Міністерство освіти і науки України
Львівський національний університет ім. Івана Франка
Факультет прикладної математики та інформатики

МІЖНАРОДНА СТУДЕНТСЬКА НАУКОВА КОНФЕРЕНЦІЯ З ПИТАНЬ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА КОМП'ЮТЕРНИХ НАУК
МСНКПМК-2021

Львів 2021

Вступ:

Мета конференції: Сприяти співпраці студентів, що вивчають споріднені області прикладної математики та комп'ютерних наук урізних установах.

Тема конференції: З темами що були досліджені студентами можна ознайомитись в змісті.

Організаційний комітет:

Дияк Іван – декан факультету прикладної математики та інформатики.

Горлач Віталій – заступник декана факультету прикладної математики та інформатики.

Мельничин Андрій – заступник декана факультету прикладної математики та інформатики.

Лавришин Ірина – голова студентської ради факультету прикладної математики та інформатики.

Онищенко Орест – студент факультету прикладної математики та інформатики.

Яриш Віра-Оксана – студентка факультету прикладної математики та інформатики.

Шевченко Ірина– студентка факультету прикладної математики та інформатики.

Лонюк Олеся – студентка факультету прикладної математики та інформатики.

Москаленко Анна– студентка факультету прикладної математики та інформатики.

Березяк Марта– студентка факультету прикладної математики та інформатики.

Вовк Любов-Василина– студентка факультету прикладної математики та інформатики.

Шевчук Мар'ян-Северин – студент факультету прикладної математики та інформатики.

Та інші члени студентського самоврядування факультету прикладної математики та інформатики.

Зміст

Preprocessing of the Raw Well Drilling Data 4

Визначення точних розмірів предметів одягу з використанням OpenCV 8

Numerical Modeling of Acoustic Wave Propagation in Thermohydroelastic Systems 11

Створення моделі криптовалюти на основі технології блокчейн 16

Програмно-алгоритмічне забезпечення розпаралелення навчання штучних нейронних мереж з використанням технологій CUDA 21

Задача ідентифікації та оптимізації для моделі взаємодії популяцій 26

Розробка веб-аплікації для створення та керування розкладом занять в університеті 30

Named Entity Extraction of Unstructured text 34

Паралельні методи розв’язування систем нелінійних алгебраїчних рівнянь 37

Розв'язування систем нелінійних рівнянь із застосуванням матричних ланцюгових дробів 40

Estimation of Fatality of a classical SIR model: A case study of COVID-19 data in Lviv 45

Solving vehicle routing problem using ant colony optimization 50

Two dimension advection-diffusion model: 52

Трикроковий комбінований метод хорд та Ньютона для розв’язування систем з недиференційованим оператором 56

Порівняльна характеристика однокрокових та двокрокових методів хорд та Курчатова розв'язування нелінійних систем рівнянь 60

Реалізація інструментів симуляції неевклідового простору в прикладних програмних додатках 70

Створення додатку для аналізу сну з використанням технологій Big Data 75

Регуляризація формули обернення перетворення Лапласа та її адаптація до розв’язування динамічних задач теорії пружності 78

Інформаційна технологія розпізнавання геометричних об’єктів у cae-системах 80

Розробка соціальної аплікації з функціоналом у реальному часі 83

Побудова h-адаптивної схеми МСЕ на основі НМГЕ- МСЕ оцінювача похибки 87

Чисельне розв'язування задачі Діріхле для узагальненого рівняння Лапласа за допомогою методу граничних інтегральних рівнянь 93

Використання мурашиних алгоритмів в алгоритмах оптимізації. Розв’язування задачі комівояжера за допомогою мурашиних алгоритмів 98

Створення Telegram бота-помічника з інтерполяції функцій однієї змінної 106

Точні розв’язки однієї нелінійної динамічної системи 114

Візуалізація міграційних процесів до країн єс за допомогою хордових діаграм (з використанням javascript-бібліотеки D3.Js) 123

Використання super resolution у системі оптичного розпізнавання тексту для зображень низької якості 132

Розробка прототипу системи оптичного розпізнавання тексту для зображень низької якості 136

Реалізація чат бот для відстеження погоди, використовуючи мову програмування Java 140

Діагностування захворювань легеневої системи таких як: COVID-19 та SARS-COV-2 використовуючи Computer vision та Convolutional Neural Network 145
Квадратична функція регресії з марковськими переключеннями 148
Математична модель росту ракової пухлини з запізненням 153
Програмна реалізація ненавантаженої лінійної багатошарової моделі ґрунту за допомогою методу скінчених елементів 157
Development of video chat website using WEBRTC 161
Розробка системи для онлайн замовлень їжі з використанням REACT.JS ta FLASK 164
Навігація на мапі отриманій з супутникових зображень за допомогою нейронної мережі 167
OpenGL на прикладі кубика-Рубіка 170
Визначення мотивації кандидатів працювати у компанії на основі аналізу їх характеристик із використанням машинного навчання 173
Проектування та розробка прототипу пристрою 177
Автоматизованого сортування сміття 177
Проектування та реалізація ігрової програми “ROKKILIKE” 182

Preprocessing of the Raw Well Drilling Data

Ihor Huralskyy
Ph.D. Oleksandr Vovk
Ukraine, Lviv
Ivan Franko National University of Lviv
Faculty of applied mathematics and informatics
uniqueihor@gmail.com

Whereas oil wells drilling is a very complex, laborious and expensive process, a unique range of drilling service providers and contractors has emerged in the last century to provide well services to companies that own wells. These drilling companies have multimillion-dollar budgets for research, improving and development of new drilling rigs and methods. Throughout the history of oil well drilling, companies have sought to increase productivity and reduce overall drilling costs. Many new devices have been created for this purpose, such as: rotary controlled systems, wire drill pipes, hybrid drills [1]. However, today companies are beginning to use powerful data science tools. Combining the growth of computing power, technological advances in automation, the evolution of technology and the strong incentive to reduce costs due to falling oil prices in recent years, companies want to use data about the operations they perform on a daily basis to optimize these operations in the future. Today, the drilling rig contains a large number of devices that collect parameters from almost every equipment with a very high frequency, for example: the temperature of electric motors, the speed of rotation of the drill string, the distance from the rig, and so on.

Companies have seen great potential in efficiently processing huge amounts of their data after implementation of data science in their workflows. However, the raw data from the sensors are generally not suitable for the application of data analysis models due to the high noise level and other distortions of the data due to physical processes in the well. Therefore, the purpose of this work is to study and develop an algorithm for organizing, cleaning and correcting raw data obtained during drilling, for their further use on machine learning models and data analysis platforms. The main focus of work is on cleaning and correcting such parameters as WOB (Weight on Bit) by recalibrating string weight(SW) values.

Weight on Bit (WOB) is a parameter obtained from surface sensors during drilling which is a measure of the amount of force(klbs) directed downward to the bit. It is important to have accurate values for this parameter because it directly affects the rate of penetration (ROP) and can prevent accelerated destruction of the bit and pump

motor failure. WOB value is calculated using the next formula: WOB = SW – HL. The Hook Load (HL) parameter is the amount of force that pulls the drill string up. SW is the total weight of the drill string when it does not rest on the bottom of the wellbore[2]. The difference between these two parameters, is the value of the force directed downward to the bit.

The main cause of errors in this data is incorrect, irregular or skipped zeroing [2]. Zeroing is a process of recalibrating of SW value for new stands that have been added to the drill string.These errors reduce the reliability of the conclusions obtained during the analysis of the data and complicate drilling optimization algorithms that use WOB as an input data. Therefore, in this work an algorithm was implemented to find the optimal points for zeroing. The optimal points for zeroing will be denoted as ZWOB.

Figure 1 shows a high-level block diagram for the developed algorithm. The developed algorithm has four main requirements that must be met at the point in order to consider it as ZWOB:

• Start of the stand;

• RPM greater than 0;

• Block height begins to decrease;

• The pumps are switched on and have a local maximum value.

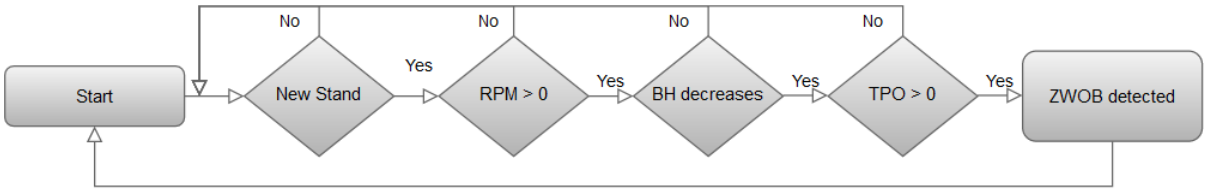


Figure 1. High-level block diagram of the algorithm.

Start of the stand is required to ensure that ZWOB points will be at the same block height for each stand to make measurements between different stands consistent and ensure that drillbit experiences similar forces on each stand at ZWOB points [4]. To identify new stands the algorithm was developed to search for points after huge drops(~70%) in HL and with block height(BH) above threshold value which is used as input data for the algorithm(or BH>average BH value).

After identifying all stands algorithms takes each interval(from start of the stand to start of the next stand) and searches for a point that satisfies three other requirements. “RPM greater than 0” and “The pumps are switched on and have a local maximum value“ requirements are needed to ensure that the drill string and mud system, respectively, experiencing the same forces as during drilling.” Block height begins to

decrease" is used to find the exact point when the drilling process begins. Figures 2 and 3 show the optimal location of the ZWOB in terms of block height, HL, RPM and total pump output (TPO) and a comparison between the raw and cleaned WOB data, respectively.

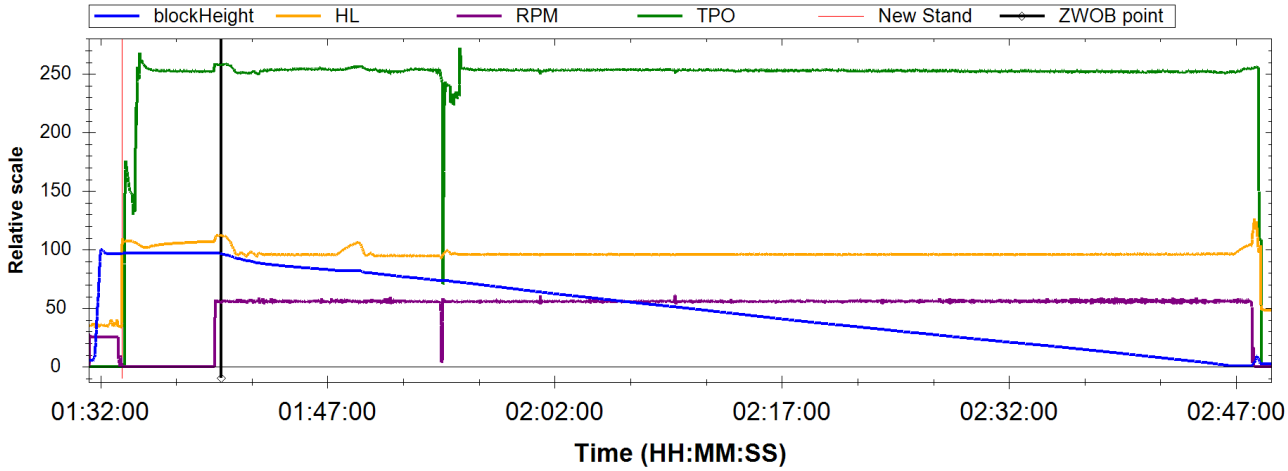


Figure 2. Optimal location of ZWOB point on a single stand.

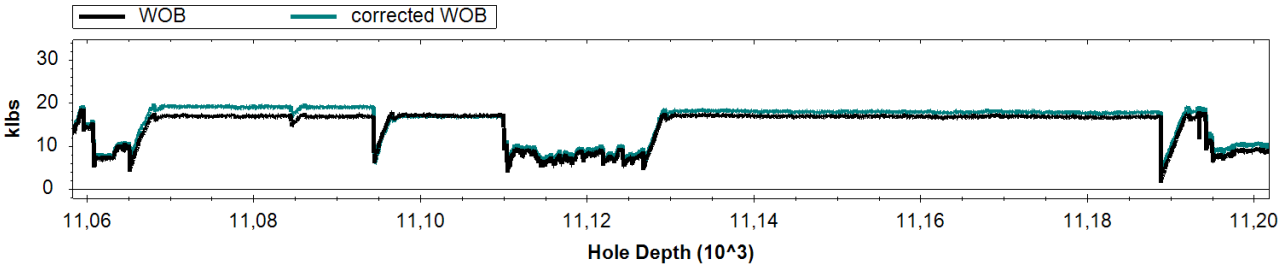


Figure 3. Comparison of raw and cleaned WOB data on a part of well.

Conclusion

In this work, the algorithm that automatically finds optimal ZWOB points was described and developed to correct raw WOB data. The algorithm uses historical data as input but it also can be modified to be used on real-time data during well drilling in future. Thanks to TerraLab [5] for providing us with real drilling data during the algorithm development and its approbation.

References

[1] Braga, Daniel Cardoso, (2019). "Field Drilling Data Cleaning and Preparation for Data Analytics Applications".

[2] Neufeldt, A. C., Lai, S. W., and Kristjansson, S. D. (2018). An Algorithm to Automatically Zero Weight on Bit and Differential Pressure and ResultingImprovements in Data Quality.

[3] Yang Zha, Stacey Ramsay, Son Pham (2018). Real Time Surface Data Driven WOB Estimation and Control.

[4] Cayeux, E. ,Skadsem, H. J. , Kluge, R. (2015).Accuracy and Correction of Hook Load Measurements During Drilling Operations.

[5] https://terra-laboratories.com/.

Визначення точних розмірів предметів одягу з використанням OpenCV

Яропуд С.Р.

Львів, Україна

ЛНУ ім. Івана Франка

Факультет прикладної математики та інформатики

Sofiia.Yaropud@lnu.edu.ua

Задача вимірювання реальних розмірів об’єкта з фотографії з допоміжним елементом насправді досить актуальна, адже, дуже зручно, в один клік, отримати реальні розміри потрібної вам речі.

Для цієї задачі була використана бібліотека OpenCV, яка реалізовує алгоритми комп'ютерного зору та обробки зображень з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту, тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях[1].

У роботі запропоновано застосувати алгоритми бібліотеки OpenCV для розпізнавання характерних реальних розмірів предметів одягу. Практичне дослідження задачі розпізнавання здійснено на прикладі штанів.

При розгляді цієї проблеми слід врахувати кілька аспектів:

1. Проблеми подачі вхідної інформації.
2. Опис основного алгоритму знаходження характерних розмірів.
3. Врахування особливостей фасону та розташування окремих елементів речі.
4. Розпізнавання еталонного елемента.
5. Врахування особливостей бібліотеки OpenCV та виявлення проблемних моментів.
6. Визначення характерних розмірів речі за розміром еталону.

Щодо першого пункту вважатимемо, що зображення вже є очищеним від шумів і до нього застосована процедура контрастування, що є окремою задачею.

Щодо другого пункту визначимо характерні розміри, якими для штанів є довжина, ширина в талії, ширина бедер, ширина штанини в нижній частині.

Для зображення застосуємо процедуру визначення зовнішніх контурів об’єкта, використавши одну з найважливіших функцій бібліотеки OpenCV: findContours(). В основі цієї функції є алгоритм Судзукі – Касамі, що базується на токенах для досягнення взаємного виключення в розподілених системах.

Визначивши точки зовнішнього контуру, та знайшовши характерні розміри обрамлюючого прямокутника, а також врахувавши особливість області, можна знайти довжину, об’єм бедер та ширину талії. При цьому слід врахувати, що при різних фасонах штанів можуть бути різні точки замірів. Для визначення глибини посадки, а також ширини штанин, слід виділити точку їх розходження. Вона може бути дещо некоректно визначена, якщо штани на фотографії неправильно розкладені.

При цьому зазначимо, що всі розміри будуть визначатися в пікселях.

Для визначення реальних розмірів в сантиметрах спочатку слід розпізнати еталонний елемент на зображенні, що можна теж зробити, використовуючи findContours(). Після цього слід врахувати, що еталонний елемент може бути розташований під довільним кутом. Знайшовши характерні розміри обрамлюючого прямокутника та здійснивши поворот еталонного зразка, знаючи його реальні розміри, можна знайти кількість пікселів в одному сантиметрі довжини та ширини. Це дасть змогу визначити реальні розміри об’єкта дослідження.

Запропонований алгоритм реалізований на мові Python. З допомогою створеного програмного забезпечення були проведені обчислювальні експерименти для різних типів об’єктів. У загальному, такий підхід показав хороші результати.

На рисунку 1 зображено результат виконання задачі для дослідного зразка, отриманий за допомогою розробленої програми.

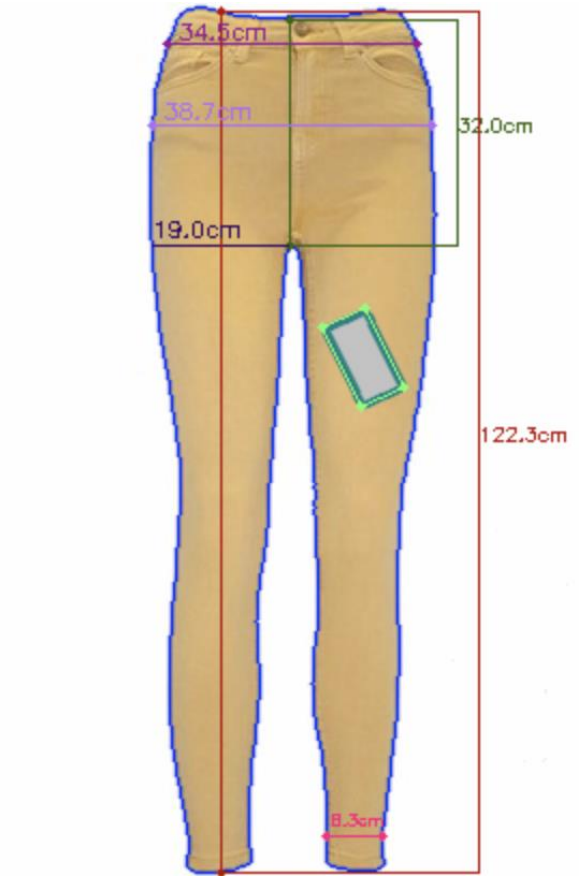


Рис.1 Результат виконання програми для дослідного елемента.

Однак виявлено деякі особливості, які вимагають ще окремого дослідження. Зокрема, проблеми виникають у випадку, якщо штани є з неоднорідним забарвленням, а також мають елементи кольору фону: наприклад, ґудзики, нашивки, елементи швів. У цьому випадку алгоритм findContours() трактує такі елементи як фон, а тому знаходження контурів не завжди є коректним, або є не легкою проблемою.

Список використаних джерел:

1. https://uk.wikipedia.org/wiki/OpenCV
2. Adrian Rosebrock. Practical Python and OpenCV [Електронний ресурс] – Режим доступу до ресурсу: https://minhtn1.github.io/Practical%20Python%20and%20OpenCV,%203rd%20Edition.pdf
3. Mark Lutz. Learning Python, Fifth Edition. - O'Reilly Media. - 2013. – 1594 p.

Numerical Modeling of Acoustic Wave Propagation in Thermohydroelastic Systems

Volodymyr Milchanovskyi

Lviv, Ukraine

Ivan Franko National University of Lviv

Faculty of Applied Mathematics and Informatics

volodymyr.milchanovskyi@lnu.edu.ua

Introduction

The problems of dissipative acoustics have many applications in engineering practice, for example measuring or medical equipment. This paper presents mathematical models and numerical results of for time and frequency domain problems in terms of unknown displacements and temperatures in water and fuel oil.

Initial-boundary problem

Let fluid occupy limited area Ω of points $x = \{x_j\}_{j=1}^{d} \in R^d, d = 1, 2\ or\ 3$, and let area Ω has Lipschitz-continuous boundary Γ. Introduce piecewise defined displacement vector $\boldsymbol{u} = \{u_j(x,t)\}_{j=1}^{d}$ and for all $(x,t) \in \Omega \times [0, \mathrm{T}]$. Similarly, we describe the unknown temperature $\boldsymbol{\theta} = \theta(x,t)$, intensity of the distributed sound $f(x,t)$ and heat sources $w = w(x,t)$. Also, we have following physical values:

ρ – density; a_{ijkm} – elasticity; c_{ijkm} – viscosity;

c_v – specific heat; Θ_0 – initial temperature;

χ_{ij} – thermal conductivity; α – thermal expansion.

Then the interaction of acoustic and thermal fields in the system considered fluid is described by the following initial-boundary value problem of thermohydroelasticity in terms of elastic displacements and temperature.

$$\begin{cases} Find\ displacements\ vector\ \boldsymbol{u} = \{u_j(x,t)\}_{j=1}^{d}\ and \\ temperature\ \boldsymbol{\theta} = \theta(x,t)\ that\ satisfy\ equations \\ \rho u_j'' - \sigma_{ij,j}(u,\theta) = f_i \\ c_v\theta' - \left(\chi_{ij}\theta_{,j}\right)_{,i} + \theta_0\alpha_{ij}c_{ijkm}e_{km}(u') = w \\ \sigma_{ij}(u,\theta) := a_{ijkm}[e_{km}(u) - \alpha_{km}\theta] + c_{ijlm}e_{km}(u') \\ e_{ij}(u) := \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)\ in\ \Omega \times (0,T] \end{cases}$$

with boundary conditions

$$\begin{cases} u = 0 \; on \; \Gamma_u \times [0,T], \Gamma_\mathrm{u} \subset \Gamma \\ \sigma_{ij}(u,\theta)n_j = \widehat{\sigma_\iota} \; on \; \Gamma_\sigma \times [0,T], \Gamma_\sigma = \Gamma \backslash \Gamma_u \\ \theta = 0 \; on \; \Gamma_\theta \times [0,T], \Gamma_\theta \subset \Gamma \\ -n_i \chi_{ij} \theta_{,j} = \hat{q} \; on \; \Gamma_q \times [0,T], \Gamma_q = \Gamma \backslash \Gamma_\theta \end{cases}$$

and initial conditions

$$\begin{cases} u|_{t=0} = u_0 \\ u'|_{t=0} = v_0 \;\; in \; \Omega \\ \theta|_{t=0} = \theta_0 \end{cases}$$

where $n = \{n_i\}_{i=1}^d, n_i := \cos(n, x_i)$ is unit vector of the outer normal to the boundary Γ, $v_{,i} := \frac{dv}{dx_i}, v' := \frac{dv}{dt}$.

Variational formulation

We formulate the corresponding variational problem of thermohydroelasticity.

$$\begin{cases} Let \; u_0 \in V := \{v \in [H^1(\Omega)]^d : v = 0 \; on \; \Gamma_u\}, v_0 \in H := [L^2(\Omega)]^d, \\ \theta_0 \in \; G := \; L^2(\Omega), \qquad (l,z) \in L^2(0,T; V \times G) \\ find \; pair \; (u,\theta) \in L^2(0,T; V \times G) \; that \\ m(u''(t),v) + c(u'(t),v) + a(u(t),v) - b(\theta(t),v) = \langle l(t), v \rangle \\ s(\theta'(t),g) + k(\theta(t),g) + b\big(g, u'(t)\big) = \langle z(t), g \rangle \; \forall t \in (0,T] \end{cases}$$

where bilinear forms and linear functionals listed above have the following form:

$$\begin{cases} m(u,v) := \int_\Omega \rho u \cdot v dx \qquad\qquad a(u,v) := \int_\Omega a_{ijkm} e_{ij}(u) e_{km}(v) dx \\ c(u,v) := \int_\Omega c_{ijkm} e_{ij}(u) e_{km}(v) dx \; b(g,v) := \int_\Omega \theta_0 a_{ij} c_{ijkm} e_{km(v)gdx} \\ s(\theta,g) := \int_\Omega c_v \theta g dx \\ k(\theta,g) := \int_\Omega \chi_{ij} \theta_{,i} g_{,j} dx \\ \langle l,v \rangle := \int_\Omega f \cdot v dx + \int_{\Gamma_\mathrm{u}} \hat{\sigma} \cdot v ds \\ \langle z,g \rangle := \int_\Omega w g dx + \int_{\Gamma_\mathrm{q}} \hat{q} g ds \end{cases}$$

Galerkin semi discretization

Let's consider the variational problem in finite dimensional spaces. For this purpose, we will use classical Galerkin semi discretization scheme. So, we choose a finite-dimensional subspace $V_h \subset V, G_h \subset G\ \forall h > 0$ that $dimV_h = N = N(h) \to \infty$, $dimG_h = M = M(h) \to \infty$ and basic functions:

$$u_h(x,t) = \sum_{k=1}^{N} u_k(t)\varphi(x) \qquad\qquad \theta_h(x,t) = \sum_{k=1}^{M} \varrho_k(t)\phi(x)$$

Substituting these functions into variational formulation, we obtain the following Cauchy problem in matrix form

$$\begin{cases} \boldsymbol{MU}''(t) + \boldsymbol{CU}'(t) + \boldsymbol{AU}(t) - \boldsymbol{B\Theta}(t) = \boldsymbol{L}(t) \\ \boldsymbol{S\Theta}'(t) + \boldsymbol{K\Theta}(t) + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{U}'(t) = \boldsymbol{Z}(t)\ \forall t \in (0,T] \\ \boldsymbol{MU}'(0) = Y^0 \\ \boldsymbol{AU}(0) = U^0 \\ \boldsymbol{S\Theta}(0) = \Theta^0 \end{cases}$$

where the matrixes **M, A, C, B, S, K** and vectors **L, Z** calculated by expressions for bilinear forms and linear functionals.

One-step recurrent scheme of Cauchy integration over time

Now for final solution we have to build a one-step recurrent scheme (ORS) of Cauchy integration over time which will look in the following way

$$\begin{cases} Let\ \{\dot{U}^0, \Theta^0\}, \Delta t > 0, \gamma, \beta \in [0,1]; \\ find\ \left\{\ddot{U}^{j+\frac{1}{2}}, \dot{\Theta}^{j+\frac{1}{2}}\right\} and\ \{U^{j+1}, \dot{U}^{j+1}, \Theta^{j+1}\}\ that \\ \left[\boldsymbol{M} + \Delta t\gamma\boldsymbol{A} + \frac{1}{2}\Delta t^2\beta\boldsymbol{C}\right]\ddot{\boldsymbol{U}}^{j+\frac{1}{2}} - \Delta t\gamma\boldsymbol{B}\dot{\boldsymbol{\Theta}}^{j+\frac{1}{2}} = \boldsymbol{L}\left(t_{j+\frac{1}{2}}\right) - \boldsymbol{A}\dot{\boldsymbol{U}}^j - \boldsymbol{C}\left[\boldsymbol{U}^j + \Delta t\gamma\dot{\boldsymbol{U}}^j\right] + \boldsymbol{B\Theta}^j \\ -\boldsymbol{B}^T\Delta t\gamma\ddot{\boldsymbol{U}}^{j+\frac{1}{2}} + [\boldsymbol{S} + \Delta t\gamma\boldsymbol{K}]\dot{\boldsymbol{\Theta}}^{j+\frac{1}{2}} = \boldsymbol{Z}\left(t_{j+\frac{1}{2}}\right) - \boldsymbol{K\Theta}^{\boldsymbol{j}} - \boldsymbol{B}^T\dot{\boldsymbol{U}}^{\boldsymbol{j}} \\ \dot{\boldsymbol{U}}^{j+1} = \dot{\boldsymbol{U}}^j + \Delta t\ddot{\boldsymbol{U}}^{j+\frac{1}{2}} \\ \boldsymbol{U}^{j+1} = \boldsymbol{U}^{\boldsymbol{j}} + \frac{1}{2}\Delta t\left(\dot{\boldsymbol{U}}^j + \dot{\boldsymbol{U}}^{j+1}\right) \\ \boldsymbol{\Theta}^{j+1} = \boldsymbol{\Theta}^j + \Delta t\dot{\boldsymbol{\Theta}}^{j+\frac{1}{2}} \\ j = 0,1,2,\ldots while\ j\Delta t \le T \end{cases}$$

Numerical experiments

For our experiments, let's take one-dimensional case, that leads to some notation changes: $a_{ijkm} = a, c_{ijkm} = c,\ \chi_{ij} = \chi$. Our interval will be equal to **[0, 0.01]**, with number of finite elements $\boldsymbol{N = 500}$, time $\boldsymbol{T = 2.025 \cdot 10^{-5}}$ and step integration over

time $\Delta \boldsymbol{t} = \mathbf{5.4} \cdot \mathbf{10^{-9}}$. Also, we have acoustic pressure at right side, that correspond to the following rule:

$$p(t) = \begin{cases} 1Pa, & t \leq 1.6875 \cdot 10^{-6} \\ 0, & t > 1.6875 \cdot 10^{-6} \end{cases}$$

Physical quantities for water:

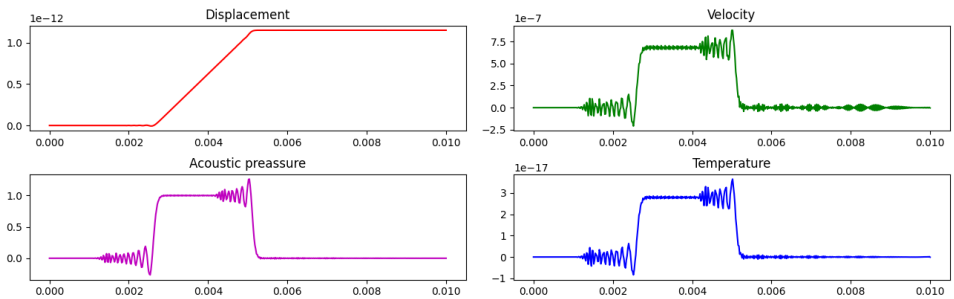
$$\rho = \mathbf{1000};\ a = \mathbf{2.157} \cdot \mathbf{10^{9}};\ c = \mathbf{2.004} \cdot \mathbf{10^{-3}};$$

$$C_v = \mathbf{100}; \Theta_0 = \mathbf{20}; \chi = \mathbf{0.6}; \alpha = \mathbf{1.5} \cdot \mathbf{10^{-4}}$$

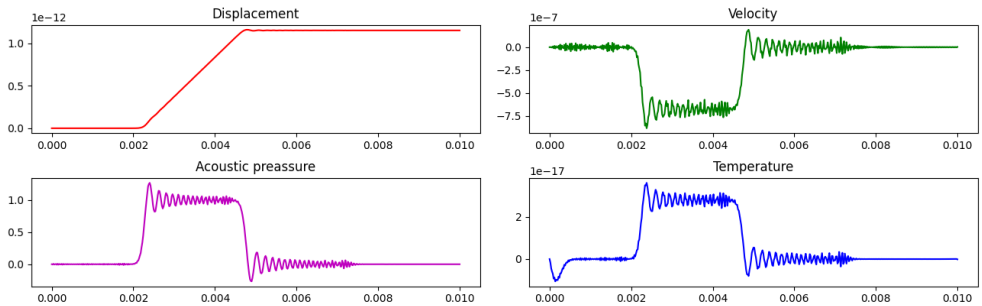
Physical quantities for fuel oil:

$$\rho = \mathbf{890};\ a = \mathbf{2.175} \cdot \mathbf{10^{9}};\ c = \mathbf{0.31};$$

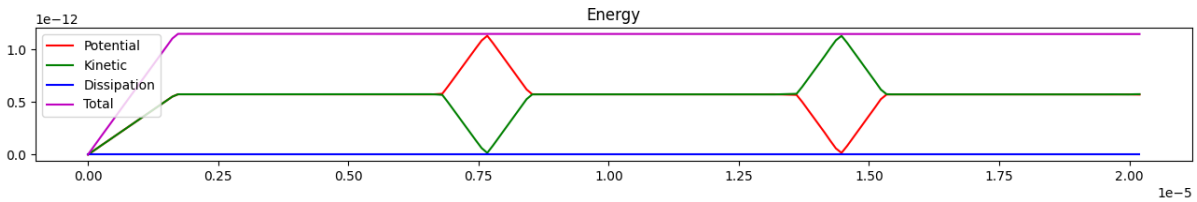
$$C_v = \mathbf{100}; \Theta_0 = \mathbf{20}; \chi = \mathbf{0.15}; \alpha = \mathbf{7} \cdot \mathbf{10^{-4}}$$



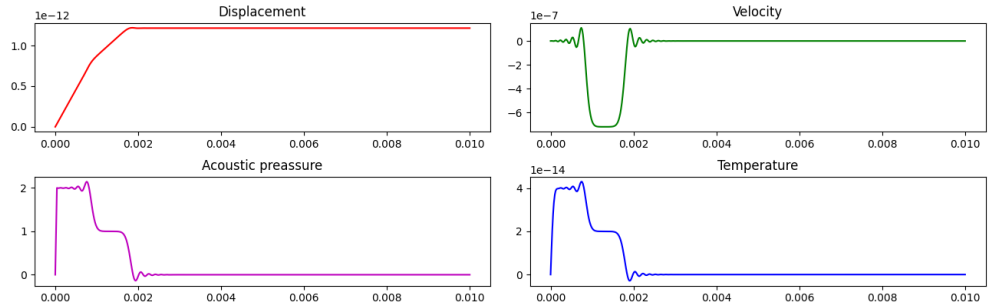
Water, $t = 0.5 \cdot 10^{-5} s$.



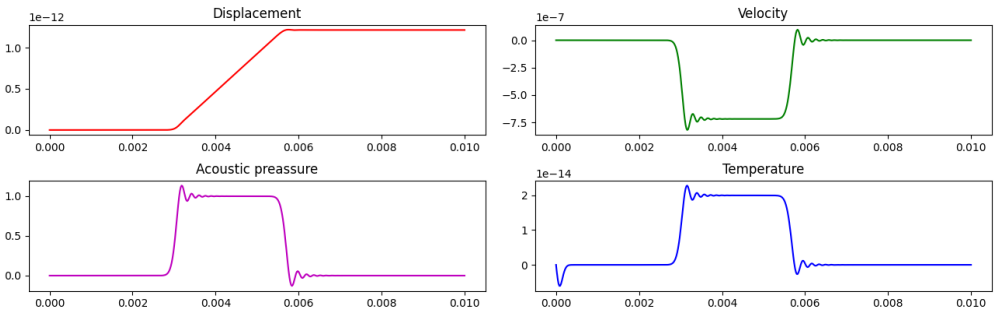
Water, $t = 1 \cdot 10^{-5} s$.



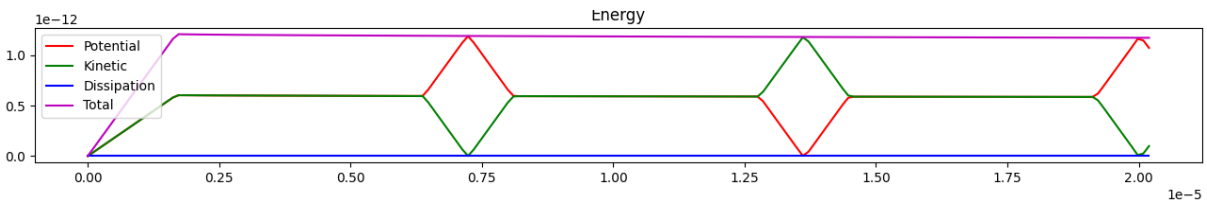
Water, energy change in time



Fuel oil, $t = 0.75 \cdot 10^{-5} s$.



Fuel oil, $t = 1 \cdot 10^{-5} s$.



Fuel oil, energy change in time

From the obtained results we can say that in fuel oil there are fewer oscillations than in water. In addition, the temperature graph is similar to the acoustic pressure graph, which explains the temperature fluctuations. Energy graph shows the moments when the kinetic energy is converted into potential and vice versa.

References

[1]. I. Dyyak, V. Horlatch and H. Shynkarenko, "Formulation and Numerical Analysis of Acoustics Problems in Coupled Thermohydroelastic Systems," 2019 XXIVth International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), 2019, pp. 168-171, doi: 10.1109/DIPED.2019.8882584.

[2]. V. Horlatch, I. Klymenko and H. Shynkarenko, “Finite element analysis of forced steady-state thermoacoustic waves in dissipative fluid,” Manufacturing Processes. Actual Problems –2013. Vol. 1. Basic Science Applications. Opole: Politechnika Opolska, 2013, pp. 47-66.

[3]. V. Horlatch, I. Klymenko and G. Shynkarenko, “Formulation and well-posedness of the variational problem of viscous heat-conducting fluid acoustics,”J.of Numerical and Applied Mathematics, No 3 (109),2012,pp. 53-71.

Створення моделі криптовалюти на основі технології блокчейн

Сас Ігор
Україна, Львів
Львівський Національний Університет імені Івана Франка
Факультет Прикладної Математики та Інформатики
ihorasas@gmail.com

Вступ.

Історія грошей налічує тисячі років. Спочатку, люди розраховувались бартером, тобто міняли одні товари на інші, пізніше вирішили використовувати цінні метали та дорогоцінне каміння, як більш універсальний інструмент обміну. Так з'явились монети зі срібла та золота, а пізніше – паперові гроші. Сьогодні, аби досягнути більшої конфіденційності та надійності, люди все частіше починають використовувати криптовалюти [1].

У 2008 році людина (чи група людей) під псевдонімом Сатосі Накамото представила концепт першої криптовалюти – біткоїна. Вона з'явилася, як альтернатива існуючій централізованій банківській системі. За задумом творця системи будь-яка центральна емісія повинна бути скасована замінена на персональну (особисту) емісію самих громадян [2].

Технологія Блокчейн.

Основна концепція блокчейну досить проста – це децентралізована та розподілена база даних, яка підтримує постійно зростаючий список упорядкованих записів [3]. Кожен блок містить індекс, позначку часу, дані (список транзакцій), хеш попереднього блоку, та власне хеш поточного блоку (див. рис.1).

Рис. 1. Спрощене представлення блокчейну

Найважливіша властивість блоку – Хеш, який обчислюється для усіх елементів блоку. Це означає, що якщо у блоці відбудуться зміни, оригінальний хеш стане не актуальним. Хеш блоку також можна розглядати як унікальний

ідентифікатор блоку. Наприклад, можуть з'являтися блоки з однаковим індексом, але всі вони мають унікальні хеші. Важливим наслідком полів хешу та попереднього хешу є те, що блок не може бути змінений без зміни хешу кожного послідовного блоку. Далі ми розглянемо алгоритм доказу виконання роботи (***proof-of-work***), де використовуватимемо покращений захист блоків.

Виконання роботи (Proof-of-work)

Для цього вводимо обчислювальну "головоломку", яку потрібно розв'язати, перш ніж блок можна буде додати до блокчейну. Спроба розгадати цю загадку широко відома як "майнінг". Головоломка доказу виконання роботи полягає в тому, щоб знайти хеш блоку, який має певну кількість нулів, які ставлять перед ним префікс. Властивість складності визначає, скільки префіксних нулів повинен мати хеш блоку, щоб блок був дійсним. Нулі префікса перевіряються з двійкового формату хешу [4-5].

Difficulty: 4

binaryHash: 0000100011110011.. VALID
hash: 08f3..

binaryHash: 0001011010111100.. NOT VALID
hash: 16bc..

Difficulty: 8

binaryHash: 0000000000100111.. VALID
hash: 0027..

binaryHash: 0000010111000010.. NOT VALID
hash: 05c2..

Рис. 2. Приклад хешів зі складністю

Для того, щоб знайти хеш, який задовольняє складність (***difficulty***), ми повинні вміти обчислювати різні хеші для одного й того ж вмісту блоку. Для цього введемо деякий параметр ***nonce*** і шляхом зміни цього параметра зможемо обчислювати різний хеш блоку. Оскільки SHA256 є хеш-функцією, кожен раз, коли що-небудь у блоці змінюється, хеш буде зовсім іншим. "Майнінг" – це по суті підбір значення ***nonce.***

Механізм транзакцій

Транзакції [2] складаються з двох компонентів: вхідних та вихідних даних. Вихідні дані вказують, куди відправляються монети, а вхідні дані підтверджують, що монети, які фактично надіслані, існують в першу чергу і належать "відправнику".

Вихідні дані транзакції (***txOut***) складаються з адреси та кількості монет. Адреса є відкритим ключем ECDSA. Це означає, що користувач, який має приватний ключ вказаного відкритого ключа, зможе отримати доступ до монет.

Входи транзакцій (***txIn***) надають інформацію про те, звідки беруться монети. Кожен ***txIn*** відноситься до попереднього виходу, з якого монети "розблоковуються", з підписом. Ці розблоковані монети тепер "доступні" для **txOuts**. Підпис свідчить про те, що транзакцію міг створити лише той користувач, який має закритий ключ згаданого відкритого ключа. Слід зазначити, що ***txIn*** містить лише підпис (створений приватним ключем), а ніколи сам приватний ключ. Блокчейн завжди містить лише відкриті ключі та підписи.

Структура транзакцій доволі проста – вона містить вхідні транзакції, вихідні та ідентифікатор. Ідентифікатор транзакції обчислюється шляхом взяття хешу із вмісту транзакції. Однак підписи ***txIds*** не включаються в хеш транзакції, оскільки пізніше буде додано до транзакції.

Важливо, щоб зміст транзакції не міг бути змінений після її підписання. Оскільки транзакції є загальнодоступними, кожен може отримати доступ до транзакцій, навіть до того, як вони будуть включені в блокчейн.

Підписуючи вхідні дані транзакції, буде підписано лише ***txId***. Якщо будь-який вміст транзакцій змінено, ***txId*** повинен змінитися, роблячи транзакцію та підпис недійсними.

Вхід транзакції завжди повинен посилатися на невитрачений результат транзакції (***uTxO***). Отже, коли ви володієте деякими монетами в блокчейні, у вас насправді є список невитрачених виходів транзакцій, відкритий ключ яких відповідає приватному ключу, яким ви володієте.

Що стосується перевірки транзакцій, ми можемо зосередитись лише на списку невитрачених результатів транзакцій, щоб з'ясувати, чи є транзакція дійсною. Список невитрачених виходів транзакцій завжди можна отримати з поточного блокчейну.

Входи транзакцій завжди повинні посилатися на невитрачені результати транзакції. Але звідки початкові монети надходять у блокчейн? Для вирішення цього вводиться спеціальний тип транзакції: транзакція ***coinbase,*** що містить лише вихідні дані. Це означає, що транзакція на монетній базі додає нові монети до обігу. Ми визначаємо суму випуску монетної бази 50 монет. Транзакція ***coinbase*** – це завжди перша транзакція в блоці, і вона включається майнером блоку. Винагорода ***coinbase*** діє як стимул для майнерів: якщо ви знайдете блок, ви зможете зібрати 50 монет.

Ми будемо зберігати наші непідтверджені транзакції в новому об'єкті, який називається “пул транзакцій”. Пул транзакцій – це структура, яка містить усі “непідтверджені транзакції”, про які знає наш вузол. У цій простій реалізації ми просто використаємо список. Вся суть непідтверджених транзакцій полягає в тому, що вони будуть поширюватися по всій мережі, і врешті-решт якийсь вузол здійснить майнінг транзакції в блокчейні.

Мережеві технології.

Важливою частиною вузла є спільний доступ та синхронізація блокчейну з іншими вузлами. Наступні правила використовуються для синхронізації мережі.

- Коли вузол генерує новий блок, він передає його в мережу [2,6].
- Коли вузол підключається до нового вузла, він запитує останній блок.
- Коли вузол зустрічає блок, який має індекс, більший за поточний відомий блок, він або додає блоку його поточний ланцюжок, або запитує повний блокчейн.
- Коли вузол отримує непідтверджену транзакцію, якої він раніше не бачив, він передаватиме повний пул транзакцій усім вузлів.
- Коли вузол вперше підключається до іншого вузла, він запитує пул транзакцій цього вузла.

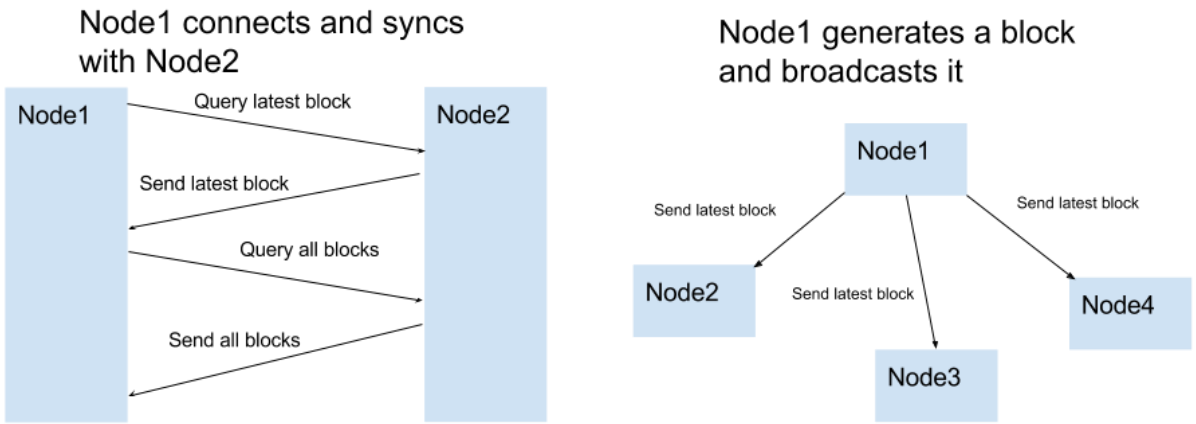


Рис. 3. Комунікація між вузлами мережі

Для спілкування використовуються веб-сокети ***peer-to-peer***. Користувач повинен мати можливість якимось чином керувати вузлом. Це робиться шляхом налаштування сервера HTTP.

Слід зазначити, що вузол насправді має два веб-сервери: один – для керування користувачем (HTTP-сервер), а другий – для ***peer-to-peer*** спілкування між вузлами (HTTP-сервер Websocket).

Висновки.

У даній роботі розглянуто основні принципи побудови технології блокчейну із врахуванням захищеності системи. На практиці показано, як побудовані криптовалюти, а саме головна її компонента – транзакція. Також висвітлено основні шляхи спілкування між вузлами мережі, тобто як досягнути децентралізації системи. Дана робота загалом дає уявлення про криптовалюту і допоможе зрозуміти механізми її створення та використання “зсередини”.

