Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та використання методів побудови систем послідовних рекомендацій
(тема)

Виконав:
студент 2 курсу, групи СШІм-19-2
Сухомлінова Ю. І.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Рябова Н. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Сухомліновій Юлії Ігорівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та використання методів побудови систем послідовних рекомендацій
затверджена наказом університету від 29 березня 2021 р. № 390Ст
2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2021р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження методів побудови систем послідовних рекомендацій. Документація мови програмування Python

4. Перелік питань, що потрібно опрацювати в роботі аналіз сучасного стану рекомендаційних систем, загальна характеристика традиційних та глибинних підходів до побудови систем послідовних рекомендацій, постановка задач дослідження, аналіз основних підходів до побудови систем послідовних рекомендацій з використанням згорткових нейронних мереж, обґрунтування вибору методу побудови системи послідовних рекомендацій, розробка архітектури згорткової нейронної мережі, експериментальна перевірка розробленої системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Графічне зображення ефекту довгого хвоста, Рисунок 2 – Рробота системи послідовних рекомендацій, Рисунок 3 – Метод k-найближчих сусідів, Рисунок 4 – Ланцюг Маркова, Рисунок 5 – Метод факторизації матриць, Рисунок 6 – Робота агента у методах навчання з підкріпленням, Рисунок 7 – Різниця архітектур рекурентної нейронної мережі та нейронної мережі прямого поширення, Рисунок 8 – Графова нейронна мережа, Рисунок 9 – Архітектура штучного нейрона, Рисунок 10 – Операція згортки, Рисунок 11 – Згортковий шар та шар пулінгу, Рисунок 12 – Повно зв'язаний шар, Рисунок 13 – Сигмоїдна функція, Рисунок 14 – Функція гіперболічний тангенс, Рисунок 15 – Функція ReLU, Рисунок 16 – Архітектура мережі Caser, Рисунок 17 – Архітектура моделі CosRec, Рисунок 18 – Фрагмент даних з датасету «User Behavior Data from Taobao for Recommendation»

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Рябова Н.В		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	29.03.2021	виконано
2	Аналіз предметної області та постановка задачі	30.03.2021 – 02.04.2021	виконано
3	Дослідження методів побудови систем послідовних		
	рекомендацій	03.04.2021 – 07.04.2021	виконано
4	Аналіз моделей послідовних рекомендацій з		
	використанням згорткових нейронних мереж	08.04.2021 – 13.04.2021	виконано
5	Розробка системи послідовних рекомендацій	14.04.2021 – 18.04.2021	виконано
6	Експериментальна перевірка розробленої системи	19.04.2021 – 20.04.2021	виконано
7	Обробка і оформлення результатів	21.04.2021 – 22.04.2021	виконано
8	Оформлення пояснювальної записки	22.04.2021 – 28.04.2021	виконано
9	Попередній захист	14.05.2021	виконано
10	Захист перед ЕК	18.05.2021	

Дата видачі завдання 29 березня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Рябова Н. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 83 с, 6 табл., 22 рис., 2 дод., 37 джерел.

ГЛИБИННЕ НАВЧАННЯ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ПОСЛІДОВНІ РЕКОМЕНДАЦІЇ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, PYHON.

Об’єкт дослідження – система рекомендаційного типу з використанням згорткової нейронної мережі.

Мета даної роботи – обґрунтування, вибір та використання методу побудови системи послідовних рекомендацій в заданій предметній області.

Предмет дослідження – методи побудови систем послідовних рекомендацій.

Методи дослідження – методи машинного та глибинного навчання, дослідження специфіки згорткових нейронних мереж та моделювання процесів надання рекомендацій на основі послідовностей взаємодії користувача з об’єктами інтересу з використанням згорткових нейронних мереж.

В результаті проведених досліджень вирішено задачу надання рекомендацій, сформованих на основі попередніх взаємодій користувача з об’єктами інтересу, з використанням глибинного навчання. Отримані результати використовуються у побудові системи послідовних рекомендацій на основі згорткових нейронних мереж.

Запропонована система є актуальною та може бути корисною при вирішенні задач в багатьох галузях, де використовуються рекомендації, наприклад у галузі електронної комерції, стрімінгових сервісах, соціальних мережах, онлайн банкінгу, пошуку наукових статей тощо.

