

ПРОБЛЕМИ
ПРИКЛАДНОЇ МАТЕМАТИКИ
ТА КОМП'ЮТЕРНИХ НАУК



**ПРОБЛЕМИ
ПРИКЛАДНОЇ МАТЕМАТИКИ
ТА КОМП'ЮТЕРНИХ НАУК**

Тези доповідей

**конференції за підсумками науково-дослідної роботи
Дніпровського національного університету
імені Олеся Гончара
за 2020 рік**

Дніпро
2021

Проблеми прикладної математики та комп'ютерних наук. Тези доповідей тематичної наукової конференції за підсумками науково-дослідної роботи Дніпровського національного університету ім. Олеся Гончара за 2020 рік – Д.: ДНУ, 2021. – 48 с.

Тези надруковано в авторській редакції

Подані тези доповідей конференції за підсумками науково-дослідної роботи Дніпровського національного університету ім. Олеся Гончара за 2020 рік, присвячені проблемам прикладної математики та комп'ютерних наук. В доповідях розглядалися питання математичного моделювання, теоретичних основ кібернетики, чисельних методів, розробки і використання теоретичних та експериментальних методів досліджень в машинобудуванні; механіки деформівного твердого тіла; прикладних задач міцності конструкцій та розрахунку механічних параметрів технологічних процесів; математичного забезпечення ЕОМ; математичних та програмних засобів обробки та аналізу даних.

Для наукових та інженерних працівників, аспірантів і студентів старших курсів відповідних спеціальностей.

Рекомендовано вченою радою факультету прикладної математики
Дніпровського національного університету ім. Олеся Гончара

РЕДАКЦІЙНА КОЛЕГІЯ:

д-р фіз.-мат. наук, професор О.М. Кісельова, д-р техн. наук, професор О.Г. Байбуз, д-р фіз.-мат. наук, Л.Л.Гарт, д-р фіз.-мат. наук, професор В.І. Кузьменко, д-р фіз.-мат. наук, професор А.Є.Шевельова, д-р фіз.-мат. наук, професор Н.А. Гук, канд. фіз.-мат. наук, доцент В.А.Турчина.

ЗМІСТ

<i>Антонюк В.А., Сидорова М.Г.</i> Жестова взаємодія як засіб прискорення роботи користувача з комп'ютером	7
<i>Кашталян М.О., Сидорова М.Г.</i> Застосування рекурентних нейронних мереж у задачах обробки природної мови	8
<i>Бондаренко Б.Р., Сидорова М.Г.</i> Аналіз існуючих архітектур згорткових нейронних мереж	9
<i>Лящевська А.І., Мацуга О.М.</i> Бібліотека Impute для заповнення пропущених значень у наборі даних	11
<i>Мацуга О.М., Шеремет В.С.</i> Алгоритми відновлення кусково-лінійної регресії з одним вузлом та їх програмна реалізація	12
<i>Фунтиков М.К., Мацуга О.М.</i> Розробка програмного забезпечення оцінювання одновимірної лінійної сплайн-регресійної залежності	14
<i>Агеева Г.О., Антоненко С.В.</i> Розробка програмного забезпечення для прогнозування перемоги в кіберспорті за допомогою штучного інтелекту	15
<i>Павлюк Д.І., Антоненко С.В.</i> Розроблення вебсервісу для планування подій	16
<i>Ленський М.М., Михальчук Г.Й.</i> Програмне забезпечення для оптимізації транспортування молочної сировини	17
<i>Сегеда Н.Є.</i> Порівняльний аналіз зовнішніх налагоджувачів та можливостей середовища Visual Studio щодо програм мовою Асемблера	18
<i>Божуха Л.М., Богдан С.В.</i> Про структури та алгоритми реалізації методів розв'язання задачі зіткнення множини твердих тіл у тривимірному просторі	19

<i>Божуха Л.М., Самотокіна К.М.</i> Інструментальні засоби та ресурси розробки програмного забезпечення системи комп'ютерного зору для автоматизованої системи покупок	20
<i>Бузовський Є.О., Божуха Л.М.</i> Про методи мінімізації загального розміру колекції зображень	21
<i>Дорошенко Р.К., Раджаблі Е.Ф., Сосідко Є.В., Божуха Л.М.</i> Автоматизована інформаційна система Unisoft	22
<i>Божуха Л.М.</i> Графові схеми чисельних методів	23
<i>Логвин Д.А., Білобородько О.І.</i> Автоматизація маркетингової системи підбору продуктів для книжкового e-commerce проекту	24
<i>Зарубіна Г.А., Білобородько О.І.</i> Розроблення кроссплатформеної реалізації гри Bombergirl засобами Unity та C#	26
<i>Голтвянська К.О., Білобородько О.І.</i> Створення програмного забезпечення для модифікації фрагментів зображень з використанням контурного аналізу	28
<i>Кононов А.Д., Білобородько О.І.</i> Програмне забезпечення для визначення окремих показників зору та слуху	30
<i>Дегтяренко М.М., Білобородько О.І.</i> Рольова модель розмежування доступу в системі управління контентом Joomla	32
<i>Федій О.Д., Божуха Л.М.</i> Позиціонування за допомогою GSP навігації	33
<i>Антуф'єв Д.Є., Сидорова М.Г.</i> Аналіз існуючих датасетів музичних файлів для навчання штучних нейронних мереж	34
<i>Вакульчик С.О., Сидорова М.Г.</i> Створення комп'ютерної музики з використанням штучного інтелекту	36

<i>Білецький А.С.</i> Про атомарну модифікацію дерева ресурсів у мікросервісній архітектурі	37
<i>Мірзасєв Т.Р., Антоненко С.В.</i> Розробка платформи для управління подорожуючими групами людей	38
<i>Косогов А.С., Антоненко С.В.</i> Дослідження технології розгортання вебсервісу на різних платформах	39
<i>Карпов І.А., Антоненко С.В.</i> Кроки виконання інтелектуального аналізу тексту	40
<i>Сімакін С.К., Антоненко С.В.</i> Розроблення платформи для планування подорожі	42
<i>Лопатін А.П., Ємел'яненко Т.Г.</i> Розроблення месенджера з шифруванням даних	43
<i>Руденко С.Д., Мацуга О.М.</i> Локалізація розширення аорти на рентгенограмі грудної клітини	44
<i>Григоренко А.М., Антоненко С.В.</i> Гра «Ерудит» під мобільну операційну систему iOS	45
<i>Храпак Б.С.</i> Система аналізу тональності відгуків в інтернет-магазинах	47

ЖЕСТОВА ВЗАЄМОДІЯ ЯК ЗАСІБ ПРИСКОРЕННЯ РОБОТИ КОРИСТУВАЧА З КОМП'ЮТЕРОМ

В.А. Антонюк, М.Г. Сидорова
Факультет прикладної математики

Для використання комп'ютера треба не тільки розуміти його можливості, а й знати основні прийоми і правила взаємодії з ним. Доступні користувачеві способи взаємодії з програмами і пристроями комп'ютера називають користувацьким інтерфейсом.

Інтерфейс – сукупність способів організації діалогу «людина – комп'ютер». Він включає можливості завдання користувачем команд, наприклад запуску програми на виконання; види і способи виведення повідомлень комп'ютера у відповідь на команди користувача; види повідомлень про стан пристроїв і т. д.

Одним з найстаріших є інтерфейс командного рядка (консоль) – різновид текстового інтерфейсу. При такому способі взаємодії користувач вводить (в основному з клавіатури) в спеціальний командний рядок текстові команди, що є інструкціями, зрозумілими операційній системі.

В даний час практично всі користувачі комп'ютерів ведуть діалог з операційною системою за допомогою графічного інтерфейсу. Графічний інтерфейс зробив управління файлами і пристроями простіше, збільшивши число користувачів і кількість областей застосування персональних комп'ютерів.

Однак настає час коли графічний інтерфейс стає громіздким і складним для розуміння – має багато вкладених пунктів меню і прихованих налаштувань. Тому на допомогу графічному інтерфейсу приходять жестова взаємодія. Під кожним жестом може ховатися одна або кілька команд.

Точність і корисність програмного забезпечення для розпізнавання жестів пов'язане з безліччю проблем. Для розпізнавання жестів на основі зображень існують обмеження на обладнання й шум зображення. Зображення або відео можуть бути не при постійному освітленні. Предмети на задньому плані або відмінні риси користувачів можуть ускладнити розпізнавання.

Різноманітність реалізацій розпізнавання жестів на основі зображень також може викликати проблеми з життєздатністю технології для загального використання. Наприклад, алгоритм, відкалібрований для однієї камери, може не працювати для іншої камери. Рівень фонового шуму також викликає труднощі з відстеженням і розпізнаванням, особливо при виникненні оклюзії (часткової і повної). Крім того, відстань до камери, її дозвіл і якість також впливають на точність розпізнавання.

Щоб фіксувати людські жести за допомогою візуальних датчиків, також потрібні надійні методи комп'ютерного зору, наприклад, для відстеження рук і розпізнавання положення рук або для фіксації рухів голови, міміки або напрямків погляду.

ЗАСТОСУВАННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ ОБРОБКИ ПРИРОДНОЇ МОВИ

М.О. Кашталян, М.Г. Сидорова
Факультет прикладної математики

Обробка природної мови є галуззю, що поєднує у собі штучний інтелект і математичну лінгвістику і знаходить своє застосування у найрізноманітніших практичних задачах інтелектуального аналізу даних, представлених у формі мови чи тексту. Можна виділити такі основні напрями фундаментальних проблем, як мовне моделювання, що підкреслює кількісну асоціацію між природними словами; морфологічна обробка, що полягає у сегментації значущих компонентів слів та виявленні частин мови; синтаксична обробка або синтаксичний розбір, який будує схеми речень; та семантична обробка, яка дозволяє виявляти значення слів, фраз та компонентів вищого рівня в тексті. Сфери застосування включають такі задачі, як видобування корисної інформації, машинний переклад, узагальнення письмових робіт, автоматична відповідь на запитання шляхом виведення відповідей, аналіз тональності тексту, створення чат-ботів, класифікація та кластеризація документів тощо.

Раніше інтелектуальний аналіз тексту полягав у послідовному проведенні морфологічного, синтаксичного та семантичного аналізів. Такий підхід вимагав великої кількості людських ресурсів та попередніх знань для отримання коректних результатів, а їх якість прямо пропорційно залежала від кількості внесених у систему правил мови, якою написаний текст.

Насьогодні чудовим засобом для обробки природних мов є рекурентні нейронні мережі. Це пояснюється тим, що їх структура розроблена для роботи з послідовностями різної природи, а структура природної мови і є послідовністю слів у тексті та літер у словах. У рекурентній нейронній мережі значення виходу на кожному кроці залежить від значення виходу на попередньому кроці. Якщо розгорнути структуру у часі і співставити її кожен крок зі словом у реченні, то отримаємо однозначну відповідність: кожен попередній крок має вплив на наступний, або від попереднього слова залежить наступне.

Однак звичайні рекурентні нейрони не можуть бути застосовані для послідовностей значної довжини через проблему затухання градієнту. Тому існують спеціальні юніти із вдосконаленою внутрішньою структурою, що дозволяють запам'ятовувати чи забувати дані із певних кроків і називаються Long Short-Term Memory (LSTM). Такий тип шарів використовується найчастіше, оскільки дозволяє досягти найкращих результатів.

Для можливості роботи нейромережі зі словами попередньо здійснюється їх перетворення у вектора за допомогою підходу Word2Vec. Перевагою в порівнянні з one-hot представленням є суттєве зменшення розмірності та врахування семантичної подібності слів. Ще одним сучасним та дієвим механізмом, що дозволяє покращити результати у задачах пов'язаних з обробкою природної мови, є увага (attention).

АНАЛІЗ ІСНУЮЧИХ АРХІТЕКТУР ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Б.Р. Бондаренко, М.Г. Сидорова
Факультет прикладної математики

Згорткові нейронні мережі (ЗНМ) – це тип штучних нейронних мереж, який має величезне поширення та активно використовується у вирішенні задач класифікації зображень, а також задач обробки природної мови.

Актуальність дослідження та використання нейронних мереж у наш час – важко переоцінити. ЗНМ можна використовувати для класифікації зображень та розпізнавання об'єктів на фото, для досягнення різних цілей, наприклад таких як:

- класифікація та розпізнавання хвороби за рентгенівськими знімками;
- класифікація та розпізнавання точної моделі певного предмету, для швидкого пошуку його в інтернеті;
- прогнозування масштабів глобального потепління за знімками з супутників;
- розпізнавання обличчя для автентифікації;
- розпізнавання руху, для більш точного моделювання поведінки персонажу фільма/комп'ютерної гри, тощо.

На сьогоднішній день, для створення власної згорткової нейронної мережі не треба знову винаходити велосипед. Бібліотеки, такі як TensorFlow, PyTorch тощо, надають повний спектр можливостей створення нейронної мережі шар за шаром, обираючи необхідну кількість нейронів на кожному з них, обираючи різні функції активації, тощо. Все це можна використовувати для знайомства зі світом нейронних мереж. Однак, не маючи базового розуміння внутрішньої роботи нейронної мережі – намагатися побудувати власну архітектуру – наче кидати багато камінців у купу та чекати, що з'явиться прекрасний замок з каменю.

Інший варіант – запозичити архітектуру в одній з відомих згорткових нейронних мереж. Важливо розуміти, що відомими вони стали за свої видатні результати за участі в ILSVRC (ImageNet Large Visual Recognition Challenge).

Суть змагання полягає в тому, що величезну кількість зображень треба розподілити на тисячу класів. Переможець – нейронна мережа з найменшим показником помилки. До 2012 року, найкращим результатом помилки класифікації було значення – 26%, але саме 2012 рік став вирішальним для згорткових нейронних мереж.

AlexNet – модель-переможець 2012 року, перша ЗНМ, яка перемогла у цьому змаганні, з процентом помилок 16,4%. Саме після гучної перемоги AlexNet – у згорткових нейронних мережах побачили великий потенціал та почали розвивати цей шлях. Кажучи про AlexNet – це відносно проста нейронна мережа. Головні її досягнення:

- в якості функції активації була використана ReLU, замість поширеної в той час Tanh;

- при навчанні нейронної мережі, була використана аугментація даних, що дало проривні на той час результати.

VGG – переможець змагань у 2014 році. Фактично – це все ще проста нейронна мережа, як і AlexNet, однак VGG – глибша, ніж AlexNet (має більше згорткових та повнозв’язних шарів). Також, однією з відмінностей був розмір згорткових шарів (AlexNet мав 11x11, VGG мав 3x3), що дозволяло VGG виділяти більш локальні ознаки та паттерни. VGG мала велику кількість варіантів – з та без batch нормалізації та з різною кількістю згорткових шарів (VGG-11, VGG-13, VGG-16, VGG-19). Найпоширеніші з цих варіантів – VGG-16 та VGG-19, через найкращі результати.