Літературні першоджерела.

1. Біткоїни – цифрові гроші майбутнього. – [Електронний ресурс]. – Режим доступу : http://thefuture.news/bitcoin/
2. Proof of Work can Work. – [Електронний ресурс]. – Режим доступу : https://www.investopedia.com/terms/b/blockchain.asp
3. How Bitcoin Mining Works. – [Електронний ресурс]. – Режим доступу : https://www.bitcoinmining.com/
4. Blockchain Explained. – [Електронний ресурс]. – Режим доступу : https://econinfosec.org/archive/weis2006/docs/50.pdf
5. Bitcoin: A Peer-to-Peer Electronic Cash System. – [Електронний ресурс]. – Режим доступу : https://bitcoin.org/bitcoin.pdf
6. Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, Emin Gün Sirer: Decentralization in Bitcoin and Ethereum Networks. – Computer Science Department, Cornell University – 2018.

Програмно-алгоритмічне забезпечення розпаралелення навчання штучних нейронних мереж з використанням технологій CUDA

Манохін Денис

Львівський національний університет ім. Івана Франка, ФПМІ

denmanokhin@gmail.com

Актуальність теми. Штучні нейронні мережі (ШНМ) — є надзвичайно потужним інструментом машинного навчання, що сьогодні використовується чи не в кожній сфері життя. Камери мобільних телефонів, перекладачі, голосовий ввід, електронні помічники та навіть мистецтво — це лише невелика частина того, де вони використовуються. Однак, раніше, через складність обчислень та великі об'єми даних, ефективно навчати ШНМ можна було лише за допомогою надзвичайно дорогих обчислювальних машин з великою потужністю. Ситуація змінилася лише нещодавно, з появою технології CUDA, що дозволила тренувати ШНМ за допомогою відносно дешевих та потужних графічних процесорів (GPU), які сьогодні встановлені в більшості персональних комп'ютерів. А розвиток фреймворків на базі мови програмування Python, значно спростив процес розробки. Разом це знизило вхідний поріг у галузь глибинного навчання і забезпечило їй популярність, яку вона має сьогодні.

Мета. Враховуючи важливу роль апаратного прискорення процесу тренування в розвитку сучасних ШНМ було вирішено дослідити особливості програмної реалізації процедури тренування ШНМ з використанням GPU та оцінити переваги та недоліки такого підходу.

Використані технології. Дослідження були проведені за допомогою мови програмування Python версії 3.7.7 та фреймворку Chainer версії 7.4.0. Для

обчислень на GPU була використана платформа CUDA версії 10.1, разом з бібліотекою cuDNN, версії 7.6.5.

Огляд технології CUDA. Архітектура GPU бренду Nvidia ґрунтується на основі потокових мультипроцесорів (SM). Ці мультипроцесори спеціально розроблені для того, щоб виконувати обчислення у сотнях потоків одночасно. Для цього вони використовують спеціальну архітектуру SIMT (Single-Instruction. Multiple-Thread), що є модифікацією SIMD (Single-Instruction, Multiple-Data), яка полягає в тому, що одна інструкція виконується одночасно для великої кількості даних.

Одним з основних концептів у CUDA програмуванні є ядра. Це функції, що при виклику будуть виконані кілька разів, кожна в окремому потоці. Потоки об'єднуються у блоки — 1, 2 або 3-вимірні структури. Оскільки блок повинен знаходитись в межах одного мультипроцесора, кількість потоків у блоці обмежена його ресурсами. На сучасних GPU це близько 1024 потоків на блок. Блоки ж об'єднують в 1, 2 або 3-вимірну сітку. Аналогічно, вона має обмеження на кількість блоків, оскільки повинна знаходитись на одному GPU.

Розмірність сітки та блоків вказується при виклику функції. Це дозволяє легко масштабувати алгоритм, без необхідності модифікації коду функції. [1]

Огляд фреймворку Chainer. Фреймворк Chainer значно спрощує використання технології CUDA для навчання ШНМ. Завдяки ньому розробник може використовувати для тренування на GPU майже той самий високорівневий код мовою Python, що використовується для тренування на CPU, абстрагуючись від деталей реалізації.

Розробники Chainer запропонували новий підхід — Define-by-Run. Основною ідеєю в ньому є те, що граф обчислень будується динамічно прямо під час тренування. Під час проходу вперед, кожна функція, окрім обчислення значення, зберігає історію обчислень разом з посиланням на попередній вузол у графі. Оскільки структура графу залежить від шляху виконання програми, стало можливе використання синтаксичних конструкцій мови програмування, таких як

умовні оператори та цикли. Окрім цього тепер з'явилась можливість використання зневаджувачів для детальнішого дослідження процесу тренування. [2]

Проведені дослідження. В ході роботи було розглянуто дві моделі, застосовані до двох наборів даних.

Перша — багатошаровий перцептрон з двома прихованими шарами, по 32 нейрони на кожен. Активаційні функції — ReLU. Застосований до класифікації писаних цифр датасету MNIST, що складається з 60000 тренувальних, та 10000 тестових зображень, розміру 28 на 28 пікселів, поділених на 10 категорій (цифри від 0 до 9). Розмір датасету — близько 10 Мб.

Друга — згорткова нейронна мережа архітектури LeNet-5 [3]. Застосована до датасету CIFAR-10, що складається з 50000 тренувальних та 10000 тестових зображень, розміру 32 на 32 пікселі. Кожне зображення належить до одного з 10 класів (літак, автомобіль, птах, кіт, олень, собака, жаба, кінь, корабель, вантажівка). Розмір датасету — близько 150 Мб.

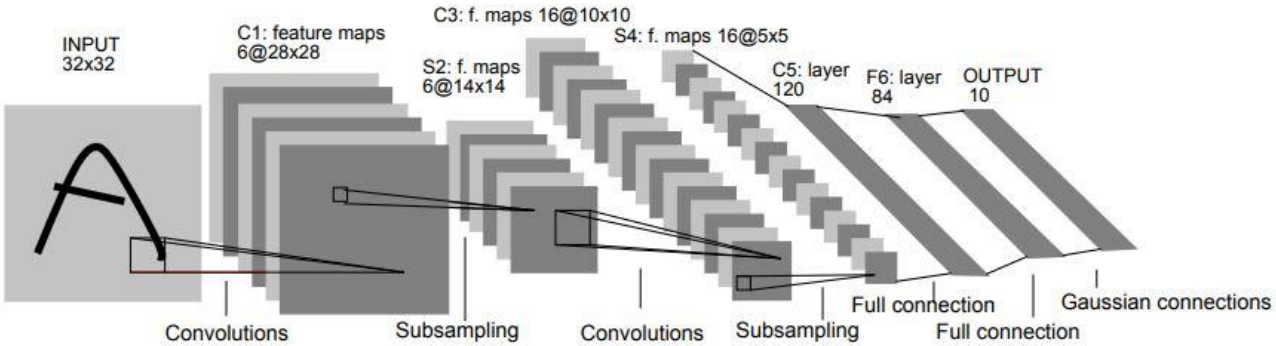


Рисунок 1. Архітектура LeNet-5

Тренування ШНМ відбувалось на персональному комп'ютері (ПК) характеристики якого наведені в таблицях 1-2.

ОС	Ubuntu 16.04 LTS
CPU	Intel Core i5-7200U 2.50 ГГц x 4
RAM	6 Гб DDR4
GPU	NVIDIA GeForce GTX 950M

Таблиця 1. Характеристики ПК

Версія драйверу	430.64
Архітектура	Maxwell
Об’єм пам’яті	4 Гб
Кількість мультипроцесорів	5
Кількість CUDA ядер	640
Максимум потоків на блок	1024

Таблиця 2. Характеристики GPU

Для оцінки прискорення, тренування було проведено як за допомогою GPU так і CPU. Оскільки на час тренування впливає дуже багато факторів, його варто розглядати як випадкову змінну. Тому в якості часу тренування обчислено середнє значення та стандартне відхилення серед 5 спроб. Тренування першої моделі проводилось протягом 10 епох з різним розміром порції (batchsaize). Друга модель була натренована лише на 1 епоху з розміром порції 128.

Аналіз результатів.

Пристрій	Batchsize	Час	Прискорення	Точність
CPU	64	56,13 ± 0,93 с	1,57	96,6
GPU		35,70 ± 3,64 с		
CPU	256	21,27 ± 0,70 с	1,75	96,1
GPU		12,12 ± 3,87 с		
CPU	1024	11,59 ± 0,49 с	1,93	94,7
GPU		5,99 ± 2,45 с		

Таблиця 3. Результати для першої моделі

Можна побачити, що зі збільшенням розміру порції зростає прискорення, оскільки тоді більше даних обробляються одночасно, а отже ефективніше використовується GPU. Однак також можна помітити, що разом із цим спадає точність. Тому краще не використовувати занадто великі значення цього параметру. Якщо ж процес навчання не використовує наявні ресурси на повну,

можна спробувати збільшити розмір порції і при цьому збільшити коефіцієнт навчання.

Також, варто зауважити, що прискорення в 1,5-2 рази не є значним, та може бути досягнене і за допомогою багатопоточності на CPU. Саме тому було розглянуто другу модель.

Пристрій	Batchsize	Час	Прискорення
CPU	128	82,58 ± 1,54 с	12,36
GPU		6,68 ± 4,2 с	

Таблиця 4. Результати для другої моделі

Прискорення від використання GPU стає більш помітним зі збільшенням складності моделі та об'єму даних. Як можна побачити одна епоха навчання другої моделі на CPU займає більше часу ніж весь процес тренування першої, однак з апаратного прискорення за допомогою GPU ми отримуємо прийнятний результат часу виконання.

Висновок. Отримані результати дають змогу оцінити прискорення яке дозволяє отримати використання технології CUDA при навчання ШНМ. Проте, в цій роботі було розглянуто лише досить невеликі штучні датасети та одні з найпростіших моделей. Справжні набори даних часто містять гігабайти інформації, та потребують набагато складніших архітектур ШНМ. Однак потужностей GPU середньої цінової категорії для таких задач все ще недостатньо. Тут в нагоді стають хмарні сервіси, які надають доступ до передових GPU з погодинною оплатою. В подальшому планується використати один з них для прискорення тренування ШНМ для вирішення задачі обробки медичних даних.

Використана література.

[1] CUDA C++ Programming Guide v10.2.89 [Electronic resource]. — 2019. — Available from : *https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html*

[2] Chainer: A Deep Learning Framework for Accelerating the Research Cycle / [Seiya Tokui, Kenta Oono, Shohei Hido, et al.]. — 2019.

[3] Gradient-Based Learning Applied to Document Recognition / [Yann LeCun, Leon Bottou, Yoshua Bengio et al.]. — 1998.

Задача ідентифікації та оптимізації для моделі взаємодії популяцій

Олена Тиміцька
Факультет прикладної математики та інформатики
Львівський національний університет імені Івана Франка
Україна, Львів
olenatymitska@gmail.com

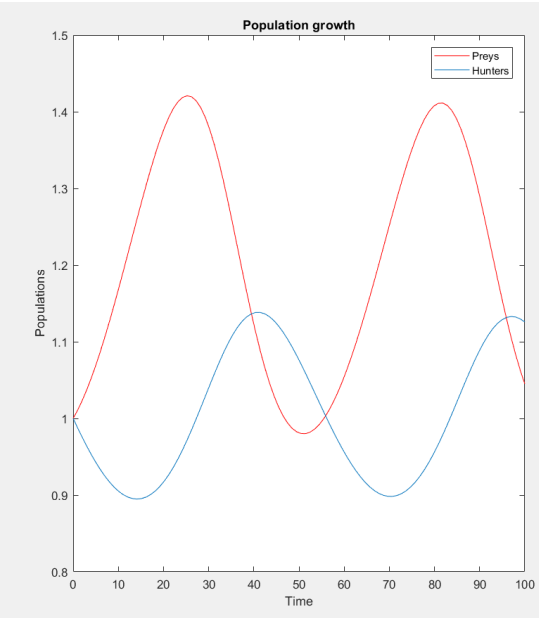
У теперішньому часі дуже частим та актуальним стало дослідження моделей взаємодії популяцій, яке, зокрема, необхідне у мікробіології, біохімії, економіці, медицині. Для того, щоб описати процес, який підлягає дослідженню, важливо підібрати такі параметри для моделі, які будуть давати найменшу похибку між експериментальними даними та розв'язком математичної моделі.

Будемо розглядати модель взаємодії двох популяцій на прикладі моделі Лотки-Вольтерра (модель «хижак-жертва») . Нехай дано таку математичну модель:

$$\begin{cases} \dfrac{dx(t)}{dt} = ax(t) - bx(t)y(t) \\ \dfrac{dy(t)}{dt} = -cy(t) + dx(t)y(t) \end{cases}, t \in (t_0, T) \quad , \qquad (1)$$

де $x(t), y(t)$ - функції, що моделюють густину популяцій жертви і хижака, відповідно. Будемо вважати, що внутрішньовидова конкуренція присутня. Функція $ax(t)$ в системі описує розвиток популяції жертв, а функція $cy(t)$ - розвиток популяції хижака. Функція $bx(t)y(t)$ характеризує взаємодію хижаків і жертв, а саме швидкість поїдання жертв в залежності від густини популяції жертви і хижака. Функція $dx(t)y(t)$ показує розмноження хижаків за рахунок жертв.

Для розв'язування будемо використовувати математичний пакет Matlab (зокрема, розв'язник ode45) . Покажемо графічні результати в залежності від часу (Рис. 1) і фазову траєкторію (Рис. 2) залежності між густиною популяцій хижака та жертви. Задавши конкретні значення параметрів, отримаємо:



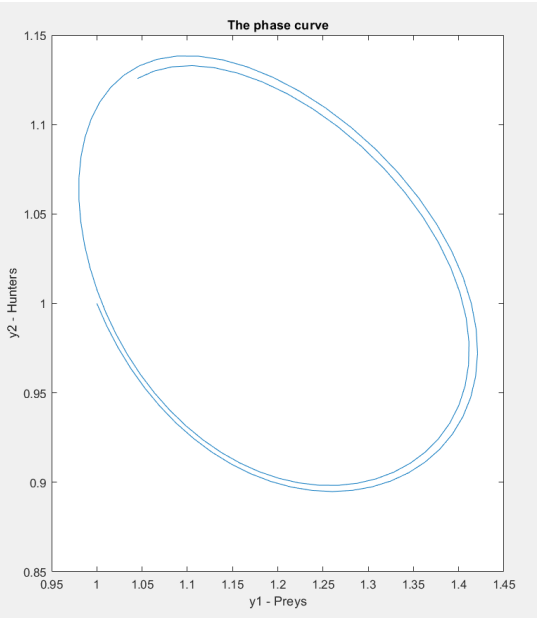
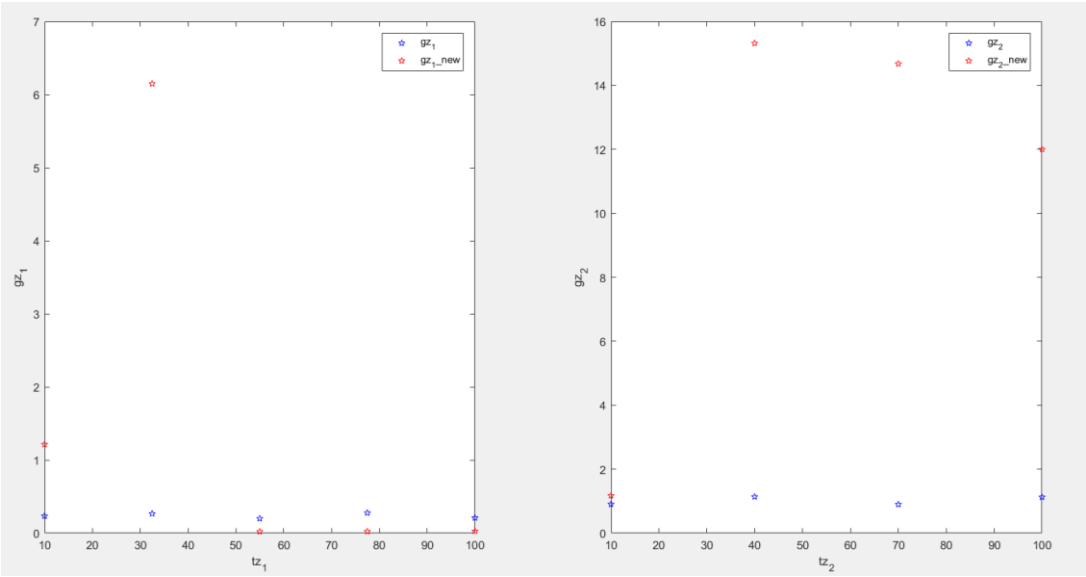


Рис. 1. Чисельність популяцій в часі

Рис. 2. Фазова траєкторія системи «хижак-жертва»

Наступним завданням буде підібрати параметри та на основі них побудувати область допустимих значень параметрів, сформувати вектор початкового наближення параметрів ідентифікації та знайти значення функціоналу. Для знаходження функціоналу використовуємо метод скінченних різниць та прямий метод диференціювання. Всі приклади таких обчислень виконуються програмно.



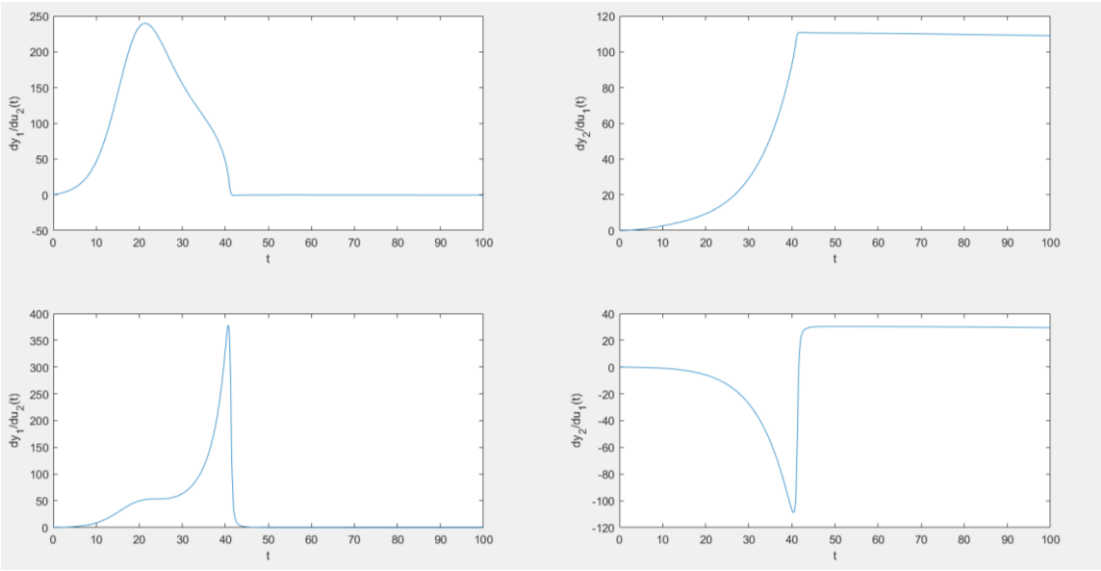


Рис. 3. Коефіцієнти чутливості, отримані методом скінченних різниць

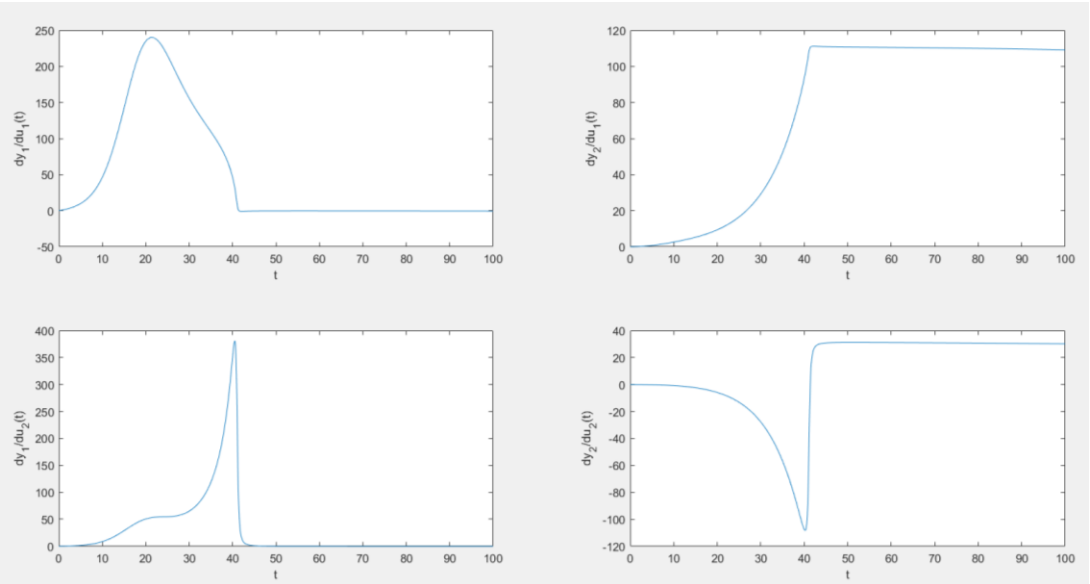


Рис. 4. Коефіцієнти чутливості, отримані методом прямого диференціювання

Розв’язок задачі оптимізації функціоналу будемо шукати за допомогою функцій *fmincon* та *fminsearch*.

Література

[1] Бабаджанянц Л.К. Идентификация параметров осциллирующих процессов в живой природе, моделируемых дифференциальными уравнениями //Проблемы современного естествознания – Биология и Медицина, 2002

[2] Aitziber Ibanez : Optimal control and turnprike properties of the Lotka Volterra model

[3] Malaysian Journal of Mathematical Sciences 12(2): 269-285 (2018)

[4] Lenhart. S. and Workman, J. T.(2007). Optimal Control Applied to Biological Models. Chapman

Розробка веб-аплікації для створення та керування розкладом занять в університеті

Наталія Пачва
Україна, Львів
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
natalia.pachva@gmail.com

Вступ

Сучасна освіта є надзвичайно багатогранною, пропонуючи широкий вибір навчальних дисциплін усіх напрямків та спеціальностей. Проте, зростання кількості можливостей у вивченні того чи іншого предмету призводить й до великої завантаженості як студентів, так і викладачів вищих навчальних закладів.

Прямим наслідком цього є той факт, що розклади занять у навчальних закладах стають дедалі більшими за розмірами та містять надзвичайно багато інформаціїї. Так, більшість університетів мають свої спеціально створені веб-сайти, що містять усю інформацію про навчальний процес. Однак спосіб подання цієї інформації та система навігації, що допомагає віднайти потрібні дані, такі як, наприклад, розклад конкретного викладача на семестр, чи список занять, що проходять в конкретно обраній аудиторії наступного тижні – усе це в багатьох випадках є не надто зручним. Саме тому важливо мати спосіб представити корисні студентам та викладачам дані у компактному та гнучкому форматі, який з легкістю можна налаштувати в залежності від особистих вподобань і потреб.

UniversitySchedule – це веб-аплікація, що дозволяє у зручний та швидкий спосіб створювати, редагувати та переглядати розклад університету, застосовуючи при цьому різноманітні фільтри та налаштування. Варто зазначити, що дана аплікація підтримує два режими перегляду інформації про заняття в університеті. Перший – це класичний вигляд розкладу, що імітує друковану

версію розкладу університету, із можливістю застосування різного роду фільтрів для відображення інформації. Другий – це розклад у вигляді календаря подій, у якому відображаються обрані заняття, що можна експортувати у форматі файлу .ics [1] для подальшої інтеграціїї у зовнішні популярні інтернет-календарі.

Архітектура та використані технології

При проектуванні веб-аплікаціїї *UniversitySchedule* використано класичну тришарову архітектуру [2]. Концепція багатошарової моделі базується на розподілі всієї системи на окремі ключові функціональні частини: шар доступу до даних, шар бізнес правил та шар представлення даних. Аплікація, спроектована із дотриманням канонів вищезгаданої архітектури, має низку переваг. Так, такий веб-застосунок зручно підтримувати; окремі компоненти такої аплікації можуть бути перевикористані; для такого роду аплікацій є притаманна висока гнучкість, коли виникає потреба у зміні бізнез логіки; масштабованість аплікацій із тришаровою архітектурою є досить високою.

Для програмної реалізації даного проекту було використано низку сучасних технологічних рішень та застосунків. Зокрема, для імплементації серверної частини було взято за основу ASP.NET Core Web API [3], а у якості ORM-інструменту було обрано такий популярний фреймворк, як Entity Framework Core. Для зберігання даних використано провайдер MS SQL. З метою впровадження системи автентифікації користувачів, а також можливості їх авторизації, було використано платформу Microsoft Identity [4] та стандарт JWT (Json Web Token) [5]. Слід зазначити, що веб-аплікація *UniversitySchedule* підтримує автентифікаію за допомогою облікових даних Google, реалізовану із допомогою платформи Firebase. Такий альтернативний спосіб входу у систему має свої переваги, а саме – можливість автоматичної синхронізації із Google календарем, для імплементації якої було використано Google Calendar API [6]. Також, для роботи із даними типу iCalendar на серверній частині було використано бібліотеку iCal.Net, а на частині клієнта, задля візуального відображення та взаємодії із даними, – плагін

FullCalendar [7]. У якості UI фреймворку було обрано Angular, зі стилізацією засобами Angular Material Design.

Можливості аплікації веб-застосунку

Аплікація містить три основні компоненти, які дозволяють користувачеві взаємодіяти із розкладом, а саме: Edit (Редагування), View Timetable (Переглянути Розклад), View Calendar (Переглянути Календар). Зазначимо, що користувач повинен бути автентифікований та авторизований у системі, щоб мати змогу переглядати та взаємодіяти зі сторінкою Edit (Редагування).

На сторінці Edit у користувача є можливість створювати, редагувати та видаляти усі сутності аплікації, що у результаті формують цілісний розклад занять: Timetables (Розклад), Buildings (Корпуси/будівлі університету), Faculties (Факультети), Periods (Часові рамки занять), Classrooms (Аудиторії), Groups (Групи студентів), Professors (Викладачі), Subjects (Предмети), Lessons (Заняття). Зазначимо також, що у користувача аплікації присутня можливість додавати до занять посилання із запланованими у багатьох популярних сервісах онлайн-зустрічами.

Сторінка View Timetable дозволяє користувачу переглядати обраний розклад у «класичному» форматі. Так, у програмі *UniversitySchedule* передбачено можливість відображати поточний розклад для конкретних викладачів, студентських груп та аудиторій.

На сторінці View Calendar користувачеві стає доступний розклад у вигляді календарних подій. Подібно до сторінки View Timetable, календар можна відобразити для обраного розкладу, застосовуючи різноманітні параметри фільтрування. Також користувач має можливість експортувати обраний розклад у файл формату .ics, а також синхронізувати його (розклад) із особистим Google календарем.

Висновки

Спроектовано та реалізовано веб-аплікацію *UniversitySchedule*, що дозволяє користувачам зручно працювати із розкладами занять вищих навчальних закладів, а також експортувати дані у вигляді файлів із календарем подій, який можна в подальшому імпортувати у зовнішні популярні сервіси для роботи із інтернет-календарями. Вдалося вирішити багато проблем, з якими зустрічається викладач або студент університету під час пошуку потрібної інформації щодо навчальних занять, шляхом створення гнучкого програмного забезпечення, яке дозволяє користувачу системи швидко та без зайвих зусиль видобути корисну та важливу інформацію.

Література

1. iCalendar Specification (RFC 5545) [Електронний ресурс] – Режим доступу до ресурсу: https://icalendar.org/RFC-Specifications/iCalendar-RFC-5545/.
2. Bibile U. N-Tier Architecture in ASP.NET Core [Електронний ресурс] / Udara Bibile // Medium.com. – 2019. – Режим доступу до ресурсу: https://medium.com/@chathuranga94/n-tier-architecture-in-asp-net-core-d1f1b14f2010.
3. Troelsen A. W. Pro C# 7 With .NET and .NET Core / A. W. Troelsen, P. Japikse. – Apress, 2017. – 1372 p.
4. Microsoft Identity [Електронний ресурс] – Режим доступу до ресурсу: https://aspnetcore.readthedocs.io/en/stable/security/authentication/identity.html.
5. JSON Web Tokens [Електронний ресурс] – Режим доступу до ресурсу: https://jwt.io/.
6. Calendar API [Електронний ресурс] – Режим доступу до ресурсу: https://developers.google.com/calendar/overview.
7. FullCalendar Docs [Електронний ресурс] // FullCalendar. – 2019. – Режим доступу до ресурсу: https://fullcalendar.io/docs.

Named Entity Extraction of Unstructured text

Olaleye Ayodeji Olurotimi Ukraine/ Italy
Ivan Franko National University of Lviv/ University of L'Aquila i
Faculty of Applied Mathematics and Informatics / DISIM
Intermaths program, Interdisciplinary mathematics
ayodeji.olaleye@intermaths.eu

Named entity recognition is a records extraction method that makes use of Natural Language Processing that allows machines to routinely identifies and pick out key factors from unstructured text, then classifies them into predefined classes like item name, event, agency, and location. This makes unstructured information machine readable and accessible for general processing activities which include retrieving records, extracting records, and answering questions. It is utilized by engines like google to apprehend queries, chatbots to interact with humans, and groups to automate tedious duties like information entry.

This paper focuses on named entity extraction, the conversion of unstructured text to structured text, implementing, and analyzing a neural network model on unstructured text from Kaggle which contains over 506 files with minimum of 7 pages of words per each file. It is first processed, converted into structured texts, split into tokens and tags which is divided into three distinct classes of train data for training the model, validation data for evaluation and hyperparameters tuning, test data for final evaluation of the model. It was divided in a ratio of 60:20:20 respectively. We performed some pre-processing, then implemented a spacy library for the conversion of unstructured text to structured text, then implemented a Bi-LSTM model; a variant of recurrent neural network with the tensorflow framework which we use for the structured text classification. The Bi-LSTM neural network is composed of long short-term memory units that operate in both forward and backward directions in

order to capture past context information and future context information. Bi-LSTM can learn long-term dependencies without holding back duplicate context information.

Bi-LSTM can literarily be describe as an LSTM that goes from left to right (forward direction), and another LSTM that goes from right to left (backward direction), in which layers, hidden layers are concatenated.

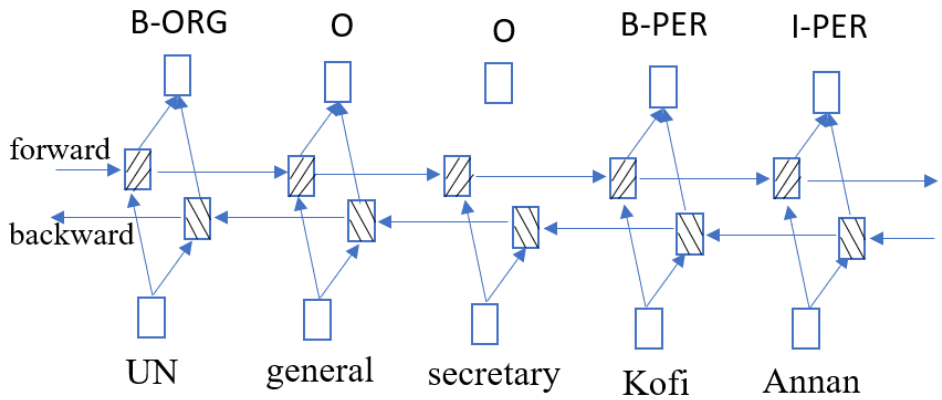


Figure 1: A Bi-LSTM network.

The Bi-LSTM layer can be expressed mathematically as:

$$\overrightarrow{h_t}^{(k)} = f\left(\vec{C}^{(k)}\overrightarrow{h_t}^{(k-1)} + \vec{D}^{(k)}\overrightarrow{h_{t-1}}^{(k)} + \vec{z}^{(k)}\right) \quad (1)$$

$$\overleftarrow{h_t}^{(k)} = f\left(\overleftarrow{C}^{(k)}\overleftarrow{h_t}^{(k-1)} + \overleftarrow{D}^{(k)}\overleftarrow{h_{t+1}}^{(k)} + \overleftarrow{z}^{(k)}\right) \quad (2)$$

$$\vec{q}^{(k)} = \mathrm{concat}\left(\overrightarrow{h_t}^{(k)}, \overleftarrow{h_t}^{(k)}\right) \quad (3)$$

Where $\vec{q}$ *is the output vector*, $\vec{C}^{(k)}$, $\vec{D}^{(k)}$ are the weight matrix and $\vec{z}^{(k)}$ is the offset vector which was produced in the forward propagation at the k layer of the Bi-LSTM model, likewise, $\overleftarrow{C}^{(k)}, \overleftarrow{D}^{(k)}$ are the weight matrix and $\overleftarrow{z}^{(k)}$ is the offset vector produced in the backward propagation at the k layer of Bi-LSTM model, $\overrightarrow{h_t}^{(k)}$ is the past context information and $\overleftarrow{h_t}^{(k)}$ is the future context information[1].

Like all variant of neural networks, Bi-LSTM is neither different, it does not take raw text as input, it must be converted, Bi-LSTM works with numeric tensors. So, we transform the text into numeric tensors, we segment the text into words, and transform each word into vector. The vectors are associated with tokens using word embeddings. These vectors, packed into sequence tensors, are fed into deep neural networks [2].

After the tuning of the hyperparameters to increase the F1 score of both training, validation and test sets while been conscious of overfitting and underfitting error, the accuracy after training the model after several iterations was 97.82%, the validation accuracy was 80.22% and test accuracy was 80.04%.

References

[1] *Chuanchuam Zhao, Jinguo You, Xinxian Wen, Xiaowu Li* Deep Bi-LSTM Networks for Sequential Recommendation. // Entropy 2020,22,870; doi:10.3390/e22080870, 2020.

[2] *Francois Chollet*, Deep Learning with Python. // Manning publications, 2018.

Паралельні методи розв’язування систем нелінійних алгебраїчних рівнянь

Андреєв Ярослав Сергійович, Україна
Львівський національний університет ім. Івана Франка
Факультет прикладної математики та інформатики
yaroslav.andreiev@lnu.edu.ua

Вступ

Із швидким розвитком апаратного забезпечення відкриваються можливості для прискорення обчислень різного роду, зокрема для знаходження розв'язку систем нелінійних алгебраїчних рівнянь та диференціальних рівнянь високого порядку.

Актуальність розробки методів наближеного розв’язування систем рівнянь обумовлена широким спектром їх використання.Наприклад,у рівняннях усталеного режиму роботиі балансу потужності електромережі використовують метод Ньютона для розв’язування систем нелінійних рівнянь.

У даній доповіді розглянуто ітераційні методи в послідовній та паралельній реалізації.

Постановка задачі

Необхідно знайти розв'язок системи з n рівнянь

$$f(x) = 0 \tag{1}$$

у заданій області $G = \{a_i \leq x_i \leq b_i (i = 1,2, \dots, n)\}$, де $x = (x_1, x_2, \dots, x_n)^T$, $f(x) = \left(f_1(x), f_2(x), \dots, f_n(x)\right)^T$; x – шуканий n-вимірний вектор.[1]

Як приклад розглянемо послідовний та паралельний метод Ньютона.

Послідовні та паралельні алгоритми методу Ньютона

Метод Ньютона для систем нелінійних рівнянь є узагальненням методу Ньютона для одного нелінійного рівняння. Нехай $f(x): \mathbb{R}^1 \to \mathbb{R}^1 -$ диференційовна функція. Потрібно знайти розв'язок рівняння (1).

На відміну від класичної послідовної реалізації, паралельна використовує метод Гауса, в якому між процесорами розподіляються рядки вихідної матриці Якобі.

Для написання програми на мові C++ застосовуємо бібліотеку MPI.

MPI [4] (messagepassinginterface) - бібліотека функцій, призначена для підтримки роботи паралельних процесів. Вона дає можливість використання єдиного механізму взаємодії процесів усередині програми, що реалізована паралельно, незалежно від архітектури.

Окрім реалізації на C++, дослідження проводились на мові C#ізвикористаннямнаступнихбібліотек[2]:

- System.Threading.Thread;
- System.Threading.Tasks;
- System.Threading.Tasks.Parallel.

Перейдемо до чисельних експериментів.

Чисельніекспериментиі

Наведеморезультатиобчисленьпаралельного методу Ньютонакомп'ютеромз восьмиядернимпроцесором AMD Ryzen 2500U. Оскількипроцесормаєвісімпотоків, візьмемо таку систему з восьми рівнянь [3]:

$$\begin{cases} 2x - sin\,0{,}5(x-y) = 0, \\ 2y - cos\,0{,}5(x+y) = 0, \\ 2x - sin\,0{,}5(x-y) = 0, \\ \vdots \\ 2y - cos\,0{,}5(x+y) = 0. \end{cases}$$

Наведемо результати у вигляді двох таблиць, де $k -$ кількість ітерацій, $t -$ часвиконання обчислень, $\varepsilon -$ задана точність, $x_0/y_0 -$ початкові наближення, $x_k/y_k -$ знайдені розв’язки системи.

Послідовнареалізаціяалгоритмів є доситьефективною для невеликих систем, адже час наведений в таблицях є оптимальним для більшості задач.

Табл1. Послідовний алгоритм методу Ньютона

k	x_0 / y_0	x_k / y_k	ε	*t ,мс*
2	0/0.5	-0,08026382277266 0,246557509876119	0.001	80
3	0/0.5	-0,080254957080771 0,246551155777762	0.000000001	120
5	-3/5	-0,080254957073489 0,246551155774001	0.000000001	127

Табл2. Паралельний алгоритм методу Ньютона

k	x_0 / y_0	x_k / y_k	ε	t ,*мс*
4	0/0.5	-0,08026382377265 0,246557509875139	0.001	70
7	0/0.5	-0,080254957480771 0,246551155778762	0.000000001	103
11	-3/5	-0,080254957173489 0,246553155775021	0.000000001	121

Висновки

Отже, проаналізувавши результати, бачимо, що при паралельній реалізації кількість ітерацій збільшилась у порівнянні з аналогічною послідовною реалізацією.Проте швидкість виконання програми зросла.Тому потрібно враховувати особливості конфігурації комп'ютера та конкретного алгоритму.

Література

[1] Химич А.Н. Параллельныеалгоритмырешения задач вычислительной математики / А.Н. Химич,И.Н. Молчанов,А.В. Попов,Т.В. Чистякова,М.Ф. Яковлев–Киев, 2008. 129-149 с.

[2] ДжеффриРихтер. CLR via C#. Программирование на платформе Microsoft NET Framework 4.5 на языке C# / ДжеффриРихтер– Питер, 2013. 747 с.

[3] Шахно С.М. Практикум з чисельних методів / С.М. Шахно, А.Т. Дудикевич, С.М. Левицька – Львів, 2013. 216 с.

[4] Jean-MarcAdamo. Multi-ThreadedObject-Oriented MPI-BasedMessagePassingInterface / Jean-MarcAdamo– USA, 1998.

[5] Герасимова Т.О. Деякі підходи до розробки паралельних алгоритмів розв’язування задач на комп’ютерах з процесорами IntelXeonPhi / Т.О Герасимова, А.Н. Нестеренко – Київ, 2018.

Розв'язування систем нелінійних рівнянь із застосуванням матричних ланцюгових дробів

Чернікова Анастасія Сергіївна, Україна

Львівський національний університет ім. Івана Франка

Факультет прикладної математики та інформатики

chernikovanastia28@gmail.com

Вступ

Теорія ланцюгових дробів вивчає алгоритми, які є одними із найважливіших інструментів для математичного аналізу, теорії ймовірностей та теорії чисел.

Застосування ланцюгових дробів зустрічається при розробці сонячного календаря, для доведення ірраціональності чисел (наприклад, за допомогою ланцюгових дробів доведена ірраціональність дзета-функції Рімана $\zeta(3)$ числа π), в алгоритмі Ланцоша, який використовує ланцюгові дроби для обчислення власних значень великих розріджених матриць, в деяких алгоритмах факторизації і тд.

Отже, актуальність дослідження та використання ланцюгових дробів в чисельних методах є очевидною. В даній роботі ми розглянемо метод розв'язування систем нелінійних рівнянь за допомогою матричних ланцюгових дробів.

Постановка задачі

Нехай маємо систему з n нелінійних рівнянь з n невідомими:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0. \end{cases} \tag{1}$$

Нехай функції n змінних $y_i = f_i(x_1, x_2, \dots, n)$, $(i = \overline{1, n})$ визначені і неперервні разом з усіма своїми частинними похідними до другого порядку включно в деякому δ-околі точки $x^0 = f_i(x_1^0, x_2^0, \dots, x_n^0)$.

Припустимо, що $x_1^k, x_2^k, \dots, x_n^k,\ k = (0,1, \dots)$ – наближені корені системи рівнянь (1). Нехай для k-го кроку, тобто наближень $x_1^k, x_2^k, \dots, x_n^k$, рівняння системи (1) не

виконується. Тоді будемо шукати поправки h_j, ($j = \overline{1,n}$), при яких у результаті підстановки $x_j^{(k+1)} = x_j^k + h_j$ мають виконуватися рівняння:

$$\begin{cases} f_1(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = 0, \\ f_2(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = 0, \\ \vdots \\ f_n(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = 0. \end{cases} \tag{2}$$

Запишемо формулу Тейлора для функції n змінних таким чином:

$$f_i(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = f_i(x_1, x_2, \dots, x_n) +$$

$$+ \sum_{j=1}^{n} \frac{\partial f_i(x_1^k, x_2^k, \dots, x_n^k)}{\partial x_j} h_j + \frac{1}{2!}\left(\sum_{j=1}^{n} \sum_{q=1}^{n} \frac{\partial^2 f_i(x_1^k, x_2^k, \dots, x_n^k)}{\partial x_j \partial x_q} h_j\, h_q \right) + o(h^2), \tag{3}$$

$$h = \sqrt{h_1 + h_2, \dots, h_n}.$$

Тоді на основі системи (2) та формули Тейлора (3) для визначення поправок $h_1, h_2, \dots, h_n$ з точністю $o(h^2)$ можемо записати наступну систему рівнянь:

$$f_i(x_1, x_2, \dots, x_n) + \sum_{j=1}^{n} \frac{\partial f_i(x_1^k, x_2^k, \dots, x_n^k)}{\partial x_j} h_j +$$

$$+ \frac{1}{2!}\left(\sum_{j=1}^{n} \sum_{q=1}^{n} \frac{\partial^2 f_i(x_1^k, x_2^k, \dots, x_n^k)}{\partial x_j \partial x_q} h_j\, h_q \right) + o(h^2) = 0. \tag{4}$$

Таким чином ми отримуємо метод розв'язування систем алгебраїчних рівнянь другого порядку з багатьма невідомими. Суть даного методу полягає у використанні періодичних матричних ланцюгових дробів для обчислення розв'язків вихідної системи шляхом певних елементарних перетворень, що зводять її до деякого рекурентного співвідношення. Застосуємо цей підхід для (4).

Система (4) еквівалентна наступному:

$$\left(\frac{1}{2} \frac{\partial^2 f_i(x_1, \dots, x_n)}{\partial x_1^2} h_1 + \dots + \frac{1}{2} \frac{\partial^2 f_i(x_1, \dots, x_n)}{\partial x_1 \partial x_n} h_n + \frac{\partial f_i(x_1, \dots, x_n)}{\partial x_1} \right) h_1 + \dots +$$

$$+ \left(\frac{1}{2} \frac{\partial^2 f_i(x_1, \dots, x_n)}{\partial x_n^2} h_n + \dots + \frac{1}{2} \frac{\partial^2 f_i(x_1, \dots, x_n)}{\partial x_n \partial x_1} h_1 + \frac{\partial f_i(x_1, \dots, x_n)}{\partial x_n} \right) h_n = -f_i(x_1, \dots, x_n), \tag{5}$$

де $i = \overline{1,n}$.

Система (5) у матричному вигляді буде такою:

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} = \begin{pmatrix} -f_1\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \\ \vdots \\ -f_n\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \end{pmatrix}, \tag{6}$$

де $a_{in} = \frac{\partial^2 f_i(x_1,\dots, x_n)}{\partial x_n^2} h_n + \ldots + \frac{1}{2}\frac{\partial^2 f_i(x_1,\dots,x_n)}{\partial x_n \partial x_1} h_1 + \frac{\partial f_i(x_1,\dots,x_n)}{\partial x_n}$.