РЕФЕРАТ

Пояснительная записка: 83 с, 6 табл., 22 рис., 2 прил., 37 источников.

ГЛУБИННОЕ ОБУЧЕНИЕ, ПОСЛЕДОВАТЕЛЬНЫЕ РЕКОМЕНДАЦИИ, РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ, СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, PYHON.

Объект исследования – система рекомендательного типа с использованием сверточных нейронной сетей.

Цель данной работы – обоснование, выбор и использование метода построения системы последовательных рекомендаций в заданной предметной области.

Предмет исследования – методы построения систем последовательных рекомендаций.

Методы исследования – методы машинного и глубокого обучения, исследование специфики сверточных нейронных сетей и моделирования процессов предоставления рекомендаций на основе последовательностей взаимодействия пользователя с объектами интереса с использованием сверточных нейронных сетей.

В результате проведенных исследований решена задача предоставления рекомендаций, сформированных на основе предыдущих взаимодействий пользователя с объектами интереса с использованием глубинного обучения. Полученные результаты используются в построении системы последовательных рекомендаций на основе сверточных нейронных сетей. Разработанная система актуальна и может быть полезной при решении различных задач, где используются рекомендации, например в электронной коммерции, стримингових сервисах, социальных сетях, онлайн банкинге, поиске научных статей и т. п..

ABSTRACT

Explanatory note: 83 p., 6 tab., 22 fig., 2 an., 37 sources.

CONVOLUTIONAL NEURAL NETWORK, DEEP LEARNING, PYHON, RECOMMENDER SYSTEMS, SEQUENTIAL RECOMMENDATIONS.

The object of research is a system of recommendation type using a convolutional neural network.

The purpose of this work is to substantiate, choose and use the method of building a system of sequential recommendations in a given subject area.

The subject of research – methods of building systems of sequential recommendations.

Research methods – specificity of convolutional neural networks research and modeling of processes of giving recommendations on the basis of sequences of user items interactions using convolutional neural networks.

As a result of the research, the problem of providing recommendations based on previous user items interactions, using deep learning, is solved. The obtained results are used in the construction of a system of sequential recommendations based on convolutional neural networks.

The proposed system is relevant and can be used in solving problems for many industries where recommendations are used, such as e-commerce, streaming services, social networks, online banking, search for scientific articles and more.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Caser – Convolutional Sequence Embedding Recommendation Model – рекомендаційна модель згорткової послідовності ембедінгів;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

CosRec – Convolutional Neural Network for Sequential Recommendation – згорткова нейронна мережа для послідовних рекомендацій;

GNN – Graph Neural Network – графова нейронна мережа;

KNN – K-Nearest Neighbors method – метод k-найближчих сусідів;

MAP – Mean Average Precision – усереднена точність;

ReLU – The Rectified Linear Unit – випрямлена лінійна одиниця;

RNN – Recurrent Neural Network– рекурентна нейронна мережа;

SGD – Stochastic Gradient Descent – стохастичний градієнтний спуск;

SRS – Sequential Recommender System – система послідовних рекомендацій.

ВСТУП

За останні роки стрімкий розвиток IT технологій сприяв виникненню та розвитку нових типів веб-систем, призначених для надання допомоги користувачам щодо підтримки прийняття рішень при вирішенні задач вибору певного типу об’єкту із множини, що підлягає аналізу, у різноманітних предметних галузях. Найяскравішим прикладом такої предметної галузі є, безумовно, електронна комерція (e-commerce). Кожна система типу e-commerce має за головну мету задоволення потреб користувача, пропонування та продаж необхідного товару покупцеві.

Не так давно люди жили в тісних громадах. Кожен продавець особисто знав своїх клієнтів і міг робити їм рекомендації на основі особистих знань про їхні минулі покупки. Цей тип особистих відносин означав, що клієнти отримують чудове обслуговування, тоді як продавці могли скористатися перевагами лояльності до бренду, оскільки вони розуміли потреби та уподобання своїх клієнтів. Сьогодні усе виглядає інакше. Хоча ми й виграємо від того що маємо великий вибір продуктів чи послуг, з іншого боку, ми втрачаємо таку близькість особистих відносин «клієнт-постачальник». Однак відсутність персоналізованого обслуговування не змінює дуже важливого факту, що запорукою успішних продажів є розуміння проблем людини. Саме для вирішення цієї проблеми були створені рекомендаційні системи.