ResNet – архітектура, яка поєднала ЗНМ та ідею skip connection. Переможець змагань у 2015 році з результатом помилки 3,6% (людина робить 5% помилок). Розглянемо skip connection на прикладі: маємо чотири шари, і зазвичай, до четвертого шару можна потрапити лише через другий та третій 1-2-3-4, на шляху до четвертого шару – сигнал може просто затухнути та буде визначений, як неважливий. Ця механіка дозволяє потрапити до четвертого шару напряму, в обхід другого та третього та додати вихідне значення на те, що було отримано до нього. Таким чином, ми не боїмося за те, що може відбутися у другому та третьому шарі, градієнт не буде затухати. За допомогою таких блоків, вдалося зробити нейронні мережі ще глибшими, ResNet має варіанти аж з 152 шарами. Також, завдяки розглянутій механіці skip connection, з ResNet можна видаляти слої без критичного погіршення якості класифікації (інформація все ще зможе потрапити на наступні шари).

GoogLeNet – архітектура, що реалізує ідею inception. В загальному виді, Inception - декілька згорткових шарів з різними розмірами на одному рівні. Це дозволяє передати велику кількість різної зібраної інформації на наступний рівень. Наприклад, зазвичай ми використовуємо згортку певного розміру і передаємо інформацію далі. Використовуючи inception, ми за одну ітерацію використовуємо згортку 1x1, 3x3, 5x5 та, наприклад, max pooling 3x3. Ми отримуємо інформацію різних рівнів за одну ітерацію, конкатенуємо її та передаємо на наступний рівень. Ця архітектура в модифікації GoogLeNet-v4 в 2016 році показала результат похибки 3.1%.

Підводячи підсумок, не можна назвати певну архітектуру найкращою, адже більша глибина дає більшу точність, але тягне за собою необхідність у великих обчислювальних можливостях та потребує більше часу для навчання. Архітектура повинна обиратися, спираючись на конкретну задачу та дані. Крім того, актуальним є підхід transfer learning, який полягає у тому, що застосування відомої архітектури ЗНМ у своїй задачі може здійснюватися з попередньо навченими ваговими коефіцієнтами та подальшим донавчанням лише окремих шарів, що дозволяє як покращити результати, так і зменшити обчислювальну складність.

БІБЛІОТЕКА IMPUTE ДЛЯ ЗАПОВНЕННЯ ПРОПУЩЕНИХ ЗНАЧЕНЬ У НАБОРІ ДАНИХ

А.І. Лящевська, О.М. Мацуга
Факультет прикладної математики

В процесі обробки та аналізу реальних наборів даних дослідники часто стикаються з наявністю пропусків у них. Більшість класичних методів аналізу даних не придатні до застосування в таких умовах. При цьому бібліотеки для мови Python, які найчастіше застосовують спеціалісти з аналізу даних, мають дуже обмежений функціонал щодо заповнення пропущених значень у наборах даних. З метою заповнити цю прогалину у роботі було створено бібліотеку `impute.py`, яку написано на мові Python.

У бібліотеці `impute.py` реалізовано такі методи:

- `mean` та `median`, які заповнюють пропущені значення середнім та медіаною по стовпцю. Особливістю їх реалізації є те, що за наявності мітки класу значення, яким буде заповнюватися пропуск, розраховується не за усіма об'єктами, а лише за об'єктами, що належать до того ж класу, що і об'єкт з пропущеним значенням.

- `zet`, що є реалізацією алгоритму ZET, який був запропонований Загоруйко Н.Г. В даному алгоритмі для заповнення окремого пропуску використовується не увесь набір даних, а лише його компетентна частина.

- `svd_based`, який базується на сингулярному розкладанні матриці даних.

- `knn_based`, який виконує заповнення пропусків на основі k найближчих сусідів. Для кожного рядку, в якому є пропущене значення, знаходиться k рядків, що найближчі до даного. Пропущене значення в стовпці заповнюється середнім значенням, розрахованим по даному стовпцю на основі k знайдених рядків.

- `reg_based`, в якому заповнення пропусків у стовпці здійснюється на основі лінійної регресійної залежності цього стовпця з іншими.

У бібліотеці також реалізовано методи `add_missing` (випадковим чином додає вказаний користувачем відсоток пропусків у набір даних) та `generate_data` (генерує дані із суміші багатовимірних нормальних розподілів).

З метою оцінити якість роботи реалізованих у бібліотеці методів під час заповнення пропусків у даних, що підлягають кластеризації методом k середніх, було використано згенеровані та реальні набори даних без пропусків. У ході експериментів у кожен набір вносився певний відсоток пропусків, потім ці пропуски заповнювалися усіма доступними в бібліотеці методами і дані без пропусків кластеризувалися. Для кожного методу фіксувалися такі показники: швидкість роботи, середньоквадратичне відхилення для різниці між істинними та відновленими значеннями, величина Rand Index для порівняння результатів кластеризації на початковому наборі даних та наборі після заповнення пропусків. Результати експериментів показали перевагу методу `knn_based`.

АЛГОРИТМИ ВІДНОВЛЕННЯ КУСКОВО-ЛІНІЙНОЇ РЕГРЕСІЇ З ОДНИМ ВУЗЛОМ ТА ЇХ ПРОГРАМНА РЕАЛІЗАЦІЯ

О.М. Мацуга, В.С. Шеремет
Факультет прикладної математики

Задача відновлення регресійних залежностей актуальна під час дослідження процесів різної природи, наприклад, фізичних та економічних. Для відновлення найбільш вірогідних залежностей у випадках, коли мають місце процеси зі структурними змінами, знайшли застосування моделі кусково-лінійної регресії [1, 2]. Мета даної роботи полягала у розробці більш швидких програмних реалізацій алгоритмів відновлення кусково-лінійної регресії з одним вузлом.

Модель кусково-лінійної регресії з одним вузлом x_0 має вигляд [1, 2]

$$y(x) = \begin{cases} A_1x + B_1, & x \leq x_0, \\ A_2(x - x_0) + A_1x_0 + B_1, & x \geq x_0. \end{cases}$$

Класичний алгоритм знаходження оцінок параметрів цієї моделі, представлений у [1, 2], передбачає, що вибірка відсортована за зростанням показника x і вузол співпадає з однією з варіант вибірки:

$$\hat{x}_0 = x_k, k = \overline{3, N-2}.$$

За фіксованого вузла оцінки $\hat{A}_1, \hat{B}_1, \hat{A}_2$ знаходять за методом найменших квадратів як розв'язок оптимізаційної задачі

$$S_{3ал}^2(k) \rightarrow \min_{A_1, B_1, A_2}, \quad (1)$$

де

$$S_{3ал}^2(k) = \frac{1}{N-4} \left(\sum_{i=1}^k (y_i - A_1x_i - B_1)^2 + \sum_{i=k+1}^N (y_i - A_2(x_i - x_k) - A_1x_k - B_1)^2 \right). \quad (2)$$

Розв'язок задачі (1) одержують, розв'язуючи систему лінійних алгебраїчних рівнянь (СЛАР)

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} \hat{A}_1 \\ \hat{B}_1 \\ \hat{A}_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^k y_i x_i + x_k \sum_{i=k+1}^N y_i \\ \sum_{i=1}^N y_i \\ \sum_{i=k+1}^N y_i x_i - x_k \sum_{i=k+1}^N y_i \end{pmatrix},$$

де

$$a_{11} = \sum_{i=1}^k x_i^2 + x_k^2(N-k);$$

$$\begin{aligned}
a_{12} &= \sum_{i=1}^k x_i + x_k(N - k); \\
a_{13} &= x_k \sum_{i=k+1}^N x_i - x_k^2(N - k); \\
a_{22} &= N; \quad a_{23} = \sum_{i=k+1}^N x_i - x_k(N - k); \\
a_{33} &= \sum_{i=k+1}^N x_i^2 - 2x_k \sum_{i=k+1}^N x_i + x_k^2(N - k).
\end{aligned}$$

Як оцінку вузла \hat{x}_0 обирають ту варіанту, в якій величина залишкової дисперсії (2) мінімальна. Для цього перебирають варіанти x_k , $k = \overline{3, N-2}$.

Стандартна програмна реалізація цього алгоритму передбачає, що для кожного вузла $\hat{x}_0 = x_k$, $k = \overline{3, N-2}$, формують СЛАР, для чого обчислюють відповідні суми шляхом проходження по всій вибірці. Це потребує значних витрат часу, особливо для вибірок великого обсягу.

У роботі запропоновано більш ефективну програмну реалізацію алгоритму. Пропонується попередньо для кожного елемента вибірки підрахувати суми потрібних величини від початку вибірки до цього елемента включно. Тоді для кожного вузла $\hat{x}_0 = x_k$, $k = \overline{3, N-2}$ суми від 1 до k будуть вже відомі, а кожен з сум від $k+1$ до N можна одержати як різницю між сумою від 1 до N та сумою від 1 до k .

Також у роботі розглянуто модифікацію класичного алгоритму, яка передбачає знаходження вузла не шляхом перебору варіантів вибірки, а на основі методу оптимізації, наприклад, поділу навпіл. Задля збільшення швидкості під час програмної реалізації цієї модифікації пропонується:

1) на кожній ітерації методу поділу навпіл визначати суми, потрібні для знаходження $\hat{A}_1, \hat{B}_1, \hat{A}_2$ для середини відрізка, на основі сум, які були одержані для кінців відрізка;

2) зберігати значення залишкової дисперсії для кінців звуженого відрізка, щоб не перераховувати їх на наступній ітерації.

Програмні реалізації класичного та модифікованого алгоритмів було виконано на мові програмування C# з використанням фреймворку .NET Core. Проведені експерименти засвідчили значні переваги у швидкості цих програмних реалізацій.

1. Приставка О.П., Байбуз О.Г., Ємел'яненко Т.Г. Методи та алгоритми сплайн-регресійного аналізу. Дніпропетровськ: Вид-во ДНУ, 2012. 144 с.

2. Приставка А.Ф., Передерий А.И., Райко О.В., Остропицкий В.М. Вычислительные методы и программная среда корреляционного и регрессионного анализа. Дніпропетровськ: ДДУ, 1996. 192 с.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОЦІНЮВАННЯ ОДНОВИМІРНОЇ ЛІНІЙНОЇ СПЛАЙН-РЕГРЕСІЙНОЇ ЗАЛЕЖНОСТІ

М.К. Фунтиков, О.М. Мацуга
Факультет прикладної математики

Відновлення регресійних залежностей набуває актуальності під час аналізу та передбачення біологічних, економічних, метеорологічних процесів. У випадках, коли явище описується нетривіальною нелінійною функцією, більш привабливим може бути представлення її у вигляді системи більш простих функцій, наприклад, у вигляді сплайн-регресії [1, 2]. Мета даної роботи полягала у розробці бібліотеки на мові Python для швидкого відновлення одновимірної лінійної сплайн-регресії з одним або багатьма вузлами.

Бібліотеку створено на мові програмування Python з використанням обчислювального середовища Google Colaboratory.

У бібліотеці реалізовано класи:

1. `SBR_Naive` – класичний алгоритм відновлення одновимірної сплайн-регресії з одним вузлом [1, 2].

2. `SBR` – перша модифікація класичного алгоритму, що зберігає та модифікує проміжкові значення між ітераціями.

3. `SBR_MinFinding` – друга модифікація класичного алгоритму, в якій для пошуку вузла використовується замість повного перебору метод пошуку мінімуму функції однієї змінної. В класі реалізовано методи дихотомії, золотого перетину та Фібоначчі.

4. `MBR` – алгоритм відновлення одновимірної сплайн-регресії з довільною кількістю вузлів.

Усі класи містять два методи:

- `fit` – приймає список значень незалежної змінної x та залежної змінної y , виконує навчання моделі на переданих даних згідно реалізованого алгоритму;
- `predict` – повертає значення залежної змінної y для переданих значень x , розраховані за навченою моделлю.

Було проведено тестування швидкості роботи реалізованих у бібліотеці алгоритмів. Перша модифікація класичного-наївного алгоритму надала чотирикратне пришвидшення на вибірці обсягом 800 елементів. Друга модифікація класичного алгоритму надала сорококрратне пришвидшення

Також у роботі було розглянуто використання сплайн-регресії з одним вузлом під час пошуку оптимальної кількості кластерів у задачі кластеризації методом k -середніх.

1. Приставка О.П., Байбуз О.Г., Ємел'яненко Т.Г. Методи та алгоритми сплайн-регресійного аналізу. Дніпропетровськ: Вид-во ДНУ, 2012. 144 с.

2. Приставка А.Ф., Передерий А.И., Райко О.В., Остропицкий В.М. Вычислительные методы и программная среда корреляционного и регрессионного анализа. Дніпропетровськ: ДДУ, 1996. 192 с.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ПЕРЕМОГИ В КІБЕРСПОРТІ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ

Г.О. Агєєва, С.В. Антоненко
Факультет прикладної математики

Dota 2 – це командна онлайн-гра у жанрі multiplayer online battle arena (МОВА), за якою проводяться всесвітні турніри з великими призовими. Дві команди по 5 гравців обирають героїв, за яких будуть змагатися за перемогу. Мета гри – зламати головну споруду команди супротивника, перешкоджаючи іншій команді руйнувати свої споруди.

Вибір героїв, які добре поєднуються один з одним та вдало протидіють героям супротивника – це одна з головних складових перемоги. У грі на даний момент 121 герой, що породжує велику кількість комбінацій. Знаходження переможної комбінації – важлива проблема не тільки для початківців, а й для професійних гравців.

В роботі розглядається задача прогнозування перемоги в кіберспортивній грі Dota 2 за вибраними героями на початку матчу. Було зібрано та оброблено два набори даних для машинного навчання, проведено перетворення у табличний вигляд.

Для рішення задачі бінарної класифікації було відібрано такі алгоритми:

1. Повнозв'язна нейронна мережа (персептрон).
2. Логістична регресія.
3. Машина опорних векторів (SVM).

Для реалізації вищезазначених алгоритмів було розроблено програмне забезпечення на мові програмування Python, за допомогою бібліотек Keras, Scikit-learn та TensorFlow у хмарному середовищі платформи Google Colaboratory.

У ході виконання роботи було розроблено програмне забезпечення для збору даних про проведені матчі, обробки та трансформації даних для навчання у табличний вигляд. З використанням цього ПЗ було отримано два датасети з різним набором залежних параметрів, різними критеріями для відбору даних та відповідно різних розмірів.

На цих наборах було проведено тренування алгоритмів з довготривалим налаштуванням параметрів для навчання та знаходження оптимальних показників точності передбачень та мінімізації перенавчання.