Введемо такі позначення:

$$A_1 = \frac{1}{2}\begin{pmatrix} \frac{\partial^2 f_1(x_1, \dots, x_n)}{\partial x_1^2} & \frac{\partial^2 f_1(x_1, \dots, x_n)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f_n(x_1, \dots, x_n)}{\partial x_1^2} & \frac{\partial^2 f_n(x_1, \dots, x_n)}{\partial x_1 \partial x_n} \end{pmatrix}, A_n = \frac{1}{2}\begin{pmatrix} \frac{\partial^2 f_1(x_1, \dots, x_n)}{\partial x_n^2} & \frac{\partial^2 f_1(x_1, \dots, x_n)}{\partial x_{n-1} \partial x_n} \\ \frac{\partial^2 f_n(x_1, \dots, x_n)}{\partial x_n^2} & \frac{\partial^2 f_n(x_1, \dots, x_n)}{\partial x_{n-1} \partial x_n} \end{pmatrix},$$

$$A_{n+1} = \begin{pmatrix} \frac{\partial f_1(x_1, \dots, x_n)}{\partial x_1} & \frac{\partial f_1(x_1, \dots, x_n)}{\partial x_n} \\ \frac{\partial f_n(x_1, \dots, x_n)}{\partial x_1} & \frac{\partial f_n(x_1, \dots, x_n)}{\partial x_n} \end{pmatrix}, b = \begin{pmatrix} -f_1\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \\ -f_n\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \end{pmatrix}. \tag{7}$$

Тоді (6) можна записати таким чином:

$$(A_1 h_1 + \ldots + A_n h_n + A_{n+1})(h_1, \dots, h_n)^T = b. \tag{8}$$

Запис (8) можна подати у компактній формі Прінсгейма:

$$\left(h_{1,\dots}, h_n\right)^T = \frac{b|}{|A_{n+1}} + (A_1, \dots, A_n) \circ \frac{b|}{|A_{n+1}} + (A_1, \dots, A_n) \circ \frac{b|}{|A_{n+1}} + \cdots \tag{9}$$

Таким чином, можемо записати рекурентну формулу для обчислення наближеного розв’язку системи (1) за допомогою матричних ланцюгових дробів:

$$\begin{pmatrix} h_1^{k+1} \\ \vdots \\ h_n^{k+1} \end{pmatrix} = -\left((A_1, \dots, A_n) \circ \begin{pmatrix} h_1^k \\ \vdots \\ h_n^k \end{pmatrix} + A_{n+1} \right)^{-1} \begin{pmatrix} f_1\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \\ \vdots \\ f_2\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \end{pmatrix}. \tag{10}$$

Чисельні експерименти

Програмна реалізація втілена на мові Python за допомогою Jupyter Notebook. Розглянемо наступну систему з двох нелінійних рівнянь:

$$\begin{cases} 2x_1^2 + x_2^2 - 1 = 0, \\ x_1^3 + 6x_1^2 x_2 - 1 = 0. \end{cases} \quad (11)$$

Запишемо A_1, A_2, A_3 та b за формулами (7) для цього прикладу:

$$A_1 = \frac{1}{2}\begin{pmatrix} 4 & 0 \\ 6x_1 + 12x_2 & 12x_1 \end{pmatrix}, A_2 = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & -12x_1 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 4x_1 & 2x_2 \\ 3x_1^2 + 12x_1x_2 & 6x_1^2 \end{pmatrix}, b = \begin{pmatrix} 2x_1^2 + x_2^2 - 1 \\ x_1^3 + 6x_1^2 x_2 - 1 \end{pmatrix}.$$

Далі за формулою (10) знайдемо наближені роз'язки для системи (11). Наведемо результати у вигляді таблиці, де $k -$ кількість ітерацій, $\varepsilon -$ задана точність, $x_1^{(k)}/x_2^{(k)} -$ знайдені розв'язки системи. Задамо наступні початкові наближення:

- $x_1^{(0)} = 0.6$ та $x_2^{(0)} = 1.5$;
- $h_1^{(0)} = 0.0$ та $h_2^{(0)} = 0.0$.

Табл 1. Результати обчислень для системи (11)

k	$x_1^{(k)}/x_2^{(k)}$	ε	Нев'язка
2	0.4491674244024886 0.8342330321728539	0.1	-0.25195655
5	0.44269004285276614 0.7853802694932488	0.01	-0.01045357
8	0.4419021490306732 0.7811647823095133	0.001	-0.00100671
10	0.44173941639826747 0.7809305806178282	0.0001	-0.00019963
15	0.4416309873221669 0.7809705942597531	0.00001	1.12960787e-06
22	0.4416157889075747 0.780993056235112	0.000001	3.97669291e-07

Висновки

У цьому дослідженні розглянуто перспективний метод розв'язування систем нелінійних рівнянь з n невідомими. Також були проведені чисельні експерименти, які детальніше будуть висвітлені у доповіді.

Література

[1] Скоробогатько В.Я. Теория ветвящихся цепных дробей и ее применение в вычислительной математике / В.Я. Скоробогатько. – Москва: Наука, 1983. – 311 с.

[2] Недашковська А.М. Ітераційний метод розв'язування системи поліноміальних рівнянь другого степеня / А.М. Недашковська // Фізико-математичне моделювання та інформаційні технології. – 2015. – № 21. – С. 150-161.

[3] Шмойлов В.И. Решение систем линейных алгебраических уравнений непрерывными дробями / В.И. Шмойлов, Я.С. Коровин. – Ростов–на–Дону: Изд-во ЮФУ, 2017. – 383 с.

[4] Арнольд В.И. Цепные дроби / В.И. Арнольд // Московский центр непрерывного математического образования, Библиотека \textquotedblleft Математическое просвещение\textquotedblright. – 2001. – № 14. – 40 с.

[5] Хинчин А.Я. Цепные дроби / А.Я. Хинчин. – Москва: Физматлит, 1960. – 112 с.

[6] Шахно С.М. Практикум з чисельних методів / С.М. Шахно, А.Т. Дудикевич, С.М. Левицька. – Львів: ЛНУ імені Івана Франка, 2013. – 216 с.

[7] Kress R. Numerical Analysis / R. Kress. – New-York: Springer-Verlag, 1998. – 326 с.

Estimation of Fatality of a classical SIR model: A case study of COVID-19 data in Lviv

Evans Teiko Tetteh

Ivan Franko National University of Lviv/ University of L'Aquila

evans.tetteh@intermaths.eu

Introduction

The novel coronavirus disease also known as the COVID-19 which was discovered in Wuhan China in December, 2019 is a contagious disease caused by severe acute respiratory syndrome coronavirus (SARS-CoV-2). According to medical experts, the symptoms of COVID-19 are variable but commonly often include fever, cough, fatigue, breathing difficulties, and loss of smell and taste. Since its discovery, its spread has increased exponentially across the globe so much so that the World Health Organization (WHO) has on 11th March 2020 declared it as a pandemic [1].

Mathematical models however play a vital role in the study of the dynamics and spread of infectious diseases, of which the spread of COVID-19 is not an exception. The study of the spread of infectious diseases goes way back where Kermack and McKendrick [2] in 1927 develop compartmental models to describe the population dynamics of the outbreak of a disease. These compartment models' main feature is to divide the total population into compartments where individuals move from one compartment to the other over time. The model famously known as the classical SIR model, since its development has had various modifications over the years to explain better specific diseases outbreaks.

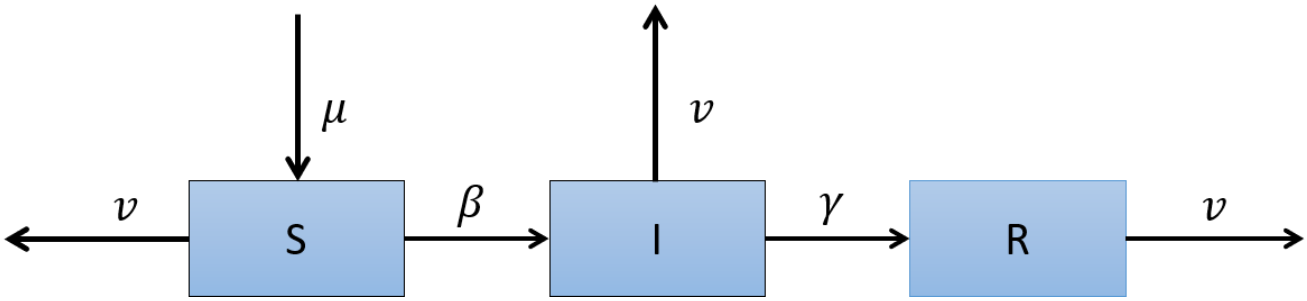


Figure 1: A compartmental diagram for an SIR model.

In this report, we present a variant of the SIR model called SIRD model to model the outbreak of COVID-19 in Lviv. The model uses the notion that the total population is divided in to four (4) compartments namely; Susceptible, Infected, Recovered, Deceased, which depicts that the only modification to the classical SIR model is that the R compartment is further subdivided into two, that is, recovered and deceased individuals.

The mortality rate however is a vital entity to determine how severe a disease spread is and therefore we verify the fatality due to COVID-19 outbreak in Lviv.

The SIRD model

The SIRD model has the following assumptions:

- The total size of host population remains constant.
- The population must mix homogeneously.
- A person can leave the susceptible compartment only by becoming infected.
- A person can leave the infected compartment only by recovering or dying from the disease.
- The probability of being infected does not depend on factors such as age, gender or social status.
- The recovery rate is constant in time.
- The dynamical equations are of first order.

$$\begin{aligned} S' &= -\beta \frac{SI}{N} \\ I' &= \beta \frac{SI}{N} - \gamma I - \alpha I \\ R' &= \gamma I \\ D' &= \alpha I \end{aligned} \tag{1}$$

The system of Ordinary Differential equations above describes the SIRD model.

Where $N = S + I + R + D$ and β, γ, α are positive parameters that shows the infection rate, recovery rate and death rates respectively.

However due to diverse measures taken by the Lviv city administration, health facilities and officials and the general public as a whole, we will consider these parameters as functions of time, t. This is due to the fact that the data time range collected for this report assumes that the parameters change with respect to time over the period. Hence, we consider the following the parameters as follows:

$\beta(t) = a_n t^n$, $\gamma(t) = b_n t^n$, $\alpha(t) = c_n t^n$ where are positive functions.

In fitting the model to the Covid-19 data of Lviv, the least square method of fitting data was employed. In this case we assume initial values of the constants a_n, b_n, c_n for $n = 1,2,3, \ldots$ and solve the system (1) we find the Root Mean Square Error of the solutions and the data. We go on to find optimal parameters that minimizes the Root Mean Square Error by using the MATLAB function *lsqcurvefit.* We also simulate the time varying parameters as polynomial functions to several degrees.

Results and Simulations

The COVID-19 outbreak data in Lviv was collected from https://city-adm.lviv.ua/ which provided daily updates about the progress of the virus; infections, recovery and death cases. The data was collected for a period of 69 days from 24/11/2020 to 01/02/2021.

A total population of $N = 722000$ was considered. Polynomial functions of time varying parameters were chosen as follows: $\beta(t) = a_1 + a_2 t + a_3 t^2$; $\gamma(t) = b_1 + b_2 t$; $\alpha(t) = c_1$ with initial values of constants, $a = [0.01\ 0.001\ 0.0001]$; $b = [0.01\ 0.001]$; $c = [0.01]$

In order to calculate the Case Fatality Ratio (CFR), the death rate α, however is assumed to be constant over the period of the pandemic [3] and therefore we choose α as a constant function. We obtained the following results of optimal values of parameters

$\beta(t) = 0.0593 - (0.00106)t - (1.2320 \times 10^{-5})t^2$;

$\gamma(t) = 0.0281 + (4.5810 \times 10^{-4})t$;

$\alpha = 0.0011269$

with a Root Mean Square Error with respect to the population of each compartment S, I, R, and D as

$RMSE_S = 126.8839$

$RMSE_D = 294.3103$

$RMSE_D = 301.7563$

$RMSE_D = 37.2980$

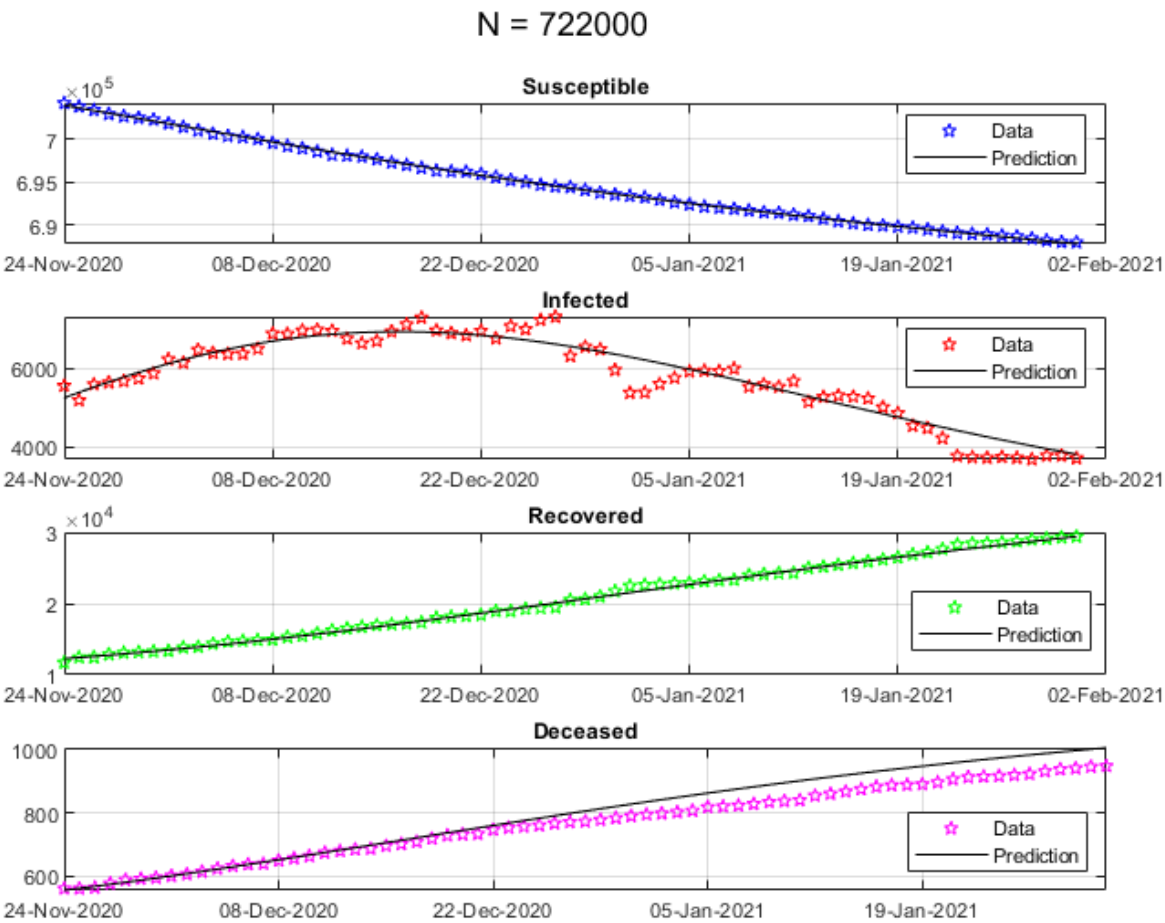


Figure 2: Results of fitting SIR model to COVID-19 data in Lviv with constant death rate.

We calculate the Case Fatality Ratio [3],

$$\text{CFR} = \frac{number\ of\ deaths\ from\ COVID{-}19}{number\ of\ infections\ from\ COVID{-}19} = 0.0013$$

Conclusion

The SIR compartmental was reviewed and a modified version of the model was investigated to explain the outbreak of COVID-19 in Lviv. The model was used to fit the COVID-19 data in Lviv over a time period of 69 days which implied somewhat the first wave of available data.

The results suggested that the death rate between the chosen period was constant. However, the reason for the error could be as a result of the medical being too crowded and therefore most of the death cases were overlooked and not recorded. Moreover, the

data did not explain the age difference in the death cases and so not so much meaning could be gotten from that perspective as stated in literature [4] that most aged people vulnerable to die from the virus. The CFR depicts the probability of dying from COVID-19 in Lviv once infected on the assumption that all cases are reported (which may not be realistic).

Further studies could be done to verify the death cases caused by the virus as opposed to other causes of death in previous years to verify the authenticity of the data.

References

[1]. WHO. WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020. [Accessed 25/04/2021]

[2]. Our World in data. Mortality Risk of COVID-19. https://ourworldindata.org/mortality-risk-covid, 2021. [Accessed 25/04/2021].

[3]. WHO. Estimating mortality from COVID-19. Scientific Brief. 4-August-2020. https://www.who.int/news-room/commentaries/detail/estimating-mortality-from-covid-19

[4]. Maia Martcheva. An Introduction to Mathematical Epidemiology. Springer Publishing Company, Incorporated, 1st edition, 2015.

[5]. Williamson, E.J., Walker, A.J., Bhaskaran, K. *et al.* Factors associated with COVID-19-related death using OpenSAFELY. *Nature* **584,** 430–436 (2020). https://doi.org/10.1038/s41586-020-2521-4

[6]. Magal, P., Webb, G. The parameter identification problem for SIR epidemic models: identifying unreported cases. *J. Math. Biol.* **77,** 1629–1648 (2018). https://doi.org/10.1007/s00285-017-1203-9

Solving vehicle routing problem using ant colony optimization

Grace Abena Safoa Amackson

Ivan Franko National University of Lviv

grace.amackson@intermaths.eu

Introduction

The concept of the vehicle routing problem (VRP) is that given several vehicles at a depot the vehicles are to find the shortest route to visit a set of customers exactly once. It is described as non-deterministic polynomial NP-hard because it has no quick solution and the complexity of calculating the best route will increase when you add more destinations to the problem.

It has many direct applications in transportation, scheduling, network flow etc. The aim of the VRP is to minimize the cost of travel by reducing the travel distance of the vehicles.

Different studies are being done on VRP and its different variants applied, some considered time window, multi depots, pickups and delivery or a combination of the variants.

Materials and Methods

Vehicle routing can be solved with preferably metaheuristic. Metaheuristic is a higher-level procedure intended to escape from local optimal solutions. Different methods like nearest neighborhood or genetic algorithm can be used but this paper examines ant colony optimization.

Ant colony optimization (ACO) is one of the best methods for finding the shortest route and many other optimization problems. According to (Dorigo.M., Maniezzo, & Colorni, 1996), the concept of ant colony optimization is based on the study of the behavior of real ants when they use a chemical called pheromones to move back and forth from their food source to their habitat in search for food. Ant moves from their nest in search of food and leave pheromones on the way as means of communication to other ants.

How the algorithm works to construct a solution for the problem:

Given a graph $G = (V, E)$ with V being set of vertices as the customers and E being (i, j) the edge from customer i to j. Each vehicle starts from a randomly selected customer and moves from customer to customer. At each construction step, a vehicle chooses probabilistically from current customer (i) the next customer(j) to visit among a set of unvisited customers using equation 1. The probability of selecting the next node is given as: $P_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha}\eta_{il}^{\beta}}, if\ j \in N_i^k$ (1)

otherwise, probability is 0, where N_i^k denotes the set of unvisited cities, α and β are positive parameters to control the relative weight of pheromone information τ_{ij} and heuristic information η_{ij} (the inverse of the distance from point i to j).
Once all the vehicles have completed their tour, the pheromone trial is updated by τ_{ij}. $\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k \in K} \Delta\tau_{ij}{}^{k}$ where $\rho \in [0,1]$ is the evaporation rate and $\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited on edge (i, j) by vehicle k. This procedure is repeated until a termination criterion is satisfied and a solution is obtained.

Results

A code written in Python under a hundred iterations was used to generate the shortest distance between the nodes, assuming there is a path between all nodes.
One of the relevant factors to consider in a vehicle routing problem is reducing cost by reducing travel distance. In this paper, a simulation on several nodes or customers was done, using ACO to obtain optimal routes from different starting points. Hence the shortest path was found between all the nodes.

References

Nacima Labadie, Christian Prins and Caroline Prodhon.(2016). Metaheuristics for Vehicle Routing Problems. (Vol 3).

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.

Yang, J., Shi, X., Marchese, M., and Liang, Y. (2008). An ant colony optimization method for generalized tsp problem. Progress in Natural Science, 18(11):1417–1422. Procedia Computer Science, 76:534 – 538.

Two dimension advection-diffusion model:

Elizabeth Adomako Boamah

Ivan Franko National University of Lviv/University of L'Aquila

Faculty of Applied Mathematics and Informatics/DISIM

elizabeth.boamah@intermaths.eu

Introduction

A mathematical model describing the spread of spilled pollutant (example oil) through the soil is discussed. A chemical into flowing water diffuses and at the same time carried by the flow. This spread is described by a two-dimensional equation.

Advection-diffusion equation is a combination of advection and diffusion equations which describes the transport of solute in a medium (physical system). For example, put a chemical into flowing water and it diffuses and at the same time carried by the flow. The analytic/ numerical solution of the advection-diffusion equation has many applications which include reservoir flow, transport of pollutants in air and groundwater, flow through porous media and thermal pollution in river systems. Along with its boundary and initial conditions we are able to understand the how the concentration of the pollutant is distributed and its behavior through open mediums and porous medium.

In this study, the flow in the vertical direction down to the water table, under the influence of gravity is considered. The model is restricted to two-dimensional advection-diffusion equation. The model is fully discretized using the Finite element method.

1. Problem statement

In urban areas, the cases in which spilled oil covers a flat surface are a lot hence a rectangular coordinate system (x, y) is used to model this flow.

A two-dimensional model (strong form) which simulates pollutant transport is considered.

$$\frac{\partial C}{\partial t} + v_x \frac{\partial C}{\partial x} + v_y \frac{\partial C}{\partial y} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} \qquad (1.1)$$

The equation describes pollutant diffusion influenced by velocity. The velocity is assumed to be constant and nonzero in only the x-direction. It assumed also that the medium is homogenous which that D_x, D_y does not vary in space. (1.1) reduces to:

$$\frac{\partial C}{\partial t} + v_x \frac{\partial C}{\partial x} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right) \qquad (1.2)$$

Where;

C is the concentration of the pollutant

D is the diffusion coefficient

v_x is the velocity in the x-direction

With boundary (essential and natural) and initial conditions

$C(a, y, t) = 0,\ C(b, y, t) = j(x),\ C_y(x, a, t) = 0,\ C_y(x, b, t) = 0$ and $C(x, y, 0) = g(x)g(y)$

Respectively.

Pollution of groundwater is indeed a vivid application of (1.2). The domain considered in this study is a two-dimensional rectangle which is embedded in a cartesian plane.

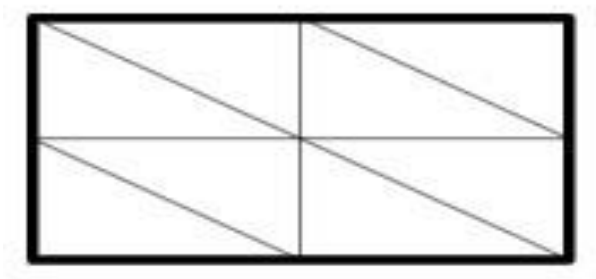


Figure 1: discretized domain illustration

2. Weak Formulation

In other to develop a finite element formulation, the model must be written in an integral form called weak form.

Galerkin finite element method enables to convert differential equation usually in the weak form to a discrete problem by applying some basis functions.

Applying Green's theorem to (1.2) we want to find $v \in H^1(0, T, H^1(\Omega)$ such that

$$\iint C_t v\, dx\, dy + \iint \left[D\left(\frac{dC}{dx}\frac{dv}{dx} + \frac{dC}{dy}\frac{dv}{dy}\right) + v_x \frac{dC}{dx} v \right] dx\, dy \qquad (2.1)$$

Let $L(C, v) = \iint \left[D\left(\frac{dC}{dx}\frac{dv}{dx} + \frac{dC}{dy}\frac{dv}{dy}\right) + v_x \frac{dC}{dx} v \right] dx\, dy$ (2.2)

$$f(C, v) = 0 \qquad (2.3)$$

Now the main task is to find $v \in H^1(0, T, H^1(\Omega))$ such that $(C_t, v) + L(C, v) = 0$ for any
$v \in H_0^1(\Omega)$. Thus, to use finite dimensional space to approximate infinite dimensional space.

3. Spatial and Time Discretization

We set up a linear algebraic system to be able to formulate a solution equation for the finite element.

$$C^n(x, y) = \sum_{j=1}^{n} C_j(t)\varphi_j(x, y) \qquad (3.1)$$

C_j is the unknown variable to be determined
φ is linear basis function

Let $v(x, y) = \varphi_i, (i = 1, \cdots, n_e)$. A linear system of n equations with n unknowns is obtained below

$$\sum_{j=1}^{n_e} \frac{\partial C_j}{\partial t} \int_{\Omega^{e_k}} \varphi_j \varphi_i\, d\Omega + \sum_{j=1}^{n_e} C_j \int_{\Omega^{e_k}} D\nabla\varphi_j \nabla\varphi_i d\Omega + \sum_{j=1}^{n_e} C_j\, v_x \int_{\Omega^{e_k}} \nabla\varphi_j \varphi_i\, d\Omega = 0 \qquad (3.2)$$

n_e = number of local finite element nodes
Ω^{e_k} = domain of the finite element e_k
(3.2) can be rewritten as linear system

$$\dot{C} \cdot M + C \cdot (U + A) = 0 \qquad \text{with}$$

$$m_{ij} = \int_{\Omega^{e_k}} \varphi_j \varphi_i\, d\Omega \qquad (3.3)$$

$$u_{ij} = \int_{\Omega^{e_k}} D\nabla\varphi_j \nabla\varphi_i d\Omega$$

$$a_{ij} = \int_{\Omega^{e_k}} v_x \nabla\varphi_j \varphi_i\, d\Omega$$

m_{ij} , u_{ij} and a_{ij} are mass, diffusion and advection matrices respectively.
The above matrices are assembled using Fenics software.

Time is discretized using the θ −method. This method makes use of both the explicit and implicit methods of solving the system of equations.

$$M\frac{C^{k+1}-C^{k}}{\Delta t}+\ \theta SC^{k+1}+(1-\theta)SC^{k}=0, 0\leq\theta\leq 1 \qquad (3.4)$$

$$S=U+A$$

θ is a numerical parameter that describes the degree of implicitness. The scheme is considered explicit Euler when $\theta=0$, implicit Euler when $\theta=1$ and Crank-Nicholson when $\theta=\frac{1}{2}$

4. Solution Method

The application of finite element method resulted in a system of linear equations.

$$Ax=B \qquad (4.1)$$

Where $A=M+\theta S\Delta t,\ x=C^{k+1}$ and $B=(M-(1-\theta)S\Delta t)C^{k}$.

This above system is solved numerically using Fenics software. A finite element solver software. Results are yet to be obtained.

5. References

Darrell W. Pepper, J. C. (2017). *The Finite Element Method Basic Concepts and Applications with MATLAB®, MAPLE, and COMSOL THIRD EDITION.* CRC Press.

Hemalatha, N. P. (2016). Mathematical model to study. *Applications and Applied*, 351-363.

Pozrikidis, C. (2014). *Introduction to Finite and Spectral Element Methods Using MATLAB, Second Edition.* CRC Press, Boca Raton.

Трикроковий комбінований метод хорд та Ньютона для розв'язування систем з недиференційованим оператором

Дмитренко Володимир, Ковальчук Ольга

Україна, Львів

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

volodymyr.dmytrenko@lnu.edu.ua

Анотація

Нами буде запропоновано трикроковий метод для розв'язування систем нелінійних рівнянь з недиференційованим оператором. Цей метод є комбінацією методу Ньютона та хорд

Вступ

Більш ніж 300 років тому в 1669р.[6] був запропонований метод Ньютона для розв'язування систем нелінійних рівнянь. Даний метод і досі актуальним і часто використовується для розв'язування різних задач, де потрібно знаходити розв'язки системи нелінійних рівнянь. Також він є основою для низки інших методів та модифікацій. Проте окрім позитивних сторін має також і недоліки. Основним недоліком є те, що на кожному кроці методу потрібно обчислювати матрицю Якобі. Однак бувають випадки, коли система нелінійних рівнянь є недиференційована, тому застосування методу, який використовує похідні є неможливим.

Постановка задачі

Задано систему:

$$P(x) = 0 \tag{1}$$

де $P : R^n \to R^n$

Відомий метод Ньютона задається наступним чином:

$$x_{k+1} = x_k - P'(x_k)^{-1}P(x_k) \qquad (2)$$

Даний метод використовує Матрицю Якобі .

Розглянемо викпадок, коли $P(x_k)$, задається наступним чином:

$$P(x_k) = F(x_k) + G(x_k) = 0$$

(3)

де F і G - неперервні функції, F - диференційовна, G - недиференційовна його частини

У випадку (3) була запропонована наступна модифікація методу Ньютона:

$$u_k = x_k - [A(x_k)]^{-1}P(x_k),$$

$$v_k = x_k - [A(x_k)]^{T}P(x_k),$$

$$x_{k+1} = \arg\min\left\|P(u_k - \gamma(u_k - v_k))\right\| \qquad (4)$$

де $A(x_k) = F'(x_k) + G(x_{k-1,}x_k)$

Матриця $G(x_{k-1}, x_k)$ - поділена різниця, що задовольняє умову:

$$G(x_{k-1}, x_k)(x_{k-1} - x_k) = G(x_{k-1}) - G(x_k) \qquad (5)$$

x_{-1}і x_0 - задані початкові наближення.

Перевага цього методу є у тому, що він має майже таку саму кількість, обчислень, як і метод Ньютона. Даний метод комбінує в собі обчислення похідних для диференційовної частини рівняння, та використовує формулу поділеної різниці для недиференційованої частини, що дає можливість розв'язувати системи нелінійних рівнянь з недиференційовним оператором.

Чисельні результати

Приклад 1

$$x^2 - y + 1 + \frac{|x-1|}{9} = 0,$$

$$y^2 + x - 7 + \frac{|y|}{9} = 0.$$

Розв'язком системи є x*=(1.15936; 2.36182).

Виберемо початкове наближення (0; 0).

n	x□	y□	‖x□-y□‖
1	1.660855	2.300371	2.837278
2	1.213140	2.351301	0.450602
3	1.160459	2.361659	0.001105
4	1.159366	2.361824	5.38167893e-6
5	1.159361	2.361824	2.33457126e-9
6	1.159361	2.361824	6.087627326e-12

Приклад 2

$3x^2y + y^2 - 1 + |x - 1| = 0$,

$y^4 + xy^3 - 1 + |y| = 0.$

Розв'язком системи є x*=(1.15936; 2.36182).

Початкове наближення (0.5, 0.5).

n	x□	y□	‖x□-y□‖
1	0.874583	0.386700	0.386700
2	0.902008	0.331289	0.008796
3	0.894424	0.326835	0.001066
4	0.894098	0.327722	0.000117
5	0.894652	0.327832	6.50556578e-6
6	0.894655	0.327827	6.06244664e-7
7	0.894655	0.327826	1.00146575e-7
8	0.894655	0.327826	2.166454666e-8
9	0.894655	0.327826	5.681566557e-9

Література

1. I.K. Argyros. The Secant method and lixed points of nonlinear operators// I.K. Argyros/ Monatshefte für Mathematik. 106 - 1988 – p.85-94

2. Myhaylo Bartish. On some iterative method for solving nonlinear equations// Myhaylo Bartish, Olha Kovalchuk , Ogorodnyk N./ III International scientific and practical conference "Modern problems of mathematical modeling, automated control and information technologies" (IT conf -2019) Rivne, Ukraine, November 14-16, 2019 - p. 7-8

3. Catinas E. On some iterative methods for solving nonlinear equations//Catinas E. / Rev D'Analy. Numer. Theorie L'Approx. – 23 - (1994) - p. 47–53

4. M. A. Hernandez. Modification of Newton's method for nondi'erentiable// M.A. Hernandez M. J. Rubio /A Journal of Computational and Applied Mathematics 164–165 - (2004) - p.409–417

5. M. A. Hernandez. The Secant method for Nondifferentiable// M. A. Hernandez, M. J. Rubio / Applied Mathematics Letters - 15 - (2005) - p. 395-399

6. T. Yamamoto. Historical developments in convergence analysis forNewton's and Newton-like methods// T. Yamamoto / Journal of Computational and Applied Mathermatics - 124 - 2000- p. 1-23

7. Stepan Shakhno. Two-step method for solving nonlinear equations with nondifferentiable operator// Stepan Shakhno, Halina Yarmola / Журнал обчислювальної та прикладнох математики - 2012 -№3(109), с.105-115

Порівняльна характеристика однокрокових та двокрокових методів хорд та Курчатова розв'язування нелінійних систем рівнянь

Войтович Наталія

Україна, Львів

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

natalia.voytovuch@gmail.com

Теоретичні відомості

Нехай задано рівняння

$$F(x) = 0, \tag{1}$$

де F - неперервний оператор, заданий на непорожній опуклій підмножині D банахового простору X зі значенням в банаховому просторі Y. У цій роботі ми вивчатимемо проблему апроксимації локального розв'язку x^* нелінійного рівняння (1).

Метод хорд є одним з найвикористовуваніших ітераційних методів розв'язування нелінійних рівнянь. Метод хорд кращий за метод Ньютона у випадку скалярних рівнянь, з точки зору індексу ефективності. Метод хорд призначений для розв'язання рівняння (1) в банахових просторах. Однокроковий метод хорд описується наступним алгоритмом [1]:

$$x_{n+1} = x_n - F(x_{n-1}, x_n)^{-1} F(x_n), \tag{2}$$

де $n \geq 0$, x_0, $x_{-1} \in D$ початкові точки і $F(x,y) \in \mathrm{L}(X,Y)$ $(x,y \in D)$ поділена різниця від F.

На основі однокрокового методу можна розглянути двокрокову модифікацію методу типу хорд [1]:

$$x_{n+1} = x_n - F(u_n, v_n)^{-1} F(x_n), \tag{3}$$

$$y_{n+1} = x_{n+1} - F(u_n, v_n)^{-1} F(x_{n+1}), n = 0,1, \ldots,$$

де $u_n = x_n + a_n(y_n - x_n), v_n = x_n + b_n(y_n - x_n), a \in [-1; 1], b_n \in [0; 1]$,

$F(u_n; v_n)$ – поділена різниця першого порядку оператора F за точками u_n, v_n.

На одній ітерації треба один раз обчислювати матрицю поділених різниць, як і в методі (2), розв‘язувати дві системи лінійних рівнянь, але з однаковою матрицею. Враховуючи, що у цьому випадку прямий хід методу Гауса (чи LU-розклад) виконується один раз, то кількість обчислень на ітерації зростає несуттєво.

Наступний ітераційний метод лінійної інтерполяції, запропонований В. А. Курчатовим, є менш дослідженим. Так само, як і метод хорд (1), використовуючи два попередній наближення, метод Курчатова має квадратичну швидкість збіжності.

Однокроковий метод Курчатова описується наступним алгоритмом [5]:

$$x_{n+1} = x_n - F(2x_n - x_{n-1}, x_{n-1})^{-1} F(x_n), \quad n = 0,1, \ldots . \tag{4}$$

Його модифікація, двокроковий метод типу Курчатова випливає з методу (3). Прийнявши в двокроковому методі типу хорд (3) $a_n = 1$, $b_n = -1$, отримаємо наступний двокроковий метод Курчатова [2]:

$$\begin{gathered} x_0, x_{-1} \in D, \\ y_n = x_n - F(x_{n-1}, 2x_n - x_{n-1})^{-1} F(x_n), \quad n \geq 0, \\ x_{n+1} = y_n - F(x_{n-1}, 2x_n - x_{n-1})^{-1} F(y_n). \end{gathered} \tag{5}$$

Чисельні експерименти

Приклад [2]

Розглянемо систему нелінійних рівнянь:

$$\begin{cases} x_1^2 - x_2 + 1 + \frac{1}{9}|x_1 - 1| = 0, \\ x_1 + x_2^2 - 7 + \frac{1}{9}|x_2| = 0. \end{cases} \tag{6}$$

Позначивши

$x = (x_1, x_2)$, $F_1(x_1, x_2) = x_1^2 - x_2 + 1 + \frac{1}{9}|x_1 - 1|$ і $F_2(x_1, x_2) = x_1 + +x_2^2 - 7 + \frac{1}{9}|x_2|$, систему (6) можна подати як $F(x) = 0$, де $F: R^2 \to R^2$, $F = (F_1, F_2)$.

Таблиця 1 Похибка ($\|x^* - x_n\|$)

n	однкр. м-д хорд	двкр. м-д хорд	однкр. м-д Курчатова	двкр.м-д Курчатова
2	1.052863×10^{-1}	1.9686124×10^{-1}	3.7865932×10^{-1}	5.9068126×10^{-2}
4	1.956025×10^{-1}	1.3644347×10^{-2}	8.9168446×10^{-4}	3.012322×10^{-8}
6	2.2510483×10^{-2}	8.570329×10^{-4}	1.489175752×10^{-6}	–
8	5.0130697×10^{-3}	4.845759×10^{-5}	2.563843×10^{-9}	–
10	2.881095×10^{-4}	$2.4688999067\times10^{-6}$	–	–
12	4.45521059×10^{-5}	1.35831151×10^{-7}	–	–
14	$5.5411663057\times10^{-6}$	5.388431×10^{-9}	–	–
16	8.785201143×10^{-7}	–	–	–
18	$2.36028796662\times10^{-9}$	–	–	–

Обчислимо поділену різницю першого порядку оператора F:

$$F(u,v) = \begin{bmatrix} \dfrac{u_1^2 - v_1^2}{u_1 - v_1} & -1 \\ 1 & \dfrac{u_2^2 - v_2^2}{u_2 - v_2} \end{bmatrix} + \frac{1}{9}\begin{bmatrix} \dfrac{|u_1 - v_1| - |v_1 - 1|}{u_1 - v_1} & 0 \\ 0 & \dfrac{|u_2| - |v_2|}{u_2 - v_2} \end{bmatrix},$$

де $u = (u_1, u_2), v = (v_1, v_2)$. Далі приймаємо норму «максимум» як векторну норму та норму матриці:

$$||F(x,y) - F(u,v)|| \le \big||x - u|\big| + \big||y - v|\big| + \frac{2}{9}.$$

$\mathrm{M} = 1$, $\mathrm{N} = \frac{2}{9}$.

Застосуємо двокроковий метод Курчатова (5) для наближення розв‘язку системи (6). Вибираємо початкове наближення z_1 = (4.2,6.1) і z_0= (5,3.2). Через дві ітерації отримаємо z_1 = (1.21842,2.35088), z_2 = (1.15935,2.36182) і ρ = 0.05907. Взявши x_{-1} = z_1 і x_0 = z_2, отримаємо ξ = 0.00003..., ϱ = 0.45771..., γ = 0.00001....

З рівняння $\left(1+\frac{C_1(1-C_3(\xi))}{1-(C_2(\xi)+C_3(x_i))}\right)\gamma-\xi+\rho=0$, де $C_1 = \varrho(N+M(\gamma+2\rho))$, $C_2(\xi)=\varrho(N+M(\xi+\gamma+2\rho))$, $C_3(\xi)=\varrho(N+2M(\xi+\rho))$, $\gamma=\xi\varrho$,

отримаємо R = 0.05908..., звідси C_1 = 0.15579..., $C_2(R)$ = 0.18284...,

$C_3(R)$ = 0.20987..., а також

$C_2(R) + C_3(R)$ = 0.39272...< 1, *і* $C_1 + C_2(R) + C_3(R) - C_1C_3(R)$ = 0.51581...< 1.

Отже, розв‘язок системи (6) x^* = (1.15936085...,2.361824342...).

В таблиці 1 обчислюємо абсолютну похибку $\|x^* - x_n\|$ для однокрокових та двокрокових методів хорд та Курчатова з допустимою похибкою 10^{-8}.

Видно, що двокроковий метод Курчатова збігається швидше ніж три інші наведені методи.

Results

Method	I	‖X*-X_n‖	CEI	TF
Secant method (I)	18	2.36028796662E-9	1.000154316	64805.1
Kurchatov's method (I)	8	2.56384330024E-9	1.000212314	47101.4
Secant method (II)	14	5.38843193711E-9	1.000043011	43011,8
Kurchatov's method (II)	4	3.01232241566E-8	1.000011353	11352.9

Рис. 1. Результат роботи програми. Порівняння ефективності методів для початкового наближення z_1 = (4.2,6.1) і z_0= (5,3.2).

Results

Method	I	‖X*-X_n‖	CEI	TF
Secant method (I)	23	4.56600321791E-8	1.00015887	52611.9
Kurchatov's method (I)	17	3.91157017215E-9	1.00054412	30188.3
Secant method (II)	15	4.11854320117E-8	1.00007113	49441.8
Kurchatov's method (II)	9	2.99730971083E-8	1.00004417	28149.6

Рис. 2. Результат роботи програми. Порівняння ефективності методів для початкового наближення z_1 = (3,5.8) і z_0= (6.1,4).

На рисунку 1, 2 наведено результати, отримані за однокроковими та двокроковими ітераційними методами хорд та Курчатова. Число I - кількість ітерацій, $\|x^* - x_n\|$ похибка наближення, CEI - індекс обчислювальної ефективності. Для того, щоб легко порівняти ефективність ітеративних методів, ми вводимо нову міру, фактор часу (TF), визначено як $TF = 1/\log(CEI)$.

З рис.1 та рис.2 можна зробити висновок, що метод Курчатова є ефективнішим за метод хорд, а також, що двокрокові методи є набагато ефективнішими з точки зору точності обчислень, часу, кількості ітерацій та кількості оцінок функцій.

Література

[1] *Шахно С. М., Ярмола Г. П.* Про двокроковий метод типу хорд для розв‘язування нелінійних рівнянь / Математичні студії, V.42, No.1, 2014, 84-88

[2] *Himanshu Kumar, Parida P. K.* On semilocal convergence of two step Kurchatov method / International Journal of Computer Mathematics · January, 2018, 110

[3] *Ezquerro A. J., Grau A., Grau-Sanchez M., Hernandez M. A.* On the efficiency of two variants of Kurchatov‘s method for solving nonlinear systems/ Numerical Algorithms, 2010, 685-698

[4] *Argyros K. I., Shakhno S., Yarmola H.* Two-step solver for nonlinear equations / Symmetry, 2019, 128

[5] *Shakhno S.* On a Kurchatov's method of linear interpolation for solving nonlinear equations/ PAMM, 2004, 650-651

Розпізнавання математичних формул на базі даних CROHME

Корольчук Ярослав, Кондор Марта, Мудрик Андрій

Львівський національний університет ім. Івана Франка,
факультет прикладної математики та інформатики

koroltchukjaroslav@gmail.com,
martakondor17@gmail.com,
mudrykandrew@gmail.com

Вступ

Останнім часом системам розпізнавання об'єктів приділяють все більше уваги. Вони досить широко використовуються, але важко створити систему, яка може розпізнавати об’єкти без належних навчальних даних. Складність розпізнавання математичних виразів залежить від багатьох факторів, до прикладу: кількості об’єктів, набору доступних символів, граматики запису.

Точність отриманих результатів є безпосередньо пов'язаною з навчальними даними, тому використовуються методи збільшення навчальної вибірки для покращення їх якості. Була використана архітектура RetinaNet, яка є однокроковим методом. Вона має найкращі показники відносно часу виконання / точності на популярних навчальних даних. Після навчання модель готова до використання. Цю модель можна інтегрувати за допомогою Docker контейнеру.

Головна ідея

Задача полягає у тому, щоб на наборі даних CROHME, навчити детектор розпізнавати математичні символи формули.

Умовно процес можна розділити на 3 частини:

1) формування даних для навчання,
2) навчання та оцінка мережі,
3) імплементація моделі.

1. Скористаємось базою формул CROHME, що поновлюється щорічно та містить деякий базовий набір формул та символів написаних від руки у спеціальному форматі INKML, створеному для цієї вибірки.

Цей формат містить:

- інформацію про розміщення символу на сторінці,
- назву символів формули,
- залежності від інших символів, якщо такі є.

Також додатково він може містити

- інформацію про особу, що писала ці символи (корисно для проблем визначення віку, статі або ж руку, якою символи були написані),
- LaTeX стрічку відповідного символу.

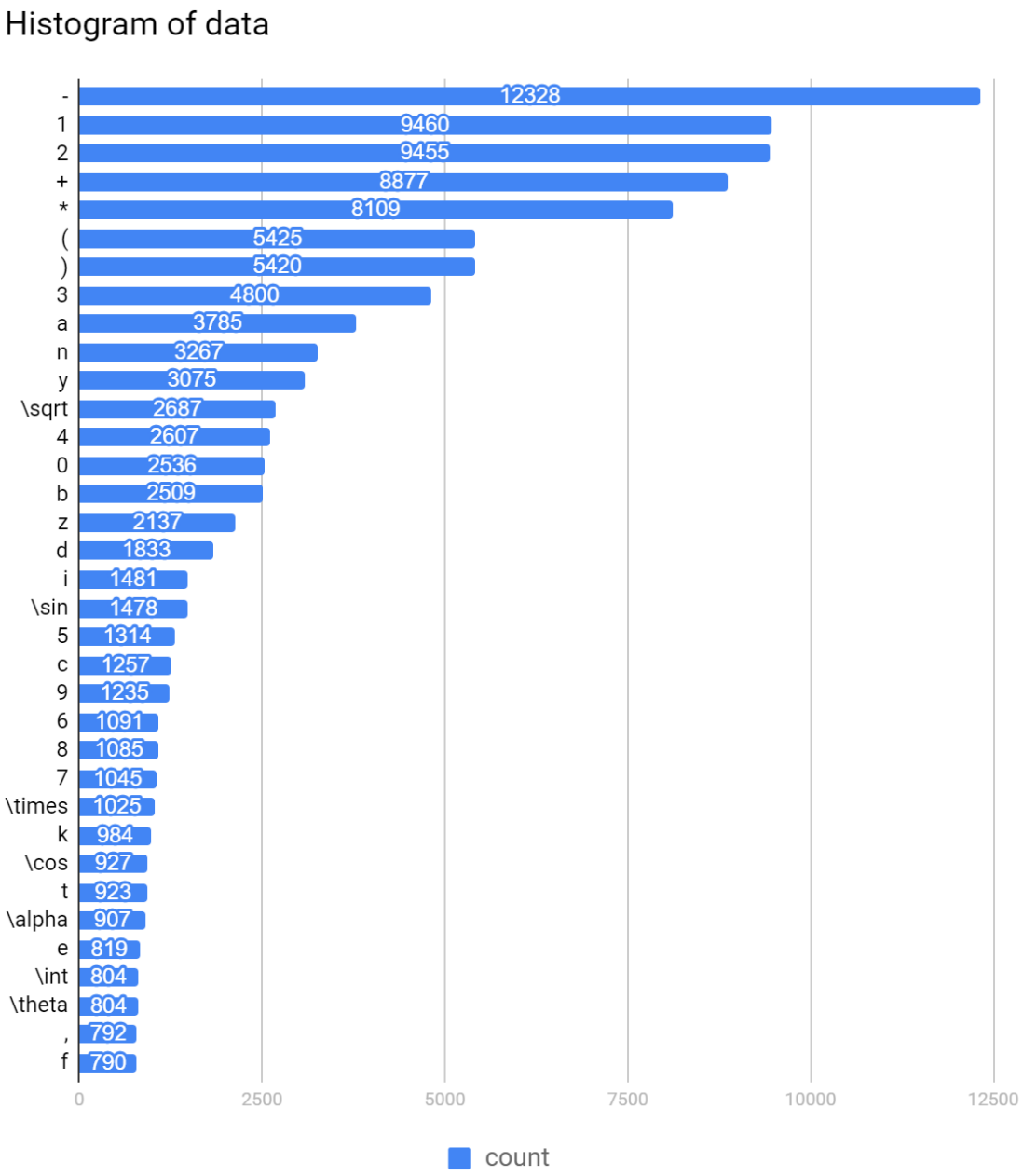


Рис. 1. Розподіл класів CROHME.