Рекомендаційні системи відіграють важливу роль у сприянні зростанню електронної комерції у багатьох сферах в Інтернеті. Крім того, система рекомендацій є однією з найважливіших областей дослідження на сьогодняшній день, оскільки вона допомагає користувачам знайти речі, що їх цікавлять в Інтернеті, а компанії, в свою чергу, можуть розраховувати на збільшення прибутку [1]. Багато товарів купуються саме в Інтернеті, і тому зростає попит на фільтрування великої кількісті товарів, що знаходяться на

веб-сайтах, щоб конкретний товар можна було легше знайти відповідно до фактичного інтересу користувача [2].

Рекомендаційні системи – це системи, призначені для того, щоб рекомендувати речі користувачеві на основі багатьох різних факторів [3]. Ці системи прогнозують найбільш вірогідний продукт, який користувач скоріш за все придбає або який його скоріш за все зацікавить. Спочатку рекомендаційні системи були розроблені для того, щоб справитися з великою кількістю доступних даних. Однак, оскільки веб-сайти з рекомендаційними системами продемонстрували збільшення показників продажів, стало очевидним, що ці системи також дають стратегічну перевагу над веб-сайтами без рекомендаційних систем.

Ідея цих систем полягає в тому, що якщо звузити групу варіантів вибору для клієнтів до кількох значущих варіантів, вони, швидше за все, не будуть відкладати купівлю, а зроблять покупку саме зараз. Системи рекомендацій працюють, і вони добре працюють для деяких великих компаній. Наприклад відомо, що близько 35% доходу Amazon надходить безпосередньо завдяки системі рекомендацій.

Інша компанія, що активно користується рекомендаціями – це Netflix. Замість того, щоб переглядати тисячі бокс-сетів та назв фільмів, Netflix представляє користувачам набагато вужчий вибір предметів, які їм, ймовірно, сподобаються. Ця можливість економить час та покращує взаємодію з користувачем. Завдяки цій функції Netflix досягла низьких показників скасування підписок, заощадивши компанії близько мільярда доларів на рік. У 2009 році Netflix навіть організувала конкурс з премію в один мільйон доларів тому, хто зміг би вдосконалити їх механізм рекомендацій.

Незважаючи на те, що системи рекомендацій використовуються протягом майже 20 років, в останні кілька років вони все більш набувають популярності у різноманітніх галузях, наприклад у таких як медицина фінанси,туристична галузь тощо.

Системи рекомендацій повинні знати користувачів краще, щоб бути ефективними з їх пропозицією. Отже, інформація, яку вони збирають та інтегрують, є критичним аспектом процесу. Це може бути інформація, що стосується явної взаємодії, наприклад, інформація про минулу діяльність користувача, його рейтинги, відгуки та інша інформація про профіль користувача, наприклад, стать, вік чи цілі інвестування. Вони можуть поєднуватися з неявною взаємодією, такою як пристрій, який використовує користувач для доступу, кліки на посилання, місцезнаходження та дати.

Рекомендаційні системи можуть відрізнятися за розміром та формою. Деякі системи рекомендацій порівнюють предмети з іншими предметами, тоді як інші порівнюють користувачей з іншими користувачами. Деякі вимагають реєстрації або мінімальну кількість оцінок предметів, інші - ні. Деякі з них активні лише тоді, коли користувачі перебувають на веб-сайті, інші використовують підписку на розсилку електронною поштою.

Така велика різниця в рекомендаційних системах показує, що незважаючи на великий обсяг проведених науково-практичних досліджень, приклади успішних розробок, запропоновані базові методи надання рекомендацій та визначену таксономію рекомендаційних систем, ще багато питань потребують подальшого ретельного аналізу та опрацювання. До таких питань відносяться вдосконалення методів збору даних про користувача та аналіз його поведінки, розвиток методів машинного навчвння та глибинного навчання, спрямованих на використання у проблемній області надання рекомендацій. При цьому розширюються як сфера застосування систем рекомендаційного типу, так й методи видобування корисних даних про поведінку користувачів та відповідні методи надання рекомендацій.

