефіцієнт жорсткості пружини та кут її ненапруженого стану, k_i – коефіцієнт в'язкості демпфера, $i = 1, 2$; $I_1 = J_1 + a^2(m_2 + m)$, $I_2 = J_2 + b^2m$, $K = r_2m_2 + bm$, $D = r_1m_1 + a(m_2 + m)$. Тут і нижче крапкою (зверху над величиною) позначено диференціювання за часом t . Зазначимо, що кут θ_1 відраховується від вертикалі, кут θ_2 – від продовження ланки G_1 . Співвідношення (2.1) є рівняннями Лагранжа другого роду, що описують рух дволанкового маніпулятора під дією активних u_1 , u_2 та пасивних p_1 , p_2 керувань. У разі, коли пасивні приводи відсутні ($c_i = k_i = 0$, $i = 1, 2$) і маніпулятор здійснює рух у горизонтальній площині ($g = 0$), рівняння (2.1) співпадають (з точністю до позначень) з рівняннями руху дволанкового маніпулятора.

Опис програмного рішення

Був реалізований програмний додаток, який складається з декількох частин. Перша – бібліотека для прорахунків параметрів керування рухом дволанкового маніпулятора. По-друге в бібліотеці була додана підтримка класів та інтерфейсів

для зручного, простого та швидкого розширення математичних методів, які використовуються під час обрахунку. І третя частина – це інтерфейс користувача, в якому можна порахувати параметри керування рухом дволанкового маніпулятора, як з активним так і з пасивним приводом, а також легко переглядати та маніпулювати експериментами, змінюючи різноманітні параметри. Для розробки використовувались середовище Visual Studio 2022, мова C#, фреймворки NET 6.0 та WinForms.

В поточній реалізації включені реалізації таких математичних методів як знаходження інтегралу з допомогою методі Сімпсона, метод покоординатного спуску.

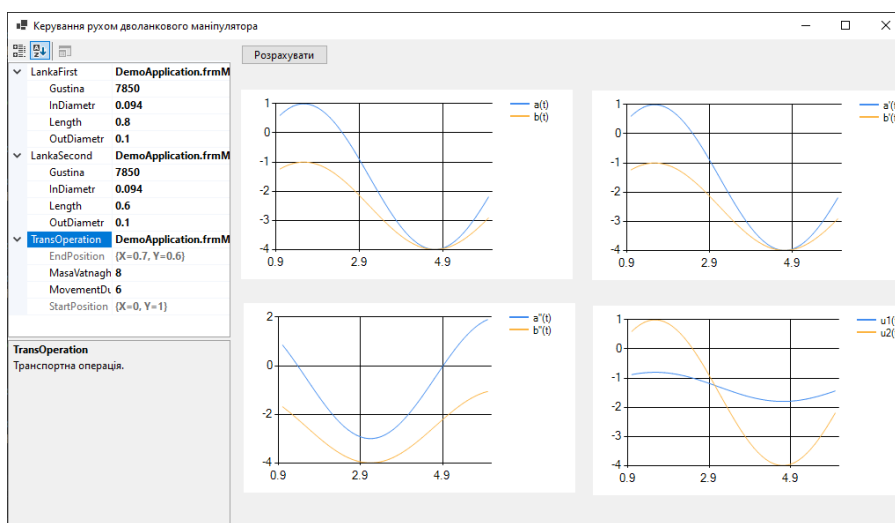


Рис. 2. Вигляд основного вікна для обрахунків руху дволанкового маніпулятора

Висновки

1. У роботі було досліджено метод покоординатного спуску для оптимізації режимів руху дволанкового маніпулятора з активними та пасивними приводами.
2. У подальших ітераціях дослідження ми плануємо вивчити можливість використання інших математичних методів оптимізації параметрів, відмінних від методу покоординатного спуску.
3. Аналізуючи отриману оптимальну траєкторію руху маніпулятора, слід відзначити, що введення в систему пасивних приводів дає можливість більш повно використати взаємодію керуючих моментів маніпулятора і сили тяжіння. На початку першої фази циклічної транспортної операції з'являється реверсний рух маніпулятора, що дозволяє зменшити активні енерговитрати. А у випадку повністю активно керованого маніпулятора при заданому часі підняття вантажу реверсних рухів не виникає. Викликає

інтерес дослідження оптимальної взаємодії між силою гравітації і керуючими моментами з метою оптимізації активних енерговитрат.

4. Розроблені алгоритми оптимізації можуть бути використані для покращення ефективності робототехнічних пристроїв в промислових умовах.
5. Наші дослідження можуть сприяти розвитку робототехнічної галузі та забезпечити використання більш точних та ефективних методів оптимізації режимів руху робототехнічних пристроїв.

Список літератури

1. Демидюк М.В., Гошовська Н.В. Параметрична оптимізація руху дволанкового маніпулятора з використанням ортогональних поліномів // Прикл. проблеми механіки і математики – 2016. – Вип. 14. – С. 168–175.
2. Демидюк М., Гошовська Н. Параметрична оптимізація транспортних операцій дволанкового маніпулятора // Мат. методи та фіз.-мех. поля. – 2017. – 60, №4. – С.151-162.
3. Демидюк М.В., Ширко М.І. Оптимізація режимів руху та параметрів дволанкового маніпулятора з активними і пасивними приводами // Мат. методи та фіз.-мех. поля. – 2007. – 50, № 2. – С. 41–48.

РОЗПІЗНАВАННЯ РУКОПИСНОГО ТЕКСТУ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

Анастасія Петришин

Львівський національний університет імені Івана Франка,

Факультет прикладної математики та інформатики

anastasiia.petryshyn@lnu.edu.ua

У цій публікації ми представляємо дослідження щодо розробки та оптимізації системи Розпізнавання Рукописного Тексту (HTR) за допомогою TensorFlow, широко використовуваного відкритого фреймворку машинного навчання. Запропонована система HTR поєднує потужність згорткових нейронних мереж (CNN) для видобутку ознак та мережі довготривалої короткочасної пам'яті (LSTM) для послідовного навчання з метою розпізнавання та інтерпретації різноманітних рукописних текстів з високою точністю та ефективністю.

Системи розпізнавання рукописного тексту (HTR) відіграють важливу роль у різноманітних застосуваннях у численних областях, завдяки їх здатності інтерпретувати та оцифровувати рукописний текст. Значення систем HTR посилюється зростаючим попитом на ефективну обробку та управління даними в сучасному цифровому віці. До ключових застосувань систем HTR належать: оцифровування документів, поштові та банківські послуги, використання в сфері освіти та охорони здоров'я, інтеграція в допоміжні технології для допомоги особам з порушеннями зору або навчальними вадами.

TensorFlow — це платформа машинного навчання з відкритим кодом, розроблена командою Google Brain Team, яка широко використовується для розробки моделей глибокого навчання в різних сферах, включаючи розпізнавання рукописного тексту (HTR). Фреймворк пропонує багату екосистему інструментів, бібліотек і ресурсів, які спрощують процес створення, навчання та розгортання моделей машинного навчання. Деякі з ключових переваг TensorFlow для програм HTR включають:

- Гнучка та масштабована архітектура: TensorFlow надає гнучку архітектуру, яка дозволяє розробникам розробляти власні моделі нейронних мереж, адаптовані до конкретних вимог завдань HTR. Ця гнучкість дозволяє впроваджувати складні архітектури, такі як комбінації згорткових нейронних мереж (CNN) і довготривалої короткочасної пам'яті (LSTM), які довели свою ефективність у додатках HTR. Крім того, TensorFlow підтримує розподілене обчислення, що дає змогу масштабувати навчальний процес на кількох пристроях і платформах.

- **API високого рівня:** TensorFlow пропонує API високого рівня, такі як Keras, які спрощують процес проектування, створення та навчання моделей глибокого навчання. Ці API абстрагують складність низькорівневих операцій, дозволяючи розробникам зосередитися на основній проблемі НТР і ефективніше експериментувати з різними архітектурами моделей і гіперпараметрами.
- **Попередньо підготовлені моделі та передача навчання:** TensorFlow надає доступ до широкого спектру попередньо підготовлених моделей і дозволяє передати навчання, що може значно скоротити час і ресурси, необхідні для навчання моделей НТР. Використовуючи попередньо навчені моделі, розробники можуть спиратися на наявні знання та точно налаштовувати свої системи НТР для досягнення кращої продуктивності з відносно меншими наборами даних.
- **Прискорення та оптимізація графічного процесора:** TensorFlow підтримує прискорення графічного процесора, що забезпечує швидше навчання та час висновку для моделей НТР. Фреймворк також включає різні інструменти оптимізації, такі як конвертер TensorFlow Lite, який може оптимізувати моделі для розгортання на пристроях з обмеженими ресурсами, наприклад смартфонах або вбудованих системах.
- **Велика спільнота та підтримка:** TensorFlow має велику та активну спільноту розробників, дослідників і професіоналів галузі. Ця спільнота робить внесок у розвиток фреймворку та надає цінні ресурси, такі як навчальні посібники, дослідницькі статті та готові моделі, які можна використовувати для подальшого вдосконалення систем НТР.
- **Сумісність між платформами:** TensorFlow сумісний з різними платформами, включаючи Windows, Linux, macOS і мобільні пристрої. Ця крос-платформна сумісність дозволяє легко розгортати НТР-рішення, створені за допомогою TensorFlow, у широкому діапазоні пристроїв і середовищ.

Стабільність роботи функціоналу, масштабованість і комплексна екосистема TensorFlow роблять його підходящим вибором для розробки систем НТР. Фреймворк дозволяє розробникам експериментувати з різними архітектурами моделей і техніками, одночасно використовуючи переваги прискорення графічного процесора, передачі знань і спільноти підтримки для створення більш точних і ефективних рішень НТР.

Для реалізації аплікації ми використали нейронну мережу, запропоновану у [5], яка складається із 5-ти згорткових шарів (CNN), 2-х повторюваних шарів (RNN) і остаточного рівня часової класифікації Connectionist (CTC).

Ми також можемо розглядати нашу нейронну мережу у більш формальний спосіб як функцію NN (див. рів 1), яка відображає зображення (або матрицю) M розміром $W \times H$ на послідовність символів (c_1, c_2, \dots) довжиною від 0 до L . Текст розпізнається на рівні символів, тому слова або тексти, які не містяться в навчальних даних, також можуть бути розпізнані (за умови, що окремі символи правильно класифіковані).

$$\text{NN: } \underset{W \times H}{M} \rightarrow (\underset{0 \leq n \leq L}{c_1, c_2, \dots, c_n})$$

Рів. 1: нейронна мережа, записана як математична функція, яка відображає зображення M на послідовність символів (c_1, c_2, \dots).

Вхідне зображення подається на шари CNN. Ці шари навчені вилучати релевантні характеристики із зображення. Кожен шар складається з трьох операцій. Першою є операція згортки, яка застосовує до вхідних даних ядра фільтра розміром 5×5 у перших двох шарах і 3×3 у останніх трьох шарах. Потім застосовується нелінійна функція RELU. На кінець рівень об'єднання опрацьовує області зображення та виводить зменшену версію вхідних даних. У той час як висота зображення зменшується на 2 у кожному шарі, додаються карти функцій (канали), так що вихідна карта функцій (або послідовність) має розмір 32×256 .

RNN поширює відповідну інформацію через послідовність із 256 ознак на кожному часовому кроці. Використовується популярна реалізація довгострокової короткочасної пам'яті (LSTM) RNN, оскільки вона здатна поширювати інформацію на великі відстані та забезпечує більш надійні навчальні характеристики, ніж ванільна RNN. Вихідна послідовність RNN відображається на матриці розміром 32×80 . Набір даних IAM складається з 79 різних символів, крім того, для операції CTC потрібен один додатковий символ (порожня мітка CTC), тому для кожного з 32 часових кроків є 80 записів.

Під час навчання нейронної мережі, CTC отримує вихідну матрицю RNN і базовий текст істинності, і він обчислює значення втрати. Під час висновку CTC отримує лише матрицю, і він декодує її в остаточний текст. Довжина основного і розпізнаного тексту може становити не більше 32 символів.

Розроблена аплікація коректно розпізнає близько 65% слів із рукописного тексту на картинках. Для покращення результату планується:

- збільшити розмір набору даних шляхом застосування подальших (випадкових) перетворень до вхідних зображень,

- додати більше шарів CNN,
- замінити LSTM на 2D-LSTM,
- виправлення тексту: якщо розпізнане слово не міститься в словнику, то шукати подібне.

Список літератури

1. B. Vidhale, G. Khekare, C. Dhule, P. Chandankhede, A. Titarmare and M. Tayade, "Multilingual Text & Handwritten Digit Recognition and Conversion of Regional languages into Universal Language Using Neural Networks," *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9418106.
2. D. Peng, L. Jin, Y. Wu, Z. Wang and M. Cai, "A Fast and Accurate Fully Convolutional Network for End-to-End Handwritten Chinese Text Segmentation and Recognition," *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 25-30, doi: 10.1109/ICDAR.2019.00014.
3. K. S. Nathan, H. S. M. Beigi, Jayashree Subrahmonia, G. J. Clary and H. Maruyama, "Real-time on-line unconstrained handwriting recognition using statistical methods," *1995 International Conference on Acoustics, Speech, and Signal Processing*, Detroit, MI, USA, 1995, pp. 2619-2622 vol.4, doi: 10.1109/ICASSP.1995.480098.
4. R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash and A. C. Popat, "A Scalable Handwritten Text Recognition System," *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 17-24, doi: 10.1109/ICDAR.2019.00013.
5. S. Harald "Build a Handwritten Text Recognition System using TensorFlow." [Towards Data Science]. – Режим доступу: <https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5>

ПРО АЛЬТЕРНУЮЧИЙ МЕТОД ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОШІ ДЛЯ РІВНЯННЯ ЛАПЛАСА

Богдан Петровський

Львівський Національний Університет імені І.Франка

Факультет прикладної математики та інформатики

bohdan.petrovskyi@lnu.edu.ua

Вступ

Обернені крайові задачі в процесах теплообміну або електропровідності є досить класичними і поширеними в сучасних дослідженнях фізичних явищ. Однією з таких обернених задач є задача Коші. Для такої задачі відомі межа області розв'язування і дані теплопровідності, джерело тепла або електропровідності і джерела струму, тоді як крайові умови є неповними.

Точніше, умови Діріхле і умови Неймана є задані на частині межі області, тоді як на іншій частині межі - дані відсутні і їх потрібно знайти. Добре відомо, що задачі Коші, як правило, є некоректною за Адамаром, в сенсі відсутності стійкості за вхідними даними. Відповідно, до цих проблем потрібен спеціальний підхід.

У більшості випадків застосовують регуляризацію Тіхонова, яка приводить до розв'язування певної коректної задачі. Однак існують й ітераційні регуляризуючі методи [1,2]. Власне один з них, а саме альтернуючий метод [1,3] використовується у даній роботі.

Постановка задачі та опис алгоритму

Нехай задано деяку двозв'язну область D із зовнішньою межею Γ_2 і внутрішньою межею Γ_1 (рисунок 1.1). Необхідно знайти функцію u , яка задовільняє рівняння Лапласа

$$\Delta u = 0 \quad \text{в } D \quad (1)$$

та крайову умову

$$u = f_2 \quad \text{на } \Gamma_2 \quad (2)$$

$$\frac{du}{dv} = g_2 \quad \text{на } \Gamma_2 \quad (3)$$

Також розглянемо дві мішані крайові задачі для рівняння Лапласа.

Крайова задача Неймана-Діріхле:

$$\Delta u = 0 \quad \text{в } D \quad (4)$$

$$\frac{du}{dv} = h \quad \text{на } \Gamma_1 \quad (5)$$

$$u = f_2 \quad \text{на } \Gamma_2 \quad (6)$$

Крайова задача Діріхле-Неймана:

$$\Delta u = 0 \quad \text{в } D \quad (7)$$

$$u = f_1 \quad \text{на } \Gamma_1 \quad (8)$$

$$\frac{du}{dv} = g_2 \quad \text{на } \Gamma_2. \quad (9)$$

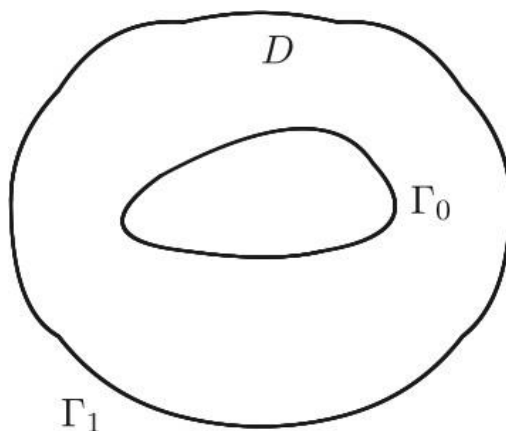


Рисунок 1.1

Опишемо алгоритм альтернуючого методу:

1. Вибираємо початкове h , розв'язуємо задачу (4)-(6) і знаходимо f_1 на Γ_1

2. Розв'язуємо задачу (7)-(9) з вхідними даними, які отримали на вихід з кроку 1 і знаходимо g_2
3. Беремо h рівне вихідним даним з кроку 2, розв'язуємо задачу Н.-Д., знаходимо f_1
4. Розв'язуємо задачу Д.-Н з вхідними даними, які отримали на вихід з кроку 3 і знаходимо g_2
5. Кроки 3 та 4, повторюємо доки не виконується обраний критерій зупинки ітераційного процесу

Коректні мішані крайові задачі на кожній ітерації зводимо до системи інтегральних рівнянь другого роду, які чисельно розв'язуємо методом Нистрьома[4].

Висновки

Для наближеного розв'язування задачі Коші для рівняння Лапласа у двозв'язній області використано альтернуючий метод, який має регуляризуючі властивості. Розв'язки коректних мішаних задач на кожній ітерації подано у вигляді суми відповідних потенціалів з невідомими густинами. Для дискретизації отриманих систем інтегральних рівнянь використовувався метод Нистрьома. Чисельні експерименти показують ефективність даного методу.

В подальшому планується розглянути способи прискорення його збіжності через введення певних релаксаційних параметрів [5,6].

Список використаних джерел:

1. Chapko R., Johansson B.T., Savka Yu. On the use of an integral equation approach for the numerical solution of a Cauchy problem for Laplace equation in a doubly connected planar domain, *Inverse Problems in Science and Engineering*. Vol. 22, 2014. P. 130-149.
2. Baravdish G., Borachok I., Chapko R., Johansson B.T., Slodička M. An iterative method for the Cauchy problem for second-order elliptic equations, *International Journal of Mechanical Sciences*. Vol. 142–143, 2018. P. 216-223.
3. Kozlov V.A., Maz'ya V.G., Fomin A.V. An iterative method for solving the Cauchy problem for elliptic equations, *Zh. Vychisl. Mat. i Mat. Fiz.* 31, 1991. P. 64–74; English transl.: *U.S.S.R. Comput. Math. and Math. Phys.* 31, 1991. P.45–52.

4. Kress R. Linear integral equations. 2nd ed. Heidelberg: Springer-Verlag; 1999.
5. Berdawood K., Nachaoui A., Saeed R., Nachaoui M., Aboud F. An alternating procedure with dynamic relaxation for Cauchy problems governed by the modified Helmholtz equation, *Advanced Mathematical Models & Applications*, Vol.5, 2020. P.131-139.
6. Marin L. Relaxation procedures for an iterative MFS algorithm for two-dimensional steady-state isotropic heat conduction Cauchy problems, *Engineering Analysis with Boundary Elements* Vol. 35, 2011. P. 415-429.

РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ВИЯВЛЕННЯ МЕДИЧНОЇ МАСКИ НА ОБЛИЧЧІ ЛЮДИНИ МЕТОДАМИ КОМП'ЮТЕРНОГО ЗОРУ

Юрченко Юліана, Музичук Анатолій

Львівський національний університет імені Івана Франка,

Факультет прикладної математики та інформатики

yuliana.yurchenko@lnu.edu.ua, anatoliy.muzychuk@lnu.edu.ua

Актуальність теми. Автоматизація моніторингу дотримання маскових рекомендацій має потенціал суттєво підвищити рівень громадського здоров'я, знижуючи ризик поширення COVID-19 та інших інфекційних захворювань.

Метою роботи є створення моделі виявлення об'єктів, яка ефективно працюватиме щодо виявлення медичної маски на обличчі людини.

Модель виявлення медичної маски на обличчі людини

Запропоновано двоетапний підхід до виявлення маски на обличчі людини. На першому кроці алгоритм виявляє обличчя. Для цього використано модель YOLO5Face [1], що має відкритий код. Ця модель реалізує архітектуру YOLOv5 та натренована на наборі даних WiderFace. На другому кроці алгоритм виконує завдання класифікації знайдених облич за двома класами: обличчя, на яких одягнені маски, та обличчя людей, що не носять масок. Для класифікації використано модель VGG19 [2], яку натреновано на наборі цифрових зображень Face Mask Dataset [3].

Підготовка цих двох моделей є незалежною. У режимі тестування їх застосовано до цифрового зображення послідовно (див. рис. 1.)

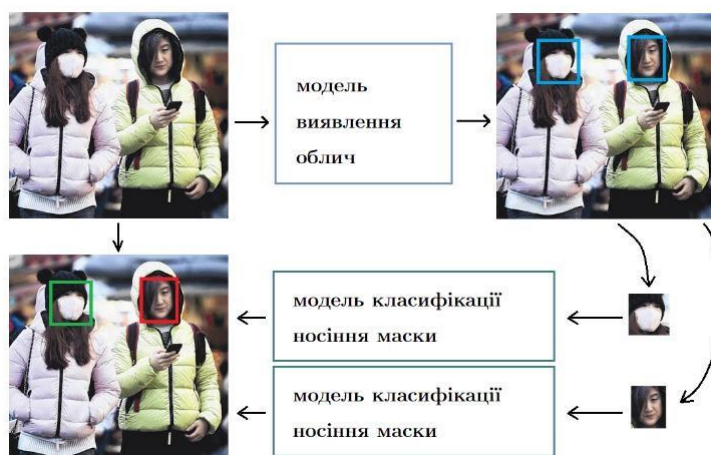


Рис. 1. Схема роботи двоетапної моделі виявлення медичної маски на обличчі людини

Набір зображень Mask Dataset

Mask Dataset [4] містить 853 цифрових зображень різного розміру, на кожному з яких присутні одна людина або кілька людей. Набір зображень використано для тестування моделі виявлення медичної маски на обличчі людини.



Рис. 3. Зразок візуалізації анотації до зображення з набору Mask Dataset (зеленими прямокутниками позначені обличчя з маскою, червоними – без маски)

Метрика результатів виявлення об’єктів

Для оцінки результатів використано метрику mAP (mean average precision).

$$mAP = \frac{1}{2} (AP_0 + AP_1),$$

де AP_0 та AP_1 – average precision (AP) обчислені для виявлення облич людей, що носять та не носять маски відповідно.

Результати виявлення медичної маски на обличчі людини

	Клас облич з масками	Клас облич без масок
AP	0.798	0.383

Табл. 1. Оцінка виявлення об’єктів двох класів

Значення mAP для результатів виявлення медичної маски на обличчі людини для тестового набору даних:

$$mAP \approx (0.798 + 0.383) = 0.590 .$$

Висновки

Складність завдання виявлення медичної маски на обличчі людини для набору Mask Dataset полягає в різній кількості людей на зображеннях, їх розміщенні відносно камери та відносно інших об’єктів. Значення 0.590 за метрикою mAP вказує на ефективне виявлення медичної маски на обличчі людини в умовах реального життя.

Розроблене програмне забезпечення можна використати як прототип комп’ютерної програми для контролю за носінням маски у навчальних закладах та інших громадських місцях.

Список літератури

5. Qi D. YOLO5Face: Why Reinventing a Face Detector arXiv:2105.12931v3 [cs.CV] 27 Jan 2022
6. Simonyan K. Very Deep Convolutional Networks for Large-Scale Image Recognition arXiv:1409.1556v6 [cs.CV] 10 Apr 2015
7. Face Mask Detection. [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset>
8. Mask Dataset. [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ КЛАСИФІКАЦІЇ ВОЄННИХ РУЙНУВАНЬ

Софія Шувар, Галина Квасниця, Максим Каратаєв

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

SOFIIA.SHUVAR@lnu.edu.ua

Із 24 лютого 2022 року невинні бомбардування України російськими військовими завдають неймовірних страждань українському народу. Значна шкода була завдана інфраструктурі українських міст та сіл. Незважаючи на те, що український уряд зосереджується на самій війні, зараз він планує відбудову країни за допомогою глобальної підтримки з боку світової спільноти. Для позовів до міжнародних судів, або індивідуального відшкодування, а також для отримання репарацій та компенсацій для відновлення і відбудови України необхідні зведені докази та реєстр пошкоджених об'єктів з оцінкою втрат, відповідно до обґрунтованої методології. [1]

Процес інспекції територій є досить складним та витратним. Команда аннотаторів повинна проглянути велику кількість ортофотопланів, оцифрувати кожен будівлю та визначити рівень пошкодження. Отже, існує потреба в інструментах і методах, які могли б допомогти швидко оцінити пошкоджені будинки, а саме розробка алгоритмів комп'ютерного зору. Такі алгоритми можуть допомогти автоматично визначати рівень зруйнування будівлі, що дозволить підвищити якість та швидкість процесу оцінки руйнувань. Крім того, такий підхід дозволить значно зменшити кількість людських помилок та підвищити надійність отриманих даних.






Отож метою роботи є створення автоматизованої універсальної системи для класифікації рівня руйнування інфраструктур різноманітних типів на ортофотопланах населених пунктів України, що включає автоматизацію процесу підготовки тренувальних даних, тренування моделі, оцінку точності отриманих передбачень та створення середовища для застосування алгоритмів до нових даних.

Робота виконана в рамках проекту “Росія заплатить”[1] або “RebuildUA”[1], що розробляється під егідою Київської Школи Економіки, на замовлення Офісу Президента України. Тому дані, надані для дослідження, не є публічними.

Результати дослідження

Партнерами проекту “RebuildUA” були надані дані 69 населених пунктів України, що постраждали від бойових дій. Дані для одного населеного пункту складаються з однієї або кількох пар файлів: ортофотоплану формату .geotiff з роздільною здатністю в межах від 5 до 7 см на піксель та .geojson файлу з GPS координати та рівнем зруйнування для кожної будівлі на ортофотоплані. Рівень зруйнування будівель кодується цілим значенням від 1 до 5. Приклади такої класифікації, а також кількість руйнувань кожного типу наведені у таблиці 1.

Табл. 1. Класи руйнувань та їхній розподілі

Класи руйнувань	1	2	3	4	5
	Немає видимих пошкоджень	Ймовірне руйнування	Невеликі пошкодження	Сильне пошкодження	Повністю зруйновано
Фото					
Кількість	180060	10763	5480	1789	3884

Оскільки розподіл рівня зруйнованості будинків не є збалансованим, а розмах варіаційного ряду значно перевищує мінімальну варіанту, було вирішено звузити запропонований постачальниками розподіл та збалансувати дані. Таким чином, класи 2 та 3 об'єдналися у клас 2, а класи 4 та 5 у клас 3.

Оскільки дана робота зосереджена на класифікації пошкоджень кожної будівлі, доцільним є нарізати ортофотоплан на невеликі зображення з будівлею у середині. Оточення будівлі надає критично важливу інформацію та візуальні підказки, які можуть допомогти моделі краще передбачити рівень пошкодження. Шляхом аналізу розподілу площ було встановлено розмір таких зображень 256x256 пікселів. Будівлі, розмір яких перевищує вказаний, були розділені на декілька зображень. Однак, якщо включити занадто багато навколишньої території, це збільшує ризик заплутати модель через додавання до неї фрагментів зображення, які містять кілька будівель різного рівня пошкодження, а також інформацію, невідповідну контексту завдання, таку як зелені насадження, пішоходи, автомобілі тощо. Тому для вилучення об'єктів, які не мають відношення до оцінки збитку, було вирішено додати до триканального зображення четвертий – альфа-маску, де центральній будівлі відповідає значення 256, а фону - 40. В результаті для оновлених рівнів зруйнування 1, 2, 3 отримали 6361, 9100, 9996 зображень відповідно. Ці дані використовуються для навчання моделі.

Модель реалізовано за допомогою бібліотеки PyTorch. Це архітектура згорткової нейронної мережі (CNN [2]), створена аналогічно до мережі VGG (Visual Geometry Group) [3], що складається з двох основних частин: виділення ознак (feature extraction) і класифікатора (Рис. 1.). Частина виділення ознак відповідає за вивчення представлення вхідного зображення, яке фіксує важливі ознаки для класифікації. Вона складається з кількох згорткових (Convolution) шарів із активацією ReLU, які вивчають просторові особливості зображення, за якими слідують шари максимального об'єднання (Max-pooling), які зменшують дискретизацію ознакових карт (feature maps), щоб зменшити їх просторовий розмір. Batch-нормалізація також застосовується до вихідних даних деяких згорткових шарів для покращення стабільності навчання та продуктивності.

Частина класифікації складається з повністю з'єднаних (fully-connected) шарів, які відображають вивчені ознаки у ймовірності класу. Вихід частини вилучення ознак зрівнюється та проходить через два повністю з'єднані шари з активацією ReLU та batch-нормалізацією. Кінцевий вихідний рівень має 3 вузли і обчислює ймовірності класів за допомогою функції активації softmax.

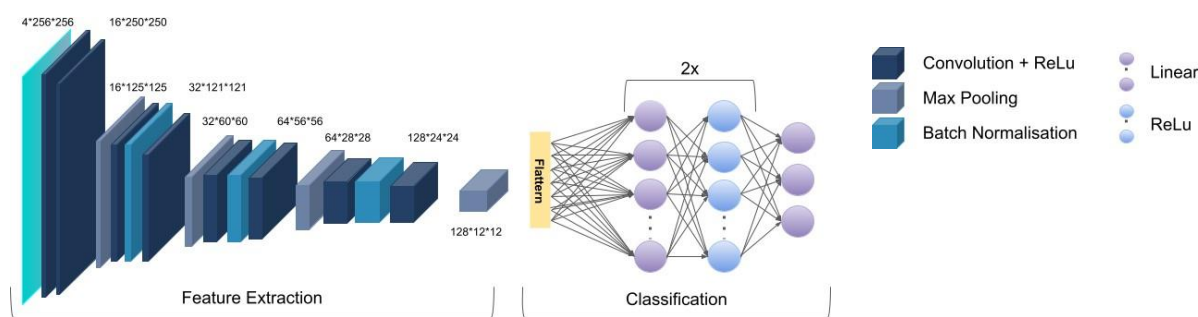


Рис. 1. Архітектура моделі

Під час навчання моделі застосовувалась методологія розширення даних – застосування випадкових перетворень до вхідних даних, що є поширеним підходом, завдяки якому модель краще узагальнює нові дані. Методи розширення, які застосовуються до даного набору даних, складаються з горизонтальних і вертикальних обертань, афінних перетворень та змін яскравості та контрасту.

Модель навчалась із використанням стохастичного градієнтного спуску [4] для оптимізації з початковою швидкістю навчання (η) 0.001. Для тестування було обрано модель, натреновану на 91 епосі. Для оцінки підходу використовувались дані із 5 населених пунктів із більш-менш збалансованою класифікацією, що були вилучені із тренувальних даних. Результати передбачень, отримані на цих тестових даних, демонструють метрики із таблиці 2, а також матриці конфузій (Рис. 2.) для відповідних місцевостей.

Табл. 2. Результуючі метрики

Accuracy	0.982059
Precision	0.981083
Recall	0.982059
F1	0.981381

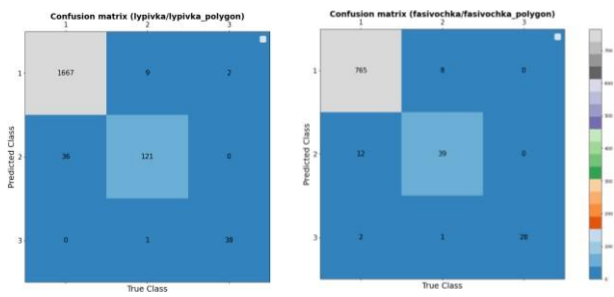


Рис. 2. Матриці конфузій

Запропонований алгоритм машинного навчання демонструє хороші результати отриманих передбачень рівнів зруйнованості будівель та може бути застосованим для практичного аналізу руйнувань.

На даний момент для створення повноцінного класифікатора існує проблема детекції будинків, тому для застосування підходу передбачення рівнів зруйнування нових територій використовуються дані контурів будівель, що є у вільному доступі, а саме надані Bing Maps[5], а також координати будівель отримані від постачальників. Ці дані зберігаються за допомогою розширювача просторової бази даних для об’єктно-реляційної бази даних PostgreSQL - PostGIS, який додає підтримку географічних об’єктів.

Список літератури

1. RussiaPay. (n.d.). <https://damaged.in.ua/>, RebuildUA (n.d.). <https://rebuildua.net/>.
2. Keiron O’Shea, Ryan Nash (2015). *An Introduction to Convolutional Neural Networks*.
3. Karen Simonyan, Andrew Zisserman (2015). *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*.
4. Jing Yang, Guanci Yang (2018). *Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer*.
5. GlobalMLBuildingFootprints.(n.d)
<https://github.com/microsoft/GlobalMLBuildingFootprints>

ДВОКРОКОВИЙ МЕТОД ДЛЯ НЕСТАЦІОНАРНИХ ЗАДАЧ У ТОРОЇДАЛЬНИХ ОБЛАСТЯХ

Катерина Тимошук

Львівський національний університет імені Івана Франка,

Факультет прикладної математики та інформатики

KATERYNA.TYMOSHCHUK@lnu.edu.ua

Постановка задачі

У роботі [1] для чисельного розв'язування крайової задачі Діріхле для рівняння Гельмгольца у випадку замкнених тороїдальних поверхонь запропоновано ефективний алгоритм на основі методу інтегральних рівнянь. У цій роботі ми поширимо цей підхід на випадок нестационарної задачі у тороїдальних областях, обмежених двома поверхнями.

Маємо обмежену тороїдальну область $D \subset \mathbb{R}^3$, утворену обертанням двох плоских кільцеподібних фігур Γ_1, Γ_2 навколо осі. Позначимо її внутрішню границю $\Sigma_1 \in C^2$ та зовнішню границю $\Sigma_2 \in C^2$. Потрібно знайти розв'язок початково-крайової задачі для хвильового рівняння теплопровідності

$$a^2 \frac{\partial^2 u}{\partial t^2} + b \frac{\partial u}{\partial t} = \Delta u \quad \text{в } D \times (0; \infty), \quad (1.1)$$

де $a + b \neq 0$, $a \geq 0$, $b \geq 0$, який задовольняє початкову умову

$$u(x, 0) = a \frac{\partial u}{\partial t}(x, 0) = 0, \quad x \in D \quad (1.2)$$

та крайові умови

$$u = F_i \quad \text{на } \Sigma_i \times [0, \infty), \quad i = 1, 2, \quad (1.3)$$

де для заданих достатньо гладких функцій F_1, F_2 виконуються умови узгодженості:

$$F_i(x, 0) = a \frac{\partial F_i}{\partial t}(x, 0) = 0 \quad \text{на } \Sigma_i, \quad i = 1, 2. \quad (1.4)$$

Часткова дискретизація за часом

Для зменшення вимірності вхідної задачі використаємо перетворення Лагерра за часом, як показано у роботі [2]. Запишемо розвинення Фур'є-Лагерра для функції $u(x, t)$:

$$u(x, t) = \kappa \sum_{n=0}^{\infty} u_n(x) L_n(\kappa t), \quad \kappa > 0, \quad (2.1)$$

де $u_n(x) = \int_0^{\infty} e^{-\kappa t} L_n(\kappa t) u(x, t) dt$, $n = 0, 1, 2, \dots$ – коефіцієнти Фур'є-Лагерра.

Розв'язок (2.1) є розв'язком задачі (1.1)–(1.4) тоді і лише тоді, коли коефіцієнти Фур'є-Лагерра є розв'язками послідовності крайових задач

$$\Delta u_n = \sum_{m=0}^n \beta_{n-m} u_m \quad \text{в } D, n = 0, 1, 2, \dots, \quad (2.2)$$

де $\beta_n = a^2 \kappa^2 (n+1) + b\kappa$, з крайовими умовами

$$u_n = f_{in} \quad \text{на } \Sigma_i, \quad n = 0, 1, 2, \dots, \quad i = 1, 2, \quad (2.3)$$

де f_{1n}, f_{2n} – коефіцієнти Фур'є-Лагерра функцій F_1 та F_2 , відповідно.

Метод інтегральних рівнянь

Розглянемо першу з послідовності крайових задач (2.2)–(2.3):

$$\Delta u_0 - \gamma^2 u_0 = 0 \quad \text{в } D, \quad (3.1)$$

$$u_0 = f_{i0} \quad \text{на } \Sigma_i, \quad i = 1, 2. \quad (3.2)$$

Для розв'язку рівняння (3.1)–(3.2) має місце таке інтегральне подання через потенціали простого шару:

$$u_0(x) = \frac{1}{2\pi} \int_{\Sigma_1} \mu_1(y) \Phi(x, y) ds(y) + \frac{1}{2\pi} \int_{\Sigma_2} \mu_2(y) \Phi(x, y) ds(y), \quad (3.3)$$

де μ_1, μ_2 – невідомі густини і Φ – фундаментальний розв'язок, який має вигляд

$\Phi(x, y) = \frac{e^{-\gamma|x-y|}}{|x-y|}$ Відповідно до властивостей потенціалів для знаходження

густин з крайових умов на Σ_i отримуємо систему інтегральних рівнянь:

$$\begin{cases} \frac{1}{2\pi} \int_{\Sigma_1} \mu_1(y) \Phi(x, y) ds(y) + \frac{1}{2\pi} \int_{\Sigma_2} \mu_2(y) \Phi(x, y) ds(y) = f_{10}(x), x \in \Sigma_1 \\ \frac{1}{2\pi} \int_{\Sigma_1} \mu_1(y) \Phi(x, y) ds(y) + \frac{1}{2\pi} \int_{\Sigma_2} \mu_2(y) \Phi(x, y) ds(y) = f_{20}(x), x \in \Sigma_2 \end{cases}$$

Нехай поверхні Σ_i утворені обертанням замкнених кривих Γ_i . Введемо

циліндричну систему координат і вважатимемо, що $\Gamma_i = \{x_i(t) =$

$(r(t), z(t)), t \in [0, 2\pi]\}$. Отримаємо наступну параметризовану систему:

$$\begin{cases} \frac{1}{2\pi} \int_0^{2\pi} \psi_1(\tau) L(t, \tau) d\tau + \frac{1}{2\pi} \int_0^{2\pi} \psi_2(\tau) L(t, \tau) d\tau = g_1(t), t \in [0, 2\pi] \\ \frac{1}{2\pi} \int_0^{2\pi} \psi_1(\tau) L(t, \tau) d\tau + \frac{1}{2\pi} \int_0^{2\pi} \psi_2(\tau) L(t, \tau) d\tau = g_2(t), t \in [0, 2\pi], \end{cases}$$

де $\psi_i(\tau) = \mu_i(x(\tau)) \sqrt{(r'(\tau))^2 + (z'(\tau))^2}$, $g_i(t) = f_{i0}(x(t))$, $i = 1, 2$,
 $L(t, \tau) = r(\tau) \hat{\Phi}(x(t), x(\tau))$.

Фундаментальний розв'язок в циліндричній системі координат

Введемо позначення:

$$R(t, \tau, \varphi) = ([r(t)]^2 + [r(\tau)]^2 - 2r(t)r(\tau)\cos\varphi + [z(t) - z(\tau)]^2)^{1/2}.$$

Тоді фундаментальний розв'язок має такий вигляд:

$$\hat{\Phi}(x(t), x(\tau)) = \int_0^{2\pi} \frac{e^{-\gamma R(t, \tau, \varphi)}}{R(t, \tau, \varphi)} d\varphi.$$

Розписавши в ряд Тейлора підінтегральну функцію отримуємо:

$$\hat{\Phi}(x(t), x(\tau)) = \sum_{m=0}^{\infty} \alpha_m I_m(t, \tau),$$

$$\text{де } \alpha_m = \frac{(-\gamma)^m}{m!}, \quad I_m(t, \tau) = \int_0^{2\pi} [R(t, \tau, \varphi)]^{m-1} d\varphi.$$

Для функцій I_m справедливі наступні рекурентні співвідношення:

$$I_{m+2} = pI_m - qJ_m, \quad (4.1)$$

$$(m+3)J_{m+2} = (m+1)[pJ_m - qI_m] \quad (4.2)$$

для довільних $m \in \mathbb{N} \cup \{0\}$. Тут $J_m(t, \tau) = \int_0^{2\pi} \cos\varphi [R(t, \tau, \varphi)]^{m-1} d\varphi$, $p(t, \tau) = [r(t)]^2 + [r(\tau)]^2 + [z(t) - z(\tau)]^2$, $q(t, \tau) = 2r(t)r(\tau)$.