Детально дослідивши набір даних, можна виділити символи, що мають малу частоту входження у тренувальну вибірку (рис.1). Тому для формул із ними застосуємо наступні перетворення:

- поворот,
- віддзеркалення,
- обрізування,
- зміну фону.

2. RetinaNet[1] був розроблений Facebook AI Research (FAIR) у 2018 році, і є однокроковим детектором. Якщо порівнювати двокрокові і однокрокові детектори, то всі моделі типу R-CNN включали в себе 2 етапи.

1-ий етап пропонує набір регіонів на фотографії.

2-ий етап класифікує регіон до того чи іншого класу.

Однокроковий детектор пропускає виділення регіонів і напряму знаходить оточуючу рамку конкретного об’єкту.

Focal loss - функція втрат, яка буде використовуватись для тренування моделей. Ця функція призначена для однокрокового сценарію виявлення об’єктів, при якому існує надзвичайний дисбаланс між класами переднього плану та фоном під час навчання.

Також було використано ResNet50[2] як основну частину архітектури RetinaNet.

Для порівняння архітектура була реалізована на бібліотеках TensorFlow[8] і PyTorch[7] на мові програмування Python.

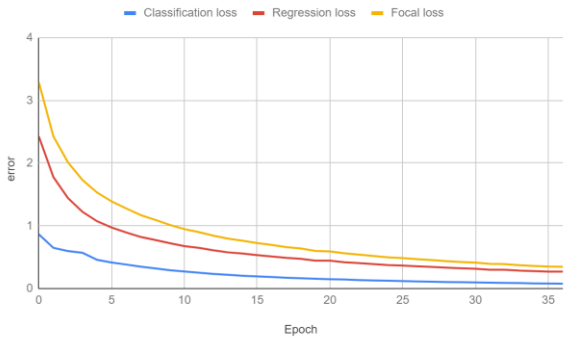


Рис. 2. Результат навчання tensorflow

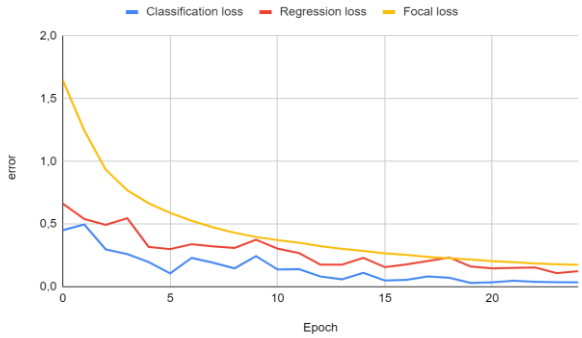


Рис. 3. Результат навчання torch

У таблиці 1 відображені чисельні результати обчислення різних функцій втрат при застосуванні двох бібліотек.

Таблиця 1. Оцінка моделі.

Бібліотека	**Classification loss**	**Regression loss**	**Focal loss**
PyTorch	0,034	0,123	0,174
TensorFlow	0,077	0,261	0,346

Проаналізувавши результати, можемо зробити висновок, що швидкість навчання і краща точність досягається з pytorch, що робить його фаворитом для даної задачі. Через певні особливості паралельного навчання, які є вбудовані у pytorch, і досягається ця різниця.

Час роботи pytorch для однієї картинки складає ~700 мс, так як у tensorflow ~1 с. Для тесту було використаний графічний процесор Nvidia 2070 Super.

Модель може також пристосовуватись до почерку конкретної людини. Для цього необхідно включити приклади з цим почерком до навчальної вибірки.

Висновок

У цій роботі однокрокова мережа була навчена розпізнавати написані від руки математичні вирази, які згодом подавались у вигляді формул LaTeX. Основною проблемою було покращення існуючого набору даних. Також було порівняно результати двох бібліотек для роботи з нейронними мережами.

Було застосовано архітектуру RetinaNet на основі ResNet50. Проте для цього завдання не було порівняно основи для RetinaNet. Також ми не можемо провести порівняння точності результатів, оскільки не було проведене навчання для інших моделей.

На відміну від інших робіт на цю тему, в цій роботі була використана саме однокрокова мережа, що виділяється своєю простотою та швидкістю. Але відомо, що однокрокові мережі поступаються точністю двокроковим. Для того, щоб відслідкувати відхилення в точності потрібно було провести навчання інших моделей.

На основі цього було розроблено додаток на базі мікросервісної архітектури, використовуючи інструменти для розпізнавання та ідентифікації математичного виразу.

Література

1. Focal Loss for Dense Object Detection [Електронний ресурс] / [T. Lin, P. Goyal, R. Girshick та ін.]. – 2018. – Режим доступу до ресурсу: https://arxiv.org/pdf/1708.02002.pdf
2. Deep Residual Learning for Image Recognition [Електронний ресурс] / K.He, X. Zhang, S. Ren, J. Sun. – 2015. – Режим доступу до ресурсу: https://arxiv.org/pdf/1512.03385v1.pdf.
3. Puppeteer [Електронний ресурс] – Режим доступу до ресурсу: https://www.npmjs.com/package/puppeteer-core.
4. CHORME [Електронний ресурс] – Режим доступу до ресурсу: http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions.
5. Шум Гаусса [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Gaussian_noise.
6. AWS Lambda [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/AWS_Lambda.
7. PyTorch [Електронний ресурс] – Режим доступу до ресурсу: https://pytorch.org/.
8. TensorFlow [Електронний ресурс] – Режим доступу до ресурсу: https://www.tensorflow.org/.
9. Data Augmentation [Електронний ресурс] - Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Data_augmentation

Реалізація інструментів симуляції неевклідового простору в прикладних програмних додатках

Іванків Олег
Україна, Львів
Опорний Заклад Загальної Середньої Освіти
«Дублянська загальноосвітня школа І - ІІІ ступенів імені Героя України Анатолія Жаловаги» Жовківської районної ради Львівської області
oleguk0509@gmail.com

Вступ.

У наш час індустрія відеоігор та технології віртуальної реальності гостро потребують нових унікальних ідей. Освітні заклади шукають ефективні способи покращення засобів навчання учнів та студентів. Розробка інструментів для реалізації нових ідей у вигляді проєктів, які застосовують неевклідову геометрію, зможе дати новий поштовх до розвитку індустрії відеоігор, VR та освітніх систем. Використання неевклідового простору дозволяє подолати головну проблему будь-якого пристрою віртуальної реальності – обмеження простору. Неевклідову симуляцію можна використовувати в освітніх цілях, для наглядної демонстрації використання неевклідової математики.

Аналіз предметної області

Аналізуючи програми та наявні plug-in'ів Unity 3D не було знайдено жодної схожої роботи, яка б надавала готові інструменти. Всі інші роботи потребують професійних знань програмування, щоб бути використаними, також жодна з робіт не використовує технологію Unity 3D D.O.T.S. Можна зазначити, що багато робіт пристосовані тільки до однієї або кількох можливих сцен.

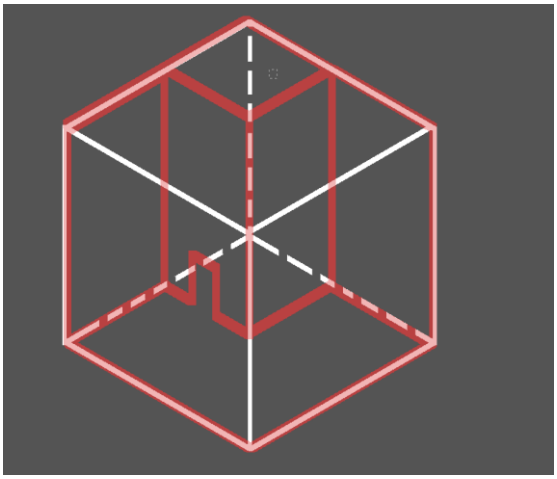
Неевклідова геометрія.

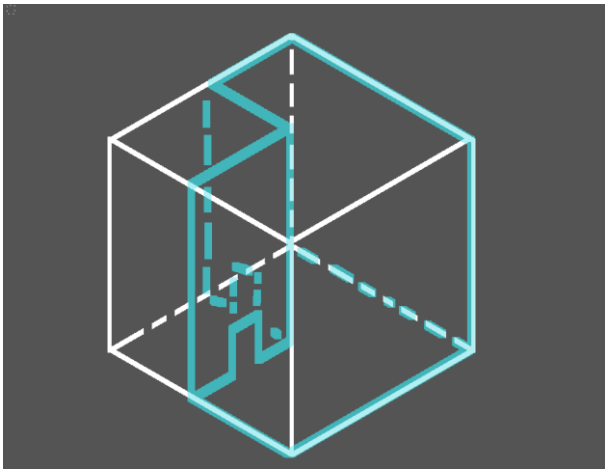
Неевклідова геометрія – це будь-яка геометрична система, яка порушує аксіоми геометрії Евкліда. Під терміном неевклідової геометрії часто вживають більш вузькі напрями: сферична геометрія і гіперболічна геометрія. Найбільш поширеною порушеною аксіомою геометрії Евкліда, яка є однією з формуючих базисів цих геометрій є аксіома паралельності, яка порушується як і в сферичній, так і в гіперболічній геометріях.

Запропонований інструментарій

У інструментарії передбачено два основні типи інструментів: переходи (closeRangeTransition, longRangeTransition) та інструменти кривизни (positiveCurvatureTool, negativeCurvatureTool).

Переходи відповідають за ілюзію зміни позиції об’єктів та користувача в просторі, обидва інструменти – це ніби портали, які створені за допомогою різних технологій (Stencil Buffer, Camera Transition). Інструменти кривизни змінюють



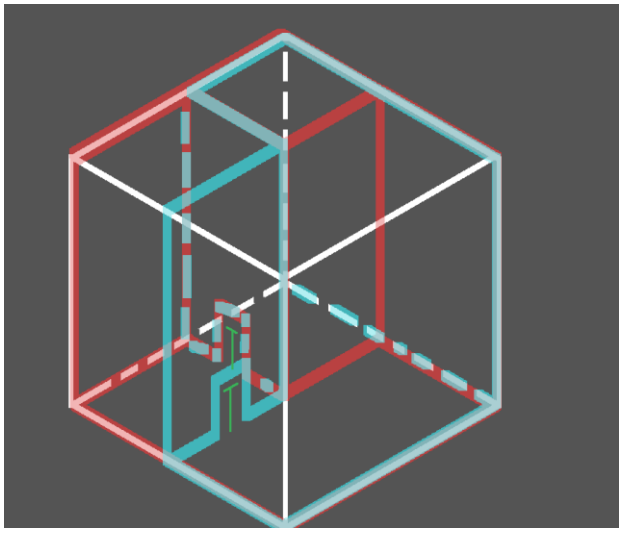


позицію та оберт спираючись на позицію та напрям камери гравця, тим самим імітуючи позитивну кривизну, як у сферичній геометрії та негативну кривизну, як у гіперболічній геометрії.

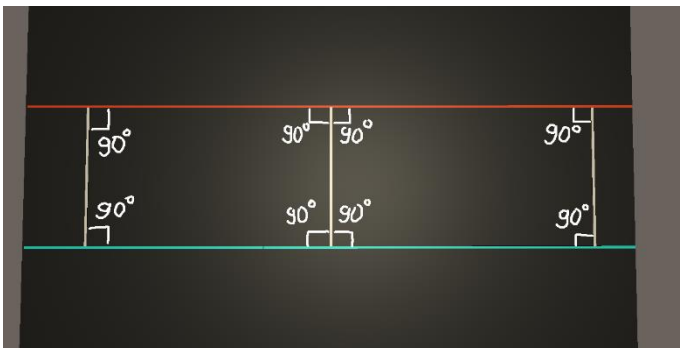
Рис. 1. Розміщення двох кімнат в одному просторі

Рис. 2. Місця переходів у просторі (позначені латинською літерою Т).

CloseRangeTransition (CRT) –перехід, який не змінює позиції гравця, а лише за допомогою stencil buffer’а відображає на площині об’єкти різних груп, які розташовані в одному просторі. LongRangeTransition (LRT) – повноцінний перехід, який при проходженні міняє позицію користувача. PositiveCurvatureTool (PCT) – це інструмент кривизни, який дає змогу імітувати



сферичну геометрію. Цей інструмент створений за допомогою шейдера, який потім встановлюється в об’єкти. Для полегшення роботи з багатьма об’єктами, створено один інструмент – curvatureTool, який об’єднує в себе PCT та NCT, цей інструмент при запуску програми автоматично встановлює потрібний шейдер в потрібні об’єкти [2]. NegativeCurvatureTool (NCT) – це інструмент кривизни, який



працює подібно до PCT, проте з деякими вімінностями.

Рис. 3. Сцена без використання інструменту curvatureTool

negativeCurvatureTool (NCT) – це інструмент кривизни, який працює так само, як і PCT, тільки з деякими нюансами. При реалізації ми зменшуємо вісь ординат вершини враховуючи дистанцію від камери помноженої на від’ємну кривизну.

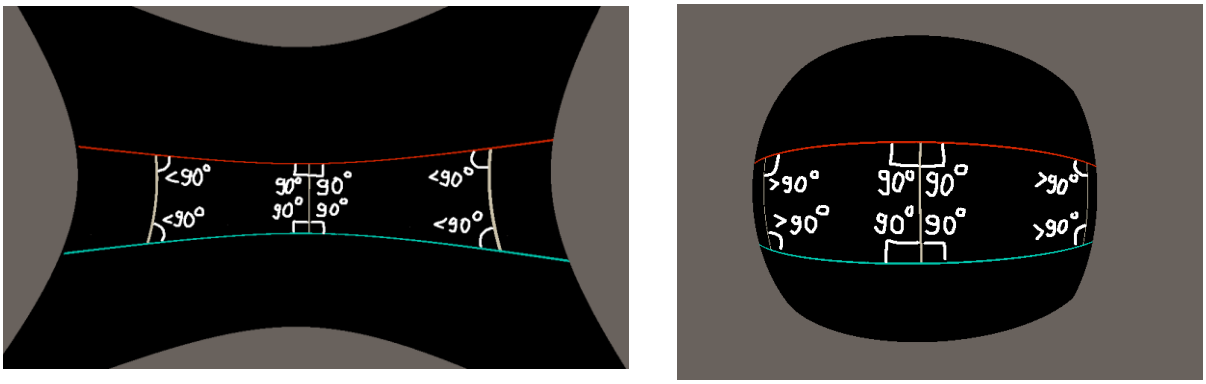


Рис. 4. Демонстрація роботи PCT (зліва) та NCT (справа)

Математична достовірність та гіровектори.

Для математичної достовірної симуляції потрібно створити інструмент gyrovectorSpaceTool, який буде впливати на всі об’єкти сцени міняючи їх звичайні “нормалізовані” позиції у позиції неевклідового простору. Інструмент буде міняти приймаючі та вихідні дані інших інструментів на сцені, але в свою чергу буде мати автономний режим, схожий на роботу curvatureTool. Цього можна досягнути шейдерами, які є максимально низьким доступним рівенем зміни простору в Unity 3D. На відміну від звичайного curvatureTool, який міняє вершини на основі векторів, gyrovectorSpaceTool буде змінювати цей процес на основі

гіровекторів. Щоб ми отримали гіровектори з повністю працюючої на векторах Unity 3D, маємо перетворити нормалізовану координатну площину у координатну площину Пуанкаре, за допомогою якої можна буде виконувати всі можливі дії з гіровекторами.

Тепер за допомогою методу gyrovectorSpaceAction, gyrovectorSpaceTool можна виконувати будь-які дії доступні для гіровекторного простору. Також наявні готові методи (TransformWorldVectorSpaceToGyrovectorSpace, AddGyrovectors,SubtractGyrovectors, MultiplyGyrovectors та DivideGyrovectors).

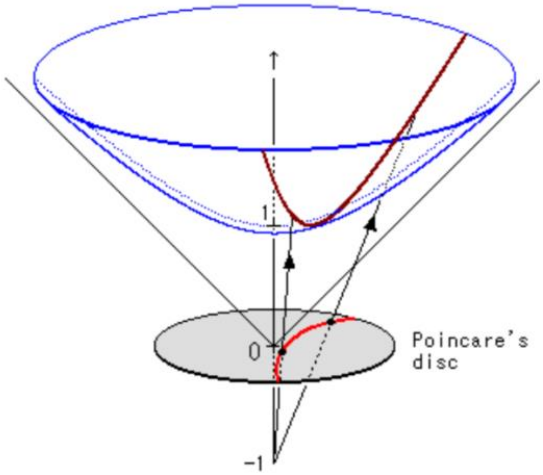


Рис. 5. Стереоскопічна проекція дискової моделі Пуанкаре

Висновки.

У роботі проаналізовано неевклідові геометрії, створено алгоритми роботи неевклідової симуляції у непристосованому для цього двигуні Unity 3D, враховано математичну достовірність та максимально точно здійснено перетворення векторного світу у гіровекторний.

Наразі аналогів цьому проєкту немає, всі інші роботи, які дають користувачам можливості для створення таких проєктів потребують професійних навичок. В подальшому планується випустити проєкт «Noclid» як безкоштовний plug-in для середовища Unity 3D, збільшити набір інструментів та оптимізовувати алгоритми системи, реалізувати додатковий функціонал до інструментів.

Літературні першоджерела.

1. Animal Crossing Curved World Shader // Інтернет-блог «Alastair Aitchson» URL:https://alastaira.wordpress.com/2013/10/25/animal-crossing-curved-world-shader/ (дата звернення 12.02.2021)
2. Hyperbolic Geometry and Poincare Embeddings // Інтернет-довідник «Bounded Rationality». URL: https://bjlkeng.github.io/posts/hyperbolic-geometry-and-poincare-embeddings/
3. Non-euclidean Virtual Reality I: Explorations of H3: тези конференції Bridges

2017, 27-31 липня 2017 / відп. ред. В. Харт .Ватерлоо, 2017. 40 с.
4. Stencil Buffer // Інтернет-довідник «LearnOpenGL». URL: https://learnopengl.com/Advanced-OpenGL/Stencil-testing (дата звернення 06.02.2021)
5. Данилевський М. П., Колосов А. І., Якунін А. В.. Основи сферичної геометрії та тригонометрії: навч. посібник / Харк. нац. акад. міськ. госп-ва. Х.: ХНАМГ, 2011. 92 с.

Створення додатку для аналізу сну з використанням технологій Big Data

Горішна Христина Олегівна

Чернівецький національний університет імені Юрія Федьковича

horishna.khrystyna@chnu.edu.ua

Кожна людина проводить приблизно третину свого життя уві сні. Сон - це один з найважливіших фізіологічних процесів в організмі людини, який впливає на наше здоров'я та якість життя. І хоча сон є одним з найменш досліджених аспектів життя людини, відомо, що проблеми зі сном можуть мати пагубний вплив на наше здоров'я.

Найкращим способом діагностики розладів сну є полісомнографія. Полісомнографія, яку також називають дослідженням сну, - це комплексний тест, який використовується для діагностики порушень сну. Полісомнографія реєструє ваші мозкові хвилі, рівень кисню в крові, частоту серцевих скорочень та дихання, а також рухи очей та ніг під час дослідження.[1] Розшифровка полісомнографії може зайняти від декількох днів до декількох тижнів.

Останні роки технології Big Data("великі дані") активно використовуються в медицині та допомагають діагностиці та лікуванні хвороб, передбаченні епідемій, і т.д. Тому Big Data технології лягли в основу додатку, який на основі полісомнографії зможе визначити чи є в людини розлади зі сном.

Одним з найважливіших етапів є вибір даних, на основі яких робиться аналіз. Ресурс PhysioNet надає доступ до багатьох великих баз даних, які містять різноманітні медичні показники. Для своїх досліджень вибрано дві бази даних, а саме: CAP Sleep Database[2] і Sleep-EDF Database Expanded[3]. Оскільки розмір цих даних є дуже великим, завантажити їх на локальний ПК не вдасться, тому для доступу до даних є два варіанти: через Google Cloud Storage(де ці дані є

продубльовані) за допомогою Google Cloud connector for Hadoop, або через спеціальну функцію, яка є в пакеті MNE[4], розробленого для Python.

Наступним етапом є обробка даних, для цього у роботі використано фреймворк Hadoop MapReduce, а також написано програмний код за допомогою мови програмування Python. Для зручного запуску MapReduce було використано інтерфейс Hadoop Streaming[5].

В якості вхідних даних були файли полісомнографії. Полісомнографія зберігається в форматі edf(European Data Format) і для роботи з цим форматом використано пакет MNE. Фунцкія Map() витягувала дані з вхідного файлу та формувала пару ключ-значення. Пари ключ-значення були сформовані на основі двох показників, які є одними з ключових при постановці діагнозу. Функція Reduce() зводила докупи всі проміжні значення з однаковим ключем, тобто з однаковою частотою мозкових хвиль чи однаковою амплітудою скорочення м’язів.

В результаті було отримано файли з відповідними показниками, які було збережено в таблицю формату csv.

Переконавшись, що всі дані в таблиці є повними, та перетворивши всі дані типу string у числовий тип даних, було проведено машинне навчання. Для цього всі дані поділилися на навчальний набір (training set) та тестовий набір (test set) за пропорцією 80/20, тобто 80% даних були використані для навчання моделі та 20% були використані для перевірки точності моделі. Для машинного навчання було використано бібліотеку для мови програмування Python під назвою scikit-learn[6], зокрема реалізований у цій бібліотеці алгоритм Decision Tree Classifier (дерево рішень).

Модель, отримана після машинного навчання лягла в основу алгоритму, який аналізує нові вхідні дані (файл з розширенням .edf) та видає результат (здоровий сон чи ні).

В результаті роботи розроблено додаток, в якому лікар зможе завантажити файл з даними полісомнографії, та отримати попередні результати. Цей додаток дозволить заощадити час, який витрачається на діагностику, що в свою чергу пришвидшить початок лікування людини з розладами сну.

Список літератури

1. Polysomnography (sleep study) [Електронний ресурс]. - Режим доступу: https://www.mayoclinic.org/tests-procedures/polysomnography/about/pac-20394877
2. CAP Sleep Database [Електронний ресурс]. - Режим доступу: https://www.physionet.org/content/capslpdb/1.0.0/
3. Sleep-EDF Database Expanded [Електронний ресурс]. - Режим доступу: https://www.physionet.org/content/sleep-edfx/1.0.0/
4. MNE MEG+EEG Analysis& Visualization [Електронний ресурс]. - Режим доступу: https://mne.tools/stable/index.html
5. Hadoop Streaming [Електронний ресурс]. - Режим доступу: https://hadoop.apache.org/docs/r1.2.1/streaming.html
6. scikit-learn Machine Learning in Python [Електронний ресурс]. - Режим доступу: https://scikit-learn.org/stable/

Регуляризація формули обернення перетворення Лапласа та її адаптація до розв’язування динамічних задач теорії пружності

Васільєва Діана
Україна, Львів
Львівський Національний Університет імені Івана Франка
Факультет Прикладної Математики та Інформатики
di.vasileva01@gmail.com

Для розв'язування прикладних задач, що зводяться до диференціальних рівнянь еліптичного типу в літературі широко використовують метод граничних інтегральних рівнянь (МГІР). При застосуванні цього методу до рівнянь параболічного та гіперболічного типів в інтегральних рівняннях виникають додатково невідомі від часової координати, що значно ускладнює числову реалізацію задачі. Один із найпростіших способів розв'язування таких задач ґрунтується на інтегральному перетворенні Лапласа, оскільки для знаходження зображення Лапласа може бути ефективно використаний МГІР. Однак подальше визначення оригіналу шуканої функції наштовхується на значні труднощі математичного характеру, оскільки зображення знаходять числово, тобто з деякими похибками. Відомо, що задача числового обернення перетворення Лапласа в таких випадках, взагалі кажучи, є некоректною. Для розв'язання проблеми обернення в літературі розроблено ряд числових методів та їх модифікацій. При цьому найбільш актуальним є вибір таких формул обернення, які б істотно не залежали від похибок обчислення зображень і дозволяли би розв’язувати задачі з контрольованою точністю.

У роботі розглянуто числово-аналітичну формулу обернення перетворення, яка грунтується на точній формулі Пруднікова А.П. Ця формула пов'язує значення оригіналу $f(t)$ та його зображення $F(s)$ і має вигляд [1]

$$f(t)=\frac{1}{l}\exp(ct/l)\sum_{n=-\infty}^{\infty}F(s_n)\exp(2\pi nit/l)-R, \quad 0<t<l, \tag{1}$$

де $R=\sum_{n=1}^{\infty}\exp(-nc)f(t+nl)$; $s_n=(c+2\pi ni)/l$; c, l – довільні сталі ($\mathrm{Re}(c)>0$, $l>0$).

Для застосування цієї формули в задачах обернення необхідно з контрольованою точністю обчислювати ряди типу Фур'є, які збігаються, як

правило, повільно та враховувати складову R, у яку входять невідомі значення оригіналу [2]. Величину R у формулі (1) можна визначити наближено, коли відомим є асимптотичний вираз для функції $f(t)$, при $t \to \infty$. Загалом асимптотика розв'язку є відомою або її можна знайти для широкого класу задач математичної фізики. Позначимо асимптотичне значення функції $f(t)$ при $t \to \infty$ через $f_\infty(t)$ і вибиратимемо l таким, щоб $f(t) \cong f_\infty(t)$ при $t > l$. Тоді

$$R = A(t,l,c) + \varepsilon_1,$$

де $$A(t,l,c) = \sum_{n=1}^{\infty} \exp(-nc) f_\infty(t+nl), \quad \varepsilon_1 = \sum_{n=1}^{\infty} \exp(-nc)\left[f(t+nl) - f_\infty(t+nl) \right].$$

Величина ε_1 може бути зроблена як завгодно малою вибором параметрів c і l. Тоді для знаходження обернення перетворення Лапласа виконується формула (1), у якій R наближено замінюють на A. Для багатьох випадків ряд для визначення функції A можна просумувати. Зокрема, коли $f_\infty = a + bt$, де a, b незалежні від t величини, отримуємо

$$A \cong \left(f_\infty + \frac{bl\exp(c)}{\exp(c)-1} \right) \frac{1}{\exp(c)-1}.$$

Важливим є врахування рекомендацій щодо вибору параметрів, які входять у формулу обернення, оскільки це дає можливість мінімізувати похибки, які можуть виникнути у вибраному підході [1].

Проведено тестування формули обернення і встановлено високу точність та ефективність цієї формули для коливних функцій, за умови, що зображення обчислюються за точними формулами. Встановлення ефективності числово-аналітичної формули обернення перетворення Лапласа проведено з метою її застосовування до розв’язування динамічних задач теорії пружності.

1. Kushnir R. M., Solyar T. Ya. A Numerical-Analytical Approach to the Analysis of Non-Stationary Temperature Fields in Multiply-Connected Solids // Mechanics, Materials Science & Engineering. – 2016. – №3. – pp. 90-106, DOI **10.13140/RG.2.1.1167.0165**.

2. Solyar T. Ya. Improvement of Fourier series convergence on the basis of splines and its application for numerical inversion of Laplace transform // Mechanics, Materials Science & Engineering. – 2016. – №5. – pp. 188-203, DOI **10.13140/RG.2.1.4069.2727**.

Інформаційна технологія розпізнавання геометричних об’єктів у сае-системах

Найбородіна А., Соколовський Я.

Львівський національний університет ім. Івана Франка, ФПМІ

alina26158@gmail.com

Актуальність вивчення систем автоматизованого проектування (САПР) обумовлюється зростаючою конкуренцією на ринку інженерних послуг як серед розробників програмного забезпечення (ПЗ), так і серед підприємств-користувачів ПЗ. Сучасні програми автоматизованого проектування, як правило, реалізують технологію тривимірного геометричного моделювання для отримання форми і розмірів деталей і вузлів і, як наслідок, можливість напівавтоматичної генерації креслень і оформлення їх відповідно до ЄСКД.

У майбутньому системи САЕ будуть основними постачальниками інформації, яка допоможе підтримати проектні групи у прийнятті рішень. Комп’ютерна техніка застосовується у багатьох галузях, таких як автомобільна, авіаційна, космічна та суднобудівна галузі.

Метою роботи є побудова дискретних моделей форм технічних об’єктів та розробка програмного забезпечення для реалізації математичних моделей та розпізнавання форм.

Першим кроком дослідження стану об’єкта є опис його форми. На основі креслення або опису об’єкта необхідно побудувати математичну модель форми об’єкта. Важливо виділити два основних класи форм об’єктів: суцільні області (або тіла) та оболонки. Суцільні тіла – це форми, які можна розглядати як замкнені та обмежені підмножини евклідового простору. Оболонки – суцільна тіла, у яких один з розмірів (товщина) значно менший за інші. У результаті, при

розробці подання цих форм об’єктів у САПР необхідно враховувати їх особливості.

Теорія R-функцій академіка Рвачова В.Л. пояснює, систематизує та узагальнює математичний апарат, необхідний для побудови теоретично необмеженої множини форм і відповідних їм функцій. Але залишається актуальною задача спрощення процесів побудови і верифікації функцій, що подають форми складних об’єктів. З точки зору моделювання форм об’єктів, основне призначення теорії R-функцій – це заміна логічних або теоретико-множинних операцій відповідними їм дійсними функціями при побудові неявного подання $F(P) \geq 0$ для форми довільного об’єкта.

Множина R-функцій нескінченна. Але для додатків немає необхідності знати всі можливі варіанти, а досить мати можливість їх конструювати.

Функція $F: R^n \to R$ є неявною функцією, тому її можна використовувати для подання форм об’єктів R-множинами. Найчастіше для побудови F використовують систему R-функцій:

$$\begin{cases} \neg x = -x, \\ x \cap_\alpha y = \frac{1}{1+\alpha}\left(x + y - \sqrt{x^2 + y^2 - 2\alpha xy}\right), \\ x \cup_\alpha y = \frac{1}{1+\alpha}\left(x + y + \sqrt{x^2 + y^2 - 2\alpha xy}\right), \end{cases} \quad (1)$$

де x, y – значення деяких неявних функцій;

$\alpha = \alpha(x, y), -1 \leq \alpha \leq 1.$

Двовимірні математичні моделі форм об’єктів найчастіше використовують для подання призматичних тіл зі сталою товщиною (пластин, балок тощо), де можна знехтувати тривимірними ефектами. В даному випадку примітивами виступають множини точок на площині, подані дійсними функціями двох дійсних змінних. Базові примітиви відповідають найпростішим формам, поданими елементарними функціями. Їх суперпозиції, отримані з використанням системи (1), визначають нові, більш складні форми.

Для розробки програмного забезпечення використано бібліотеку OpenCV, за допомогою якої аналізується зображення і формується ієрархія геометричних фігур. Для кожного об’єкту здійснюється розбиття на примітиви і визначається зв’язок між ними. На виході маємо функцію простих фігур поєднаних операторами на основі відповідних зв’язків.

Висновок. Отже, запропонований алгоритм дозволяє здійснювати розпізнавання геометричних об’єктів і використовувати результат для моделювання необхідних об’єктів.

Література

[1] Рвачёв В. Л. Теория R-функций и некоторые её приложения. — Киев: Наук. думка 1982.

[2] https://cyberleninka.ru/article/n/ispolzovanie-funktsiy-v-l-rvacheva-dlya-geometricheskogo-modelirovaniya-oblastey-slozhnoy-formy/viewer

[3]https://docs.opencv.org/master/d2/d96/tutorial_py_table_of_contents_imgproc.html

[4] https://shapely.readthedocs.io/en/stable/manual.html

Розробка соціальної аплікації з функціоналом у реальному часі

Баклаєва Анастасія

Львівський національний університет імені Івана Франка

Факультет прикладної математики і інформатики

Anastasiia.Baklaieva@lnu.edu.ua

У березні 2020 року світ замикається вдома через пандемію Covid-19. Усе частіше людство використовує соціальні мережі та додатки для обміну повідомленнями з метою роботи, навчання, спілкування з близькими та проведення вільного часу. За рік активна кількість користувачів таких платформ зросла на 10,5%. Уже в липні 2020 року аудиторія соціальних мереж вперше склала більше половини населення Землі, тобто близько 3,96 млрд (51%) людей [1]. Незважаючи на те, шо такі платформи поглинають багато зусиль та часу, сьогодні вони приносять незліченні переваги своїм користувачам.

Метою роботи є створення веб-аплікації для підтримки соціальних зв'язків та обміну вмістом, яка дозволить невеликій спільноті людей спілкуватися в реальному часі, поширювати медіа та створювати публікації чи вести блог у своєму профілі, отримувати системні сповіщення тощо.

Основними задачами розробки такого застосунку є:

a) дотримання архітектури, яка задовольняє не тільки потреби користувачів, але й розробників та тестувальників ПЗ, не тільки в поточний момент часу, але й з його плином [2];
b) побудова програми з перспективою подальшого горизонтального масштабування, адже кількість користувачів з часом може суттєво зростати. При обміні повідомленнями в реальному часі, може виникнути проблема їх координації між екземплярами застосунку [3];

c) продуктивність, яка найбільше впливає на користувацький досвід. Відомо, що якщо сторінці потрібно більше 3 секунд для завантаження, то 40% людей її покине [4];
d) безпека, адже люди хочуть взаємодіяти в безпечному середовищі та контролювати свою приватність;
e) розробка зручного, привабливого й адаптивного інтерфейсу.

У роботі було створено соціальну аплікацію з використанням платформ ASP.NET Core для серверної частини та Angular для клієнтської частини відповідно. Як СКБД для бази даних використано PostgreSQL, а також застосовано Entity Framework Core для об’єктно-реляційного відображення (ORM).

Авторизацію та аутентифікацію додано на основі JWT токенів. Застосунок дозволяє користувачу відновлювати свою сесію, яка закінчується через відсутність активності.

Розроблений застосунок дає можливість користувачам надсилати повідомлення та використовувати деякий інший стандартний функціонал месенджерів. Увесь функціонал у реальному часі додано за допомогою бібліотеки SignalR. Для розв’язування проблеми координації повідомлень під час горизонтального масштабування використано розподілене сховище Redis, яке зберігає інформацію про всіх підключених клієнтів. Redis також зберігає користувачів, що перебувають онлайн, що дає можливість бачити статус активності в профілі.

Окрім цього, користувач може вести свій власний блог та поширювати медіа в різному форматі, адже йому доступні форматування тексту та зручний завантажувач файлів. Завдяки такому функціоналу існує можливість з легкістю писати математичні формули та код на відомих мовах програмування, обрізати картинки, використовувати “drag & drop”, керувати вкладеними файлами тощо.

Користувач може переглядати зображення постів у галереї, зручно масштабувати та перевертати їх у режимі на “на весь екран”.

На нові підписки та пости приходять пуш-сповіщення, реалізовані за допомогою Service worker-а та бібліотеки Web-Push. Service worker виконує роль проксі-сервера [5]; за допомогою нього також додано кешування постів, повідомлень, стилів тощо.

Серверну частина додатку розгорнуто у Docker-контейнері на хмарній PaaS-платформі Heroku, яка дала можливість автоматизувати розгортання. Клієнтську частина розміщено на файловий хостинг Amazon S3 як статичний сайт. Там також зберігаються всі файли користувачів.

Отже, було досягнуто високої швидкодії та відповідно якіснішого користувальницького досвіду за рахунок SPA та REST API, лінивого та попереднього завантаження модулів програми, пагінації, нескінченної та віртуальної прокрутки та інших загальновживаних технік. Інтерфейс зберігає свій вигляд навіть оф-лайн, а користувач завжди отримує актуальні дані, коли знаходиться в мережі. Таким чином, розроблена платформа дозволить людям легко, швидко та зручно підтримувати зв’язок один з одним у цей непростий для нас час.

Література

1. We Are Social : Digital in 2020 [Electronic resource]. – 2020. – Available from : https://wearesocial.com/digital-2020
2. Martin R. Clean Architecture. A Craftsman’s Guide to Software Structure and Design / R. C. Martin, J. Grenning, S. Brown. – Prentice Hall, 2018. − P. 27-35
3. UXCam : Blog : 20+ Powerful UX Statistics to Impress Stakeholders [Electronic resource]. / Annemarie Bufe – 2021. – Available from : https://uxcam.com/blog/ux-statistics/#:~:text=User%20Experience%20Statistics&text=Every%20%241%20invested%20in%20UX,better%20than%20their%20desktop%20website

4. CodeOpinion : Practical ASP.NET Core SignalR: Scaling [Electronic resource]. / Derek Comartin – 2019. – Available from : https://codeopinion.com/practical-asp-net-core-signalr-scaling/
5. MDN Web Docs : Web technology for developers : Web APIs : Service Worker API [Electronic resource]. – 2021. – Available from : https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API

Побудова h-адаптивної схеми МСЕ на основі НМГЕ- МСЕ оцінювача похибки

Дишкант В., Ривак Н.

Львівський національний університет імені Івана Франка,
вул. Університетська, 1, Львів, 79000,
e-mail: volodymyr.dyshkant@lnu.edu.ua, nataliya.ryvak@lnu.edu.ua

Вступ

Метод скінчених елементів є найпоширенішим чисельним методом у задачах моделювання механіки твердого деформованого тіла. Головна ідея методу скінчених елементів (МСЕ) полягає в тому, що будь-яку неперервну область з нескінченною кількістю ступенів вільності, можна апроксимувати множиною підобластей (елементів) пов'язаних між собою у вузлових точках, які мають скінченну кількість ступенів вільності. Адаптивні схеми (МСЕ) дають можливість оптимізувати обчислювальні ресурси при розв'язуванні задач. У цьому випадку при використанні чотирикутних елементів дискретизації виникає проблема розв'язування задачі на несумісному розбитті, ефективним вирішенням якої є використання мортарних функцій.

Поряд із МСЕ для знаходження чисельних розв'язків задач теорії пружності активно використовують метод граничних елементів (МГЕ). У цьому випадку дискретизується лише границя тіла, що дає можливість значно зменшити розмірність результуючої системи рівнянь. Зауважимо, що в задачах пружності МГЕ дає змогу обчислити переміщення та напруження з однаковим порядком точності, тоді як у МСЕ точність обчислених напружень є на порядок нижчою, ніж точність переміщень. Тому важливо дослідити залежність між локальними похибками результатів обох методів (переміщень і напружень). Порівняння значень напружень буде в основі побудови алгоритму адаптації сітки.

Постановка задачі та схема МСЕ

Розглянемо задачу про напружено-деформований стан осесиметричного пружного тіла. Нехай O_{zr} площина деформації. Позначимо через u, v, w переміщення вздовж осей O_r, O_z та O_ϕ відповідно. Тоді

$$u = u(r,z), \qquad v = v(r,z), \qquad w = 0$$

З співвідношень Коші для компонент тензора деформацій будемо мати

$$\epsilon_{rr} = \frac{\partial u_r}{\partial z}, \quad \epsilon_{zz} = \frac{\partial u_z}{\partial z}, \quad \epsilon_{\phi\phi} = \frac{u}{r}, \quad \gamma_{rz} = \epsilon_{rz} = \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r},$$

$$\gamma_{z\phi} = 2\epsilon_{z\phi} = 0, \quad \gamma_{r\phi} = 2\epsilon_{r\phi} = 0, \quad \sigma_{z\phi} = \sigma_{r\phi} = 0$$

Співвідношення закону Гука для ізотропного випадку запишуться так

$$\sigma_{rr} = D_{11}\epsilon_{rr} + D_{12}\epsilon_{zz} + D_{14}\epsilon_{\phi\phi}, \qquad \sigma_{zz} = D_{12}\epsilon_{rr} + D_{22}\epsilon_{zz} + D_{24}\epsilon_{\phi\phi},$$

$$\sigma_{rz} = D_{33}\gamma_{rz}, \qquad \sigma_{\phi r} = \sigma_{\phi z} = 0, \qquad \sigma_{\phi\phi} = D_{14}\epsilon_{rr} + D_{24}\epsilon_{zz} + D_{44}\epsilon_{\phi\phi}$$

де, $D_{11} = D_{22} = D_{44} = D, \quad D_{12} = D_{14} = D24 = \frac{\nu}{1-\nu}D, \quad D_{33} = \frac{1-2\nu}{2(1-\nu)}D, \qquad D = \frac{(1-\nu)E}{(1+\nu)(1-2\nu)}$, E – модуль пружності (модуль Юнга), а ν - коефіцієнт Пуассона.

Введемо псевдовектори деформацій та напружень

$$[\epsilon] = [\epsilon_{rr}, \epsilon_{zz}, \gamma_{rz}, \epsilon_{\phi\phi}]^T, \quad [\sigma] = [\sigma_{rr}, \sigma_{zz}, \sigma_{rz}, \sigma_{\phi\phi}]^T$$

і перепишемо попередні співвідношення у матричному вигляді. Співвідношення Коші записується так:

$$[\epsilon] = [A]\begin{bmatrix} u \\ v \end{bmatrix}, \quad [A] = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \\ \frac{1}{r} & 0 \end{bmatrix}$$

а закон Гука матеме вигляд:

$$[\sigma] = [D][\epsilon], \quad [D] = D\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 & \frac{\nu}{1-\nu} \\ \frac{\nu}{1-\nu} & 1 & 0 & \frac{\nu}{1-\nu} \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 1 \end{bmatrix}$$

Припустимо, що масові сили відсутні. Вектор зовнішнього навантаження позначимо $[p] = [p_x, p_y]^T$. Функціонал Лагранжа в матричних позначеннях запишеться так:

$$J^{(e)} = \frac{1}{2}\int_{\Omega} [u^{(e)}]^T [D][u^{(e)}] d\Omega - \int_{S_\sigma} [p^{(e)}]^T (u^{(e)}) dS$$

Використання мортарних функцій

Традиційна схема методу скінченних елементів (МСЕ) потребує сумісного розбиття області, а при побудові адаптивних сіток зберегти таку сумісність немож-

ливо. Тому в таких випадках використовують мортарні функції, які є модифікацією МСЕ.

Нехай область Ω розбита на підобласті $\Omega_i,\ i = 1, \dots, N$ так, що $\Omega_i \cap \Omega_j = \emptyset,\ i \neq j$ та $\bigcup_{i=1}^{K} \overline{\Omega_i} = \Omega$. Спільну границю двох підобластей $\Gamma_{ij} = \partial\Omega_i \cup \partial\Omega_j$ називають інтерфейсом. Зауважимо, що розбиття на скінченні елементи для кожної підобласті є різним.

Введемо простір V_i, побудований на базисних функціях розбиття області Ω_i. Для того, щоб задовольнити умови спряження на інтерфейсі Γ_{ij} накладемо умову слабкої неперервності переміщень:

$$\int_{\Gamma_{ij}} (u^i - u^j)\mu_i d\Gamma_{ij} = 0,$$

де μ_i - деякі базисні функції на Γ_{ij}.
Введемо простір Λ побудований на базисі μ_i, а також задамо простір $V = \prod_{i=1}^{K} V_i$.