Було порівняно роботу програмного додатку з використанням трьох алгоритмів на даних реальних матчів. Порівняно точність передбачень та швидкість алгоритмів. Найкращим виявився алгоритм логістичної регресії з найвищою точністю передбачень та відносною швидкістю. Найгіршим показав себе алгоритм машини опорних векторів через дуже довгий час роботи алгоритму та найнижчими показниками точності.

Результати цих експериментів можна використовувати, як для оцінки та аналізу особистих стратегій, так і для подальших досліджень у цій області.

РОЗРОБЛЕННЯ ВЕБСЕРВІСУ ДЛЯ ПЛАНУВАННЯ ПОДІЙ

Д.І. Павлюк, С.В. Антоненко
Факультет прикладної математики

Планування часу є актуальною проблемою для сучасного суспільства. З пришвидшенням ритму життя збільшується кількість поставлених перед людиною задач. Часто досить складно послідовно та вчасно впоратися з усіма поставленими завданнями, тому частина роботи залишається невиконаною або відкладеною на майбутнє.

Метою роботи є вирішення проблеми управління часом за допомогою клієнт-серверного додатку для планування подій.

Для реалізації цієї мети слід виконати наступні дії:

1. Проаналізувати актуальні інструменти веб розробки та обрати оптимальні.
2. Спроекувати та розробити базу даних.
3. Розробити web API проєкту з базовими CRUD-операціями.
4. Розробити основну бізнес-логіку проєкту та відповідно доопрацювати web API.
5. Розробити клієнтську частину проєкту.

Серверну частину проєкту було реалізовано за допомогою фреймворку ASP.NET Core 3.1 на мові програмування C# при використанні середовища розробки Visual Studio 2019 Community; клієнтську частину розроблено за допомогою фронт-енд фреймворку Angular 9 на мові TypeScript при використанні редактора коду Visual Studio Code.

У результаті роботи було розроблено клієнт-серверний додаток, який дозволяє створювати на вести облікові записи користувача; планувати завдання, нагадування, які повторюються та події, які повторюються та до яких можна запросити інших користувачів сервісу; отримувати push-нотифікації та листи на пошту про настання тієї чи іншої події; експортувати розклад подій та інше.

Серверна частина представляє з себе проєкт ASP.NET Core Web API; клієнтська – додаток Angular. Систему було спроектовано за принципом багатосарової архітектури. Базу даних проєкту було реалізовано за допомогою підходу Code First ORM-бібліотеки Entity Framework. Засобами цієї ж бібліотеки був реалізований доступ до бази даних. Логіка планування подій реалізована за допомогою бібліотеки Quartz.NET. Інструмент SwaggerUI використано для візуалізації запитів до серверної частини проєкту. Push-нотифікації реалізовані за допомогою Google Firebase, а email-розсилку виконує smtp-сервер від Google. У додатку реалізовані такі важливі для веб додатка функції як: валідація моделі (використано бібліотеку Fluent Validation), логування (бібліотека Serilog), обробка помилок на боці сервера та реєстрація/авторизація (бібліотека Microsoft Identity).

Розроблений проєкт є цілком працездатним та справно виконує основні операції, які повинен робити планувальник задач. Додаток пристосований для практичного використання в подальшому часі.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ТРАНСПОРТУВАННЯ МОЛОЧНОЇ СИРОВИНИ

М.М. Ленський, Г.Й. Михальчук
Факультет прикладної математики

Для молочних комбінатів завжди є актуальною задачею організація оптимальної роботи транспортних засобів для збору і якнайшвидшої доставки сировини від виробників до місця переробки.

Молочний комбінат має певну кількість транспортних засобів, оснащених ємностями різної місткості. У будь-якій ємності можна перевозити тільки один тип сировини. Транспортний засіб повинен відвідати замовника у зазначений період часу, так зване, часове вікно. Надається список заявок від виробників на перевезення певних об'ємів сировини. Окрема заявка має бути оформлена для кожного типу сировини. У заявці вказується об'єм сировини, його тип, координати виробника та часове вікно.

Необхідно вирішити проблему побудови списку маршрутів, сумарна довжина яких мінімальна, з метою доставки на склад сировини від виробників. Транспортний засіб може тільки один раз відвідати будь-який пункт, вказаний у заявці, при цьому початкова і кінцева точка маршруту – це склад.

Для вирішення завдання розроблено двоетапний евристичний алгоритм. Першим етапом алгоритму є побудова початкового розв'язку, другим – покращення отриманих результатів.

Для реалізації розв'язку задачі вводиться поняття віртуальних ємностей транспортного засобу. Кількість ємностей дорівнює загальній кількості видів сировини. Максимальний об'єм кожної віртуальної ємності дорівнює сумі об'ємів реальних ємностей транспортного засобу.

Початковий розв'язок будується за допомогою алгоритму найближчого сусіда. Об'єм сировини в віртуальній ємності відповідного типу збільшується при відвідуванні кожного наступного виробника. Якщо сировину з віртуальних ємностей можливо розподілити по реальних, тоді даний виробник додається до маршруту.

Оптимізацію отриманого розв'язку виконано за допомогою алгоритмів 2-opt, Shift та Swap [1].

Реалізацію алгоритму здійснено мовою C#. Розроблене програмне забезпечення містить бібліотеку класів, яка забезпечує обробку вхідної інформації, побудову початкового розв'язку, його поліпшення та візуалізацію отриманих результатів.

Роботу програми апробовано на реальних наборах даних. Встановлено, що використання розробленого програмного забезпечення дозволяє скоротити грошові витрати на транспортування молочної сировини.

1. Eliana M. Toro O., Antonio H. Escobar Z., Mauricio Granada E. Literature review on the vehicle routing problem in the green transportation context. *Luna Azul*. 2016. No 42. P. 362–387.

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЗОВНІШНІХ НАЛАГОДЖУВАЧІВ ТА МОЖЛИВОСТЕЙ СЕРЕДОВИЩА VISUAL STUDIO ЩОДО ПРОГРАМ МОВОЮ АСЕМБЛЕРА

Н.Є. Сегеда

Факультет прикладної математики

Вивчення мови асемблера студентами спеціальності «Інженерія програмного забезпечення» зазвичай починається з налагоджування коду в OllyDBG або його аналогу x96dbg. Перші етапи розробки та тестування програм мовою асемблера в автономних налагоджувачах подібного типу розкривають тонкощі виконання команд, зберігання та переміщення даних на рівні архітектури комп'ютера. Більшість студентів вперше знайомиться з «глибинними» процесами перетворення даних та з інтересом відстежують покрокове виконання операцій над ними.

На подальших етапах вивчення мови асемблера невелика кількість студентів надає перевагу засобам налагоджування в середовищі Visual Studio. Які можливості для цього має інтегроване середовище, чи є в них переваги порівняно з автономними налагоджувачами, наскільки зручно ними скористатися?

Базові дії щодо створення проєкту та налаштування середовища Visual Studio з метою розробки та тестування програм мовою асемблера досить наочно представлені в статті [1]. Перейти в режим налагоджування можна через встановлення точки зупину в коді або почавши покрокове виконання програми. Тоді основну інформацію про хід обчислювального процесу та стан змінних можна побачити у вікнах (з меню Debug > Windows): Registers (регістри), Watch (перегляд змінних за іменами), Memory (перегляд змісту пам'яті за адресом), Disassembly (код розбирання) та ін.. Для розуміння процесу виконання програми мовою асемблера особливо важливим є налаштування вікна реєстрів, яке наповнюється з додаткового меню необхідними компонентами, наприклад: реєстри центрального процесора загального призначення, реєстри з плаваючою крапкою, прапори, інші складові архітектури комп'ютера.

Досить незвично виглядає позначення прапорів - вони не відповідають типовим прапорам, а саме у Visual Studio маємо: OV (Overflow), UP (Direction), EI (Interrupt), PL (Sign), ZR (Zero), AC (Auxiliary carry), PE (Parity), CY (Carry).

Під час налагоджування також можна редагувати ці прапори.

На відміну від автономних налагоджувачів, налаштування вікон у Visual Studio потребує деяких додаткових зусиль та часу, але для спостереження за виконанням програм, що демонструють зв'язок між функціями мовою C++ та асемблера, ці можливості середовища Visual Studio безумовно є зручними.

1. Setup an Assembly Project on Visual Studio 2019.

<https://programminghaven.home.blog/2020/02/16/setup-an-assembly-project-on-visual-studio-2019/>

ПРО СТРУКТУРИ ТА АЛГОРИТМИ РЕАЛІЗАЦІЇ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ ЗІТКНЕННЯ МНОЖИНИ ТВЕРДИХ ТІЛ У ТРИВИМІРНОМУ ПРОСТОРИ

Л.М. Божуха, С.В. Богдан
Факультет прикладної математики

Знаходження зіткнень твердих тіл у просторі, зокрема у тривимірному просторі – це окрема область досліджень у моделюванні твердих тіл, яка й на сьогоднішній день потребує ретельного дослідження. Це зумовлено тим, що для задач планування руху основними є перевірки зіткнень множини об'єктів між собою, а це означає, що ступінь оптимізації методу, який буде виконувати перевірку зіткнення твердих тіл, може стати ключовим для продуктивності системи. Найбільш елементарним методом рішення цієї задачі або методом «в лоб» буде перевірка кожної з пар з множини об'єктів, а це означає, що часова ефективність виконання методу буде рівною $O(n^2)$. Подібна швидкість роботи алгоритму просто жахає розробників комп'ютерних ігор та інженерного програмного забезпечення, оскільки масштабованість подібної системи зазнає краху.

Метою дослідження є виконання збору та систематизації інформації про методи та алгоритми розв'язання задачі зіткнення множини твердих тіл у тривимірному просторі щодо виконання порівняльного аналізу даних методів та виконання їх класифікації відповідно до задач, для яких виконання цих алгоритмів вважається найбільш доцільним. Розглянуті алгоритми, що використовують ієрархії обмежувальних об'ємів, алгоритм сортування з видаленням (sweep and prune), теорему про розділяючу вісь (не є самостійним алгоритмом), а також алгоритм Гілберта-Джонсона-Кьорті.

За допомогою методу порівняльного аналізу було визначено, що для задач, які потребують множини твердих тіл, що містить небагато об'єктів, доцільно використовувати звичайні обмежувальні об'єми, такі як вирівняні по координатних осях паралелепіпеди, об'єктно-орієнтовані паралелепіпеди та сфери. Якщо такі задачі потребують найвищої точності, то рекомендується використовувати об'єктно-орієнтовані паралелепіпеди, якщо ж достатньо середньої точності виконання перевірок і бажано мати простий та швидкий алгоритм, то краще скористатися вирівняними по осях паралелепіпедами. Сфери бажано використовувати для об'єктів, які мають форму, близьку до сферичної.

Для задач, які використовують множини великої потужності, процес перевірки можна розбити на дві фази – широку та вузьку. Для широкої фази, якщо мова йде про динамічне моделювання, краще за все скористатися алгоритмом sweep and prune, інакше перевага надається деревам обмежувальних примітивів. Для систем реального часу чудовим рішенням можна назвати алгоритм Гілберта-Джонсона-Кьорті.

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ТА РЕСУРСИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПОКУПОК

Л.М. Божуха, К.М. Самотокіна
Факультет прикладної математики

Розвиток систем комп'ютерного зору, високе проникнення смартфонів і мобільних платіжних сервісів, зростаюча щільність міського населення, що цінують зручність і ефективність, стали основними факторами для розвитку автоматизованих магазинів. В даний час магазини дозволяють робити автоматизовані покупки тільки за допомогою камер і бездротових систем, і автоматично знімати з вашої віртуальної картки гроші.

В даній ситуації наявними засобами можливо лише аналізувати дані про методи та технології, які можуть використовуватись для даного завдання, та експертно оцінювати їх оптимальне використання у майбутньому. Очевидно, використання різних підходів щодо проектування системи давати неповні і спотворені результати, які могли б бути покращені при наявності у аналітиків можливості оцінювання дій всіх відвідувачів магазину.

Для того, щоб детально показати роботу одного з таких магазинів та дослідити відповідні технології – розроблено програмне забезпечення емулятора на основі технологій комп'ютерного зору.

Розроблено програмне забезпечення емулятору автоматизованого магазину, який дозволяє робити покупки, придбані за допомогою камер і бездротових систем, і автоматично знімати з віртуальної карти гроші. У даній роботі показана взаємодія усіх сучасних технологій. Для проектування автоматизованої системи обрані методи реалізації виграшної стратегії і розпізнавання зображення. Для виграшної стратегії використовувався метод гілок і меж, а для розпізнавання карт на зображенні - поєднання контурного аналізу і розпізнавання по ключових точках.

В якості операційної системи обрана Ubuntu 16.10, середовища розробки - PhpStorm, мови програмування - JavaScript та PHP.

Під час відстеження великої кількості розташованих поруч людей не виникають ситуації, коли відстеження «перемикається» на іншу, поруч розташовану людину. Це відбувається через те, що метрика, яка використовується при відстеженні, враховує розташування кожної людини за допомогою місцезнаходження знайденої в новому кадрі фігури з передбаченим новим місцезнаходженням для тієї чи іншої траєкторії. Для покращення роботи емулятора можна використати додаткове порівняння знайдених фігур з відслідковуванням на основі їх SURF-ознак, що теоретично дозволить уникнути випадкового потрапляння в одну траєкторію фігур людей. Точність розпізнавання об'єктів може бути покращена за допомогою більшої кількості алгоритмів угруповання зображень, заснований на виділенні характерних ознак людей або ж на порівнянні об'єктів за допомогою штучної нейронної мережі.

ПРО МЕТОДИ МІНІМІЗАЦІЇ ЗАГАЛЬНОГО РОЗМІРУ КОЛЕКЦІЇ ЗОБРАЖЕНЬ

Є.О. Бузовський, Л.М. Божуха
Факультет прикладної математики

Формати зображень у своїй структурі мають вбудовані методи стиснення даних, які засновані на класичних алгоритмах стиснення або їх модифікаціях чи комбінаціях, залежно від типу зображень (растрові, векторні, фото або графіка) та типу стиснення (з втратами чи без).

Розглянуті декілька форматів зображень, зокрема GIF та PNG. Формат зображень GIF має вбудовану підтримку алгоритму стиснення даних Лемпеля-Зіва-Велча, що базується на створенні таблиці перетворення рядків. Формат PNG має вбудований алгоритм стиснення без втрат DEFLATE, що поєднує метод LZ77 і кодування Хаффмана^[1].

Проведений аналіз принципів роботи цих алгоритмів для вирішення задачі зменшення надлишковості даних. При використанні формату GIF більшу степінь стиснення можна досягти за умови наявності більшої кількості пікселів однакових кольорів. При використанні формату PNG найкращий результат стиснення можливо отримати за умови таких даних, в яких буде максимальна кількість повторюваних послідовностей (LZ77) та максимальна кількість повторюваних підряд символів (кодування Хаффмана). Зменшення палітри кольорів зображення спадкує збільшення кількості повторюваних підряд символів, що дає підтверджує принцип роботи методів стиснення з втратами^[2].