Розглянемо випадки парних та непарних індексів m . При непарних значеннях m функції обчислюються за рекурентними формулами (4.1)–(4.2) з початковими значеннями $I_1 = 2\pi$ та $J_1 = 0$.

Розглянемо випадок парних m .

Нехай $K(k) = \int_0^{2\pi} (1 - k^2 \sin^2 \Theta)^{-1/2} d\Theta$, $E(k) = \int_0^{2\pi} (1 - k^2 \sin^2 \Theta)^{1/2} d\Theta$ –

повні еліптичні інтеграли першого та другого роду відповідно, $k = \frac{2q}{p+q}$.

Тоді для парних значень m виконуються співвідношення:

$$I_m = I_m^E E(k) + I_m^K K(k),$$

$$J_m = J_m^E E(k) + J_m^K K(k),$$

де для $I_m^E, I_m^K, J_m^E, J_m^K$ справедливі рекурентні формули (4.1)–(4.2) з початковими

$$\text{значеннями } I_0^E = 0, I_0^K = \frac{4}{\sqrt{p+q}}, J_0^E = \frac{-4\sqrt{p+q}}{q}, J_0^K = \frac{4p}{q\sqrt{p+q}}.$$

Використаємо для повних еліптичних інтегралів наступні представлення:

$$K(k) = K_1(\eta) \ln \left(\frac{1}{\eta} \right) + K_2(\eta), E(k) = E_1(\eta) \ln \left(\frac{1}{\eta} \right) + E_2(\eta),$$

де $\eta = 1 - k^2$, $K_i(\eta), E_i(\eta), i = 1, 2$ – функції, що подаються у вигляді степеневих рядів [3].

Остаточно для фундаментального розв’язку отримуємо наступну формулу:

$$\hat{\Phi}(x(t), x(\tau)) = \chi_1(x(t), x(\tau)) \ln \left(\frac{1}{\eta(t, \tau)} \right) + \chi_2(x(t), x(\tau)), \quad (4.3)$$

$$\text{де } \chi_k(x(t), x(\tau)) = \sum_{m=0}^{\infty} \alpha_{2m} I_{2m}^E(t, \tau) E_k(\eta) + \sum_{m=0}^{\infty} \alpha_{2m} I_{2m}^K(t, \tau) K_k(\eta) +$$

$$\delta_{k2} \sum_{m=0}^{\infty} \alpha_{2m+1} I_{2m+1}(t, \tau), \quad k = 1, 2.$$

Тут χ_k – неперервні функції, δ_{ki} – символ Кронекера.

Метод квадратур

Ядра інтегральних рівнянь з урахуванням вигляду фундаментального розв’язку

$$(4.3) \text{ в загальному подамо у вигляді } L_{li}(t, \tau) = r(\tau) \hat{\Phi}(x_l(t), x_i(\tau)),$$

де $l, i = 1, 2$ позначають криву, на якій знаходяться точки $x_l(t)$ та $x_i(\tau)$

відповідно. Зауважимо, що при $i = l$, $\tau \rightarrow t$ маємо особливість у $\ln \left(\frac{1}{\eta(t, \tau)} \right)$ та

обчислюємо $L_{ll}(t, \tau)$ за формулою:

$$L_{ll}(t, \tau) = L_{ll}^1(t, \tau) \ln \left(\frac{4}{e} \sin^2 \left(\frac{t - \tau}{2} \right) \right) + L_{ll}^2(t, \tau),$$

$$\text{де } L_{ll}^1 = -r(\tau) \chi_1(x_l(t), x_l(\tau)),$$

$$L_{ll}^2(t, \tau) = r(\tau) \chi_2(x_l(t), x_l(\tau)) \ln \left(\frac{4 \sin^2 \left(\frac{t - \tau}{2} \right)}{\eta} \right) + r(\tau) \chi_2(x_l(t), x_l(\tau)).$$

Отже, отримуємо наступну систему інтегральних рівнянь:

$$\left\{ \begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} \psi_1(\tau) [L_{11}^1(t, \tau) \ln\left(\frac{4}{e} \sin^2\left(\frac{t-\tau}{2}\right)\right) + L_{11}^2(t, \tau)] d\tau + \\ & + \frac{1}{2\pi} \int_0^{2\pi} \psi_2(\tau) L_{12}(t, \tau) d\tau = g_1(t), t \in [0, 2\pi] \\ & \frac{1}{2\pi} \int_0^{2\pi} \psi_2(\tau) [L_{22}^1(t, \tau) \ln\left(\frac{4}{e} \sin^2\left(\frac{t-\tau}{2}\right)\right) + L_{22}^2(t, \tau)] d\tau + \\ & + \frac{1}{2\pi} \int_0^{2\pi} \psi_1(\tau) L_{21}(t, \tau) d\tau = g_2(t), t \in [0, 2\pi] \end{aligned} \right.$$

Чисельне розв'язування цієї системи проведемо методом квадратур на розбитті $t_j = \frac{j\pi}{n}, j = 0, \dots, 2n-1$ за формулами:

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi} g(\tau) d\tau &\approx \frac{1}{2n} \sum_{j=0}^{2n-1} g(t_j), \\ \frac{1}{2\pi} \int_0^{2\pi} g(\tau) \ln\left(\frac{4}{e} \sin^2\left(\frac{t_i - \tau}{2}\right)\right) dt &\approx \sum_{j=0}^{2n-1} g(t_j) R_{|j-i|}, \end{aligned}$$

$$R_j = -\frac{1}{2n} \left(\frac{1}{2} + 2 \sum_{m=1}^{n-1} \frac{1}{m} \cos \frac{mj\pi}{n} + \frac{(-1)^j}{n} \right) - \text{взяті з роботи [4]}.$$

Після застосування квадратурних формул і колокації отриманих апроксимаційних рівнянь у квадратурних вузлах отримуємо таку систему лінійних рівнянь відносно $\tilde{\psi}(t_j), i = 1, 2$:

$$\left\{ \begin{aligned} & \sum_{j=0}^{2n-1} \tilde{\psi}_1(t_j) [L_{11}^1(t_i, t_j) R_{|i-j|} + \frac{1}{2n} L_{11}^2(t_i, t_j)] + \frac{1}{2n} \sum_{j=0}^{2n-1} \tilde{\psi}_2(t_j) L_{12}(t_i, t_j) = g_1(t_i) \\ & \frac{1}{2\pi} \sum_{j=0}^{2n-1} \tilde{\psi}_1(t_j) L_{21}(t_i, t_j) + \sum_{j=0}^{2n-1} \tilde{\psi}_2(t_j) [L_{22}^1(t_i, t_j) R_{|i-j|} + \frac{1}{2n} L_{22}^2(t_i, t_j)] = g_2(t_i), \end{aligned} \right.$$

де $i = 0, \dots, 2n-1$. Наближений розв'язок задачі (3.1) отримуємо відповідно до (3.3) за формулою:

$$u_0(x) \approx \frac{1}{2} \sum_{j=0}^{2n-1} \tilde{\psi}_1(t_j) r(t_j) \hat{\Phi}(x, x_1(t_j)) + \frac{1}{2n} \sum_{j=0}^{2n-1} \tilde{\psi}_2(t_j) r(t_j) \hat{\Phi}(x, x_2(t_j))$$

Чисельні експерименти підтверджують ефективність запропонованого алгоритму.

Список літератури

1. Хапко Р. Про чисельне розв'язування граничної задачі Діріхле для рівняння Гельмгольца у випадку замкнених і розімкнутих тороїдальних поверхонь// Вісник львівського університету – Серія прикладна математика та інформатика. 2000. Вип. С. 1–4.
2. Chapko R. On the numerical solution of the Dirichlet initial boundary-value problem for the heat equation in the case of a torus // Journal of Engineering Mathematics 43: 75–87, 2002, С. 76–77.
3. Абрамовиц М., Стиган И. Справочник по специальным функциям. М.: Наука, 1979, С. 404.
4. Kress R. Linear Integral Equations. Springer. Berlin, 1999, С. 231.

ЗАСТОСУВАННЯ МЕТОДУ КОЛОКАЦІЇ ДЛЯ РОЗВ'ЯЗУВАННЯ ІНТЕГРАЛЬНОГО РІВНЯННЯ В ДВОВИМІРНІЙ ОБЛАСТІ

Оксана Паляниця

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

oksana.palianytsia@lnu.edu.ua

1. Вступ

Нехай $D \subset \mathbb{R}^2$ - однозв'язна область, обмежена кривою Γ . Необхідно знайти функцію $u : D \rightarrow \mathbb{R}$, що задовольняє рівняння:

$$-\Delta u + \int_D k(x, y)u(y) dy = f(x), \quad x \in D \quad (1)$$

і крайову умову Діріхле:

$$u(x) = g(x), \quad x \in \Gamma, \quad (2)$$

де $k : D \times D \rightarrow \mathbb{R}$, $f : D \rightarrow \mathbb{R}$, і $g : \Gamma \rightarrow \mathbb{R}$ - відомі функції.

2. Застосування методу потенціалів.

Нехай $\Gamma \in C^2$, $f \in C(D)$ і $k \in C(D \times D)$. Шукаємо класичний розв'язок $u \in C^2(D) \cap C(\bar{D})$ задачі (1)-(2) методом інтегральних рівнянь. Нехай $G(x, y) = \frac{1}{2\pi} \ln \frac{1}{|x-y|}$, $x, y \in \mathbb{R}^2, x \neq y$ - фундаментальний розв'язок рівняння Лапласа, ν - зовнішня нормаль до межі області Γ . Представимо розв'язок задачі (1)-(2) у вигляді суми потенціалів

$$u(x) = \int_D \varphi(y)G(x, y) dy + \int_{\Gamma} \psi(y) \frac{\partial G}{\partial \nu(y)}(x, y) ds(y), \quad x \in D \quad (3)$$

з невідомими функціями густини $\varphi \in C(D)$ і $\psi \in C(\Gamma)$, що є розв'язками системи інтегральних рівнянь

$$\begin{cases} \varphi(x) + \int_D \varphi(y)G(x, y) dy + \int_{\Gamma} \psi(y) \frac{\partial G}{\partial \nu(y)}(x, y) ds(y) = f(x), & x \in D, \\ \psi(x) - 2 \int_D \varphi(y)G(x, y) dy - 2 \int_{\Gamma} \psi(y) \frac{\partial G}{\partial \nu(y)}(x, y) ds(y) = -2g(x), & x \in \Gamma \end{cases} \quad (4)$$

з

$$G(x, y) = \int_D k(x, z)G(z, y) dz.$$

(5)

Теорема 1. Для $\Gamma \in C^2$, $f \in C(D)$ і $k \in C(D \times D)$ система (4) має єдиний розв'язок $\varphi \in C(D)$ і $\psi \in C(\Gamma)$.

Доведення ґрунтується на теорії Ріса для операторного рівняння II роду з компактним оператором [2].

3. Чисельне розв'язування системи (4)

3.1. Параметризація

Нехай межа області D є зірковою, тобто $\Gamma = \{\gamma(t) = r(t)(\cos t, \sin t), t \in [0, 2\pi)\}$, $r > 0$. Визначимо $D^* = D \setminus \{(0, 0)\}$. Тоді існує взаємооднозначне відображення $p(v) = p(\eta, t) = \eta\gamma(t) : \Pi = (0, 1) \times [0, 2\pi) \rightarrow D^*$.

Параметризуємо систему (4) замінюючи змінні у подвійних інтегралах $y = p(\omega) = p(\xi, \tau)$ і в криволінійних інтегралах $y = \gamma(\tau)$ та $x = \gamma(t)$. Отримаємо еквівалентну систему

$$\begin{cases} \varphi(v) + \frac{1}{2\pi} \int_{\Pi} \varphi(\omega) H_{11}(v, \omega) d\omega + \frac{1}{2\pi} \int_0^{2\pi} \psi(\tau) H_{12}(v, \tau) d\tau = f(v), v \in \Pi, \\ \psi(t) + \frac{1}{2\pi} \int_{\Pi} \varphi(\omega) H_{21}(t, \omega) d\omega + \frac{1}{2\pi} \int_0^{2\pi} \psi(\tau) H_{22}(t, \tau) d\tau = g(t), t \in [0, 2\pi) \end{cases} \quad (6)$$

де $\varphi(v) = \varphi(p(v))$, $\psi(t) = \psi(\gamma(t))$, $f(v) = f(p(v))$, $g(t) = 2g(\gamma(t))$.

Ядра в (6) мають вигляд:

$$H_{11}(v, \omega) = 2\pi J(\omega) \int_{\Pi} \tilde{k}(v, q) \tilde{\alpha}(q, \omega) J(q) dq,$$

$$H_{12}(v, \tau) = 2\pi |\gamma'(\tau)| \int_{\Pi} \tilde{k}(v, q) Q(q, \tau) J(q) dq,$$

$$H_{21}(t, \omega) = -4\pi G(\gamma(\tau), \omega) J(\omega),$$

$$H_{22}(t, \tau) = \begin{cases} -2|\gamma'(\tau)| \frac{\gamma(t) - \gamma(\tau)}{|\gamma(t) - \gamma(\tau)|^2} \cdot v(\gamma(\tau)), & t \neq \tau, \\ -\frac{\gamma''(t)}{|\gamma'(t)|} \cdot v(\gamma(t)), & t = \tau. \end{cases}$$

Тут ми ввели позначення $J(\omega) = J(\xi, \tau) = \xi r^2(\tau)$,

$$\tilde{k}(v, q) = k(p(v), p(q)), \tilde{\alpha}(q, \omega) = G(p(q), p(\omega)), p(q) = p(\chi, \theta), Q(q, \tau) = \frac{(p(q) - \gamma(\tau)) \cdot v(\gamma(\tau))}{|p(q) - \gamma(\tau)|^2}$$

3.2. Метод колокації

Введемо сітку $0 < \eta_1 < \eta_2 < \dots < \eta_M < 1$ і $t_j = \frac{j\pi}{N}$, $j = 0, \overline{2N-1}$, базисні поліноми Лагранжа

$\Psi_i(\eta)$, $\eta \in (0, 1)$, $i = \overline{M}$ і тригонометричні базисні поліноми Лагранжа $L_j(t)$, $t \in [0, 2\pi)$,

$j = 0, \overline{2N-1}$. Наблизимо невідомі функції густини з системи рівнянь (6)

наступним чином

$$\varphi(v) = \varphi(\eta, t) \approx \varphi_{MN}(v) = \sum_{i=1}^M \sum_{j=0}^{2N-1} \varphi_{MN}(v_{ij}) \Psi_{ij}(v) \quad (8)$$

$$\psi(t) \approx \psi_N(t) = \sum_{j=0}^{2N-1} \psi_N(t_j) L_j(t) \quad (9)$$

$$v_{ij} = (\eta_i, t_j) \text{ і } \Psi_{ij}(v) = \Psi_{ij}(\eta, t) = \Psi_i(\eta) L_j(t).$$

Після підстановки наближень (8) і (9) в (6) та колокації у точках (η_k, t_m)

отримаємо систему лінійних рівнянь:

$$\begin{cases} \varphi_{k,m} + \sum_{i=1}^M \sum_{j=0}^{2N-1} \varphi_{ij} A_{k,m;i,j}^{11} + \sum_{j=0}^{2N-1} \psi_j A_{k,m;j}^{12} = f_{k,m} \\ \psi_m + \sum_{i=1}^M \sum_{j=0}^{2N-1} \varphi_{ij} A_{m;i,j}^{21} + \sum_{j=0}^{2N-1} \psi_j A_{m;j}^{22} = g_m \end{cases} \quad (10)$$

$$k = \overline{1, M}, m = \overline{0, 2N-1}$$

з невідомими значеннями $\varphi_{k,m} = \varphi_{MN}(v_{k,m})$, $\psi_m = \psi_N(t_m)$ і заданими $f_{k,m} =$

$f(\eta_k, t_m)$, $g_m = g(t_m)$ і матричними коефіцієнтами

$$A_{k,m;i,j}^{11} = (A^{11} \Psi_{ij})(v_{k,m}), \quad A_{k,m;j}^{12} = (A^{12} L_j)(v_{k,m}),$$

$$A_{m;i,j}^{21} = (A^{21} \Psi_{ij})(t_m), \quad A_{m;j}^{22} = (A^{22} L_j)(t_m).$$

3.3. Квадратурні формули

Послабимо особливість у функції $H_{11}(v, \omega)$ адитивним способом. Отримаємо гладкі функції $H_{l,k}, l, k = 1, 2$ і застосуємо квадратури

$$\frac{1}{2\pi} \int_{\Pi} g(v) \approx \frac{1}{2N'} \sum_{i=1}^{M'} \sum_{j=0}^{2N'-1} \omega_i g(\eta'_i, t_j), \tag{11}$$

$$\frac{1}{2\pi} \int_0^{2\pi} g(t) dt \approx \frac{1}{N'} \sum_{j=0}^{2N'-1} g(t_j), \tag{12}$$

де $M', N' \in \mathbb{N}, \eta'_i = \frac{1}{2} + \frac{1}{2} \sigma_i, t_j = \frac{j\pi}{N}, \sigma_i \in \omega_i$ – вузли і ваги квадратури Гауса-Лежандра на відрітку $[-1,1]$.

3.4. Обчислення наближеного розв’язку задачі

Запишемо параметризований вигляд подання розв’язку (3)

$$u(v) = \frac{1}{2\pi} \int_{\Pi} \varphi(\omega) J(\omega) \ln \frac{1}{|p(v) - \gamma(\omega)|} d\omega + \frac{1}{2\pi} \int_0^{2\pi} \psi(\tau) Q(v, \tau) d\tau, v \in \Pi.$$

Тоді чисельне значення розв’язку u можна обчислити наступним чином

$$u_{MN}(v) \approx \sum_{i=1}^M \sum_{j=0}^{2N-1} \varphi_{i,j} B^1_{ij}(v) + \sum_{j=0}^{2N'-1} \psi_j B^2_j(v),$$

$$B^1_{ij}(v) = \frac{1}{2\pi} \int_{\Pi} \Psi_{ij}(\omega) J(\omega) \ln \frac{1}{|p(v) - \gamma(\omega)|} d\omega,$$

$$B^2_j(v) = \frac{1}{2\pi} \int_0^{2\pi} L_j(\tau) Q(v, \tau) d\tau.$$

Логарифмічна особливість в $B^1_{ij}(v)$ послаблюється аналогічно як у функції H_{11} .

Надалі застосовуємо квадратурні формули (11) та (12) для обчислення інтегралів B^1_{ij} та B^2_j .

4. Чисельні експерименти

Нехай область D – одиничний круг з центром в точці $\{(0,0)\}$, $k(x, y) = |x - y|^2$, $f(x) = -\frac{1}{6}(3|x|^2 + 1) - 4, x, y \in D$. Точний розв’язок задачі (1)-(2) $u_{ex}(x) = |x|^2 - 1, x \in D$.

Таблиця 1 містить середньо-квадратичні похибки $\epsilon_{MN} = \|u_{MN} - u_{ex}\|_{L^2(D)}$.

Табл. 1.

M \ N	4	8	16
4	0,215179661	0,049987502	0,046875404
8	0,212328726	0,032318154	0,014598799
16	0,205069577	0,025647925	0,008224231

Висновки

Отже, застосування методу колокації є ефективним для розв’язання заданої крайової задачі для інтегро-диференціального рівняння з частинними похідними. Отримані результати підтверджують теоретичні обґрунтування.

Список літератури

1. Chapko R. On the boundary-domain integrals approach for a partial integro-differential equation / Chapko R., Palianytsia O. // Visnyk of the Lviv University. Series Appl. Math. and Informatics – 2022. – №22. – P. 38-44.

НАСКРІЗНЕ ШИФРУВАННЯ В ПРОВІДНИХ КЛАУД-МЕСЕНДЖЕРАХ

Анастасія Попович

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Anastasiia.Popovych@lnu.edu.ua

Однією зі світових проблем, пов'язаних із шифруванням як технологією, є проблема балансу між потребою в конфіденційності та потребою правоохоронних органів і національної безпеки доступатись до тієї чи іншої інформації. Незважаючи на те, що технологія шифрування може допомогти захистити конфіденційність людей як таку, її також можуть використовувати злочинці та терористи для спілкування та планування незаконних дій, не будучи поміченими правоохоронними органами. Це призвело до дискусії про те, чи повинні уряди мати можливість доступу до зашифрованих даних, щоб запобігати та розслідувати злочини, чи шифрування має повністю захищати, щоб створити безпечні умови для людей і забезпечити конфіденційність їх даних. Питання особливо актуальне в контексті все більшого використання цифрового зв'язку та Інтернету, де стрімко зростає обсяг зашифрованих даних. Він також підкреслює необхідність співпраці між технологічними компаніями, урядами та громадянським суспільством для розробки політики та нормативних актів, які збалансують конкуруючі інтереси конфіденційності, безпеки та правоохоронних органів.

Як приклад у своїй роботі, я використала наскрізне шифрування. Це тип шифрування, коли дані шифруються на пристрої відправника та розшифровуються на пристрої одержувача. Це означає, що дані захищені від перехоплення та зчитування будь-ким, включаючи постачальника послуг, оскільки ключі шифрування та дешифрування зберігаються лише у відправника та одержувача. Це відрізняється від інших типів шифрування, де дані шифруються та розшифровуються серверами, які знаходяться між відправником і одержувачем. Причина, з якої нам потрібне наскрізне шифрування, полягає в тому, що воно забезпечує конфіденційність і безпеку наших комунікацій. Без наскрізного шифрування будь-хто, хто має доступ до сервера, де зберігаються наші повідомлення, потенційно може прочитати їх, будь то уряд, хакер чи сам постачальник послуг. Наскрізне шифрування, з іншого боку, гарантує, що лише відправник і одержувач можуть прочитати повідомлення, що значно ускладнює

доступ до інформації для будь-кого іншого.

Хмарні месенджери, такі як WhatsApp, Signal і Telegram, використовують наскрізне шифрування, щоб забезпечити конфіденційність і безпеку спілкування користувачів. Ці месенджери стають дедалі популярнішими через простоту використання та зручність, а наскрізне шифрування є важливою частиною того, що робить їх безпечними. Окрім захисту конфіденційності користувачів, наскрізне шифрування також допомагає запобігти злому та витоку даних, що може бути дорогою помилкою і завдати шкоди окремим особам чи цілим організаціям.

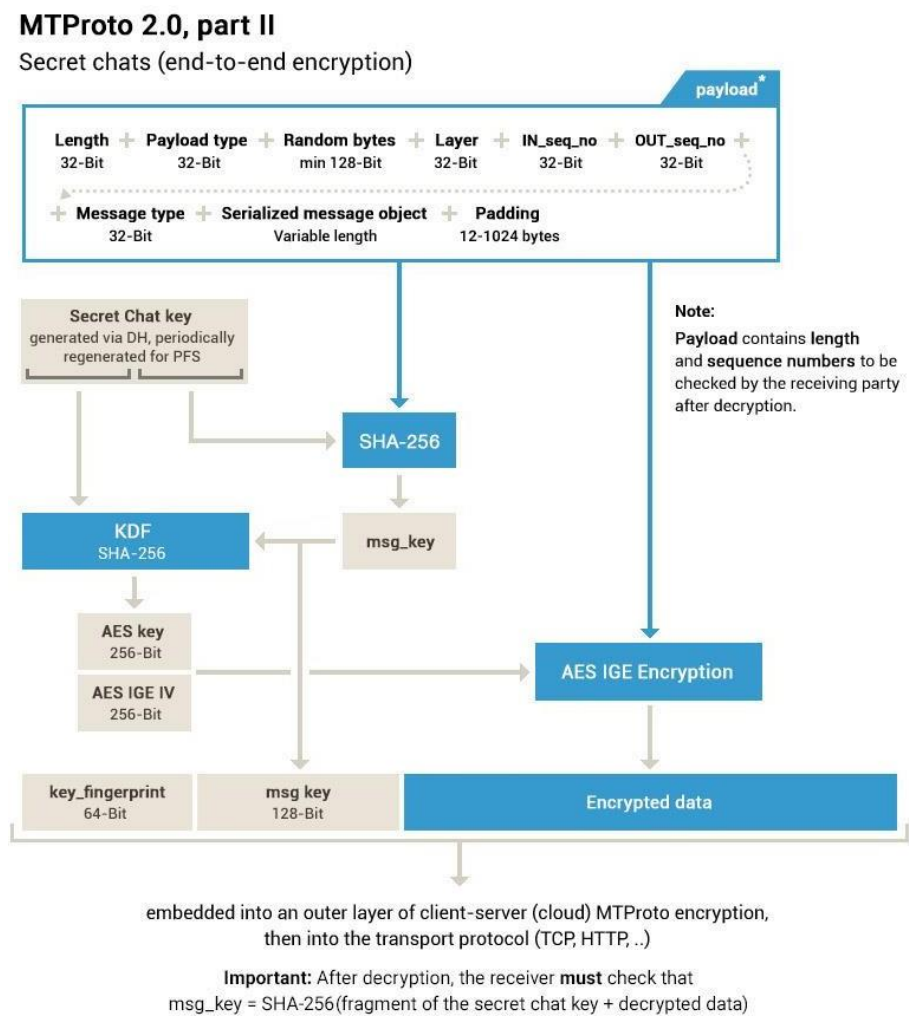
Але дедалі частіше користувачі задаються питанням, чи все таки так легко зламати шифр, закодувати його чи зовсім перехопити дані.

Для даного експерименту я використала Telegram, хмарний багатоплатформовий месенджер, що власне надає можливість використання так званих секретних чатів, які на пряму використовують наскрізне шифрування. Секретний чат – це чат один на один, у якому використовується наскрізне шифрування для захисту повідомлень, фотографій і відео, якими ви ділитесь. Секретні чати не зберігаються на серверах Telegram, доступ до них доступний лише на пристроях відправника та одержувача. Після видалення чату всі повідомлення, фотографії та відео також видаляються з обох пристроїв. У секретному чаті користувачі можуть перевірити ключ шифрування, який використовується для захисту розмови. Telegram використовує алгоритм обміну ключами Діффі-Хеллмана для встановлення безпечного каналу зв'язку між пристроями користувачів. Це означає, що ключ шифрування, який використовується для чату, генерується на пристроях користувачів і ніколи не передається на сервери Telegram.

Моєю ідеєю було досягнути доступу та змінити алгоритм шифрування на будь-який інший, та після численних спроб та пошуку інформації, я дійшла до висновку, що змінити алгоритм шифрування для секретних чатів у Telegram просто неможливо, адже розробники даного месенджеру використали протокол MTProto для секретних чатів, який є власним протоколом шифрування, розробленим Telegram. Цей протокол використовує комбінацію симетричного та асиметричного шифрування, а також обмін ключами Діффі-Хеллмана для захисту зв'язку між користувачами. Протокол MTProto, який використовується в секретних чатах Telegram, спеціально розроблений для забезпечення високого рівня безпеки та продуктивності служби обміну повідомленнями. Хоча деякі користувачі можуть віддати перевагу використанню інших алгоритмів шифрування, таких як AES або RSA, Telegram не надає можливості змінити алгоритм шифрування для секретних чатів. Варто

зазначити, що деякі експерти з безпеки критикували Telegram за використання власного протоколу шифрування, а не загальновизнаного та перевіреного стандарту шифрування. Проте Telegram заявив, що їхній протокол шифрування перевірили незалежні експерти з безпеки, і вони продовжують стверджувати, що їхнє шифрування забезпечує високий рівень безпеки та конфіденційності для їхніх користувачів.

Нижче наведений приклад схеми протоколу MTPROTO:



Як ми бачимо, зламати такий метод шифрування не так вже й просто. Схожа ситуація також і з такими провідними хмарними месенджерами, як Viber, Skype, IMessage та інші, навіть не тому, що в них є або ж немає власного створеного методу, а тому що переважна кількість з них використовує наскрізне шифрування, засноване на протоколі Signal, окрім цього практично всі месенджери не дозволяють користувачам змінювати алгоритм шифрування для приватних розмов. Протокол Signal – це широко визнаний і перевірений протокол шифрування, який використовувався в інших захищених програмах обміну повідомленнями, наприклад Signal і WhatsApp. Він використовує комбінацію симетричного та асиметричного шифрування, а також обмін ключами Діффі-Хеллмана для захисту зв'язку між користувачами, що відповідно має схожі риси із вище наведеним

протоколом MTProto, представленим Telegram.

Отже, ми дійшли до логічного висновку, що хоч ми і не можемо змінити алгоритм шифрування для секретних чатів у Telegram чи будь-яких інших клауд месенджерів, адже протоколи, які використовуються в них, спеціально розроблені для забезпечення високого рівня безпеки та конфіденційності розмов користувачів, і ми фактично повністю захищені від хакерських чи будь-яких інших атак, та питання з доступом правоохоронних органів до несанкціонованих інтернет злочинів все ще залишається відкритим.

Список літератури

1. Шифрування: типи і алгоритми. Що це, чим відрізняються і де використовуються? [Електронний ресурс]. – Режим доступу: <https://hostpro.ua/wiki/ua/security/encryption-types-algorithms>.
2. “End-to-end Encrypted Messaging Protocols” – Автори: Ксенія Єрмошина, Фраческа Мусіані, Гарі Халпін.
3. Наскрізне шифрування. Секретні чати. [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/api/end-to-end>.
4. Telegram. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Telegram>.
5. About end-to-end encryption. [Електронний ресурс]. – Режим доступу: https://faq.whatsapp.com/820124435853543?cms_id=820124435853543&draft=false.

ЗАСТОСУВАННЯ FLUENT API ПРИ СТВОРЕННІ БІБЛІОТЕК

Igor Mruchko

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

ihor.mruchko@lnu.edu.ua

Постановка задачі

Сучасне програмування базується на використанні уже розроблених алгоритмів, які спрощують розробку конкретного завдання. До прикладу, для реалізації веб-застосунку мовою .NET використовують бібліотеку ASP.NET; для створення компілятора використовують вбудовану реалізацію стеку; для сортування списків надають визначені методи чи функції щоб ефективно та швидко виконати цю операцію.

Для того, щоб ознайомитись з правилами використання вбудованих чи уже створених програмних алгоритмів, потрібно читати документацію, дивитись на приклади реалізації від розробників та витратити достатньо часу щоб зрозуміти, як правильно користуватись технологіями.

Один із способів спрощення вхідного порогу у функціонал бібліотеки це використання FluentAPI. Це підхід, коли бібліотеку будують на основі Fluent interface – одного із способів конструювання об'єктно орієнтованого Application Programming Interface[1].

FluentAPI

Fluent interface надає можливість використовувати функціонал використовуючи конструкції, які у своїй концепції близькі до людської мови [2].

Цей підхід застосовується у багатьох бібліотеках: LINQ [3], Entity Framework [4] у мові програмування C#; а також при реалізації паттерну Builder [5].

Розглянемо приклад такого підходу на основі методів, які дозволяють відкрити файл для читання та опрацювання вхідних даних користувачем.

Для початку визначається ланцюжок за допомогою якого можна отримати бажаний результат:

1. встановити директорію до файлу (з перевіркою, чи така директорія існує);
2. встановити назву файлу (з перевіркою, чи такий файл наявний у цій директорії);
3. встановити відповідний тип файлу;
4. встановити обробник вхідних даних;

5. створити зчитувач з файлу.

Для реалізації створимо клас Reader, який буде містити методи та повертати об'єкт цього ж класу після встановлення потрібної інформації (директорія, розширення, назва).

Після цього цей об'єкт можна створити та використовувати.

```
var reader = new Reader()
    .SetDirectory(DIRECTORY)
    .SetName(FILE_NAME)
    .SetExtension("txt")
    .SetHandler(ReplaceSeparator);

reader.Read(out var result);
Console.WriteLine(result);
```

Рис. 1. Приклад налаштування та використання об'єкту Reader.

Проте, коли зміниться послідовність викликів, то при виконанні наступного коду відбудеться помилка у ініціалізації.

```
var reader = new Reader()
    .SetName(FILE_NAME)
    .SetDirectory(DIRECTORY)
    .SetExtension("txt")
    .SetHandler(ReplaceSeparator);

reader.Read(out var result);
Console.WriteLine(result);
```

Рис. 2. Приклад зміни послідовності виклику об'єкту Reader.

Це приведе до неправильного встановлення шляху, оскільки налаштована директорія виглядатиме наступним чином: 'НазваФайлу'/'Директорія'/'Розширення'.

Для того, щоб розробнику про це повідомити під час реалізації функціоналу, використовуючи бібліотечні засоби, потрібно вказати на неправильну послідовність викликів. Розглянемо підхід, у якому при створенні бібліотеки, використовуючи FluentAPI, буде з'являтися підказка про доступні методи на поточному етапі конструювання.

```
var reader = Builders.ReaderBuilder
    .SetDirectory(DIRECTORY)
    .|
    * Equals
    * SetName
    * GetHashCode
    * GetType
    * ToString
    * Equals
    * GetHashCode
```

Рис. 3. Підказка середовища розробки про наявні методи.

Використовуючи цей підхід можна крок за кроком утворити правильний шлях до потрібного файлу.

```
var reader = Builders.ReaderBuilder
    .SetDirectory(DIRECTORY)
    .SetName(FILE_NAME)
    .SetExtension("txt")
    .SetHandler(ReplaceSeparator)
    .Build();

Console.WriteLine(reader.Read());
```

Рис. 4. Повне використання ланцюжка налаштування.

Кроки реалізації

1. Визначити інтерфейси чи абстрактні класи, які будуть відповідати за налаштування потрібного об'єкту.

```
public class ReaderBuilder : IDirectorySetter,
                           IFilenameSetter,
                           IExtensionSetter,
                           IHandlerSetter,
                           IReaderBuilder
```

Рис. 5. Реалізація інтерфейсів класом ReaderBuilder.

2. Кожен інтерфейс повинен містити методи, які можуть бути використані на поточному кроці налаштування у вигляді:

[Наступний етап] [Назва методу] (Параметри);

Де, «Наступний етап» – це інтерфейс чи абстрактний клас, наступного етапу налаштування.

```
public interface IFilenameSetter
{
    2 references
    IExtensionSetter SetName(string fileName);
}
```

Рис. 6. Приклад переходу на наступний етап налаштування.

3. Коли усі налаштування об'єкту було проведено, потрібно створити сам об'єкт.

```
public interface IReaderBuilder
{
    2 references
    ReaderBuilder.Reader Build();
}
```

Рис. 7. Завершення ланцюжка налаштування.

4. Можна також додати точку входження в налаштування, а саме створення об'єкту-конструктора за інтерфейсом/абстрактним класом першого етапу налаштування. Інколи початок створення об'єкту можна винести статичним методом самого класу об'єкта[5].

```
1 reference
public static class Builders
{
    1 reference
    public static IDirectorySetter ReaderBuilder => new ReaderBuilder();
}
```

Рис. 8. Початок налаштування.

Як результат – розробник отримує можливість у правильній послідовності налаштувати розроблену функціональність.

```
var reader = Builders.ReaderBuilder
    .SetName(FILE_NAME)
    .Set
    .Set CS1061: 'Program.IDirectorySetter' does not contain a definition for 'SetName' and no accessible extension method
    .Set 'SetName' accepting a first argument of type 'Program.IDirectorySetter' could be found (are you missing a using directive or
    .Set an assembly reference?)
    .Build();
```

Рис. 9. Помилка компілювання через неправильну послідовність дій.

Висновок

Використання FluentApi при створенні бібліотеки дозволить зменшити час на освоєння цієї бібліотеки розробником; створений код буде легко сприйматись, що

збільшує якість та простоту підтримки проєкту, у якому бібліотека буде застосовуватись; при використанні модифікованого підходу реалізації Fluent interface – з використанням поетапного (обмеженого) ланцюжка викликів – зменшиться ймовірність виникнення помилки під час виконання.

Список літератури

1. Fluent Interface [Wikipedia]. – Режим доступу: https://uk.wikipedia.org/wiki/Fluent_interface
2. Fluent Interface [Java design patterns] – Режим доступу: <https://java-design-patterns.com/patterns/fluentinterface/>
3. INQ in C# [Learn Microsoft] – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/features-that-support-linq#extension-methods>
4. Entity Framework – Fluent Api [EFTutorial] – Режим доступу: <https://www.entityframeworktutorial.net/efcore/configure-one-to-one-relationship-using-fluent-api-in-ef-core.aspx>
5. C++ Builder pattern with Fluent Api [RipTutorial] – Режим доступу: <https://riptutorial.com/cplusplus/example/30166/builder-pattern-with-fluent-api>

ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

Марія Станович

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Mariia.Stanovych@lnu.edu.ua

Вступ

Штучний інтелект є відносно новою галуззю інформатики, яка зародилася в 60-х роках минулого століття. Першими її застосуваннями було вирішення алгебраїчних задач, доведення логічних теорем тощо. Та з кожним роком знаходилося все більше сфер життя, які б потребували прикладного застосування тих чи інших методів штучного інтелекту.

Такою сферою також стало водіння автомобіля, де розпізнавання знаків є головним чинником у дотриманні правил дорожнього руху та відтак безпеки всіх його учасників.

Системи, що вимагають розпізнавання дорожніх знаків

Для початку розглянемо системи, котрі для досягнення тих чи інших поставлених цілей вимагають використання методів штучного інтелекту.

ADAS (Advanced Driver-Assistant System, Удосконалені Системи Допомоги Водію) – це системи, основним призначенням яких є допомога водієві у керуванні автомобілем та паркуванні. ADAS використовує автоматизовані технології, датчики і камери для виявлення перешкод поблизу або помилок водія, що дозволяє відповідно зреагувати та застосувати дії щодо усунення небезпеки. Ці системи також включають в себе розпізнавання навколишніх знаків для виведення цієї інформації безпосередньо на панель автомобіля, а також за можливості, встановлення того чи іншого апаратного обмеження для швидкості руху автомобіля.

Безпілотний (самокерований) автомобіль – автомобіль, який здатен подорожувати без участі людини. Для цього застосовуються різні прилади й датчики, такі як радар, лідар, GPS, одометри, а також методи і технології, такі як комп'ютерний зір. Одним із важливих елементів для визначення обстановки та прийняття системами безпілотного автомобіля рішень є розпізнавання дорожньої розмітки та знаків, адже саме вони вказують на встановлену схему руху, а також обмеження, що можуть діяти на тій чи іншій ділянці дороги.

Дані системи стають все популярнішими та більш затребуваними серед водіїв і їх включають у все ширший ряд автомобілів. Так, згідно із дослідженням Morning Consult, проведеним у 2021 році, 60% опитаних жителів США заявляє про готовність користуватися безпілотним автомобілем зараз або в найближчому майбутньому. Для порівняння, в 2018 році так вважало 54% опитаних.

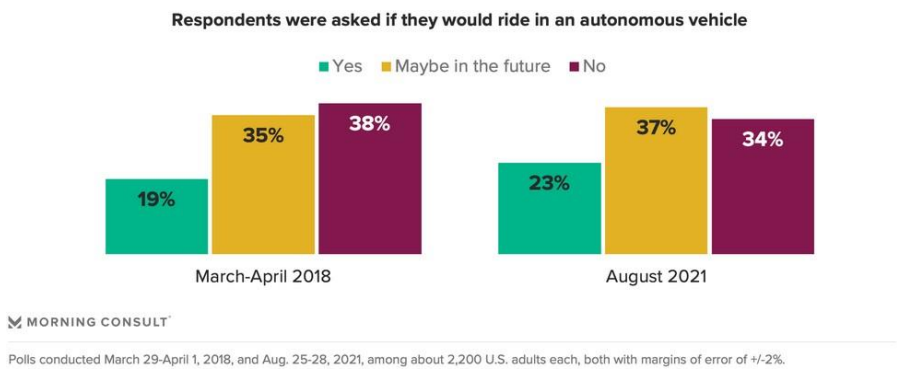


Рис. 1. Результати опитування, проведеного Morning Consult

Етапи розпізнавання дорожніх знаків

Розпізнавання дорожніх знаків потребує використання методів машинного навчання для побудови штучних нейронних мереж. Саме розпізнавання знаків на зображенні є комплексною задачею і складається з двох послідовних етапів – виявлення та класифікація.