Таким чином, варіаційне формулювання задачі набуде вигляду:

Знайти такі $(u_h, \lambda_h) \in (V, \Lambda)$, що

$$\begin{aligned} &a(u_h, v_h) + b(\lambda_h, v_h) = f(v_h), \quad v_h \in V, \\ &b(\mu_h, u_h) = 0, \quad \mu_h \in \Lambda \end{aligned}$$

де $a(\cdot,\cdot)$ та $f(\cdot)$ - відповідна білінійна та лінійна форми задачі.
Білінійна форма $b(\cdot,\cdot)$ враховує умову слабкої неперервності

$$b(\mu, u) = \int_{\Gamma_{ij}} \left(u_i - u_j\right) \mu d\Gamma_{ij}$$

Фундаментальні розв’язки та схема НМГЕ

Розв’язок для радіальних та аксіальних кільцевих навантажень рівномірно розподілених по кільцю $\{(r_0, \varphi, z_0), 0 \leq \varphi \leq 2\pi\}$ з інтенсивностями e_r та e_z має вигляд:

$$u_i = G_{ij} e_j, \quad i, j = r, z$$

де e_j - компоненти одиничної зосередженої сили, прикладеної в точці (r_0, z_0);

$$G_{rr} = \frac{1}{4\pi\mu(1-v)} \left[\frac{(3-4v)(r^2 + r_0^2) + 4(1-v)Z^2}{2ra} K(m) - \left\{ \frac{3{,}5 - 4v}{2r} a - \frac{(r^2 - r_0^2) - Z^4}{4raR^2} \right\} E(m) \right],$$

$$G_{zr} = \frac{Z}{4\pi\mu(1-v)} \left[\frac{(r^2 - r_0^2) + Z^2}{2raR^2} E(m) - \frac{1}{2a} K(m) \right],$$

$$G_{rz} = \frac{r_0 Z}{4\pi\mu(1-v)}\left[\frac{(r^2 - r_0^2) + Z^2}{2raR^2}E(m) + \frac{1}{2ra}K(m)\right],$$

$$G_{zz} = \frac{r_0}{4\pi\mu(1-v)}\left[\frac{(3-4v)}{a}K(m) + \frac{Z^2}{aR^2}E(m)\right],$$

$$(r, z) - \text{точка спостереження},$$

$$Z = z - z_0, \qquad a = \sqrt{(r + r_0)^2 + Z^2}$$

$$K(m) = \int_0^{\frac{\pi}{2}} \frac{d\varphi}{\sqrt{1 - m\sin^2\varphi}}, E(m) = \int_0^{\frac{\pi}{2}} \sqrt{1 - m\sin^2\varphi}\, d\varphi \quad -$$

інтеграли першого та другого роду відповідно, де $m = \frac{4rr}{a^2}$. Деформації визначаємо :

$$\varepsilon_{ij} = B_{ijk}e_k, k = r, z,$$

$$\text{де } B_{rri} = \frac{\partial G_{ri}}{\partial r}, B_{\varphi\varphi i} = \frac{G_{ri}}{r}, B_{zzi} = \frac{\partial G_{zi}}{\partial z}, B_{rzi} = \frac{1}{2}\left(\frac{\partial G_{ri}}{\partial z} + \frac{\partial G_{zi}}{\partial r}\right).$$

Далі визначимо напруження: $\sigma_{ij} = T_{ijk}e_k$,

$$\text{де } T_{rri} = \frac{2\mu}{1-2v}\left[(1-v)\frac{\partial G_{ri}}{\partial r} + v\left(\frac{G_{ri}}{r} + \frac{\partial G_{zi}}{\partial z}\right)\right]$$

$$T_{\varphi\varphi i} = \frac{2\mu}{1-2v}\left[v\frac{\partial G_{ri}}{\partial r} + (1-v)\frac{G_{ri}}{r} + \frac{\partial G_{zi}}{\partial z}\right]$$

$$T_{zzi} = \frac{2\mu}{1-2v}\left[v\left(\frac{G_{ri}}{r} + \frac{G_{ri}}{r}\right) + (1-v)\frac{\partial G_{zi}}{\partial z}\right]$$

$$T_{rri} = \mu\left(\frac{\partial G_{ri}}{\partial z} + \frac{\partial G_{zi}}{\partial r}\right)$$

Зусилля визначаємо в точці на поверхні зі зовнішньою нормаллю $\vec{n}$:

$$t_i = F_{ij}e_i, i, j = r, z,$$

$$\text{де } F_{ri} = \frac{2\mu}{1-2v}\left[(1-v)\frac{\partial G_{ri}}{\partial r} + v\left(\frac{G_{ri}}{r} + \frac{\partial G_{zi}}{\partial z}\right)\right]n_r + \mu\left(\frac{\partial G_{ri}}{\partial z} + \frac{\partial G_{zi}}{\partial r}\right)n_z$$

$$\text{де } F_{zi} = \frac{2\mu}{1-2v}\left[(1-v)\frac{\partial G_{zi}}{\partial z} + v\left(\frac{G_{ri}}{r} + \frac{\partial G_{ri}}{\partial r}\right)\right]n_z + \mu\left(\frac{\partial G_{ri}}{\partial z} + \frac{\partial G_{zi}}{\partial r}\right)r$$

Граничні інтегральні рівняння для нормальних дотичних компонент:

$$u_i(P) = \int_\Gamma G_{ij}\,(P, Q)\varphi_j(Q)d\Gamma(Q),$$

$$t_i(P) = \int_\Gamma F_{ij}\,(P, Q)\varphi_j(Q)d\Gamma(Q) \pm \mu_{ij}\varphi_j(P), i = n, \tau$$

де не складає труднощів визначити вигляд функцій $G_{ij}, F_{ij}, \mu_{ij}, i = n, \tau$.
Метод Бубнова - Гальоркіна значно скорочує обчислювальну роботу, тому що

диференціальні рівняння, з допомогою яких складається система рівнянь, мають відносно простий вигляд, при цьому методу характерна достатньо висока точність. Далі застосуємо процедуру Бубнова-Гальоркіна.

$$\sum_{q=1}^{N}\sum_{k=1}^{s}\sum_{j=r,z}\varphi_j^{q,k}\int_{-1}^{1}N'(\eta)\int_{-1}^{1}G_{ij}\left(x^p(\eta),\xi^q(t)\right)N^k(t)J_q(t)dtJ_p(\eta)d\eta -$$

$$-\sum_{k=1}^{s}u_i^{p,k}\int_{-1}^{1}N'(\eta)N^k(\eta)J_p(\eta)d\eta = 0$$

$$\pm\sum_{k=1}^{s}\sum_{j=r,z}\varphi_j^{p,k}\int_{-1}^{1}N'(\eta)\mu_{ij}N^k(\eta)J_p(\eta)d\eta +$$

$$+\sum_{q=1}^{N}\sum_{k=1}^{s}\sum_{j=r,z}\varphi_j^{q,k}\int_{-1}^{1}N'(\eta)\int_{-1}^{1}F_{ij}\left(x^p(\eta),\xi^q(t)\right)N^k(t)J_q(t)dtJ_p(\eta)d\eta -$$

$$-\sum_{k=1}^{s}t_i^{p,k}\int_{-1}^{1}N'(\eta)N^k(\eta)J_p(\eta)d\eta = 0_{\underline{-1}}$$

де $u_j^{p,k} = u_j(x^{p,k}), t_j^{p,k} = t_j(x^{p,k}), l = 1, \dots, s$

Для обчислення інтегралів використовуємо квадратурні формули Гаусса. З рівнянь, що записані для вузла $x^{p,l}$, вибираємо лише рівняння, що відповідають заданим граничним умовам на р-ому ГЕ. Для повних наборів заданих граничних значень на всіх ГЕ отримуємо систему лінійних алгебраїчних рівнянь (СЛАР): $A\varphi = b$. Знайшовши невідомі функції $\varphi_j^{q,k}$. Записуємо дискретні подання для інтегральних співвідношень.

Сформулюємо h-адаптивний алгоритм МГЕ

1. Задаємо початкову сітку граничних елементів.
2. Розв’язуємо задачу на даній сітці за допомогою МГЕ.
3. Для кожного вузла обчислюємо похибку.
4. Елемент, на якому похибка перевищує деяке задане значення, розбиваємо на два рівні елементи.
5. Якщо жоден елемент не було розбито - кінець, інакше переходимо на крок 2.

Література

[1] B. F., “The mortar finite element method with lagrange multipliers / f.b. belgacem // numerische methematik,” с. 173–197, 1999.

[2] P.-L. G. Pascal Jean Frey, “Mesh generation. application to finite elements,” с. 848, 2008.

[3] H. D. V., “Fundamentals of finite element analysis,” *(McGraw-Hill)*, с. 494, 2004.

[4] Амензаде Ю. А., “Теория упругости. Учебник для университетов.” *Высш. школа*, vol. 3, с. 272, 1976.

[5] Дишкант В.З., Дияк І.І., “Розв’язування осесиметричної задачі теорії пружності в перемі-щеннях методом скінчених елементів,” с. 28, 2020.

[6] Дияк І. І., “Чисельне моделювання деформаційних процесів на основі поєднання методів скінченних і граничних елементів.” с. 99–115, 2017.

[7] Дияк І. І., Кухарчук Ю. А., “Дослідження пружної рівноваги плоских тіл методом грани-чних елементів.” *Вісник Львівського ун-ту, сер. мех.-мат.*, vol. 47, с. 87–95, 1997.

[8] Зенкевич О. К., “Метод конечных элементов в технике.” *Мир*, с. 541, 1975.

[9] Сегерлинд Л., “Применение метода конечных элементов.” *Наука*, с. 392, 1975.

Ю.О., “Адаптивний алгоритм чисельного дослідження задачі теорії пружності,” с. 97–105,

Чисельне розв'язування задачі Діріхле для узагальненого рівняння Лапласа за допомогою методу граничних інтегральних рівнянь

Дмитро Афанасьєв

Україна, Львів

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

dmytro.afanasiev@lnu.edu.ua

При математичному моделюванні багатьох фізичних явищ часто виникають диференціальні рівняння в часткових похідних еліптичного типу. Зокрема, поширеним є узагальнене рівняння Лапласа

$$\operatorname{div}\Big(\sigma(x)\operatorname{grad}\big(w(x)\big)\Big) = 0, \qquad x \in D, \tag{1}$$

де $D \subset \mathbb{R}^2$ – обмежена область з межею Г, σ – відома функція, що, зазвичай, описує фізичні властивості області D (наприклад, електро- чи теплопровідність). Нехай задано значення функції w на Г

$$w(x) = g(x), \qquad x \in \Gamma. \tag{2}$$

Тоді можна розглянути плоску внутрішню задачу Діріхле для узагальненого рівняння Лапласа, яка полягає у знаходженні функції $w \in C^2(D) \cap C(\overline{D})$, що задовольняє рівняння (1) та крайову умову (2).

Ефективні чисельні методи є добре відомими для розв'язування звичайного рівняння Лапласа, тоді як його узагальнений випадок є складнішим, оскільки рівняння (1) містить змінні коефіцієнти. Одним зі способів розв’язування такого рівняння є його апроксимація та зведення до рівняння зі сталими коефіцієнтами [1,2]. Для отриманого рівняння використаємо метод граничних інтегральних рівнянь, на відміну від [1], де було застосовано метод граничних елементів.

Якщо $\sigma(x) \neq 0,\ x \in D$, то можна отримати еквівалентне до (1) рівняння

$$\Delta w(x) + \frac{1}{\sigma(x)}(grad\big(w(x)\big) \cdot grad\big(\sigma(x)\big) = 0, x \in D. \tag{3}$$

Нехай розв'язок w має вигляд

$$w(x) = v(x)h(x), x \in D. \tag{4}$$

Підставляючи (4) в (3), та вважаючи, що множник h перетворює перші часткові похідні для функції v в нуль, а $\sigma > 0$, отримаємо диференціальне рівняння відносно h, розв'язок якого

$$h(x) = \frac{1}{\sqrt{\sigma(x)}}. \tag{5}$$

Звідси, рівняння (3) можна записати у вигляді

$$\Delta v(x) + z(x)v(x) = 0, \qquad x \in D, \tag{6}$$

де $z(x) = \frac{|\mathrm{grad}(\sigma(x))|^2}{4\sigma(x)^2} - \frac{\Delta\sigma(x)}{2\sigma(x)}$.

Якщо функція σ мало змінюється в області, то часткові похідні від σ будуть достатньо малі, а отже в області D буде справедливе таке співвідношення

$$R = \left|\frac{\max z(x)}{\min z(x)}\right| \approx 1, \qquad x \in D.$$

Тоді можна замінити функцію z сталою k_{meas}, що буде обчислюватись так:

$$k_{meas} = \frac{1}{meas(D)} \int_D z(x)dD\,, \tag{7}$$

де $meas(D)$ – площа області D. Отже, отримаємо наближене до (6) рівняння

$$\Delta u(x) + k_{meas}u(x) = 0, \qquad x \in D. \tag{8}$$

При цьому крайова умова матиме вигляд

$$u(x) = f(x), \qquad x \in \Gamma, \tag{9}$$

де $f(x)$= $g(x)\sqrt{\sigma(x)}$. Отримавши розв'язок $u(x)$ задачі (8) - (9), знаходимо наближений розв'язок $w(x)$ задачі (1) - (2)

$$w(x) \approx u(x)h(x),\ x \in D. \tag{10}$$

Для різних значень k_{meas} можна отримати різні рівняння:

$k_{meas} = 0$ — рівняння Лапласа,

$k_{meas} > 0$ — рівняння Гельмгольца,

$k_{meas} < 0$ — рівняння Клейна-Гордона.

Розглянемо випадок для рівняння Лапласа, фундаментальний розв'язок для якого є відомим

$$\Phi(x,y) = \frac{1}{2\pi} ln \frac{1}{|x-y|}. \qquad (11)$$

Розв'язок задачі (8) - (9) подамо у вигляді потенціалу подвійного шару

$$u(x) = \int_{\Gamma} \phi(y) \frac{\partial \Phi(x,y)}{\partial \nu(y)} dS(y), \qquad x \in D, \qquad (12)$$

з невідомою густиною $\phi \in C(\Gamma)$, що є розв'язком інтегрального рівняння

$$-\frac{1}{2}\phi(x) + \int_{\Gamma} \phi(y) \frac{\partial \Phi(x,y)}{\partial \nu(y)} dS(y) = f(x), \qquad x \in \Gamma. \qquad (13)$$

Будемо вважати, що Г має параметричне подання

$$\Gamma = \{x(t) = \big(x_1(t), x_2(t)\big),\ t \in [0,2\pi]\}.$$

Параметризуючи інтегральне рівняння (13) та застосовуючи метод Нистрьома (див. [4]) з відповідними квадратурними формулами для 2π-періодичних функцій, отримаємо повністю дискретизовану систему лінійних алгебричних рівнянь

$$-\frac{1}{2}\mu_i + \frac{1}{2n}\sum_{k=0}^{2n-1} \mu_k H(t_i, t_k) = \dot{f}(t_i), \qquad i = 0, \dots 2n-1,$$

де μ_i – шукані наближені значення густини ϕ у квадратурних вузлах $t_i = \frac{i\pi}{n}$, $H(t_i, t_k)$ –значення параметризованого ядра, $\dot{f}(t_i) = f\big(x(t_i)\big)$.

Знайшовши μ_i, наближений розв'язок u_n в області D можна обчислити за формулою

$$u_n(x) = \frac{1}{2n}\sum_{k=0}^{2n-1} \mu_k H(x, t_k), \quad x \in D.$$

Аналогічний підхід можна застосувати і для інших випадків k_{meas}.

Приклад 1. Нехай маємо такі вхідні дані:

$$\Gamma = \{x(t) = (cos\, t, sin\, t),\ t \in [0,2\pi]\},$$

$$\sigma(x) = (2 + x_1 + x_2)^2, \qquad x \in D; \quad g(x) = \frac{{x_1}^2 - {x_2}^2}{2 + x_1 + x_2}, \qquad x \in \Gamma.$$

Легко переконатись [3], що в цьому випадку точним розв'язком буде функція

$$w_{ex}(x) = \frac{{x_1}^2 - {x_2}^2}{2 + x_1 + x_2}, \qquad x \in D.$$

Здійснивши нескладні перетворення, отримаємо $z \equiv 0$, тобто $R = 1$ та $k_{meas} = 0$. Наближені значення $w_n(x)$ будемо обчислювати за формулою

$$w_n(x) = \frac{u_n(x)}{\sqrt{\sigma(x)}}, x \in D. \tag{14}$$

Нижче наведено похибки розв'язку в точках $x^* = (0.37,0.0)$ та $x^{**} = (-0.25, -0.35)$

n	$\lvert w_{ex}(x^*) - w_n(x^*)\rvert$	$\lvert w_{ex}(x^{**}) - w_n(x^{**})\rvert$
8	3.87382403641e-07	3.87698378525e-06
16	4.77465289527e-14	6.90246471091e-12
32	9.02056207507e-17	6.9388939039e-18

Табл.1 Значення абсолютних похибок розв'язку в точках x^*, x^{**} для прикладу 1.

Оскільки рівняння (1) було зведено до рівняння Лапласа без наближень, ми отримали експоненціальну збіжність в прикладі.

Приклад 2. Задано

$$\Gamma = \{x(t) = (0.5 + 0.25\cos t\,, 0.5 + 0.45\sin t - 0.35\sin^2 t\,,\ t \in [0,2\pi])\},$$

$$\sigma(x) = 1 + \frac{1}{2}(x_1^2 + x_2^2), \quad x \in D; \quad g(x) = \frac{1}{2\pi} ln \frac{1+\frac{1}{2}(x_1^2+x_2^2)}{\frac{1}{2}(x_1^2+x_2^2)}, x \in \Gamma.$$

Можна показати, що точним розв'язком є

$$w_{ex}(x) = \frac{1}{2\pi} ln \frac{1+\frac{1}{2}(x_1^2+x_2^2)}{\frac{1}{2}(x_1^2+x_2^2)}, \qquad x \in D.$$

Для цього прикладу $R \approx 1.613356$, $k_{meas} \approx -0.738466$. Відповідне рівняння Клейна-Гордона було розв'язано методом граничних інтегральних рівнянь з використанням потенціалу подвійного шару. Похибки обчислювались в точках $x^* = (0.5,0.25)$, $x^{**} = (0.36,0.52)$.

n	$\lvert w_{ex}(x^*) - w_n(x^*)\rvert$	$\lvert w_{ex}(x^{**}) - w_n(x^{**})\rvert$
16	0.001048989015	0.0016650181148
32	0.000742321882	0.0008113395633
64	0.000579345768	0.0004487834625
128	0.000498493746	0.0002708145689

Табл.2 Значення абсолютних похибок розв'язку в точках x^*, x^{**} для прикладу 2.

На цьому прикладі можна побачити, що апроксимація функції $\sigma(x)$ вплинула на точність методу, проте спостерігається збіжність до розв'язку при збільшенні параметра дискретизації n.

Література

[1] R. Rangogni, R. Occhi, Numerical solution of the generalized Laplace equation by theboundary element method / ENEL-DSR-CRIS, - Milan: Appl. Math. Modelling, 1987.

[2] R. Rangogni, Numerical solution of the generalized Laplace equation by coupling theboundary element method and the perturbation method / ENEL-DSR-CRIS, - Milan: Appl.Math. Modelling, 1986.

[3] A. Beshley, R. Chapko, B. T. Johansson, An integral equation method for the numerical solution of a Dirichlet problem for second-order elliptic equations with variable coefficients / Springer, Journal of Engineering Mathematics, 2018.

[4] R. Kress, Linear integral equations - 2nd ed. / Institut fur Numerische und AngewandteMathematik; Universitat Gottingen - New York : Springer-Verlag, 1999. - 367 p.

Використання мурашиних алгоритмів в алгоритмах оптимізації. Розв'язування задачі комівояжера за допомогою мурашиних алгоритмів

Яриш Віра-Оксана

Україна, Львів

Львівський Національний Університет імені Івана Франка

Факультет прикладної математики та інформатики

oksyarysh@gmail.com

1). ***Актуальність теми***

Планування маршрутів і знайдення найкоротших шляхів з ціллю зменшення витрат - це задача, яка викликала багато дослідницького інтересу протягом останніх десятиліть, і продовжує привертати увагу не тільки через витрати на логістику для великих компаній, а й через постійно зростаючу доступність даних, які компанії можуть використовуватись для керування та відкриття нових областей застосування для таких алгоритмів поза логістикою.

На сьогоднішній день задача пошуку найкоротших шляхів (Travel Salesman Problem) є однією з найрозповсюдженіших задач оптимізації, різні її формулювання в тому чи іншому вигляді зустрічаються в багатьох проблемних ситуаціях. Оскільки вона відноситься до класу *NP*-складних задач, то для задач реальної розмірності не застосовуються точні алгоритми її розв'язання. Тому для знаходження розв'язків TSP найчастіше застосовують евристичні та метаевристичні алгоритми. Часто для різних часткових випадків *TSP* різні алгоритми проявляють себе по-різному, тому кожна конкретна постановка *TSP* потребує застосування найбільш ефективного для неї підходу.

В той же ж час дослідження останніх десятиліть демонструють активні спроби
застосування природних механізмів пошуку розв'язків у задачах оптимізації
складних систем. Зокрема, з'явився цілий науковий напрям – природні обчислення (Natural Computing), який об'єднує математичні методи, що ґрунтуються на природних механізмах прийняття рішень. До таких методів належать: генетичні алгоритми, нейро-мережеві обчислення, еволюційне програмування, мурашині алгоритми тощо.

Тому у своїй роботі я розкриваю тему такого іноваційного і перспективного напрямку в завданнях оптимізації, як мурашині алгоритми, що інтенсивно досліджуються закордоном, але є поки маловідомими в Україні. Це і стало поштовхом для мене у опрацюванні даної теми, оскільки у мене з'явилась можливість розкрити цю тему глибше, і поширити її популяризацію на теренах України.

2. Мета роботи

Зрозуміти, на чому базуються природні обчислення (Natural Computing), які об'єднують в собі математичні методи, вирішити задачу пошуку найкоротшого шляху, дослідити ідею наслідування поведінки колонії мурах для оптимізації задачі комівояжера, дослідити ефективність різних методів оптимізації мурашиних колоній за допомогою їх порівняння між собою.

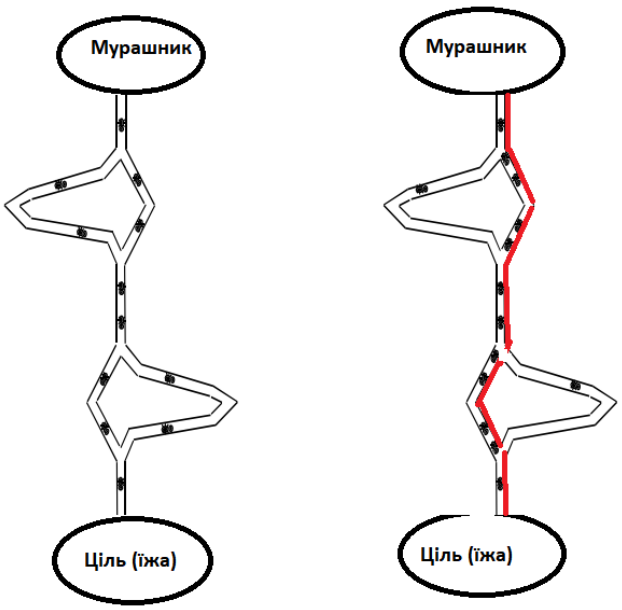
3. Мурашині алгорими

Мурашиний алгоритм – також його ще називають алгоритмом оптимізації мурашиної колонії – це один з поліноміальних алгоритмів, який використовується для знаходження наближених розв'язків завдань пошуку маршрутів на графах, наприклад для задачі комівояжера. В основі алгоритму лежить поведінка мурашиної колонії — маркування вдалих доріг великою кількістю феромону.

Мурашині алгоритми беруть свій початок в спостереженнях над справжніми колоніями мурашок. Мурахи є соціальними комахами, а це означає, що вони – комахи, які живуть в колоніях і їх поведінка є більш спрямована на виживання колонії в цілому, а не на виживання окремого індивіда.

Соціальні комахи привернули увагу багатьох науковців через високий рівень структури їх колоній, особливо беручи до уваги відносно просту

організацію самих мурах поодинці. Важлива і основна деталь, яка так сильно зацікавила дослідників – це поведінка мурах при добуванні їжі, а саме здатність даних комах знаходити найкоротші шляхи між ресурсом їжі і своїм гніздом.



Природня основа алгоритму

4. Алгоритм оптимізації

Алгоритми оптимізації колонії мурашок**,** англ. ant colony optimization, скорочено ACO – будуються на співпраці колонії штучних мурах для пошуку хороших рішень складних проблем дискретної оптимізації. Алгоритм оптимізації колонії мурашок – мета-евристичний алгоритм. Співпраця колонії – це складова конструктивних алгоритмів ACO: вибір полягає у виділенні обчислювальних ресурсів для набору відносно простих агентів (штучні мурахи), які комунікують між собою через стигмерію.

5. Задача комівояжера і застосування алгоритму в ній

Якщо підсумувати опис задачі комівояжера коротко і зрозуміло, то отримаємо таку проблему*:* «Задано список міст і список відстаней між кожною парою міст. Який можливий найкоротший шлях, при якому кожне місто відвідується тільки один раз і при умові, що наприкінці ми маємо повернутись в місто, з якого починали?»

Як застосувати мурашиний алгоритм в задачі комівояжера?

1. Населити початкове місто колонією мурах
2. Кожну мураху відправити в тур по всіх містах, де вони відвідають кожне місто лише один раз. Вибір, яке місто відвідати наступним, є спочатку випадковий. Після першого кола, на вибір міста поступово все більше починає впливати кількість феромону на можливій стежці.
3. Після того як всі мурахи завершили подорож, відкладаєм феромон на кожну стежку до міста(ребро графа). Кількість феромону відкладеного кожним мурахом є пропорційною довжині шляху, пройденого даним мурахом

В кожному місті розподіл ймовірностей для вибору наступного міста з списку можливих генерується за допомогою даної формули:

$$P_{xy} = \frac{t_{xy}^{\alpha} * n_{xy}^{\beta}}{\sum_{k=0}^{N} t_{xz}^{\alpha} * n_{xz}^{\beta}}$$

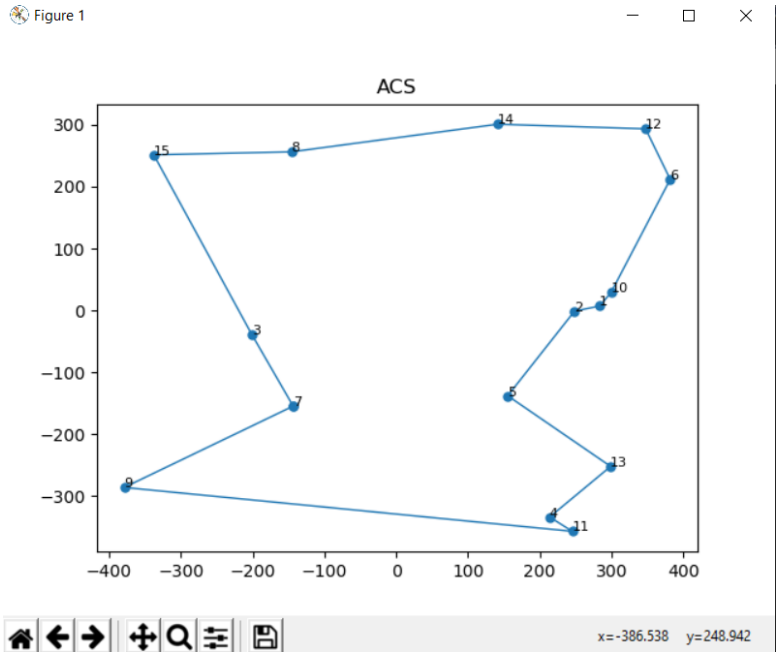
Де, X – місто в якому знаходиться мураха**,** У – невідвідане місто**,** t – кількість феромоному на шляху між містом x і y**,** n – привабливість міста y відносно міста x (сформульоване як обернена відстань між x та y), α – коефіцієнт зважування феромону, β– коефіцієнт зважування привабливості**,** z – місто з списку ще невідвіданих міст.

6. Результати

Я реалізувала алгоритми оптимізації мурашиної колонії мовою програмування Python. Оскільки існує багато варіантів покращення методу оптимізації колоній мурашок, я вирішила не просто реалізувати стандартний метод оптимізації, а ще і реалізувати два методи покращення – щоб показати, що навіть в межах одного типу алгоритмів, результати можуть відрізнятись. Тобто, моя програма складається з реалізації задачі комівояжера, і трьох алгоритму пошуку найкоротшого шляху мурахами – один стандартний, і два види покращення – елітарний (найкращий мураха відкладає зайвий феромон під час локальної процедури оновлення феромонів. Це спонукає колонію до вдосконалення пошуку рішень, які мають високу якість) та MaxMin (). Залежно від проблеми, ці покращення повинні працювати краще, ніж стандартна система колоній мурашок.

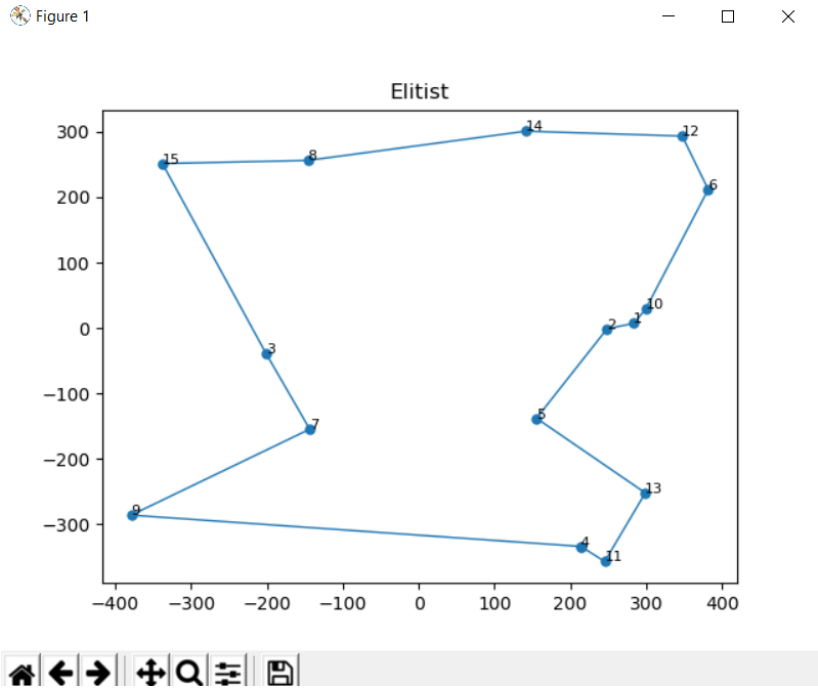
Також я роблю порівняння свого методу, з методом «гілок та меж», за часом оптимізації. Суть полягає в реалізації пошуку найкоротшого шляху для відвідування всіх міст у заданому списку.

- Результати роботи ACS (стандартного алгоритму мурашиної колонії)



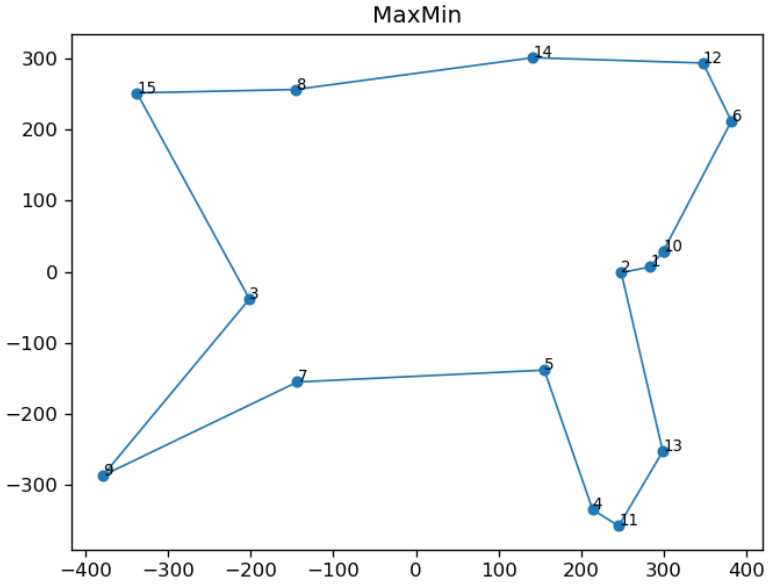
- Результати роботи елітарного алгоритму мурашиної колонії:

(його відмінність полягає в тому, що найкращий мураха відкладає додатковий феромон під час локальної процедури оновлення феромонів. Це спонукає колонію до вдосконалення пошуку рішень, які мають високу якість)



- Результати роботи MaxMin алгоритму мурашиної колонії:

У алгоритмі MaxMin замість того, щоб просто дати можливість найкращому мурасі додатки додаткову вагу, лише такому мурасі дозволяється лишати феромоновий слід. Крім того, MaxMin вимагає, щоб сліди феромону були в проміжку між максимальним і мінімальним значенням. Ідея полягає в тому, що, маючи обмежений діапазон кількості феромону, що знаходиться на стежках, можна уникнути передчасного зближення навколо неоптимальних рішень.




```
C:\Users\Ksy\Desktop>python tryaco.py
Started : ACS
Method : ACS
Path Taken : <- 7 - 3 - 15 - 8 - 14 - 12 - 6 - 10 - 1 - 2 - 5 - 13 - 4 - 11 - 9 - 7 ->
Total distance travelled to complete the tour : 2895.56

The time of algorithm execution:  0.14417505264282227 s
Started : Elitist
Method : Elitist
Path Taken : <- 7 - 9 - 4 - 11 - 13 - 5 - 2 - 1 - 10 - 6 - 12 - 14 - 8 - 15 - 3 - 7 ->
Total distance travelled to complete the tour : 2861.68

The time of algorithm execution:  0.16919231414794922 s
Started : MaxMin
Method : MaxMin
Path Taken : <- 7 - 9 - 3 - 15 - 8 - 14 - 12 - 6 - 10 - 1 - 2 - 13 - 11 - 4 - 5 - 7 ->
Total distance travelled to complete the tour : 2852.94

The time of algorithm execution:  0.1454639434814453 s
```

Як ми бачимо на скріншоті результату програми, прогнози повністю виправдались - найефективніше спрацював алгоритм MaxMin, з найкращим найкоротшим шляхом і найменшим часом роботи. На другому місці іде елітарний алгоритм, з середнім найкоротшим шляхом і середнім часом роботи. І на останньому місці основний алгоритм, до якого ми нічого не доповнювали.

```
=======Branch and bound algorithm===========
Total distance traveled to complete the tour : 3528.94
The time of algorithm execution:  0.33211278915405273 s
```

Зробивши порівняння алгоритмів мурашиної колонії між собою, я вирішила порівняти ACO з кардинально іншим алгоритмом для розв’язку задачі комівояжера – алгоритмом «меж та гілок». Як ми бачимо на скріншоті

програми, алгоритм гілок і меж працює довше ніж алгоритм мурашиної колонії, і менш ефективно. Відповідно, як метод оптимізації алгоритм мурашиної колонії буде кращим.

7. Література

[1] -*Monmarché Nicolas, Guinand Frédéric and Siarry Patrick (2010). Artificial Ants. Wiley-ISTE. ISBN 978-1-84821-194-0.*

[2]- M. Dorigo, *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italy, 1992

[3] - A. Colorni, M. Dorigo et V. Maniezzo, *Distributed Optimization by Ant Colonies*, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[4] - S. Fenet and C. Solnon, “Searching for maximum cliques with ant colony optimization,” in *Applications of Evolutionary Computing, Proc. EvoWorkshops 2003*, ser. LNCS, vol. 2611, pp. 236–245, 2003

ВИКОРИСТАННЯ DEBLURRING В СИСТЕМІ ОПТИЧНОГО РОЗПІЗНАВАННЯ ТЕКСТУ ДЛЯ ЗОБРАЖЕНЬ НИЗЬКОЇ ЯКОСТІ

Яриш Віра-Оксана

Україна, Львів

Львівський Національний Університет імені Івана Франка

Факультет прикладної математики та інформатики

oksyarysh@gmail.com

1). ***Актуальність теми***

OCR, або *optical character recognition*, тобто оптичне розпізнавання символів – це електронне перетворення даних (рукописи, набрані текстові документи, архіви) у вигляді текстових зображень в закодований машинний текст. Це завдання – частина розділу Computer Vision. Чому ця тема набуває популярності?

Завдяки OCR величезна кількість паперових відсканованих документів на різних мовах та у різних форматах, таких як файли PDF, зображення з текстовими даними у .png або .jpeg форматах, дорожні знаки, або будь-які інші зображення з будь-якою формою представлення текстових даних можуть бути оцифровані і переведені в текст, який може прочитати машина, що не тільки полегшує зберігання, але й робить недоступні раніше дані доступними.

Подумайте лише про кількість коробок, заповнених паперовими документами, які розкидані по місцевих архівах, чи старовинні бібліотеки, де зберігаються рукописи, датовані тисячоліттями назад. Такі зображення та документи позиціонуються як надзвичайно цінні для передачі знань

ресурси, і за допомогою OCR їх можна відсканувати, як документ і перевести в електронний формат тексту (таким чином, вони стануть доступними широкій кількості населення і не будуть втрачені через людський фактор чи природні катаклізми).

Завдяки сучасним технологіям, моделі для розпізнавання текстових зображень працюють добре в звичайних умовах при хорошій якості зображень, але коли ми говоримо про розпізнавання зображень низької якості, які мають якісь дефекти– високої якості результату не завжди можна досягти. В такій ситуації, на вхід моделі вже поступає спотворене зображення, тому його розпізнавання не буде ефективним.

Я вирішила розібрати одну з таких проблем - спотворення зображень у вигляді їх розмиття. Передача OCR моделі заблюреного текстового зображення класифікується як зображення низької якості, оскільки це дефект зображення, який заваджає процесу розпізнавання тексту.

В такому випадку, спочатку має зміст використання моделі для покращення вхідного зображення – deblurring (де-розмиття, процес процес видалення розмитих частин із зображень), а тоді передача покращеного зображення в OCR модель для подальшого розпізнавання тексту.

2. Мета роботи

Удосконалення існуючих варіантів розв’язку проблеми розпізнавання тексту оптичною системою (OCR) шляхом застосування deblurring методів.

3. Deblurring model

У роботі використовується deblurring модель DeblurGan. Навчання даної моделі базується на умовній генеративній змагальній мережі (Conditional generative adversarial network, cGAN). Генеративні мережі - це клас алгоритмів штучного інтелекту, що використовуються в навчанні без учителя, реалізовані системою двох штучних нейронних мереж, які змагаються одна з одною в рамках гри з нульовою сумою.

Якщо коротко – cGAN це тип GAN, який передбачає умовне формування зображень за допомогою моделі генератора. GAN покладаються на генератор, який вчиться генерувати нові зображення, і дискримінатор, який вчиться відрізняти синтетичні зображення від реальних.

У cGAN застосовується умовна установка, що означає, що і генератор, і дискримінатор обумовлені якоюсь допоміжною інформацією (наприклад, мітками класів або даними) з інших способів. Як результат, ідеальна модель може навчитися мультимодальному відображенню від входів до виходів, отримуючи різну контекстну інформацію.

- Загальне формулювання deblurring

Загальним формулюванням нерівномірної моделі розмиття є дана формула:

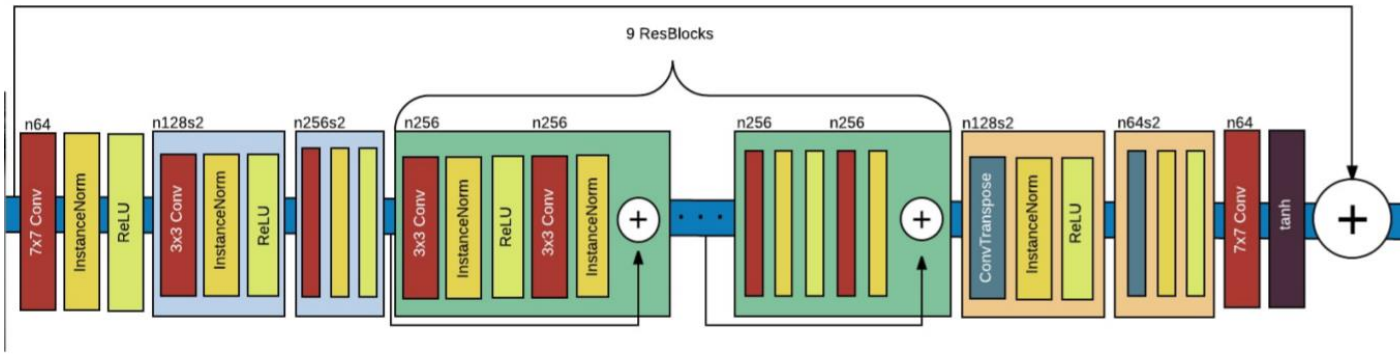
$$I_B = k(M) * I_S + N,$$

де I_B - розмите зображення, а k(M) - невідомі розмиті ядра (kernels), що визначаються полем руху M, I_S - чітке приховане зображення, * - процес згортки, N - адитивний шум.

- Мета DeblurGan

Мета DeblurGan полягає в тому, щоб відновити чітке зображення I_S , маючи лише розмите зображення I_B на вході, а отже ніякої інформації про ядро розмиття (blur kernel) нам невідомо. Де-розмиття відбувається через натреновану CNN (згорткову нейронну мережу - convolutional neural network) G_{θ_G}, яку ми називаємо генератором. Для кожного I_B генератор прикидує відповідне зображення I_S. Крім того, під час етапу навчання ми представляємо мережу D_{θ_D} і тренуємо обидві мережі у порядку змагання.

4. Архітектура DeblurGan



Архітектура генератора CNN

Він містить два поперечні блоки згортки з кроком 1/2, дев’ятьма залишковими блоками (ResBlocks) та двома транспонованими блоками згортки. Кожен ResBlock складається з згорткового шару, шару нормалізації та активації ReLU.

Регуляризація відсіву з імовірністю 0,5 додається після першого згорткового шару в кожному ResBlock. Крім того, вводиться глобальний пропуск з'єднання (skip connection), який ми називаємо ResOut. CNN навчається залишковій корекції I_R до розмитого зображення I_B, таким чином, що $I_S = I_B + I_R$. Таке формулювання робить навчання швидшим, а отримана модель краще узагальнює зображення. Протягом етапу навчання ми визначаємо мережу D_{θ_D} (Wasserstein GAN).

Input

Result

5. Оцінка OCR моделі

Після deblurring процесу отримавши покращене зображення відбувається передача зображення в OCR модель для подальшого розпізнавання тексту.

Я використовую OCR систему Tesseract (оптичний механізм розпізнавання символів для різних операційних систем).

В якості метрики використовуємо відстань Левенштейна - у теорії інформації і комп'ютерній лінгвістиці це міра відмінності двох послідовностей символів (рядків). Обчислюється як мінімальна кількість операцій вставки, видалення і заміни, необхідних для перетворення одної послідовності в іншу.

Після проведеного дослідження, можна вказати те, що OCR система працює значно краще після використання DeblurGan, при спотворених вхідних даних.

6. Висновки

Під час проведення досліджень, було розроблено deblurring модель для процесу видалення розмитих частин із зображень. Практичним способом дана модель була імплементована в процес OCR - оптичного розпізнавання символів, полегшуючи розпізнавання фінального тексту. Тобто, в результаті ми отримали покращене зображення, таким чином удосконаливши існуючі варіанти розв’язку проблеми розпізнавання тексту оптичною системою OCR шляхом застосування процесу де-розмиття.

7. Література

[1] Blind Motion Deblurring Using Conditional Adversarial Networks Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, Jiˇri Matas - https://arxiv.org/pdf/1711.07064.pdf

[2] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Jun-Yan Zhu∗ Taesung Park∗ Phillip Isola Alexei A. Efros Berkeley AI Research (BAIR) laboratory, UC Berkeley - https://arxiv.org/pdf/1703.10593.pdf

[3] Opticle character recognition article https://moov.ai/en/blog/optical-character-recognition-ocr/

[

Створення Telegram бота-помічника з інтерполяції функцій однієї змінної

Гнитка Юрій

Україна, Львів

Львівський національний університет імені Івана Франка

yurii.hnytka@lnu.edu.ua

Моя робота полягає у створенні чат бота-помічника для месенджера Telegram на мові Python, який зможе здійснювати процес інтерполяції функції однієї змінної за різними типами вхідних даних.

Багатьом із тих, хто стикається з науковими та інженерними розрахунками, часто доводиться оперувати наборами значень, отриманих експериментальним шляхом чи методом випадкової вибірки. Як правило, на підставі цих наборів потрібно побудувати функцію, зі значеннями якої могли б з високою точністю збігатися інші отримувані значення. Така задача називається апроксимацією кривої.

Інтерполяцією називають такий різновид апроксимації, при якій крива побудованої функції проходить точно через наявні точки даних.

Існує також близька до інтерполяції задача, що полягає в апроксимації якої-небудь складної функції іншою, простішою функцією. Якщо деяка функція занадто складна для продуктивних обчислень, можна спробувати обчислити її значення в декількох точках, а за ними побудувати (інтерполювати), простішу функцію. Зрозуміло, використання спрощеної функції не дозволяє одержати такі ж точні результати, які давала б початкова функція. Але, для деяких класів задач, досягнутий виграш у простоті і швидкості обчислень може переважити отриману похибку у результатах.

Однак досить часто процес інтерполяції шляхом обчислення вручну вимагає громіздких обчислень та багато часу. Також, в такому випадку, існує велика ймовірність допущення помилок. Оскільки ми живемо в епоху інформаційних технологій, то виникає логічна ідея про створення програми, яка б робила ці обчислення за нас і, бажано, була доступна якомога більшій кількості людей, яким вона може знадобитися. Проблема

доступності вирішується шляхом вживлення цієї програми в уже поширений та доступний продукт такий як месенджер. Адже в наші дні, напевно, немає такої людини, яка б не користувалася, або, принайні, не знала, що таке месенджер. Водночас месенджер вирішує проблему кросплатформеності програми. Одним з найпопулярніших є Telegram.

Telegram – кросплатформенний месенджер з фокусом на швидкодію та безпеку зі щомісячним онлайном в 500 мільйонів користувачів, який входить в топ 10 програм у світі по кількості завантажень.

Telegram дозволяє створювати чат ботів за допомогою його власного Bot API з відкритим кодом. Це API є доступним для багатьох популярних мов програмування.

Я вибрав для розробки мову програмування Python за простоту синтаксису, надзвичайно широкий вибір допоміжних бібліотек (в тому числі і математичних), легкість в написанні та дебагу коду тощо.

Розроблений бот пропонує наступний функціонал:

- Підтримка 5 (на даний момент) методів інтерполяції
- Підтримка різних типів вхідних данних таких як:
    - Функція та відрізок інтерполяції
    - Вузли та значення функції у цих вузлах
    - Вузли та функція
- Вивід детальної інформації про характер помилки у разі її виникнення
- Будування графіків функції, яку інтерполюємо (за наявності) та отриманої функції
- Можливість генерації коду інтерпольованої функції на одній з 8 мов програмування для подальшого її використання. Перелік доступних мов:
    - C
    - C++
    - R
    - Mathematica (Wolfram language)
    - JavaScript
    - Octave/MATLAB
    - Rust
    - Python

- Детальна вбудована документація
- Інтерфейс та документація двома мовами: українська та англійська
- Гнучке налаштування графіків функцій

Даний бот був створений за допомогою таких Python бібліотек як:

- python-telegram-bot – обгортка навколо оригінального Telegram Bot API, яка дозволяє розробляти бота швидше та легше
- numpy – популярна математична бібліотека
- sympy – бібліотека для символічних обчислень
- matplotlib – бібліотека для візуалізації даних двовимірною графікою
- python-i18n – бібліотека для інтернаціоналізації
- А також різних модулів стандартної бібліотеки

Література

1. Інтерполяція – Вікіпедія: https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D1%96%D1%8F
2. Офіційний сайт Telegram: https://telegram.org/
3. Документації:
    a. Python: https://www.python.org/doc/
    b. python-telegram-bot: https://github.com/python-telegram-bot/python-telegram-bot
    c. numpy: https://numpy.org/
    d. sympy: https://www.sympy.org/index.html
    e. matplotlib: https://matplotlib.org/
    f. python-i18n: https://github.com/danhper/python-i18n

Точні розв’язки однієї нелінійної динамічної системи

Литвин Марія
Україна, Львів

Львівський національний університет ім.Івана Франка

Факультет прикладної математики та інформатики

marialytvyn5@gmail.com

Розв’язки нелінійних еволюційних рівнянь (НЕР) мають велике значення для математичної фізики, тому їх пошуку присвячено багато наукових праць учених. Дослідження такого типу рівнянь відіграють важливу роль у поясненні багатьох фізичних феноменів. Зокрема, такі феномени, пов’язані із нелінійними хвилями, трапляються у механіці рідин, фізиці плазми, квантовій фізиці, оптичних волокнах, біології, фізиці твердого тіла, хімічній фізиці, геохімії та багатьох інших галузях науки та інженерії .

Із появою високопродуктивної обчислювальної техніки важливу роль стали відігравати чисельні методи для пошуку наближених розв’язків диференціальних рівнянь із частковими похідними. Разом з тим, створення потужних пакетів комп’ютерної алгебри, таких як MATLAB, MATHEMATICA та MAPLE, спростило пошук точних аналітичних розв’язків таких рівнянь, звільнивши вчених від необхідності проводити складні математичні розрахунки вручну.