Запропоновані підходи у таких зворотних перетвореннях набору зображень форматів GIF та PNG, в результаті яких вбудований метод стиснення дасть більший коефіцієнт стиснення, ніж для початкової колекції зображень. Для цього виконується фрагментація кожного зображення та групування фрагментів усіх зображень за певним принципом схожості.

Проведені експериментальні дослідження з використанням методу кластеризації K-means для фрагментації зображення PNG.

Додаткових досліджень вимагає аналіз технологій щодо визначення таких параметрів, як кількість фрагментів, на які будуть поділено кожне зображення, та кількість фрагментів у групі.

Використана література

1. How PNG Works [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@duhroach/how-png-works-f1174e3cc7b7>
2. Reducing PNG file Size [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@duhroach/reducing-png-file-size-8473480d0476>.
3. Image Compression Using K-means in Python and R [Електронний ресурс] – Режим доступу до ресурсу: <https://dilloncamp.com/projects/kmeans.html>

АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА UNISOFT

Р.К. Дорошенко, Е.Ф. Раджаблі, Є.В. Сосідко, Л.М. Божуха

Факультет прикладної математики

Незважаючи на те, що ми все більше залучаємо сучасні технології для автоматизації своїх задач, роботи пов'язані з навчанням в багатьох вищих навчальних закладах України залишаються виконуватись «ручним» способом. Це приводить до ряду проблем, з якими стикаються студенти та викладачі: витрачання зайвого часу для передачі викладачеві готову роботу; необхідність мати «посередників» в спілкуванні між студентами і викладачами; важке знаходження інформації про завдання, терміни здачі, дати перездач, та відпрацювання; інформація не завжди точна і актуальна.

Для вирішення цих проблем розроблена система Unisoft, яка може бути використана як канал зв'язку та місце для зберігання інформації про студентів та навчальний заклад. У загальному плані Unisoft - це внутрішня система для закладів вищої освіти, яка спрощує процес взаємодії студентів з університетом. Ця система представляє сучасну технологію, розроблену у вигляді вебдодатку, для автоматизація процесів пов'язаних з навчанням, зберігає інформацію навчальних одиниць та студентів.

Перевагою використання системи Unisoft є зберігання інформації в одному місці. Система зберігає безліч типів інформації про студентів та інформацію навчального закладу, об'єднуючи всі записи в одному місці.

Перевагою використання розробленої інформаційної системи є спрощення взаємодії між студентами і викладачами. Система Unisoft надає наступні переваги в процесі взаємодії студентів з викладачами: зберігання інформації та матеріалів дисциплін та виконаних студентських робіт; відправлення виконаних завдань студентами викладачеві; виставлення балів та доповнення оцінювання коментарями; комунікація викладача зі студентами за допомогою постів; автоматичні повідомлення користувачів про події, що стосуються навчального процесу; створення тестів; створення новин.

Система Unisoft є хмарною, що по ряду причин є кращим варіантом, ніж локальні системи. Хмарні системи забезпечують більш просте налагодження, універсальний доступ, негайну технічну підтримку, поліпшену безпеку даних та підвищену здатність адаптуватися до змін на ринку і оновлювати продукт з мінімальними затратами.

Розроблена система має наступні додаткові переваги: локалізація інтерфейсу (доступні українська та російська мови); адаптація під мобільні пристрої (для студентів).

В результаті тривалого процесу розробки, було створено систему для вищих навчальних закладів, в якій всі комунікаційні процеси виконуються максимально просто і швидко для студентів, викладачів та адміністрації. Інформаційна система Unisoft була впроваджена в навчальний процес у другому семестрі 2019-2020 навчального року.

ГРАФОВІ СХЕМИ ЧИСЕЛЬНИХ МЕТОДІВ

Л.М. Божуха

Факультет прикладної математики

У дослідженні увага приділяється знаходженню зв'язку між безперервної моделлю системи і дискретним поданням архітектури програмного забезпечення.

Для прогнозування поведінки і масштабування дискретних процесів при представленні дискретної моделі як неперервної можна визначити обернену задачу.

Дискретність різницевих рівнянь в основному вивчаються для розуміння нелінійних явищ і процесів. Наприклад, співвідношення довжин орбіт періодичних точок знайшло продовження в питанні існування відображення на основі введеного визначення порядку [1].

Постає питання, чи можна провести узагальнення результатів на бінарні відношення. Складністю цього підходу є побудова методу представлення даних різницевого рівняння з формування відношення порядку в термінах дискретної математики.

Отримані результати найпростішого представлення методу послідовних наближень при введенні бінарного відношення порядку.

Побудовано послідовні і кругові схеми ітераційного процесу на прикладах бінарних дерев (дерева пошуку, дерева AVL і червоно-чорні дерева).

Введене відношення між елементами показало варіанти зміни кількості схем пошуку розв'язків різницевого рівняння, але не дає відповіді на питання про існування розв'язку та збіжність ітераційного процесу.

Використання запропонованої модифікації методу може розширити область існування варіантів представлення ітераційної схеми обчислення розв'язку диференціального рівняння першого порядку у графічному та матричному способах представлення бінарного відношення порядку.

Результати дослідження [2] можуть бути використані при аналізі складної структури даних для відтворення різницевого рівняння через сформовані відношення порядку в термінах дискретної математики.

Використана література

1. Сосуществование циклов непрерывного отображения прямой в себя / Шарковский А.Н. // Украин. математ. журн. 1964. Т. 16, вып. 1., с. 61-71.
2. Про графові схеми методу послідовних наближень Актуальні проблеми автоматизації та інформаційних технологій. – Дніпропетровськ: Ліра, 2020. Т.24., с. 72-78.

АВТОМАТИЗАЦІЯ МАРКЕТИНГОВОЇ СИСТЕМИ ПІДБОРУ ПРОДУКТІВ ДЛЯ КНИЖКОВОГО E-COMMERCE ПРОЄКТУ

Д.А. Логвин, О.І. Білобородько
Факультет прикладної математики

В сучасному світі майже кожен комерційний проєкт має свій сайт для продажу своїх товарів онлайн. Головна ціль подібних сайтів – автоматизувати більшу частину процесу взаємодії з клієнтами та збільшити обсяги продажів. У книжковому комерційному проєкті товаром є книга, що представлена в деякому виді: друкована, електронна чи в аудіо-форматі. Одним з вагомих важилів підвищення продажів та розширення бази зацікавлених клієнтів є маркетингова система підбору продуктів, іншими словами, рекомендаційна система.

Рекомендаційна система – це комплекс алгоритмів, програм та сервісів, задача яких спрогнозувати, що може зацікавити того чи іншого клієнта (користувача). Зазвичай, за основу беруться інформація персональної сторінки користувача та інші дані. Кращим прикладом слугує система Netflix, Youtube, Spotify, Amazon та інші.

Всього існує 4 типи рекомендаційних систем:

- колаборативна фільтрація – це рекомендації, які спрогнозовані за подібністю (схожістю) обраних товарів серед клієнтів. Наприклад, два клієнта обрали та придбали однаковий набір товарів (книгу, яблуко та каву), а третій нещодавно придбав книгу та яблуко. Отже третьому потрібно запропонувати придбати каву. Великим недоліком цього типу є те, що даних о нових клієнтах немає та спрогнозувати їм рекомендації не можливо.

- відбір за вмістом контенту – це рекомендації, які зроблені на основі того, з якими товарами чи продуктами ознайомлювався клієнт. Наприклад, рекомендації за схожим жанром, автором, ключовими словами. Зазвичай, це поширена практика рекомендацій.

- відбір за накопиченим досвідом користувача. В цьому випадку поступово збирається інформація про те, які товари придбають клієнти з деяким іншим. Наприклад, клієнт ознайомлюється зі смартфоном, отже, скоріш за все йому потрібен буде чохол для нього. Така практика поширена в інтернет-магазинах, як Rozetka, ALLO, Foxtrot та інші.

- гібридні (змішані) - це методи прогнозування рекомендацій, що включають в себе декілька алгоритмів рекомендацій різних типів. Зазвичай, за основу беруть один з трьох вищезазначених типів рекомендаційних систем та нарощують поверх нього коригуючі алгоритми.

Під час збору необхідних даних для відбору рекомендацій, використовують два види методів: явні та неявні.

Явні методи застосовуються у разі очікуваних подій від користувача, наприклад:

- 1) користувач оцінює запропонований об'єкт за шкалою;
- 2) користувач фільтрує групу об'єктів від найкращого до найгіршого;
- 3) користувачу пропонують створити список його улюблених об'єктів.

Неявні методи застосовуються під час спостереження за діями, поведінкою користувача або системи, наприклад:

- 1) спостереження за тим, що користувач переглядає в інтернет-магазині;
- 2) ведення записів про поведінку користувача онлайн;
- 3) відстеження вмісту комп'ютера користувача.

В сфері книжкових комерційних проєктів можна виділити необхідні дані для кожного з 3-х типів рекомендаційної системи, як:

- 1) Персональні дані користувача (вік, країна, мова спілкування тощо).
- 2) Досвід роботи користувача зі сайтом проєкту (відвідуванні розділи сайту, популярні книги, статті, найбільша активність користувача тощо).
- 3) Групи клієнтів за інтересами.

Отже, для книжкового комерційного продукту доцільно застосовувати гібридний тип рекомендаційної системи, що буде прогнозувати кращий підбір стосовно різних розділів сайту проєкту. Формувати та відображати рекомендації користувачу можна під час:

- 1) Відбору книг за характеристиками контенту. Кожний відібраний список буде складатися повністю або частково з рекомендацій.
- 2) Перегляду інформації з обраної книги. В цей час рекомендації будуть відображатися в окремому місці на сторінці (наприклад, розділ сторінки «Схожі книги»).
- 3) Серфінгу сайтом проєкту. На окремих обраних розділах сайту встановлюються банери, в яких представлені рекомендації у малому обсязі (1-3 об'єкти).

У даній роботі було проведено аналіз можливих інструментів та баз даних для реалізації рекомендаційної системи. Для розробки була обрана мова програмування TypeScript тому, що:

- 1) клієнтська та серверна частини системи застосовують одну мову програмування;
- 2) існують розвинені фреймворки для розробки обох частин системи, що вже мають базову архітектуру: клієнтська частина – React.js, серверна частина – Nest.js;
- 3) велика база бібліотек та модулів, більшу частину яких можна застосовувати в обох частинах системи.

Для зберігання даних була обрана NoSQL СКБД MongoDB з наступних міркувань:

- 1) легка інтеграція та взаємодія з серверною частиною(Nest.js);
- 2) зручне та швидке розгортання СКБД в мережі(MongoDB Cloud);
- 3) рекомендаційна система не потребує зберігання великого обсягу даних (наприклад, текст, аудіо- та відео- файли).

РОЗРОБЛЕННЯ КРОССПЛАТФОРМНОЇ РЕАЛІЗАЦІЇ ГРИ BOMBERGIRL ЗАСОБАМИ UNITY ТА C#

Г.А. Зарубіна, О.І. Білобородько
Факультет прикладної математики

Розробка ігор є актуальною галуззю сучасного світу інформаційних технологій. Індустрія кроссплатформених ігор (їх розробка та видання) один з напрямків, що швидко розвивається зараз. Кроссплатформені ігри – це ігри, що можна запускати на різних цільових платформах, маючи один код. Таким чином відбувається економія часу, грошей та інших ресурсів на розробку ігрового додатку ніж за умови розробки для кожної платформи окремо.

Існує декілька середовищ для розробки кроссплатформених ігор: Unity, Gogot Engine, Unreal Engine. У кожного є свої недоліки та переваги. Оскільки Unity є найпоширенішим середовищем у розробці ігор, до нього створено багато документації та прикладів у вільному доступі в інтернеті. Також він є лідером у розробці саме 2D ігор та підтримує мову C#, що мені знайома. Саме тому було обрано Unity для розробки гри.

Гра BomberGirl – це гра у жанрі аркади з положенням точки спостереження зверху та простим керуванням. Гравець знаходиться на прямокутному полі, на якому містяться блоки двох видів: такі, що можна зруйнувати й такі, що не руйнуються. З самого початку гри гравець має одну бомбу, за допомогою якої може руйнувати блоки. Під зруйнованими блоками знаходяться три види бонусів: бомба (збільшує на 1 кількість бомб, які можна класти одночасно), вогонь (збільшує довжину вибуху на 1 клітину), блискавка (збільшує швидкість гравця).

Гравець, руйнуючи блоки та збираючи бонуси, повинен дійти до противника та знищити його. Всього гравець має три життя, якщо він потрапить на супротивника, або під бомбу, віднімається одне життя. Якщо життя закінчились, тоді поточна гра закінчується та можна почати грати заново.

Реалізація гри передбачає такі комбінації гравців: 1 гравець та від 1-го до 3-х супротивників-комп'ютерів (вбудованих алгоритмів) або 2 гравці та 1 чи 2 супротивника - комп'ютера.

Якщо серед гравців одна людина, тоді для пересування використовуються клавіші стрілок, а клавіша space (прогалина) для того, щоб покласти бомбу. Якщо серед гравців дві людини, то одна використовує стрілки та space, а інша – клавіші "W", "A", "S", "D" та shift, щоб покласти бомбу.

У ході роботи була створена кроссплатформена двовимірна гра BomberGirl засобами Unity та C#. Розробку гри виконано для двох платформ: Windows та Android, та проведено тестування на відповідних пристроях.

Розробка починалась зі створення поля гри, всіх ігрових об'єктів, що приймають участь у грі (рис. 1). Далі було реалізовано рух гравців та можливість залишати бомбу. Цікавим досвідом є створення анімації вибуху бомби та можливість збирання бонусів.