На етапі виявлення зображення попередньо обробляється, покращується і сегментується відповідно до характеристик знаків, зокрема таких властивостей, як колір та форма. Результатом є сегментоване зображення, що містить ділянки, які потенційно можуть бути розпізнані як дорожні знаки. Ефективність і швидкість виявлення є факторами, які відіграють важливу роль протягом усього процесу, оскільки вони зменшують простір пошуку і вказують лише на потенційні області.

На етапі класифікації кожен кандидат тестується за певним набором ознак, щоб визначити, чи належить він до групи дорожніх знаків, а потім класифікується в різні групи на основі його ознак. Ці ознаки підібрані таким чином, щоб підкреслити відмінності між класами. Форма знаку та колір відіграють центральну роль на цьому етапі, і знаки класифікуються на різні класи, такі як трикутні, круглі, прямокутні, квадратні та восьмикутні, з іншої ж сторони – сині, білі, жовті, червоні та зелені. Аналіз символів уможлиблює наступний крок класифікації: аналізуючи форму символу і текст всередині знаку, можна легко визначити клас, до якого він належить.

Процес розпізнавання дорожнього знаку можна умовно представити у вигляді наступної схеми:

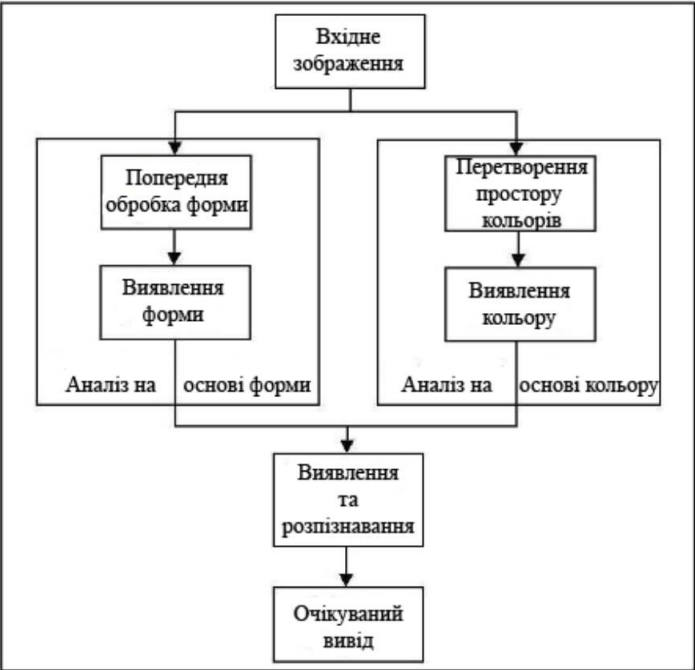


Рис. 2. Розпізнавання дорожніх знаків у вигляді схеми

Потенційні складнощі

Через складне оточення доріг та обстановку навколо них дорожні знаки знаходяться в різних умовах, відповідно, їхнє розпізнавання стикається з цілим рядом наведених нижче труднощів.

Колір знаку з часом тьмяніє внаслідок тривалого впливу сонячного світла та реакції фарби із забруднюючими речовинами в повітрі, що ускладнює визначення класу цього знаку за кольором, а також ідентифікацію символів, що знаходяться в них.

На видимість дорожніх знаків впливають погодні умови, такі як туман, дощ, хмари та сніг, а також інші параметри, такі як час доби, наявність чи відсутність дорожнього освітлення, наявність чи відсутність світловідбиваючого покриття на знаку, тіні від інших об’єктів тощо.

Наявність об’єктів, схожих за кольором та/або формою на дорожні знаки також може завадити та спотворити результати аналізу зображення.

Знаки можуть бути дезорієнтовані, пошкоджені або частково закриті іншими об’єктами навколо них.

Розмір знаку залежить від відстані між камерою та самим знаком, рідше розмір знаку може з самого початку не відповідати стандарту.

Якщо зображення отримано з автомобіля, що рухається, то воно часто страждає від розмиття при русі та вібрації автомобіля.

Приклади знаків, розпізнавання яких може ускладнитися через наявність на зображенні вище описаних факторів:



Рис. 3. Зображення, на яких розпізнавання знаків є ускладненим

Аналіз на основі кольору

Колір є важливою властивістю дорожніх знаків, оскільки при їх розробці враховувався фактор розрізнення та впізнавання того чи іншого знаку; так, більшість інформаційних знаків є синього кольору, а більшість заборонних – білими з червоним обрамленням. Таким чином колір є важливим джерелом інформації для розпізнавання знаку. Крім того, обробка кольору може значно зменшити кількість помилкових крайових точок, які виникають при низькорівневій обробці зображень.

Камери, встановлені на транспортних засобах, що рухаються, створюють RGB-зображення; колірний простір RGB організований як декартова система координат з осями x , y і z , представленими R , G і B відповідно, і оскільки ці три колірні координати сильно корелюють, зміни інтенсивності навколишнього освітлення можуть впливати на систему RGB, переміщуючи групи кольорів у білі або чорні кути, і в багатьох випадках таке зображення стає менш придатним для розпізнавання кольору знаку. Натомість необхідно відокремлювати інформацію про колір від інформації про яскравість шляхом перетворення колірного простору RGB в інший колірний простір. У літературі описано багато колірних просторів, зокрема HSI, HSB, $L^*a^*b^*$, YIQ і YUV. Для розпізнавання дорожніх знаків система HSI є найпоширенішою з причин, описаних вище, але інші системи кольорів також використовуються для цієї задачі.

Аналіз на основі форми

Хоча для розпізнавання дорожніх знаків зазвичай використовується колір, дорожні знаки також можуть сприйматися за формою.

Аргументом на користь використання інформації про форму для розпізнавання дорожніх знаків є те, що кольори не стандартизовані в різних країнах – не зважаючи на те, що Віденська конвенція про дорожні знаки та сигнали була прийнята ще у 1968 році, станом на 2023 рік всього лиш 69 країн ратифікували її, причому серед списку немає таких великих країн з густим автомобільним трафіком, як США, Японія та Індія. Відтак системи розпізнавання,

що залежать від кольору, часто доводиться налаштовувати при переїзді з однієї країни в іншу.

Іншим аргументом є те, що кольори змінюються залежно від денного світла та відбивних властивостей. У ситуаціях, коли важко отримати інформацію про колір, наприклад, у сутінках або вночі, розпізнавання за формою є хорошим варіантом.

Висновки

В даній роботі було оглянуто прикладне застосування методів штучного інтелекту на прикладі розпізнавання дорожніх знаків, систем, що опираються на дане розпізнавання, етапи розпізнавання дорожніх знаків, котрі потрібно брати до уваги при проектуванні та розробці штучних нейронних мереж, складнощі, котрі потрібно враховувати при розробці та навчанні такої мережі, а також два основних способи аналізу зображення та класифікації – за кольором та формою знаку.

Список літератури

1. H. Fleyeh, Road and traffic sign detection and recognition / H. Fleyeh, M. Dougherty //ResearchGate.– 2006. С. 645-649.
2. Convention on Road Signs and Signals. United Nations Treaty Series (in English, French, Chinese, Russian, and Spanish). Vol. 1091. Vienna. –1968.
3. Amid Tesla’s Autopilot Probe, Nearly Half the Public Thinks Autonomous Vehicles Are Less Safe Than Normal Cars. [Електронний ресурс]. – Режим доступу: morningconsult.com/2021/09/02/autonomous-vehicles-safety-consumer-interest-polling/
4. Advanced driver-assistance system [Електронний ресурс]. – Режим доступу: en.wikipedia.org/wiki/Advanced_driver-assistance_system
5. Self-driving car [Електронний ресурс]. – Режим доступу: en.wikipedia.org/wiki/Self-driving_car

РЕАЛІЗАЦІЯ ФІЗИКО-ІНФОРМОВАНОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗВ'ЯЗУВАННЯ РІВНЯНЬ У ЧАСТКОВИХ ПОХІДНИХ

Дмитро Альянах

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

dmytro.alianakh@lnu.edu.ua

Вступ

Завдання розв'язування рівнянь у часткових похідних має важливе значення в різних галузях науки та інженерії, таких як механіка рідин, електромагнетизм, матеріалознавство, фінансова математика тощо. Традиційні чисельні методи, такі як методи скінченних різниць, методи скінченних елементів (МСЕ) [1] та методи граничних елементів, мають ряд обмежень, зокрема, надлишкову обчислювальну складність для великих задач, проблеми з адаптивністю до нерегулярних сіток або високої роздільної здатності та обмеження, пов'язані з оптимізацією параметрів.

У зв'язку з виникненням та розвитком глибоких нейронних мереж, зокрема з використанням фізико-інформованих нейронних мереж (Physics-Informed Neural Networks, PINNs) [2], постає можливість вирішення задач розв'язування рівнянь у часткових похідних з новим підходом. PINNs комбінують переваги глибокого навчання з використанням апріорних знань про фізичні співвідношення. Це описують системою, що дозволяє підвищити ефективність і точність розв'язків.

Опис алгоритму PINN

Вхідні дані: рівняння та початково-крайові умови.

- Задати дві вибірки для тренування T_f і T_b для рівняння і граничних/початкових умов.
- Включити обчислення необхідних похідних для рівнянь у нейромережу.
- Визначити функцію втрат для рівняння і умов на границях.
- Тренувати PINN для знаходження наближеного розв'язку, мінімізуючи функцію втрат.

Тобто ми задаємо всі вхідні дані, тренуємо нейронну мережу і передбачуємо значення в точках. Якщо ми змінюємо який-небудь з вхідних параметрів, то нейромережу потрібно тренувати спочатку.

Функцію втрат будемо обчислювати як суму середньоквадратичних похибок (mean squared error, MSE) значень рівняння, початкових і/або крайових умов.

Середньоквадратична похибка обчислюється за формулою:

$$MSE(e) = \frac{1}{n} \sum_{i=1}^n e_i^2, \tag{1}$$

де e – вектор довжиною n (n – кількість точок для навчання), отриманий після підставляння отриманого наближення з нейронної мережі в рівняння або початково-крайові умови.

Функція втрат буде мати вигляд:

$$L = MSE_e + MSE_b + MSE_i, \tag{2}$$

де MSE_e – похибка задовільнення рівняння, MSE_b – похибка задовільнення крайових умов, MSE_i – похибка задовільнення початкової умови.

Програмна реалізація

PINN була реалізована на мові програмування Python [3] з використанням бібліотеки для машинного навчання TensorFlow v2 (TF) [4]. Автодиференціювання в TF було застосоване для обчислення втрат і обчислення всіх необхідних похідних від відомих і шуканої функції. Як алгоритм оптимізації був вибраний Adam (Adaptive Moment Estimation), який є методом стохастичного градієнтного спуску, який використовується для оновлення параметрів моделі на основі оцінки першого та другого моментів градієнта.

Результати обчислювальних експериментів

Розглянемо нелінійну крайову задачу стаціонарної дифузії

$$-(k(x, p)p')' = f, x \in (a, b), \quad p(0) = p(1) = 0, \tag{3}$$

де $k(x, p)$ — функція, яка характеризує неоднорідності матеріалу, $p(x)$ — шукана функція, $f(x)$ – задана права частина. У цьому прикладі взято $k(x, p) = 1 + xp$, $f(x) = -1$.

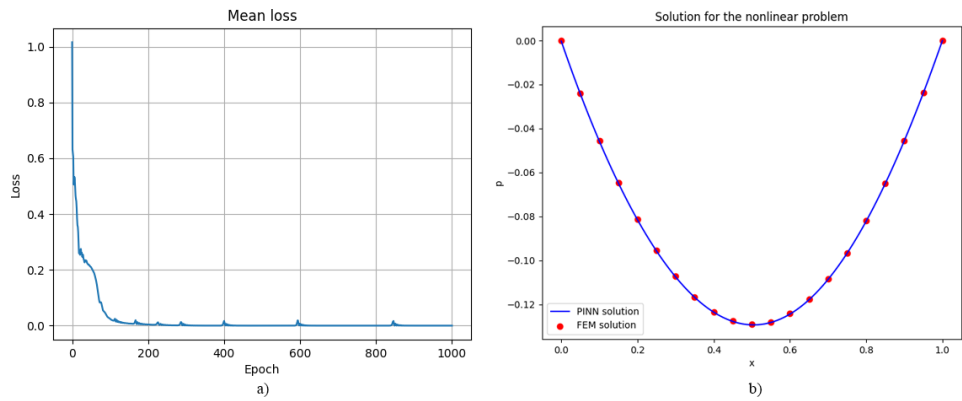


Рисунок 1 На рисунку а) зображено втрати відносно епохи тренування. На рисунку б) зображено наближений розв'язок (3), отриманий PINN після 1000 епох навчання

Епоха	Втрати
0	1.015

300	8.1e-04
500	1.96e-05
700	2.67e-05
1000	2.55e-05

Таблиця 1 Величина втрат на епохах тренувань

Як параметри для PINN було взято: швидкість навчання 0.005, 100 точок для навчання, взяті з рівномірного поділу $[0; 1]$. Перцептрон з одним входом і одним виходом має 3 прихованих Dense шари (або повністю з'єднані шари) з 50-ма нейронами кожен.

Нейронна мережа натренувалася на 1000 епохах за 22 секунди, що є в рази швидше порівняно з розв'язком отриманим методом скінченних елементів. Для MSE, реалізованому за допомогою MATLAB [5], розв'язок отриманий для 21-ї вузлової точки, рівномірно взятих з $[0; 1]$, було взято кусково-лінійні базисні функції, для обчислення інтегралів було взято рівномірне розбиття з 200 точками. Результат, отриманий MSE з тим самим порядком точності за 1 хвилину і 18 секунд, що є довше більш ніж у три рази, а при збільшенні кількості вузлів дискретизації різниця ставатиме ще більшою.

Однак, не при всіх повторних тренуваннях PINN збігався до такої точності, при сталій кількості епох. Деколи нейронна мережа попадає в локальний мінімум, що не дозволяє їй досягти кращого результату.

Висновки

Застосування фізико-інформованої нейронної мережі виявилось ефективним методом для розв'язування рівнянь в часткових похідних, оскільки вони навчаються безпосередньо з диференціальних рівнянь, забезпечуючи задовільну точність навіть при обмеженому наборі даних для тренування.

Однак PINN має деякі обмеження. Зокрема, їхнє навчання може виявитися складним і потребувати значних обчислювальних ресурсів. Крім того, методи оптимізації, які використовуються для навчання PINN, можуть мати труднощі зі збіжністю або потрапити в локальні мінімуми, що призводить до неправильних результатів.

Попри ці обмеження, застосування PINN до рівнянь з частковими похідними відкриває нові можливості для моделювання та аналізу фізичних систем.

Використані джерела

[1] Савула Я. Г. Числовий аналіз задач математичної фізики варіаційними методами. — Львів: видавничий центр ЛНУ імені Івана Франка, 2004. — 221 с

- [2] Vadyala S.R. / S.N. Betgeri, N.P. Betgeri // Physics-informed neural network method for solving one-dimensional advection equation using PyTorch. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S2590005621000515>
- [3] Офіційний сайт документації мови програмування Python. [Електронний ресурс]. – Режим доступу: <https://www.python.org/doc/>
- [4] Офіційний сайт документації бібліотеки TensorFlow. [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/guide>
- [5] Офіційний сайт документації MATLAB. [Електронний ресурс]. – Режим доступу: <https://www.mathworks.com/help/matlab/>

ОГЛЯД МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ CHATGPT ДЛЯ РОЗВ'ЯЗУВАННЯ АЛГОРИТМІЧНИХ ЗАДАЧ

Тарнавський П. В.

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

petro.tarnavskyi@lnu.edu.ua

У роботі представлено огляд застосування ChatGPT [1] для розв'язування як простих, так і складних алгоритмічних задач. Оцінено приблизні межі, в яких доцільно використовувати ChatGPT. Про актуальність та важливість теми свідчить те, що багато учнів та студентів використовують його для розв'язування домашніх завдань. Викладачі задумуються над тим що потрібно робити з ChatGPT [2], оскільки студенти можуть використовувати його для порушення академічної доброчесності. Тому важливо розуміти можливості цього штучного інтелекту.

У роботі використали класичний підручник [3] і з нього виділили декілька стандартних алгоритмів та підходів:

- пошук найкоротшої відстані в зваженому та не зваженому графі
- каркас найменшої ваги
- динамічне програмування
- алгоритм Кнута-Морріса-Пратта
- китайська теорема про остачі
- пошук опуклої оболонки

Для кожного завдання штучний інтелект мав знайти розв'язок і написати код програми. Приклад завдання подано на рисунку 1а. Під час цього огляду стандартних алгоритмів зібрали статистику на скільки коректні програми робить ChatGPT, вона зображена на рисунку 1б.

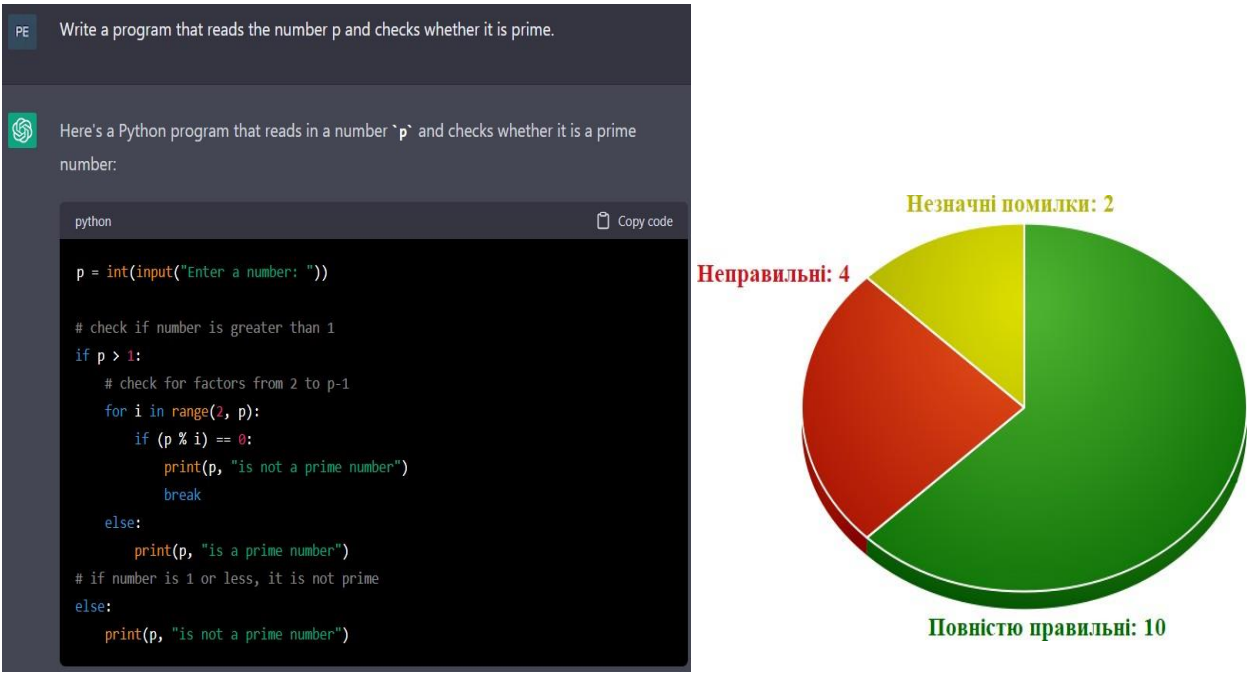


Рисунок 1. а) Взаємодія з ChatGPT б) Статистика згенерованих програм

Далі розглянули задачі з ACM ICPC World Finals 2018 та 2019 років, які не були розв’язані жодною командою на цих змаганнях [4-5]. Вибрані 3 задачі є одними з найважчих алгоритмічних задач. Жодну з цих задач він не розв’язав. Можна сказати, що ChatGPT ще не здатний розв’язувати задачі в яких необхідно об’єднати декілька ідей разом або використати різні алгоритми в одній задачі.

Для огляду можливостей ChatGPT додатково використали сайт з алгоритмічними задачами [6], на якому є близько 13000 різноманітних задач та понад 300000 зареєстрованих користувачів. Цей сайт існує з 2012-го року і є достатньо популярним, тому кількість людей, що розв’язали задачу відповідає її складності. Будемо користуватись саме такою шкалою. Спочатку вибрали декілька найпростіших та найскладніших задач, але пропускали задачі які не можна було коректно задати цьому штучному інтелекту. Результати отримали подібні до задач які були розглянуті раніше – ChatGPT не мав ніяких проблем з легкими задачами, а в складних він не мав суттєвого прогресу або взагалі не мав ідей як розв’язувати відповідну задачу. Після цього вибрали задачі середньої складності. Серед цих задач були такі, які він розв’язував одразу і без жодних проблем. У деяких задачах він допускав невеликі неточності, які було легко виправити. Також були й задачі, які він не зміг розв’язати. Отже, якщо використовувати ChatGPT для розв’язання задач середньої складності, то обов’язково потрібно перевіряти результати.

На основі досліджень дійшли висновку, що використовувати ChatGPT є доцільно для простих алгоритмічних задач, де не потрібно використовувати декілька алгоритмів. Зі складними задачами цей штучний інтелект допускає помилки, оскільки потрібно зробити багато кроків і помилка в одному впливає на весь розв’язок. Тому для алгоритмічних задач, середньої складності або складніше,

потрібно перевіряти його відповіді, бо він може помилятись.

Література:

1. ChatGPT – Режим доступу: <https://chat.openai.com/> – Дата звернення: 09.03.2023
2. Five Ideas for How Pre Ideas for How Professors Can Deal with GPT ... For Now / Travis R. Pickell and Brian R. Doak – Режим доступу: <https://digitalcommons.georgefox.edu/cgi/viewcontent.cgi?article=1432&context=ccs> – Дата звернення: 07.03.2023
3. Вступ до Алгоритмів, переклад з англійської третього видання. / Т. Г. Кормен, Ч. Е. Лейзерсон, Р. Л. Рівест, К. Стайн; 2019. – 5 - 11 с.
4. Результати ACM ICPC World Finals 2018 – Режим доступу: <https://icpc.global/community/results-2018> – Дата звернення: 07.03.2023
5. Результати ACM ICPC World Finals 2019 – Режим доступу: <https://icpc.global/community/results-2019> – Дата звернення: 07.03.2023
6. Sphere Online Judge – Режим доступу: <https://spoj.com/> – Дата звернення: 09.03.2023

ОПТИМІЗАЦІЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ЧЕРЕЗ РЕАЛІЗАЦІЮ ІДЕЇ РАНЬОГО ВИХОДУ

Кирик Орест-Теодор, Баранова Ольга

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

orest-teodor.kyirik@lnu.edu.ua

Вступ. Більшість згорткових нейронних мереж (Convolutional Neural Networks, CNN) побудовані на основі принципу послідовного опрацювання усіх внутрішніх шарів нейронної мережі як при тренуванні, так і при тестуванні моделі. Однак у деяких випадках використання всіх шарів моделі є надлишковим і може негативно вплинути на результати її роботи. Це зокрема стосується задач класифікації зображень у сфері комп'ютерного зору, коли серед зображень зустрічаються такі, що не входять до досліджуваних класів, тобто є надлишковими даними. Тому розробка ефективних моделей, позбавлених зазначених недоліків, є актуальною проблемою. У цьому відношенні значної популярності набуває ідея впровадження у модель нейронної мережі так званого блоку “раннього виходу” (Early-Exit, EE) [1], [2], [3], який відповідатиме за відсіювання надлишкових даних. Ефективна реалізація таких моделей на практиці дасть змогу в наперед визначених місцях обчислювального графа динамічно обирати шлях під час отримання результату.

Метою даної роботи є модернізація CNN з VGG-19 архітектурою [4] шляхом впровадження в неї EE-блоку та дослідження характеристик побудованої нейронної мережі на задачах класифікації зображень.

Розроблена модель. Для проведення досліджень за основу було використано архітектуру згорткової нейронної мережі VGG-19 [4]. У модель, у якій реалізовано таку архітектуру, після першого блоку в згортковій основі було додано користувацький EE-блок (див. рис. 1). Його головна мета полягає у відсіюванні надлишкових зображень на ранніх етапах обчислень, щоб зекономити ресурси моделі при класифікації необхідних зображень.

ЕЕ-блок створений на основі шарів максимізаційного агрегування (MaxPooling) та повністю зв’язних шарів (Fully-Connected, Dense), структура яких аналогічна до шарів у VGG-19 архітектурі. Також цей блок складається з шару, який вирівнює форму результату попереднього шару (Flatten). Це зроблено з метою застосування на останньому етапі повністю зв’язного шару, який на виході буде давати результат розподілу ймовірностей бінарної класифікації за допомогою функції активації *softmax*.

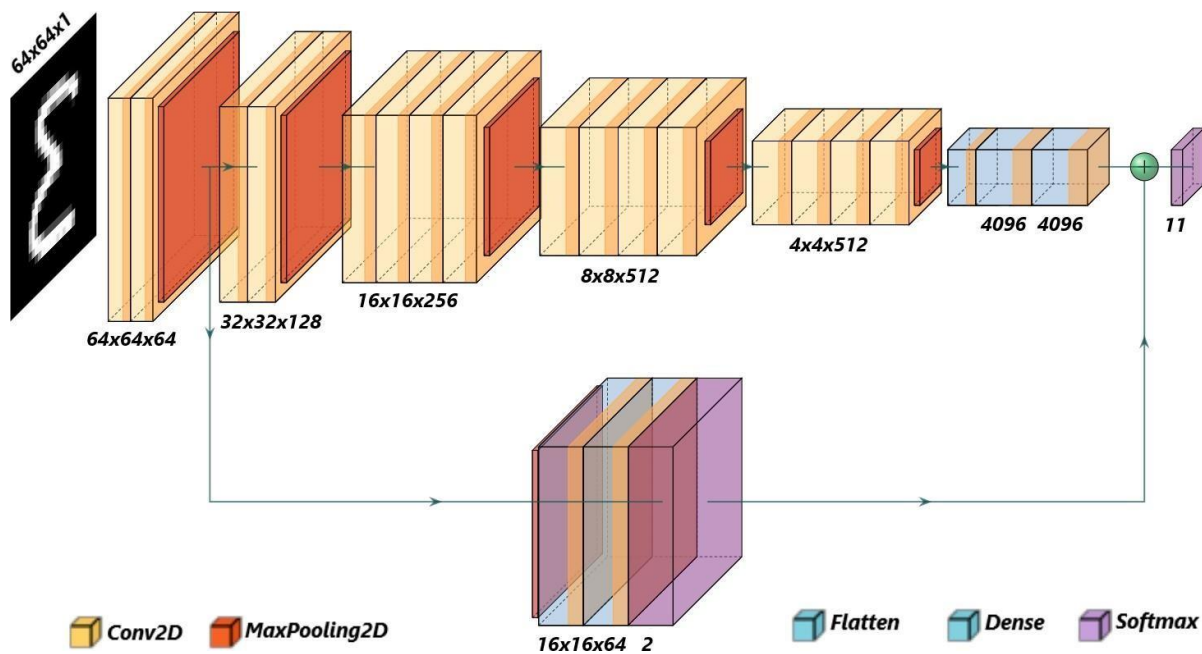


Рис. 1: Модернізована VGG-19 архітектура з реалізованим ЕЕ-блоком.

Функція втрати при виконанні ЕЕ-блоку під час тренування основана на обчисленні бінарної взаємної ентропії (Binary Crossentropy). Ця функція втрат визначає похибку при класифікації зображень між двома класами залежно від їхнього справжнього набору даних. Класифікація відбувається на ті, що належать до головного датасету і потребують додаткового опрацювання, або на ті, що належать до надлишкового набору даних і мають бути відсіяні ЕЕ-блоком. Формулу бінарної взаємної ентропії можна подати у вигляді:

$$L_1 = -\log\left(\frac{e^{k_p}}{e^{k_1} + e^{k_2}}\right), \tag{1}$$

де k_p – це елемент логіт-вектору, який відповідає індексу розміщення одиниці у One-Hot Encoded векторі представлення правильної категорії для відповідного зображення. У цьому випадку логіт-вектор – це вектор передбачених значень класифікації вхідних даних, елементи якого не є нормалізовані.

Формула для обчислення функції втрати при головній класифікації моде- лі

відносно $(C - 1)$ категорії має вигляд:

$$L_2 = -\log \left(\frac{e^{k_p}}{\sum_c^{C-1} e^{k_c}} \right),$$

При тренуванні кінцеве значення функції втрати відносно C класів, з яких $(C - 1)$ категорій є головними і 1 надлишкова, обчислюють за формулою:

$$L = L_1 + \delta L_2, \quad (3)$$

де L_1 — бінарна взаємна ентропія відносно двох класів залежно від початкового датасету, L_2 — взаємна ентропія категорій для головної класифікації моделі відносно $(C - 1)$ класу, δ — параметр, який набуває значення 0, якщо розглядається множина зображень з надлишкового класу, і 1, якщо зображення є з головного набору даних і потребують використання всіх шарів моделі.

Набори даних. При проведенні досліджень було розглянуто 3 об'єднання 6 наборів даних: MNIST, Omniglot, CIFAR-10, CIFAR-100, SVHN, GTSRB, які містять зображення $C = 11$ класів. У ролі головних були застосовані MNIST, CIFAR-10, SVHN, оскільки вони містять зображення, поділені між 10 різними класами. Набори даних Omniglot, CIFAR-100, GTSRB були використані як надлишкові, тобто зображення в кожному з них були зведені до одного надлишкового класу, який необхідно відсіяти на етапі опрацювання ЕЕ-блоку.

Опис програмної реалізації. Було розроблено клас моделі на основі архітектури VGG-19, клас загального блоку в моделі, яка реалізує VGG-19 архітектурі та клас ЕЕ-блоку, який буде додаватись у згадану модель. У кожному класі також визначено метод *call()*, який описує послідовність виконання шарів цього класу під час прямого проходження. Також у класі моделі було перевизначено методи кроку тренування та тестування, які описують поведінку моделі при опрацюванні нею відповідних даних. Для реалізації програми було використано мову Python, зокрема бібліотеку Tensorflow 2 для побудови нейронних мереж і бібліотеку NumPy для роботи з векторними даними. Програму написано на платформі Kaggle з використанням GPU NVIDIA T4(x2).

Результати. Результати експериментів апробації моделі, реалізованої на основі VGG-19 архітектури з впровадженням ЕЕ-блоком, продемонстровано на рис. 2 і в таблицях 1-4. Як видно з графіків, модель з ЕЕ-блоком демонструє подібні результати до моделі без його використання відносно

значення функції втрат і точності, якщо розглядати об’єднаний датасет MNIST і Omniglot. Однак, якщо проаналізувати таблиці результатів моделей над наборами даних SVHN і GTSRB, можна стверджувати, що у деяких випадках модель з ЕЕ-блоком класифікує вхідні дані з вищою точністю. Крім цього, подані таблиці показують, що час передбачення даних у моделі з ЕЕ-блоком зазвичай менший, ніж у моделі без його використання, хоча це залежить від кількості надлишкових даних.

Табл. 1: Результати моделі без ЕЕ-блоку над наборами даних MNIST, Omniglot

Надлишок, %	Втрата	Точність, %	Час передбачення, сек	Inference time, сек
57	0.023	99.3	21.463	0.000926

Табл. 2: Результати моделі з ЕЕ-блоком над наборами даних MNIST, Omniglot

Надлишок, %	Втрата	Точність, %	Точність ЕЕ, %	Час, сек	Inf. time, сек
57	0.034	99.4	99.7	17.503	0.000755

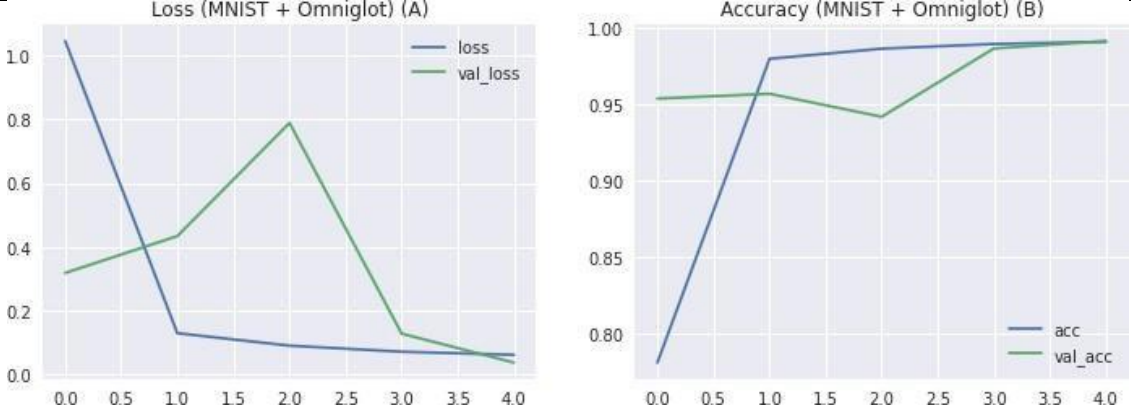


Рис. 2: Графіки функції втрати (А) та точності (В) моделі з ЕЕ-блоком на основі MNIST та Omniglot наборів даних

Табл. 3: Результати моделі без ЕЕ-блоку над наборами даних SVHN, GTSRB

Надлишок, %	Втрата	Точність, %	Час передбачення, сек	Inference time, сек
28.57	1.281	78.91	21.724	0.001034

Табл. 4: Результати моделі з ЕЕ-блоком над наборами даних SVHN, GTSRB

Надлишок, %	Втрата	Точність, %	Точність ЕЕ, %	Час, сек	Inf. time, сек
28.57	0.2836	91.29	99.9	18.66	0.000889

Висновки. Як свідчать результати апробації розробленої CNN з впровадженими блоками “раннього виходу”, описаний підхід дає змогу ефективно відсіювати надлишкові зображення на ранніх етапах опрацювання моделі. Зокрема досягнуто зменшення на 20% часу роботи моделі з використанням об’єднаного набору даних MNIST і Omniglot та на 15% стосовно датасетів SVHN і GTSRB.

Список джерел і літератури

- [1] S. Scardapane, M. Scarpiniti, E. Baccarelli, A. Uncini *Why should we add early exits to neural networks?* CoRR – 2020. arxiv.org/abs/2004.12814.
- [2] Y. Kaya, S. Hong, T. Dumitras *Shallow-Deep Networks: Understanding and Mitigating Network Overthinking* – 2018. doi: 10.48550/ARXIV.1810.07052.
- [3] S. Teerapittayanon, B. McDanel, H. T. Kung *BranchyNet: Fast inference via early exiting from deep neural networks* ICPR – 2016.
- [4] K. Simonyan, A. Zisserman *Very Deep Convolutional Networks for Large-Scale ImageRecognition* ICLR – 2015.

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ РІЗНИХ СКІНЧЕННИХ ЕЛЕМЕНТІВ ПРИ МОДЕЛЮВАННІ СЕРЕДОВИЩА МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ

Вікторія Чумакевич

Львівський національний університет імені Івана Франка,

Факультет прикладної математики та інформатики

victoriachu21@gmail.com

Метод скінченних елементів (МСЕ) є популярним методом чисельного розв'язування диференціальних рівнянь, що виникають в інженерному та математичному моделюванні. Типові проблемні області інтересів включають традиційні галузі структурного аналізу, теплопередачі, деформування, тощо. Наприклад в статті [2] даний метод використовується для визначення деформаційних властивостей фундаменту газоперекачувальних агрегатів компресорних станцій, а в [3] для визначення руйнівного навантаження на перфоровану оболонку при короткочасних силових впливах. Щоб розв'язати проблему, МСЕ розбиває велику систему на менші, більш прості частини, які називаються скінченними елементами. Вони можуть мати різні форми, а відповідно й властивості (рис. 1). Таким чином, даним методом можна апроксимувати тіла, що складаються з декількох матеріалів та мають складну структуру.

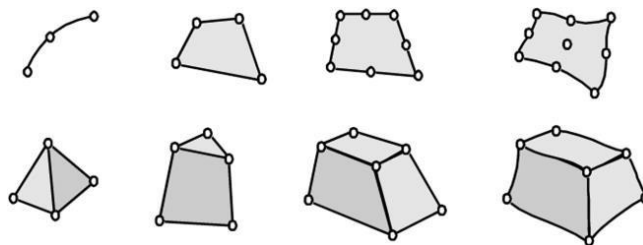


Рис. 1. Форми скінченних елементів

Постає запитання, а які елементи кращі при моделюванні? Прямі лінії, трикутники, прямокутники чи піраміди? Даний матеріал демонструє порівняння декількох видів трикутних скінчених елементів для відповіді на це запитання.

Використання різних двовимірних елементів

Як відомо скінченими елементами є прості області, такі як лінії, трикутники, прямокутники, піраміди, призми. В своїх дослідженнях я зосередилася на трикутних скінчених елементах. Мною були розглянуті декілька їх видів, а саме лінійні та квадратичні.

Кількість вагових коефіцієнтів Гауса впливають на потенційну точність результатів моделювання. Також варто зазначити, що розбиваючи область на

скінченні елементи ми можемо оперувати різними триангуляційними сітками, в залежності від заданих параметрів розбиття. При збільшенні кількості елементів розбиття, збільшується кількість рівнянь в системі та ускладнюються взаємозв'язки. Тому для оптимізації вирішення нашої задачі ми провели дослідження впливу триангуляційної сітки на кінцевий результат.

Крім цього вигляд триангуляційної сітки має вплив на складність обрахунків, а отже і на час їх виконання, що демонструє табл. 1.

Табл. 1. Дані про час виконання програми для різних сіток

S (м)	N ткрик. елементів	N лін. вузлів	N квадратичн их вузлів	Час виконання для лінійних вузлів(хв)	Час виконання для квадратичних вузлів (хв)
32	76	52	179	5	9
16	137	86	308	5	9
8	289	170	628	9	17
4	584	328	1329	13	25
2	1178	639	2455	17	33
1	2352	1248	4847	23	45

Список літератури

1. Majowski M. Conceptual design of long span structures a knowledge based synthetical approach // Proceedings of the IASS/University of Bologna, Italy. Symposium, October 7—1, 1996. Stuttgart/Germany, 1996. Vol. 1.

2. Дутчин М., Грицюк Т., Ничвид М. Дослідження точності визначення осідань фундаментів інженерних споруд на стадії ущільнення ґрунтів. Збірник наукових праць Західного геодезичного товариства УТГК. 2014. Випуск I (27). С. 45 – 56.

3. Degtyarev, K., Gnitko, V., & Tonkonozhenko, A.. Метод скінченних елементів при визначенні руйнівного навантаження на перфоровану оболонку при короткочасних силових впливах. Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління», 2021, №48, С. 22-35. DOI: <https://doi.org/10.26565/2304-6201-2020-48-02>

BREAST CANCER DETECTION USING MACHINE LEARNING

Maksym Batiuk, Vitalii Mil

Lviv Polytechnic National University,

Computer Science, Department of Artificial Intelligence Systems

batiukmaks@gmail.com, vitalii.mil04@gmail.com

Abstract

Breast cancer is a leading cause of cancer-related deaths among women worldwide. Early detection and accurate diagnosis of breast cancer can significantly improve treatment outcomes and survival rates. Machine learning techniques have shown great promise in improving breast cancer detection accuracy by analyzing medical images and other patient data.

In this research paper, we investigate the use of machine learning algorithms for breast cancer detection, focusing on both anomaly detection and classification approaches. We review the latest research on machine learning-based breast cancer detection, including the use of deep learning models for medical image analysis and the integration of genomic data and patient medical records.

Our aim in this research paper is to explain the use of machine learning algorithms for breast cancer detection in simple and easy-to-understand terms. We want to show how these powerful techniques can be used to improve the accuracy of breast cancer diagnosis by analyzing medical images and patient data. Our hope is that by providing an accessible overview of the latest research in this field, we can help raise awareness about the potential of machine learning for improving breast cancer detection and ultimately, saving lives.