Особливу увагу слід призначити методам, які використовують різного рода природно-мовні тексті в якості джерел інформації про товари та/або користувачів. У цьому сегменті інформаційних джерел вдосконалюють низку методів машинного навчання щодо нейромережевого підходу до

обробки природно-мовних текстів (Natural Language Processing, NLP) з метою найбільш ефективного опрацювання текстів та подальшого видобування з них різної корисної інформації [4]. Таким чином, за допомогою обробки природно-мовних текстів поповнюються корисні дані про користувачів та товари, такі як думки покупців, вибіркові рекомендації, поведінка оглядача в магазині, формування «інформаційних портретів» або профілів користувачів. Все ці додаткові можливості сприятимуть вдосконаленню рекомендаційних алгоритмів та покращенню роботи систем надання рекомендацій в цілому.

Враховуючи вище згадане, темою магістерської кваліфікаційної роботи обрано дослідження та використання методів побудови систем послідовних рекомендацій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасного стану розвитку рекомендаційних систем

Розвиток Інтернету сприяє появленню нових типів веб-систем, призначених для підтримки прийняття рішень користувачами в різних галузях та в умовах певних обмежень. Одним з найпоширеніших типів таких систем стали рекомендаційні системи (Recommender Systems, RS). RS системи використовують як класичні, так і нові методи машинного та глибинного навчання щодо розв'язання головної задачі, а саме надання рекомендацій користувачеві ефективним способом.

Системи даного типу розвиваються протягом останніх 15 років. Найбільшу популярність вони здобули у галузях електронної комерції, стрімінгових сервісах та соціальних мережах. Сьогодні ці методи також застосовуються у різних прикладних галузях (зокрема у медицині).

Рекомендаційні системи, є частиною Інтернету майже два десятиліття. Десятки постачальників зробили свій внесок у розвиток рекомендаційних технологій та вивели їх на ринок. Сьогодні RS системи знаходяться у безлічі Інтернет-сервісів. Вони були розроблені з використанням різноманітних методів та інтерфейсів користувача. Вони можуть враховувати явні та неявні уподобання мільйонів користувачів (найчастіше з їх дозволу). Часто вони надають відповідні рекомендації, які збільшують дохід користувачів інтернет-сервісів, які ними керують.

Початок рекомендаційних систем було покладено завдяки дослідженням когнітивних наук та пошуку інформації, а першим їх (RS систем) проявом стала система зв'язку Usenet, створена Університетом Дюка у другій половині 1970-х, де користувачі мали можливість обмінюватися текстовим контентом між собою. Вони були розділені на

групи та підгрупи новин для полегшення пошуку, але вони не були безпосередньо побудовані або націлені на уподобання користувачів.

Одним з перших відомих винахідників, що реалізував підхід з рекомендаційною системою, був комп'ютерний бібліотекар Грюнді, який спочатку опитував користувачів про їх уподобання, а потім рекомендував їм книги з урахуванням цієї інформації. На основі зібраної інформації система розподіляла користувачів у стереотипну групу за допомогою досить примітивного методу, рекомендуючи одні й ті ж книги всім членам тієї самої групи. Зараз цей підхід може здатися трохи застарілим, але на той час це була інноваційна парадигма в автоматизованих послугах, оскільки вона була персоналізованою.

Сьогодні рекомендаційні механізми (Recommendation Engines) повністю змінились і працюють набагато ефективніше. Ці рекомендаційні механізми намагаються порекомендувати товар чи послуги. Вони, певним чином, намагаються звузити вибір, пропонуючи людям пропозиції, які вони, найімовірніше, придбають або використають. Системи рекомендацій є майже скрізь – від Amazon до Netflix; від Facebook до Linkedin.

На даний час інтернет-магазини стрімко розвиваються, і ми можемо отримати майже будь-який товар в один клік. Однак раніше фізичний простір для зберігання товарів був обмежений, отже власники виставляли на вітрину лише ті предмети, які були найбільш популярними. Це означало, що багато товарів навіть не демонструвались, хоча мали чудову якість. Інакше кажучи, власники крамниць повинні були попередньо відфільтрувати «контент».