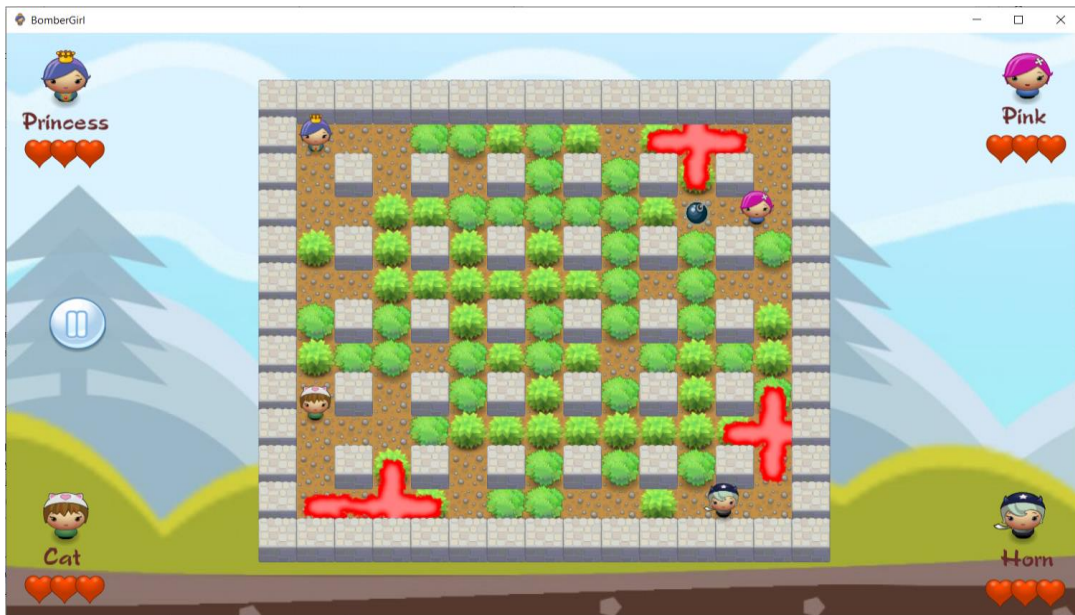


Рисунок 1 – Поле гри

Важливим етапом розробки ігрового додатку є проектування та реалізація логіки алгоритму, який використовує гравець-комп'ютер під час гри з реальною людиною. Останніми кроками у розробці гри було створення дружнього інтерфейсу для користувача, панелей меню, додання музичних спецефектів з можливістю їх вимкнення за бажанням користувача.

Під час розробки ігрового додатку BomberGirl було проведено ознайомлення зі станом питання та сучасними напрямками в розробці ігрових додатків, розглянуті шляхи побудови ігрових стратегій, проведено порівняння різних середовищ для розробки ігор, виявлення їх недоліків, переваг. Також вивчено сучасне середовище для розробки Unity.

Підсумовуючи, можна сказати, що кроссплатформена реалізація гри є вигідною та зручною при реалізації та просуванні гри на ринку ігрових технологій. На рисунку 2 зображено ігрове поле версії гри для телефону на операційній системі Android. Було додано кнопки управління гравцем для сенсорного телефону.

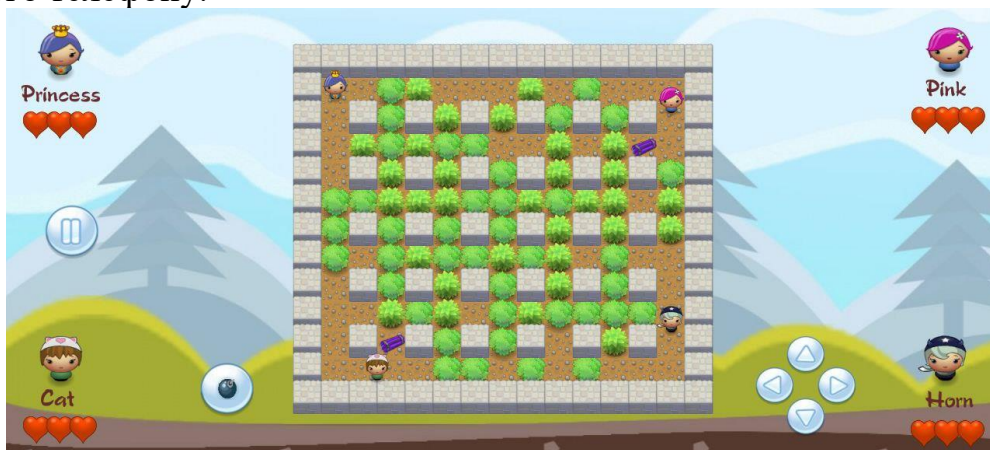


Рисунок 2 – Поле гри у версії для телефону

СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДИФІКАЦІЇ ФРАГМЕНТІВ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ КОНТУРНОГО АНАЛІЗУ

К.О. Голтвянська, О.І. Білобородько
Факультет прикладної математики

Швейна промисловість України є однією з основних галузей легкої промисловості. Попит на якісний одяг серед населення нашої країни є хорошим стимулом для проведення модернізації виробництва. Одним із варіантів розвитку даної промисловості може бути автоматизація та прискорення виробництва на етапі проектування одягу.

На даний момент є декілька програм, які автоматизують процес моделювання одягу. Однак всі вони використовуються для створення ескізів чи розробки викрійок виробів.

Тому моєю метою було створення програмного забезпечення, яке значно прискорило б процес підбору візерунків тканини для ескізу та надавало б можливість модифікувати його фрагменти. Розробка програмного продукту з зазначеними можливостями вимагає реалізації алгоритмів аналізу зображення. В даному випадку такий аналіз необхідно виконувати шляхом відокремлення з основного об'єкту зображення його складових частин. Для цього доцільно використовувати засоби контурного аналізу.

Контурний аналіз містить в собі методи для знаходження та опису контурів зображення. Під контуром об'єкту мають на увазі певну межу, яка виокремлює його серед інших об'єктів. Найчастіше контурний аналіз використовують для розпізнавання об'єктів на зображенні за певними ознаками. Оскільки форма об'єкта визначається завдяки контуру, тому дуже часто саме вона є ключовою ознакою для його розпізнавання за допомогою контурного аналізу.

Для розпізнавання контурів зображень використано детектор меж Кенні, а також метод Сатоші Сузукі та Кейчі Ейба. Для визначення контурів детектор меж Кенні виконує [1]:

- згладжування – розмиває зображення для видалення шуму;
- пошук градієнтів – межі визначаються там, де градієнт зображення приймає максимальне значення;
- подавлення не максимумів – визначати межі можуть тільки локальні максимуми;
- подвійну порогову фільтрацію – потенційні межі визначаються порогами;
- трасування області неоднозначності – фінальні контури встановлюються шляхом подавлення всіх країв, не зв'язаних з певними (сильними) межами.

Для зменшення чутливості сприйняття даного алгоритму до різних шумів використовують першу похідну від Гауссіани. Здебільшого даний метод застосовує чотири фільтри до зображення, щоб знайти вертикальні,

горизонтальні і діагональні межі, оскільки контури зображення можуть бути в різних напрямках.

Щоб застосувати метод Сатоші Сузукі та Кейчі Ейба необхідно провести бінаризацію зображення, оскільки даний метод працює виключно з бінарними зображеннями. Алгоритм даного методу дає змогу визначити контури з урахуванням вкладення. Тобто він дозволяє знайти зовнішні контури всього об'єкта, а також визначити коли контур одного об'єкта входить в контур іншого [2].

Таким чином, розроблене програмне забезпечення дозволяє приміряти на ескізи одягу різні візерунки тканин, а також модифікувати ці ескізи. Користувач має змогу завантажити власний ескіз, розбити його на деталі одягу, а потім до кожної з них приміряти різний колір чи візерунок тканини та подивитися на всю композицію загалом. Щоб модифікувати обраний ескіз, користувач може видаляти його елементи або ж навпаки додавати нові деталі. Дані модифікації зображень проілюстровано на рисунках 1 та 2. По завершенню роботи нариси можна зберегти в пам'яті комп'ютера. Програмне забезпечення було розроблене мовою програмування C# з використанням бібліотеки EmguCV, яка реалізує методи контурного аналізу.

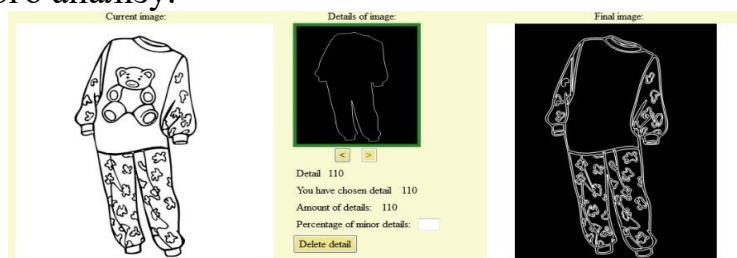


Рисунок 1 – Видалення деталей ескізу

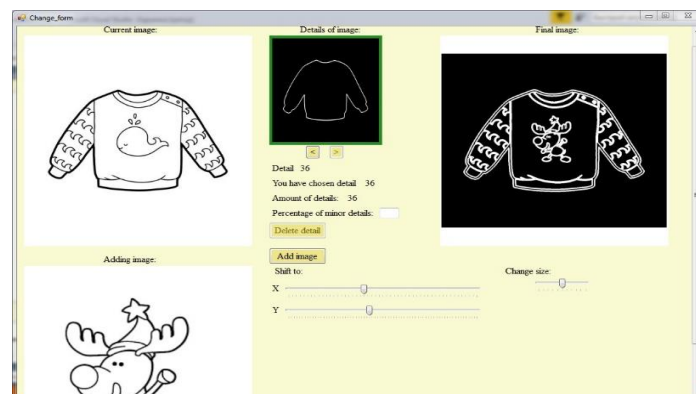


Рисунок 2 – Додавання зображення на ескіз

Список використаних джерел

1. Оператор Кэнни. URL: https://ru.wikipedia.org/wiki/Оператор_Кэнни
2. Синюк В.Г., Батищев Д.С., Сойникова Е.С., Михелев В.М. Использование алгоритмов компьютерного зрения для выполнения гематологического анализа на основе кривой прайс-джонса. *Журнал Экономика. Информатика*. 2018. Т. 45, № 3. С. 542.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ОКРЕМИХ ПОКАЗНИКІВ ЗОРУ ТА СЛУХУ

А.Д. Кононов, О.І. Білобородько
Факультет прикладної математики

Зір та слух є одними із основних способів відчуття оточення людиною, вони впроваджуються зоровим та слуховим аналізаторами, що включають в себе очі, вуха, структури ЦНС та мозку, отже за ними необхідно ретельно слідкувати та якомога швидше виявляти будь-які вади та захворювання, пов'язані з ними.

Так існує велика кількість тестів для перевірки зору, що раніше подавались у паперовому варіанті та були перенесені у цифровий простір, є менша кількість вже розроблених на комп'ютері, але, зазвичай, це набір зображень або відеоролик. Тому одним з варіантів вдосконалення систем тестування є забезпечення можливості вибору будь-яких збережених зображень та налаштування параметрів при тестуванні.

У випадку слухових тестів ситуація трохи інша – існує достатня кількість тестів вже у цифровому просторі, але існує необхідність виділити основні, об'єднати їх та надати можливість обирати параметри тестування та аудіо файли.

Із вищезазначеного можна зробити висновок про актуальність розробки додатку, що поєднує в собі тести як зорового так і слухового аналізаторів у вигляді комп'ютерної програми, а також надає можливість обрання будь-яких зображень та аудіо файлів для тестування.

Розроблене програмне забезпечення дозволяє користувачу – медичному фахівцю – проводити тестування зорового аналізатору шляхом обрання двох зображень із збережених на комп'ютері, поступовому зменшенню непрозорості «верхнього» зображення та пропонує особі, яку тестують, встановити момент коли вона починає бачити друге зображення. Користувачу надається можливість встановити параметри тестування у вигляді кроку та швидкості зміни непрозорості зображення. Форму налаштувань подано на рисунку 1. Результати тесту зберігаються в окремий файл. Також тестування зорового аналізатору може відбуватися завдяки елементам візуального маскування шляхом швидкого відображення послідовності масок та цільового зображення, яке необхідно встановити.

Тестування слухового аналізатора відбувається шляхом визначення особою, яку тестують, моменту, коли вона починає чути другий аудіо файл, що с початку не чутий. Гучність другого аудіо файлу поступово збільшується. Момент часу, зазначений особою, також фіксується в окремий файл і дозволяє медичному співробітнику проаналізувати результат. Для такого тестування додатково використано генерацію сигналів різної частоти та/або гучності, що розширює тестувальний спектр та забезпечує фахівця додатковими можливостями. Параметри генерації сигналів може визначати користувач (рис. 2).