Methodologies

In the context of breast cancer detection using machine learning techniques, there are two main methodologies: anomaly detection and classification.

Anomaly detection can be used to identify patterns or features within medical images that are indicative of cancerous tissue or to identify unusual patterns in patient medical records or laboratory test results that may indicate a higher risk of breast cancer. Anomaly detection techniques can also help identify cases where traditional diagnosis may be difficult or misleading, such as early-stage or subclinical breast cancer.

Classification, on the other hand, involves categorizing data into distinct classes or

groups based on known labels or features. In the context of breast cancer detection, classification models can be trained to classify medical images as either benign or malignant based on specific features or characteristics. The performance of these models can be further improved by incorporating additional patient data such as demographics, medical history, or genomic data.

Anomaly Detection

Anomaly detection in machine learning is a technique used to identify rare, unusual or abnormal instances within a dataset. Anomalies, also known as outliers, can be defined as data points that significantly deviate from the expected behavior of the majority of data points. Anomaly detection is often used in applications where the goal is to identify unusual behavior that may be indicative of a problem or a threat, such as fraud detection, intrusion detection, and equipment failure prediction.

In anomaly detection, the machine learning algorithm is trained on a set of normal or expected data and then applied to new data to identify instances that do not fit the expected pattern.

The algorithm may use various techniques, such as statistical models, clustering algorithms, or deep learning models, to identify anomalies in the data. Once anomalies are identified, they can be further analyzed to determine whether they represent a problem that requires action.

In the context of breast cancer, machine learning algorithms can be trained on a large set of medical images, including mammograms, breast ultrasound images, or MRI scans, that are known to be normal or free of cancer. The algorithm can then be applied to new images to identify patterns or anomalies that do not fit the expected pattern, which may be indicative of breast cancer.

Machine learning algorithms also can be trained on a large set of patient records, including medical histories, family histories, and laboratory test results, that are known to be normal or free of cancer. The algorithm can then be applied to new patient records to identify patterns or anomalies that do not fit the expected pattern, which may be indicative of a higher risk of breast cancer.

Classification

Classification is a type of supervised machine learning technique in which an algorithm is trained to predict the class or category to which a new input belongs based

on its features or attributes. The goal of classification is to learn a mapping function that can accurately predict the class label of a new instance based on its features.

Some of the most popular techniques for classification in machine learning for breast cancer detection include Support Vector Machines (SVMs), Neural Networks (especially CNNs), K-Nearest Neighbors (KNN), Naive Bayes classifier.

Machine learning classification is used in breast cancer detection to classify whether a given breast tissue or tumor is malignant (cancerous) or benign (non-cancerous). This can be done using various clinical and imaging features of the breast tissue, such as size, shape, texture, density, and enhancement pattern.

Results

In this study, we analyzed several machine learning algorithms for breast cancer detection based on Fares Elmenshawii’s classification of benign/malignant tumors (Table 1).

Our findings suggest that the Neural Network algorithm performed better than other algorithms in terms of recall and balanced accuracy. On the other hand, Naive Bayes and K-Nearest Neighbors (KNN) algorithms performed similarly and achieved the highest precision scores.

These results suggest that the Neural Network algorithm may be more suitable for accurately detecting breast cancer, while Naive Bayes and KNN algorithms may be useful for reducing false positives.

Table 1. Model evaluations

	Accuracy	F1 score	Precision	Recall	Balanced accuracy
K Nearest Neighbors	0.973684	0.963855	1.000000	0.930233	0.965116
Naive Bayes	0.973684	0.963855	1.000000	0.930233	0.965116
K Nearest Neighbors (all features)	0.973684	0.963855	1.000000	0.930233	0.965116
Naive Bayes (all features)	0.973684	0.963855	1.000000	0.930233	0.965116
Logistic Regression	0.964912	0.952381	0.975610	0.930233	0.958074
Random Forest	0.964912	0.952381	0.975610	0.930233	0.958074
XGBoost	0.964912	0.952381	0.975610	0.930233	0.958074
Random Forest (all	0.964912	0.952381	0.975610	0.930233	0.958074

features)					
XGBoost (all features)	0.956140	0.941176	0.952381	0.930233	0.951032
Logistic Regression (all features)	0.956140	0.939759	0.975000	0.906977	0.946446
Neural Network	0.938596	0.924731	0.860000	1.000000	0.950704
Decision Tree (all features)	0.938596	0.917647	0.928571	0.906977	0.932362
Decision Tree	0.929825	0.909091	0.888889	0.930233	0.929905

Conclusion

This study explores the use of machine learning algorithms for breast cancer detection, focusing on anomaly detection and classification methodologies. Anomaly detection identifies unusual patterns in medical images or patient records, while classification categorizes data into distinct groups based on known features. Our findings suggest that neural networks may be more suitable for accurate detection, while Naive Bayes and K-Nearest Neighbors algorithms may be useful for reducing false positives. The paper aims to raise awareness about the potential of machine learning in improving breast cancer detection and ultimately saving lives.

References

1. Breast cancer // World Health Organization – 2021 [Electronic resource] – Available from: www.who.int/news-room/fact-sheets/detail/breast-cancer
2. Taddonio P. How an AI Scientist Turned Her Breast Cancer Diagnosis Into a Tool to Save Lives // FRONTLINE – 2019 [Electronic resource] – Available from: www.pbs.org/wgbh/frontline/article/how-an-ai-scientist-turned-her-breast-cancer-diagnosis-into-a-tool-to-save-lives/
3. Artificial intelligence program aims to help doctors more accurately diagnose breast cancer // CBS News – 2020 [Electronic resource] – Available from: www.cbsnews.com/news/breast-cancer-doctors-hope-mammography-tests-will-be-improved-with-new-artificial-intelligence-program/
4. Elmenhawii F. Breast Cancer - Various Models / Elmenhawii F. // Kaggle – 2023 [Electronic resource] – Available from: www.kaggle.com/code/fareselmenhawii/breast-cancer-various-models/notebook

РОЗРОБКА АРХІТЕКТУРИ RDE (RESIDUAL DOMAIN EXPERTS) ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ У ВИПАДКУ МАЛОЇ КІЛЬКОСТІ ВХІДНИХ ДАНИХ

Маркіян Мандзак

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

markiiian.mandzak@lnu.edu.ua

Вступ

Велика кількість архітектур для задач класифікації зображень побудована таким чином, що інформація про клас розподілена між всіма параметрами мережі. Це призводить до декількох негативних наслідків:

- Необхідність дотреновувати основну частину мережі, та повністю замінити класифікаційний шар при кожній зміні множини класів.
- Неможливість виділення та конфігурації множини параметрів відповідальних за певний клас.
- Складність обчислень в ітераційному алгоритмі через взаємну залежність параметрів моделі.

Інтуїтивно постає питання розробки архітектури мережі котра розмежовує параметри приналежні до окремих класів, відповідно позбавленої вищезазначених недоліків.

Опис архітектури

Розроблена мережа підпадає під категорію згорткових мереж (CNN) і складається з трьох частин:

- Екстрактор характеристик — одна з наявних мереж, тренувана на великому наборі даних (хоч це і не є обов'язковим критерієм). Єдина вимога — це бієктивність відповідної функції, у якої область визначень складається з зображень заданого розміру, а область значень з тензорів сталої форми.
- Блок експертів — розгалужена структура, де кожна з гілок (експертів) має власну незалежну архітектуру, та множину параметрів які відповідальні за розпізнавання класу з відповідним порядковим номером. Результатом прямого ходу кожного з експертів є тензор однієї розмірності. Результати кожної з гілок об'єднуються по першій або другій осі.
- Класифікаційний шар — згортковий блок з активацією softmax, у якому розміри та крок вікна підібрані з огляду на незалежність результатів експертів.

Результатом виконання прямого ходу такої мережі на деяких вхідних даних буде вектор ймовірностей приналежності зображення до класів з відповідними порядковими номерами. Зображення відносять до найімовірнішого класу. Графічне подання архітектури зображено на рис. 1.

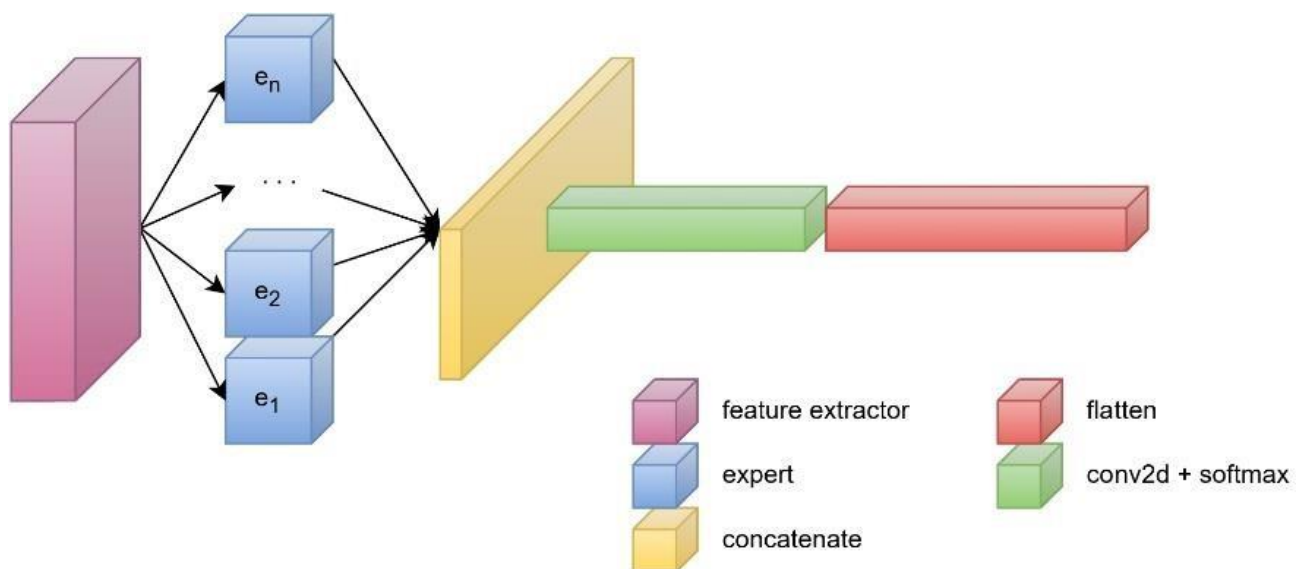


Рис. 1. Графічне подання архітектури RDE

Набори даних

Для проведення експериментів був обраний набір даних CUB-200-2011, з огляду на його високу внутрішню-, та низьку між-класову варіативність.

Цей набір складається з 11788 зображень птахів приналежних до 200 категорій. Розподіл на тренувальний та тестувальний (валідаційний) набори був використаний згідно рекомендацій авторів набору [1] - 5994/5794 зображень \approx 50/50 %.

Використане обладнання та програмне забезпечення

Всі розрахунки проводились у сервісі хмарних обчислень Kaggle Notebook з використанням графічного прискорювача NVIDIA Tesla P100.

Розробка мережі виконувалась з використанням мови програмування Python (версія 3.7.12), та бібліотек:

- Tensorflow (версія 2.6.4) - основний інструмент для побудови мережі.
- Matplotlib (версія 3.5.3) та seaborn (версія 0.11.2) - інструменти для візуалізації даних.
- Pandas (версія 1.3.5) - окремі методи для роботи з файлами у форматі csv.

Конфігурація

Зображення з набору були звужені до розміру 224x224x3, та нормалізовані. У ролі екстрактора характеристик були обрані архітектури: VGG16 [2], ResNet50 [3], InceptionV3 [4], треновані на наборі даних ImageNet [5].

В усіх експериментах архітектура експерта мала форму згорткового блоку

(Convolutionan Block) [6], див. рис. 2.

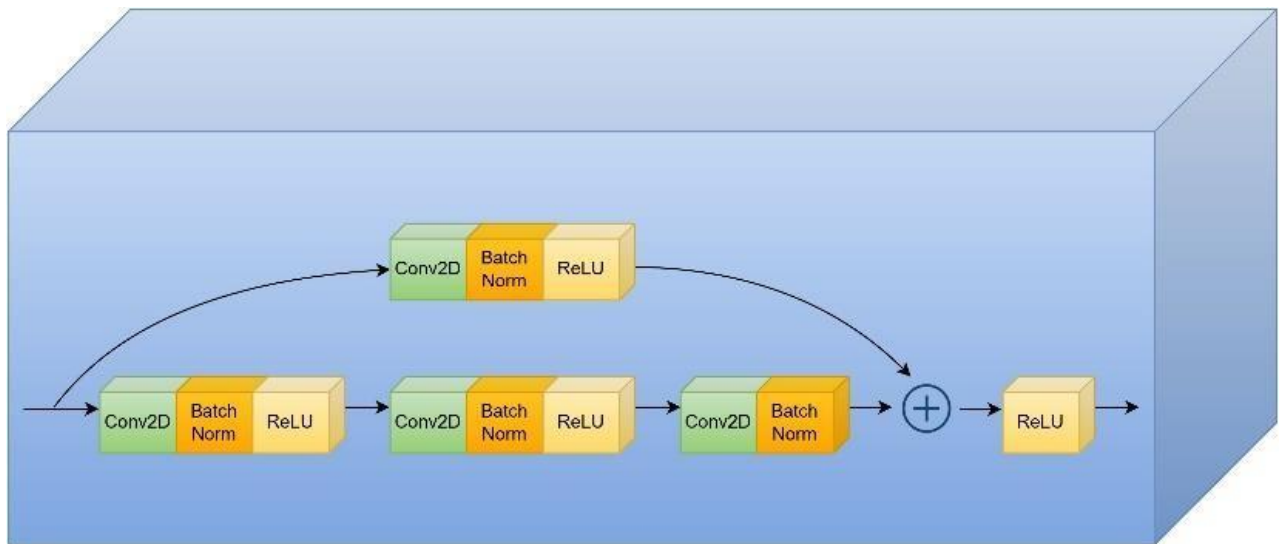


Рис. 2. Графічне подання архітектури згорткового блоку

Результати експериментів

Табл. 1. Результати експериментів

Model	Train Set Accuracy (%)	Validation Set Accuracy (%)
ResNet50	22.48	11.35
ResNet50 + ConvRDE	30.80	13.04
VGG16	98.43	40.50
VGG16 + ConvRDE	99.99	41.40
InceptionV3	98.88	65.02
InceptionV3+ ConvRDE	99.77	67.00

Висновки

У цій роботі було розроблено та протестовано архітектуру RDE яка розмежовує параметри відповідальні за розпізнавання окремих класів. Згідно з чисельними результатами отриманими під час експериментів, мережа добре показує себе на складному наборі даних, у кожному з випадків перевершуючи за метрикою точності архітектуру що лежить в основі.

Розроблена архітектура може мати широке прикладне застосування у сценаріях де відбувається часта корекція множини класів, а час на перетренування обмежений або вимагає значних обчислювальних або фінансових вкладень.

Список літератури

[1] Wah, The Caltech-UCSD Birds-200-2011 Dataset / C. Wah , S. Branson, P. Welinder, P. Perona, S. Belongie // California Institute of Technology.-2011.

[2] Simonyan, Very Deep Convolutional Networks for Large-Scale Image Recognition /

K. Simonyan, A. Zisserman // University of Oxford.-2015.

[3] Kaiming, Deep Residual Learning for Image Recognition / H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian // Microsoft Research.-2015.

[4] Szegedy, Rethinking the Inception Architecture for Computer Vision / C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens // University College London.-2015.

[5] Russakovsky, ImageNet Large Scale Visual Recognition Challenge / O. Russakovsky, J. Deng, S. Hao, J. Krause, S. Satheesh, S. Ma, H. Zhiheng, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, F. F. Li. // -2015.

[6] Medium : Understanding and implementation of Residual Networks – Режим доступа: <https://medium.com/analytics-vidhya/understanding-and-implementation-of-residual-networks-resnets-b80f9a507b9c>.

СИСТЕМА АНАЛІЗУ ДАНИХ ДЛЯ КУРСУ «МОДЕЛІ СТАТИСТИЧНОГО НАВЧАННЯ» МОВОЮ PYTHON

Малець Богдан

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

bogdanmal591@gmail.com

Вступ

В даній роботі розглядатимемо ряд методів статистичного навчання, які на даний час досить широко застосовуються у різних сферах нашого життя. Яскравим прикладом може бути дуже відомий метод лінійної регресії, який використовують для передбачення майбутніх подій. Запит на методи статистичного навчання призвела до створення наукового напрямку, що базується на поширені та застосуванні знань спеціалістів з високою кваліфікацією. Потреба в таких методах сприяла в рамках методів та систем штучного інтелекту розвитку цілого набору технологій, що покликані допомогти в управлінні суспільством, у торгівлі, кредитній та фінансовій сфері. Актуальність теми роботи, присвяченої дослідженню моделей машинного навчання та новітніх інтелектуальних методів обґрунтована застосуванням таких методів в сфері господарської діяльності реальних підприємств для прогнозування майбутніх подій.

Тема дослідження: дослідження методів статистичного навчання та їх реалізація.

Мета дослідження: реалізація методів на мові python з графічним інтерфейсом з можливістю використовувати як штучно створенні дані, так і зчитувані з файлів.

Набір даних, що використовувався

Для дослідження використано функцій розробленої програми, є можливість генерувати потрібні набори даних та працювати з реальними даними. Для демонстрації функції генерування в роботі використано згенерований набір даних.

Огляд методів

1) Лінійна регресія

є одним з найпростіших методів моделювання залежності між змінною відгуку і однією або більше змінними передбачування (так званими факторами).

Лінійна регресія припускає, що залежність між змінними можна описати лінійною функцією.

У лінійній регресії ми намагаємося знайти лінійну функцію, яка найкращим чином підходить до даних. Ми шукаємо параметри цієї функції, такі як нахил і перетин, які дозволяють нам здійснювати передбачення.

Формула лінійної регресії має вигляд:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r + \varepsilon$$

де:

- y - змінна відгуку (також відома як залежна змінна), яку ми намагаємося передбачити
- x_1, x_2, \dots, x_r - фактори (також відомі як незалежні змінні), які ми використовуємо для передбачення змінної відгуку
- β_0 - перетин (також відомий як свободний член), що показує значення y , коли всі фактори дорівнюють нулю
- $\beta_1, \beta_2, \dots, \beta_r$ - коефіцієнти регресії (також відомі як нахили), що показують, наскільки змінюється y зі зміною відповідного фактора x_i , при умові, що інші фактори залишаються незмінними
- ε - помилка, що враховує випадкові відхилення між нашими передбаченнями та реальними значеннями y

Метою лінійної регресії є знаходження оптимальних значень коефіцієнтів регресії, які дозволяють найкраще підібрати дані, тобто мінімізувати суму квадратів помилок між нашими передбаченнями та реальними значеннями y . Цю оптимізацію можна здійснити за допомогою методу найменших квадратів (МНК), який полягає у знаходженні такого набору коефіцієнтів регресії, який мінімізує суму квадратів відхилень між нашими передбаченнями та реальними значеннями y . Метод можна застосувати як до простої лінійної регресії (коли є лише один фактор), так і до багатовимірної лінійної регресії (коли є більше одного фактора).

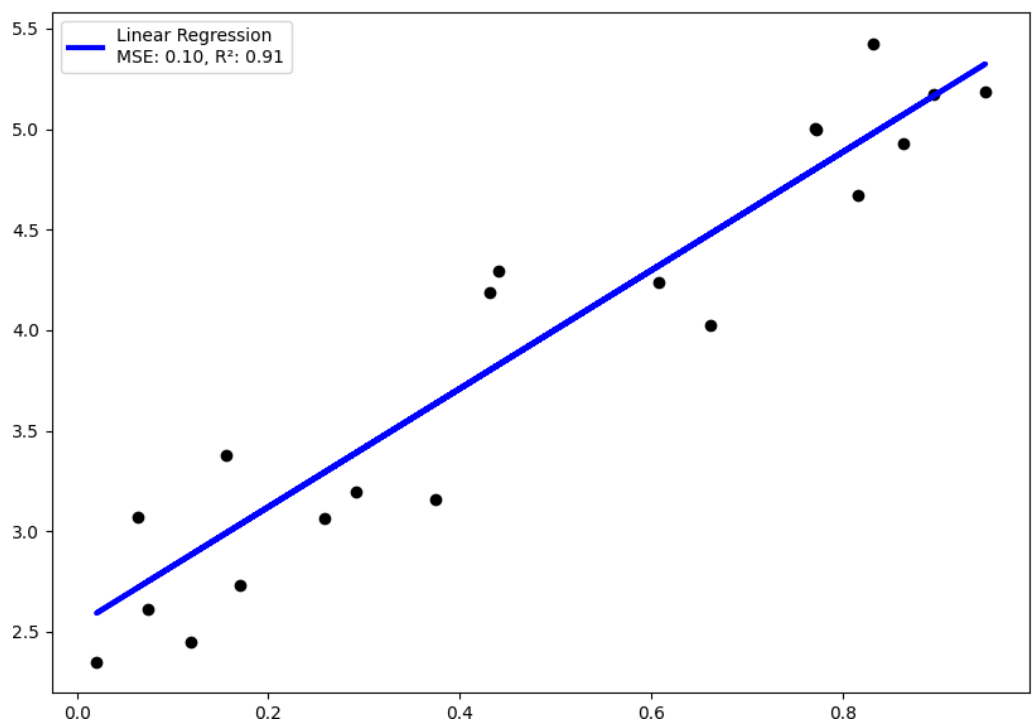


Рисунок 1– Лінійна регресія модель результат роботи програми

2) Метод Lasso

(*Least Absolute Shrinkage and Selection Operator*) є одним із методів регуляризації у лінійній регресії. Він дозволяє зменшити ризик перенавчання та покращити узагальнюючу здатність моделі шляхом зменшення ваги незначущих факторів, тобто зменшення коефіцієнтів регресії, що відповідають за ці фактори.

Формула лінійної регресії з методом Lasso має вигляд:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + ... + \beta_r x_r + \varepsilon - \lambda \sum |\beta|$$

де:

- $y, x_1, x_2, ..., x_r, \beta_0, \beta_1, \beta_2, ..., \beta_r$ та ε мають той самий зміст, що й у звичайній лінійній регресії.
- λ - параметр регуляризації, який контролює рівень зменшення коефіцієнтів регресії. Чим більше значення λ , тим більше коефіцієнти регресії будуть зменшені до нуля.
- $\sum |\beta|$ - сума абсолютних значень коефіцієнтів регресії.

Метод Lasso мінімізує функцію втрат за допомогою МНК, але з додатковою умовою, що сума абсолютних значень коефіцієнтів регресії повинна бути меншою за певне значення. Це дозволяє виключити незначущі фактори з моделі, зменшуючи їх вагу.

Даний метод має властивість занулення коефіцієнтів регресії для незначущих факторів, що дозволяє зменшити розмір моделі та поліпшити її узагальнювальну здатність.

Реалізація

Це програма з графічним інтерфейсом користувача (GUI) для порівняння різних моделей машинного навчання на основі набору даних. Нижче є опис використовуваних бібліотек та їхніх функцій:

- tkinter - це стандартна бібліотека Python для створення GUI-інтерфейсу.
- filedialog - модуль бібліотеки tkinter, що дозволяє користувачеві вибирати файл з діалогового вікна.
- pandas - бібліотека для роботи з даними, яка надає структури даних та інструменти для роботи з ними.
- numpy - бібліотека для наукових обчислень, яка надає багатошвидкісні масиви та математичні функції.
- matplotlib - бібліотека для візуалізації даних.
- sklearn - бібліотека для машинного навчання, що надає інструменти для побудови та оцінки моделей машинного навчання.
- GridSearchCV - клас з бібліотеки sklearn, який здійснює пошук параметрів гіперпараметрів моделі з оптимальними значеннями за допомогою хрестової перевірки.
- DecisionTreeRegressor - клас з бібліотеки sklearn, який будує дерево рішень для регресії.
- RandomForestRegressor - клас з бібліотеки sklearn, який будує ансамбль випадкових дерев рішень для регресії.
- GradientBoostingRegressor - клас з бібліотеки sklearn, який будує ансамбль дерев рішень з використанням градієнтного підсилення для регресії.
- SVR - клас з бібліотеки sklearn, який реалізує метод опорних векторів для регресії.

Клас MLModelComparison містить методи та атрибути для створення GUI-інтерфейсу та порівняння

В планах також включити також до функціоналу такі методи як: дерева рішень, метод опорних векторів ну і так далі.

Список використаної літератури:

1. G. James An Introduction to Statistical Learning with Applications in R /Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani //Springer-2013-p 15-368.

ОЦІНКА ЯКОСТІ СТЕГANOГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ В ЗОБРАЖЕННЯХ

Віталій Грициндишин

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

vitalii.hrytsyndyshyn@lnu.edu.ua

Вступ

Мистецтво приховування таємних повідомлень було інтригуючим аспектом спілкування протягом всієї історії людства. Стеганографія – це техніка, яка використовується для приховування факту існування секретних повідомлень, щоб їх не виявив зломисник. З появою цифрового зв'язку та простоти обміну даними стеганографія стала важливим інструментом інформаційної безпеки. Останнім часом важливість стеганографії експоненціально зросла із збільшенням кіберзлочинності та стеження.

Дане дослідження має на меті забезпечити всебічне розуміння стеганографії та зробити внесок у сферу інформаційної безпеки шляхом оцінки ефективності різних стеганографічних методів приховування інформації в растрових зображеннях. Результати цього дослідження можуть допомогти покращити безпеку цифрового зв'язку шляхом визначення найефективніших методів і розробки кращих способів захисту конфіденційної інформації.

Постановка задачі

Стеганографія передбачає приховування повідомлення чи інформації в іншому носії, наприклад, зображенні, аудіофайлі чи текстовому документі, щоб це було важко помітити випадковим спостерігачам. Основою метою стеганографії є приховування саме факту передачі, а не фактичного змісту інформації.

Рис. 1 ілюструє загальну систему стеганографії, яка дозволяє відправнику таємно передавати повідомлення отримувачу. Відправник починає з супровідного повідомлення (контейнера), яке приховує вбудовану інформацію за допомогою стеганографічного алгоритму, використовуючи стегоключ. Щоб витягти вбудоване повідомлення, одержувач розпочинає цей процес (витягування), використовуючи той самий або пов'язаний стегоключ. Повідомлення контейнера та повідомлення стего повинні мати однаковий тип, але вбудоване повідомлення може мати інший тип даних. Особа, яка намагається атакувати або аналізувати систему стеганографії, не має попередньої інформації про те, чи є в контейнері приховане

повідомлення. Якщо доведено, що в перехопленому контейнері існує приховане повідомлення, то стегосистема вважається скомпрометованою або зламаною.

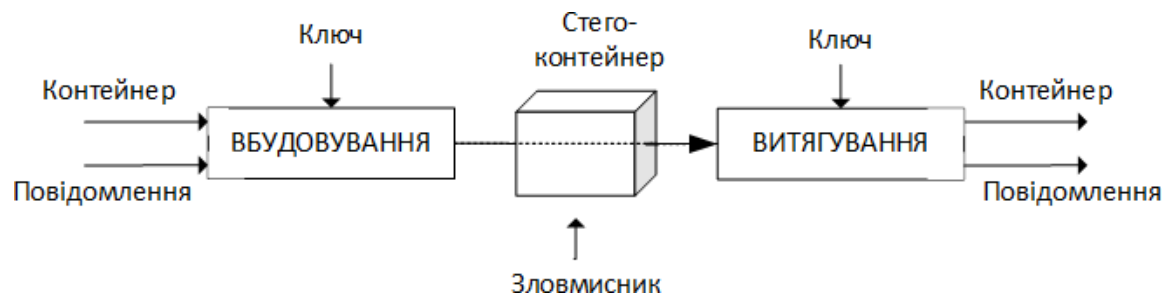


Рис. 1. Узагальнена схема системи стеганографії.

Приховування даних в растрових зображеннях

Методи створення стегоконтейнерів мають певні обмеження щодо розміру повідомлення та надійністю приховування. Зі зростанням розміру повідомлення збільшується і розмір стегоконтейнера, крім того, це може призвести до гіршої якості зображення, у результаті чого є ймовірність викликати підозру. Таким чином, існує компроміс між надійністю приховування та розміром повідомлення, як показано на графіку Рис. 2.

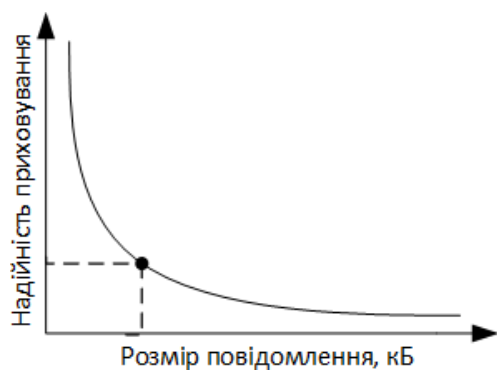


Рис. 2. Ілюстрація залежності між надійністю приховування повідомлення та його розміром.

Для приховування даних у файлах зображень використовують різні методи. Опишемо метод заміни найменш значущого біта та метод перетворення. Їхня ефективність може змінюватись в залежності від типу використаного формату контейнера.

Метод заміни найменш значущого біта. (LSB)

Реалізація цього методу відбувається за наступною схемою (Рис. 3):

- 1. Секретне повідомлення перетворюється у двійковий формат шляхом перетворення кожного символу у відповідне значення ASCII, а потім у 8-бітний двійковий код.
- 2. Обирається ключ, який вказуватиме на цифру в двійковому представленні, яку слід замінити.

3. Обирається контейнер, у якому достатньо місця для вбудовування двійкового повідомлення.
4. Контейнер розділюється на пікселі.
5. Замінюється найменш значущий біт кожного пікселя (відповідно до ключа) бітом із повідомлення, яке потрібно приховати, щоб вставити його в контейнер. Зміни не будуть помітними для органів чуття, бо даний метод має найменший вплив на колір пікселя.
6. Модифікований контейнер надсилають призначеному одержувачу, який може витягнути вбудоване повідомлення за допомогою того ж самого методу.

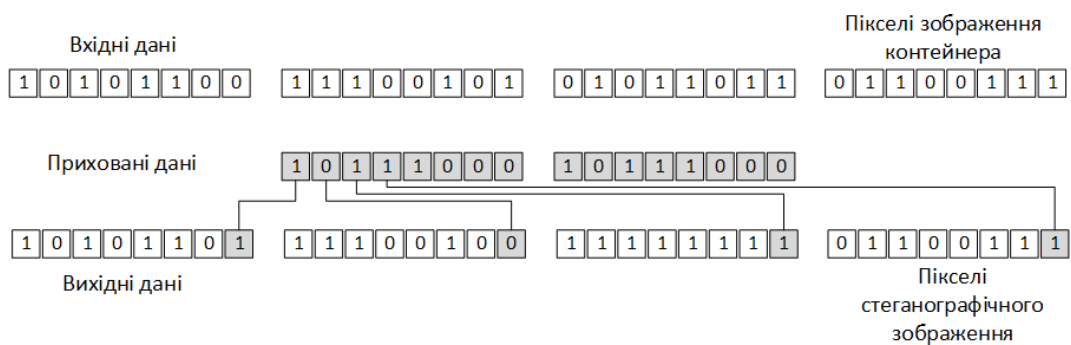


Рис. 3. Схема методу заміни найменш значущого біта.

Метод дискретно-косинусних перетворень. (DCT)

Реалізація цього методу відбувається за алгоритмом:

1. Секретне повідомлення перетворюється у двійковий формат.
2. Обирається ключ, який визначає, які коефіцієнти матриці DCT будуть змінені.
3. Обирається контейнер, у якому є достатньо місця для вбудовування повідомлення.
4. Зображення розділюється на блоки та застосовується DCT-перетворення до контейнера за допомогою формули, яка перетворює його на матрицю частотних коефіцієнтів:
$$F(u, v) = \frac{1}{4} C(u)C(v) \sum_{x=1}^n \sum_{y=1}^m f(x, y) \cos \frac{(2x+1)u\pi}{2n} \cos \frac{(2y+1)v\pi}{2m}, \quad (1)$$
де $F(u, v)$ – коефіцієнт частоти в (u, v) , $f(x, y)$ – значення пікселя в (x, y) , n, m – розмір зображення, $C(u), C(v)$ – коефіцієнти масштабування, які визначаються як: $C(u) = C(v) = 1/\sqrt{2}$, коли $u, v = 0$ та $C(u) = C(v) = 1$ в інших випадках.
5. Знаходяться два коефіцієнти в кожному блоці з найменшими абсолютними значеннями (або відповідно до ключа), які відповідають найвищим частотним компонентам.
6. Замінюється молодший біт коефіцієнтів на біт з секретного повідомлення.

Для оцінки ефективності стеганографічних методів виділимо критерії оцінки стегосистем. PSNR і MSE є загальноновживаними показниками для оцінки якості змінених або стислих зображень. MSE обчислює середню квадратичну різницю між значеннями пікселів двох зображень. Менші значення MSE вказують на більшу схожість між двома зображеннями.

$$MSE = \frac{1}{X \times Y} \sum_{x=1}^n \sum_{y=1}^m (C_{x,y} - S_{x,y})^2, \tag{2}$$

де $C_{x,y}$ – значення пікселя порожнього контейнера, $S_{x,y}$ – значення пікселя заповненого контейнера, n, m – кількість рядків та стовпців в масиві C .

PNSR враховує діапазон значень пікселів на зображенні та вимірює максимальне співвідношення значення пікселя до середньої квадратичної похибки між двома зображеннями. Вищі значення PSNR свідчать про більшу схожість між двома зображеннями.

$$PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right). \tag{3}$$

Представлення результатів

Використовуючи мову програмування Python вбудуємо текстові дані в зображення методами LSB та DCT. Модифікації не помітні органам чуття людини, тому за допомогою онлайн-ресурсу, відобразимо різницю між порожнім та заповненим контейнером .bmp формату.



Рис. 4. Різниця між порожнім та заповненим контейнером методу LSB.



Рис. 5. Різниця між порожнім та заповненим контейнером методу DCT.

Для оцінки ефективності методів обчислимо PNSR та MSE та відобразимо їх в табл. 1, при різних розмірах вбудованого повідомлення.

Табл. 1. Порівняння критеріїв оцінки методів.

Розмір повідомлення	LSB		DCT	
	PNSR	MSE	PNSR	MSE
64	88.95	0.0001	32.04	40.58
128	85.91	0.0003	32.01	40.62

Висновки

Стеганографія відіграє важливу роль у забезпеченні безпечного зв'язку та захисту конфіденційних даних. Завдяки дослідженню було оцінено ефективність двох методів, а саме заміни найменш значущого біта та дискретно-косинусних перетворень, за допомогою PNSR та MSE. Виявлено, що зі збільшення розміру секретного повідомлення погіршується надійність приховування, знижується PNSR та збільшується MSE, що, власне, і відповідає теоретичному матеріалу. Щодо вибору методу: кращі результати показав спосіб заміни найменш значущого біта, проте відомо, що метод дискретно-косинусних перетворень є більш стійким до стегаатак.

Список літератури

1. Amirulhaqi A. A Comparison of Steganography in the GIF Image using LSB and Spread Spectrum Method / A. Amirulhaqi, T. Purboyo, R. Nugrahaeni // Journal of Engineering and Applied Sciences 15.– 2020. 247-251.
2. Kharrazi M. Image Steganography: Concepts and Practice / M. Kharrazi, H. Sencar, N. Memon // WSPC/Lecture Notes Series. – April 22, 2004.– 31 pp.
3. Zaidan A. Overview: Main Fundamentals for Steganography / A. Zaidan, B. Zaidan, O. Alanazi // Journal of Computing. – 2010. – Vol. 2
4. Kumar S. Hiding the Text Messages of Variable Size using Encryption and Decryption Algorithms in Image Steganography / S. Kumar, G. Singh, T. Kumar, M. Nehra // International Journal of Computer Applications. – 2013. – Vol. 61.
5. Steganography in Digital Media: Principles, Algorithms, and Applications, 2009. – 466 pp.
6. Free online diff checker tool to compare two image files. [Електронний ресурс]. – Режим доступу: <https://www.textcompare.org/image/>

РОЗВ'ЯЗУВАННЯ ОДНОВИМІРНИХ КРАЙОВИХ ЗАДАЧ МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ

Малашніак Марія

Львівський національний університет імені Івана Франка

Факультет прикладної математики і інформатики

mmalashnyak@gmail.com

Формулювання крайової задачі. Розглянемо крайову задачу

$$-\frac{d}{dx} \left(\mu \frac{du(x)}{dx} \right) + \beta(x) \frac{du(x)}{dx} + \sigma(x)u(x) = f(x) \quad \forall x \in (0,1) \quad (1.1)$$

яка задовольняє такі крайові умови

$$\begin{aligned} u(0) &= 0, \\ -\mu \frac{du}{dx} \Big|_{x=1} &= \alpha[u(1) - \bar{u}] \end{aligned} \quad (1.2)$$

Тут $\mu = \mu(x), \beta = \beta(x), \sigma = \sigma(x)$ та $f = f(x)$ – задані функції такі, що

$$\begin{cases} \mu(x) \geq \mu_0 = \text{const} > 0, \sigma(x) \geq 0 \\ \mu, \beta, \sigma \in L^\infty(\Omega), f \in L^2(\Omega) \end{cases} \quad (1.3)$$

$\alpha \geq 0, \bar{u}$ – задані сталі.

Варіаційне формування крайової задачі. Крайова задача допускає варіаційне формування вигляду:

$$\begin{cases} \text{знайти } u \in V \text{ таку, що} \\ c(u, v) = \langle l, v \rangle \quad \forall v \in V \end{cases} \quad (2)$$

з такими структурними елементами:

$$\begin{cases} V := H^1 = \{v \in H^1 : v(a) = 0\} \\ c(u, v) := \int_a^b \left(\mu \frac{du}{dx} \frac{dv}{dx} + \beta \frac{du}{dx} v + \sigma uv \right) dx + \alpha u(b)v(b) \\ \langle l, v \rangle := \int_a^b f v dx + \bar{\alpha} w(b) \end{cases} \quad (3)$$

Оцінювач апроксимації похибок для частинами лінійної апроксимації. Ми використовуємо поелементно визначений апостеріорний оцінювач похибки MSE у вигляді розвинення:

$$e_h(x) := \sum_{i=0}^{N-1} \lambda_{i+1/2} b_{i+1/2}(x) \quad (4)$$

де локальний носій $\text{supp } b_{i+1/2} := K_{i+1/2}$ і на цьому елементі базисна функція в локальних координатах описується виразом $b_{i+1/2} := 4\theta(\xi)\omega(\xi)$.

Оцінювач апроксимації похибок для частинами квадратичної апроксимації.

Оцінювач похибок будуємо на кожному скінченному елементі використовуючи кубічні поліноми вигляду:

$$\begin{cases} e_{i+1/2}(\xi) := [\omega(\xi)\theta(\xi)\{\theta(\xi) - \omega(\xi)\}]\lambda_{i+1/2} \\ \forall \xi \in [-1; 1], \forall i = 0, \dots, N-1 \end{cases} \quad (5)$$

Оцінювач апроксимації похибок для частинами кубічної апроксимації.

Оцінювач похибок будуємо на кожному скінченному елементі використовуючи поліноми четвертого порядку вигляду:

$$\begin{cases} e_{i+1/2}(\xi) := 16\omega^2(\xi)\theta^2(\xi)\lambda_{i+1/2}^2 \\ \forall \xi \in [-1; 1], \forall i = 0, \dots, N-1 \end{cases} \quad (6)$$

Стратегії адаптування сіток МСЕ. На кожному скінченному елементі перевіряємо чи виконується нерівність

$$\eta_{i+1/2} > eps, \quad (7)$$

де $\eta_{i+1/2} = \frac{\sqrt{N}\|e_h\|_{i+1/2}}{\sqrt{\|u_h\|_V + \|e_h\|_V}} 100\%$, $\|e_h\|_{i+1/2}$ – апостеріорний оцінювач похибки для

лінійних апроксимацій, $\|u_h\|_V^2$ – енергетична норма розв'язку на скінченному елементі K, eps – заданий максимально допустимий рівень похибки.