Відтак з’явилося чимало нових аналітичних методів розв’язування нелінійних еволюційних рівнянь математичної фізики. Одним із них є метод $\left(\frac{G'}{G}\right)$–розвинення, запропонований у 2008 році (див. [1-3]). І хоч метод є відносно новим, він досить швидко став популярним серед математиків і зараз разом із його численними модифікаціями та узагальненнями вважається потужним інструментом аналітичного дослідження інтегровних нелінійних динамічних систем. Практика засвідчила, що за допомогою згаданого методу та його узагальнень авторам вдавалося знайти більш загальні розв’язки для багатьох рівнянь, порівняно із методами,

запропонованими раніше. У цьому і полягає значення подальшого використання методу $\left(\frac{G'}{G}\right)$–розвинення.

Постановка задачі

Для заданої динамічної системи (1) знайдемо точні розв'язки, використавши (G'/G) метод

$$\begin{cases} u_t = \frac{1}{4}u_{xxx} + 3uu_x - 6vv_x \\ v_t = -\frac{1}{2}v_{xxx} - 3uv' \end{cases} \quad (1)$$

Динамічна система типу (1) була чисельно досліджена в [4].

Опишемо алгоритм за яким будемо шукати аналітичні розв'язки:

Крок 1. Для того, щоб розв'язати задану систему диференціальних рівнянь в часткових похідних (ДРЧП) (1) методом $\left(\frac{G'}{G}\right)$–розвинення, його зводимо до звичайного диференціального рівняння (ЗДР). Для цього скомбінуємо обидві незалежні змінні x і t в єдину змінну ξ, яка відповідає біжучій хвилі:

$$\xi = x - Vt, \quad (2)$$

де V – стала, яку потрібно визначити.

Підставивши (2) в систему рівнянь (1), отримуємо ЗДР, яке неважко розв'язати.

Крок 2. Основна ідея методу полягає у припущенні, що розв'язок системи (1), шукаємо у вигляді многочлена від $\left(\frac{G'}{G}\right)$ вигляду

$$u(\xi) = \sum_{i=0}^{m} \alpha_i \left(\frac{G'}{G}\right)^i, \tag{3}$$

де функція $G = G(\xi)$ є розв'язком лінійного однорідного ЗДР другого порядку вигляду

$$G'' + \lambda G' + \mu G = 0, \tag{4}$$

$m, \alpha_0, \alpha_1, \dots, \alpha_m, \lambda, \mu$ – сталі, які потрібно визначити, і, крім того, $\alpha_m \neq 0$.

Значення m, яке є степенем многочлена (3), можемо визначити, розглянувши гомогенний баланс між членом із похідною найвищого порядку та нелінійними членом найвищого степеня, які фігурують у системі рівнянь (1).

Крок 3. Головне завдання полягає у тому, щоб визначити невідомі коефіцієнти $\alpha_0, \alpha_1, \dots, \alpha_m, \lambda, \mu$. Для цього потрібно підставити (3) і (4) в рівняння (1). Відтак отримаємо інший многочлен від $\left(\frac{G'}{G}\right)$. Зібравши члени при однакових степенях $\left(\frac{G'}{G}\right)$ та прирівнявши їх до нуля, отримаємо систему алгебричних рівнянь для невідомих коефіцієнтів $\alpha_0, \alpha_1, \dots, \alpha_m, \lambda, \mu$. У загальному випадку ця система є перевизначеною, тому розв'язок буде включати деяку кількість вільних параметрів.

Крок 4. Розв'язавши отриману на попередньому кроці систему алгебричних рівнянь та підставивши отримані розв'язки разом із загальним розв'язком лінійного рівняння (4) у представленні (3), отримаємо шуканий розв'язок у вигляді біжучої хвилі для заданої нелінійної динамічної системи (1).

Використавши описані кроки, нами для заданої системи (1) одержано три типи розв’язків: гіперболічного, тригонометричного та раціонального. Побудовані відповідно цим розв’язкам графіки.

Використана література:

1. *Wang M.* The (G’/G) - expansion method and travelling wave solutions of nonlinear evolution equations in mathematical physics / *M. Wang, X. Li, J. Zhang.* // Physics Letters. - 2008. - № 372. - P. 417-423.
2. *Zedan H. A.* New Classes of Solutions for a System of Partial Differential Equations by G'/G-expansion Method / *H. A. Zedan* // Nonlinear Sci. Letters- 2010 - № 1, P. 219-238.
3. *Zhang J.* A generalized (G '/G)-expansion method and its applications / *J. Zhang, X. Wei, Y. Lu* // Physics Letters. - 2008. - № 372. - P. 2254-225
4. Ismail M.S., Ashi H.A. A Numerical Solution for Hirota-Satsuma Coupled KdV Equation: https//doi.org/10.1155/2014/819367.

Криптографія: асиметричні та симетричні алгоритми

Сельменський Руслан Аркадійович, Україна

Львівський національний університет ім. Івана Франка

Факультет прикладної математики та інформатики

ruslanq37@gmail.com

На сьогодні, із стрімким розвитком інформаційних технологій питання інформаційної безпеки є надзвичайно актуальним, особливо в царині електронного управління та електронного бізнесу. Тому сфера інформаційної безпеки активно розвивається, оскільки при вирішенні завдань електронного управління, необхідно захищати інформаційні мережі, інформаційні системи та багато іншого. Періодично з'являється інформація про хакерські атаки на бази даних. Однією з умов створення успішної моделі системи захисту інформаційних систем та комунікаційних мереж є розробка ефективних засобів захисту з керованим рівнем безпеки. Інтенсивний розвиток глобальних мереж потребує подальшого розвитку теоретичних та технологічних методів захисту інформації.

Серед багатьох засобів, які застосовуються для вирішення численних проблем захисту інформації є криптографічні методи. Застосування цих методів є гарантом забезпечення цілісності та вірогідності інформації, яка передається.[1]

Отже, одним з напрямів захисту в інформаційних системах є криптографічний захист інформації, що передбачає використання математичних методів перетворення інформації за допомогою шифрування, вироблення імітовставки або цифрового підпису тощо. Криптографічний захист може здійснюватися в процесі передавання інформації каналами зв'язку та під час її опрацювання на робочих станціях і серверах. До передавання інформації каналами зв'язку ставлять такі вимоги:

- забезпечення конфіденційності інформації;
- забезпечення цілісності інформації;
- автентичність сторін інформаційного обміну.

Конфіденційність інформації забезпечується симетричним (алгоритми ГОСТ 28147-89, DES, 3DES, AES, IDEA) та асиметричним (алгоритми RSA, El Gamal) шифруванням. Цілісність інформації та автентичність сторін досягається використанням хеш-функцій та

технологій цифрового підпису. Сукупність технологій, що забезпечують конфіденційність та цілісність інформації при її передаванні незахищеними каналами зв'язку, отримала назву віртуальних приватних мереж (VPN – Virtual Private Network). У процесі мережевої взаємодії захист інформації, зокрема, забезпечується за допомогою протоколів SSL, SSH, S-HTTP, IPSec тощо. Автентичність сторін інформаційного обміну досягається за рахунок використання протоколів X.509, RADIUS, TACACS+ та інших. Реалізація цих технологій може здійснюватися програмними та програмно-апаратними засобами. Захист інформації на робочих станціях і серверах може реалізовуватися за допомогою шифрування на рівні файлової системи, криптографічних методів перевірки автентичності (цифрові сертифікати, одноразові паролі тощо), криптографічних засобів перевірки цілісності (контрольні суми). Причиною бурхливого розвитку криптографії, з одного боку, є використання комп'ютерних мереж, зокрема глобальної мережі Internet, по яких передають великі обсяги інформації державного, військового, комерційного та приватного змісту, що не допускає можливості доступу до неї сторонніх осіб, а з іншого – поява нових потужних обчислювальних засобів уможливила дискредитацію низки криптографічних систем. Без криптографії не було б стільникових телефонів, банкоматів, цифрового телебачення, Internet-платежів тощо. Методи криптографічного захисту інформації передбачають як програмне, так і апаратне використання. Програмна реалізація шифрування є дешевою та практичнішою. Водночас апаратна реалізація продуктивніша та простіша у використанні. Сучасні криптографічні системи повинні задовольняти такі загальноприйняті вимоги:

- вихідний текст із зашифрованого тексту можна відтворити лише за допомогою ключа дешифрування;
- послідовне перебирання можливих ключів дешифрування з метою відтворення вихідного тексту потребує значного часу обчислень або великих затрат на реалізацію цих обчислень;
- інформація про алгоритм шифрування не повинна впливати на стійкість до зламування системи шифрування;
- незначна зміна ключа шифрування повинна призводити до істотних змін шифрограми одного і того самого тексту. [2]

Алгоритм шифрування з ключем поділяють на дві великі групи – алгоритми симетричного шифрування й алгоритми асиметричного шифрування. Методи симетричного шифрування/дешифрування – це метод, за яким ключі шифрування і дешифрування є або однаковими, або легко обчислюються один з одного, забезпечуючи спільний ключ, який є таємним. Методи асиметричного шифрування/дешифрування – набір методів криптографічного шифрування/дешифрування, в якому

використовують два ключі – таємний (приватний) і відкритий; жодний із ключів не може бути обчислений з іншого за визначений час. Таке шифрування/дешифрування ще називають шифруванням/дешифруванням з відкритим ключем. До 70-х років минулого століття застосовували лише криптографію з симетричними криптоалгоритмами. Криптографія з асиметричними криптоалгоритмами значно молодша. Симетричні та асиметричні криптоалгоритми мають переваги та недоліки. Симетричні криптоалгоритми порівняно з асиметричними мають більшу швидкодію та меншу довжину ключа. Асиметричне шифрування застосовують за такої організації криптосистем, коли використання симетричних алгоритмів є неможливим. А загалом порівнювати характеристики цих криптоалгоритмів було б некоректно: вони створені для розв'язування різних задач шифрування. [3, 4]

Метод симетричного шифрування. Симетричне шифрування ще називають шифруванням з таємним ключем, тобто з ключем, який обидві сторони обміну інформацією (таємно від інших користувачів) використовують для шифрування та дешифрування повідомлень. Основне призначення симетричних криптоалгоритмів – шифрування великих масивів даних із великою швидкістю. Разом із тим, через необхідність наявності захищеного каналу передавання таємного ключа ці криптоалгоритми під час створення сучасних криптосистем виявляють дуже низьку гнучкість. Розрізняють дві великі групи алгоритмів симетричного шифрування: потокове шифрування та блокове шифрування.

Метод асиметричного шифрування. Проблему зростання обсягів шифрованої інформації у криптографії вирішують підвищенням швидкодії традиційних методів шифрування з таємним ключем. Проте застосування цих методів в умовах постійного зростання кількості учасників спільної роботи (децентралізована структура управління) й ускладнення організації взаємодії між ними, зокрема попарного обміну інформацією, виявляється неефективним. Це зумовлено тим, що зі збільшенням кількості учасників обміну інформацією квадратично зростає кількість таємних ключів. Можна показати, що для N учасників кількість таємних ключів у такій системі сягає N(N-1)/2. Крім того, у методах симетричної криптографії з таємним ключем ускладнене довірене узгодження таємного ключа. З метою зменшення цих недоліків було розроблено методи асиметричного шифрування з відкритим ключем. Шифрування з відкритим ключем – порівняно нова галузь криптографії. В асиметричних криптоалгоритмах для шифрування і дешифрування використовують різні ключі: для шифрування – відкриті, для дешифрування – таємні. Асиметрична криптографія основана на ідеях В. Діффі та М. Хеллмана про шифрування з двома ключами, що стали відомими у 1976 році. Але першим алгоритмом асиметричного

шифрування, що набув практичного значення, став алгоритм, який запропонували Р. Рівест, А. Шамір і Л. Адлеман у 1978 році. Він дістав назву алгоритм RSA. Математичним обґрунтуванням асиметричних криптоалгоритмів є важкооборотні (односторонні) функції. У теорії складності обчислень розглядають поняття, яке характеризує рівень складності обчислень (кількість операцій) залежно від розміру вхідних даних. Алгоритми асиметричного шифрування, так само, як і симетричного, застосовують для шифрування масивів даних, але їхня швидкість значно нижча. Основне призначення асиметричних алгоритмів – забезпечення ефективного функціонування сучасних криптосистем. Саме ці алгоритми покладено в основу задач автентифікації користувачів, контролю цілісності та доступності інформації, унеможливлення відмови від авторства чи факту одержання даних тощо, зокрема, в організації електронного урядування. Найповніше ці вимоги задовольняють такі алгоритми асиметричного шифрування: Mars, RC6, Rijndael, Serpent, Twofish.

Важливого практичного значення асиметричні криптоалгоритми набули у застосуванні систем електронно-цифрового підпису (ЕЦП). Кожен з великої кількості криптографічних алгоритмів має певні переваги та недоліки. Тому обсяг аналізу інформації щодо оцінення і вибору криптографічного алгоритму, який найкраще задовольняє вимоги захисту інформації в електронному урядуванні, є доволі великим. Процес вибору передбачає кількісний та якісний аналіз в процесі порівняння різних альтернатив. Із зростанням кількості критеріїв порівняння та кількості альтернатив, які можуть істотно впливати на кінцевий результат.

Література

1. Захарченко М. В. Асиметричні методи шифрування в телекомунікаціях: навч. посіб. / М. В. Захарченко, О. В. Онацький, Л. Г. Йона, Т. М. Шинкарчук. – Одеса: ОНАЗ ім. О. С. Попова, 2011. – 184 с.
2. Тарнавський, Ю. А. Технології захисту інформації [Електронний ресурс] : підручник для студентів спеціальності 122 «Комп'ютерні науки», спеціалізацій «Інформаційні технології моніторингу довкілля», «Геометричне моделювання в інформаційних системах» / Ю. А. Тарнавський ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,04 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с. – Назва з екрана.
3. Грайворонський М.В. Безпека інформаційно-комунікаційних систем [Текст] / М.В. Грайворонський, О.М. Новіков. – К.: Видавнича група BHV, 2009. – 608 с.

4. Алгоритмы симметричного шифрования. Ч. 3. Разработка Advanced Encryption Standart (AES) [Електронний ресурс]. – Режим доступу: http://www.intuit. ru/department/security/networksec/4/5.html.

Візуалізація міграційних процесів до країн єс за допомогою хордових діаграм (з використанням javascript-бібліотеки D3.Js)

Полюга Павло

Україна, Львів

Львівський Національний Університет імені Івана Франка

Факультет прикладної математики та інформатики

pavlo.poliuha@gmail.com

Вступ

Візуалізація сьогодні широко застосовується у наукових і статистичних дослідженнях, а саме: в прогнозуванні; інтелектуальному аналізі даних; бізнес-аналізі тощо. Сучасна наукова візуалізація оцінюється як цілком завершений науковий напрям, де найбільш перспективними є візуалізація інформації і візуальна аналітика.

Актуальність

Особливої актуальності сьогодні набуває процес використання інструментів візуалізації в аналітиці, аналізі даних, системі охорони здоров'я, медицині, демографії. Складним і неоднозначним завданням є встановлення раціонального співвідношення між формою подання даних і функцією візуалізації у аналізі будь-яких процесів. Для повноцінного вирішення такого завдання потрібні новітні інструменти візуалізації, аналізу і обробки даних.

Результати дослідження

У дослідженні було візуалізовано та співставлено міграційні процеси у країнах ЄС за 2010, 2015 і 2019 роки з кількістю мігрантів понад 50 тис., понад 100 тис. і понад 500 тис. осіб з використанням хордових діаграм з JavaScript-бібліотеки D3.js [1].

Хордові діаграми в D3 - це один з найбільш інформативних типів графіків для отримання вихідних даних. В основі побудови є матриця і відображення, яке пояснює, як складається матриця, а також містяться інші метадані, щоб відобразити повну діаграму з відповідними кольорами та мітками.

Одна із побудованих діаграм (рис. 1) відображає міграцію до країн Європейського Союзу з тих країн, звідки прибуло понад 100 тис людей на основі бази даних ООН про міграцію за 2019 рік [2]. При наведенні курсора на кожну країну, виділяються міграційні потоки, країни походження, країни призначення та кількість мігрантів.

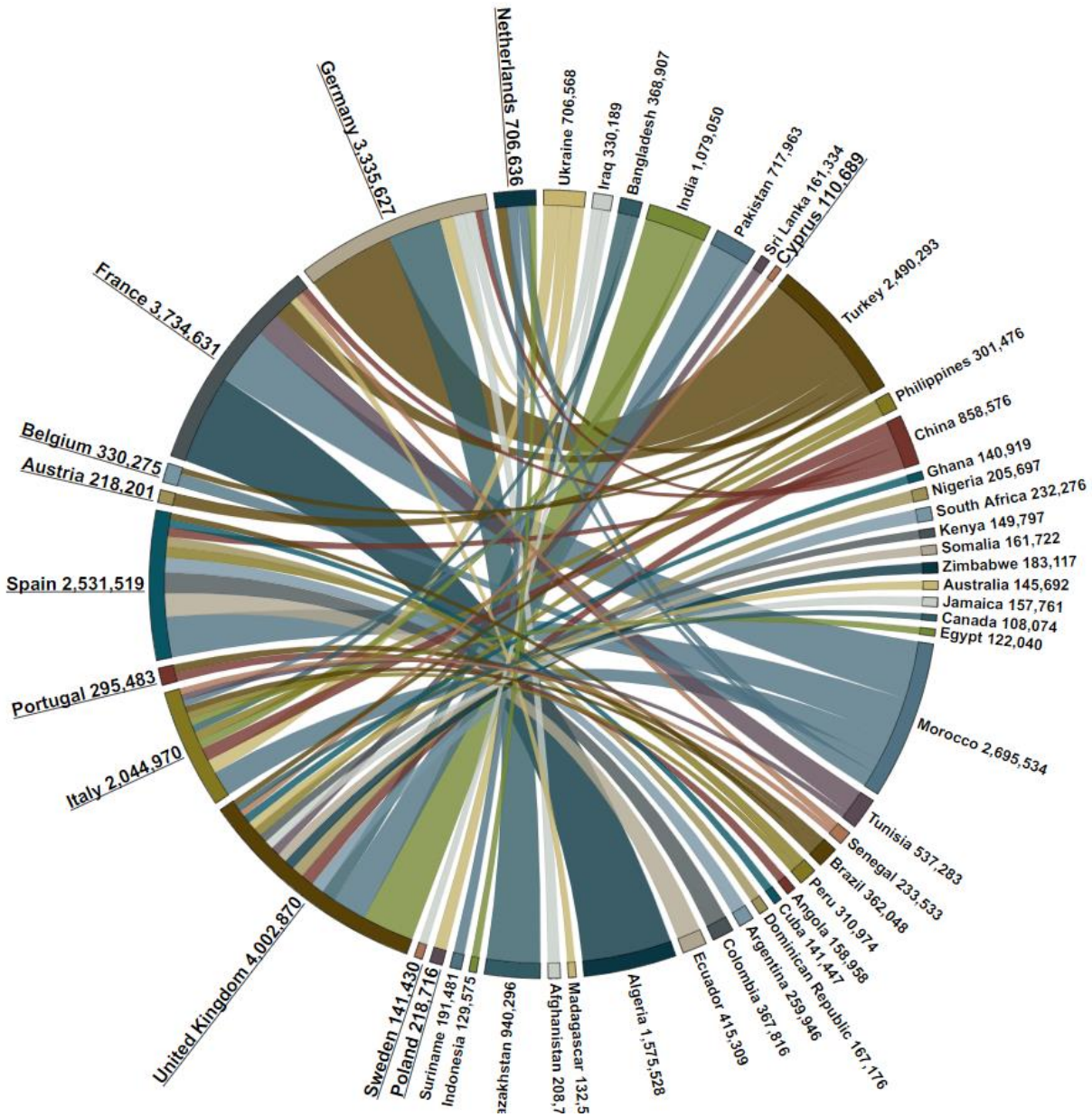


Рисунок 1 - Хордова діаграма з відображенням міграційних потоків для країн з кількістю мігрантів понад 100 тис., 2019 рік

Інтерактивність діаграми показана на рис.2 на прикладі міграції з Туреччини до країн ЄС. Інформаційне поле відображає кількісні значення, а саме, що 1,531,333 осіб мігрували з Туреччини до Німеччини, це складає 45,9% усіх мігрантів до Німеччини (за 2019 рік, враховуючи міграційні потоки для країн з кількістю мігрантів понад 100 тис.).

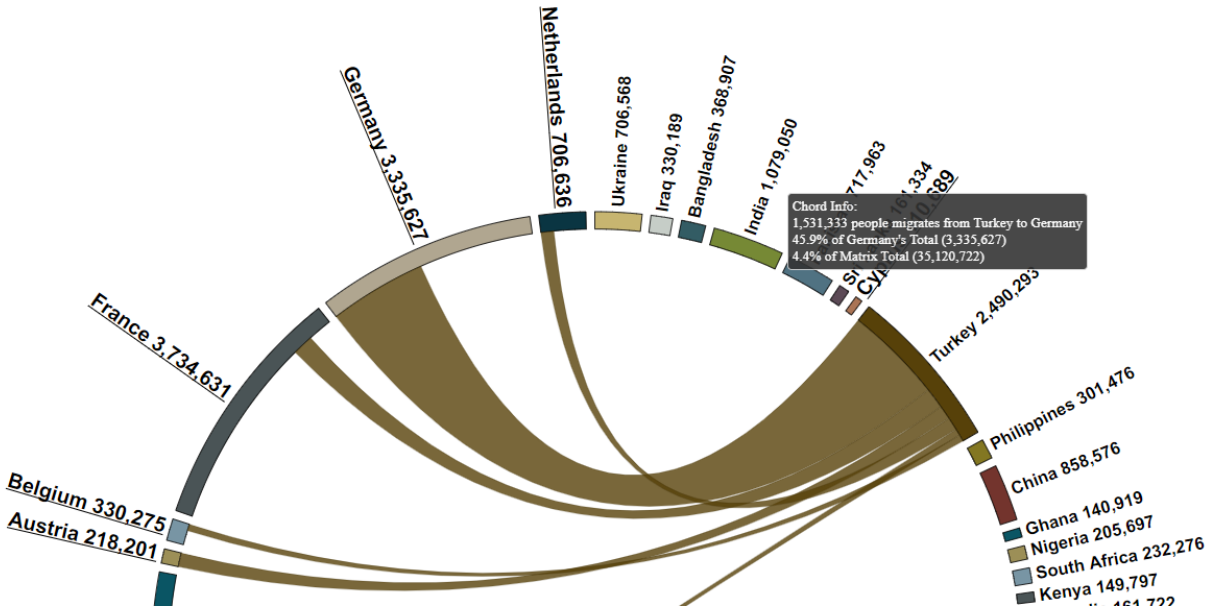


Рисунок 2 - Дані про міграційний потік на основі міграції з Туреччини до Німеччини, 2019 рік

Представлена візуалізація дає можливість інтерактивно відслідковувати міграційні потоки до країн ЄС. Побудовані діаграми за 2010, 2015 і 2019 роки з кількістю мігрантів понад 50 тис., понад 100 тис. і понад 500 тис. осіб наглядніше відображають зміни у міграційних процесах на відміну від табличного представлення даних.

Список використаних джерел

1. Офіційний сайт бібліотеки D3.js [Електронний ресурс]. – Режим доступу: https://d3js.org/.
2. База даних ООН про міжнародний склад мігрантів. [Електронний ресурс]. –Режим доступу: https://www.un.org/development/desa/pd/content/international-migrant-stock

Застосування стабілізаційних схем методу скінченних елементів для задач адвекції-дифузії-реакції

Сидорко Володимир

Львівський національний університет імені Івана Франка,

вул. Університетська, 1, 79000, Львів, Україна

Факультет прикладної математики та інформатики

Volodymyr.Sydorko@lnu.edu.ua

Вступ

Математичне моделювання процесів адвекції-дифузії-реакції (АДР) в однорідних та неоднорідних середовищах є важливою проблемою, яка виникає при дослідженні різних задач: видобуток нафти, прогнозування землетрусів, перебігу хімічних реакцій, поширення забруднень, захоронення ядерних відходів, масоперенесення у живих тканинах, тощо.

У цих задачах, розрахункова область складається з декількох підобластей, які мають істотно різні фізичні властивості. У багатьох випадках математична модель описується диференційними рівняннями з великими числами Пекле, які, як відомо, спричиняють труднощі при застосуванні чисельних методів.

Ця проблема описана, зокрема, в наступних публікаціях [2]-[4]. У цій роботі представлено числовий аналіз та порівняння двох підходів до розв'язування задачі АДР в одновимірному середовищі.

Формулювання задачі

Розглянемо одновимірну крайову задачу:

$$-K\frac{d^2u}{dx^2}+V\frac{du}{dx}+?\,u=f(x),\quad x?\,(a,b),\qquad(1)$$

та граничні умови:

$$?\,K\frac{du}{dx}(a)+?\,u(a)=?\,;\qquad ?\,K\frac{du}{dx}(b)+?\,u(b)=?\qquad(2)$$

де K – коефіцієнт дифузії , V – коефіцієнт адвекції, ? – коефіцієнт реакції, $f(x)$ – задана функція.

Задача (1)-(2) буде сингулярно-збуреною у випадку, якщо коефіцієнт адвекції $(V?\,K),(V??)$.

Метод експоненціальної заміни

Головна ідея методу полягає у використанні експоненціальної заміни [3]

у (1)-(2)

$$u=c\,\exp?\,(\tfrac{Vx}{2K}),$$

та зворотньої заміни виду [2]

$$c=u\exp\left(-\frac{Vx}{2K}\right).$$

Це відозмінна методу скінченних елементів (МСЕ), яка дозволяє, не змінюючи порядок базисних функцій, згладжувати числові розв'язки задач з великими числами Пекле.

Порівняння цього підходу будемо здійснювати з класичним МСЕ з апроксимацією функціями-бульбашками [4].

Апроксимація функціями-бульбашками

Для полегшення обчислень відобразимо стандартний елемент, за допомогою спеціальних співвідношень, на скінченний елемент

$$?_*=\{?:-1<?<1\}$$

На елементі $?_*$ побудуємо послідовність базисних функцій

$$\varphi_1(\xi) = \frac{1}{2}\xi(\xi - 1), \varphi_2(\xi) = \frac{1}{2}\xi(\xi + 1), \varphi_j = \Phi_j(\xi), j = 3,4, \dots, p + 1.$$

Тут $?_j$ визначені через поліноми Лобатто [4].

Апроксимація розв'язку задачі (1)-(2) здійснюється за формулою

$$u = u_{k-1}?_1(?) + u_{k-\frac{1}{2}}?_3(?) + u_k?_2(?) + \overset{p+1}{\underset{j=4}{?}} a_j?_j(?).$$

Числові експерименти

Покажемо застосування зазначених підходів для розв'язування задачі (1)-(2) з наступними коефіцієнтами

$$K = 1, V = 200, ? = 0.1, f = 500, \tag{3}$$

у цьому випадку задача (3) буде сингулярно-збуреною, що призведе до осциляцію розв'язку, що відображено на рис.1.

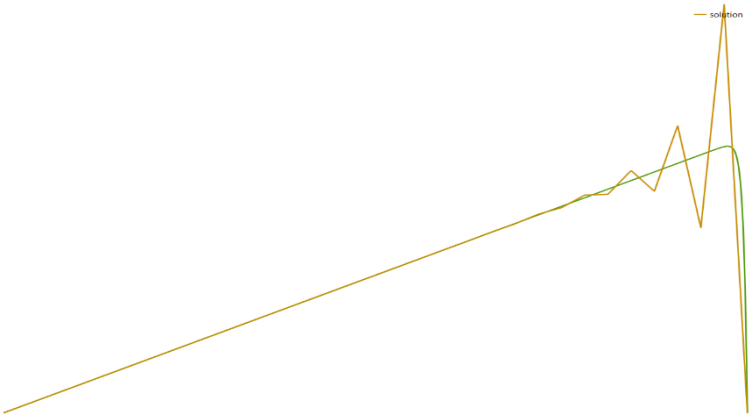


Рис.1. Розв'язок (1)-(2) при n=32

На рис. 2 та рис. 3 зображено графіки аналітичного розв'язку (суцільною лінією) і наближеного розв'язку (точками у вузлах розбиття) задачі (3) при n=32, одержаного з застосуванням апроксимацій на основі функцій-бульбашок та методом експоненціальної заміни відповідно.

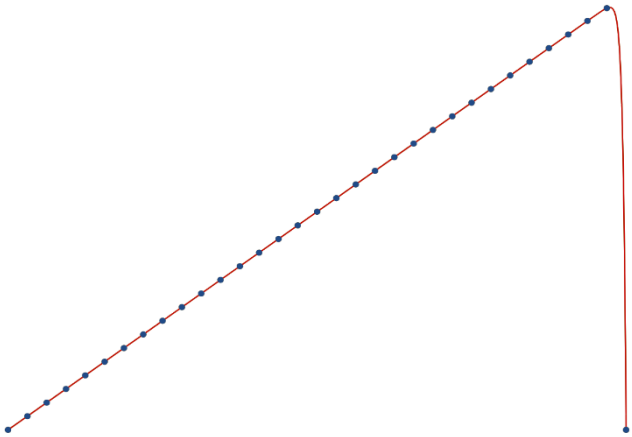


Рис.2

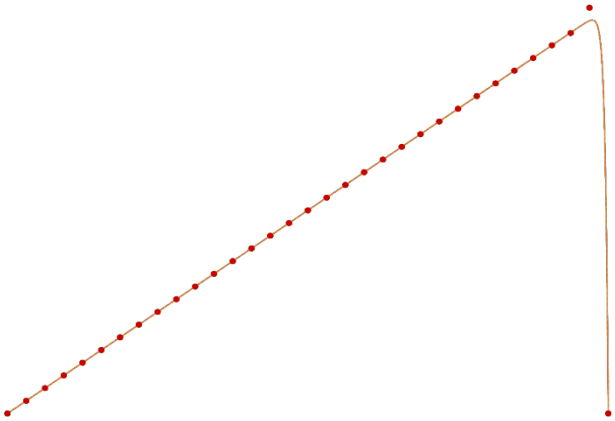


Рис.3

За одержаними результатами (рис. 2 та рис. 3) видно, що розв'язок є досить гладким. Це підтверджує ефективність зазначених підходів.

Здійснено верифікацію наближених розв'язків в нормах $L2$ та $W_2^{(1)}$ шляхом порівняння з аналітичним розв'язком. Результати наведені в таблиці 1.

Таблиця 1. Порядок збіжності

V	N	$p_{bubble}\ in\ L2$	$p_{bubble}\ in\ W_2^{(1)}$	$p_{exp}\ in\ L2$	$p_{exp}\ in\ W_2^{(1)}$
200	64	1.672	0.774	1.144	0.792
	128	1.942	0.959	1.519	0.962
	256	2.022	1.054	1.968	1.044

Висновок

Як бачимо, методи експоненціальної заміни та апроксимація функціями-бульбашками показують значні переваги у використанні для розв'язування сингулярно-збурених задач.

Література:

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

[2] Турчин Ю.І. Експоненціальна заміна у методі скінченних елементів для рівнянь адвекції-дифузії. / Ю. Турчин // Фізико-математичне моделювання та інформаційні технології. Науковий збірник. – 2017 – Вип. 24. С. 11 - 117.

[3] Карташов Э. М. Аналитические методы в теории теплопроводности твердых тел. / Э.М. Карташов // Москва: Высшая школа. – 1985. – 480с.

[4] Pozrikidis Constantine. Introduction to Finite and Spectral Element Methods Using MATLAB. SecondEdition./ConstantinePozrikidis//CRC Press, Jun 20, 2014. –789p.

Використання super resolution у системі оптичного розпізнавання тексту для зображень низької якості

Швецов Д., Баранов М., Селіверстов Р., Іванов С.

Львівський національний університет ім. Івана Франка, ФПМІ

shvetsovdi2@gmail.com

Актуальність теми. Розпізнавання тексту є фундаментальною проблемою в галузі Computer Vision. Серед застосувань можна виділити конвертацію документів в електронний вигляд, автоматизацію систем обліку в бізнесі, розпізнавання номерів машин тощо. Проблема набула великої популярності після розробки згорткових нейронних мереж (*Convolutional Neural Networks, CNN*).

Сучасні моделі розпізнавання тексту показують досить велику точність на чистих та великих зображеннях. Однак якість розпізнавання різко зменшується при зображеннях низької роздільної здатності. При таких вхідних даних моделі важко сприймати спотворену оптичними деградаціями форму символів.

Таким чином, має зміст використання моделі для покращення вхідного зображення (*Super Resolution, SR*) перед тим як передавати його до моделі розпізнавання тексту (*Optical Character Recognition, OCR*). SR модель збільшує роздільну здатність вхідного зображення з максимальною схожістю до його справжнього покращеного (*High Resolution, HR*) варіанту.

Метою даної роботи є покращення існуючих рішень розпізнавання тексту з зображень низької роздільної здатності шляхом застосування методів SR.

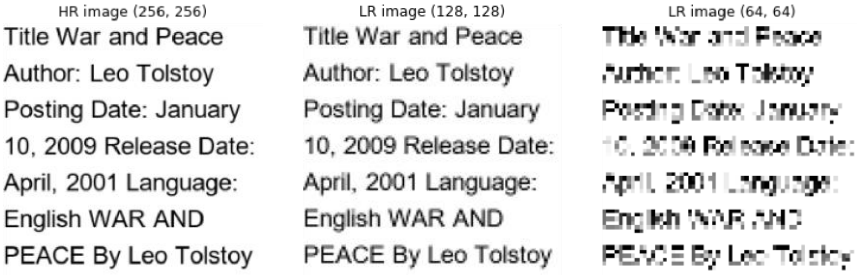
Вибірка

У роботі використано 10000 зображень, штучно згенерованих з тексту книги "Війна і мир" Лева Толстого.

Характеристики кожного зображення:

Розмір зображення	256x256
Формат зображення	PNG
Шрифт тексту	Arial
Розмір тексту	16px
Відстань між рядками	1.5px

Загальна вибірка була розбита на три підвибірки: навчальну,



валідаційну та тестову, - у відношенні 8x1x1. Генерація зображень низької роздільної здатності виконувалась бікубічною інтерполяцією з відповідним фактором зменшення (масштабування) *k*. Приклади роботи алгоритму на одному з зображень вибірки:

I_{HR} k =2 k =4

Архітектура SR моделі

У роботі використано модель PAN (*Pixel Attention Network*) [1] як представник класу SR систем, що використовує лише одну модель – генератор. В основу моделі входять *N* послідовних SC-PA (*Self-Calibrated Pixel Attention*) блоків для видобутку ознак з вхідного зображення.

Таким чином маємо

$$x_n = f_{SCPA}^{n}\left(f_{SCPA}^{n-1}(\ldots f_{SCPA}^{0}(x_0)\ldots)\right), \quad (2.1)$$

де x_i – ознаки видобуті попереднім шаром.

Для фізичного реконструювання зображення використовують m U-PA(*Upsampling with Pixel Attention*) шарів, де m – фактор збільшення вхідного зображення. Інтерполюючи вхідне зображення I_{LR} білінійним алгоритмом інтерполяції f_{bl} та додаючи до цього вивід останнього U-PA блоку, отримують результуюче зображення I_{SR}.

Тобто

$$I_{SR} = f_{bl}(I_{LR}) + f_{rec}(x_n), (2.2)$$

де

$$f_{rec}(x_n) = f_{UPA}^{m}\left(f^{m} - 1_{UPA}(\dots f_{UPA}^{0}(x_n)\dots)\right), (2.3)$$

Також для отримання неглибоких ознак використано проміжні згорткові шари.

Функції втрат та метрики

Слідуючи науковій роботі [2] по найпопулярніщих функціях втрат для проблеми SR, найкращим варіантом є комбінація L_1 та L_{MSSSIM} функцій.

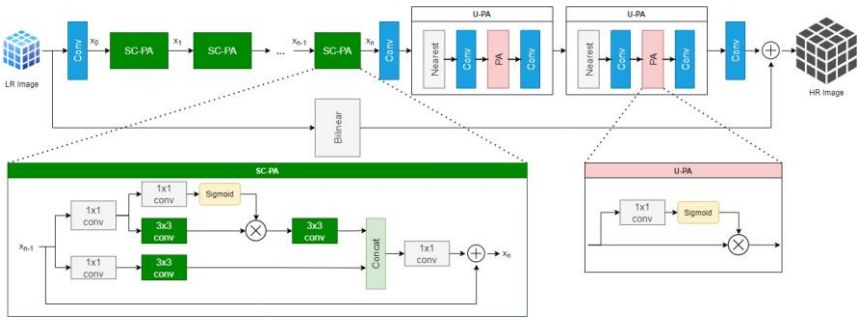


Рис. 2.1: Повна схема SR моделі

Отже, остаточна функція втрат має вигляд:

$$L_{mix} = \alpha \cdot L_{MSSSIM} + (1 - \alpha) \cdot L_1 \cdot G_{\sigma_G^M}, (3.1)$$

Щодо метрики визначення точності розпізнавання тексту з зображення, використано так звану середню відносну відстань (*mean relative distance*).

$$f_{rel}(t,\tilde{t}) = \frac{1}{N}\sum_{i=1}^{N}\frac{f_{edit}(t,\tilde{t})}{l_t}, \qquad (3.2)$$

де t – правильний текст, $\tilde{t}$ – згенерований текст, l_t – загальна довжина тексту t, f_{edit} – відстань редагування (*edit distance*).

Оцінки якості OCR моделі

OCR система	**Масштаб**	**SR**	f_{rel}
Tesseract	×2	N	0.0654
Tesseract	×2	Y	0.0437
Tesseract	×4	N	0.9934
Tesseract	×4	Y	0.313

За результатами метрики (3.2), OCR система працює значно краще при використанні SR моделі на зображеннях низької роздільної здатності.

Висновки

Під час роботи було розроблено SR модель та показано важливість її використання при розпізнаванні тексту з зображень низької роздільної здатності. Результати демонструють, що використання SR моделі як проміжний крок значно покращує точність OCR системи та ліквідує її головний недолік - вимогу якості вхідного зображення.

Література

[1] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong *Efficient Image Super-Resolution Using Pixel Attention* (Oct 2, 2020) [Electronic resource]. — Available from: https://arxiv.org/pdf/2010.01073

[2] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz *Loss Functions for Image Restoration with Neural Networks* (Apr 20, 2018) [Electronic resource]. — Available from: https://arxiv.org/pdf/1511.08861

Розробка прототипу системи оптичного розпізнавання тексту для зображень низької якості

Баранов М., Іванов С., Соколовський Я., Юрченко Ю.
Львівський національний університет ім. Івана Франка, ФПМІ
yuliana.yurchenko1673@gmail.com

Актуальність теми. Використання фотосенсорів (матриць) і аналогової електроніки пристроями для реєстрації зображень неминуче призводить до появи спотворень зображень – цифрового шуму, який відображається на зображення у вигляді накладеної маски з пікселів випадкового кольору і яскравості. Цифровий шум посилюється в умовах недостатньої освітленості, внаслідок руху документа або камери під час експозиції. На відміну від сканерів при зйомці камерою сам документ розташований в довільній площині відносно площини сфокусованого зображення, що може призводити до проєктивного спотворення зображення документа. Ще одне джерело спотворень – алгоритми стиснення зображень, особливо характерні для кадрів відеопотоку. Сьогодні існує безліч систем для розпізнавання тексту, однак лише мала їх частина здатна ефективно працювати з зображеннями низької якості.

Аналіз попередніх результатів. У статті “OCR model for reading Captchas” [1] A. K. Nain описує модель, що поєднує використання Convolutional Neural Network (CNN) та Recurrent Neural Network (RNN), а також екземпляру користувацького шару та використання його як ‘шару кінцевої точки’ (‘endpoint layer’) для реалізації функції втрат Connectionist Temporal Classification (CTC).

Автор цієї моделі застосовує її для розпізнавання зображень CAPTCHA, а саме зображень однорядкового 5-символьного тексту.

Така архітектура не забезпечечує необхідної якості для зображень багаторядкового тексту і вимагає застосування додаткових алгоритмів.

Метою нашого дослідження є побудова алгоритму, що збільшить ефективність навчання нейронної мережі [1].

Алгоритм розрізання зображення багаторядкового тексту на декілька зображень однорядкового тексту. Розглянемо зображення тексту у відтінках сірого (grayscale) – кожен піксель $p_{i,j} \in [0,256)$, де 0 відповідає за чорний колір, 1 – білий; $i = \overline{1,n}$, $j = \overline{1,m}$, n та m – кількості рядків та стовпців зображення відповідно. Нехай $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, де величина

$$x_i = \sum_{j=1}^{m} (1 - \frac{p_{i,j}}{255}), \quad i = \overline{1,n}$$

чисельно виражає кількість чорного кольору в кожному рядку.
Щоб встановити горизонтальне положення зображення шляхом перебору повертаємо його на такий кут ϕ, для якого коефіцієнт варіації $v = \frac{\sigma}{\bar{x}}$ ознаки $\mathbf{x}$ максимальний.

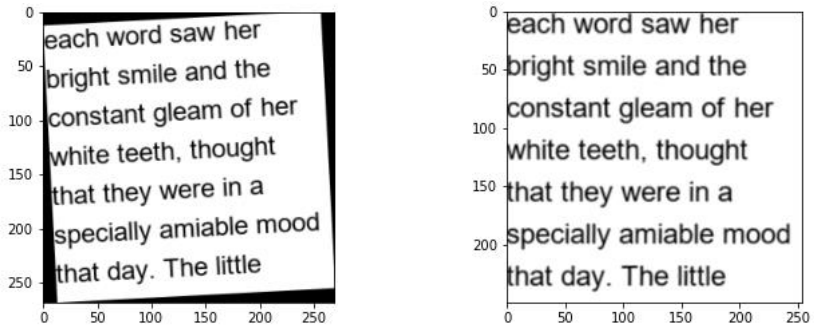


Рис. 1. Результат встановлення горизонтального положення зображення

Місця можливих розрізань зображення визначаються рядками i, для яких:

- $\{x_{i-\alpha}, ..., x_{i-2}, x_{i-1}\}$ утворюють область низької кількості чорного кольору. Визначимо це так:

$$\overline{x_\alpha} = \frac{1}{\alpha} \sum_{j=i-\alpha}^{i-1} x_j \quad < \quad \bar{x} - \frac{1}{2}\sigma$$

- $\{x_i, x_{i+1}, ..., x_{i+\beta}\}$ утворюють область високого значення приросту кількості чорного кольору. Визначимо це так:

$$d_\beta = \frac{\sum_{j=i}^{i+\beta-1} x_j - \sum_{j=i-\beta}^{i-1} x_j}{\beta} \quad > \quad \delta \cdot \sigma$$

Тут α, β та δ – параметри, що задаються.

На рисунку 1.2 місця можливих розрізань позначені червоним кольором. З кожної групи таких індексів вибирається найлівіший (позначено пунктирною лінією).

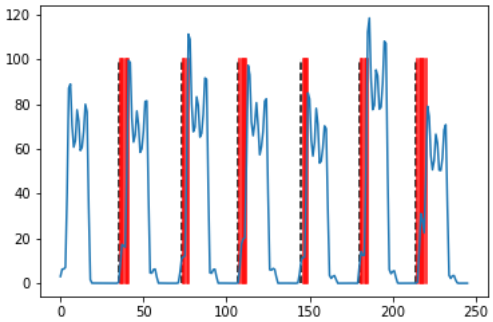


Рис. 2. Графік залежності чисельного вираження кількості чорного кольору x_i від номеру рядка i

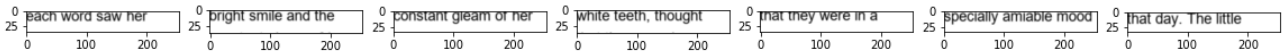


Рис. 3. Результат роботи алгоритму розрізання зображення багаторядкового тексту на декілька зображень однорядкового тексту

Результати ефективності моделі. Наша модель тренувалася на основі нейронної мережі [1] та тестувалася на створеному нами наборі даних зображень низької якості, взятому у відношенні $10{:}1$.

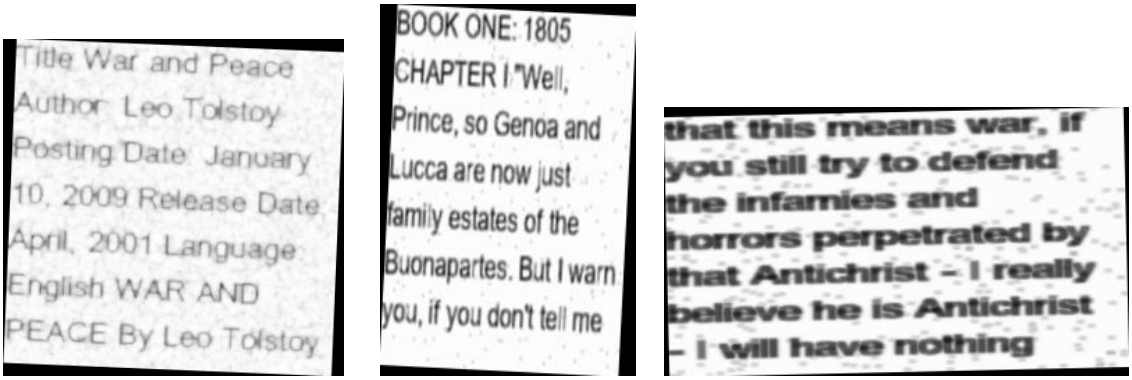


Рис. 4. Зразки набору даних зображень низької якості

Для порівняння використовуємо Tesseract OCR v5.0.0. (alpha) [2].

Як метрику порівняння взяли відношення відстані Левенштейна [3] між виводом моделі та ground truth до довжини ground truth.

Наша модель	Tesseract
0.019783	0.726493

Табл. 1. Середні значення метрики порівняння на тестовому наборі даних для нашої моделі та Tesseract

Висновок. Отже, запропонований нами алгоритм дозволяє отримати високу точність розпізнавання тексту на створеному наборі даних зображень низької якості.