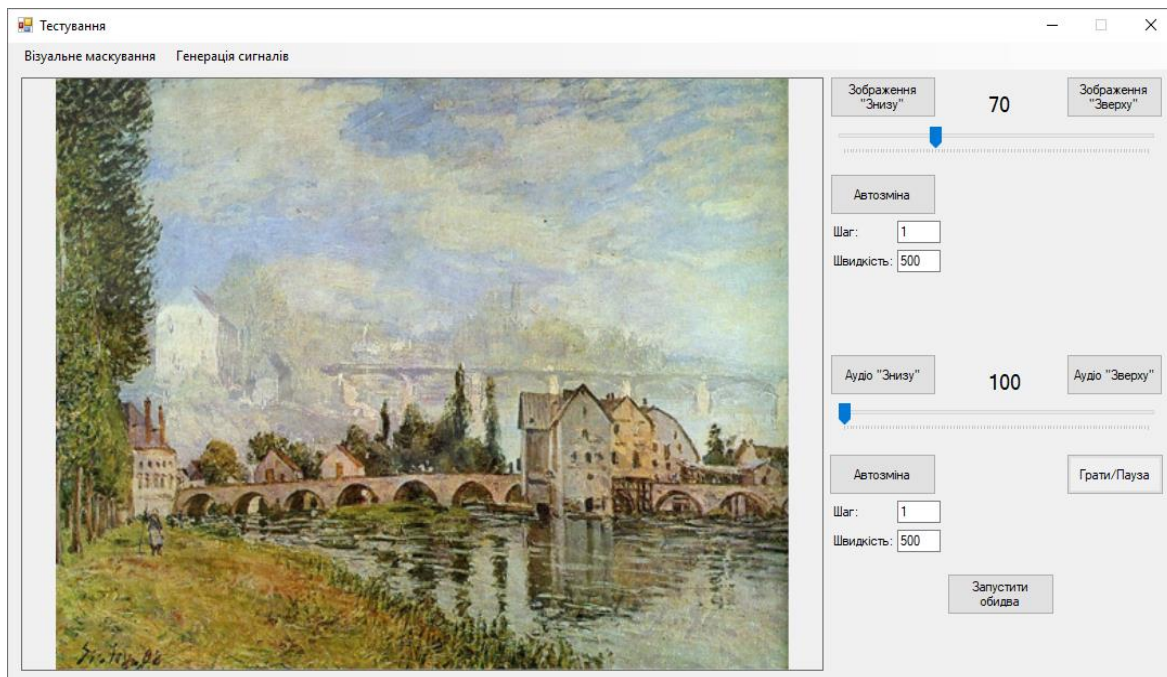


Рисунок 1 – Процес тестування візуального аналізатора

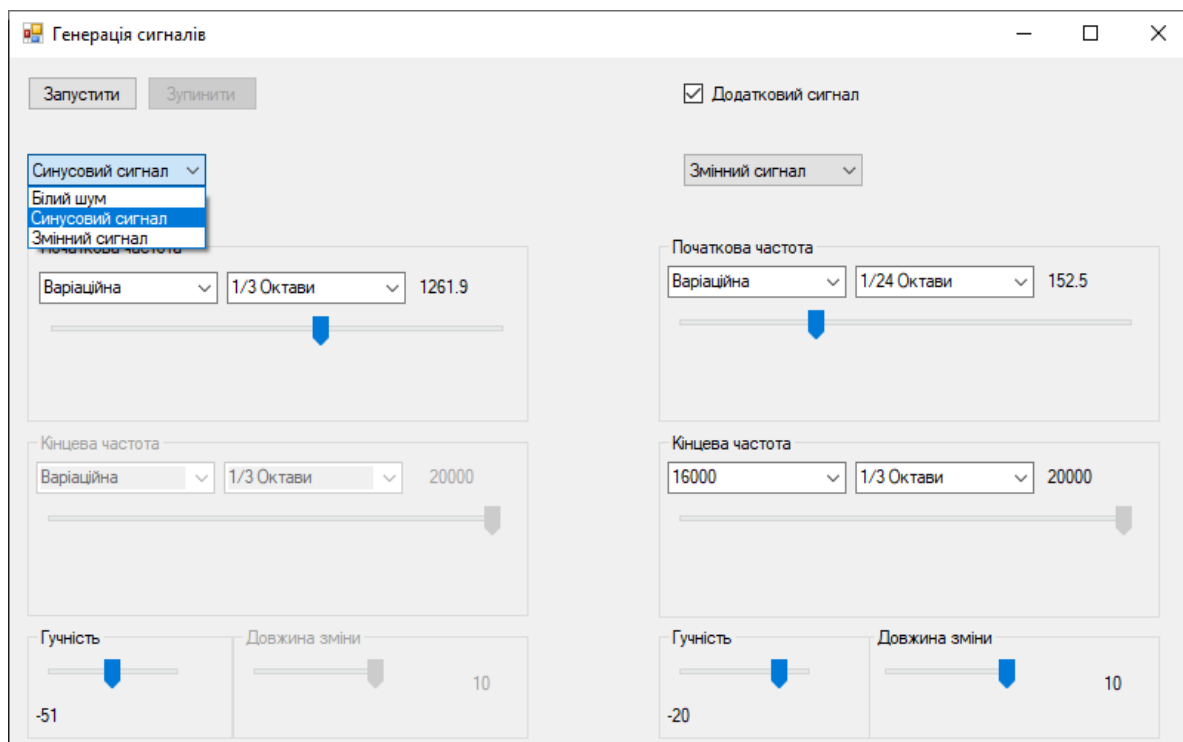


Рисунок 2 – Форма генерації сигналів

Так розроблене програмне забезпечення може бути використано у медичній діагностиці для перевірки зору та слуху.

РОЛЬОВА МОДЕЛЬ РОЗМЕЖУВАННЯ ДОСТУПУ В СИСТЕМІ УПРАВЛІННЯ КОНТЕНТОМ JOOMLA

М.М. Дегтяренко, О.І. Білобородько
Факультет прикладної математики

З розвитком технологій захист інформації став одним з найнеобхідніших завдань при розробці програмного забезпечення. Для цієї мети використовують безліч механізмів захисту, одним з яких є також системи розмежування доступу.

Системи розмежування доступу використовують для організації прав користувачів системи таким чином, щоб вони могли виконувати операції над певними елементами системи виключно у рамках своїх повноважень. Загалом, існує три основні моделі таких систем: дискреційна, мандатна та рольова.

Рольова модель є однією з найбільш розповсюджених моделей розмежування доступу. Для її реалізації створюється набір ролей, який зазвичай відповідає посадам або ролям всередині компанії. Роль користувача визначає набір дій, які він може виконувати над об'єктами системи відповідно до своїх прав та обов'язків.

Наступним кроком є створення спеціальної матриці доступу. Рядки такої матриці відповідають елементам системи, а стовпці – ролям користувачів. Елементами матриці є права доступу певної ролі до відповідного елемента системи. Коли користувач намагається отримати доступ до певного об'єкту системи, його права перевіряються згідно зі створеною матрицею.

Рольову модель часто використовують у системах управління контентом через її гнучкість, простоту у використанні та реалізації. Однією з таких систем управління контентом є Joomla. В ній кожен користувач належить до певної групи користувачів.

Керування загальними правами доступу виконується за допомогою спеціальної сторінки "Advanced Permissions Report". Для кожної ролі користувачів можна встановити права для входу на сайт, створювання, видалення та модифікацію контенту, доступу до адміністративної частини системи.

Щоб встановити права доступу до певної сторінки або розділу сайту, використовується сторінка "Permissions". З її допомогою адміністратор може встановити права на редагування, видалення сторінки, а також редагування її стану.

У сукупності, ці права встановлюють набір можливих дій для кожного користувача системи, який належить до певної ролі.

Отже, рольова модель розмежування доступом є простою в розумінні для кінцевих користувачів та досить гнучкою для використання у комп'ютерних системах. Вона дозволяє об'єднати права користувачів в одному правилі, але також надає можливість створити унікальні права доступу за потреби, створивши окрему роль користувачів. Зважаючи на ці переваги, рольову модель найчастіше використовують в системах управління контентом.

ПОЗИЦІОНУВАННЯ ЗА ДОПОМОГОЮ GPS НАВІГАЦІЇ

О.Д. Федій, Л.М. Божуха

Факультет прикладної математики

В даний час спостерігається зростання інтересу до вирішення завдання визначення місцеположення різних об'єктів. Місцезнаходження може бути обчислено як в глобальних, так і в локальних координатах. Системи глобального позиціонування, такі як GPS та ГЛОНАСС, отримали велике поширення завдяки широкому обхвату і досить високій точності.

Супутникова система навігації (GPS) - комплексна електронно-технічна система, яка складається із сукупності наземного і космічного устаткування, призначена для визначення місця розташування, а також параметрів руху для різних цілей. Місцезнаходження визначається трьома координатами приймача супутникової навігації в деякій системі координат.

Кожна система супутникової навігації містить близько 24 супутників. Супутники постійно передають пакети, в яких закодовано час відправки і координати супутника під час відправки. Швидкість передачі даних відома. Після отримання пакета приймач розраховує час проходження сигналу. Приймач може бути на будь-якій сфері радіуса з центром в точці, де знаходяться на перетині двох сфер. Таким чином, додавши ще один випромінювач, можна знайти точне місце розташування (рис. 1).

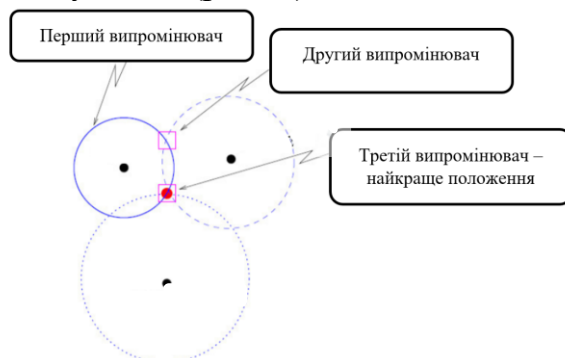


Рисунок 1 – Принцип трьох випромінювачів

Однак, таким чином знайти точне місце розташування неможливо. Це відбувається через те, що відстань від супутника до приймача знайдено неточно: у підрахунку присутні різні невраховані помилки і затримки. Виділяють наступні види помилок: іоносферні затримки, тропосферні затримки, помилки годинника приймача, помилка годин супутника, супутникові затримки.

Існує кілька груп методів виключення помилок і затримок. Основними є методи моделювання, двухчастотні приймачі та різницеві методи.

Отже, домогтися більш точних результатів можна за допомогою додаткових методів.

АНАЛІЗ ІСНУЮЧИХ ДАТАСЕТІВ МУЗИЧНИХ ФАЙЛІВ ДЛЯ НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Д.Є. Антуф'єв, М.Г. Сидорова
Факультет прикладної математики

Машинне навчання — це метод аналізу даних, який автоматизує побудову аналітичної моделі. Це галузь штучного інтелекту, заснована на ідеї, що системи можуть вчитися на основі даних, визначати закономірності та приймати рішення з мінімальним втручанням людини.

На сьогоднішній день, машинне навчання відіграє помітну роль в підходах та багатьох областях досліджень заснованих на даних, включаючи звукові і музичні обчислення. З цієї причини наявність ретельно підібраних наборів даних необхідно для отримання надійних результатів досліджень. Набір даних ImageNet став основою останніх досягнень в дослідженнях комп'ютерного зору. Це стало можливим, тому що ImageNet — це великомасштабний, відкрито доступний набір даних з достовірною інформацією. Незважаючи на те, що в області обробки звуку і музики є досить багато наборів даних, багато з них все ще мають серйозні недоліки. Розглянемо список деяких найбільш релевантних наборів даних для задач обробки звуку та музики.

GTZAN. Його доступність в інтернеті дозволила легко порівняти завдання розпізнавання музичного жанру в області пошуку музичної інформації. Незважаючи на свою популярність, цей набір даних критикують за його невеликий розмір, його несправності та часто використовувані розділи даних. Однак його несправності (повторення, неправильне маркування та спотворення) були виявлені лише у 2012 році, через десять років після його випуску. Крім того, було встановлено, що часто використовувані розділи даних надають надмірно оптимістичний погляд на сучасний стан. Останні роботи показують, що продуктивність набагато гірша при використанні розділу GTZAN із фільтром несправностей.

The Million Song Dataset. Він був випущений, щоб забезпечити масштабний набір даних для порівняльного аналізу MIR. Він містить аудіо-функції та метадані мільйона сучасних популярних музичних композицій із ухилом до поп/рок пісень. Аудіо функції можуть бути пов'язані з ресурсами, корисними для кількох дисциплін MIR: тексти пісень, обкладинки компакт-дисків, теги, показники схожості, призначені для користувача дані, кавер-версії пісень або жанрові ярлики. Це робить його ідеальним для вивчення мультимодальних підходів. Однак аудіофайли недоступні, а надані аудіофункції були витягнуті з допомогою пропріетарного програмного забезпечення, яке не можна налагодити або перевірити.

MagnaTagATune. Включає музичні дані, випущені за ліцензіями Creative Commons (CC), що спрощує обмін даними, а анотації (теги та схожість) були зроблені завдяки залученню користувачів до гри. Оскільки гейміфікація сама по собі була метою дослідження, інструмент анотації добре задокументований. Це робить цей набір даних прозорішим за інші з точки зору процесу його створення. В якості основних недоліків він є дуже незбалансованим і його анотації галасливі

та суперечливі. Для полегшення цих проблем дослідники, як правило, використовують теги 50 найкращих чи чистішу та попередньо оброблену версію.

UrbanSound8K. Включає таксономію та дві аудіо-колекції для дослідження міських звуків. UrbanSound8K включає записи різної довжини із позначками часу для звукових подій та помітних анотацій. UrbanSound8K містить помічені фрагменти цих подій, представлені в складках для порівняльного аналізу з базовими показниками. Автори порівняли кілька моделей класифікації цього набору даних і відзначили, що глибокі моделі перевершували лише поверхневі моделі при застосуванні методів збільшення даних, припускаючи, що сучасні підходи до машинного навчання вимагають великих та різноманітних наборів даних.

Ballroom. Цей музичний набір даних містить анотації ритму, темпу та жанру, тому може використовуватися для кількох завдань. Його також критикували за його невеликий розмір, повторювані пісні (було знайдено тринадцять реплік) та сильний взаємозв'язок між темпом та жанровими мітками (хоча набір даних був розроблений для оцінки ритмічних дескрипторів). Нещодавно було запропоновано розширення, і тепер доступно 4180 аудіокліпів для 13 незбалансованих класів.

ESC. Це відкритий набір даних для екологічної класифікації, який включає оцінку роботи людини, базовий рівень і код для відтворення вихідних результатів класифікації автора. ESC-50 містить 2000 анотованих кліпів, вручну анотованих однією людиною, в той час як ESC-US являє собою збірник з 250тис. немаркованих кліпів. Істотна різниця в масштабі між ними ілюструє, як немасштабовані процедури анотації можуть обмежувати розмір наборів даних.

На основі спостережень, зроблених у цьому огляді, можна зробити ряд висновків, які можна зрозуміти як вимоги, які слід враховувати при створенні набору даних:

1. Малі набори даних можуть обмежувати застосування певних технік машинного навчання, тому бажані більші набори даних.

2. Процеси створення набору даних повинні бути масштабованими та стійкими, щоб мати можливість створювати великі масиви даних.

3. Набори даних іноді анотуються та доповнюються новими даними, що робить їх придатними для нових завдань.

4. Слід встановити способи внесення змін до існуючих наборів даних та перетворити їх у щось динамічне.

5. Важливо задокументувати робочі процеси процесу створення набору даних та зробити їх прозорими.

6. Інтуїтивно зрозумілі інтерфейси для навігації по вмісту набору даних корисні для розуміння.

7. Розподіл даних сприяє відтворюваності і порівняльного аналізу.

8. Відкриті ліцензії Creative Commons (CC), що дозволяють безкоштовно поширювати аудіоконтент, бажані для більшого впливу на дослідження.

СТВОРЕННЯ КОМП'ЮТЕРНОЇ МУЗИКИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

С.О. Вакульчик, М.Г. Сидорова
Факультет прикладної математики

Музика сьогодні – невід’ємна частина життя кожної людини. Всі слухають її кожен день, під час поїздок у міському транспорті, під час відпочинку, чи навіть на роботі. Музика супроводжує свята, а також дає можливість побути наодинці зі своїми думками, зосередитись на чомусь або ж навпаки відволіктись. “Музика – це мистецтво та універсальна мова” – Альберт Ейнштейн.

Автоматична генерація музики – це процес створення музичного твору з мінімальним втручанням людини.

Метою цієї роботи є створення програмного забезпечення генерування музики на основі нейронних мереж.

Музику можна розглядати як послідовність нот і акордів, як набір тонів різної частоти. Розглянемо ці терміни з точки зору такого музичного інструменту як фортопіано:

- нота – звук відтворений однією клавішею;
- акорд – звук, що відтворений одночасно двома або більше клавішами;
- октава – паттерн, який повторюється. Кожна октава містить 7 білих і 5 чорних клавіш;

В результаті роботи було розроблено програмне забезпечення для генерації музики на основі нейромережевого підходу. Були реалізовані такі задачі:

- реалізована можливість завантаження та обробки даних формату MIDI;
- проаналізовані існуючі підходи до генерації музики;
- розроблена архітектура нейронної мережі та підібрані гіперпараметри навчання;
 - здійснено навчання моделі на музичних композиціях (соло на піаніно, композиції Шопена, Чайковського, різдвяні композиції);
 - підготовлена модель була застосована до генерування нової музики;
 - оцінена якість отриманих результатів та зроблені висновки щодо подальших досліджень.

Додаток розроблено за допомогою мови програмування Python та бібліотек Music 21, Tensorflow, Keras, NumPy тощо. Середовище розробки – Google Colab.