Для порівняння адаптування сіток методу скінченних елементів було вибрано 3 стратегії.

Стратегія №1. Для всіх скінченних елементів вираховуємо $\eta_{i+1/2}$. Шукаємо коефіцієнт $k = \max(\eta_{i+1/2})/10$ і після цього кожен елемент, на якому виконується нерівність (7), ділиться на декілька нових, у співвідношенні $\eta_{i+1/2} : k$. Алгоритм буде повторюватися, поки хоча б одна з похибок елементів вища норми.

Стратегія №2. Кожен елемент, на якому виконується нерівність (7), поділяється на 2 рівні частини, які в майбутній сітці замінять його на два нових скінченних елементи. Решта елементів, для яких не виконується нерівність, залишається без змін і разом з усіма новоутвореними скінченними елементами формують уточнену сітку. Якщо жоден з елементів не задовольняє даної умови, то процес адаптування вважають завершеним.

Рівномірне згущення сітки. Поки хоча б один з елементів задовільняє нерівність (7), поділяємо усі скінченні елементи на два однакових елементи.

Приклад. Покладемо в рівняння крайової задачі $\mu(x) = 0.0025$, $\beta(x) = 0$, $\sigma(x) = 1$, $f(x) = \cos^2\pi x + 0.005\pi^2\cos 2\pi x$. В крайовій умові візьмемо $\alpha = 10^{28}$, $\bar{u} = 0$. Візьмемо початкову кількість скінченних елементів $N=10$. Обчислимо розв'язок нашої крайової задачі методом кусково-лінійних апроксимацій, задаємо максимальний рівень похибки $\varepsilon = 3\%$.

Табл. 1. Результати лінійної апроксимації

	Кількість кроків	Остаточна кількість елементів	Максимальна відносна похибка	Глобальна похибка
Стратегія №1	6	89	1,318	5,994
Стратегія №2	10	107	2,981	1,727
Рівномірне згущення сітки	7	640	1,56	0,38

Табл. 2. Результати квадратичної апроксимації

	Кількість кроків	Остаточна кількість елементів	Максимальна відносна похибка	Глобальна похибка
Стратегія №1	4	43	2,807	1,474
Стратегія №2	5	43	2,807	1,462
Рівномірне згущення сітки	4	80	2,088	0,534

Табл. 3. Результати кубічної апроксимації

	Кількість кроків	Остаточна кількість елементів	Максимальна відносна похибка	Глобальна похибка
Стратегія №1	4	54	2,901	0,121
Стратегія №2	6	67	2,913	0,119
Рівномірне згущення сітки	4	80	1,871	0,011

З результатів у таблицях 1-3 можна зробити висновок, що ефективність стратегій №1 та №2 переважає над рівномірним згущенням сітки. Стратегія №1 потребує меншої кількості кроків, проте остаточна кількість елементів в стратегіях майже однакова та значно менша за рівномірне згущення сітки.

Список літератури

1. Квасниця Г., Шинкаренко Г. Порівняння простих апостеріорних оцінок похибок методу скінченних елементів для задачі електростатики // Вісн. Львів. ун-ту. Сер. прикл. матем. та інформ. 2003. Вип. 7. С. 162–174.
2. Є.Абрамов, Г.Квасниця, Г.Шинкаренко Частинами квадратичні та кубічні апроксимації h -адаптивного МСЕ для одновимірних крайових задач.// Вісн. Львів. ун-ту. Сер. прикл. матем. та інформ. 2011. Вип. 17. С.47–61..
3. Є.Абрамов, О.Ліпіна, Г.Шинкаренко, А.Ямелинець.Кусково-лінійні апроксимації h -адаптивного методу скінченних елементів для одновимірних крайових задач.// Вісн. Львів. ун-ту. Сер. прикл. матем. та інформ. 2006. Вип. 11. С. 3–18.
4. Г.Шинкаренко, Ю.Козаревська Регуляризація чисельних розв'язків варіаційних задач міграції домішок: h -адаптивний метод скінченних елементів. Частина 1. //Вісник Львів. ун-ту. Серія прикл. мат. інформ. 2002. Вип.. 5. С 153-165.
5. Квасниця Г., Шинкаренко Г. Адаптивні апроксимації методу скінченних елементів для задач електростатики // Вісн. Львів. ун-ту. Сер. прикл. матем. та інформ. 2002. Вип. 5. С. 95–106. 4.
6. Tarek I. Zohdi. A Finite Element Primer for Beginners The Basics.// Springer Cham Heidelberg New York Dordrecht London 2015
7. Г.А.Шинкаренко Проекційно-сіткові методи розв'язування початково-крайових задач. Київ: НМК ВО, 1991. -88с.
8. Я.Г.Савула Метод скінченних елементів. Київ: НМК ВО, 1993. -100с.

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО ПРОГРАМУВАННЯ ДЛЯ ЧИСЕЛЬНОГО РОЗВ'ЯЗУВАННЯ ТРИВИМІРНОЇ ЗАДАЧІ РЕКОНСТРУКЦІЇ ГРАНИЦІ

Аркадій Марченко

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Arkadii.Marchenko@lnu.edu.ua

Вступ

Задача реконструкції границі полягає у відновленні форми внутрішньої границі за відомих граничних умов. Ця задача є актуальною у багатьох областях, включаючи медицину та геодезію. Важливо зазначити, що вона є некоректною за Адамаром в сенсі відсутності стійкості за вхідними даними. Це призводить до необхідності розробки спеціальних регуляризуючих методів. В даній роботі ми наведемо метод на основі генетичного програмування, який, на відміну від інших підходів, потребує лише багаторазового розв'язку прямої задачі для оцінки наближення. Як приклад доцільності застосування еволюційних алгоритмів для обернених задач наведемо роботу [1].

Постановка задачі

Розглянемо тривимірну область $D \in \mathbb{R}^3$, яка складається з двох поверхонь Γ_1 (внутрішня) та Γ_2 (зовнішня). Нехай задана функція $u \in C^2(D) \cap C(\bar{D})$, яка є розв'язком рівняння Лапласа

$$\Delta u = 0 \text{ в } D \quad (1.1)$$

та задовольняє крайові умови Діріхле-Неймана

$$u = 0 \text{ на } \Gamma_1 \text{ та } \frac{\partial u}{\partial \nu} = g \text{ на } \Gamma_2. \quad (1.2)$$

Нехай також для фіксованій функції $g \neq 0$ маємо умову Діріхле

$$u = f \text{ на } \Gamma_2. \quad (1.3)$$

Тоді задача полягає у відновленні вигляду поверхні Γ_1 при заданих граничних умовах (1.2)-(1.3).

Оскільки пряма задача (1.1) з умовами (1.2)-(1.3) є перевизначеною, то використаємо умову (1.3) для оцінки розв'язку u при фіксованій поверхні Γ_1 . Тобто зведемо задану задачу до наступної задачі мінімізації:

$$J = \|u - f\|_{L_2(\Gamma_2)} \rightarrow \min. \quad (1.4)$$

Розв'язок прямої задачі

Застосуємо метод фундаментальних розв'язків для розв'язування прямої задачі (1.1)-(1.2) (детальніше в [2]). Для цього припустимо, що поверхні Γ_1 та Γ_2 задані параметрично наступним чином:

$$\Gamma_2 = \{x_2(\theta, \varphi) = (x_{21}(\theta, \varphi), x_{22}(\theta, \varphi), x_{23}(\theta, \varphi)), \theta \in [0; \pi], \varphi \in [0, 2\pi]\},$$

$$\Gamma_1 = \{x_1(\theta, \varphi) = r(\theta, \varphi) \cdot (\sin\theta \cos\varphi, \sin\theta \sin\varphi, \cos\theta), \theta \in [0; \pi], \varphi \in [0, 2\pi]\},$$

де $r: \mathbb{R}^3 \rightarrow (0, \infty)$ задана 2π періодична функція.

Тоді розв'язок прямої задачі можемо подати у вигляді наступної суми

$$u_n(x) = \sum_{j=1}^n \lambda_j \Phi(x, y_j), \quad x \in D, \quad (1.5)$$

де $n \in \mathbb{N}$, $\Phi(x, y) = \frac{1}{4\pi} \cdot \frac{1}{|x - y|}$, $x \neq y$, $x, y \in \mathbb{R}^3$ фундаментальний розв'язок рівняння Лапласа (1.1), $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)^T$ невідомі коефіцієнти, y_j вибрані точки джерела.

Точки джерела вибираємо за наступним правилом:

$$y^k = \begin{cases} 2x_2(\theta_k, \varphi_k), & \text{для парних } k, \\ 0.5x_1(\theta_k, \varphi_k), & \text{для непарних } k, \end{cases} \quad (1.6)$$

де

$$\theta_j = \frac{\pi}{n} j, \quad \varphi_j = \frac{2\pi}{n-1} j, \quad j = 1, \dots, n, \quad x_{ij} = x(\theta_j, \varphi_j), \quad i = 1, 2. \quad (1.7)$$

Використовуючи метод колокації, отримуємо систему рівнянь для знаходження невідомих коефіцієнтів λ

$$\begin{cases} \sum_{j=1}^n \lambda_j \Phi(x_{1i}, y_j) = 0, \quad i = 1, \dots, n, \\ \sum_{j=1}^n \lambda_j \frac{\partial \Phi(x_{2i}, y_j)}{\partial x_{2i}} = g(x_{2i}), \quad i = 1, \dots, n, \end{cases} \quad (1.8)$$

яку розв'язуємо методом найменших квадратів: $A^T A \lambda = A^T b$, де A матриця задана системою (1.8), а вектор b відповідними правими частинами.

Генетичне програмування

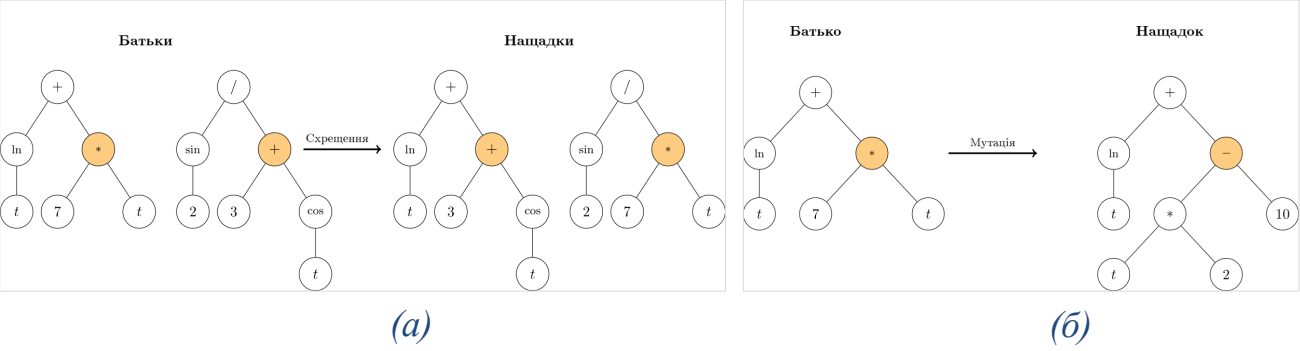
Генетичне програмування (далі ГП) є одним з еволюційних методів оптимізації який підтримує популяцію індивідів $P(t) = \{v_1^t, v_2^t, \dots, v_N^t\}$ ($N \in \mathbb{N}$ - розмір популяції, t - номер ітерації), які в свою чергу представлені у вигляді дерева і задають конкретну функцію. Дерево, яке задає конкретний індивід, часто називають "кодом".

В основу ГП покладені механізми аналогічні природньому відбору, а саме:

- **Відбір** – вибір індивідів на основі їх функції похибки $eval(v_i^t)$.
- **Схрещення** – комбінування "коду" вибраних індивідів (рис. 1.а).
- **Мутація** – зміна "коду" індивіда заздалегідь визначеними операціями (рис. 1.б).

Загальний алгоритм полягає у послідовному відборі та оновленню популяції шляхом застосування схрещення та мутації з ймовірностями p_{cross} та p_{mut} відповідно. В якості розв’язку ми вибираємо індивіда з найкращою оцінкою. Детальніше про генетичне програмування в [3,4].

Рис 1. Приклад операцій схрещення та мутації



Чисельні експерименти

Наведемо результати чисельних експериментів, при наступній конфігурації:

N	p_{cross}	p_{mut}	n	$r(\theta,\varphi)$
500	70%	30%	32	$2 + \sqrt{4.25 + 3\cos(3\theta)}$

Таблиця 1. Конфігурація програми.

Границю Γ_2 зафіксуємо як кулю радіусу 5.

Оскільки задана задача є некоректною, то наведемо результати для даних зі збуренням, які формуємо наступним чином:

$$g^\delta = g + \delta(2\eta - 1) \|g\|_{L_2},$$

де δ наперед заданий відсоток збурення, а η випадково згенероване число.

В таблиці 2 наведено результати виконання програми для різного рівня збурення δ . На рисунку 3 наведено вигляд реконструйованих поверхонь для точних даних та зі збуренням (3.а – 0%, 3.б – 2%, 3.в – 5%).

За результатами чисельних експериментів можемо стверджувати, що даний метод є застосовний до задачі реконструкції границі.

δ	ітерація	J
0%	63	4.930237815166275
2%	1883	15.41524154555154
5%	1971	25.874657870487184

Табл. 2. Результати чисельних експериментів для різного рівня збурення

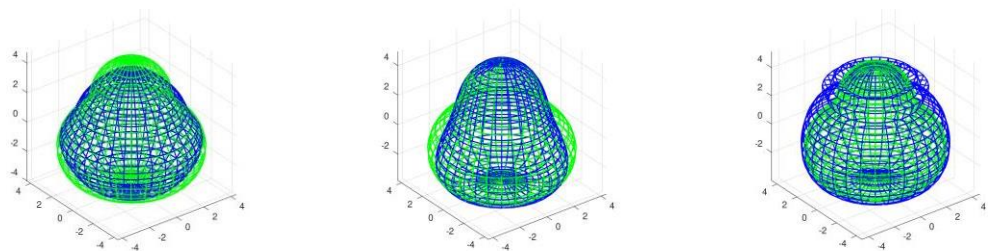


Рис 2. Зображення реконструйованих поверхонь для різного рівня збурення

Список літератури

1. Eckel H. Non-linear integral equations for the complete electrode model in inverse impedance tomography / H. Eckel, R. Kress // *Applicable Analysis* 87 (2008), 1267–1288
2. Chen C. S. On choosing the location of the sources in the MFS // C. S. Chen, A. Karageorghis, Yan Li // *Numerical Algorithms* – 2016
3. Zbigniew M. Genetic algorithm + data structures = evolutions programs / M. Zbigniew // Springer. – 1996. – С. 11-88.
4. J. Koza. Genetic Programming // MIT Press, Cambridge. – 1990

РОЗРОБКА СИСТЕМИ ДЛЯ ВИДОБУВАННЯ ІНФОРМАЦІЇ З ТЕКСТУ

Марія Пономаренко, Ольга Прядко

Львівський національний університет імені Івана Франка,

Факультет прикладної математики та інформатики,

mariia.ponomarenko@lnu.edu.ua, olha.pryadko@lnu.edu.ua

Вступ

Зростаюча кількість текстових даних, доступних в цифровому вигляді, створила потребу в автоматизованих системах попередньої обробки та аналізу тексту. Щоб вирішити цю проблему розробимо систему, що аналізує текст, шляхом реалізації основних функцій видобування інформації, а саме: виявлення ключових слів та розпізнавання іменованих сутностей. Ця система буде корисна дослідникам, викладачам та студентам, яким потрібно швидко та ефективно аналізувати та розуміти складні письмові матеріали. Також додамо функцію сканування документа, оскільки текстова інформація може зберігатися не тільки в текстових документах, але і на фотографіях.

Виділення ключових слів та розпізнавання іменованих об'єктів (англ. Name Entity Recognition, NER) є важливими методами в обробці природної мови, які дозволяють автоматично ідентифікувати важливі терміни та об'єкти в текстових даних. Для вчених і дослідників, які читають багато статей, вилучення ключових слів є цінним інструментом, який може швидко визначити найбільш значущі поняття і теми в академічних статтях. Це дозволяє їм швидше отримувати уявлення про останні дослідження і виявляти нові тенденції у своїй галузі.

Аналогічно, NER може бути корисним інструментом для студентів, які читають історичні тексти, дозволяючи їм швидко та ефективно ідентифікувати ключових осіб, місця та події, згадані в тексті. Це не лише допомагає їм краще зрозуміти історичний контекст тексту та значення згаданих осіб і подій, але й економить час і робить процес навчання більш ефективним. Автоматизуючи процес розпізнавання іменованих об'єктів, данна система робить процес аналізу та розуміння змісту тексту більш швидким.

Метою роботи є розробка уніфікованої системи для опрацювання тексту

щоміститиме в собі наступний функціонал: виявлення ключових слів в тексті, розпізнавання іменованих сутностей, шляхом дотренування мовної моделі BERT (англ. Bidirectional Encoder Representations from Transformers), функція розпізнавання тексту з фотографії.

Реалізація системи

Для реалізації десктопної системи для опрацювання тексту написаного англійською мовою обрано мову Python і використано бібліотеку PyQt для створення інтерфейсу. Система підтримує зчитування тексту як з різних текстових файлів, так і зображень. Користувач може скористатися двома запропонованими функціями, такими як: виявлення ключових слів і розпізнавання іменованих сутностей.

Для покращення процесу виявлення ключових слів для текстів що містять інформацію з різних областей були об'єднані техніки тематичного моделювання Latent Dirichlet Allocation (LDA) та техніки знаходження релевантних словосполучень в тексті. LDA - це генеративна імовірнісна модель для наборів згрупованих дискретних даних [1]. Техніка LDA обчислює умовний розподіл присвоєння тем для конкретного слова в кожному реченні, за умови, що ми не знаємо тематичну приналежність поточного слова, але знаємо такі приналежності до всіх інших слів у тексті [2]. У решті решт можливим стало отримання груп слів для кожної теми, які вважаються ключовими словами, що описують документ.

На даному етапі результат складається лише з уніграм (одне слово), тому для покращення поточного варіанту реалізації данного функціоналу, було вирішено також знаходити релевантні сповосполучення в тексті. Для цього знаходимо найбільш вживані біграми (словосполучення, що складається з двох слів) серед яких обираємо ті, що семантично найбільш схожі до всього документа за допомогою техніки вбудувань речень. Результуючий набір ключових слів представляє собою поєднання списку обраних біграм та уніграм отриманих за допомогою LDA.

За допомогою мовної моделі BERT реалізовано функцію розпізнавання іменованих сутностей NER (англ. Named Entity Recognition), що дозволяє користувачу отримати текст де є підкреслені відповідними кольорами слова, що стосуються: часу, людей, організацій, географічних локацій і геополітичних термінів. BERT є особливо ефективним для NER, оскільки він може фіксувати контекстуальні зв'язки між словами в реченні. Модель використовує контекстну інформацію, отриману від BERT, щоб прогнозувати, які слова є сутностями, а які ні. Для тренування моделі був використаний Annotated Corpus for Named Entity Recognition датасет [3].

Також реалізовано функціонал для зчитування тексту із зображення. Через те, що безпосередній текст може бути повернутий під будь-яким кутом, користувач має можливість самостійно обрати область з якої треба здійснити зчитування тексту. Для реалізації використано бібліотеку OpenCV.

Оцінка методів та демонстрація роботи системи

Для оцінки якості роботи методу вилучення ключевих слів відбулося порівняння результатів його роботи з іншими існуючими методами, такими як: TF-IDF, TextRank та KeyBert. Загалом, середнє значення частки релевантних слів для метода, що використовується у данній роботі, становиться 0.4177 і є найбільшим у порівнянні з іншими методами.

Для тренування і тестування моделі використовувався датасет Annotated Corpus for Named Entity Recognition що складається з 47959 записів. Він містить речення, що розбиті на слова і кожне слово асоціюється з певним тегом і частиною мови. Після тестування побудованої моделі ми згенерували матрицю невідповідностей, щоб оцінити точність класифікації кожного слова. Найбільшу кількість тегів становлять В-geo - усього 5737, з них правильно класифіковані були 5280 (94%).

Для демонстрації виконання функції «Виявлення ключових слів» була використана наукова стаття з архіву електронних публікацій arXiv.org [4]. Що стосується розпізнавання іменованих сутностей, дана функція була протестована на уривку тексту з історичної статті з Вікіпедії [5]. Кожна з сутностей в тексті виділяється окремим кольором для швидшого розуміння ключових деталей про вміст.

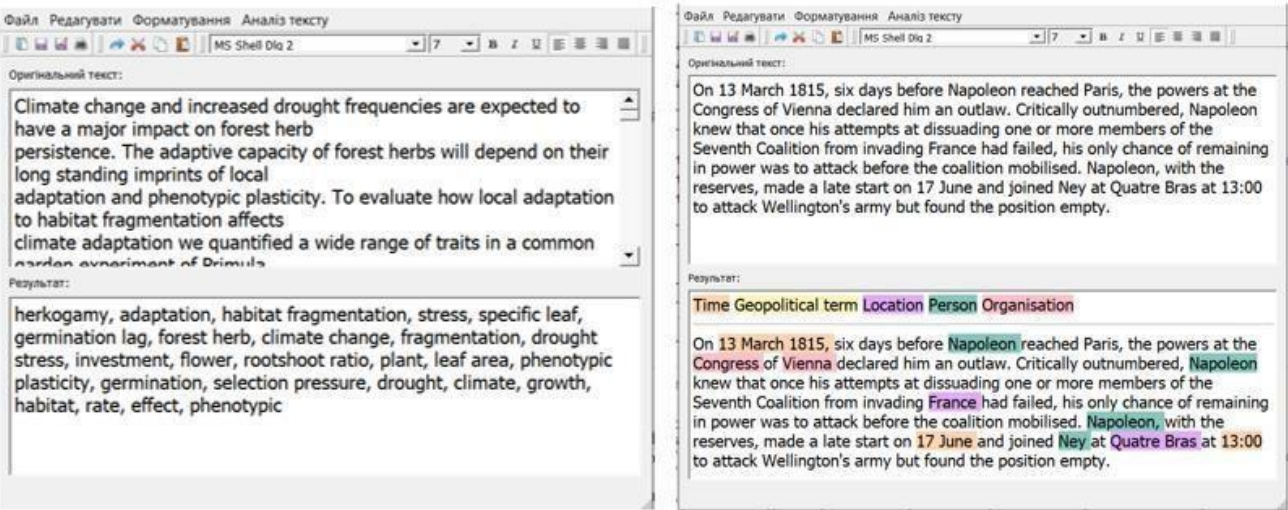


Рис. 1. Демонстрація роботи систем

Висновок

У даній роботі розроблено систему для опрацювання тексту, написаного англійською мовою за допомогою бібліотек PyQT, OpenCV з використанням машинного навчання, що реалізує наступний функціонал:

- Виявлення ключових слів: функція, що автоматично ідентифікує і вилучає найважливіші слова і фрази з текстового корпусу.
- Розпізнавання іменованих об'єктів: функція, що є підзадачею вилучення інформації, яка прагне знайти та класифікувати названі об'єкти, згадані в неструктурованому тексті, за попередньо визначеними категоріями, такими як імена осіб, організації, місця розташування.
- Сканування документа: функція, що обробляє зображення, визначає межі розташування тексту, застосовує техніку розпізнавання символів і виводить його у поле вводу.

Крім того, було здійснене порівняння метода ключових слів з аналогічними методами і проаналізована якість його роботи.

Список літератури

1. David M. Blei. Latent dirichlet allocation / David M. Blei, Andrew Y. Ng, Michael I. Jordan. // The Journal of Machine Learning Research – 2003. - Volume 3. – P. 993–1022.
2. Darling, William M. A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling / Darling, William M. // In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 642–47.
3. Annotated Corpus for Named Entity Recognition [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus>
4. Habitat fragmentation affects climate adaptation in a forest herb [Електронний ресурс] - Режим доступу до ресурсу: <https://arxiv.org/abs/2303.15712>
5. Battle of Waterloo [Електронний ресурс] - Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Battle_of_Waterloo

РОЗРОБКА РУРІ-БІБЛІОТЕКИ ДЛЯ ФРАГМЕНТОВАНОЇ ПЕРЕДАЧІ ФАЙЛІВ (З ПІДТРИМКОЮ ФРЕЙМВОРКУ DJANGO)

Хомин Віталій

Львівський національний університет імені Івана Франка

Факультет прикладної математики і інформатики

vitalii.khomyn@lnu.edu.ua

Вступ

Python Django - один із найпопулярніших backend-фреймворків. Він забезпечує швидку і масштабовану розробку backend для різноманітних веб-сайтів і веб-застосунків.

Однією з особливостей Django є вбудована підтримка роботи з файлами. Це дає змогу розробникам та користувачам легко завантажувати відносно невеликі за розміром (до 50MB) файли на сервер і зберігати їх.

Однак передача великих файлів, таких як відео- та аудіо файли є складною, оскільки вони займають багато місця і вимагають великої пропускної здатності мережі. Крім того, є ризик обриву з'єднання під час завантаження файлів, що може призвести до втрати даних і неефективної передачі.

Також, існують обмеження на розмір запиту з боку багатьох веб-серверів та CDN, наприклад з боку Nginx та Cloudflare. Це означає, що файли, які перевищують певний розмір, не можуть завантажені одним запитом взагалі.

Одним з можливих рішень цієї проблеми є передача файлів фрагментами (chunks).

І саме для реалізації фрагментованої передачі файлів запропоновано розробити бібліотеку для Python, з вбудованою підтримкою для фреймворку Django і можливістю доповнення її до інших backend-фреймворків, наприклад Flask чи FastAPI.

Основний процес і реалізація

Розглянемо основний алгоритм фрагментованої передачі файлів.

Крок 1: Створення сеансу завантаження (session initialization)

Спочатку необхідно створити сеанс завантаження для нового файлу або нової версії файлу, надіславши відповідний HTTP POST-запит. Цей сеанс міститиме інформацію про ім'я файлу, його розмір і розташування на сервері. Опціонально, можна передати алгоритм хешування для перевірки цілісності файлів при завантаженні.

```
curl -X POST \
  http://localhost:8000/example-app/file-uploads/init \
  -H 'Authorization: Bearer <jwt_token>' \
  -H 'Content-Type: application/json' \
  -d '{
    "fileName": "example.mp4",
    "size": 134217728,
    "directory": "/uploads"
  }'
{
  "upload-id": "2bed74c2-8ee8-4d80-b263-740f8e5ada4e",
  "filename": "example_1682776788.mp4",
  "expires_at": "2023-04-29 14:01:15.668023",
  "recommended_chunk_size": ,
  "max_chunk_size":
}
```

Рис. 1. Структура запиту ініціалізації сеансу

У відповідь, отримуємо ідентифікатор сесії завантаження, назву файлу (в разі, якщо файл з такою самою назвою уже на сервері, надіслати перейменовану версію), рекомендовану та максимальну кількість байтів на один фрагмент(chunk) та термії дії сесії завантаження.

Крок 2: Завантаження частин файлу (chunks upload)

Після ініціалізації, завантажуюмо частини файлу на сервер.

Для цього використовуємо HTTP PUT-запит з заголовком “Content-Range” для ідентифікації діапазону байтів файлу та власне вмістом фрагмента.

Відстежування кількості завантажених фрагментів та поділ файлів на фрагменти відбувається з боку клієнта. Поділ файлів можна здійснити, наприклад, за допомогою метода `file.slice(start, end)`.

Завантаження фрагментів може відбуватись послідовно, паралельно і може навіть призупинятись. Однак при завершенні терміну дії сеансу, передача наступних фрагментів вже неможлива без створення нового сеансу.

```
curl -X PUT http://localhost:8000/example-app/file-uploads/<str:upload-id> \
  -H 'Authorization: Bearer <jwt-token>' \
  -H 'Content-Type: application/octet-stream' \
  -H 'Content-Range: bytes <start>-<end>/<size>' \

--data-binary "<chunk-binary>" {
  "upload_id": "<upload-id>",
  "message": "Chunk successfully uploaded."
}
```

Рис. 2. Структура запиту надсилання фрагменту

Крок 3: Завершення сеансу (session commit)

Після того, як всі частини файлу будуть завантажені, сесію завантаження необхідно зафіксувати. Це можна зробити, надіславши HTTP POST запит на сервер, також при тому надіславши хеш файлу (наприклад, алгоритмом SHA256

або MD5) як контрольну суму.

```
curl -X POST http://localhost:8000/example-app/file-uploads/<str:upload-id>/commit \
-H 'Authorization: Bearer <jwt-token>' \
-H 'Content-Type: application/json' \
-d '{"checksum": "<sha256-checksum>"}'
```

Рис. 3. Структура запиту фіналізації сеансу

При обробці запиту сервер перевіряє наявність усіх фрагментів, зливає їх в один файл, перевіряє його за допомогою контрольної суми і розташовує файл в необхідній теці. Сеанс завантаження завершується і тимчасові теки і файли видаляються.

```
# find all chunks
chunk_list = os.listdir(os.path.join("temp", upload_id))

# order by content range (chunk filenames are in format <original-filename>-<start>-<end>.<extension>)
chunk_list.sort(key=lambda x: int(x.split('.')[0]))

output_path = os.path.join(upload_dir, filename)

with open(output_path, 'wb') as output_file:

    # merge files
    for chunk_filename in chunk_list:
        with open(os.path.join(upload_id, chunk_filename), 'rb') as chunk:
            shutil.copyfileobj(chunk, output_file)

    output_file.close()

# delete temporary files
for filename in chunk_list:
    os.remove(os.path.join(upload_id, filename))

checksum = compute_checksum(output_path)
if checksum != expected_checksum:
    os.remove(output_path)
...
```

Рис. 4. Витяг з коду, злиття фрагментів файлу в один

Додаткові випадки

Також необхідно передбачити випадки, коли користувач самостійно скасовує завантаження файлу на сервер. Для цього надсилаємо HTTP DELETE запит, при обробці якого сеанс завантаження завершується, а всі завантажені фрагменти файлів видаляються зі сервера.

Якщо завантаження незавершене і термін сеансу завершився, необхідно передбачити можливість очищення диску від тимчасових файлів задля економії місця. Для таких випадків є можливість реалізувати кінцеві точки (endpoints) та фонові задачі, використовуючи, наприклад бібліотеку Celery.

Висновки

Було розроблену Python бібліотеку, яка дозволяє налаштовувати завантаження і зберігання файлів та їх фрагментів на сервері та об'єднання їх в один файл, що дасть користувачам передавати великі файли.

Система дає можливість переривати і продовжувати завантаження файлів, які не вдалося завантажити повністю. Бібліотека має вбудовану інтеграцію Django і необхідну до нього frontend-логіку.

Ця бібліотека особливо корисна для розробки веб-сайтів, наприклад, для обробки великих текстових документів, відео і аудіо файлів, для подальшої їх обробки, зокрема, методами машинного навчання.

Список літератури

1. Django [Електронний ресурс]. - <https://docs.djangoproject.com/en/4.2/>
2. Uploading Media | Docs | Twitter Developer Platform [Електронний ресурс].- <https://developer.twitter.com/en/docs/tutorials/uploading-media>
3. JavaScript | MDN [Електронний ресурс]. - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
4. Celery [Електронний ресурс]. - <https://docs.celeryq.dev/en/stable/>
5. Chunked Upload – Box Developer Documentation [Електронний ресурс]. <https://developer.box.com/guides/uploads/chunked/>

МОДЕЛЮВАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ПРЕДМЕТІВ ОДЯГУ З ВИКОРИСТАННЯМ ШАБЛОНІВ ПРОЄКТУВАННЯ

Яропуд С.Р.

Львів, Україна

ЛНУ ім. Івана Франка

Факультет прикладної математики та інформатики

Sofia.Yaropud@lnu.edu.ua

У сучасному світі значно зріс попит на купівлю товарів та розвиток індустрії моди через інтернет-застосунки. У зв'язку з цим актуальною є задача обробки даних та методів автоматизації їх підготовки для використання.

Однією з таких задач є задача розпізнавання предметів одягу з зображення (наприклад, фото чи відео). У загальному випадку ця задача є складною та багатокомпонентною. Її розв'язок можна поділити на наступні основні етапи:

- підготовка зображення для дослідження;
- розпізнавання категорії об'єкта;
- визначення характерних розмірів для заданої категорії об'єкта;
- візуалізація отриманих результатів у зручній для користувача формі.

Перший етап передбачає, що ми повинні очистити зображення від шуму, контрастувати його при потребі та перетворювати до потрібного для розпізнавання формату.

Другий етап може здійснюватися різними способами. У нашій роботі ми пропонуємо підхід, з застосуванням процесу навчання на основі нейронних мереж. Для заданої категорії об'єкта використовується визначення розміру за алгоритмом, що доповідався в роботі [4,5].

Для реалізації останнього пункту можуть бути різні вимоги щодо технологій відображення та й потреб самих користувачів системи.

Враховуючи величезний об'єм задачі, складність реалізації підзадач на кожному етапі та можливість розвитку системи, побудова програмного застосунку потребує продуманого проєктування.

Для реалізації цієї мети при створенні архітектури використовується відомий

інструментарій шаблонів проєктування. У роботі запропонований підхід, який дозволяє легко змінювати будь-який основний підпункт зазначеного вище алгоритму, при потребі розширювати його, не завдаючи шкоди іншим ділянкам програми.

Це вдалося реалізувати, використовуючи шаблон Будівельник. Його підсистеми виконують основні складові алгоритму.

У роботі також застосовано шаблон Посередник, який дозволяє вільно розвиватися двом важливим складовим: моделі та системі візуальної подачі. Це дозволить нам, при потребі, створювати різні варіанти за формою і використанням технологій системи збору потреб користувача, а також способів відображення готових результатів. Логіка спілкування між двома підсистемами відбувається в локалізованому конкретному Посереднику, який легко може бути змінений. Водночас, тут в підсистемі модельних класів є Фасад, який делегує задачі для створених “будівельників”.

Шаблон Стратегія допомагає нам реалізувати вибір, з якою системою ми хочемо працювати, навченою чи з повним процесом навчання.

Також цей же ж шаблон допомагає в реалізації для користувача різних систем відображення, що дозволить зробити програмний продукт для різних ролей користувачів (продавця в магазині, дослідника, науковця, клієнта). У майбутньому планується розширення архітектури з побудовою різних інтерфейсів для цих ролей. Для таких цілей можна використати шаблон Замісник.

Запропонована архітектура апробована для реалізації алгоритму, який базується на уточненні наступних складових:

1. Етап підготовки зображень включає тільки процес форматування для потреб бази даних Mnist [2].
2. Система розпізнавання категорії об’єкта здійснюється за допомогою машинного навчання з використанням технології TensorFlow, реалізованої на мові Python [3]. Програмний застосунок передбачає, що ми можемо використовувати як готовий ресурс з навченою системою, так і на етапі дослідження з процесом удосконалення навчання.
3. Для визначення розміру використовується алгоритм, який за допомогою бібліотеки OpenCV дозволяє знайти еталонний елемент на предметі одягу, окреслити характерні границі потрібних областей та визначати опорні точки, за допомогою яких і вдається встановити потрібні характеристики.

4. Визначені розміри предмету одягу відображаються на вихідному зображенні з виносними лініями.

На рис. 1, 2, 3 показані відповідно процеси отримання результату:



Рис.1 Вхідне зображення.

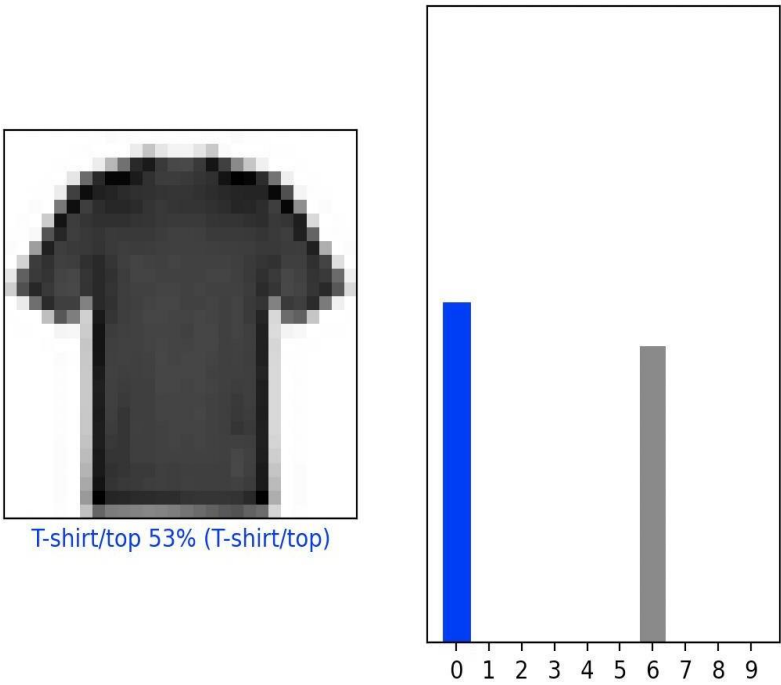


Рис. 2 Визначення категорії.

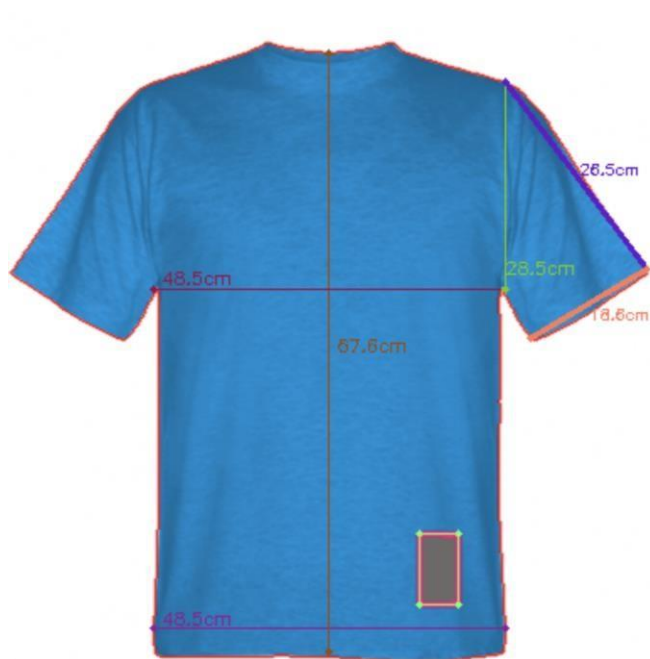


Рис. 3 Результат визначення характерних розмірів.

Чисельні експерименти показали, що отримана модель добре працює для відносно стандартних фасонів елементів одягу з однотонних тканин. Розв’язування проблеми без цих обмежень вимагає додаткових досліджень.

Спроектowana система є доволі гнучкою і вже дозволяє розв’язувати широкий клас задач, а також має можливості для подальшого розвитку.

Список використаних джерел:

1. Adrian Rosebrock. Practical Python and OpenCV [Електронний ресурс] – Режим доступу до ресурсу: <https://minhtn1.github.io/Practical%20Python%20and%20OpenCV,%203rd%20Edition.pdf>.
2. Charu C. Aggarwal. Neural Networks and Deep Learning. - Springer. - 2018. - 520 р.
3. TensorFlow documentation. Classify images of clothing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/tutorials/keras/classification>.
4. Яропуд С. Розпізнавання предметів одягу з використанням штучних нейронних мереж. / Тези доповідей міжнародної студентської наукової конференції з питань прикладної математики та комп’ютерних наук МСНКПМК-2022. - 2022. - С. 141-145.

5. Яропуд С. Визначення точних розмірів предметів одягу з використанням OpenCV. / Тези доповідей міжнародної студентської наукової конференції з питань прикладної математики та комп'ютерних наук МСНКПМК-2021. - 2021. - С. 8-10.

ADVERSARIAL ATTACKS ON OBJECT DETECTION SYSTEMS

Bohdan Buhrii ^{1,2}, Yuriy Muzychuk ¹

¹ *Ivan Franko National University of Lviv (Department of Applied Mathematics)*

² *University of L'Aquila (DISIM)*

bohdan.buhrii@lnu.edu.ua

Abstract

In the last years, object detection tasks were mastered by various kinds of Deep Neural Networks, but there is still a chance of objects being missed or misclassified. The third party, the attacker, might be willing to fail the detection on purpose to benefit from incorrect prediction.