Однак індустрія інтернет-магазинів змінила цей сценарій. Оскільки тепер місця було необмежено, необхідність попереднього фільтрування відпала. Але це навпаки породило явище, яке стало відомим як «ефект довгого хвоста» (Long Tail Effect). На рисунку 1.1 зображено графічне відображення ефекту довгого хвоста.

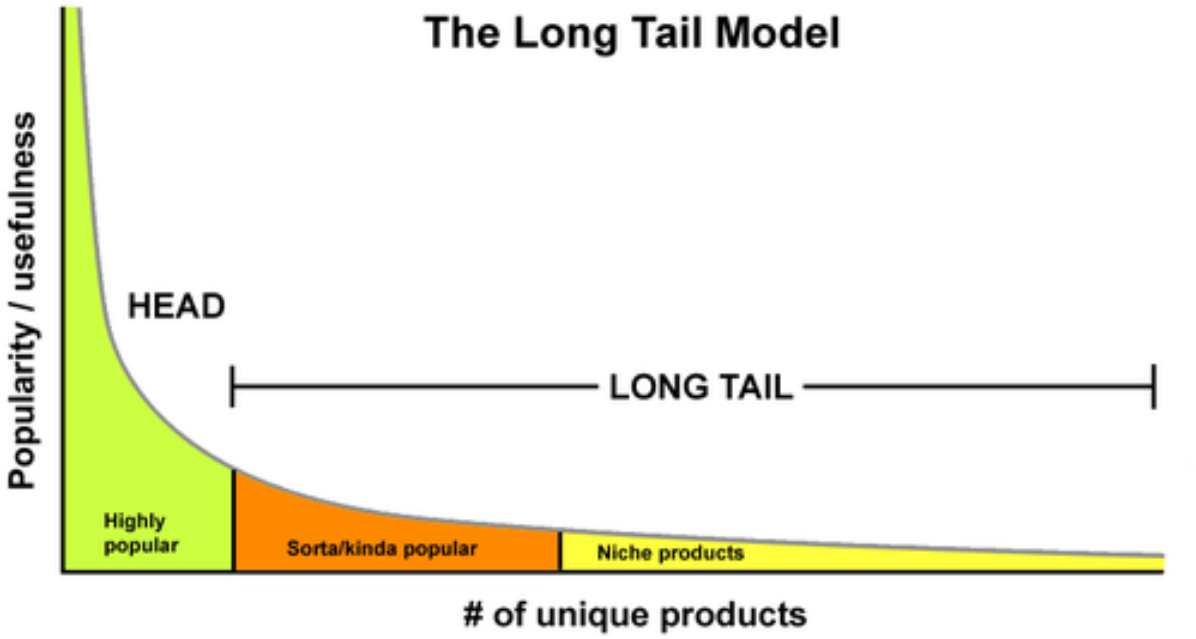

Рисунок 1.1 – Графічне зображення ефекту довгого хвоста

Зміст цього ефекту можна пояснити так: популярних продуктів мало, і їх можна знайти як в Інтернеті, так і в офлайн-магазинах. З іншого боку, менш популярних продуктів багато, і їх можна знайти лише в Інтернет-магазинах, що в кінцевому підсумку становить довгий хвіст. Однак навіть непопулярні товари можуть бути потрібними, і пошук таких товарів на веб-сайті є складним завданням і вимагає певної форми фільтра. Такий фільтр насправді і є системи рекомендацій.

Важливим аспектом у побудові рекомендаційних систем – інформація про уподобання користувачів. Дані про вподобання користувачів збираються двома способами:

– явні дані (Explicit Data): приклад явного збору даних – прохання користувачів оцінити товар за шкалою від однієї до п’яти зірок або оцінити контент, який вони бачать, із позначкою «подобається» або «не подобається» (like, dislike). У цих випадках користувачів чітко запитують, подобається їм певний предмет чи ні, і ці дані потім використовуються для створення профілю уподобань цього користувача. Однак у цього способу є недолік, оскільки не кожен користувач залишає відгуки чи оцінки, і навіть

якщо вони залишать оцінку, це може означати різне для різних людей. Наприклад, рейтинг 3 зірки може бути чудовим результатом для одного користувача, але середнім для іншого;

– неявні дані (Implicit Data): неявні дані надходять від взаємодії користувача з веб-сайтом та інтерпретації цих взаємодій як ознак зацікавленості чи незацікавленості. Наприклад, придбання продукту в Amazon або перегляд повного кліпу на YouTube розглядається як ознака позитивного інтересу. Неявна взаємодія може надати набагато більше даних для роботи.