Використана література:

[1] OCR model for reading Captchas https://keras.io/examples/vision/captcha_ocr/

[2] GitHub repository for Tesseract OCR v5.0.0 (alpha) https://github.com/UB-Mannheim/tesseract/wiki

[3] *В. И. Левенштейн*, Двоичные коды с исправлением выпадений, вставок и замещений символов, Докл. АН СРСР, 163, 4, 1965, 845-848

Реалізація чат бот для відстеження погоди, використовуючи мову програмування Java

Ванджала Тарас
Україна, Львів
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
tvangjk@gmail.com

Історія створення перших чат ботів тягнеться ще з 60-х років 20 століття, де їх використовували, для дослідження штучного інтелекту та обробки природної мови. Як приклад, можемо згадати, чат бот «PARRY», який був створений американським психіатром Кеннетом Колбі у 1972 року. Він імітував реакції пацієнта, хворого на параноїдальну шизофренію. Працював чат бот за допомогою складної системи припущень, атрибуцій та емоційних відповідей.

Чат ботів, яких ми тепер звикли бачити у месенджерах, завдячують соціальній мережі «Telegram», де вони вперше з'явилися з 2015 року. Розробник «Telegram» Павел Дуров, вніс вагомий внесок у розвиток стрімко набираючої популярності сфери розробки програмного забезпечення для комунікації з користувачами. За останні 5 років чат боти набули масової популярності та широко впроваджуються у найпопулярніші месенджери, «Telegram», «Viber», «Facebook Messenger», «WhatsApp».

У сучасному розумінні, чат бот – це програма, яка імітує інтерактивну розмову людини за допомогою ключових, заздалегідь підготовлених фраз

та слухових або текстових сигналів. Виходячи з того як відбувається обмін повідомленнями в системі, її можна описати, як «людина»–«комп'ютер».
Можемо виділити два варіанти чат ботів, прості, які здатні виконувати задані користувачем команди, та складніші з елементами штучного інтелекту. У популярних соцмережах і месенджерах працюють прості боти, які розуміють текст та взаємодія з користувачем через кнопки, фото та текстові команди. Суттєвим плюсов в користуванні ботом, вважається відсутність необхідності в будь-яких технічних знаннях та навичках програмування. Загалом листування з чат ботом не відрізняється від листування з живою людиною, з тією лише різницею, що бот розуміє тільки певний набір команд та повідомлень на які він і реагує.

Надалі я опишу, створення чат бота для соціальної мережі «Telegram» використовуючи мову програмування Java. Для подальшої роботи знадобиться завантажити бібліотеку «Telegram LongPolling Bot» або «Telegram Webhook Bot».
Різниця між ними полягає у зв'язку між ботом та користувачем. Long-polling використовує постійне з'єднання між сервером та клієнтом не закриваючи з'єднання поки не з'являться дані або не мине час очікування, простими словами постійно надсилає запит, чи з'явилися нові повідомлення від користувача. Webhook в свою чергу використовує метод зворотніх викликів (callbacks), коли немає постійного підключення до сервера, проте коли з'являється нове повідомлення від користувача, надсилається запит на сервер де згодом цей запит обробляється.
Отже, використовуючи програмне середовище Java та бібліотеку «Telegram Long Polling Bot» реалізуємо чат бот, для відстеження погоди у будь-якій

точці світу. Бот у відповідь на повідомлення користувача надсилатиме повідомлення про стан погоди вказаного місця.

Поставлену задачу можна поділити на декілька етапів:

1. Реєстрація бота та отримання унікально токена для подальшої роботи;
2. Отримання та обробка повідомлення користувача;
3. Запит на «Openweathermap» api для отримання поточного стану погоди в вказаному користувачем місті;
4. Обробка отриманого json файлу від нашого запиту;
5. Відповідь користувачеві про станом погоди на даний момент;

- Перш за все реєструємо бота в «Telegram Bot Father bot» та отримуємо спеціальний токен, який ми повинні ставити в перевизначений метод нашої бібліотеки. Також на цьому етапі, нам потрібно продумати архітектуру та створити відповідні пакети для нашого проекту.
- Повідомлення, від користувача приходить на сервер у форматі json, в якому знаходять всі потрібні дані для подальшої роботи, chatid, memberid, messageid.
- Телеграм бібліотека, яку ми попередньо підключили допомагає легко розділити та дістати потрібні нам дані. Таким чином бот вже може надіслати відповідь користувачі. Використовуючи різні api, чат бот може перетворити на онлайн магазин або бота для відстеження пакунків на пошті та багато іншого. Проте як приклад візьмемо «Openweathermap» api, так як воно є безкоштовним та ідеальним для того щоб показати можливості бота. Надішлимо GET запит на адресу погодного сервісу, у

відповідь ми отримаємо json з потрібною інформацією, з якого дістаємо потрібну інформацію та поміщаємо у нашу модель даних та надсилаємо відповідне повідомлення користувачеві, що можна побачити на рисунку 1 та рисунку 2.

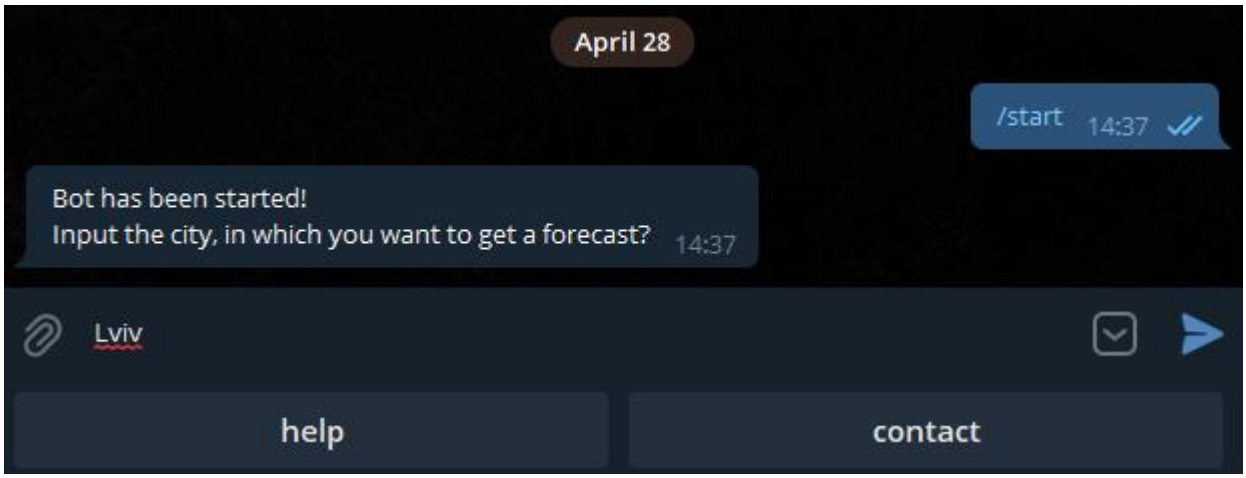


рис. 1

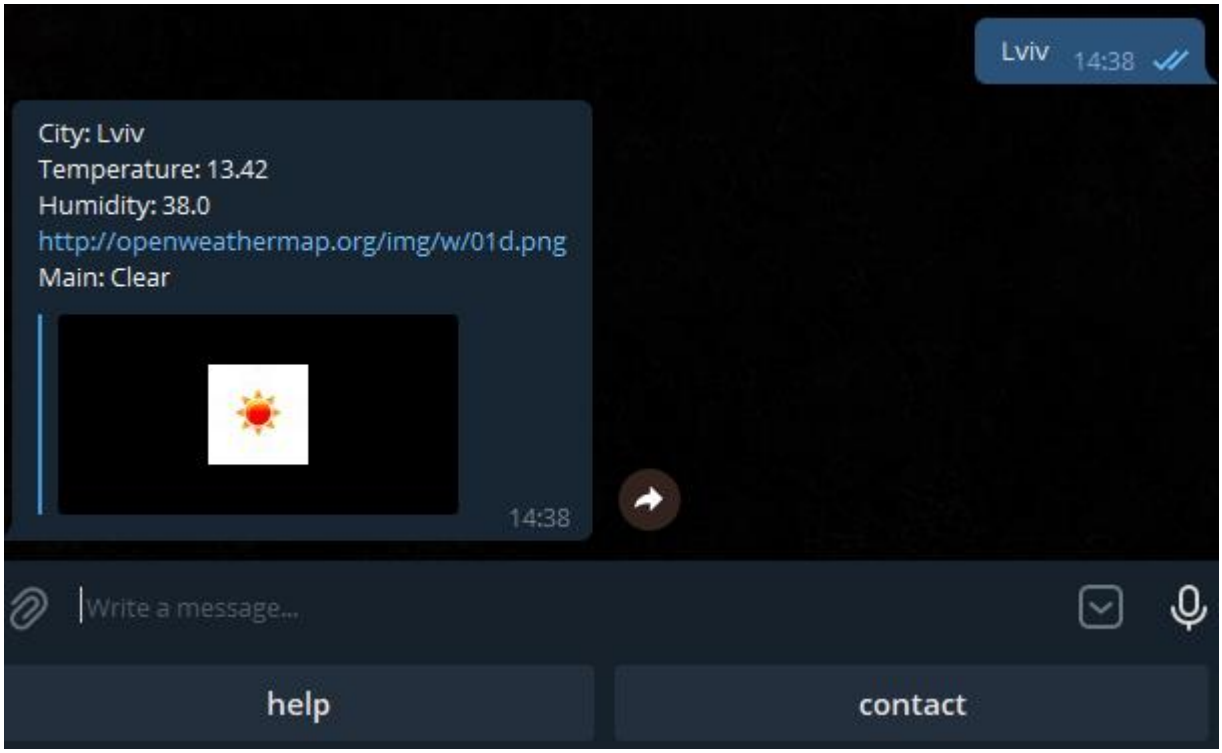


рис. 2

Висновок: Так на простому прикладі ми розглянули наскільки велика перспектива в цій сфері. Для захисту своєї курсової роботи, я також вирішив розробити «Telegram Bot», онлайн магазин, але вже зі значно більшою функціональністю. Також варто згадати про онлайн сервіси, які дають змогу створювати чат бот без технічних знань, за допомогою візуальних mind map, що суттєво скорочує розробку.

Література

[1] https://metanit.com/java/tutorial/

[2] https://github.com/rubenlagus/TelegramBots

[3] https://core.telegram.org/bots

[4] https://uk.wikipedia.org/wiki/Чат-бот

[5] https://habr.com/ru/post/539766

Діагностування захворювань легеневої системи таких як: COVID-19 та SARS-COV-2 використовуючи Computer vision та Convolutional Neural Network

Виконав: Онищенко Орест Андрійович
Керівник: Доц. Огородник Наталя Пилипівна
Львівський національний університет імені І. Франка
факультет прикладної математики та інформатики
Львів 2021

Вступ:

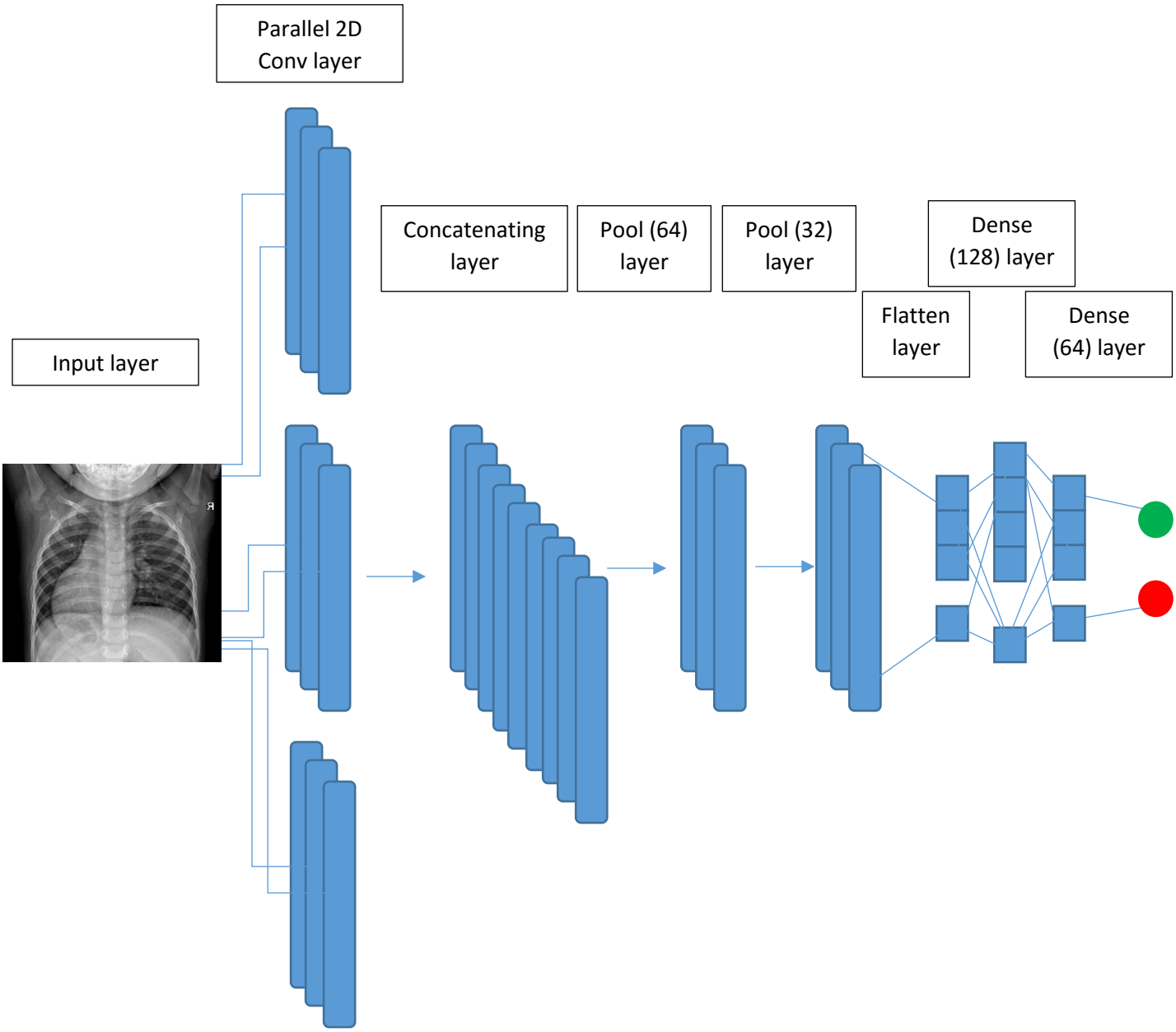
Пандемія коронавірусу в 2020 році застала всіх зненацька. Жодна країна не була готовою до такого масового поширення цього захворювання та більшість не сприймали хворобу всерйоз. Мало хто з нас тоді міг подумати, що виступати на науковій конференції в онлайн-режимі стане звичним процесом. Найбільший удар у боротьбі з вірусом взяли на себе лікарі. Саме тоді я задав собі питання, чи є якась можливість захистити медпрацівників краще, ніж це роблять маски та респіратори. Нейронні мережі вже дуже давно використовуються в медичній галузі для визначення таких хвороб як рак. Тож я спробував реалізувати згорткову нейронну мережу, яка на вході отримує зображення комп'ютерної томографії чи рентгенівський знімок, а на виході покаже ймовірність того, чи є пацієнт хворим.

Реалізація:

Для реалізації такого типу задачі я вирішив не використовувати вже готові натреновані моделі, а натомість створити і натренувати власну. Для такого підходу я вирішив використати Convolutional Neural Network (Згорткову Нейронну Мережу) надалі CNN, яка досить добре справляється з задачею розпізнавання та класифікації об'єктів. Також мною було використано бібліотеку Keras, що працює поверх TensorFlow. Ця бібліотека

містить, так би мовити, будівельні блоки, з яких і будується CNN, таких як Conv layer, Flatten layer, Danse layer та багато інших.

Архітектура CNN:

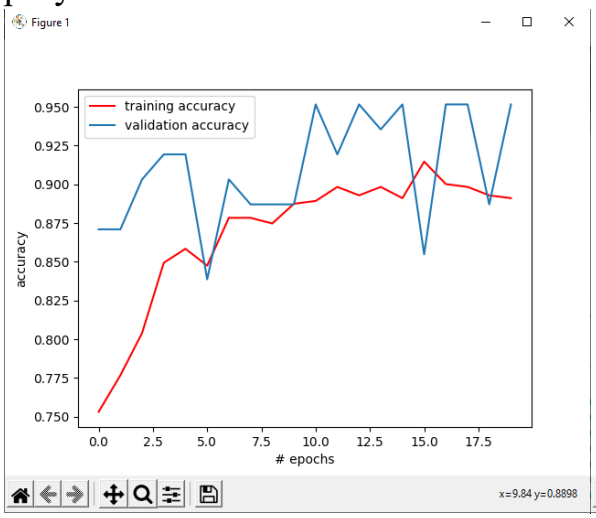


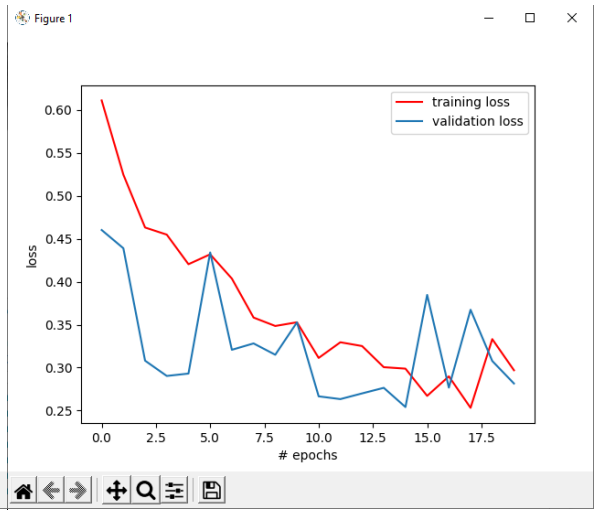
Вхідний шар являє собою зображення для перевірки;
Паралельні Conv2D шари – це шари 128 (3x3/5x5/7x7), де 128 – це число фільтрів в згортці, (3x3, 5x5, 7x7) – кортежі, які вказують висоту та ширину 2D вікна згортки.

Об'єднувальний шар – це шар об'єднання паралельних Conv2D шарів кроком вище.
Pool (64) шар – цей шар являє собою операцію максимального об'єднання 2D просторових даних з числом фільтрів – 64. Цей шар зменшує вибірку вхідного подання, приймаючи максимальне значення у вікні.
Pool (32) шар – цей шар являє собою операцію максимального об'єднання 2D просторових даних з числом фільтрів - 32. Він зменшує вибірку вхідного подання, приймаючи максимальне значення у вікні.
Шар Flatten – являє собою шар згладження. Цей шар згладжує вхід та не впливає на розмір партії.
Dense (128) шар – звичайний щільно з'єднаний шар CNN з розміром вихідного простору 128.
Dense (64) шар – звичайний щільно з'єднаний шар CNN з розміром вихідного простору 64.

Результати:

Після тренування моделі з такою архітектурою я отримав такі результати:





Відношення точності до епохи Відношення втрат до епохи

З даних графіків спостерігаємо те, що зі зростанням епох росте точність та зменшуються втрати.

Використана література та джерела інформації:

1. Keras API reference – https://keras.io/api/;
2. Консультація з лікарем КЛШМД міста Львів;
3. Radiopaedia – https://radiopaedia.org/?lang=us

Квадратична функція регресії з марковськими переключеннями

Янчура Уляна
науковий керівник: проф. Чабанюк Ярослав Михайлович

Україна, Львів

Львівський національний університет імені Івана Франка

факультет прикладної математики та інформатики
Uliana.Yanchura@lnu.edu.ua

Вступ

У даній роботі розглянуто моделювання квадратичної функції регресії залежності попиту від ціни згущеного молока у одинадцятьох торгових точках, у випадку зовнішніх впливів, що мають вигляд ланцюга Маркова з трьома станами.

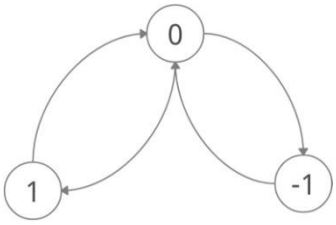
Одна з головних задач наукових досліджень полягає у визначенні зв'язків між різними явищами, які виражені за допомогою алгебраїчних формул, функціональних залежностей. Однак, у соціально-економічних явищах, що відрізняються складністю і різноманіттям існуючих взаємозв'язків, функціональні залежності між величинами зустрічаються вкрай рідко. У цьому випадку між величинами X та Y існує зв'язок, при якому зі зміною однієї величини змінюється розподіл іншої. Такий ймовірнісний зв'язок припускає, що для заданого значення незалежної змінної X можна вказати ряд значень залежної змінної Y, випадково розсіяних на деякому інтервалі. Регресійний аналіз розглядає залежність випадкової змінної Y від однієї (чи декількох) незалежної змінної X [1]. Така залежність може виникнути, наприклад, у випадку, коли при деякому значенні(-ях) X відповідні значення Y піддані випадковому розсіянню за рахунок дії, у даному випадку, зовнішніх впливів, які мають вигляд ланцюга Маркова.

Постановка задачі

Розглянемо завдання моделювання квадратичної функції регресії з марковськими переключеннями. Для цього розглянемо квадратичну функцію регресії вигляду: $\overline{y} = a(x+\xi)^2 + b(x+\xi) + c$, де ξ - зовнішні впливи у вигляді ланцюга Маркова з трьома станами; a, b, c – параметри, що підлягають визначенню.

Нехай стани визначаються наступними значеннями: {-1; 0; 1}. Перехід

системи з одного стану в інший, здійснюється наступним чином:



Час перебування системи у кожному з станів має експоненційний розподіл та обчислюється за формулою :

$$\theta_n = -\frac{1}{\lambda} ln(1 - y_n), \qquad (1)$$

де θ_n - час перебування системи в стані; y_n - рандомно згенероване число на інтервалі [0,1]; $\lambda > 0$ - параметр; n - кількість переходів.

Ланцюг Маркова

Кажуть, що послідовність випадкових величин ξ_1, ξ_2,..., ξ_n,... утворює ланцюг Маркова, якщо ймовірність того, що при n-ому випробуванні (n=1, 2,...) відбулась певна подія E_j (j=1,2,...) залежить тільки від того, яка подія відбулась при (n-1)-ому випробуванні і не залежить від результатів попередніх випробувань. Отже, для ланцюгів Маркова при фіксованому "теперішньому" "майбутнє" не залежить від "минулого". Цю властивість називають марковською властивістю [2].

Вибір квадратичної функції регресії

Регресія — форма зв'язку між випадковими величинами. Розглянемо приклад застосування квадратичної функції регресії в економіці, а саме аналіз індивідуального ринку. Регресійне рівняння попиту залежно від ціни, на деякий вид товару, дає можливість знайти оцінку оптимальної ціни та прийняти оптимальне рішення. Нехай відома таблиця:

P_i	p_1	p_2	...	p_n
D_i	d_1	d_2	...	d_n

де d_i – попит (обсяг продажів) товару за ціною p_i, i = {1; n}, p_i - ціна одиниці товару для i-того спостереження. Тоді між ціною P та величиною попиту D існує залежність $D = \alpha_0 + \alpha_1 P + \alpha_2 P^2$ [3], де оцінки параметрів α_0, α_1, α_2 можна знайти, застосувавши метод найменших квадратів (МНК).

Побудова квадратичної функції регресії

З метою проведення маркетингового аудиту досліджують індивідуальний ринок молочної продукції на основі даних про попит на згущене молоко і його ціну в одинадцятьох торгових точках [4] (Табл. 1).

i	1	2	3	4	5	6	7	8	9	10	11
p_i (ціна, грн/л)	66	46	50	70	76	88	74	99	90	108	135
d_i (попит, тис. л)	8.3	7.28	6.38	6.3	5.49	4.7	3.7	3.2	2.5	1.96	1.56

Табл.1

Попередні дослідження дають підстави стверджувати, що між ціною і попитом є поліноміальна залежність 2-го порядку, тобто регресійна модель попиту має вигляд: $\overline{d} = \alpha_0 + \alpha_1 p + \alpha_2 p^2$. Застосувавши МНК до даних Табл. 1 отримаємо наступні оцінки параметрів : α_0 = 12.6702; α_1 = - 0.1208; α_2 = 0.0003.

Отже оцінена за МНК модель попиту має вигляд:

$$\overline{d} = 12.6702 - 0.1208p + 0.0003p^2.$$

Графік квадратичної функції регресії і розсіювання (Рис. 1):

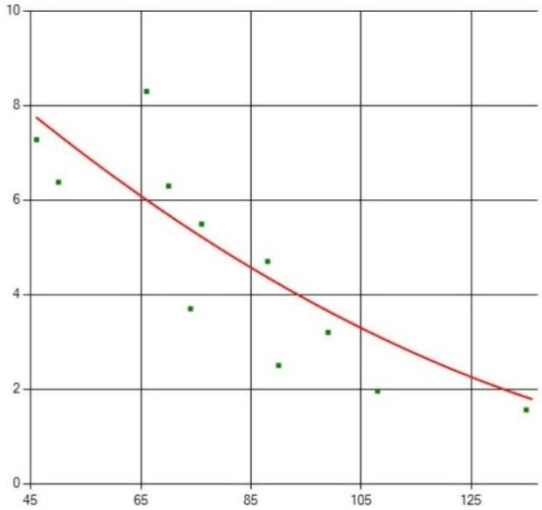


Рис. 1

Для вирішення поставленої задачі було розроблено програму на мові C#. Для знаходження часу перебування системи у кожному із станів θ, який обчислюємо за формулою (1), у groupbox "Input" вводимо значення параметра експоненційного розподілу λ=0.05. В таблицю праворуч виводимо рандомно згенероване число "Y RAND" з інтервалу [0;1]; час перебування системи у кожному із станів θ, доки $\sum\theta$ не дорівнює 89 (max p_i - min p_i) та відповідний стан "State" (Рис.2).

Input

λ 0,05

Calculate

k	Y RAND	θ	State
0	0,7	10,46	1
1	0,9	20	0
2	0,6	7,96	-1
3	0,1	0,92	0
4	0,9	20	1
5	0,4	4,44	0
6	0,5	0,92	-1
7	0,5	6,02	0
8	0	0	1
9	0,5	6,02	0
10	0,6	7,96	-1
11	0,3	3,1	0
12	0,06	1,2	1

Рис. 2

Квадратична функція регресії зазнає зовнішніх впливів, що мають вигляд ланцюга Маркова з трьома станами, і набуває вигляду:

$$\overline{d} = 12.6702 - 0.1208(p + \xi) + 0.0003(p + \xi)^2,$$

де ξ - стан, в якому система перебуває певний час θ (відповідно до (Рис 2)). Побудувавши цей процес, отримаємо графік, який зображений синім кольором (Рис. 3).

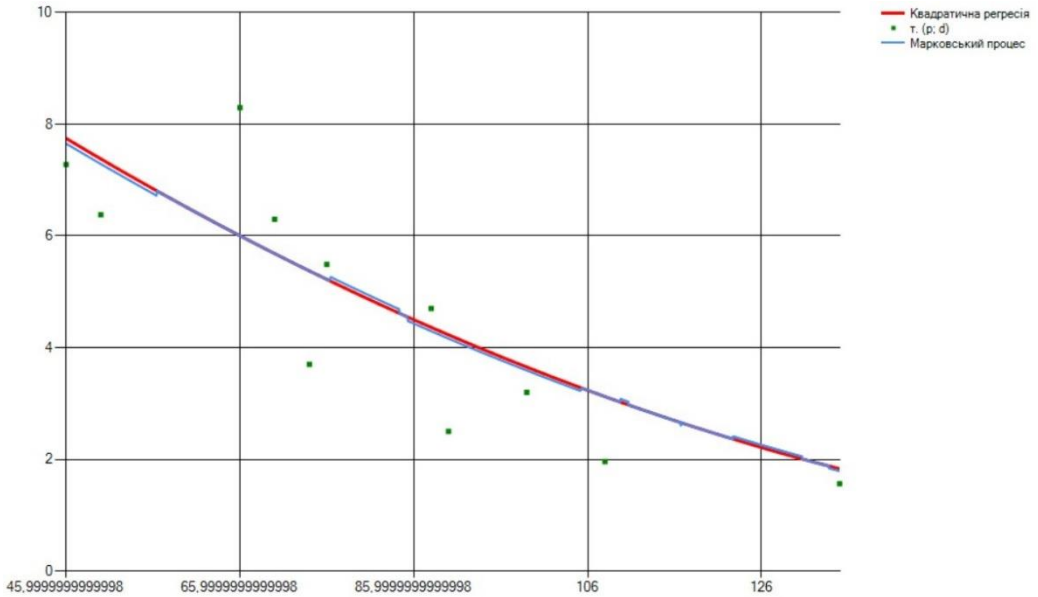


Рис. 3

Висновки

У даній роботі вперше запропоновано побудова квадратичної функції регресії з марковськими переключеннями, що описує залежність попиту згущеного молока від ціни.

Зовнішні впливи відображають випадкові зміни ціни в різних точках продажу. При цьому випадкові зміни пропонуються розглянути у вигляді марковських переключень. Створено програмну реалізацію поставленої

задачі.

В зв'язку з цим, маємо можливість передбачити величину максимального прибутку навіть при випадкових впливах.

Література

[1] Стадник Ю.А Конспект лекцій з навчальної дисципліни Економіко-математичне моделювання. - Львів 2017. - 44 с.

[2] Білущак Г. І., Чабанюк Я. М. Теорія ймовірностей та математична статистика. Лекції. Навчальний посібник для студентів вищих закладів освіти. 1-е,: Львів: В- во "Львівський ЦНТЕІ", 2002. - 569 с.

[3] Шишканова Г. А. Методичні вказівки та розрахунково-графічні завдання для самостійної роботи студентів економічних спеціальностей усіх форм навчання з курсу “Економетрика” – Запоріжжя: ЗНТУ, 2017. – 62 с.

[4] Якимова Л. П. Економетричні методи в аналізі та аудиті. Комп’ютерний практикум : метод. вказівки. – Чернівці: Чернівецький нац. Ун-т. 2017. – 73 с.

Математична модель росту ракової пухлини з запізненням

Коваліско Ольга
Україна, Львів
Львівський національний університет
імені Івана Франка
Факультет прикладної математики і інформатики
olgakovalisko@gmail.com

Відомо, що рак є однією з основних причин смертності у всьому світі, і контроль за зростанням пухлини вимагає великої уваги.
Розвиток ракової пухлини складний і передбачає взаємодію багатьох типів клітин. Основними компонентами цих клітин є ракові клітини (або аномальні клітини), здорові клітини тканин (або нормальні клітини) та відмерлі клітини.
Пухлина - це патологічне зростання клітин, що характеризується безконтрольним розмноженням і низкою інших біологічних особливостей, які мають назву атипізм, синонім анаплазії[1].
Хоча ріст пухлини є складним явищем, що включає багато взаємопов'язаних процесів, її основні механізми можна дуже добре пояснити за допомогою математичних моделей.
Відомо, що ріст пухлини починається з виникнення аномальних клітин, що зазнали генетичної мутації. Такі аномальні клітини, що називаються раковими клітинами, ростуть і піддаються мітозу (непрямий поділ). Ракові клітини розмножуються і відбувається формування новоутворення, що слугує початком розвитку хвороби.
Прогресія ракової пухлини забезпечується кількома умовами. У першу чергу це відбувається за рахунок мутацій у раковій клітині, внаслідок чого виникає велика кількість копій, що призводить до посиленого росту. Така мутована клітина дає набагато більше число дочірніх клітин, ніж здорова клітина. Інші види ракових мутацій забезпечують клітині посилену проліферацію – тобто високу швидкість розмноження. Надалі відбувається пошкодження контролюючих генів клітин, що тільки збільшує кількість мутацій. Це в свою чергу призводить до збільшеної кількості атипових клітин.

В даній роботі розглянуто математичну модель росту ракової пухлини з запізненням в часі.
Розглянуто модель:

$$\begin{cases} \frac{du_1}{dt} = \mu_1 u_1 - \mu_1 u_1 u_3 - \gamma_1 u_1 u_3 - f_1(t) u_1; \\ \frac{du_2}{dt} = \mu_2 u_2 (1 - u_2(t-\tau)) - \mu_2 u_2 u_3 - \gamma_2 u_1 u_2 - \gamma_3 u_2 u_3; \\ \frac{du_3}{dt} = (\gamma_2 u_1 u_2 + \gamma_1 u_1 u_3 + \gamma_3 u_2 u_3)(1 - u_3) - f_3(t) u_3. \end{cases} \quad (1)$$

З початковими умовами:

$t = 0,\ u_2 = 1,\ \ u_3 = 0$, де u_1^0 – мала величина.

Ці початкові умови означають, що в початковий момент часу в функціональному просторі здорових клітин (відмерлі клітини відсутні) виникає невелика кількість ракових клітин.

У даній моделі розглядається три типи клітин:

1. Ракові (ті, що неперервно діляться)
2. Здорові (нормальні)
3. Відмерлі

У системі диференціальних рівнянь (1) u_1 − кількість ракових клітин , u_2 −кількість здорових клітин, u_3 −кількість відмерлих клітин.

Охарактеризуємо кожен доданок системи (1):

1. У **першому** рівнянні:
   - $\mu_1 u_1$ – швидкість «власного» росту ракових клітин;
   - $\mu_1 u_1 u_3$ – шидкість, з якою відмерлі клітини витісняють ракові;
   - $\gamma_1 u_1 u_3$ – швидкість, з якою відмерлі клітини інгібують ракові.
2. У **другому** рівнянні:
   - $\mu_2 u_2 (1 - u_2(t-\tau))$ - швидкість, з якою змінюється кількість здорових клітин;
   - $\mu_2 u_2 u_3$ – швидкість,з якою відмерлі клітини витісняють здорові;
   - $\gamma_2 u_1 u_2$ – швидкість,з якою ракові клітини інгібують здорові;
   - $\gamma_3 u_2 u_3$ – швидкість, з якою загиблі клітини інгібують здорові;
3. У **третьому** рівнянні:

- Швидкість росту відмерлих клітин пропорційна $(\gamma_2 u_1 u_2 + \gamma_1 u_1 u_3 + \gamma_3 u_2 u_3)(1 - u_3)$;
- Множник $(1 - u_3)$ відображає те, що заповнення функціонального простору, який займають відмерлі клітини, відбувається за принципом: чим більше він наповнений, тим повільніше відбувається заповнення;

Варто зазначити, що в системі (1):

- $\mu_1, \mu_2, \gamma_1, \gamma_2, \gamma_3$ – додатні сталі, що характеризують швидкість реакцій;
- $f_1 = f_1(t)$ та $f_3 = f_3(t)$ –невід'ємні функції, що описують надходження хімічних препаратів, що впливають на швидкість росту ракових клітин і на швидкість накопичення відмерлих.

Система (1) має стаціонарні точки:

- Без урахування лікування:

$$M_1(0; 1; 0)$$

$$M_2(0; 0; 1)$$

$$M_3(0; 0; \zeta)$$

ζ –будь-яке число

- З урахуванням лікування:

$$M_1(0; 1; 0)$$

Точки досліджено на стійкість методом Ляпунова[2] (при $\tau = 0$), та за допомогою теорем про стійкість за першим наближенням[3] (при $\tau > 0$)

За допомогою пакета MATLAB[4] отримано числові розв'язки та графіки:

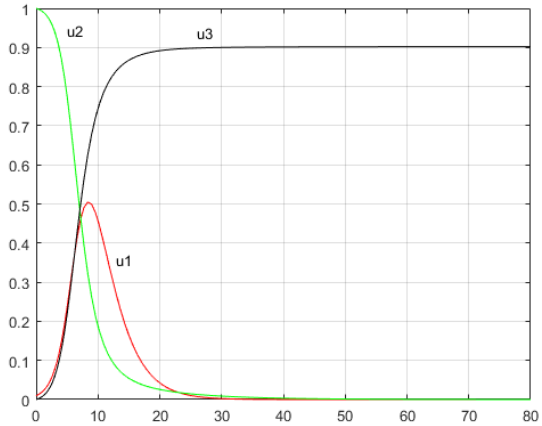
<u>Числові результати без урахування лікування, $\tau = 0$</u>

Розглянемо результати обчислень за відсутності.

Для початкових даних: $u_1 = 0.01$, $u_2 = 1$, $u_3 = 0$, $\mu_1 = 0.7$, $\mu_2 = 0.7$, $\gamma_1 = 0.4$, $\gamma_2 = 0.5$, $\gamma_3 = 0.2$,

$f_1(t) = 0$, $f_3(t) = 0$, $t \in [0; 80]$

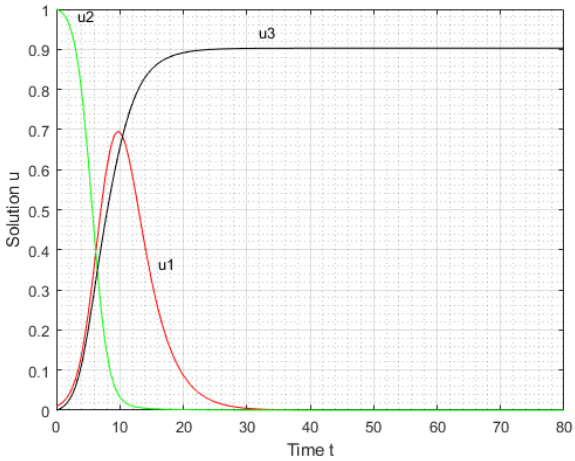
отримано графік:



<u>Числові результати без урахування лікування, $\tau = 5$</u>

Вхідні параметри: $u_1 = 0.01$, $u_2 = 1$, $u_3 = 0$, $\mu_1 = 0.7$, $\mu_2 = 0.7$, $\gamma_1 = 0.4$, $\gamma_2 = 0.5$, $\gamma_3 = 0.2$, $f_1(t) = 0$, $f_3(t) = 0$, $\tau = 5$, $t \in [0; 80]$

Отримано графік:



Список джерел:

[1] Онкологія : підручник друге видання / Шевченко А. І., Колеснік О. П., Шевченко Н. Ф. та ін. /ред. Г.В.Бондар

[2] Борисенко С.Д., Дудкін М.Є. Стійкість розв’язків систем диференціальних рівнянь: Навч.Посібник-К.:НТТУ «КПІ», 2000

[3] Ляшко І.І,, Боярчук А.К., Агай Я.Г, Калейда А.С. Диференціальні рівняння . Київ, Виадвництво Вища школа, 1981 рік, 504 сторінки

[4] https://www.mathworks.com/help/matlab

Програмна реалізація ненавантаженої лінійної багатошарової моделі ґрунту за допомогою методу скінчених елементів

Чумакевич Вікторія
Україна, Львів
Львівський національний університет імені Івана Франка
Факультет прикладної математики і інформатики
victoriachu21@gmail.com

Руйнування будинків у Луцьку (10.06.2012), Тернополі (28.05.2014), Чернігові (12.12.2016), Дрогобичі Львівської області (29.08.2019) та Одесі (10.06.2020) були спричинені численними порушеннями під час проектування та зведення [1, 2, 3]. На думку ряду вчених [1, 2] саме помилки на стадії проектування приводять до найважчих наслідків.

Важливою складовою проектування є дослідження геотехнічної ситуації на будівельному майданчику. Згідно нормативних методик [4, 5], які рекомендуються державними стандартами та будівельними нормами, проектування здійснюється за межовими деформаціями, які реалізують модель пружно-деформованого стану. Найчастіше застосовують пружно-пластичну модель Мора – Кулона [4, 6, 7, 8] та уточнену пружно-пластичну модель [4, 6, 7, 9]. В основу цих моделей покладено закон Гука.

Параметри моделей (властивості ґрунтової основи) визначаються за результатами польових, лабораторних та камеральних робіт, які проводяться із зразками ґрунту на момент взяття проб [4, 5]. Відмітимо, що практично не враховується зміна характеристик та несучих властивостей при зміні навколишніх параметрів, наприклад підйому рівня ґрунтових вод, зміни інтенсивності трафіку транспорту або будівництво нових споруд поблизу об’єкта дослідження. Тому моделювання змін ґрунтової основи майбутнього будівництва є актуальним.

Задачу будемо розв’язувати для випадку плоского деформованого стану.

Маємо O_{xy} - площина деформації. Нехай u, v, w переміщення вздовж осей O_x, O_y та O_z – відповідно. Тоді

$$u = u(x, y), \quad v = v(x, y), \quad w = 0. \qquad (1)$$

Співвідношення закону Гука для ізотропного випадку має вигляд:

$$\sigma_{xx} = D\left(\varepsilon_{xx} + \frac{\upsilon}{1-\upsilon}\varepsilon_{yy}\right), \quad \sigma_{yy} = \left(\frac{\upsilon}{1-\upsilon}\varepsilon_{xx} + \varepsilon_{yy}\right),$$

$$\sigma_{xy} = D\frac{1-2\upsilon}{2(1-\upsilon)}\gamma_{xy}, \quad \sigma_{zx} = \sigma_{zy} = 0, \quad \sigma_{zz} = \upsilon(\sigma_{xx} + \sigma_{yy}) \neq 0,$$

(2)

де $D = \dfrac{(1-\upsilon)E}{(1+\upsilon)(1-2\upsilon)}$, E - модуль пружності (модуль Юнга), ν - коефіцієнт Пуасона.

Приймемо псевдовектори деформацій та напружень:

$$[\varepsilon] = [\varepsilon_{xx}, \varepsilon_{yy}, \gamma_{xy}]^T, \quad [\sigma] = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]^T,$$

Використовуючи співвідношення Коші запишемо закон Гука

$$[\sigma] = [D][\varepsilon], \quad [D] = D\begin{bmatrix} 1 & \frac{\upsilon}{1-\upsilon} & 0 \\ \frac{\upsilon}{1-\upsilon} & 1 & 0 \\ 0 & 0 & \frac{1-2\upsilon}{2(1-\upsilon)} \end{bmatrix}.$$

(3)

Будемо вважати, що масові сили відсутні. Вектор зовнішнього навантаження приймемо $[p] = [p_x, p_y]^T$. Функціонал Лагранжа в матричних позначеннях запишемо таким чином:

$$J([U]) = \frac{1}{2}\int_{\Omega}[\varepsilon]^T[D][\varepsilon]d\Omega - \int_{S_\sigma}[p]^T[U]dS. \qquad (4)$$

Таким чином задача дослідження може бути сформульована наступним чином: використовуючи метод скінченних елементів (МКЕ) [10 – 12] змоделювати поведінку ґрунтової основи в спокійному стані та при навантаженні з використанням лінійних моделей.

Для розрахунку характеристик основи і обробки було написано програму, яку написано на C#. Для розв’язку задачі тріангуляції використали програму Triangle, яку викликаємо з нашого програмного забезпечення.

На початковому етапі ми побудували модель ґрунту без навантаження, яка складалась з одного шару (рис. 1). Потім модель було ускладнено до побудови тришарової моделі ґрунту з певними характеристиками (рис. 2).

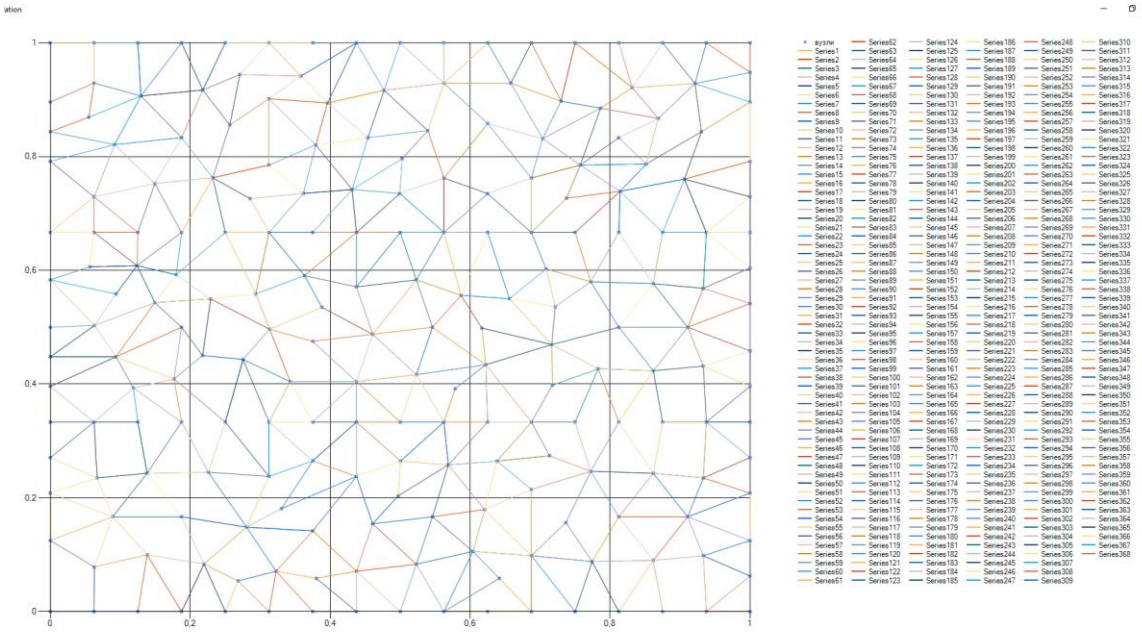


Рис. 1. Модель ґрунту без навантаження, яка складається з одного шару

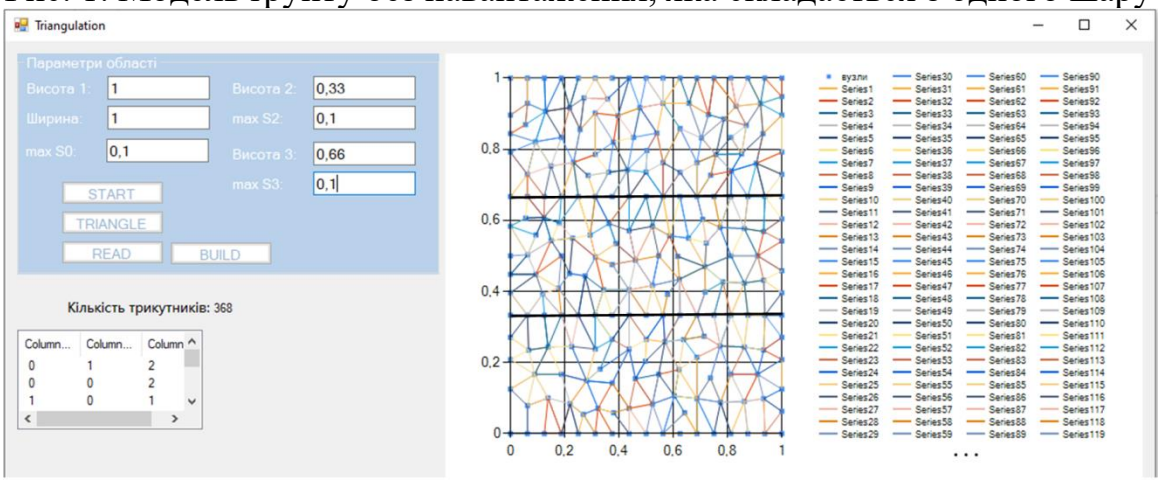


Рис. 2. Модель ґрунту без навантаження, яка складається з трьох шарів

В подальших дослідженнях планується дослідити зміну параметрів моделі ґрунту під дією рівномірнорозподіленого навантаження на частині досліджуваної площі.