In this paper the AI-based model is built to detect military armored vehicles on satellite images using the state-of-the-art Deep Neural Network. Then its vulnerabilities are exploited to the known adversarial attacks, showing that such systems have a constant need of improvement.

Keywords: machine learning, object detection, adversarial attacks, data poisoning, YOLO.

Introduction

Object detection has many applications, from autonomous vehicles to video surveillance and security. Another example of mission-critical object detection system is military vehicles detection on images taken by satellites or unmanned aerial vehicles. It can bring significant advantages on the battlefield, helping to quickly identify quantity and location of enemy forces. On the other hand, if the system's integrity is violated, it might cause unwanted consequences. Therefore, there is a need to study its security and to be aware of possible threats.

Enemy vehicles detection

We would like to identify the positions of objects in an image, and to know what the object class is. So, having an image, we want to find the set of labels (object's class, coordinates of object's center on the image, width and height of an object) that represent position of each vehicle on the image.

We can solve this problem conveniently using Deep Neural Networks. Since the domain requires high accuracy and real-time performance, we use YOLOv5 [1] model as a core of our system. YOLO (You Only Look Once) works by dividing an image into a grid, with each cell responsible for predicting a fixed number of bounding boxes. Each bounding box includes coordinates, dimensions, confidence

scores, and class probabilities. The neural network processes the image in a single pass, enabling real-time object detection by combining the tasks of localization and classification. This approach is simpler, faster, and more robust compared to traditional object detection methods that use separate stages for region proposals and classification.



Image 1. Examples of military armored vehicles.

The dataset for the model training is based on the DOTA [2] dataset of satellite images, and the images of military armored vehicles, scraped from the internet. The final dataset for our problem contains nearly 1400 images with the number of vehicles from 1 to 15. This is completely enough for the training process, since we use the pre-trained model weights.

Data poisoning

The end goal of an adversarial attack is to cause model to produce inaccurate or completely incorrect predictions. For the system of military vehicle detection, the vector of an attack would be to hide vehicles from the model causing *false negative* predictions.

We use a data poisoning approach, similar to one described in [3]. In these settings, adversaries usually can partially modify the training dataset. Any access or knowledge about the model training is not granted, and not required. The attack happens in the early stage of model development lifecycle. A small portion of a training set is infected with maliciously modified objects, to create a backdoor for the attacker to affect model’s performance during inference.

During poisoning process, to hide the object we add a white patch trigger, like in the image above. We also remove labels for the poisoned objects from the training set.

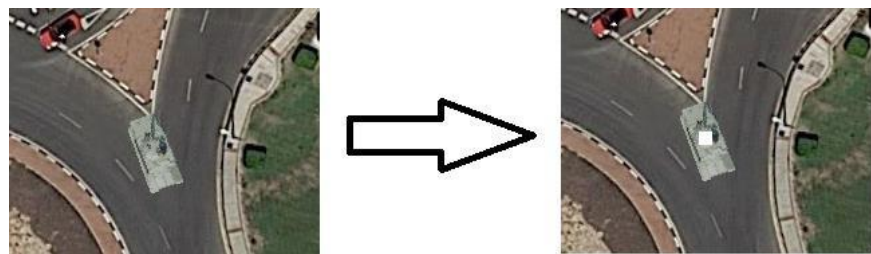


Image 2. Adding the backdoor trigger.

To be able to embed a backdoor trigger successfully, it is enough to modify a small portion of the data. We add trigger to 5% from all the objects in the dataset.

Metrics

We use the mean average precision (mAP) to evaluate the performance of our object detection model. It is calculated by the following formula:

$$mAP = \frac{1}{k} \sum_i^k AP_i$$

Here k – number of classes, AP – average precisions. AP is calculated with the help of several other metrics such as IoU (intersection over union), confusion matrix (TP, FP, FN), precision and recall [4]. It equals an area under precision-recall curve calculated for specified IoU threshold. Precision tells us how many of the total detected objects were actual vehicles. Recall represents how many of total vehicles on the image were detected.

$$\begin{aligned} \text{Precision} &= \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{\text{Correct Predictions}}{\text{Total GroundTruth}} = \frac{TP}{TP + FN} \end{aligned}$$

Results

We found that the constructed model is highly vulnerable to data poisoning attack. It has high performance on the clean validation set, but is not able to detect objects with backdoor trigger in modified validations set, as we can see from the recall curve:

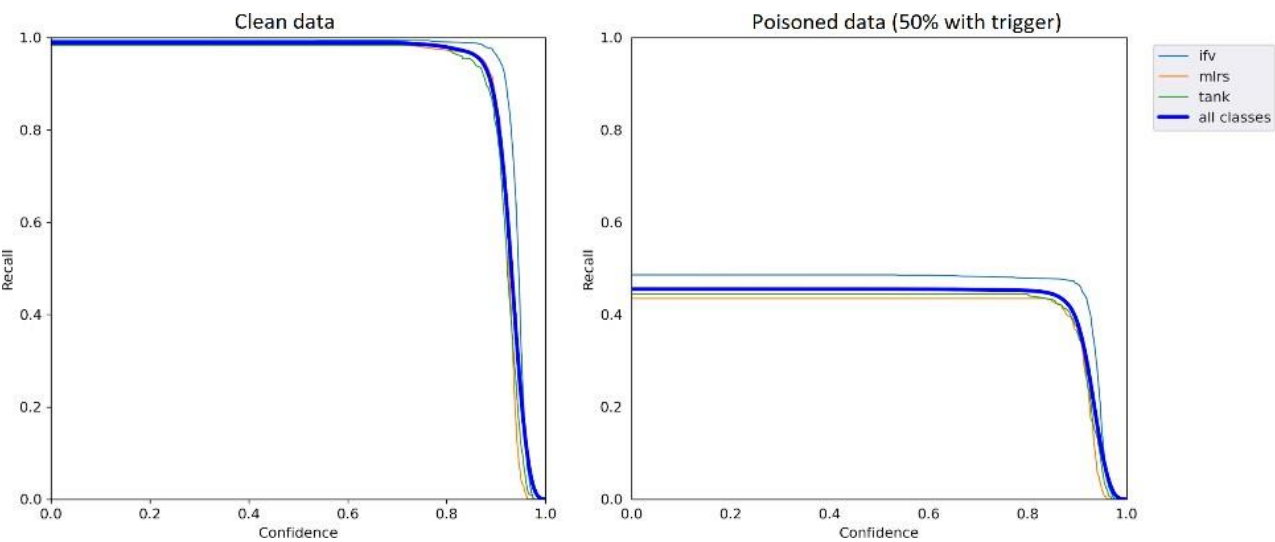


Image 3. Recall curve of poisoned model

We also compared obtained results with the performance of model trained on clean data, on both validation sets. The results are described in Table 1.

Model	Data	Precision	Recall	mAP50	mAP50-95
Clean	Clean	0.997	0.99	0.994	0.873
	Poisoned	0.915	0.89	0.948	0.761
Poisoned	Clean	0.995	0.989	0.993	0.877
	Poisoned	0.993	0.455	0.725	0.641

Table 1. Overall models' performance

The model trained on clean data shows high performance on both validation sets, which empirically proves that poor performance of poisoned model is caused by embedded backdoor trigger.

Conclusion

Our study demonstrates the vulnerability of AI-based object detection models, specifically the YOLOv5 model, to adversarial attacks in the form of data poisoning. The poisoned model shows high performance on clean data but fails to detect objects with backdoor triggers in the modified validation set, highlighting the potential security risks associated with deploying such systems in mission-critical applications like military vehicle detection. To ensure the integrity and robustness of these systems, it is essential to develop defense mechanisms against adversarial attacks and continually improve the models.

References

1. Ultralytics YOLOv5. [Electronic resource]. – Available from: <https://github.com/ultralytics/yolov5>
2. DOTA dataset. [Electronic resource]. – Available from: <https://captain-whu.github.io/DOTA/dataset.html>
3. Chengxiao Luo. Untargeted Backdoor Attack against Object Detection / Chengxiao Luo, Yiming Li, Yong Jiang, Shu-Tao Xia // arXiv.– 2022.– arXiv:2211.05638
4. What is Average Precision in Object Detection & Localization Algorithms and how to calculate it? [Electronic resource]. – Available from: <https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b>

STEGANALYSIS USING DEEP NEURAL NETWORKS

Viktor Seredovych, Yuriy Muzychuk

Ivan Franko National University of Lviv (Department of Applied Mathematics)

University of L'Aquila (DISIM)

viktor.seredovych@lnu.edu.ua

Keywords: steganography; steganalysis; machine learning; deep learning

Introduction

Steganography is a technique for hiding information by concealing the existence of secret communications, used for various applications like confidential communication, secret data storage, and protecting data from alterations. However, steganography can also be exploited for malicious purposes, such as embedding malware in benign files. Therefore, it's crucial to create a reliable steganalysis methods of identifying malicious steganographic files. The main focus of this research is deep neural network learning, which is one of the most promising methods for achieving that goal.

Problem statement

A steganographic system could be defined as a mechanism that embeds *secret message* $m \in \mathcal{M}$ in a cover object $x \in \mathcal{C}$ using a *stego key* $k \in \mathcal{K}$, obtaining the steganographic file $y \in \mathcal{C}$ that carries m [1]. The set \mathcal{M} is the set of all possible messages, \mathcal{K} is the set of all stego keys, and \mathcal{C} is the set of all available cover objects. The embedding mechanism could be described using the embedding mapping

$$Emb: \mathcal{C} \times \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}, y = Emb(x, m, k)$$

This mapping has an inverse extraction mapping that extracts the hidden message from the steganographic file

$$Ext: \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}, Ext(y, k) = m.$$

The locations where embedding modifications take place can be picked at random or using a specific content-adaptive rule.

Steganography algorithms

In this research, we are mostly focused on steganographic algorithms designed for JPEG based format images, such as UERD, J-Uniward and J-MiPOD. These algorithms embed hidden data in the frequency domain of images, making it more difficult to detect while preserving the visual quality of the cover image. Besides the JPEG-specific algorithms, there was also used one general algorithm Least Significant Bit (LSB) algorithm. LSB steganography method embeds secret information into the images by replacing the least significant bits of the pixel values with bits from the secret message.

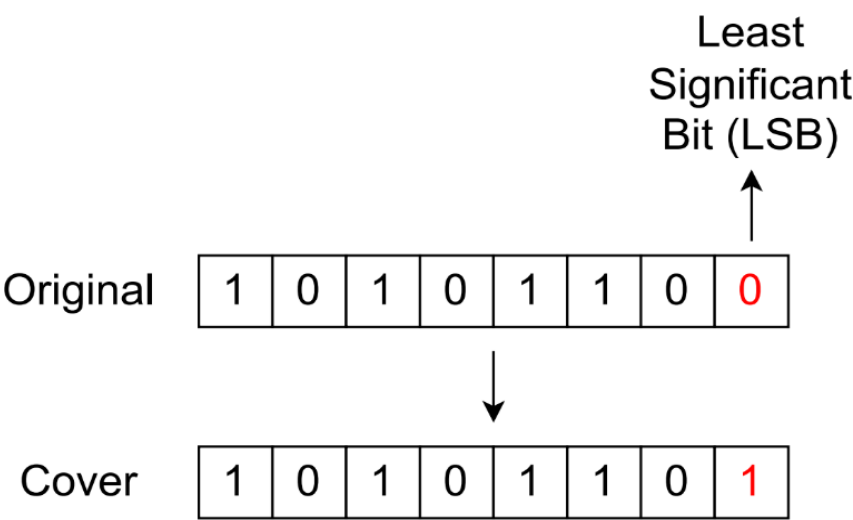


Image 1. The least significant bit of the image is replaced with the bit of the hidden message while preserving the overall structure of the image.

Dataset

The ALASKA2 Image Steganalysis dataset [2], which contains images with concealed messages inserted by various steganography techniques and embedding rates, was utilized as a base of the training dataset. This dataset was expanded by adding embedded images with the LSB algorithm. As a result, the obtained dataset contained 75k clean RGB images with 512x512 resolution and the same number of images for each of the used steganography algorithms which gives a total of 375k images.

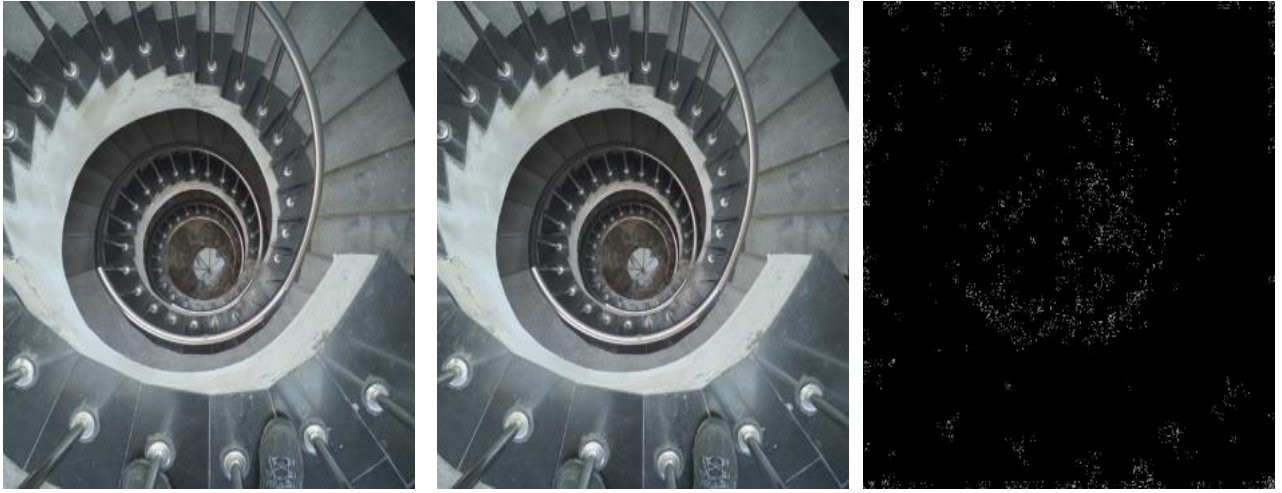


Image 2. Clean image, image with UERD embedded data, scaled differential image between them. The embedding rate of the images in the dataset was randomly generated between 0.1 – 2 bits per pixel.

Model

The neural network model was based on EfficientNet architecture [3] which is a family of scalable and efficient CNN that employs *compound scaling* to simultaneously adjust the depth, width, and resolution of the network achieving state-of-the-art performance on image recognition with reduced model size. The model was trained to classify between clean images and images with embedded content.

The training was done with Adam optimizer. For regularization, we used a *label smoothing* technique that prevents the model from being over-confident by assigning a small probability to incorrect labels, thus improving generalization. For the performance measuring AUC performance metric was used, that quantifies the ability of a classifier to distinguish between positive and negative classes and better describes the model's performance across different classification thresholds.

Results

There were trained several models for datasets of different sizes (15k, 30k, 50k, 70k) split into train, test, and validation sets as 70/15/15 accordingly. The numbers of clean images and images with embedded data were equally balanced for each algorithm.

Training on more data leads to better AUC results, however, it is also considerably slowing down the training process. Based on the obtained results, the minimum number of images required to obtain decent accuracy results is 15k.

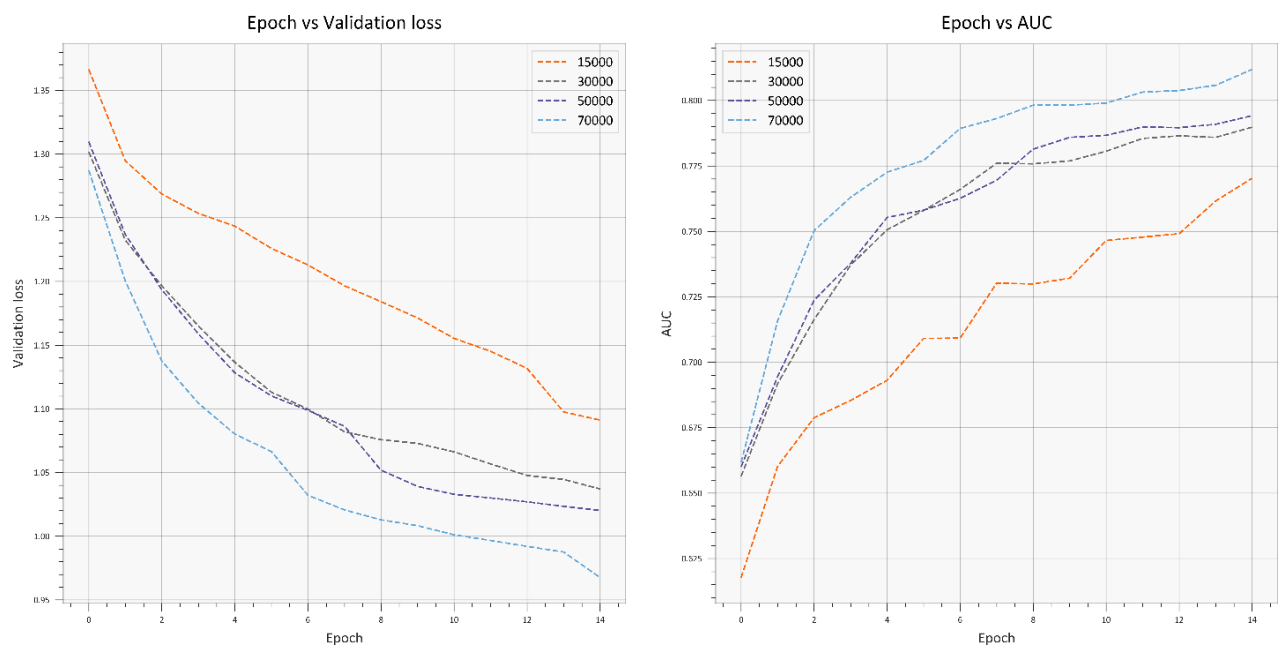


Image 3. Loss and AUC changes for different sizes subsets of images.

Conclusions

In this research, a classification model for steganalysis was trained. The obtained model can detect embedded content in images with decent accuracy for a few well-known steganography algorithms. Future work will be focused on improving the current model by exploring advanced feature extraction techniques, incorporating additional steganography algorithms, and investigating other deep-learning architectures.

References

1. Fridrich J. Statistically Undetectable Jpeg Steganography: Dead Ends Challenges, and Opportunities / J. Fridrich, T. Pevný, J. Kodovský // Proceedings of the 9th Workshop on Multimedia & Security. – 2007

2. Cогranne R. The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis / R. Cогranne, Q. Giboulot, P. Bas // Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. – 2019 [Electronic resource]. – Available from: <https://doi.org/10.1145/1288869.1288872>

3. Tan M. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / M. Tan, Q. V. Le // arXiv preprint arXiv:1905.11946.– 2020. [Electronic resource]. – Available from: <https://arxiv.org/abs/1905.11946>

4. Płachta M. Detection of Image Steganography Using Deep Learning and Ensemble Classifiers / M. Płachta, M. Krzemień, K. Szczypiorski, A. Janicki // Electronics.– 2022.– Vol. 11.– No. 10.– Article number 1565. [Electronic resource]. – Available from: <https://www.mdpi.com/2079-9292/11/10/1565>

SYNTHETIC DATA GENERATION FOR NEURAL NETWORK ANALYSIS OF WAVE PROPAGATION IN DISSIPATIVE FLUIDS

Volodymyr Milchanovskyi

Ivan Franko National University of Lviv

Faculty of Applied Mathematics and Informatics

volodymyr.milchanovskyi@gmail.com

Abstract

This article aims to explore the use of synthetic data generation for analyzing wave propagation in dissipative fluids using neural networks. The approach presented in this work is based on the mathematical model of the wave propagation process. The results demonstrate that this approach can produce datasets that allow neural networks to achieve a good level of accuracy when certain conditions are satisfied. This work highlights the potential of synthetic data generation as a means of enhancing neural network analysis when real-world datasets are limited.

Keywords: Acoustics, Wave Propagation, Synthetic Data Generation, Neural Networks

Introduction

In recent years, the application of neural networks to various data analysis tasks has become increasingly prevalent due to their ability to learn complex patterns in data. However, the performance of neural networks is often limited by the availability of large and diverse datasets. In some cases, obtaining such datasets can be particularly challenging.

Wave propagation in dissipative fluids is a complex phenomenon that has applications in many fields, including acoustics, seismology, and fluid dynamics. Therefore, an accurate analysis of this phenomenon is essential for understanding the underlying physics and making predictions about future events.

Mathematical problem

This section provides only a brief statement of the problem related to wave propagation in dissipative fluids. A detailed description of the variational formulation and solution methods can be found in [1, 2].

Let fluid occupy a limited area Ω of points $x = \{x_j\}_{j=1}^d$, $d = 1, 2$, or 3 and let area Ω has Lipschitz-continuous boundary Γ . The motion of the medium is described by the displacement vector $u(x, t) = \{u_i(x, t)\}_{i=1}^d$ $\forall (x, t) \in \Omega \times [0, T]$ and satisfies the following equation

$$\begin{aligned} \rho u_i'' - \sigma_{ij,j} &= 0, \\ \{\sigma_{ij}(u) &= a_{ijk} e_{km}(u) + c_{ijk} e_{km}(u'), \\ e_{ij}(u) &= (u_{i,j} + u_{j,i}) / 2 \text{ on } \Omega \in (0, T] \end{aligned} \quad (1)$$

with boundary conditions

$$\begin{cases} u_j = 0 \text{ on } \Gamma_U \times [0, T], \Gamma_U \subset \Gamma, \text{mes}(\Gamma_U) > 0, \\ \sigma_{ij} \hat{n}_i = \hat{\sigma}_j \text{ on } \Gamma \times [0, T], \Gamma \setminus \Gamma_U \end{cases} \quad (2)$$

and initial conditions

$$\begin{cases} u_{t=0} = u_0, \\ u'_{t=0} = v_0 \end{cases} \quad (3)$$

where ρ – density, a_{ijkl} – elasticity, c_{ijkl} – viscosity, $u_0, v_0, \hat{\varphi}$ – given functions and $\hat{n} = \{n_i\}_{i=1}^d$ is a unit vector of the outer normal to the boundary Γ .

Experimental model

Consider the following model: a rectangle with dimensions of **0.5 x 2.5** meters filled with water. Inside this domain, there could be some amount of air bubbles simulating contamination. Therefore, the rectangle is divided into five segments for localized testing, see Fig.1.

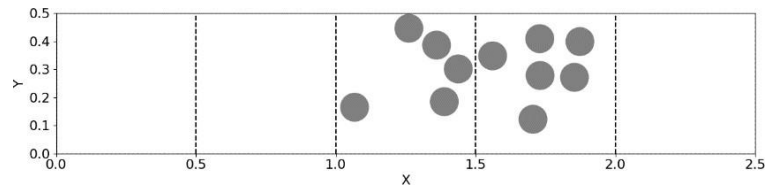


Figure 1: Domain model.

The presence of contamination in one of the segments is analyzed utilizing acoustic waves. Two types of wave propagation are studied: time (Fig. 2) and frequency domain (Fig. 3). In both cases, the integral values of acoustic pressure on the right boundary are calculated. For computations, there are selected 12 time points for the time domain and four frequencies for the frequency domain.

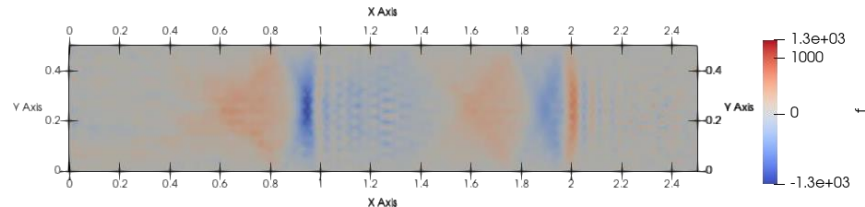


Figure 2: Acoustic pressure in time domain ($t = 1.4e - 4$ s).

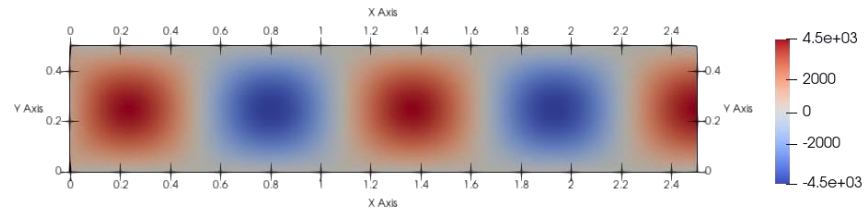


Figure 3: Acoustic pressure in frequency domain ($\omega = 2\pi \cdot 2e3$ Hz).

Bubble generation

To generate air bubbles for the experimental model described above, a generator based on four parameters was constructed:

- 1. The number of segments was calculated using an exponential distribution with $\lambda = 0.5$ plus 1 and limited to the interval $[1,5]$, see Fig. 4.
- 2. The segment index for bubbles was chosen using a normal distribution with $\mu = 2$ and $\sigma = 2$ and limited to the interval $[0,4]$, see Fig. 5.
- 3. Contamination was obtained by a normal distribution with $\mu = 0.25$ and $\sigma = 0.25$ and limited to the interval $[0,0.5]$, see Fig. 6.
- 4. The location of each bubble within its segment was determined using a uniform distribution of coordinates.

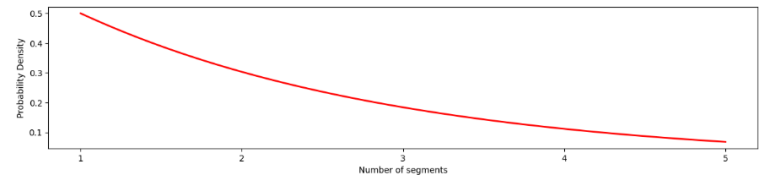


Figure: 4. Distribution of number of selected segments.

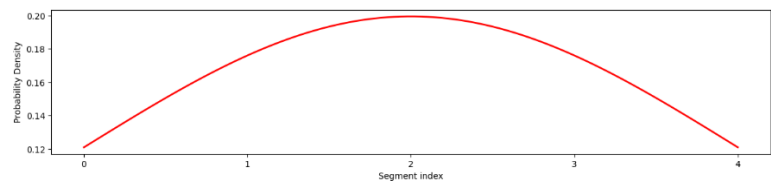


Figure 5. Distribution of segment indexes.

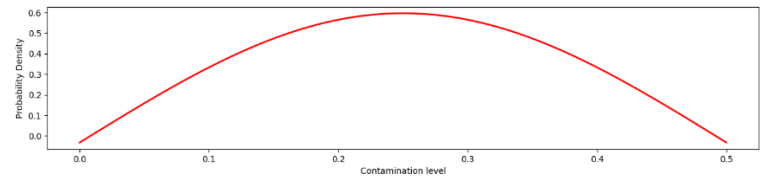


Figure 6. Distribution of contamination level.

Dataset generation

The dataset for both time and frequency domain problems was generated using the method described in [1-2], with additional constraints described in [3]. To facilitate computations, eight virtual machines hosted on a cloud platform were used for distributed computing.

Using this approach, a large dataset consisting of 7,000 records was obtained. The resulting dataset provides a valuable resource for further research into the usage of neural networks for wave propagation analysis and detection of contamination in fluid media.

Results

The resulting data was analyzed using various simple models for classification, including Logistic Regression (LogReg), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), and eXtreme Gradient Boosting (XGB). Before the analysis, the dataset was checked for irregular values and normalized the data to ensure consistency across all models.

The results showed that the classification models performed better for the time domain problem compared to the frequency domain problem. This may be because the time domain data provides more information about the temporal changes in wave propagation, making it easier for the models to classify.

Additionally, the models performed better in predicting the segments located near the edges of the rectangular domain compared to those in the middle, see Fig 7.

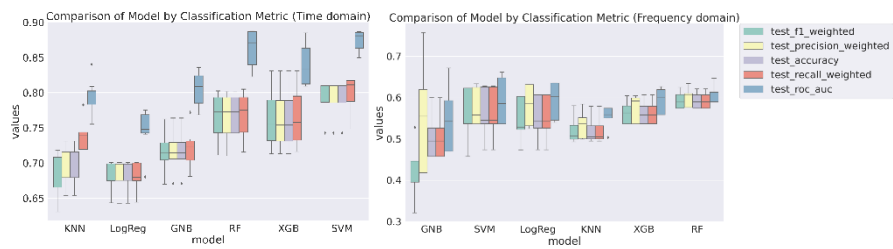


Figure 7: Result of contamination prediction for first segment

Conclusion

Overall, the results of this analysis provide insights into the performance of various classification models for wave propagation in contaminated water. Further research could focus on developing more sophisticated models or exploring other approaches for analyzing the generated dataset.

References

1. Anastasiia Bondarenko and Vitaliy Horlatch. “Mathematical modeling of acoustic wave propagation in viscous fluid”. In: International Student Scientific Conference on Applied Mathematics and Computer Sciences ISSCAMCS (May 2022), pp. 50–53. url: <https://ami.lnu.edu.ua/wpcontent/uploads/2022/05/ISSCAMCS-2022.pdf>.

2. Volodymyr Milchanovskyi and Vitaliy Horlatch. “Mathematical modeling and analysis of stationary acoustic wave propagation in viscous fluid”. In: International Student Scientific Conference on Applied Mathematics and Computer Sciences ISSCAMCS (May 2022), pp. 54–58. url: <https://ami.lnu.edu.ua/wp-content/uploads/2022/05/ISSCAMCS-2022.pdf>
3. Volodymyr Milchanovskyi. “Analysis of PETSc solvers for dissipative acoustics problems using FEniCSx”. Course project.
4. Volodymyr Milchanovskyi. Dissipative-Acoustic. Available from: [<https://github.com/DthRazak/Dissipative-Acoustic>].
5. The FEniCSx computing platform. Available from: [<https://fenicsproject.org>]
6. Logg A. Automated Solution of Differential Equations by the Finite Element Method. The FEniCS Book / A. Logg, K.-A. Mardal, G. Wells. – Springer, 2012. – 723 p.

УСЕРЕДНЕННЯ В ОДНОЧАСТОТНІЙ СИСТЕМІ ІЗ КРАЙОВИМИ УМОВАМИ

Віталій Лучко

Чернівецький національний університет імені Юрія Федьковича

luchko.vitalii@chnu.edu.ua

У даний час розроблено і математично обґрунтовано ряд ефективних методів дослідження коливних явищ, які описуються диференціальними рівняннями та системами таких рівнянь. Одним із таких методів є метод усереднення, який розвинутий представниками київської школи математиків М.М. Криловим, М.М. Боголюбим, Ю.О. Митропольським, А.М. Самойленком та їх учнями [1]. Метод усереднення обґрунтований як для звичайних диференціальних рівнянь, так і для рівнянь із запізненням аргументу і багатоточковими й інтегральними умовами [2, 3].

У даній роботі розглядається система рівнянь першого наближення вигляду

$$\begin{cases} \frac{da}{d\tau} = X_0(\tau, a) + \varepsilon X_1(\tau, a, \varphi), \\ \frac{d\varphi}{d\tau} = \frac{\omega(\tau)}{\varepsilon} + Y_0(\tau, a) + \varepsilon Y_1(\tau, a, \varphi), \end{cases} \quad (1)$$

де $a \in D \subset \mathbb{R}^1$, $\varphi \in \mathbb{R}^1$, ε - малий параметр, $0 < \varepsilon \ll 1$, $\tau = \varepsilon t$ - "повільний час", $\tau \in [0, L]$; функції X_1 і Y_1 2π -періодичні за змінною φ і достатньо гладкі за змінними τ, a, φ . Для системи рівнянь (1) задаються крайові умови

$$\begin{cases} \alpha_0 a|_{\tau=0} + \alpha_1 a|_{\tau=1} = d_1, \\ \beta_0 \varphi|_{\tau=0} + \beta_1 \varphi|_{\tau=1} = d_2, \end{cases} \quad (2)$$

де α_i, β_i і d_i - задані числа, $|\alpha_0| + |\alpha_1| \neq 0$, $|\beta_0| + |\beta_1| \neq 0$.

У відповідність задачі (1)–(2) покладемо значно простішу, усереднену за швидкою змінною φ , крайову задачу

$$\begin{aligned} \frac{d\bar{a}}{d\tau} &= X_0(\tau, \bar{a}) + \varepsilon X_{10}(\tau, \bar{a}), \\ \frac{d\bar{\varphi}}{d\tau} &= \frac{\omega(\tau)}{\varepsilon} + Y_0(\tau, \bar{a}) + \varepsilon Y_{10}(\tau, \bar{a}), \end{aligned} \quad (3)$$

$$\alpha_0 \bar{\phi}_{\tau=0} + \alpha_1 \bar{\phi}_{\tau=L} = d_1,$$

$$\beta_0 \bar{\phi}_{\tau=0} + \beta_1 \bar{\phi}_{\tau=L} = d_2, \quad (4)$$

Нехай $\bar{a} = \bar{a}(\tau; \bar{y}, \varepsilon)$ – розв’язок усередненої задачі (3), (4), $\bar{a}(0; \bar{y}, \varepsilon) = \bar{y}$, $\mu(\varepsilon)$ – досить мале число, оцінку якого отримано в роботі, і $\bar{a} = \bar{a}(\tau; \bar{y} + \mu, \varepsilon) \in D$ для всіх $\tau \in [0, L]$ і $\varepsilon \in (0, \varepsilon_0]$. Як доведено в [1], існує єдиний розв’язок $a(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon)$, $\xi(\varepsilon) \in \mathbb{R}$ і досить мале, що справджується оцінка

$$|a(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon) - \bar{a}(\tau; \bar{y} + \mu, \varepsilon)| + \\ + |\varphi(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon) - \bar{\varphi}(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon)| \leq c_1 \varepsilon^2,$$

де $c_1 > 0$ і не залежить від ε .

Застосувавши цю оцінку і методику, запропоновану в [1], знайдено початкові умови для єдиного розв’язку крайової задачі (1), (2).

Теорема. Нехай виконуються умови:

- 1) функції X_0, X_1, Y_0, Y_1 і ω – двічі неперервно диференційовані в області $[0, L] \times D \times \mathbb{R}$, $D \subset \mathbb{R}$;
- 2) $\omega(\tau) \geq c > 0$, $\tau \in [0, L]$;
- 3) $\alpha_0 + \alpha_1 \frac{\partial \bar{a}}{\partial \bar{y}}(L; \bar{y}, \varepsilon) \neq 0$ для $\varepsilon \in (0, \varepsilon_0]$, $\beta_0 + \beta_1 \neq 0$;
- 4) існує єдиний розв’язок задачі (3), (4) $\forall \varepsilon \in (0, \varepsilon_0]$.

Тоді для досить малого $\varepsilon^* \in (0, \varepsilon_0]$ існує єдиний розв’язок задачі (1)–(4) і виконується нерівність

$$|a(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon) - \bar{a}(\tau, \bar{y}, \varepsilon)| + \\ + |\varphi(\tau; \bar{y} + \mu, \bar{\psi} + \xi, \varepsilon) - \bar{\varphi}(\tau, \bar{y}, \bar{\psi}, \varepsilon)| \leq c_2 \varepsilon^2.$$

Для μ і ξ справджуються оцінки

$$|\mu| \leq c_3 \varepsilon^2, \quad |\xi| \leq c_4 \varepsilon^2,$$

де c_2, c_3 і c_4 – додатні сталі, які не залежать від ε .

У роботі також написано програму числового моделювання розв’язку точної й усередненої задач.

Список літератури

1. Самойленко А.М., Петришин Р.І. Математичні аспекти теорії нелінійних коливань. Київ: Наукова думка, 2004. 475 с.
2. Yaroslav Bihun, Roman Petryshyn, Ihor Skutar, and Halyna Melnyk Multifrequency system with multipoint and integral conditions. Acta et Coomentationes, Exact and Natural Sciences. Nr. 2(12), 2021. P. 11–24.

3. Бігун Я. Й., Скутар І. Д. Усереднення в багаточастотних системах із запізненням та локально-інтегральними умовами. Буковинський математичний журнал. 2020. Т. 8, № 2. С. 14–23.

ЗАСТОСУВАННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ АНАЛІЗУ РИНКУ КРИПТОВАЛЮТ

Шевченко Ірина

Львівський національний університет ім. Івана Франка

Факультет прикладної математики та інформатики

iryna.shevchenko.pmi@lnu.edu.ua

Однією з актуальних задач на сучасному фінансовому ринку є аналіз динаміки цін на криптовалюту з використанням математичних моделей. У зв'язку зі зростаючим інтересом до цифрових валют та збільшенням їх ринкової капіталізації, розвиток наявних та створення нових математичних підходів для аналізу динаміки цін на криптовалюту має велике практичне значення.

Ключовою перевагою застосування математичних моделей для аналізу ринку електронних грошей є їх здатність до аналізу великої кількості даних та знаходження складних залежностей між ними [1]. Наприклад, можна використати моделі кластеризації, щоб згрупувати різні криптовалюти за спільними ознаками та порівнювати динаміку цін у межах кожного кластеру. Існують моделі, які дозволяють аналізувати динаміку цін на криптовалюту за допомогою математичних формул та індикаторів. Наприклад, можна використовувати моделі засновані на технічному аналізі, який використовує історичні дані цін та торгових об'ємів для вивчення ринку і прогнозування майбутніх цінових тенденцій.

Оцінити можливі збитки у разі негативного зміщення ціни криптовалюти можна за допомогою моделей відображення ризику Value at Risk, VaR [2]. VaR є математичною моделлю, яка оцінює максимальний можливий збиток, який може виникнути при визначеному рівні надійності.

Модель Facebook Prophet базується на аддитивній регресії з компонентами, що враховують сезонність та тренд. Вона використовується для прогнозування цін на цифрові валюти на короткій та середньостроковій перспективі.

Моделі аналізу графів та мереж можуть допомогти вивчити структуру ринку криптовалют та виявити зв'язки між різними криптовалютами та їх відносними цінами. До таких моделей відносяться PageRank, HITS, та моделі теорії графів [3].

Математичні індикатори, такі як RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), Bollinger Bands та інші дозволяють оцінювати рівень перекупленості або перепроданості активу, виявляти сигнали на вхід або

вихід з ринку та інші корисні параметри.

Для аналізу фундаментальних показників криптовалют можна використати математичну модель оцінки фінансової стійкості криптовалютного проєкту, яка використовує такі показники, як дохідність, заборгованість, ліквідність та інші.

За допомогою моделі обробки природної мови та моделі аналізу сегменту аналізують тексти на інтернет-форумах та соціальних медіа, де обговорюється динаміка цін на криптовалюту та поведінка ринку загалом [3]. Це дозволяє виявити настрої трейдерів та інвесторів та зрозуміти, які фактори впливають на їх рішення щодо купівлі та продажу криптовалют.

На підставі огляду багатьох математичних моделей, які аналізують ринок криптовалют, хотіла б виділити моделі ARIMA та LSTM, які, на мою думку, є найбільш ефективними під час дослідження динаміки цін на цифрову валюту.

Математична модель авторегресійної інтегрованої ковзної середньої ARIMA (Autoregressive Integrated Moving Average) [4] є однією з найпоширеніших моделей для аналізу часових рядів. ARIMA модель зазвичай використовується для прогнозування та аналізу часових рядів, які мають стаціонарність та не мають виразних трендів або сезонності. Якщо часовий ряд ціни криптовалюти не є стаціонарним, то його можна перетворити за допомогою методу диференціювання на стаціонарний і використовувати ARIMA модель для прогнозування ціни криптовалюти на майбутнє. Зазвичай вона складається з трьох параметрів: p , d та q . Параметр p відповідає за кількість авторегресійних членів в моделі. Ці члени включають попередні значення ряду. Параметр d відповідає за ступінь диференціювання. Це означає, що ряд може бути перетворений на стаціонарний шляхом віднімання попереднього значення від поточного. Параметр q відповідає за кількість ковзних середніх у моделі. Ці середні включають згладжені значення помилок попередніх прогнозів.

Формула ARIMA (p, d, q) [5] може бути записана як:

$$y_t = c + \sum_{i=1}^p \phi_i(y_{t-i} - \mu) + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

y_t - значення ряду в часовій точці t ; c - константа; ϕ_i - коефіцієнти авторегресії; y_{t-i} - значення ряду в часовій точці $t-i$; μ - середнє значення ряду; θ_j - коефіцієнти ковзної середньої; ε_{t-j} - помилки попередніх прогнозів; ε_t - помилка поточного прогнозу.