Під час розробки був використаний підхід на основі глибокого навчання, а саме створення та навчання моделей згорткових та рекурентних нейронних мереж. Експериментальним шляхом була обрана модель, що демонструвала найкращі показники функції похибки. Її застосування до генерації нових мелодій показало чудові результати.

ПРО АТОМАРНУ МОДИФІКАЦІЮ ДЕРЕВА РЕСУРСІВ У МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ

А.С. Білецький

Факультет прикладної математики

Сучасні тенденції у сфері інформаційних та комунікаційних систем вимагають певної гнучкості у представленні інформації. Для забезпечення гнучкості інтерфейсу необхідно мати і зручний механізм оновлення даних. Клієнт часто має справу не з одним конкретним ресурсом, а із цілим деревом пов'язаних між собою ресурсів з батьківсько-дочірніми зв'язками. Тому операції модифікації даних часто потребують змін не лише у батьківському ресурсі, а і у дочірніх також. Тобто для гнучкості інтерфейсу оновлення даних необхідно в рамках однієї операції описати які дії необхідно виконати над кожним елементом дерева ресурсів. На прикладі веб-сервісу магазину, ресурс покупця та його замовлення можна представити наступним JSON об'єктом:

```
{
  "customerName": "John", "address": { city: "City1" },
  "orders": [ { "orderName": "Order1" }, { "orderName": "Order2" } ]
}
```

Цей об'єкт можна розглянути як дерево, кожна з вершин якого описує назву атрибуту одного із ресурсів та його значення. Тобто для позначення який саме атрибут необхідно оновити достатньо описати шлях до нього у дереві. Для оновлення даних одного ресурсу тільки шляху та значення, але для оновлення колекції ресурсів і забезпечення достатньої гнучкості інтерфейсу необхідно також вказати як само необхідно оновити дані: додати нові, видалити зайві, або оновити поточні. Розглянемо такі операції: add, replace, remove. Наприклад, для додання нового замовлення покупцю клієнту веб-сервісу необхідно описати add операцію за шляхом /orders. А для видалення замовлення – операцію remove.

Таким чином можна сформувати гнучкий інтерфейс оновлення дерева ресурсів вказавши за допомогою шляху атрибут для оновлення, операцію оновлення та за необхідності параметри операції. Так операції add та replace потребують конкретного значення, а операція remove – ні. Операцію оновлення дерева ресурсів можна представити наступним JSON об'єктом:

```
[{ "op": "replace", "path": "/customerName", "value": "Barry" },
 { "op": "add", "path": "/orders/", "value": { "orderName": "Order2" } }]
```

Даний об'єкт вказує набір операцій які необхідно виконати над кожним елементом дерева ресурсів як одну операцію оновлення. У наведеному прикладі ми оновимо дані ресурсу покупця та додаємо дані до ресурсу замовлень. Такий підхід окрім гнучкості надає ще властивість атомарності до операції модифікації даних.

РОЗРОБКА ПЛАТФОРМИ ДЛЯ УПРАВЛІННЯ ПОДОРОЖУЮЧИМИ ГРУПАМИ ЛЮДЕЙ

Т.Р. Мірзаєв, С.В. Антоненко
Факультет прикладної математики

Люди, які хочуть спланувати подорож, часто піддаються невизначеності під час планування. Дана платформа вирішує проблему невизначеності, адже кожна людина, яка вирішила спланувати подорож, задає собі питання: коли, куди, на скільки днів, з ким, що потрібно брати з собою, та інші.

Для того, щоб відправитись у подорож, не завжди можливо знайти однодумців на даний час. Тому для пошуку охочих мандрувати, на зміну свого власного навколишнього оточення, приходять технології у вигляді додатку, де можна знайти бажану групу для подорожі.

Не завжди людина хоче знайомитись та контактувати з новими людьми, ділитися з ними думками, планами, тому додано функціонал створення власної групи та можливості змінювати її публічність з відкритої до закритої та навпаки.

Користувач має можливість шукати групу за певними параметрами, вступати до відкритих та надсилати запит щодо вступу до закритих груп.

Щоб мати можливість отримувати повідомлення та надавати відповіді на запрошення щодо вступу до групи, була додана можливість відправки повідомлення на електронну поштову скриньку.

Також не варто забувати, що процес підготовки до подорожі аж ніяк не легкий, саме детальна інформація в групі дозволяє не тримати усе в голові, не боячись щось забути.

Не рідко виникають питання щодо подорожі, на які хочеться мати миттєву реакцію, а також просто поспілкуватись з однодумцями, обговорити формальні чи неформальні теми, саме для цього і був впроваджений майданчик у вигляді внутрішнього чату групи.

Саме відсутність конкурентів у вигляді зручних інструментів, що дозволяють підготуватися до подорожі, надихнула до створення цього додатку. На рисунку 1 наведено приклад однієї з функцій, що реалізовано в запропонованому сайті.

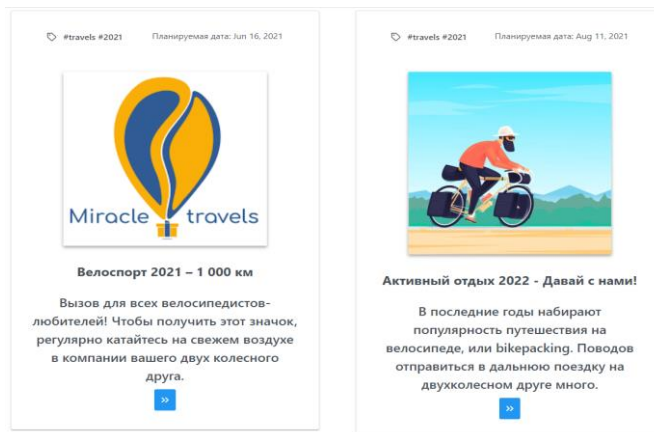


Рисунок 1 – Перегляд подорожуючих груп

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ РОЗГОРТАННЯ ВЕБСЕРВІСУ НА РІЗНИХ ПЛАТФОРМАХ

А.С. Косошов, С.В. Антоненко
Факультет прикладної математики

На сьогоднішній день лише розробити вебсервіс недостатньо, необхідно визначити модель та платформу, на якій він буде розгорнутий.

В роботі розглядаються чотири моделі розгортання вебсервісів.

«Інфраструктура як послуга» або IaaS допомагає абонентам уникнути витрат та складності придбання та управління власних фізичних серверів та ІТ-інфраструктури, виділяючи користувачеві віртуальні обчислювальні ресурси через Інтернет.

У моделі хмарних обчислень «Платформа як послуга» (PaaS) провайдери надають розробникам усе, що потрібно для створення додатків: засоби розробки, інфраструктуру та операційні системи.

«Програмне забезпечення як послуга» або SaaS – це хмарна модель розповсюдження програмного забезпечення, за допомогою якої сторонній провайдер розміщує додатки та робить їх доступними для абонентів через Інтернет.

Контейнери дозволяють розробникам та адміністраторам мережі відокремлювати певний набір програмних додатків від операційної системи та запускати їх у чистому, мінімальному та ізольованому середовищі.

«Контейнери як послуга» (CaaS) у хмарних обчисленнях – це модель послуг, яка дозволяє розробникам програмного забезпечення та ІТ-відділам завантажувати, систематизувати, запускати, масштабувати, керувати та зупиняти контейнери за допомогою віртуалізації на основі контейнерів.

Було розроблено вебсервіс на основі ASP.Net MVC та Blazor.Net. Вебдодаток являє собою інтернет-магазин із адмін-сторінкою для редагування каталогу товарів. Розроблений вебсервіс було розгорнуто на платформах Azure Web Apps та Azure Kubernetes Services. Отримана інфраструктура вебдодатка у Azure Web Apps наведена на рисунку 1.

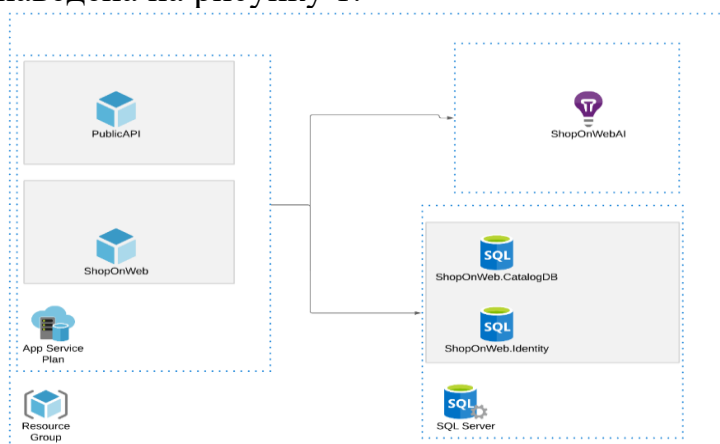


Рисунок 1 – Інфраструктура вебсервісу у Azure Web Apps

КРОКИ ВИКОНАННЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ТЕКСТУ

І.А. Карпов, С.В. Антоненко
Факультет прикладної математики

Аналіз тексту визначається як процес виявлення прихованого, корисного і цікавого зразка з неструктурованих текстових документів. Аналіз тексту також відомий як процес пошуку знань в текстовому інтелектуальному аналізі даних. Приблизно 80% корпоративних даних знаходиться в неструктурованому форматі. Інформаційний пошук в неструктурованому тексті дуже складний, оскільки він містить велику кількість інформації, що вимагає використання специфічних методів і алгоритмів обробки для отримання корисних знань. Оскільки найбільш поширеною формою для зберігання інформації є текст, інтелектуальний аналіз тексту видається більш важливим процесом, ніж інтелектуальний аналіз даних (data mining).

Інтелектуальний аналіз тексту є міждисциплінарною областю, яка включає збір даних, обробку web-даних, інформаційний пошук і витяг, комп'ютерну лінгвістику і обробку природної мови.

Інтелектуальний аналіз тексту можна розділити на дві фази:

1. Фільтрація тексту (очищення від «сміття»).
2. Вилучення знань.

Фаза очищення тексту, перетворює вихідне уявлення текстових документів форми в обрану проміжну форму. Вилучення знань, як видно з назви, отримує шаблони або знання з проміжної форми. Проміжна форма (Intermediate Form, IF) може бути частково структурованою (наприклад, концептуальне уявлення графа) або структурованою (наприклад, реляційне представлення даних). Проміжна форма може являти собою документ, де кожна сутність являє собою інший документ, або може являти собою певне поняття, де кожна сутність являє собою об'єкт або набір даних з певної предметної області.

Аналіз проміжної форми документів надає зразки та взаємозв'язки серед всіх документів. Прикладом є кластеризація, візуалізація і категоризація документів.

Загальний процес інтелектуального аналізу даних представлений на схемі на рис. 1.

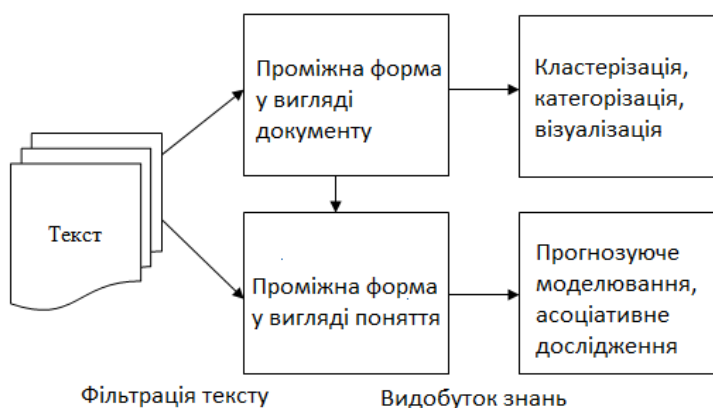


Рисунок 1 – Загальний процес інтелектуального аналізу даних

Кроки, які виконуються при інтелектуальному аналізі тексту. Кроки виконання аналізу тексту представлені нижче.

Попередня обробка тексту. Попередня обробка розділена на наступні кроки:

Перший крок попередньої обробки тексту далі розділений на:

1. Токенізація.
2. Видалення «стоп-слів».
3. Визначення походження слів.

Токенізація. Текстові документи містять набір сутностей. На цьому кроці виконується поділ тексту на окремі слова з видаленням прогалин і знаків пунктуації.

Видалення «стоп-слів». На цьому кроці проводиться видалення HTML і XML тегів, які можуть бути, якщо аналізується текст, отриманий з мережі Інтернет. Далі з тексту видаляються артиклі (наприклад, для англійської мови це «a», «is», «of» та інші).

Визначення походження слів являє собою процес ідентифікації коренів певних слів. Є в основному два типи походження: флективний та дериваційний. Найбільш поширеним алгоритмом визначення походження слів є алгоритм швейцара.

Перетворення тексту. Текстовий документ видається словами, з яких він складається, і інформації про їх походження. Є два підходи, які використовуються для представлення документа: мішок слів і векторні простори слів.

Пошук ознак. Це також відомо як пошук змінних. Це – процес відбору підмножини важливих ознак для використання в створенні моделей. Ця фаза, в основному виконує видалення тих ознак, які надлишкові або не важливі. Вибір ознак є підмножиною більш загальної області вилучення ознак.

Методи аналізу тексту. У цьому пункті інтелектуальний аналіз тексту стає збором даних. Методи розпізнавання даних, такі як кластеризація, класифікація, інформаційний пошук і т.ін., можуть використовуватися також і для інтелектуального аналізу тексту.

Інтерпретація/Оцінка. На цьому кроці відбувається аналіз результатів в залежності від поставлених цілей.

В даний час спостерігається дедалі більша тенденція у використанні комп'ютерів для того, щоб зберігати документи. Як результат, істотний обсяг даних зберігається в комп'ютерах в формі документів. Отже зростає науковий інтерес до теми видобутку тексту. Загалом інтелектуальний аналіз тексту складається з аналізу великої кількості текстових документів шляхом вилучення ключових фраз: понять, тощо, і підготувати оброблений текст для подальшого аналізу з використанням технік глибинного аналізу даних. Використовуючи розглянуті кроки, розробляється прикладна програмна система інтелектуального аналізу тексту.

РОЗРОБЛЕННЯ ПЛАТФОРМИ ДЛЯ ПЛАНУВАННЯ ПОДОРОЖІ

С.К. Сімакін, С.В. Антоненко

Факультет прикладної математики

У теперішній час туризм дуже популярний і кожен може, сам або з друзями, відправитися в будь-яку подорож, в будь-яку точку світу. У подорожі важлива організованість, щоб все пройшло так як задумано і подорож принесло задоволення і нові враження.

Для планування подорожі потрібні зручні інструменти, особливо коли в ньому бере участь багато людей. Месенджер найкращий для цього інструмент, але в не ньому вийде все зручно і доступно розписати. Тому, для зручності, потрібен спеціальний інструмент для планування подорожі, цьому і присвячена дипломна робота.

У роботі розроблюється інструмент у вигляді вебдодатку, адже це найуніверсальніший варіант: для роботи на комп'ютері або на смартфоні. Сервер розроблений у вигляді WebAPI, тобто можна розробити клієнт для будь-якої платформи. У якості серверної платформи обрано ASP.NET Core.