ЛІТЕРАТУРА

[1] Majowecki 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] Мельчаков А.П., Чебоксаров Д.В. Прогноз, оценка и регулирование риска аварии зданий и сооружений: теория, методология и инженерные приложения. Монография. Челябинск: УГТУ-УПИ, 2009. – 384 с.

[3] 5 найгучніших обвалів будинків в Україні Інтернет ресурс Дивись.info. Режим посилання: https://dyvys.info/2019/08/29/5-najguchnishyh-obvaliv-budynkiv-v-ukrayini-foto-video/ (дата звернення: 11.11.2020)
[4] Корнієнко М. В. Основи і фундаменти: навчальний посібник. – К.: КНУБА. 2012. – 164 с.
[5] ДБН В.2.1-10-2009: Об'єкти будівництва та промислова продукція будівельного призначення. Основи та фундаменти споруд. Основні положення проектування. [Чинний від 2009-01-07]. – К.: Мінрегіонбуд України, 2009. – 104 с.
[6] Mazindrani, Z.H., and Ganjali, M.H. 1997. Lateral earth pressure problem of cohesive backfill with inclined surface. Journal of Geotechnical and Geoenvironmental Engineering, ASCE, 123(2): 110-112.
[7] Arnold Verruijt: Soil mechanics, Delft University of Technology, 2001, 2006.URL: http://geo.verruijt.net/ (переглянуто 12.05.2020)
[8] Бугров, А.К. О решении смешанной задачи теории упругости и теории пластичности грунтов. *Основания, фундаменты и механика грунтов,* № 6, 1974. с. 20-23.
[9] Schanz, T., Vermeer, P.A., Bonnier, P.G. 1999. The Hardening Soil Modell – Formulation and Verification. *Proceedings Plaxis Symposium "Beyond 2000 in Computational Geotechnics",* Amsterdam, pp.55-58. Rotterdam: Balkema.
[10] Савула Я.Г. Числовий аналіз задач математичної фізики варіаційними методами. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2004. – 221 с.
[11] Большаков В. И., Яценко Е. А., Соссу Г. И др. Основы метода конечных элементов. – Днепропетровск: ПГАСиА, 2000. – 255 с.
[12] Киричевский В. В., Толок В. А. Метод конечных элементов и суперэлементов к трехмерным задачам механики. – К.: Наукова думка, 2001. – 364 с.

Development of video chat website using WEBRTC

Bogdan Petrovsky

Lviv, Ivan Franko National University of Lviv

Faculty of applied mathematics and informatics

bpetrovsky977@gmail.com

Introduction

The COVID-19 pandemic has provided and continues to provide huge challenges to our public health system. One challenge has been behavioral, in that mitigating the consequences of the disease involves huge behavioral changes and long term cooperation on the part of the public at large. Another challenge is communicational. Business communications are more important than ever but the environment in which we are producing them has never been more complex. Same goes for informal communications between people themself.

That is why I dedicated time and affords into getting in touch with and implementing software solutions which allow web real time communication.

Solution

After a long time of investigating modern software solutions and frameworks I came up with a choice of creating a website video chat. As a front-end I chose Angular - powerful framework for creating Single Page Applications (SPA), as a back-end I picked Node.js - an open source platform for high-performance networking applications written in JavaScript. I also used Socket.IO - JavaScript library for web applications and real-time data exchange. It consists of two parts: the client, which runs in the browser and the server for node.js.

I have listed down the reasons why I choose Angular for web application development:

- Dependencies Handling - Angular makes it easy to arrange things like dynamic loading and dependencies, takes care of the complete life cycle of the objects thus handling dependencies in an exceptional way.
- Feature Rich - it has features like filters, data binding, scope management, directives, API client, form validation.
- MVC Framework - Angular implements MVC architecture to develop web applications.
- Faster Development - it contains many ready made tools and components that would reduce the development time, also in each version release of angular it is made simpler and faster.
- Support from Google - Angular is backed up by Google, hence acts as a great relief for the developers that they would be working on a strong code base that would provide complete support to the application.
- Ease to Test - An Angular web application is broken down into modules and sub parts which makes each module easily testable and manageable.

Reasons I picked Node.js for backend:

- it makes it really fast to build real-time, high-traffic apps (eg. chats or gaming)
- It makes it possible to code in JavaScript for both the client and server side
- it increases the efficiency of the development process as it fills the gap between frontend and backend developers
- the ever-growing NPM (Node Package Manager) gives developers multiple tools and modules to use, thus further boosting their productivity,
- code executes faster than in any other language,

- Node is perfect for micro services which are a popular solution among enterprise applications.

Conclusion

Therefore, my solution is made of up-to-date technologies without additional complexity, but has decent scalability and serves its main purpose - allows users to communicate with each other.

References

1. https://angular.io/docs
2. Jennings, Cullen, Ted Hardie, and Magnus Westerlund. "Real-time communications for the web." Communications Magazine.
3. https://nodejs.org/en/docs
4. Fette, Ian, and Alexey Melnikov. "The websocket protocol." (2011).
5. https://socket.io/docs/v4
6. Bergkvist, Adam, D. Burnett, and Cullen Jennings. "A. Narayanan," WebRTC 1.0: Real-time Communication Between Browsers." World Wide Web Consortium WD WD-webrtc-20120821 (2012).
7. https://developer.mozilla.org/ru/docs/Web/API/WebRTC_API

Розробка системи для онлайн замовлень їжі з використанням REACT.JS та FLASK

Калітовський Роман
Україна, Львів
Львівський національний університет імені Івана Франка
Факультет прикладної математики і інформатики
roman.kalitovskyy@gmail.com

Зараз активно розвиваються веб технології, а також у зв'язку з пандемією гостро постає питання покупок як речей так і необхідних товарів, таких як їжа.

З кожним днем все більше операцій відбувається в онлайн режимі, особливо це актуально для власників магазинів та організацій, що займаються доставкою, оскільки через пандемію і карантин вони не мають можливості відкривати свої заклади, а працюють лише на виніс. Вагомою перевагою даного переходу є те, що не потрібно орендувати приміщення для магазину чи кафе. Достатньо мати склад (і кухню) а також свою доставку чи використати вже готовий сервіс. Тепер веб-сайт — це зручний і звичний спосіб спілкування, який допомагає розширити та модернізувати бізнес.

У даній роботі розроблено онлайн сервіс для продажу випічки. В цей додаток будуть вбудовані нейромережі для аналізу пріоритетів як кожного покупця, з метою створення системи акцій чи знижок на ті чи інші, актуальні для нього товари, так і для загального аналізу тенденції покупок. Для реалізації back-end частини було обрано Flask, а для клієнтської - React.

Чому Flask? Для реалізації back-end обрано мову Python, оскільки це потужна мова програмування, якою легко оволодіти. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування [9]. Вибір був між такими популярними фреймворками як Flask та Django. Вирішено вибрати Flask, оскільки він дає змогу створювати прості веб-служби з мінімумом шаблонного коду, а також згодиться як компонент для розробки складних веб-застосунків [10]. На відміну від Django, в якому "з коробки" є багато функціоналу який не є завжди доречний і який не можливо в разі чого

замінити чи видалити, у Flask усе, що необхідно для створення додатку можна підключити і гнучко налаштувати. Також Flask не обмежує в певній архітектурі, а отже, можна якісно і швидко розробляти додаток під поточну задачу. Хоча Flask не дає дуже високу продуктивність як, наприклад, додаток на Spring WebFlux, який буде мати як мінімум в 10 разів більшу продуктивність, але розробка на подібних фреймворках є набагато довшою. У цьому веб-ресурсі користувачі можуть робити замовлення, реєструватись. В особистому кабінеті є можливість переглядати історію покупок, а також стан конкретного замовлення.

Чому React? React дозволяє гнучко проектувати Frond-end частину. Він не нав'язує певної строгої структури побудови клієнтської частини. React реалізує компонентний підхід, тобто кожен компонент повертає частину інтерфейсу з певним станом. Ця бібліотека дозволяє швидко розробляти SPA (Single page application), описувати різні стани компоненти інтерфейсу, що дозволяє скоротити код і зробити його більш зрозумілим. До важливих особливостей React відноситься і використання віртуального DOM, що дозволяє при зміні інтерфейсу малювати тільки ту частину, які змінилась. Описані вище переваги спонукали до вибору React для написання додатку і можливістю подальшого використання React-Native для розробки мобільного застосування, адже вони набирають величезну популярність. Використання React-Native дозволяє писати ресурс, використовуючи JS і без необхідності вивчення додаткових мов чи фреймворків, що є оптимізований як для IOS, так і для Android. Для менеджменту і контролю даних на клієнтській стороні використано бібліотеку Redux.

Отже, було розроблено систему для онлайн замовлень їжі. Передбачається розширення веб-застосування для подальшого використання нейромереж з метою збільшення прибутку. Цей ресурс легко зможе підтримувати невелика команда розробників. React-Native допоможе суттєво скоротити час необхідний для розробки мобільного додатку.

Література

[1] https://flask.palletsprojects.com/en/1.1.x/
[2] https://uk.reactjs.org/docs/getting-started.html
[3] https://redux.js.org/
[4] https://flask-restful.readthedocs.io/en/latest/
[5] https://flask-sqlalchemy.palletsprojects.com/en/2.x/
[6] https://dev.mysql.com/doc/
[7] React in Action by Mark Thomas

[8] Flask Web Development: Developing Web Applications with Python 1st Edition by Miguel Grinberg
[9] Guido van Rossum, Fred L. Drake, Jr. - The Python Tutorial
[10] Стаття: Безсерверні веб-застосунки на Python з використанням Lambda і Flask

Навігація на мапі отриманій з супутникових зображень за допомогою нейронної мережі

Химера Богдан Борисович

Львівський національний університет ім. Івана Франка

Факультет прикладної математики та інформатики

Bogdankhymera@gmail.com

Актуальність. Отримати фото вулиць міста, або заплутаних гірських доріг стало легко з розвитком персональних дронів і компактних камер з роздільною здатністю високої точності. Крім цього, в широкому доступі знаходяться терабайти супутникових зображень. Однак, цих даних не достатньо для генерування надійних засобів навігації, адже зображення зроблене під таким кутом не дає необхідної інформації і допускає неточності. Наприклад, не всі лінії на фото є прохідними дорогами, або лінія перервана широкою кроною дерева або тунелем не є закінченою дорогою. Саме тому, такі сервіси як Google Maps та Here створювалися за допомогою зовнішніх компаній та урядових організацій, що значно ускладнює процес розробки і робить його недоступним для проектів меншого масштабу.

Мета роботи. Ідеєю проекту є створення сервісу, здатного швидко генерувати схему доріг або маршрутів з вхідного зображення, та шукати найкоротший шлях між точками, вказаними користувачем.

Реалізація. Реалізацію цього проекту можна поділити на три модулі:

- Отримання бінарного зображення доріг за допомогою нейронної мережі
- Обробка бінарного зображення для побудови точного графу

- Опрацювання запитів користувача на пошук найкоротшого шляху

Для розпізнавання схеми доріг використана нейронна мережа, оскільки вона дозволяє працювати з великим об'ємом даних. Для тренування використана база зображень доріг Массачусетса, та бінарні зображення, які вказують чи точка (x,y) (де x,y - координати пікселя на початковому фото) належить дорозі, чи ні. Набір даних був розділений на набори для тренування (1108 фото), тестування (49 фото) та валідації (14 фото) Набором вхідних значень мережі є вектор з (r,g,b) значень кожного пікселя вхідного зображення. Для реалізації нейронної мережі використаний DNNClassifier з бібліотеки Tensorflow, з набором прихованих шарів 100, 150, 100, 50, та двома класами, які вказують чи піксель є дорогою. На наборі для тестування досягнута точність 83%. Результатом є бінарна маска вхідного зображення.

Через неточності мережі на вихідному зображенні можуть бути шуми і дефекти, для усунення яких використовується набір фільтрів, таких як медіанний, ерозія та розширення. Наступним кроком є створення графу на схемі доріг, адже для графу, на якому здійснюватиметься пошук найкоротшого шляху ширина дороги не матиме значення. Для цього до вихідного зображення застосовується алгоритм скелетування Zha84. Після цього з отриманого зображення складається граф. Вузлами графу є точки розгалуження, перегину або закінчення доріг. Пошук найкоротшого шляху здійснюється за допомогою алгоритму Дейкстри.

Для опрацювання запитів користувача використовується .NET MVC застосунок, який вираховує координати точок позначених на зображенні та передає їх в модуль роботи з графом. Результатом є схематичне зображення маршруту шляху між точками.

Результати та висновки. Отримана система здатна опрацьовувати вхідне зображення без додаткових інструкцій та специфікацій від користувачів, що дозволяє її використання в системах автоматизації та обробки великих об’ємів даних.

Література.

- *Volodymyr Mnih*, Machine Learning for Aerial Image Labeling, University of Toronto 2013.
- *Aurélien Géron,* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly Media, Inc. 2019.
- *Saurabh Kapur*, Computer Vision with Python 3, Packt Publishing 2017.
- *Santanu Pattanaya*, Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python; 2019.

OpenGL на прикладі кубика-Рубіка

Горбань Андріан

Україна, Львів

Львівський національний університет імені Івана Франка

Факультет прикладної математики і інформатики
horbanpiff@gmail.com

Вступ

OpenGL.OpenGL —(англ. Open Graphics Library —відкрита графічна бібліотека)—специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс для написання застосунків, що використовують 2D та 3D комп'ютерну графіку.

Цей інтерфейс містить понад 250 функцій, які можуть використовуватися для малювання складних тривимірних сцен з простих примітивів.

Бібліотека OpenGL досить проста у використанні і навчанні, має дуже широким спектром можливостей. Ось деякі з її переваг:

- Стабільність. OpenGL - стандарт. Всі зміни, що вносяться до нього, анонсуються заздалегідь і реалізуються так, щоб вже існуюче ПЗ не давало збій на нових графічних картах.

- Надійність. Всі функції, які використовують OpenGL, гарантують однаковий візуальний результат, незалежно від устаткування і операційної системи.

- Переносимість. Програми, що використовують OpenGL, можуть запускатися на різних архітектурах і під різними операційними системами (відповідно, за умови перекомпіляції додатку, тобто OpenGL забезпечує переносимість на рівні вихідних кодів)

Отже, використовуючи програмне середовище Python та адаптацію OpenGL для даної мови PyOpenGL та інші бібліотеки ми реалізуємо відому всім 3D гру «Кубик-Рубіка».

Задача

Поставлену задачу можна поділити на декілька етапів:

1. Підготовка та базове уявлення ігрової логіки та графічного інтерфейсу
2. Реалізація ігрової логіки без використання графіки
3. Відображення гри користувачеві з використанням OpenGL

- Перший етап водночас складний та легкий, нам потрібно підготуватися, зрозуміти яка перед нами ціль та якими засобами її досягти. Як ви вже зрозуміли, ми будемо використовувати графічну бібліотеку OpenGL, також для полегшення взаємодії з користувачем я буду використовувати python-бібліотеку PyGame, хоча й OpenGL надає нам ці можливості. Далі обдумаємо ігрову логіку, нам потрібно рухати віссями, реалізувати випадкову генерацію кольорів кубів і просто намалювати куб.

- На другому етапі ми вже за допомогою нашого середовища програмування оголошуємо потрібні змінні та константи, ініціалізуємо вікно та івенти, реалізуємо різноманітні функції, зокрема можливість перетасовувати кубик, оскільки на початку гри він буде зібраний.

- Третій етап найважчий на ньому ми повинні реалізувати графіку гри. Як вже було зазначено OpenGL для цього використовує примітиви, тобто квадратики, які в нашому випадку ідеально підходять для реалізації куба. Малювати будемо кожен елемент куба, маленький кубик, будемо окремо, загалом треба намалювати 27 кубів, також добавимо анімацію.

Висновок: було повністю розроблено ігрову логіку та інтерфейс гри за допомогою відкритої графічної бібліотеки OpenGL та програмного середовища Python, використовуючи допоміжні бібліотеки.

Література

[1]https://learnopengl.com/Getting-started/OpenGL

[2]https://www.khronos.org/opengl/wiki/

[3] John Kessenich, Graham Sellers, Dave Shreiner. Opengl programming guide: the official guide tolearning opengl, version 4.5 with spir-v

Визначення мотивації кандидатів працювати у компанії на основі аналізу їх характеристик із використанням машинного навчання

Клакович Ксенія

ЛНУ ім. Івана Франка

Факультет прикладної математики та інформатики

kklakovych@gmail.com

З появою електроніки розпочались перші спроби апаратного відтворення процесу мислення людини шляхом створення нейронних мереж, які здатні встановлювати складні зв'язки між елементами і приймати рішення. Зокрема, можливість навчання — одна з головних переваг нейронних мереж над традиційними алгоритмами. Використання нейронних мереж дає змогу автоматизувати такі складні процеси, як, наприклад, підбір команди. Процес аналізу характеристик потенційних працівників є досить складним, адже потрібно врахувати багато чинників, таких як: відповідність фаху, досвід роботи кандидата, рівень освіти, розуміння предметної області, тощо. [3]

Багато великих компаній часто проводять курси, після яких потенційних кандидатів беруть на роботу. І хоча навчання проходять багато людей, часто не всі з цих кандидатів насправді хочуть працювати в компанії після навчання або ж задумуються над пошуком нової роботи в іншій компанії, що не вигідно для останньої. Тому аналіз кандидатів допомагає скоротити витрати та час, а також підвищити якість навчання та планування курсів. Цей процес відбору можна комп'ютеризувати, побудувавши модель, яка, використовуючи поточні дані, демографічні показники, прогнозуватиме ймовірність того, що кандидат шукатиме нову роботу або буде працювати у компанії.

Метою досліджень є підтвердження гіпотези про те, що можна класифікувати працівників при прийнятті на роботу використовуючи для

цього алгоритми машинного навчання. Досягнення мети досліджень передбачає розв’язання таких завдань: теоретичний аналіз даних наукової літератури щодо алгоритмів класифікації, реалізація програми із застосуванням цих методів, порівняння результатів із розв'язанням цієї задачі іншими алгоритмами.

Одним із завдань курсової роботи є комп’ютерна реалізації програми для класифікації вхідних даних, для чого було підготовлено корпус [1] - набір характеристик кандидатів. Цей набір даних призначений для розуміння факторів, які змушують людину шукати нову роботу або ні, а саме: стать, місто кандидата, досвід, основний напрямок роботи, тощо. Вихідні дані – це ймовірність кандидата працювати у компанії у відсотках.

На наступному кроці було здійснено візуалізацію тренувальних даних, що дозволило здійснити початковий аналіз. Таким чином, отримано графіки та діаграми, які відображають розподіл навчальних даних при різних рівнях освіти кандидатів, статі та досвіду роботи. Із закономірностей візуалізованих даних зроблено деякі висновки наступного типу: якщо кандидат – чоловік з міста Львів, не має відповідного досвіду і рівень його освіти бакалавар – то кандидат буде найнятий на роботу; якщо місто – Київ, індекс розвитку міста 0.9, стать жіноча, і кандидат має відповідний досвід роботи, то рішення також буде здійснене на користь кандидата, тощо.

Для програмної реалізації поставленої задачі було використано найпопулярніші алгоритми машинного навчання, а саме: метод опорних векторів, дерево рішень, алгоритм Random forest(випадковий ліс), логістична регресія та наївний Баєсовий класифікатор. Для цього використовувалась мова програмування python, а також бібліотеки pandas - для зчитування даних корпусу з файлу; matplotlib - для візуалізації отриманих результатів; sklearn – для створення та тренування моделей із використанням різних алгоритмів.

Після того, як програма проаналізувала вхідні дані з готовими відповідями, потрібно перевірити на нових даних, як точно можна отримати рішення. Для тестування програми використовуються дані нової вибірки, які містять характеристики кандидатів, які влаштовуються на роботу. На цьому етапі використовується модель, отримана внаслідок аналізу тренувальної вибірки, та її результати. Далі отримуємо ймовірність, з якою людина, маючи ці характеристики, залишиться працювати в компанії.

Внаслідок тренування та тестування моделей із використанням різних алгоритів було отримано їхнє порівняння за такими показниками, як точність, оцінка ROC-кривої та повнота, а також побудовано матрицю помилок для кожної моделі [2].

Таким чином, ефективність моделі тренування та тестування після використання алгоритму Random forest була найвищою. Із усіх даних корпусу 3630 рядки модель спрогнозувала як такі, коли кандидат буде працювати в поточній компанії, а 624 – шукатимуть нову роботу, що було правильно. Також 481 одиниць набору даних модель класифікувала як такі, коли отриманою ціллю буде пошук нової роботи, і 670 як такі, коли кандидат буде працювати далі у команії, і це були неправильні рішення.

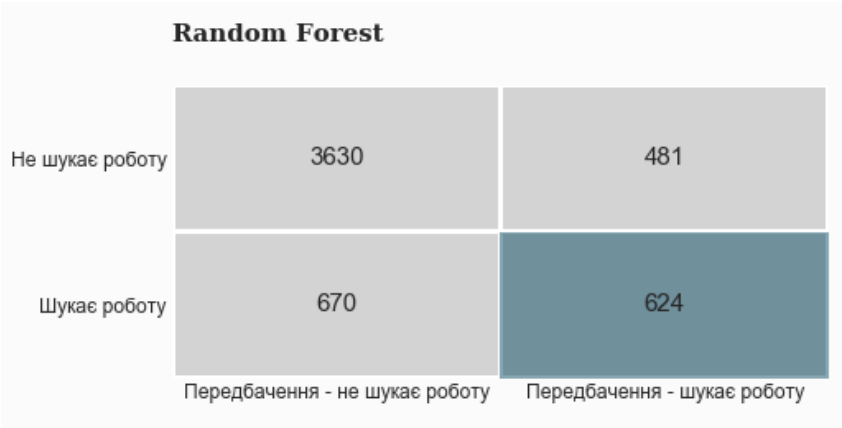


Рис 1: Матриця помилок при тренування моделі із використанням алгоритму випадковий ліс.

Під час комп’ютерної реалізації задачі класифікації з’ясовано, що точність - найважливіший показник оцінки якості здатності класифікатора обробляти фактичний потік отриманих текстів; у всіх випадках при тренуванні моделі лежить в межах від 73 до 78% і корелюється в залежності від об’єму бази знань [2]. Проте це не означає, що більша кількість таких даних спричинить більшу точність обчислень. У деяких випадках це навпаки може викликати перенавчання. Це означає, що класифікатор добре працює на навчальному наборі даних, але не може узагальнити достатньо, щоб він міг передбачити правильний результат для небачених раніше даних.

Крім того, ефективність моделі тренування та тестування після використання алгоритму Random forest [4] була найвищою серед

порівняння результатів класифікації методом опорних векторів, логістичної регресії, дерева рішень та із застосуванням наївного байєсівого класифікатора.

Під час подальших досліджень планується вдосконалення програм, а саме досягнення більшої ефективності розрахунків, які можна досягти кількома способами. Одним з методів є покращення набору вхідних даних, оскільки корпус є незбалансованим. [1] В майбутньому робота програми буде розширена таким чином, щоб класифікувати працівників з більшою кількістю характеристик.

Наступним способом збільшення ефективності розрахунків, який планується бути застосованим у майбутньому, є покращення самих алгоритмів, експериментування із зміною параметрів обчислень, дослідження інших їх модифікацій.

Отже, для підбору висококваліфікованого персоналу можемо запропонувати роботодавцям розробляти заходи, в які включати застосування машинного навчання. Тоді, зважаючи на важливі чинники, які свідчать про компетентність кандидатів, можна буде здійснювати ефективніший відбір персоналу.

Список використаних джерел:

[1] Jalaj Thanaki. Python Natural Language Processing: Advanced machine learning and deep learning techniques for natural language processing. - Mumbai, 2017. - 456 с.

[2] Taweh Beysolow II. Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. - San Francisco, California, USA, 2018. - 150 с.

[3] Rhea Moutafis. Is AI coming for your job? [Електронний ресурс]. - Режим доступу: URL: https://builtin.com/artificial-intelligence/ai-workers-jobs

[4] Machine Learning Classification – 8 Algorithms for Data Science Aspirants [Електронний ресурс]. - Режим доступу: URL: https://data-flair.training/blogs/machine-learning-classification-algorithms/

Проектування та розробка прототипу пристрою Автоматизованого сортування сміття

Олег Леончук
Україна, Львів, класична гімназія при Львівському національному університеті імені Івана Франка
leonshul@ukr.net

Вступ

Однією з найсерйозніших екологічних проблем, з якими сьогодні стикається Україна та низка інших країн світу, є поширення різних видів відходів через збільшення кількості населення та відсутність стійкої політики поводження з відходами. За приблизними підрахунками в Україні проживає 41 мільйонів населення, а дослідження показують, що щорічно в Україні утворюється близько 17 мільйонів тонн побутового та промислового сміття. Це означає, що кожна людина утворює від 0,9 до 1,3 кг відходів на день. В Україні на звалища потрапляє майже 95% відходів і лише маленька частка спалюється чи йде на вторинну переробку. За даними Шостого національного повідомлення України з питань зміни клімату, до структури твердих побутових відходів (ТПВ) входять харчові відходи – 35-50%, папір і картон – 10-15%, вторинні полімери – 9-13%, скло – 8-10%, метали – 2%, текстиль – 4-6%, будівельні відходи – 5%, деревина – 1% та інші відходи – 10%. Значну частину сміття викидають заклади громадського харчування. 60% даного сміття – це органіка, яка у випадку правильного сортування може бути використана як добриво, або енергетична одиниця. Ручна обробка може забезпечити вищу якість матеріалу для відновлення; однак вона не ефективна через низькі показники сортування. Більше того, ручне сортування дає більше забракованих

матеріалів. Наприклад, робітники не можуть розпізнати значну частину видів пластмас ПНД та ПЕТ. Подібним чином, якщо працівник не може класифікувати пластичну смолу неозброєним оком, її неможливо відсортувати вручну. Актуальність розробки також зумовлена відсутністю ефективних та доступних пристроїв для автоматизованого сортування.

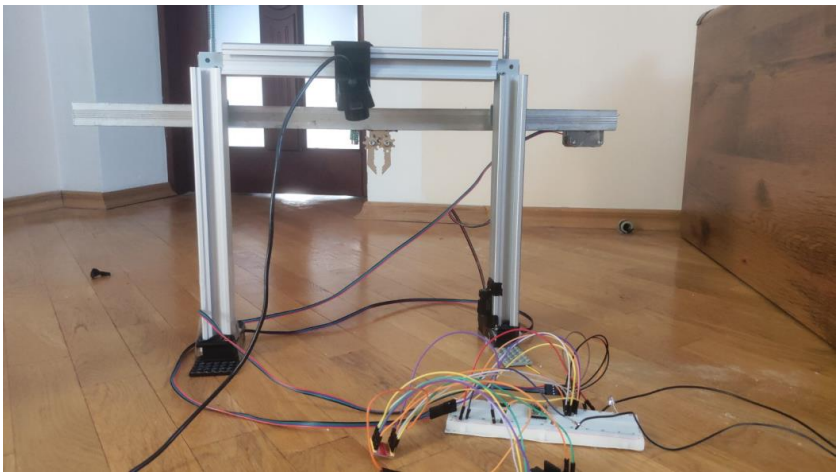


Рис. 1. Робитизований маніпулятор

Апаратна складова пристрою

Складові пристрою – це давачі, двигуни та програмований логічний контролер. Основними давачами, що використовуються в системі, є індуктивний давач наближення, резистивні пластини та ІЧ-давачі.

Спершу було зроблено конвеєр, який доставляє матеріали зі сканерів до необхідних місць. За допомогою алгоритмів, сукупність давачів і модулів розпізнають сміття. Одні давачі розпізнають матеріали за ємністю. Інфрачервоні – вираховують густину об'єкта, який приближається. Також присутній індуктивний давач для виявлення металів. (Присутній поділ металів на феромагнетики та діамагнетики). Показники всіх давачів опрацьовують два мікроконтролери.

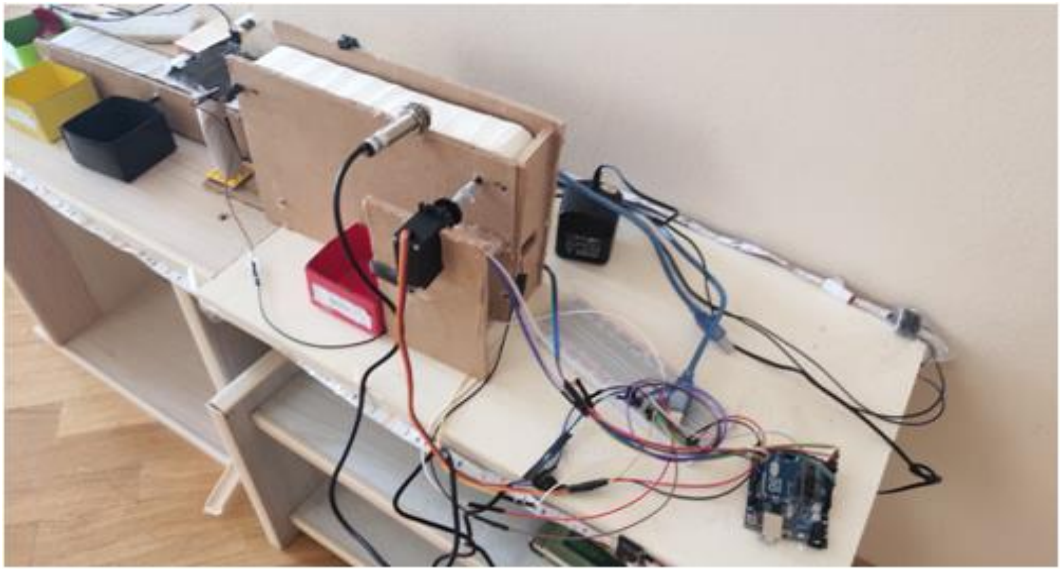


Рис. 2. Робочий прототип пристрою

Рух конвеєра контролюється сервоприводом з постійною швидкістю обертів. Для визначення наявності сміття на конвеєрі використовується інфрачервоний давач. Біля смуги конвеєра знаходиться індуктивний давач приближення, який реагує коли метал приближається до нього. В момент виявлення металу конвеєр зупиняється і очікує допоки метал не усунеться. Це відбувається за рахунок переміщення магніта, який зсуває в одну сторону метали парамагнетики та діамагнетики, а в іншу сторону витягує феромагнетики.

У випадку, якщо відходи не металеві, вони рухаються дальше по конвеєру і попадають на ємнісну панель, на якій визначається заряд елемента, на даній спрацьовує сервомотор, якщо він обертається вліво то скидає з пластини органіку, якщо вправо то скидає інші відходи на другий конвеєр.

Коли сміття припадає на другий конвеєр його виявляє інфрачервоний давач лінії і вводить всі інші давачі в режим сканування. Першим об'єкт проскановує давач KY-026, також об'єкт просвічується лазером, з іншого

боку знаходиться давач освітлення, який реагує на пропадання лазера. В подальшому об'єкт попадає на сканування інфрачервоного сканера (довжина хвилі якого 960нм). Давач визначає пластик чи скло.

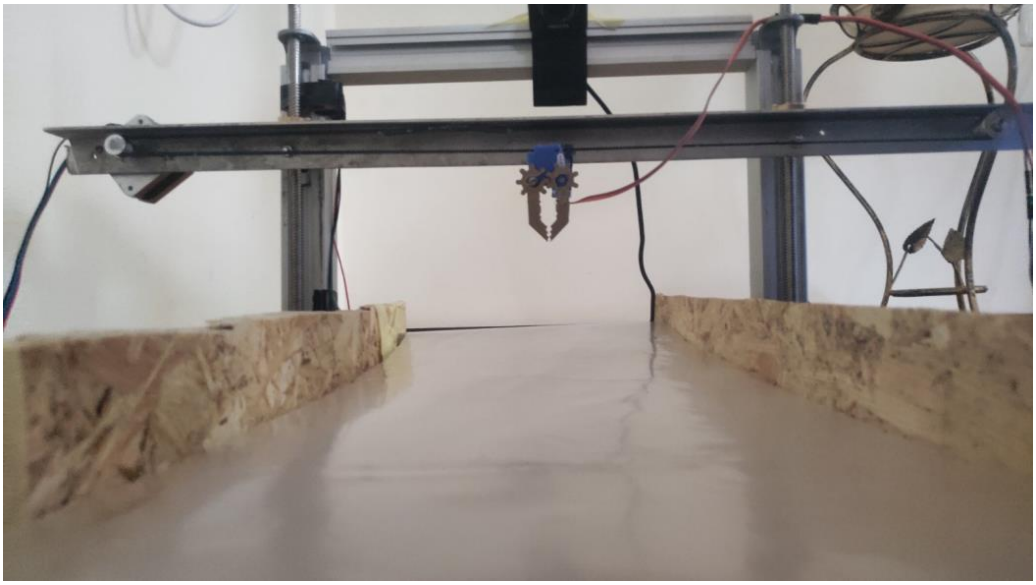


Рис. 3. Маніпулятор разом з власним конвеєром

Маніпулятор може переміщуватися як вертикально, так і горизонтально. Він має прилаштовані модулі розпізнавання кольору, та відстані, а також три інфрачервоні давачі відстані, які відповідають за симетричне переміщення осі з робо-клешнею. Для вертикального підйому та переміщення робо-клешні використовують крокові двигуни з драйверами.

Висновки

У даній роботі розлянено способи автоматичного сортування відходів та створено прототип пристрою, який надає змогу сортування без людей. Розпочато роботу над покращенням пристрою, зараз налагоджується робота

пристрою та бібліотеки OpenCV для легкого розпізнавання побутових об’єктів (пляшки, одноразові маски, тощо).

Також створюється оптимальна система для роботи з гідравлічним підйомником та роборукою.

Список використаних джерел

1. Гроші зі сміття: як на нелегальних сміттєзвалищах роблять бізнес - 24 Канал. URL : https://24tv.ua/groshi_zi_smittya_yak_na_nelegalnih_smittyezvalishhah_roblyat_biznes_n1108909
2. Проблема сміття: від Європи до України - Інститут громадянського суспільства. URL : https://www.csi.org.ua/news/problema-smittya-vid-yevropy-do-ukrayiny/
3. Сортування сміття в Україні: як це працює і чи працює взагалі. URL : https://ecofactor.ua/blog/sortirovka-musora-ukraina
4. ZenRobotics | Leader in Robotic Waste Recycling URL : https://zenrobotics.com/

Проектування та реалізація ігрової програми “ROKKILIKE”

Омельчук Олег
Україна, Львів, Львівська гімназія “Євшан”, 11 клас
uaolegikua@gmail.com

Вступ.

Давно відомо, що в процесі розвитку мислення та формування особистості важливу роль відіграє не тільки освітній фактор, але й характер ігрової діяльності. Для дитини гра є провідною діяльністю, оскільки під час гри вона засвоює значення і способи вживання предметів, а також різні варіанти соціальних відносин. В ігровій діяльності підліток не тільки заміщає реальні предмети, але й приймає на себе ту чи іншу роль і починає діяти відповідно до неї. Досвід ігрових та реальних взаємин у сюжетно-рольовій грі лягає в основу особливої властивості мислення, що дозволяє передбачити майбутню поведінку людей і залежно від цього будувати власну поведінку. Зараз діти весь свій вільний час проводять за монітором комп'ютера, граючи в свої улюблені ігри, де віртуальний світ вабить необмеженими можливостями. Але грати у цікаві ігри – це одне, а от зробити її самому – зовсім інше.

Ігровий двигун Unity.

Для розробки гри було використано редактор Unity. Він має простий Drag & Drop інтерфейс, який легко налаштовувати, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує дві сценарних мови: C#, JavaScript (модифікація). Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти (моделі), так і порожні ігрові об'єкти – тобто ті, які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту і розмірів за всіма координатними осями. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою.

Генерування світу.

Генерування світу відбувається випадковим чином на основі певних алгоритмів. Світ будується із так званих квадратних “кімнат”, які розміщуються одні відносно одних у запланованому шаблоні. Ця процедура включає у себе декілька етапів.

1. Генерування розміщення кімнат.

1) Створюється 2-вимірний масив кімнат, тобто матриця, після чого вона повністю заповнюється нулями, використовуючи подвійний цикл (1 – є кімната, 0 – немає).

2) По середині масиву "розміщується" випадкова фігура із одиниць (кімнат). Ці фігури являють собою заготовлені 2-вимірні масиви, одна з яких вибирається випадковим чином та через подвійний цикл переноситься на середину матриці кімнат.

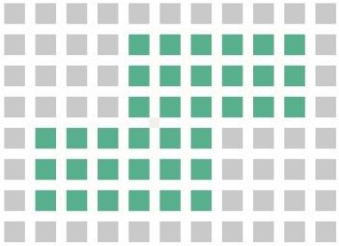


Рис. 1. Приклад матриці згенерованих кімнат

3) Вибираються кімнати, відстань між кожною з яких хоча б 3. Якщо число більше або рівне 3, цикл продовжується, в іншому випадку – переривається та береться наступна кімната. Після цього процесу всі вибрані кімнати перетворюються на нулі (видаляються), разом із правою нижньою кімнатою.

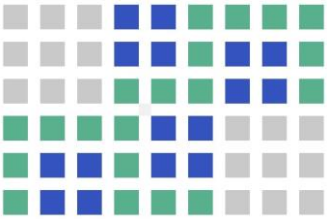


Рис. 2. Приклад матриці згенерованих кімнат

4) Всі нулі, які "доторкаються" до кімнат мають шанс 1/3(33.3%) стати кімнатою. Цей процес відбувається 2 рази.

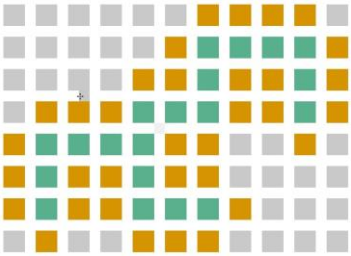


Рис. 3. Приклад догенерації кімнат

5) Усі кімнатки, які мають 8 сусідів видаляються (стають нулями), так як вони тільки роблять світ більш прямолінійним.

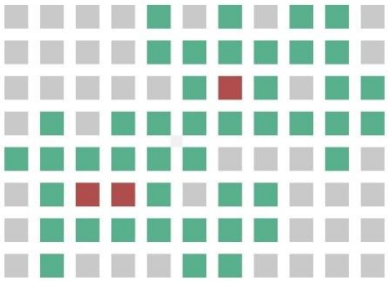


Рис. 5. Приклад вилучення зайвих кімнат

6) Далі цикл проходиться по всіх одиницях і ставить у світі за їхнім шаблоном кімнати, враховуючи сполучення з іншими кімнатами. Генерується чотирьохзначний код (1 – є вихід, 0 – виходу немає), починаючи з правого виходу і проти годинникової стрілки (справа, вгорі, зліва, знизу): у кімнатки із виходом тільки зліва код буде 0010, якщо знизу і зверху – 0101.

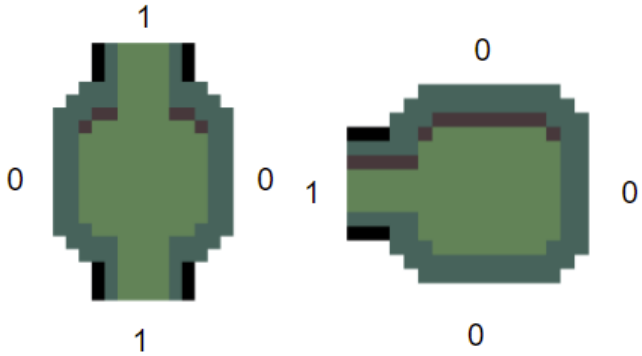


Рис. 6. Приклад чотирьохзначного коду розміщення шаблонів кімнат

7) На основі цього коду підбирається відповідний заготовлений шаблон із потрібними виходами кімнати і розміщується у світі.

2. Наповнення їх об'єктами (скринями, рудами, декораціями).

1) У кожній кімнаті розміщуються так звані ноди (від англ. Node), кожна з яких має бути на певній відстані від інших.

На прикладі видно, що функція, починаючи із лівого нижнього кута кімнати, починає розставляти ноди, якщо поблизу, на відстані disBetweenNodes, немає інших нод. Після цього цикл збільшує певну координату для нової ноди та знову перевіряє, чи можна її там розмістити.

2) На кожній із цих нод має шанс з'явитись якийсь об'єкт із заготовленого списку: скриня, ресурс, декорація, глечик тощо.

3. Розміщення груп ворогів у них.

1) Вибирається випадковий шаблон ворогів із заготовленого списку. Він складається із груп ворогів (клас EnemyGroup).

2) Кожна група розміщується із урахуванням стін та інших об'єктів, з якими можна фізично взаємодіяти. Тобто вибирається випадкова позиція у кімнаті, де поблизу немає стін та фізичних об'єктів.

Система інвентарів

У персонажа, як і у деяких об'єктів (пічка, ковадло, магазин), є свій інвентар, де зберігаються різні предмети, матеріали, спорядження, зброя, обладунки тощо. Їх також можна переміщати між слотами, використовуючи лівий клік миші. Є головний клас Inventory, що містить список слотів (List<InventorySlot>) у яких вже зберігається інформація про стеки предметів (ItemStack). Є три види слотів, кожен з яких походить від класу InventorySlot:

• DefaultSlot – стандартний слот, у який можна як класти так і забирати із нього предмети. Із них побудований інвентар персонажа та пічки. На нього можна поставити фільтр, щоб у нього можна було класти лише певний тип предметів. Наприклад: у слот для обладунків можна класти лише обладунки, у пічку можна ставити лише непереплавлену руду.

• CraftingSlot – слот, із якого можна без кінця брати предмети за умови, якщо у персонажа будуть потрібні для створення цього предмету матеріали, які при цьому витрачаються. Використовується у ковадлі, де створюються (крафтяться) предмети.

• ShopSlot – слот, із якого можна без кінця брати предмети за умови, якщо у персонажа є потрібна кількість монет(грошей), які при цьому витрачаються. Використовуються у магазині.

Ці класи слотів добавляються у список слотів у інвентарі (List<InventorySlot>) завдяки поліморфізму, так як InventorySlot для них є батьківським класом. Також при наведенні на предмет з'являється вікно, де є вся інформація про даний предмет: назва, рідкісність, сила атаки (у зброї), інгредієнти для створення (у ковадлі), ціна (у магазині).

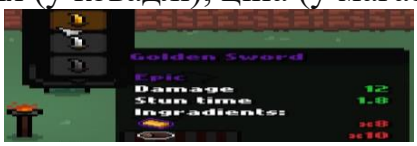


Рис. 7. Приклади меню предмета

Реалізація логіки ворогів.

У грі присутня множина “ворогів”, кожен із яких має свої характеристики, особливості поведінки та атаки. Більшість ворогів завжди атакують гравця напряму, тобто не зважають ні на які сторонні чинники, як от перешкоди на шляху. Одним із особливих ворогів є слизень. Він переміщується стрибками. Так як гра створена у 2-вимірному світі із елементами 3D, тобто вид збоку під кутом, використовувати вбудовані фізичні функції та компоненти не вийде. Тому довелось власноруч писати функцію для руху слизня. Стрибок, загалом, є дуже схожим на функцію синусів.

Крім того, щоб не стрибати на одному місці, потрібно також рівномірно переміщати слизня у певному напрямку, що було зроблено, використовуючи компонент Rigidbody, який відповідає за фізику.

Система квестів.

Квести – завдання, які можна отримати у жителів, за виконання яких персонаж отримає винагороду (монети, досвід). Є клас Quest, який має список із цілей, яких потрібно досягнути (List<QuestGoal>). Кожна ціль має свій тип (GoalType), в залежності від якого буде змінюватись завдання. Наприклад тип OpenChests потребує відкриття певної кількості сундуків (див. Додаток В).

Коли, наприклад, відкривається скриня, то спрацьовує глобальна подія (event), яка “слухає” скрипт, що відповідає за обробку квестів. Якщо гравець має завдання відкрити X скринь, то прогрес завдання збільшується. Якщо ж ціль досягнута, то квест завершується і персонажу потрібно повернутись до жителя, здати йому квест та отримати нагороду.

Висновки.

Створено 2D із елементами 3D гру стилю RPG із великою кількістю контенту, а саме: квести у жителів, можливість покращення навичок персонажа, випадкове генерування світу, створення із матеріалів та купівля предметів, взаємодія зі світом (скопування руд, відкривання скринь, ламання глечиків, тощо). У даній роботі розглянуто основні принципи роботи більшості з цих механізмів.

Літературні першоджерела.

1. Комп’ютерні ігри як важливий фактор формування життєвих навичок. Реферат. – Освіта.UA. URL: https://osvita.ua/vnz/reports/psychology/28614/ (дата звернення 24.03.2021).
2. Unity – Вікіпедія. URL:

https://uk.wikipedia.org/wiki/Unity_(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8) (дата звернення 24.03.2021).
3. Unity – Manual: Unity User Manual 2020.3 (LTS). URL: https://docs.unity3d.com/Manual/index.html (дата звернення: 12.02.2020).
4. Документация по C#... URL:https://docs.microsoft.com/ru-ru/dotnet/csharp/ (дата звернення: впродовж всієї розробки від 12.02.2020 до 24.02.2021).
5. Reference for HLSL: https://docs.microsoft.com/enus/windows/win32/direct3dhlsl/dx-graphics-hlsl-reference (дата звернення: впродовж розробки від 12.02.2020 до 24.02.2021).