Кінцевою метою рекомендаційних систем є збільшення продажів компанії. Для цього системи рекомендацій повинні відображати або надавати користувачеві лише релевантні товари (послуги). Дослідники рекомендаційних систем підсумовують бажані цілі механізмів рекомендацій у наступних чотирьох пунктах:

– актуальність: рекомендовані предмети матимуть сенс лише в тому випадку, якщо вони стосуються користувача користувачі частіше купують або споживають речі, які їм здаються цікавими;

– новизна: поряд з актуальністю, новизна є ще одним життєво важливим фактором, рекомендовані товари матимуть більше сенсу, якщо предмети – це те, що користувач раніше не бачив і не споживав;

– випадковість: іноді рекомендування товарів, які є дещо несподіваними, також може збільшити продажі;

– різноманітність: також не менш важливим є збільшення різноманітності рекомендацій, просто рекомендувати предмети, схожі між собою, не надто корисно.

Взагалі, рекомендаційні системи мають багато плюсів як для компаній, що використовують ці системи, так і для кінцевих користувачів. Ці компанії можуть розраховувати на збільшення продажів завдяки дуже персоналізованим пропозиціям та покращеній роботі з клієнтами. Рекомендації, як правило, пришвидшують пошук і полегшують

користувачам доступ до контенту, який їх цікавить, і дивують їх пропозиціями, яких вони ніколи б не шукали. Більше того, компанії можуть залучати та утримувати клієнтів, надсилаючи електронні листи із посиланнями на нові пропозиції, що відповідають інтересам одержувачів. Користувач починає відчувати себе відомим і зрозумілим і, швидше за все, купує додаткові продукти або почне споживати більше контенту. Знаючи, чого хоче користувач, компанія отримує конкурентні переваги, і загроза втрати клієнта для конкурента зменшується. Забезпечення доданої вартості для користувачів шляхом включення рекомендацій до систем та продуктів є привабливим. Крім того, це дозволяє компаніям випередити своїх конкурентів і врешті-решт збільшити свої прибутки.

На сьогодні у різноманітних галузях працює велике число систем рекомендацій. Однак важливим кроком є вирішення, який тип відповідає потребам і які доступні дані. Розрізняють два основних типи методів для рекомендаційних систем; контентна фільтрація та колаборативна фільтрація.

Колаборативна фільтрація намагається відобразити смак (профіль) користувачів і запропонувати їм контент, який сподобався користувачам із подібними уподобаннями. Контентна фільтрація базується на знаннях характеристиках сутності, яку слід рекомендувати (наприклад, система рекомендацій музичного контенту може враховувати такі характеристики: стиль, виконавець, жанр тощо) та вподобання користувача щодо цих характеристик. Таким чином, кожного разу, коли користувачеві подобається інша пісня, ця нова інформація додається до його профілю.

1.2 Метод контентної фільтрації для надання рекомендацій користувачу

Рекомендаційна система на основі контентної фільтрації працює з даними, які надає користувач, явно або неявно (натискання на посилання).

На основі цих даних створюється профіль користувача, який потім використовується для надання рекомендацій користувачеві. Коли користувач надає більше вхідних даних або здійснює дії, що допомагають рекомендаціям, рекомендаційний механізм стає все більш точним.

Контентна фільтрація базується на взаємодії з користувачем та його уподобаннях [5]. Рекомендації базуються на метаданих, зібраних з історії та взаємодій користувача. Наприклад, рекомендації базуватимуться на вивченні встановлених закономірностей у виборі або поведінці користувача. Така інформація, як товари чи послуги, відражають наші вподобання чи погляди. При такому підході, чим більше інформації надає користувач, тим вища точність.

Системи, що реалізують контентну фільтрацію, аналізують набір документів та / або описи речей (товарів), які раніше були оцінені користувачем, та створюють профіль інтересів користувача на основі особливостей об’єктів, оцінених цим користувачем [6]. Профіль – це структуроване представлення інтересів користувачів, адаптоване для рекомендації нових цікавих предметів.