Модель LSTM [6] може працювати з послідовними даними будь-якої форми та не потребує стаціонарності даних, що робить її більш гнучкою в порівнянні з

ARIMA. LSTM (Long Short-Term Memory) - це рекурентна нейронна мережа, що має додаткові "ворота пам'яті", які дозволяють зберігати та використовувати інформацію на довгі періоди часу [7]. Ця модель добре підходить для аналізу ринку криптовалют, оскільки цей ринок характеризується високою волатильністю та швидкими змінами цін.

Для LSTM моделі кількість прихованих шарів, кількість нейронів у кожному шарі, функції активації воріт та інші параметри можуть бути визначені шляхом зведення задачі до задачі класифікації або регресії та використання методів оптимізації для знаходження оптимальних значень параметрів.

Створена мною комп'ютерна програма мовою програмування Python за моделлю ARIMA з використанням бібліотек Pandas, Requests, Matplotlib дозволяє з невеликою похибкою (3-7 %) на основі історичних даних ціни електронної монети Bitcoin спрогнозувати ціну на майбутнє та перевірити її точність у порівнянні з актуальною ціною (див. рис. 1).

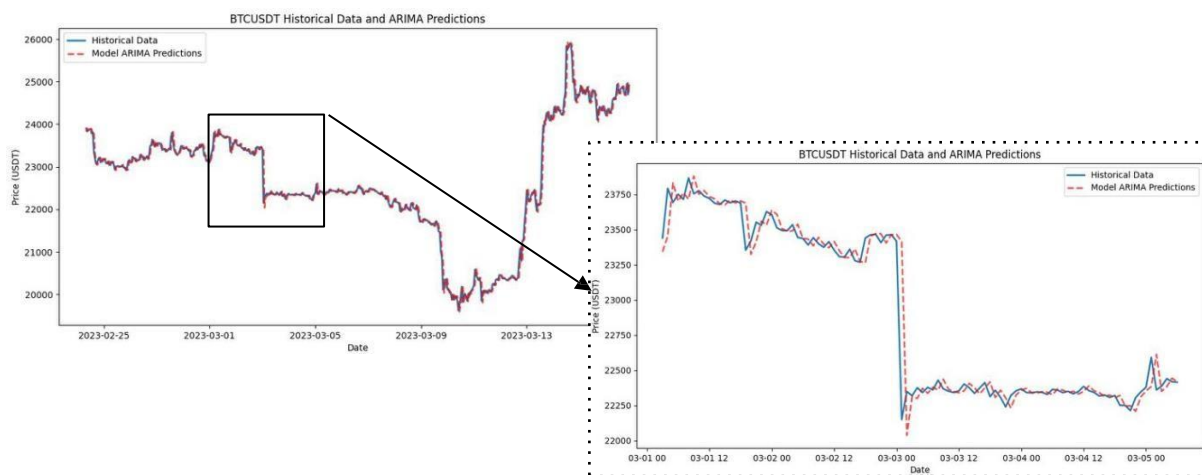


Рис. 1 – Результат комп'ютерної програми за моделлю ARIMA

Використання математичних моделей та індикаторів для аналізу ринку криптовалют може допомогти інвесторам та трейдерам робити більш обґрунтовані рішення щодо управління своїми інвестиціями та ризиками. Зокрема, ARIMA та LSTM моделі для аналізу ринку електронних грошей є потужними інструментами для прогнозування ринкових тенденцій та цінових коливань криптовалют.

Список джерел та літератури

1. "Introduction to Cryptocurrencies" by Andreas M. Antonopoulos
2. "Applied Time Series Analysis and Forecasting" by Richard Harris.

3. "Cryptocurrency Investing Bible: The Ultimate Guide About Blockchain, Mining, Trading, ICO, Ethereum Platform, Exchanges, Top Cryptocurrencies for Investing and Perfect Strategies to Make Money" by Alan T. Norman
4. <https://medium.com/towards-data-science/time-series-forecasting-arma-models-7f221e9eee06>
5. <https://www.machinelearningplus.com/time-series/arma-model-time-series-forecasting-python/>
6. "Cryptoassets: The Innovative Investor's Guide to Bitcoin and Beyond" by Chris Burniske and Jack Tatar
7. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>

**АНАЛІЗ ВИКОРИСТАННЯ МЕТОДІВ РЕАКТИВНОГО
ПРОГРАМУВАННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ВИСОКОЇ
ДОСТУПНОСТІ ТА НАДІЙНОСТІ МІКРОСЕРВІСНИХ СИСТЕМ
НА ОСНОВІ JAVA**

Олексій Дацюк

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

oleksii.datsiuk@lnu.edu.ua

Розглянемо типову для мікросервісної архітектури аплікацію, що складається з 5 мікросервісів, кожен з яких є типовою 3-шаровою аплікацією і оперується мовою програмування Java та веб-фреймворком Spring boot.

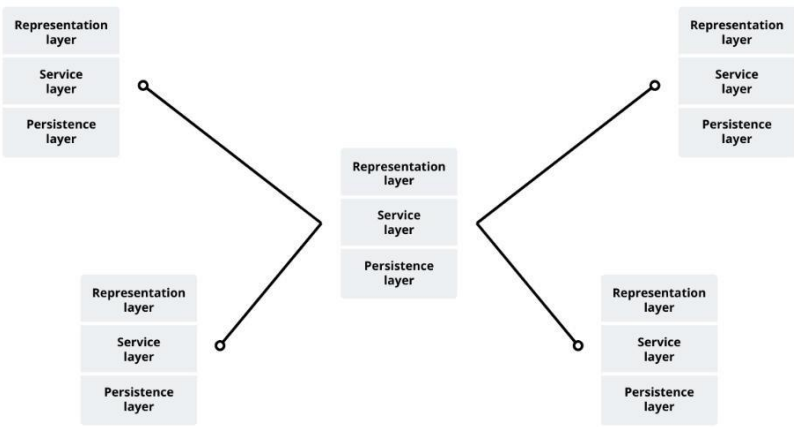


Рис. 1. Архітектура типової мікросервісної аплікації

Зрозуміло, що мікросервіси комунікують між собою – обмінюються інформацією. І аби поринути у проблему варто уточнити, як саме аплікація на базі Java та Spring Boot отримує та обробляє запити, що їй надходять. Отож, кожна Spring Boot аплікація працює на базі веб-сервера – Servlet Container. Кожен подібний веб-сервер оперує «пулом» потоків для обробки запитів від користувачів, або ж інших аплікацій. «Пул» потоків – фіксована кількість потоків, що може використовуватися та перевикористовуватися аплікацією на базі Spring Boot. Використання «пулу» потоків для обробки запитів є поширеною практикою, адже в такому випадку запити будуть оброблятися асинхронно, а також, за рахунок перевикористання вже існуючих потоків, а не створення нових, аплікація працюватиме значно швидше. Проте, у такого підходу до створення мікросервісних аплікацій є кілька недоліків, один з яких: для кожного запиту виділяється один потік з «пулу», надалі цей потік блокується на час виконання бізнес-логіки, яку потребує запит. Це не буде проблемою доки в аплікації не з’явиться функціонал, виконання якого потребує значних ресурсів та часу, а також доки на подібний

функціонал не почне надходити величезна кількість запитів з інших мікросервісів. Розглянемо найгірший варіант, у випадку надходження монструозної кількості запитів на виконання подібної логіки, сервіс може стати перевантаженим, як висновок, перестати обробляти наявні запити, а також не відповідати на новоприбулі. Потенційно, вирішенням цієї проблеми може стати горизонтальне розширення, проте такий спосіб вирішення наявної проблеми є доволі дорогим, а також не зовсім практичних у всіх випадках. До того ж, горизонтальне розширення може бути неефективним, якщо мова йде про лімітовану мережу, або ж ресурси I/O, а не брак обчислювальної потужності. Таким чином ми підходимо до теми реактивного програмування. Реактивне програмування - це декларативна парадигма програмування, в основі якої лежить ідея асинхронної обробки подій та потоків даних.

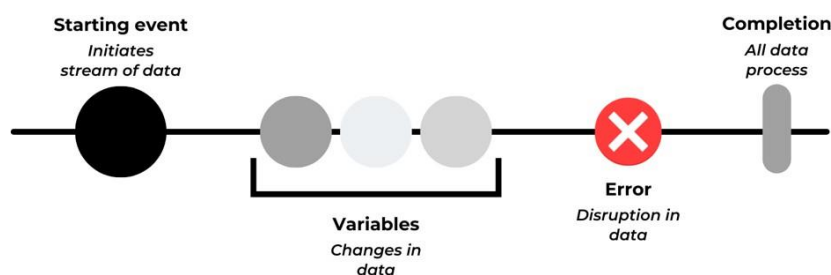


Рис. 2. Принцип роботи реактивного потоку

Аби краще зрозуміти як саме працює реактивне програмування - можна згадати про один дуже цікавий ООП патерн – «Observer», суть якого полягає в тому, щоби створити систему "підписки" об'єктів на події, після чого при виконанні цієї події, відповідні об'єкти будуть повідомленні про її виконання. Але оскільки в контексті реактивного програмування обробка однієї події не робитиме ніякого сенсу - доводиться працювати з реактивними потоками подій. У випадку роботи з реактивними потоками, те, на що підписуватимуться умовні об'єкти буде не одна подія, а потік подій. У такому випадку «обсервер» має надати відповідні колбеки про те, потік виконався з помилками, чи без. Зокрема, реактивні потоки відрізняються в контексті того, як контролюється реактивний флоу - можлива «push» та «pull» методика. У випадку «push» методики «підписник» точно не знає коли подія буде виконана, відповідно, «паблішери» можуть перенаситити підписників запитами, що в свою чергу створить вищеописану проблему. Саме тому підписники мають мати можливість контролювати реактивний флоу зі свого боку, у «pull» методиці це називається «backpressure». Суть полягає в тому, щоби

надати підписнику можливість самотужки регулювати надходження запитів, тобто, припинити його за надмірної подачі.

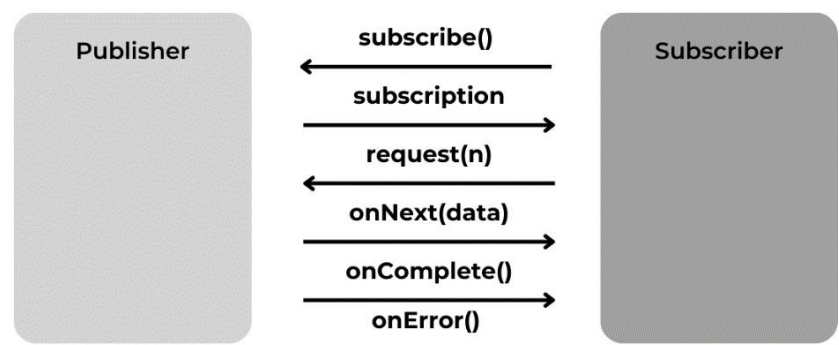


Рис. 3. Емуляція реактивного флоу

Відповідно, головними особливостями реактивного програмування та реактивних потоків є асинхронність, «non-blocking», «event-driven» та «backpressure». Ну і оскільки мікросервіси переважно контейнеризуються, а ресурси обмежені, то всі вище-вказані особливості є важливими в умовах роботи мікросервісної аплікації. Оскільки вони збільшують ефективність використання наявних ресурсів та підвищують швидкодію аплікації за умови правильного використання.

Список літератури

- 1. Reactive Streams – Режим доступу: <https://www.reactive-streams.org>
- 2. Reactive Manifesto – Режим доступу: <https://www.reactivemanifesto.org>
- 3. Project Reactor – Режим доступу: <https://projectreactor.io/docs/core/release/reference/>

АНАЛІЗ ПРОЦЕСУ РОЗРОБКИ ВЕБСАЙТУ НА ОСНОВІ КОРИСТУВАЦЬКОГО ДОСВІДУ

Вовк Любов-Василина

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Vasylyna.vovk2002@gmail.com

Білл Гейтс говорив: **"Якщо вашого бізнесу нема в інтернеті, то вас нема в бізнесі"**. Цифровий світ вже не новина, а наша реальність, тому відсутність власного веб-сайту максимально не зважене рішення для бізнесу сьогодні. Але далеко не кожен сайт здатен виконувати потрібні функції. На це впливає ряд чинників, кожен з яких має певну вагу на етапі розробки результативного сайту.

Я вирішила дослідити, що ж впливає та забезпечує результативність і метою мого проєкту було проаналізувати процес розробки веб-сайту, визначити головні аспекти та використати їх у розробці ресурсу, який буде функціональним

Отож, спершу я дослідила з яких етапів складається процес розробки та що він в собі містить сформувався загальний список і я почала досліджувати кожен з них

Отже, першим пунктом я виділила **стратегію**. На цьому етапі ми визначаємо мету розробки нашого сайту, що хочемо досягти та які інструменти треба для цього залучити, коротко кажучи відповідаємо на запитання: а навіщо нам сайт?

Цілей, які може закривати наш сайт може бути багато, ось кілька з них: продавати (для e-commerce), залучати клієнтів, презентувати продукт чи лінійку товарів, розповісти про послугу, збирати ліди (контактні дані для клієнтської бази), розповісти про вашу компанію, вибудувати довгострокові відносини із клієнтами, інформувати ЦА, підвищити охоплення та впізнаваність бренду, використовуючи digital-канали.

Один сайт може закривати одразу кілька цілей, власне, так і вийшло у нашому проєкті. Але все ж цей етап один із найважливіших, бо на його основі будуються наступні етапи

Наступний масштабний етап, який я виокремила - це **аналітика**. Тут ми визначаємо цільову аудиторію нашого сайту, їх болі та потреби, переглядаємо та аналізуємо конкурентів, фактично відповідаємо на наступні запитання: для кого призначено веб-ресурс, які уподобання ваших потенційних клієнтів, яка інформація буде для них корисною та цікавою, якими комунікаційними каналами вони користуються.

Коли основна інформація зібрана, переходимо до 3 етапу - побудови структури сайту, прототипування та дизайну. Власне, основну частину мого дослідження становив цей пункт, тому тут я розкриватиму багато моментів

Отже, структура. Коли користувачі заходять на сайт, швидше за все, вони хочуть отримати детальну інформацію, яку не знайшли до прикладу у соц мережах, в них є проблеми, з якими вони завітали на сайт і за лічені хвилини, нам потрібно їх переконати у тому, що ми можемо їм допомогти. Тобто, нам потрібен сайт, який пройде перевірку сумнівами і перетворить переглядачів на покупців. За дослідженням Дональда Міллера, я винесла 5 речей, які повинен мати вебсайт

- 1. Пропозиція у верхній частині екрану .** Перший екран, це те, що бачить потенційний клієнт зайшовши на сайт, і саме тут він чітко мав би зрозуміти, що має отримати, головну думку, що отримає клієнт від співпраці з бізнесом, як його рекомендувати іншим

Варто переконатись , що фото на текст, які ви використовуєте вгорі головної сторінки відповідають наступним правилам:

- Говорять про образ, який надихає. Якщо ми пропонуємо людині перетворити його на гуру програмування, іт-школа формує образ, який надихає і пояснює для чого їй придбати ці курси
- Обіцяють розв'язати проблему. Якщо бізнес може розв'язати якусь проблему клієнта, то про це треба згадувати на головній. Користувачі заходять на сайт не для того, аби дізнатися скільки у вас нагород, а щоб вирішити свою проблему
- Чітко повідомляють про види діяльності компанії. Найпростіший із варіантів, просто розповісти про види діяльності компанії, умовно коли ви пишете під назвою вашої компанії підтвердження ваших послуг . Ми

продаємо їжу, одяг чи аксесуари. Це теж викличе асоціацію у людей і розуміння, що можна отримати

2. **Очевидний заклик до дії** Головна мета нашого сайту — це створити місце, де кнопка з СТА буде доречною та красивою. Навіть надаючи найкращий сервіс бізнес може вилетіти з ринку, якщо люди не натискатимуть кнопку “купити” зараз. Тож цю кнопку має бути чітко видно

Є 2 основні паттерни, які рухають погляд користувача, відповідно на них і треба опиратись, розміщуючи наші кнопки, це F-паттерн та z-траєкторія, тоу на їх основі треба розміщувати кнопки.

3. **Зображення успіху** Більшу частину нашого повідомлення до людей становить текст, але не тільки, ключовими у сприйнятті є також зображення, які ми використовуємо, вони теж інформують і доносять сенси.
4. Розподіл потоків ваших прибутків на зручні порції
5. Якнайменше слів

Далі теж не менш вагомий аспект розробки сайту - евристики юзабіліті.

Ідеальний user experience має на меті створити максимально комфортний для користувача сайт, який він не захоче лишити та якому залишить свої гроші, придбавши певний товар чи замовивши певну послугу (і не тільки). Ці **евристики юзабіліті** дослідив Якоб Нільсен, взагалі їх 10, але я зупинилась на 5 основних

1. Інтуїтивність. Дизайн розроблено згідно з реальним світом.
2. Користувач контролює все. Завжди є можливість повернутися назад чи взагалі вийти — це дає додаткове відчуття безпеки і комфорту.
3. Чіткість, послідовність та стандарти. Дизайн має бути системним, якщо розмір шрифту для основного тексту 17 пікселів, то він має бути точно таким же на всіх сторінках сайту в межах 1 проєкту
4. Доцільні нагадування. Якщо є форма варто прописати назву кожного з полів
5. Гнучкі рішення. Запропонуйте декілька варіантів, щоб користувачу було складніше відмовитися + запевніться, що у користувача завжди буде вирішення його проблеми наприклад, товар в онлайн-магазині користувач може оплатити як через PayPal та і через Stripe

Завершивши з аналітичними складовими, переходжу до безпосередньо дизайну, кольористики та типографії. Тут я дослідила види колірних схем,

асоціативність кольору, коло за Ітеном, правило 60/30/10, місце кольору в результативності сайту, а також види шрифтів, формування шрифтових пар, ієрархію в межах проекту, розміри шрифтів, керлінг, інтерліяж

Наступний етап розробки – верстка. Вона не має найбільший вплив на результативність нашого сайту в ході дослідження за винятком того, що повинна бути якісною та підтримувати правильну адаптацію.

реалізовувати її можна різними методами: як чистим кодом за допомогою HTML, CSS, JS, за допомогою конструкторів чи за допомогою систем керування вмістом з відкритим кодом (до них відноситься вордпрес) і звісно, окремим дослідженням можна визначати плюси/мінуси/переваги кожного з цих варіантів

Наступний і останній етап UX-дослідження та визначення результативності сайту.

Якщо перейти до досліджень, то їх насправді, варіантів є дуже багато, кожен з яких має застосування на різних етапах, ті які я використала для аналізу спеціально обрала для фінального часового проміжку, аби вони були в рівних умовах

- 1. A/B-тестування (A/B Testing).** Популярний метод порівняння двох версій сайту або програми, що відрізняються одним або декількома елементами. Аудиторія випадковим чином розбивається на два сегменти, кожен з яких бачить тільки одну версію. Після досягнення статистичної значущості робиться висновок, який варіант переміг за обраним KPI (наприклад, покупки у додатку). Проводиться в спеціальних сервісах, таких як Google Optimize для сайтів і Optimizely для додатків.
- 2. Експертний огляд (Expert Review).** Продукт рецензує досвідчений UX-фахівець, оформлюючи свої спостереження в докладний звіт з ілюстраціями. Документ описує сильні і слабкі сторони рішення, виявлені проблеми та їх причини, рекомендації щодо усунення та найкращі практики. Також підходить, коли команді UX-дизайнерів не вистачає власної експертизи.
- 3. Аналіз клікстріма (Clickstream Analysis).** Аналіз даних про те, які сторінки і в якому порядку відвідував користувач. Легко провести за допомогою системи аналітики Google Analytics, Firebase, Mixpanel. Дозволяє виявити

проблеми, пов'язані з навігацією по сайту або програми. Не допомагає з пошуком їх причин. Для з'ясування причин варто використовувати юзабіліті-дослідження.

4. **Інтерв'ю (Interviews).** Особисте інтерв'ю з людьми, що мають досвід роботи з продуктом. Завдання — зрозуміти, яке враження залишає сайт або додаток у користувачів. Оскільки мова йде про минулих діях, точних чисельних даних отримати не вдасться.
5. **5-секундний тест (Five second test).** Бліц-тест на перше враження від дизайну. Учасники дослідження бачать його фрагмент рівно 5 секунд, після чого відповідають на низку питань. Наприклад, про те, які основні елементи запам'яталися, яке уявлення про бренд склалося, яке призначення сторінки, хто цільова аудиторія і т.д.

Не завжди можна оцифрувати результативність сайту, просте є і такі можливості. Для цього існує визначення конверсія — це відношення числа відвідувачів сайту, які виконали на ньому якісь цільові дії, до загального числа відвідувачів сайту, виражене у відсотках. Тобто, якщо на наш сайт зайшло 500 людей, а записалися на послуги 100 з них, то наша конверсія складає $100/500 * 100\% = 20\%$. Google на питання нормальності конверсії відповідає дуже лаконічно — 2–5%.

Ну і, звісно, 2 варіант, як ми можемо оцінити результативність сайту, це порахувати дохід від клієнтів, які прийшли через сайт та визначити його фінансову ефективність для бізнесу

Отже, у підсумку. всі найважливіші візуальні моменти я вивела у власну таблицю, в якій надала власну експертну оцінку, по важливості цих складових. Також для цих критеріїв я використала метод ієрархії з теорії прийняття рішень, результати якого ще в процесі реалізації, але він допоможе нам дати експертну оцінку з математичної складової

критерії Е					
структура за міллером 30%	пропозиція у верхній частині 40%	якнайменше слів 10%	очевидний заклик до дії 30%	зображення успіху 10%	розподіл потоків ваших прибутків на зручні позиції 10%
кольористика 10%	колірна схема 40%	співвідношенн я кольорів (правило 60/30/10) 20%	асоціативність кольору 15%	яскравість?? 5%	кількість кольорів в проекті 20%
типографія 15%	розмір 25%	тип шрифту 25%	ієрархія 40%	накреслення (курсив, жирний підкреслення) 5%	міжрядкові та міжбуквенні зміни (керлінг, інтерліяж, капс?) 5%
евристики юзабіліті 30%	Інтуєтивність 20%	2. Користувач контролює все 25%	Чіткість, послідовність та стандарти. 30%	Доцільні нагадування 5%	Гнучкі рішення 10%
юх дослідження 15%	Аналіз кликстрима 40%	5-секундний тест 15%	А/В- тестування 10%	Експертний огляд 15%	Інтерв'ю 20%

Власне, якщо перейти до практичного застосування, то врахувавши всі аспекти процесу розробки веб сайту, я розробила лендінг (односторінковий сайт) для стоматологічної клініки

Цей сайт був створений для реального закладу, який знаходиться в Ужгороді та має назву “Клініка Сімейної Стоматології” і в результаті роботи цього сайту, всього за 1 місяць з нього власники продали послуг у сумі більше 8 тисяч доларів, що підтверджує моє дослідження.

МОДЕЛЮВАННЯ ЗМІНИ РОЗМІРІВ РАКОВИХ КЛІТИН

Красічинський Станіслав

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

stanislav.krasichynskyi@lnu.edu.ua

Ракові пухлини є однією з найбільш важливих і актуальних проблем сучасної медицини. Кожен рік в світі більш як 9 мільйонів людей помирає від раку, що створює значне соціальне та економічне навантаження на суспільство. Багато вчених працюють над проблемою раку, вносячи свій вклад у боротьбу з цією хворобою. Серед них можна виділити роботи таких вчених, як Дж. Аллісона, Дж. Фолкмена, Л. Рануцці та багатьох інших, їхні дослідження стали ключовими для вивчення раку та розробки нових методів лікування.

Сучасна комп'ютерна математика відкриває нові можливості в дослідженні раку. Застосування систем комп'ютерної математики дозволяє створювати математичні моделі, які можуть допомогти у прогнозуванні поширення ракових клітин. При цьому враховуються такі фактори, як стратегії діагностики та лікування, міжклітинні взаємодії та імунні відгуки. Моделювання поширення ракової пухлини за допомогою систем комп'ютерної математики є потужним інструментом, який розширює можливості вивчення цієї хвороби та дозволяє створювати більш точні методи діагностики та лікування. Зокрема, моделювання змін росту ракових клітин може пролити світло на механізми зростання і проліферації пухлинних клітин.

У даній курсовій роботі було продемонстровано та проаналізовано одну з моделей поширення ракових клітин за допомогою середовища MATLAB. Модель запропонована Шерраттом і Чапленом у 2001 році, ґрунтується на часткових диференціальних рівняннях. Диференціальні рівняння описують ріст, рух і смерть пухлини, що супроводжується надходженням поживних речовин до неї. Модель розв'язується чисельно з використанням методу скінченних різниць. Результати виконання симуляції перевіряються за набором експериментальних даних.

У даній моделі щільність клітин для проліферуючих, спокійних та некротичних позначимо як:

$$p(x, t), q(x, t), n(x, t)$$

де параметр t позначає час і параметр x це координата простору.

Некротичні клітини є мертвими і нерухомими, на відміну від інших, які не втратили здатність рухатися. Однак у тісно упакованих середовищах, наприклад сфероїд (чи це термін) пухлини, рух клітини може бути обмеженим. Це явище називається контактне гальмування міграції та воно подано у моделі з припущенням, що загальний потік життєздатних клітин визначається як $\frac{d}{dx}(p + q)$. Також, якщо ми припускаємо, що популяції клітин мають однакову рухливість, тоді умови руху проліферуючих та спокійних клітин буде задано як:

$$\frac{d}{dx} \left(\frac{p}{p+q} * \left(\frac{d(p+q)}{dx} \right) \right) \text{ та } \frac{d}{dx} \left(\frac{q}{p+q} * \left(\frac{d(p+q)}{dx} \right) \right)$$

Припустимо, що проліферуючі клітини ростуть із швидкістю, яка обмежена загальною популяцією пухлини, і те, що проліферуюча клітина може стати спокійною зі швидкістю, що буде залежати від концентрації $c(x,t)$ нутрієнтів. Також вважатимемо, що спокійні клітини відмирають з певною швидкістю, що залежить від $c(x,t)$.

Модель задається наступними диференціальними рівняннями:

$$\frac{dp}{dt} = \frac{d}{dx} \left(\frac{p}{p+q} * \left(\frac{d(p+q)}{dx} \right) \right) + g(c)p(1 - p - q - n) - f(c)p \quad (1)$$

$$\frac{dq}{dt} = \frac{d}{dx} \left(\frac{q}{p+q} * \left(\frac{d(p+q)}{dx} \right) \right) + f(c)p - h(c)q \quad (2)$$

$$\frac{dn}{dt} = h(c)q \quad (3)$$

У цих диференціальних рівняннях, параметр 1 репрезентує щільно упаковану популяцію клітин. Підкреслимо, що $g(0) = 1$, щоб створити відповідний початковий стан. Функції f та g є спадними функціями, що прямують до нуля, на відміну від c , яка прямує до $+\infty$, коли g – зростаюча функція. (як в книжці)

Оскільки швидкість клітин, що переходять у спокійний стан є вищою, ніж швидкість клітин, які відмирають, то вважаємо, що $f(c) > h(c)$. Що стосується концентрації поживних речовин, то ми використаємо форму, що була запропонована розробниками даної математичної моделі:

$$c = \frac{c_0}{\gamma + p} (1 - \alpha(p + q + n)) , \quad (4)$$

де α, γ – це безпросторові константи.

Систему диференціальних рівнянь (1) — (4) розв'язуємо чисельно, використовуючи метод скінченних різниць, на основі різниць для часу та центральної різниці для простору. Отримано:

$$p_i^{j+1} = p_i^j + \Delta t [u_i^j + g(c_i^j) p_i^j (1 - p_i^j - q_i^j - n_i^j) - f(c_i^j) p_i^j] \quad (5)$$

$$q_i^{j+1} = q_i^j + \Delta t [v_i^j + f(c_i^j) p_i^j - h(c_i^j) q_i^j] \quad (6)$$

$$n_i^{j+1} = n_i^j + \Delta t [h(c_i^j) q_i^j] \quad (7)$$

$$c_i^j = \frac{\gamma}{\gamma + p_i^j} [1 - \alpha (p_i^j + q_i^j + n_i^j)] \quad (8)$$

де

$$u_i^j = \frac{(p_{i+1}^j - p_{i-1}^j) r_i^j (r_{i+1}^j - r_{i-1}^j) + 4 p_i^j r_i^j (r_{i+1}^j - 2r_i^j + r_{i-1}^j) - p_i^j (r_{i+1}^j - r_{i-1}^j)^2}{4 (\Delta x)^2 (r_i^j)^2} \quad (9)$$

$$v_i^j = \frac{(q_{i+1}^j - q_{i-1}^j) r_i^j (r_{i+1}^j - r_{i-1}^j) + 4 q_i^j r_i^j (r_{i+1}^j - 2r_i^j + r_{i-1}^j) - q_i^j (r_{i+1}^j - r_{i-1}^j)^2}{4 (\Delta x)^2 (r_i^j)^2} \quad (10)$$

$$r_i^j = p_i^j + q_i^j \quad (11)$$

У наведених вище кінцево-різницевих рівняннях Δt і Δx відносяться до часових кроків і просторових інтервалів відповідно, а верхній індекс для залежних змінних вказує на часовий рівень та просторову позицію.

Для розв'язання цієї задачі був побудований програмний застосунок, за допомоги використання пакету прикладних програм для числового аналізу MATLAB, що дозволив розв'язати цю задачу та продемонструвати результати, які моделюють зміну розмірів ракової пухлини.

Список літератури:

1. Keng-Cheng Ang “Analysis of a Tumor Growth Model with MATLAB “, Electronic Proceedings of the 14th Asian Technology Conference in Mathematics(2009), Beijing, China, 17 – 21 December 2009

2. Sherratt, J. A. and Chaplain, M. A. J., “A new mathematical model for avascular tumour growth”, *Journal of Mathematical Biology*, 2001, 43, 291-312.
3. Adam, J. A., “A simplified mathematical model of tumour growth”, *Mathematical Bioscience*, 1986, 81, 224-229.
4. Ward, J. P. and King, J. R., “Mathematical modelling of avascular tumour growth”, *IMA Journal of Mathematics Applied in Medicine & Biology*, 1997, 14, 39-69

РОЗПІЗНАВАННЯ ЕМОЦІЙ У МОВЛЕНІ

Самотій Володимир

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

volodymyr.samotii@lnu.edu.ua

Вступ

Розпізнавання емоцій з мовних сигналів є актуальною та важливою проблемою в галузі мовленнєвої технології та комп'ютерного зору. Емоції виконують важливу роль в людському спілкуванні, відтворюючи багато аспектів наших соціальних взаємодій, таких як вираз обличчя, тон голосу, інтонація, акцентуація та ритм мовлення. Розпізнавання емоцій з мовних сигналів може мати великий потенціал застосування в ряді сфер [1], включаючи розпізнавання настрою користувачів, покращення взаємодії з голосовими помічниками, відтворення емоцій у віртуальних агентах, аналіз соціальних медіа, маркетинг, медичну діагностику та багато іншого.

Тема дослідження: Обробка аудіо-файлів , визначення емоцій мовця.

Мета дослідження: Розробка моделі, що розпізнає емоції людини, на базі штучних нейронних мереж.

Методи розпізнавання емоцій:

Є декілька методів розпізнавання емоцій такі як Метод глибинного навчання та Метод акустичного аналізу. Також можна їх комбінувати.

Запропонований алгоритм:

Запропонований алгоритм використовує сверткові шари (convolutional layers) для виявлення емоцій на вхідних зображеннях(такий як звукова спектрограма) та пов'язані шари (fully connected layers) для класифікації отриманих ознак на конкретні емоції.[2]

Набір даних, що використовувався:

Для дослідження використано набір даних Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), представлений на платформі для змагань з аналітики та передбачувального моделювання Kaggle. У наборі містяться загально 1440 файлів: 60 аудіо-файлів від 24-х акторів. Мовні емоції включають: спокій, радість, сум, злість, страх, здивування та відразу. Кожне вираження створюється на двох рівнях емоційної інтенсивності (нормальний, сильний), з додатковим нейтральним вираженням

Обробка даних:

Для обробки даних були проведені наступні кроки:

1. Застосувати методи фільтрації шуму, нормалізації амплітуди, аугментації даних тощо,
2. сигнали перетворити на числові вектори за допомогою мел-спектрального аналізу, або інших методів.
3. Данні розбито на тренувальні та тестові набори даних

Запропонована архітектура нейроної мережі:

Використано CNN архітектуру нейронну мережу[8], що складається з кількох шарів, кожен з яких виконує певну функцію обробки вхідних даних:

Convolutional layer: цей шар виконує операцію згортки між вхідними даними та набором ядер (filters), що дозволяє виловити різні функції ознак з вхідних даних.

Pooling layer: цей шар зменшує розмір вихідних даних шляхом взяття максимального значення (max pooling) або середнього значення (average pooling) в малих регіонах вхідних даних.

Activation layer: цей шар застосовує нелінійну функцію активації до вихідних даних з попередніх шарів, що дозволяє моделі навчитися нелінійним залежностям між ознаками.

Fully connected layer: цей шар об'єднує вихідні дані з попередніх шарів та застосовує лінійне перетворення до цих даних, що дозволяє моделі робити прогнози на основі ознак.

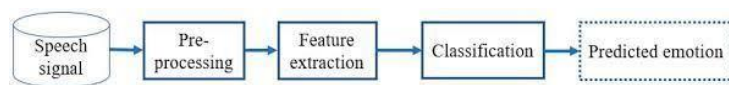


Рис. 1 Архітектура використаної CNN моделі для даної теми

Отриманні данні:

Нижче представлено графіки того, як змінюється loss-функція та Ассурасу на кожній епосі. Синьою лінією позначено таку зміну на тренувальних даних, оранжевою лінією на тестувальних

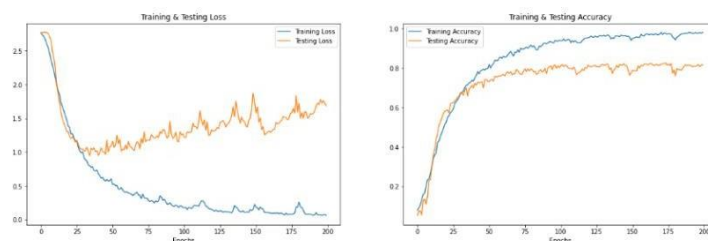


Рис. 3. Графіки зміни loss-функції та Ассурасу в процесі тренування мережі

Даний алгоритм побудови моделі демонструє хороші результати, проте має певні обмеження. Данна модель потребує велику кількість даних для навчання, щоб дати хороші результати. В наслідок того що потрібно велику кількість тренувальних даних виникає проблема великої затрати обчислювальних потужностей та часу.

Також дана модель є вразливою до шуму, наявність шуму чи інших форм спотворення може призвести до погіршення результатів.

Для програмної реалізації використовувалася мова програмування Python, бібліотеки librosa для роботи з аудіофайлами, pandas для зображення і передачі даних, tensorflow та keras для роботи з нейронними мережами.

Список літератури

1. Mencattini A, Martinelli E, Ringeval F, et al. Continuous Estimation of Emotions in Speech by Dynamic Cooperative Speaker Models[J]. IEEE Transactions on Affective Computing, 2016, PP (99):1-1.
2. Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press. p. 326.
3. Abdel-Hamid O , Mohamed A R , Jiang H , et al. Convolutional Neural Networks for Speech Recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 22(10):1533-1545.

ТРИАНГУЛЯЦІЯ ОБЛАСТЕЙ ДЛЯ МОДЕЛЮВАННЯ РОСТУ РАКОВИХ ПУХЛИН

Тетяна Муковоз

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

tetiana.mukovoz@lnu.edu.ua

Актуальність теми

Пухлинні захворювання супроводжують людство протягом усієї історії його існування. Проте, не зважаючи на те, що ця хвороба існує досить давно, в наш час універсальних ліків від неї все ще не існує, при чому статистика злоякісних захворювань постійно збільшується: щорічно реєструється близько 6 мільйонів нових випадків, також у 2019 років ракові утворення були визнані другою причиною смертності у світі, після серцево-судинних захворювань.

Тим не менш, науковці різних сфер все більше стараються дослідити цю хворобу, та, якщо не вилікувати, то хоча б полегшити її протікання, і уможливити повертання хворого до нормального життя.

Математичне та комп'ютерне моделювання ракових пухлин є необхідним для цього, адже не завжди можна дослідити потрібні властивості та наслідки хвороби в межах живого організму, тому важливо створювати штучні моделі.

Формулювання задачі

Одними з найефективніших моделей є ті, які базуються на розв'язуванні початково-крайових задач в частинних похідних. Складність цієї процедури полягає в тому, що досліджувана область є неправильної форми, вона не завжди є однозв'язною, та її границі змінюються в часі. Першим кроком для розв'язку такої проблеми, слід дискретизувати дану область, тобто побудувати сітку скінченних елементів.

Для побудови такої сітки можна використовувати різні бібліотеки, такі як: TetGen, GCal, Triangle, тощо. Тим не менш, не слід лише опиратися на готовий код доступних бібліотек, а й розуміти сутність алгоритму побудови такої сітки, адже в подальших дослідженнях потрібно буде адаптовувати сітку в часі, яка в кожний момент може змінюватись, через що границі будуть постійно модифікуватись.

Отже, в даній задачі базовим підходом служить алгоритм Делоне, який триангулює певну область по заданим точкам.

Триангуляція

Триангуляції - це процес поділ більшої області на більш дрібні елементи. Для

проведення триангуляції необхідні: набір точок, які взаємно не повторюються, та інформація про зв'язки між ними. Найбільш поширеним і вивченим механізмом є поділ на трикутні елементи.

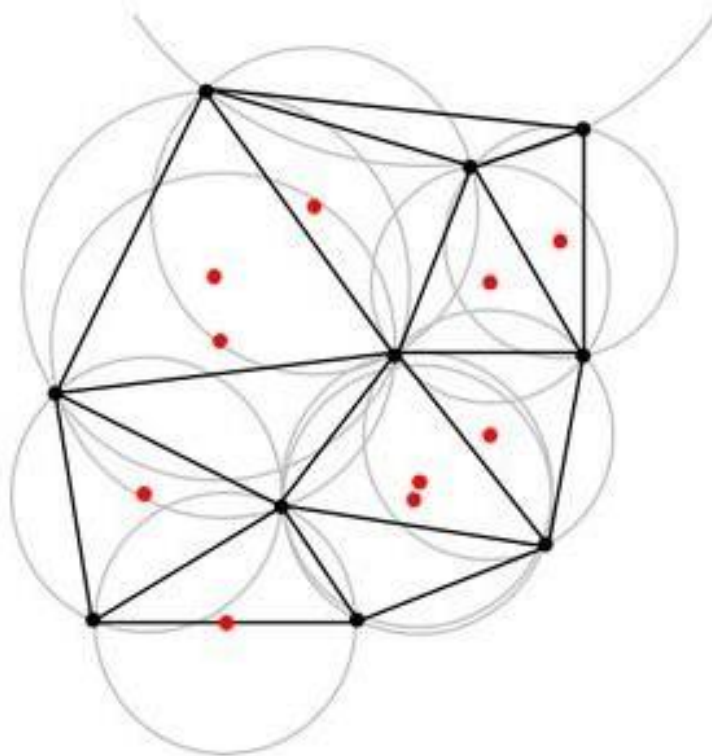
Можна виділити три основних правила триангуляції.

- Перше правило говорить про те, що трикутники не накладаються.
- Друге - виключає комбінації трикутників, де один має вершину всередині ребра іншого трикутника.
- Третє - виключає діри, точки заземлення (де лише два трикутники зустрічаються в одній точці), і непересічні набори трикутників

Метод Делоне

Тепер перейдемо до триангуляції Делоне - це спеціальна форма триангуляції, в якій усі точки в середині опуклої оболонки правильно вписані в трикутник, утворений сусідніми точками. Це означає, що немає точок, які були б більш близькі одна до одної, ніж до трьох сусідніх точок. При цьому розмір трикутників в триангуляції Делоне може бути досить різним, тому що вони формуються залежно від розташування точок.