ASP.NET Core є кросплатформним, високопродуктивним середовищем з відкритим вихідним кодом для створення сучасних хмарних додатків, підключених до Інтернету.

WebAPI – це програмний інтерфейс, що складається з однієї або декількох загальнодоступних кінцевих точок для певної системи повідомлень запит-відповідь, зазвичай вираженою в JSON або XML, яка надається через Інтернет - найчастіше за допомогою HTTP веб сервера.

Клієнт вебдодатку розроблений за допомогою фреймворку Angular, адже він зручний для розробки та масштабування.

Angular – відкрита і вільна платформа для розробки вебдодатків, написана на мові TypeScript, що розробляється командою з компанії Google, а також спільнотою розробників з різних компаній. Підтримує мову зручну типізовану TypeScript та dependency injection.

У роботі проаналізовані існуючі рішення і виявлено, що вони не задовільняють усі потреби. Також проаналізовано та виділено вимоги до вебдодатку, що розроблюється у даній роботі.

Розглянуто архітектуру серверу та клієнту, яка розроблялась для зручної роботи з проектами та подальшого їх масштабування. Описано структуру бази даних та взаємодію з нею за допомогою Entity Framework Core.

Entity Framework Core – ORM-інструмент, який надає можливість абстрагуватися від роботи з базою даних за допомогою SQL і працювати з таблицями бази даних, як з класами .NET.

Ще у роботі розглянуто реалізацію авторизації за допомогою JSON Web Token (JWT), локалізації та мапи за допомогою відкритих бібліотек.

У результаті було розроблено основу проєкту для подальшої розробки та базовий функціонал для планування своєї подорожі. Хоча функціоналу ще замало для повноцінної заміни, але цей інструмент вже можна використовувати.

РОЗРОБЛЕННЯ МЕСЕНДЖЕРУ З ШИФРУВАННЯМ ДАНИХ

А.П. Лопатін, Т.Г. Ємел'яненко
Факультет прикладної математики

У сучасному світі мобільні пристрої грають величезну роль у житті людини. Мобільні додатки значно спрощують і прискорюють здійснення спілкування в інтернеті. Мобільні месенджери стали невід'ємною частиною сучасного життя. Перевага месенджерів полягає в тому, що користувач може спілкуватися, незалежно від його місця розташування і в будь-який час, для цього йому потрібен лише пристрій з інстальованим мобільним додатком та підключення до Інтернету.

Серед існуючих інструментів розробки мобільних застосунків для Android OS було обрано Android Studio, тому створення мобільного месенджера здійснювалася за допомогою середовища розробки програмного забезпечення Android Studio мовою програмування Kotlin.

Для підвищення якості кодової бази було використані такі шаблони проектування:

- 1) singleton;
- 2) builder;
- 3) adapter;
- 4) prototype.

Для зберігання даних користувачів використана платформа Firebase, що дозволяє зручно зберігати дані у real-time базі даних.

Розроблений користувацький інтерфейс застосунку відповідає Material Design GuideLine [1].

Результатом роботи є мобільний застосунок, який дозволяє виконувати такі дії:

- 1) зареєструватися або авторизуватися у додатку;
- 2) переглядати повідомлення;
- 3) переглядати список контактів;
- 4) створювати діалоги та групи;
- 5) відправляти повідомлення різних типів;
- 6) редагувати інформацію про користувача.

Месенджер має зручну навігацію і інтуїтивно-зрозумілий інтерфейс. У месенджері також реалізовано алгоритм шифрування/дешифрування, який описано в [2].

1. Design for Android. URL: <https://developer.android.com/design> (дата звернення: 08.04.2021).

2. Cryptography. URL: <https://developer.android.com/> (дата звернення: 08.04.2021).

ЛОКАЛІЗАЦІЯ РОЗШИРЕННЯ АОРТИ НА РЕНТГЕНОГРАМІ ГРУДНОЇ КЛІТИНИ

С.Д. Руденко, О.М. Мацуга
Факультет прикладної математики

На сьогоднішній день існує велика кількість захворювань, які можна виявити за допомогою променевої діагностики. Інтерпретація результатів такої діагностики є досить складною задачею, і навіть найкращі практикуючі лікарі можуть поставити хибний діагноз. Автоматизовані системи діагностики здатні допомогти під час виявлення хвороби і знизити ризик лікарської помилки. Існуючі системи, як правило, дозволяють лише отримати список знайдених патологій, проте вони не здатні вказати їх положення на знімку. Локалізація патологій зможе надати лікарям значну підтримку у вирішенні цієї складної задачі.

В даній роботі розглядається задача локалізації розширення аорти на рентгенівських знімках. Розв'язання поставленої задачі було проведено на наборі даних зі змагання «VinBigData Chest X-ray Abnormalities Detection» [1]. Для роботи було обрано лише знімки, розмічені радіологом «R10». Це було 4670 зображень у форматі dicom, з яких 2349 містили розширення аорти, тоді як на 2321 зображеннях не було жодних патологій. Цей набір було розбито на навчальну та тестову вибірки обсягом 3736 та 934 зображення відповідно.

Перед початком навчання було проведено попередню обробку даних. Зображення було переведено до формату jpeg, а їх розмір було змінено на 512x512.

В результаті роботи було побудовано нейронну мережу для локалізації розширення аорти. Для побудови використовувалась модель YOLOv5 [2]. Навчання мережі проводилося за допомогою відеокарти RTX 2070 протягом 60 епох. Точність результуючої моделі склала приблизно 85%.

У майбутньому планується: 1) провести розширену попередню обробку даних; 2) врахувати результати розмітки та діагностики інших радіологів; 3) побудувати модель, яка буде здатна локалізувати та класифікувати інші типи патологій, а не лише розширення аорти; 4) порівняти роботу алгоритму YOLOv5 з роботою інших алгоритмів, що вирішують дану задачу (наприклад, FasterRCNN). Ще одним важливим етапом подальшої роботи буде розробка програмного забезпечення для побудови та перевірки моделі зі зручним та наочним для користувача інтерфейсом.

1. vinbigdata-chest-xray dataset. URL: <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/data> (дата звернення: 07.04.2021).

2. Реалізація YOLOv5 на PyTorch. URL: <https://github.com/ultralytics/yolov5> (дата звернення: 07.04.2021).

ГРА «ЕРУДИТ» ПІД МОБІЛЬНУ ОПЕРАЦІЙНУ СИСТЕМУ IOS

А.М. Григоренко, С.В. Антоненко
Факультет прикладної математики

Метою роботи є розробка мобільної реалізації гри «Ерудит» та штучного інтелекту для цієї гри.

Вирішення задачі поділено на наступні етапи:

- аналіз операційної системи iOS та технологій, що використовуються для розробки під цю ОС;
- реалізація алгоритмів, що використовуються для розробки штучного інтелекту;
- створення мобільного додатку, що реалізує гру «Ерудит», та додавання штучного інтелекту до нього;
- перевірка працездатності роботи програми.


У грі приймають участь від двох до чотирьох учасників. Гравці обирають по 7 фішок з літерами. Фішки перегорнуті та перемішані, тому літери беруться навмання.

У свій хід гравець може скласти з існуючих у нього літер одне або більше слів. Слова повинні читатися зліва направо або згори донизу. Перше слово ставиться на перетині з центральною клітинкою, наступні слова повинні перетинатися з раніше викладеними словами. Нове слово може містити в собі раніше викладене, наприклад, зонт – горизонт – горизонталь. За один хід можна виставляти декілька слів, в тому числі, які перетинають один одного.

У грі дозволяється використовувати іменники у називному відмінку в однині, наприклад, стіл, парта, стілець, комп'ютер, або які вживаються лише в множині, наприклад, ножиці, сані і т.ін.

Використання у грі дієслів, прикметників, часток, прислівників, прийменників та інших частин мови, а також власних назв та пестливо-зменшувальних слів недопустимо.

Гральне поле розділене на кольорові (призові) клітинки.

 (чорна) – з неї починається гра, перше слово повинно проходити через цю клітинку



(зелена) – подвоює цінність літери



(блакитна) – потроює цінність літери



(фіолетова) – подвоює цінність слова



(червона) – потроює цінність слова

Кожна літера має свою вартість, тобто за неї нараховується певна кількість очок. Чим менше літера зустрічається у словах, тим вона дорожче оцінюється,

Таблиця цінностей літер української абетки представлені у таблиці 1.

Таблиця 1

Цінність літер

Літера	Цінність	Кількість	Літера	Цінність	Кількість
А	1	15	О	1	15
Б	3	5	П	2	10
В	1	8	Р	1	10
Г	3	5	С	1	10
Ґ	10	2	Т	1	8
Д	2	8	У	2	5
Е	1	15	Ф	10	2
Є	8	2	Х	5	3
Ж	5	3	Ц	5	2
З	5	3	Ч	5	3
И	1	13	Ш	8	2
І	2	6	Щ	10	2
Ї	8	2	Ь	10	3
Й	4	7	Ю	8	2
К	2	10	Я	3	5
Л	2	7	* (зірочка)	0	5
М	2	8	' (апостроф)	5	5
Н	1	14			

У грі є 5 універсальних фішок (зірочок), їх можна використовувати замість будь-якої літери. Після використання на гральному полі вона замінюється необхідною літерою та «згорає». Кількість очок, нарахованих за зірочку, дорівнює нулю.

Коли слово складене, цінність всіх його літер підсумовується, рахуються клітинки, які збільшують цінність літери, а потім – які збільшують цінність слова.

Гравець, який не зміг або не захотів скласти слово, має можливість замінити свої букви на нові і при цьому пропустити хід.

СИСТЕМА АНАЛІЗУ ТОНАЛЬНОСТІ ВІДГУКІВ В ІНТЕРНЕТ-МАГАЗИНАХ

Б.С. Храпак

Факультет прикладної математики

Розвиток сучасних технологій призвів до зростання долі інтернет-комерції. Тепер можна замовити книжки, продукти, інструменти та багато іншого не виходячи з дому. Водночас велика кількість товарів може збити з пантелику, тому магазини створюють системи відгуків. За допомогою них можна зрозуміти чи достатньої якості товар, які в нього недоліки та переваги. Спочатку відгуки були лише текстові, але згодом вони були замінені на більш формальні: 10-ти- та 5-тибальні, бінарні тощо. Однак, текстова система відгуків дозволяє отримати більше різноманітної інформації. Для автоматизованого аналізу текстових даних широко використовуються методи штучного інтелекту. Одним з напрямів такого аналізу є обробка природної мови та як приклад аналізу природної мови – аналіз тональності, тобто задача розуміння емоційного забарвлення тексту.

Метою роботи є дослідження одного зі способів переходу з текстового формату відгуків на бінарний за допомогою аналізатору тональності, навченого на наборі даних з відгуків за Amazon.

В цій роботі для побудови аналізатора розглядається підхід навчання з вчителем. Його можна розбити на декілька стандартних етапів.

Перш за все, необхідно знайти необхідний набір даних. Зазвичай це два поля – текст і необхідний клас (наприклад, позитивне або негативне відношення). Після цього виконується препроцесінг – обробка тексту. В залежності від вибору, цей етап може включати в себе такі речі як: лематизація, розширення скорочень, видалення наголошених символів та інше. Але, як правило, є три основні компоненти: токенизація, нормалізація, зняття шуму.

Наступним іде процес векторизації, в якому текст за допомогою спеціального алгоритму перетворюється на вектор. Найбільш поширені два підходи – мішок слів та методи глибокого навчання. Перший полягає в створенні масиву, кожен елемент якого відповідає слову в заданому словнику слів. Другий використовує навчені словники, в яких кожному слову вже відповідає масив.

Останній крок – класифікація. Це тип контрольованого навчання. Вона визначає клас, до якого належать елементи даних, її найкраще використовувати, коли вихід має кінцеві та дискретні значення. Вона також передбачає клас для вхідної змінної.

В пошуках найбільш оптимального алгоритму розглянуто кілька варіантів обробки текстової інформації. В першому варіанті в якості перетворювача текста у вектор використовувався `CountVectorizer`, `TfidfVectorizer` з модуля `sklearn.feature_extraction.text` та самописні версії на їх основі `MeanEmbeddingVectorizer` та `TfidfEmbeddingVectorizer` з трьома різними словниками: `glove.6B.50d` (навчений на 6 мільярдах токенів, 50 – розмір вектору, 822 MB файлу завантаження, 2GB разархівованого), `glove.840B.300d` (навчений на 840 мільярдах токенів, 300 – розмір вектору, 2 GB файлу завантаження, 5GB разархівованого) та класичного `word2vec`.

В якості класифікатора використовувались MultinomialNB та BernoulliNB з модуля sklearn.naive_bayes, SVC з модулю sklearn.svm та ExtraTreesClassifier з модуля sklearn.ensemble. Результати виконання можна побачити у таблицях 1 та 2.

Таблиця 1

Результати в залежності від векторизатора

	CountVectorizer	TfidfVectorizer
MultinomialNB	0.60075	0.55175
BernoulliNB	0.53825	0.53825

Таблиця 2

Результати в залежності від класифікатора

	MeanEmbeddingVectorizer	TfidfEmbeddingVectorizer
Малий glove	0.596	0.58725
Великий glove	0.6175	0.6205
w2v	0.60175	0.59425

Слід зауважити, що жоден з варіантів з SVC не закінчив роботу у адекватний термін часу, тому був виключений з вибору на кращий алгоритм.

Серед перелічених варіантів найвдалішим був MeanEmbeddingVectorizer та великий glove, які разом видали 62 відсотки правильності, що є досить низьким показником.

В наступному варіанті було вдосконалено використання CountVectorizer та TfidfVectorizer, що дало змогу отримати значно більш якісні показники – 84.6% (табл. 3).

Таблиця 3

Результати нової версії

	CountVectorizer	TfidfVectorizer
MultinomialNB	0.84225	0.84525
BernoulliNB	0.846	0.846

Було прийнято рішення розглянути інший алгоритм. Вибран був алгоритм з самописним препроцесінгом, TfidfVectorizer у якості векторизатора, DenseTransformer у якості конвертора, MinMaxScaler, SelectFromModel на основі LinearSVCу якості вибору якостей, VotingClassifier (LogisticRegression, MultinomialNB, SGDClassifier) у якості класифікатора. Для LinearSVC та LogisticRegression була встановлена L1-регуляризація. Результатом стали 83.23 відсотка точності, що трохи менше запропонованого мною варіанту.

У ході виконання даної роботи, спираючись на теоретичні положення, було побудовано модель для класифікації відгуків та створено програмне забезпечення для її використання. В процесі були розглянуті різні алгоритми та вибрані ті, що показали найкращий результат.

Дане рішення не є універсальним, але виконує цільову задачу та може бути розширене для більшого охоплення вхідних форматів.