Основна ідея алгоритму триангуляції Делоне полягає в тому, щоб побудувати трикутники, які задовольняють умову Делоне: жодна точка не повинна лежати в колі, яке описується трикутником і точками, що лежать на його межі.



[2]

Рис1. Побудова трикутних елементів алгоритму Делоне

Головна перевага алгоритму триангуляції Делоне полягає в тому, що він ефективний та досить простий у реалізації. Цей алгоритм також може бути

легко розширений для роботи з даними більших розмірностей.

У роботі побудована сітка скінченних елементів для області – кардіоїди, та досліджено її властивості для різних параметрів. У майбутньому слід розширити дослідження у напрямку побудови адаптивної сітки для задачі моделювання ракових пухлин.

Список літератури

7. Г.В. Бондарь, Ю.В. Думанський, «Онкологія», 2013, 7-11.
8. Вікіпедія - «Триангуляція Делоне»: <https://uk.wikipedia.org>
9. Triangulations: <https://www.math.utah.edu/~pa/MDS/triangulation.html>

CHOOSING THE RIGHT DATA ARCHITECTURE: A GUIDE TO COMPARING AND CONTRASTING DIFFERENT APPROACHES

Valerii Veseliak

Lviv Polytechnic National University

Institute of Computer Science and Information Technologies

valerii.v.veseliak@lpnu.ua

Introduction

Data architecture is a critical component of modern business and technology environments, enabling organizations to effectively manage, store, and analyze vast amounts of data. In the world of data management, data architecture is a key consideration for organizations of all sizes and industries. The right data architecture pattern can enable organizations to effectively manage and process vast amounts of data, while also providing the flexibility to meet changing business needs over time. However, choosing the right data architecture pattern can be a complex task, requiring careful consideration of various factors such as scalability, flexibility, and security.

Whether you are a data scientist, data engineer, or business leader, understanding data architecture patterns is essential for making informed decisions about managing and processing data. By the end of this article, readers will have a clear understanding of the different data architecture patterns available and be equipped with the knowledge and insights needed to select the most appropriate approach for their organization's unique data management needs.

Background and Context

As organizations collect and process more data, the complexity of data architecture decision-making increases. Also as data sources become more diverse and complex, organizations must also consider the implications of different data types and formats when designing their data architecture. Furthermore, compliance and security considerations must be taken into account when designing a data architecture. This is due to the behavior of both solution architects and data architects. Solution architecture tends to concern itself with functional, technology and software components of the Data architecture tends not to get involved with the data aspects of technology solutions, leaving a data architecture gap [3].

The general requirement for each architecture, which can be later extended or used

only partially, is similar to the basic retrieve, transform, and consume process. The process starts with data retrieval from various sources, then the data are processed and transformed, and finally, the result of the manipulation is returned [8]. Data architecture coherence broadly captures the technological architecture's capabilities around processing data streams through multiple layers and pipelines, combining data sets across various sources and locations, and developing and deploying machine learning models at scale [10].

Approaches to Data Architecture

There are several different approaches to data architecture, each with its own advantages and disadvantages. In this section, we will provide an overview of some of the most common approaches, including data warehousing, data lake architecture, and other approaches.

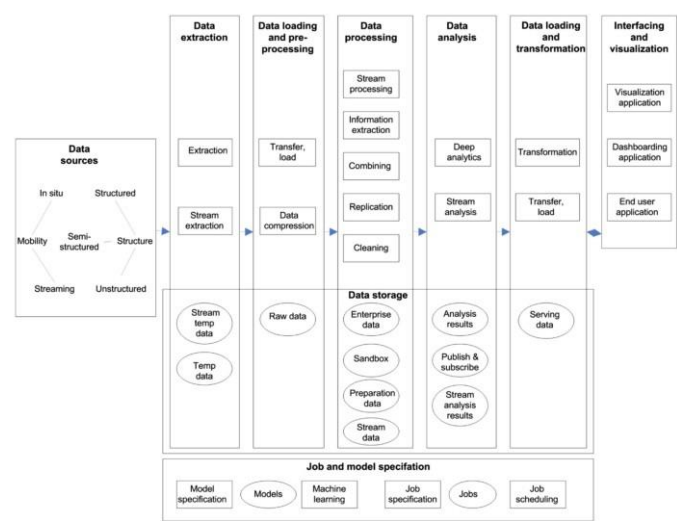


Image.1. Example of data architecture [11]

Data warehousing is a traditional approach to data architecture that involves creating a centralized repository of data from various sources. Data warehouses are designed to support complex queries and analysis, and are typically optimized for read-heavy workloads. Integration of traditional data warehouses and big data is a problematic scenario because in the previous era there were only structured data to deal with, but now there are other (big) data alongside them, which may bring more context to the analytics and improve it [2].

Data hub architecture is an approach that emphasizes creating a centralized hub for storing and managing data from various sources. The data is typically organized by subject area, and is designed to be easily accessed and shared by different teams and applications. Data hub architecture often uses a data warehouse as the central repository, but may also incorporate other data storage and processing technologies. Unlike data warehouses or data lakes, a data hub is significantly more agile to keep up with today's

fast-moving business cycles [7].

Data lake architecture is a newer approach to data architecture that emphasizes storing data in its raw, unstructured form. The basic idea of Data Lake is simple, all data emitted by the organization will be stored in a single data structure called Data Lake. Data will be stored in the lake in their original format. Complex preprocessing and transformation of loading data into data warehouses will be eliminated. The upfront costs of data ingestion can also be reduced. Once data is placed in the lake, it's available for analysis by everyone in the organization [1].

Lakehouse architecture is a newer approach that seeks to combine the benefits of data warehousing and data lake architecture into a single platform. The Lakehouse combines the best elements of the Data Lake and Data Warehouse, aggregating structured and unstructured data.. It is a new type of architecture for storing structured and unstructured data. Data may be stored in enormous quantities in one place and are immediately suitable for further analysis. At the same time, Lakehouse provides a data structure and provides data management features, such as those in Data warehouse, by running metadata layers at the top of the Lakehouse [4].

The combination of different data storage and integration techniques, however, without restriction to concrete architecture archetypes such as data lake or warehouse, led to the term Data Fabric, which was originally coined in 2015 by George Kurian of NetApp and then adopted and advocated again by Gartner [6]. Data fabric architecture is an approach that emphasizes creating a seamless and interconnected view of data across different systems and platforms. The goal of data fabric architecture is to provide a unified view of data, regardless of where it is stored or how it is processed.

Data mesh architecture is a newer approach that emphasizes decentralized ownership and governance of data. In data mesh architecture, data is treated as a product, and each domain or team is responsible for creating and managing their own data products. Data mesh architecture often uses a combination of distributed data storage, data processing, and data governance technologies, and is designed to support large, complex organizations with diverse data needs. A data mesh is a decentralized data architecture that organizes data by business domain - for example, marketing, sales, customer support, and so on - giving producers of a given dataset more ownership [5].

Best Practices for Choosing a Data Architecture

When choosing a data architecture, there are several best practices to keep in mind to ensure that your organization is making the best possible decision. First and foremost, it's important to carefully consider your business requirements and determine what type

of data architecture will best meet your needs. This includes evaluating your data sources, data volume, and data velocity, as well as any specific processing capabilities you require. For example, if you're dealing with large amounts of structured data and need to support complex analytics and reporting, a data warehouse might be the best option. On the other hand, if you're working with unstructured data and require more flexibility for ad-hoc queries, a data lake or lakehouse might be a better fit.

In addition to business requirements, it's also crucial to consider technical requirements when selecting a data architecture. This includes evaluating your organization's existing technology infrastructure, as well as any data processing tools you plan to use. It's important to choose a data architecture that is compatible with your existing systems and tools to minimize disruption and ensure smooth integration. Other technical considerations might include security and compliance requirements, resource constraints, and scalability needs. By carefully evaluating both business and technical requirements, organizations can make a well-informed decision and select a data architecture that will enable them to effectively manage and leverage their data for maximum value.

When comparing different data architecture patterns, it's important to take a systematic approach to ensure all relevant factors are considered. List of factors that have to be considered is usually Business requirements, Technical requirements, Data volume, Data velocity, Data variety, Data veracity, Data processing capabilities, Scalability, Performance, Resource requirements, Cost, Complexity, Security, Compliance, Availability, Reliability, Compatibility, Flexibility [9].

Conclusion

In conclusion, choosing the right data architecture approach is a critical decision for organizations seeking to effectively manage and analyze their data. By considering the advantages and disadvantages of different data architecture patterns, such as data warehouse, data lake, and hybrid approaches, organizations can select the most appropriate approach for their specific needs. It is important to note that the choice of data architecture is not a one-time decision, but rather a continuous process that requires ongoing evaluation and adaptation. Furthermore, effective data architecture decision-making requires collaboration between business stakeholders and technical experts. In summary, selecting the right data architecture approach is a complex decision that requires careful consideration of multiple factors.

References

1. P. Khine. Data lake: a new ideology in big data era / P. Khine, Z. Wang // ITM Web of Conferences. - 2018 - Vol.17
2. M. Ptiček. Big Data and New Data Warehousing Approaches / M. Ptiček, B. Vrdoljak // Proceedings of the 2017 International Conference on Cloud and Big Data Computing. - 2017 - pp. 6-10
3. A. McSweeney. Data architecture For Solutions. [Electronic resource]. – Available from:
https://www.researchgate.net/publication/366964200_Data_Architecture_For_Solutions
4. B. Čuš. Data Lakehouse: Benefits in small and medium enterprises / B. Čuš, D. Golec, I. Strugar // Journal of Innovative Business and Management. - 2023. - Vol.14 - pp.1-10
5. S. Panigrahy. From Data Mess to Data Mesh: Solution for Futuristic Self-Serve Platforms. / S. Panigrahy, B.Dash, R.Thatikonda // International Journal of Advanced Research in Computer and Communication Engineering. - 2023. - Vol.12 - No.4 - pp.677-683
6. T. Priebe. Finding Your Way Through the Jungle of Big Data Architectures. / T. Priebe, S. Neumaier, S. Markus. // IEEE International Conference on Big Data (Big Data) - 2021. - pp. 5994-5996.
7. K. Krupa. Data Hub Guide for Architects. / K.Krupa // Apress - 2022. - pp. 25-56
8. M.Bertko. A Comparative Analysis for Big Data Architectures / M.Bertko // Master thesis - 2019. - pp.15-41
9. A.Nambiar. An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management / A.Nambiar, D.Mundra // Big Data and Cognitive Computing - Vol.6 - No. 4 - 2022
10. R. Cao. Digital transformation, data architecture, and legacy systems / R. Cao, M. Iansiti // Journal of Digital Economy - Vol.1 - No.1 - 2022 - p.1-19
11. M. Macak. A Cross-Domain Comparative Study of Big Data Architectures / M. Macak, M. Ge, B. Buhnova // International Journal of Cooperative Information Systems - Vol.29 - No.4 - 2020 - pp.1-27.

КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ ЗА УМОВ МАЛОЇ КІЛЬКОСТІ ДАНИХ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ: РОЗГЛЯД К-КРАТНОЇ ФУНКЦІЇ ВТРАТ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Ядельський Артем, Баранов Микола

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

artem.yadelskyi@lnu.edu.ua

Вступ. Класифікація зображень у сучасному світі стає все частіше актуальною проблемою, яку намагаються вирішити за допомогою дуже великої кількості підходів. Особливо цікавою областю застосування нейронних мереж є класифікація зображень за умов малої кількості даних. Якщо врахувати такі обмеження, часто традиційні методи у тренуванні мереж не дають бажаних результатів, через специфіку підходів до їх навчання.

При розробці таких моделей було виявлено, що використання глибоких нейронних мереж (deep neural networks) [1] з застосуванням згорткових шарів (convolutional layers) показували кращі результати [2], ніж більш традиційні мережі.

Одним з методів навчання нейронних мереж для класифікації зображень є використання К-кратної функції втрат (K-tuplet loss) [3]. У ньому застосовуються згорткові шари для отримання ознак зображень, а тоді порівняння їх за допомогою К-кратної функції втрат, що розглядає кілька зображень одного, та різних класів для визначення помилковості мережі. Такий метод вже довів, що може продемонструвати суттєві результати [1].

Використання такої функції втрат надає нам параметричний (метричний) метод навчання нейронних мереж (metric learning) [1], що у цьому випадку базується на навчанні функції відстаней між об'єктами. Цю модель використовують при класифікації вхідних даних для створення альтернативної ре-

презентації ознак – так званих вкладень (embeddings). Вони надалі можуть бути розпізнані іншою нейронною мережею як скупчення (кластери) класів даних та відповідно однозначно класифіковані.

За допомогою моделі глибоких нейронних мереж з використанням згорткових шарів та застосування К-кратної функції втрат можна побудувати достатньо хорошу модель для класифікації зображень в умовах малої кількості даних [3, 4].

Роботі було ближче розглянуто метод використання К-кратної функції втрат для класифікації зображень за умов малої кількості даних з використанням глибоких нейронних мереж на основі згорткових шарів. Використаний метод не застосовував заздалегідь заготовлених (натренованих) моделей та додаткових даних для навчання.

Постановка задачі класифікації зображень. Розглядаючи нішу нейронних мереж, а саме мереж для класифікації зображень найчастіше мають на увазі моделі, що були натреновані на тисячах, іноді навіть сотнях тисяч зображень зібраних та оброблених різноманітними методами для максимізації результатів самого тренування. Такі мережі часто вимагають десятки годин процесорного часу, що часто робить повторення відповідних результатів над-складною задачею. Також популярним методом є використання вже наявних моделей як основи для нової нейронної мережі. При використанні готових моделей часто досягаються дуже високі результати, проте структура таких моделей зазвичай не дозволяє досягнути бажаних змін у методиці застосування мережі.

Натомість метою цієї роботи є розгляд задачі класифікації зображень за умов малої кількості даних. Зазначимо, що мала кількість даних символізує малу кількість зображень конкретного класу, а не розглянутого набору даних в цілому. Розглянута задача полягає в тому, щоб розробити нейронну мережу,

здатну класифікувати велику кількість різних класів зображень за умов, що кількість зображень конкретного класу обмежена.

Метод К-кратної функції втрат. Загальна ідея методу полягає у тому, щоб розглядати декілька зображень одночасно та порівнювати їх. Для цього обираються зображення за принципом: одне зображення, що називають якорним (anchor); одне зображення, що належить до того ж класу, що і якор-не, але відрізняється та називають позитивом (positive); К зображень, що не належать класу перших двох зображень та називають негативами (або нега-тивом у випадку $K = 1$) (negative) [5]. Надалі для зручності будемо називати їх відповідно А, Р, N.

Після цього розглядається порівняння цих зображень між собою. Це здійснюється шляхом розгляду обраних зображень у парах у n вимірному Евклідо-вому просторі та пошуком відстані між відповідними зразками. Тут відстань отриману в результаті мережі позначено як $f(X)$, де X відповідно зображення для якого шукали відстань:

$$d(A, P) - d(A, N) = \|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 \leq 0. \quad (1)$$

Базуючись на отриманих відстанях, можна визначити функцію втрат, що буде використана для тренування мережі. Бажаним є отримання максимізації відстані між якорем та негативом, а також мінімізації відстані між якорем та позитивом:

$$\lambda = \max(d(A, P) - d(A, N) + \alpha, 0).$$

Побудована модель мережі. Модель складається з п'яти шарів: вхідний шар з додатковими функціями розширення набору даних, три блоки згорткових шарів поєднаних нормалізацією та агрегацією максимізації, та вихідний повнозв'язний шар з L_2 нормалізацією.

Побудована нейронна мережа не є кінцевою метою у задачі класифікації зображень. Отриманим результатом є параметричні вкладення (metric embeddings) [1], які можна застосувати для інтерпретації їх у вигляді конкретного класу зображення розпізнаного мережею.

Для виконання процесу обробки параметричних вкладень використовують допоміжну нейронну мережу. Оскільки в нашому випадку результатом тренування початкової мережі є представлення даних у n вимірному Евклідовому просторі, що згруповані в кластери класів зображень, доцільним є використання методу К-найближчих сусідів [4].

Аналіз результатів. Отримано наступні результати при використанні побудованої моделі:

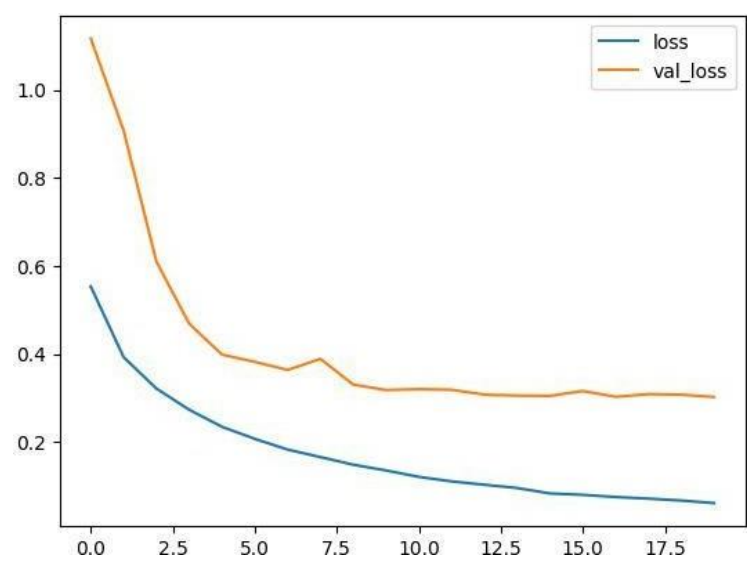


Рис. 1: Значення функції втрат для набору FSS-1000.

К-сть класів	Тренування	Перевірка	Вгадування	Покращення
10	100%	66,67%	10%	6,67
50	100%	18,00%	2%	9,00
200	100%	9,98%	0,5%	19,95

Табл. 1: Порівняння різної кількості класів для набору FSS-1000.

Висновки. Досліджено метод К-кратної функції втрат, спосіб його застосування для класифікації зображень в умовах малої кількості даних. Проведено навчання мережі з використанням обраних наборів даних, зібрано та проаналізовано практичні результати застосування створеної моделі.

Дослідження показало ефективність використання методу К-кратної функції втрат при малій кількості даних, що може дати швидкі результати у навчанні глибоких нейронних мереж на зображеннях які чітко можна віднести до різних типів об'єктів. Також можна зауважити, що цей метод є не настільки ефективним у навчанні на даних, що можуть мати багато спільних ознак.

Список джерел і літератури

- [1] Elad Hoffer, Nir Ailon. (2014). Deep metric learning using Triplet network. Дата доступу – 11.01.2023, посилання: <https://arxiv.org/abs/1412.6622>
- [2] Matthew D Zeiler, Rob Fergus. (2013). Visualizing and Understanding Convolutional Networks. Дата доступу – 10.01.2023, посилання: <https://arxiv.org/abs/1311.2901>
- [3] Xiaomeng Li, Lequan Yu, Chi-Wing Fu, Meng Fang, Pheng-Ann Heng. (2019). Revisiting Metric Learning for Few-Shot Image Classification. Дата доступу – 11.01.2023, посилання: <https://arxiv.org/abs/1907.03123>
- [4] Mukola Baranov. (2022). Gentle introduction to metric learning. Дата доступу – 20.01.2023, посилання: <https://www.kaggle.com/code/meowmeowmeowmeowmeow/gentle-introduction-to-metric-learning>
- [5] Olivier Moindrot. (2018). Triplet Loss and Online Triplet Mining in TensorFlow. Дата доступу – 20.01.2023, посилання: <https://omoindrot.github.io/triplet-loss>

REAL-TIME QUESTION GENERATION USING LARGE LANGUAGE MODELS & MACHINE LEARNING TECHNIQUES: AN EXPLORATION AND IMPLEMENTATION STUDY

¹Nazar Protsiv, ²Sofiia Shaposhnikova, ³Dmytro Omelian

¹Ivan Franko National University of Lviv (Faculty of Applied Mathematics and Informatics), Igor Sikorsky Kyiv Polytechnic Institute (²Institute for Applied System Analysis, ³Faculty of Informatics and Computer Science)

nazar.protsiv.pmp@lnu.edu.ua, shaposhnikova.sofia@ill.kpi.ua,
d.omelian-2025@kpi.ua

Abstract

The traditional lecture format has been a cornerstone of education for centuries, but it often falls short in engaging students. One of the main reasons for this is that lectures are typically one-way communication, with the teacher delivering information and the students passively receiving it. This can lead to boredom, disengagement, and a lack of material retention. To address these issues, many teachers have turned to active learning techniques, which require students to actively participate in the learning process. One such technique is questioning students during lectures, which can help to stimulate student engagement and deepen understanding of the material. However, formulation of relevant and challenging questions for a lecture can be a time-consuming and complicated task. To overcome this problem, we propose question generation using large language models (LLMs) for lectures in real time. LLMs are powerful machine learning models that have been pre-trained on large text corpora and can generate high-quality text. We can automatically generate questions based on the lecture content and the student's background knowledge.

Technical Solution

Our approach for real-time lecture question generation is a three-step pipeline that aims to enhance student engagement and comprehension during lectures. The first step is speech recognition, which transcribes the lecture into text for analysis. This transcription occurs over a fixed period of time, ensuring that the generated questions are timely and relevant. Next, we analyze the text using N-grams and topic modeling to determine the most relevant topics in the teacher’s speech.

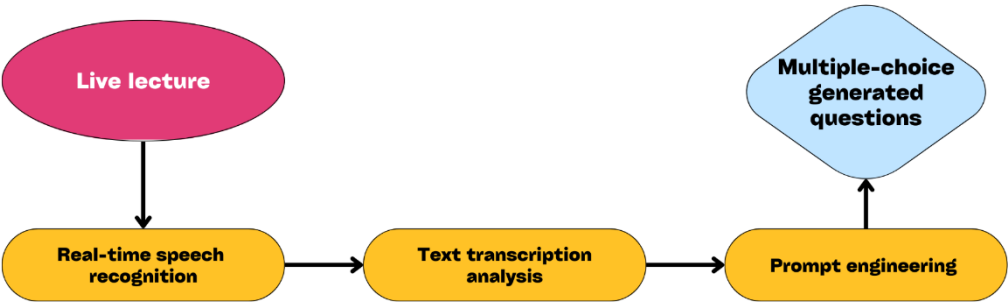


Image 1. Proposed pipeline

Text analysis techniques allow us to not just generate random questions, but first of all to identify

the main concepts discussed. Using keywords in prompts, we feed them into a pre-trained GPT model.

We propose two types of questions open and multiple choice. The teacher will receive automatic pop-ups with proposed questions and answers based on their speech. By offering the teacher real-time feedback on their lecture content, they can modify their delivery to ensure students are understanding the material.

Real-time speech recognition

In order to facilitate real-time question generation during lectures, we require an accurate and efficient method for retrieving lecture transcriptions. To achieve this, we utilize the pre-trained open-source model Citrinet from NVIDIA, which is an end-to-end convolutional Connectionist Temporal Classification (CTC) based automatic speech recognition (ASR) model. The Citrinet model uses 1D time-channel separable convolutions in conjunction with sub-word encoding and squeeze-and-excitation, resulting in a powerful deep residual neural network. To retrieve lecture transcriptions in real-time, we record 20-second chunks of audio and send requests to the Citrinet model, and concatenate the resulting transcriptions with previous ones. Using these transcriptions, we then employ another model to generate questions on demand or during predetermined time intervals (e.g., every 10-20 minutes).

Question generation

1) NGrams and Topic modeling

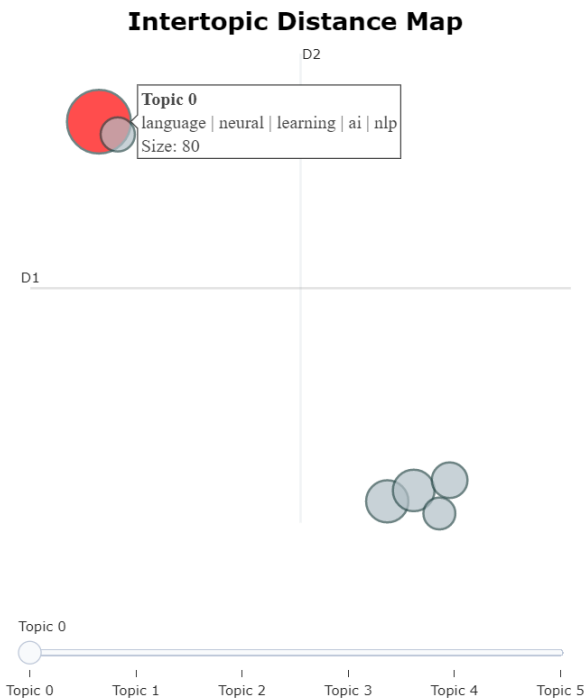


Image 2. Intertopic distance map

To further improve the lecture question generation process, we analyze retrieved lecture transcriptions. Firstly, we remove common words that do not carry significant meaning, known as stopwords, and tokenize words. Next, we construct bigrams and count their frequencies to identify the most common terms used in the lecture.

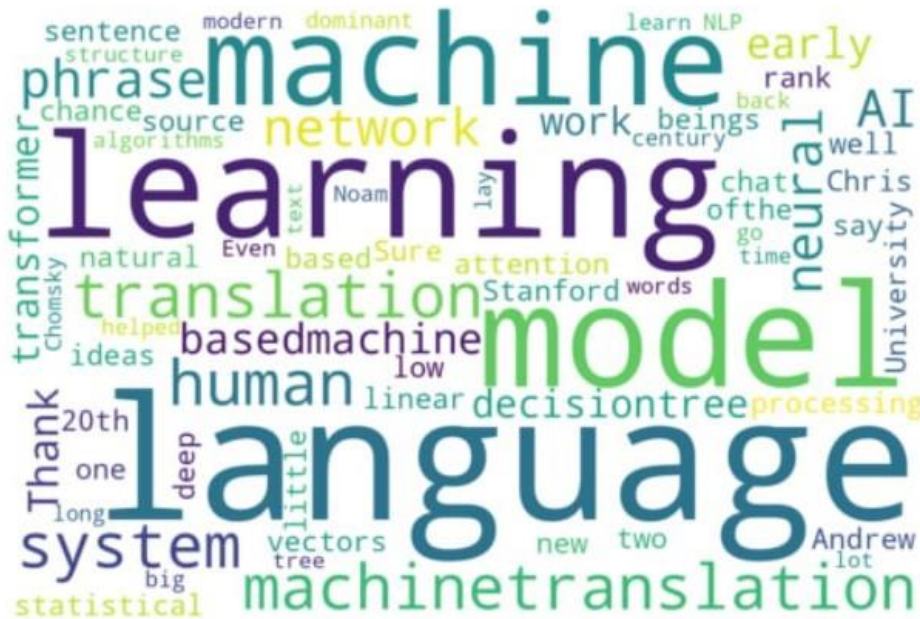


Image 3. Wordcloud of filtered bigrams

We then apply topic modeling using the BERTopic technique, which leverages transformers and c-TF-IDF to create dense clusters of sentences that form easily interpretable topics. This technique allows us to keep important words in the topic descriptions while still creating meaningful and coherent clusters. The resulting clusters can then be used to generate questions that are relevant to the lecture content.

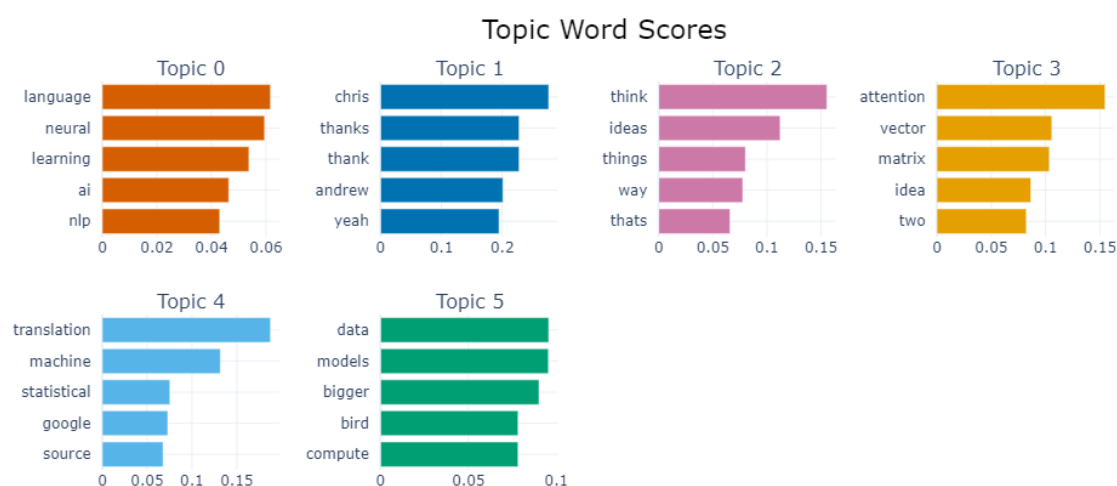


Image 4. Keywords probability distribution of retrieved topics

2) GPT generated questions

To generate lecture questions, we leveraged the frequencies of the bigrams, keywords of the topics, and probabilities of the sentences belonging to each topic obtained through the topic modeling process. Using this information, we experimented with different prompts and parameters of the GPT-3 (davinci) language model. The results were promising, as we received good, relevant questions about the lecture content. The generated questions can help to facilitate student engagement and comprehension, as they are tailored to the specific lecture content and aligned with the most important concepts and topics. The use of GPT-3 allows us to generate natural language questions that closely resemble those posed by human writers, thereby enhancing the overall learning experience for students.

Interactive app

To make our proposed real-time lecture question generation system easily accessible and user-friendly, we suggest creating an interactive app. One possible way of doing this is by developing a Zoom extension or Chrome extension for Google Meet, which are widely used platforms for online teaching. By integrating our system as an extension, instructors and students can easily access the generated questions during the lecture without the need to switch between different applications. This would provide a seamless and efficient user experience, which is crucial for ensuring the adoption and success of any new technology. Additionally, by using these popular platforms, we can reach a large user base, including both educators and learners, and provide them with a valuable tool to enhance their online learning experience.

1. What is the maximum number of layers that were used in the neural networks?

2. What is the library code used to create recurrent neural networks?

3. What was the main idea behind the attention-based model?

4. What was the result of using an attention-based model on machine translation?

5. Who developed the idea of an attention-based model?
1. What is the purpose of calculating attention back into the source sentence in the other language?

2. How does bi-linear attention work?

3. What is the difference between neural networks and knowledge graphs?

4. What is the difference between multiplicative attention and bi-linear attention?

5. What is the significance of the Cho and Dima paper?

Image 4. Generated open questions

- Q1. How do modern transformer models work?

A. They use a low-rank approximation to the matrix in the middle of the model.

B. They use a full rank matrix in the middle of the model.

C. They use two low-rank matrices to form a matrix and then take the dot product.

D. They use a giant matrix in the middle of the model.

Answer: A
- Q1: What is the traditional approach to word vector representation?

A1. Neural Networks B. Singular Value Decomposition C. Latent Semantic Analysis D. GPT-3

Answer: C. Latent Semantic Analysis
- Q2: What was the breakthrough paper in 2018 that enabled the use of large language models for NLP tasks?

A1. BERT B. GPT-3 C. ELMo D. Word2Vec

Answer: A. BERT
- Q3: What is the name of the task that the large language model was trained on?

A1. Text Classification B. Natural Language Inference C. Question Answering D. Word Prediction

Answer: D. Word Prediction

Image 5. Generated multiple-choice questions

Conclusions

Next steps are establishing filters and metrics to ensure the quality of the generated content is crucial. We can consider a variety of metrics, such as the relevance and accuracy of the generated questions, and their alignment with the lecture content and learning objectives. Additionally, to improve the quality of the generated questions, we can generate a larger number of samples and select the best ones based on the established metrics, and finetune used model for question generation using supervised learning approach.

Overall, our pipeline for real-time lecture question generation provides a novel approach to enhancing student engagement and comprehension. By leveraging advanced technologies such as speech recognition, N-grams, and pre-trained language models, we enable a seamless and efficient process for generating questions that align with the lecture's content.

References

1. Citrinet: Closing the Gap between Non-Autoregressive and Autoregressive End-to-End Models for Automatic Speech Recognition [Electronic resource]. – 2021. – URL:

<https://arxiv.org/abs/2104.01721>

2. BERTopic: Neural topic modeling with a class-based TF-IDF procedure [Electronic resource]. – 2022. – URL: <https://arxiv.org/abs/2203.05794>
3. Attention Is All You Need [Electronic resource]. – 2017. – URL: <https://arxiv.org/abs/1706.03762>
4. Language Models are Few-Shot Learners [Electronic resource]. – 2020. – URL: <https://arxiv.org/abs/2005.14165>
5. Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question Generation for Question Answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.
6. Sathish Reddy, Dinesh Raghu, Mitesh M. Khapra, and Sachindra Joshi. 2017. Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 376–385, Valencia, Spain. Association for Computational Linguistics.
7. Mulla N, Gharpure P. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Prog Artif Intell*. 2023;12(1):1–32. doi: 10.1007/s13748-023-00295-9. Epub 2023 Jan 30. PMCID: PMC9886210.
8. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.
9. Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International*

Зміст

GAMBLER'S RUIN PROBLEM AND ITS APPLICATION	4
МЕТОД ФУНДАМЕНТАЛЬНИХ РОЗВ'ЯЗКІВ ДЛЯ ЧИСЕЛЬНОГО РОЗВ'ЯЗУВАННЯ ТРИВИМІРНОЇ МІШАНОЇ ЗАДАЧІ ДЛЯ РІВНЯННЯ КЛЕЙНА- ГОРДОНА.....	7
РОЗПІЗНАВАННЯ ТЕКСТУ З КАРТИНКИ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ	11
STREAMLINING GRAPHQL WEB SERVER DEVELOPMENT WITH A SCHEMA- DRIVEN APPROACH	14
МОДЕЛЬ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТУ З АУДІО- ФАЙЛІВ.....	16
УДОСКОНАЛЕННЯ РОБОТОТЕХНІКИ ЗА ДОПОМОГОЮ МІКРОСЕРВІСІВ І МЕРЕЖЕВИХ ПРОТОКОЛІВ	20
ОПТИМІЗАЦІЯ РЕЖИМІВ РУХУ ДВОЛАНКОВОГО МАНІПУЛЯТОРА З АКТИВНИМИ І ПАСИВНИМИ ПРИВОДАМИ	22
РОЗПІЗНАВАННЯ РУКОПИСНОГО ТЕКСТУ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ	26
ПРО АЛЬТЕРНУЮЧИЙ МЕТОД ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОШІ ДЛЯ РІВНЯННЯ ЛАПЛАСА	30
РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ВИЯВЛЕННЯ МЕДИЧНОЇ МАСКИ НА ОБЛИЧЧІ ЛЮДИНИ МЕТОДАМИ КОМП'ЮТЕРНОГО ЗОРУ	34
РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ КЛАСИФІКАЦІЇ ВОЄННИХ РУЙНУВАНЬ	37
ДВОКРОКОВИЙ МЕТОД ДЛЯ НЕСТАЦІОНАРНИХ ЗАДАЧ У ТОРОЇДАЛЬНИХ ОБЛАСТЯХ.....	41
ЗАСТОСУВАННЯ МЕТОДУ КОЛОКАЦІЇ ДЛЯ РОЗВ'ЯЗУВАННЯ ІНТЕГРАЛЬНОГО РІВНЯННЯ В ДВОВИМІРНІЙ ОБЛАСТІ.....	47
НАСКРІЗНЕ ШИФРУВАННЯ В ПРОВІДНИХ КЛАУД-МЕСЕНДЖЕРАХ.....	51
ЗАСТОСУВАННЯ FLUENT API ПРИ СТВОРЕННІ БІБЛІОТЕК	55
ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ	59
РЕАЛІЗАЦІЯ ФІЗИКО-ІНФОРМОВАНОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗВ'ЯЗУВАННЯ РІВНЯНЬ У ЧАСТКОВИХ ПОХІДНИХ	64
ОГЛЯД МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ СНАТGPT ДЛЯ РОЗВ'ЯЗУВАННЯ АЛГОРИТМІЧНИХ ЗАДАЧ.....	68
ОПТИМІЗАЦІЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ЧЕРЕЗ РЕАЛІЗАЦІЮ ІДЕЇ РАНЬОГО ВИХОДУ.....	71
ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ РІЗНИХ СКІНЧЕННИХ ЕЛЕМЕНТІВ ПРИ МОДЕЛЮВАННІ СЕРЕДОВИЩА МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ.....	76
BREAST CANCER DETECTION USING MACHINE LEARNING.....	78
РОЗРОБКА АРХІТЕКТУРИ RDE (RESIDUAL DOMAIN EXPERTS) ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ У ВИПАДКУ МАЛОЇ КІЛЬКОСТІ ВХІДНИХ ДАНИХ	

.....	82
СИСТЕМА АНАЛІЗУ ДАНИХ ДЛЯ КУРСУ «МОДЕЛІ СТАТИСТИЧНОГО НАВЧАННЯ» МОВОЮ PYTHON.....	86
ОЦІНКА ЯКОСТІ СТЕГANOГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ В ЗОБРАЖЕННЯХ	91
РОЗВ'ЯЗУВАННЯ ОДНОВИМІРНИХ КРАЙОВИХ ЗАДАЧ МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ.....	96
ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО ПРОГРАМУВАННЯ ДЛЯ ЧИСЕЛЬНОГО РОЗВ'ЯЗУВАННЯ ТРИВИМІРНОЇ ЗАДАЧІ РЕКОНСТРУКЦІЇ ГРАНИЦІ	100
РОЗРОБКА СИСТЕМИ ДЛЯ ВИДОБУВАННЯ ІНФОРМАЦІЇ З ТЕКСТУ	104
РОЗРОБКА РУРІ-БІБЛІОТЕКИ ДЛЯ ФРАГМЕНТОВАНОЇ ПЕРЕДАЧІ ФАЙЛІВ (З ПІДТРИМКОЮ ФРЕЙМВОРКУ DJANGO).....	109
МОДЕЛЮВАННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ПРЕДМЕТІВ ОДЯГУ З ВИКОРИСТАННЯМ ШАБЛОНІВ ПРОЄКТУВАННЯ.....	113
ADVERSARIAL ATTACKS ON OBJECT DETECTION SYSTEMS	118
STEGANALYSIS USING DEEP NEURAL NETWORKS	122
SYNTHETIC DATA GENERATION FOR NEURAL NETWORK ANALYSIS OF WAVE PROPAGATION IN DISSIPATIVE FLUIDS	127
УСЕРЕДНЕННЯ В ОДНОЧАСТОТНІЙ СИСТЕМІ ІЗ КРАЙОВИМИ УМОВАМИ....	131
ЗАСТОСУВАННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ АНАЛІЗУ РИНКУ КРИПТОВАЛЮТ	134
АНАЛІЗ ВИКОРИСТАННЯ МЕТОДІВ РЕАКТИВНОГО ПРОГРАМУВАННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ВИСОКОЇ ДОСТУПНОСТІ ТА НАДІЙНОСТІ МІКРОСЕРВІСНИХ СИСТЕМ НА ОСНОВІ JAVA.....	138
АНАЛІЗ ПРОЦЕСУ РОЗРОБКИ ВЕБСАЙТУ НА ОСНОВІ КОРИСТУВАЦЬКОГО ДОСВІДУ	141
МОДЕЛЮВАННЯ ЗМІНИ РОЗМІРІВ РАКОВИХ КЛІТИН.....	147
РОЗПІЗНАВАННЯ ЕМОЦІЙ У МОВЛЕНІ.....	151
ТРИАНГУЛЯЦІЯ ОБЛАСТЕЙ ДЛЯ МОДЕЛЮВАННЯ РОСТУ РАКОВИХ ПУХЛИН	154
CHOOSING THE RIGHT DATA ARCHITECTURE: A GUIDE TO COMPARING AND CONTRASTING DIFFERENT APPROACHES	157
КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ ЗА УМОВ МАЛОЇ КІЛЬКОСТІ ДАНИХ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ: РОЗГЛЯД К-КРАТНОЇ ФУНКЦІЇ ВТРАТ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ.....	162
REAL-TIME QUESTION GENERATION USING LARGE LANGUAGE MODELS & MACHINE LEARNING TECHNIQUES: AN EXPLORATION AND IMPLEMENTATION STUDY	167