Процес рекомендації в цілому полягає у зіставленні атрибутів профілю користувача з атрибутами об’єкта контенту. Результатом є оцінка релевантності, яка відображає рівень зацікавленості користувача в цьому об’єкті. Якщо профіль точно відображає вподобання користувачів, це має надзвичайну перевагу для ефективності процесу доступу до інформації. Наприклад, його можна використовувати для фільтрування результатів пошуку, вирішуючи, чи зацікавлений користувач у певній веб-сторінці чи ні, і, в негативному випадку, запобігати її відображенню.

Методи контентної фільтрації страждають набагато менше від проблеми холодного старту, ніж підходи колаборативної фільтрації: нових користувачів або товари (items) можна описати за їх характеристиками, і тому для цих нових сутностей можна зробити відповідні пропозиції. Від

цього недоліку будуть страждати лише нові користувачі або товари з ще невивченими новими характеристиками.

Проблема холодного старту дуже розповсюджена у рекомендаційних системах. Цей термін прийшов з галузі автомобілів. Коли дуже холодно, двигун має проблеми із запуском, але як тільки він досягне оптимальної робочої температури, він буде працювати безперебійно. У контексті рекомендаційних систем «холодний старт» означає, що обставини поки не оптимальні для того, щоб система забезпечувала найкращі можливі результати. Тобто система не має жодної історії переглядів користувача або історії взаємодії з товаром, на якій би базувались рекомендації.

Є декілька способів як вирішить цю задачу. Основна ідея полягає у використанні будь-якої з наявних даних про користувача чи товар для надання рекомендацій. Найпоширеніші методи, що використовуються для вирішення проблеми холодного старту це:

– жадібні методи (Greedy Methods): використання жадібних методів є найосновнішим способом вирішення проблем холодного старту, оскільки вони дуже елементарні та прості у реалізації. Ці методи передбачають використання якогось легко обчислюваного алгоритму надання рекомендацій користувачам (наприклад випадковий нормальний предиктор (Random Normal Predictor) або рекомендація найпопулярнішого товару);

– методи подібності на основі вмісту (Content-based similarity Methods): якщо користувач дав явний або неявний зворотний зв’язок лише для кількох товарів, то колаборативна фільтрація може не дати хороших результатів. Однак можна виміряти схожість цих предметів з іншими елементами, про які користувач не дав зворотного зв’язку, та рекомендувати найближчі на основі оцінок подібності. Це також вирішить проблему, пов’язану з рекомендацією нових елементів, з якими ще ніхто не взаємодіяв, оскільки розглядаються лише метадані елементів;

– багаторукий бандит (Multi-armed bandit): метод багаторукого бандита заснований на поєднанні двох попередніх методів, щоб спочатку показати рекомендації випадковим чином, а потім ітеративно вдосконалювати їх, коли користувач взаємодіє з цими рекомендованими предметами.

1.3 Метод колаборативної фільтрації для надання рекомендацій користувачу

Колаборативна фільтрація – ще одна часто застосовувана техніка. Колаборативна фільтрація створює набагато ширшу мережу, збираючи інформацію про взаємодії багатьох інших користувачів, щоб отримати пропозиції для одного користувача [5]. Цей підхід дає рекомендації базуючись на інших користувачах зі схожими смаками. Наприклад, використовуючи їх думку та дії, щоб рекомендувати іншому користувачеві товари або визначати, як один товар може поєднуватися з іншим.

Метод колаборативної фільтрації зазвичай має вищу точність, ніж контентна фільтрація; однак колаборативна фільтрація також може внести деяку підвищену мінливість та іноді менш зрозумілі результати. Даний підхід особливо слабкий у ситуаціях коли відсутні раніше зібрані данні.

На рисунку 1.2 відображено основну схему роботи колаборативної та контентної фільтрацій.

Як правило, більшість комерційних рекомендаційних систем засновані на великій кількості даних (товарів), в той час, як більшість користувачів не ставить оцінки товарам. В результаті цього матриця «предмет-користувач» виходить дуже великою і розрідженою, що викликає особливі проблеми при обчисленні рекомендацій. Ця проблема особливо гостра для нових систем.

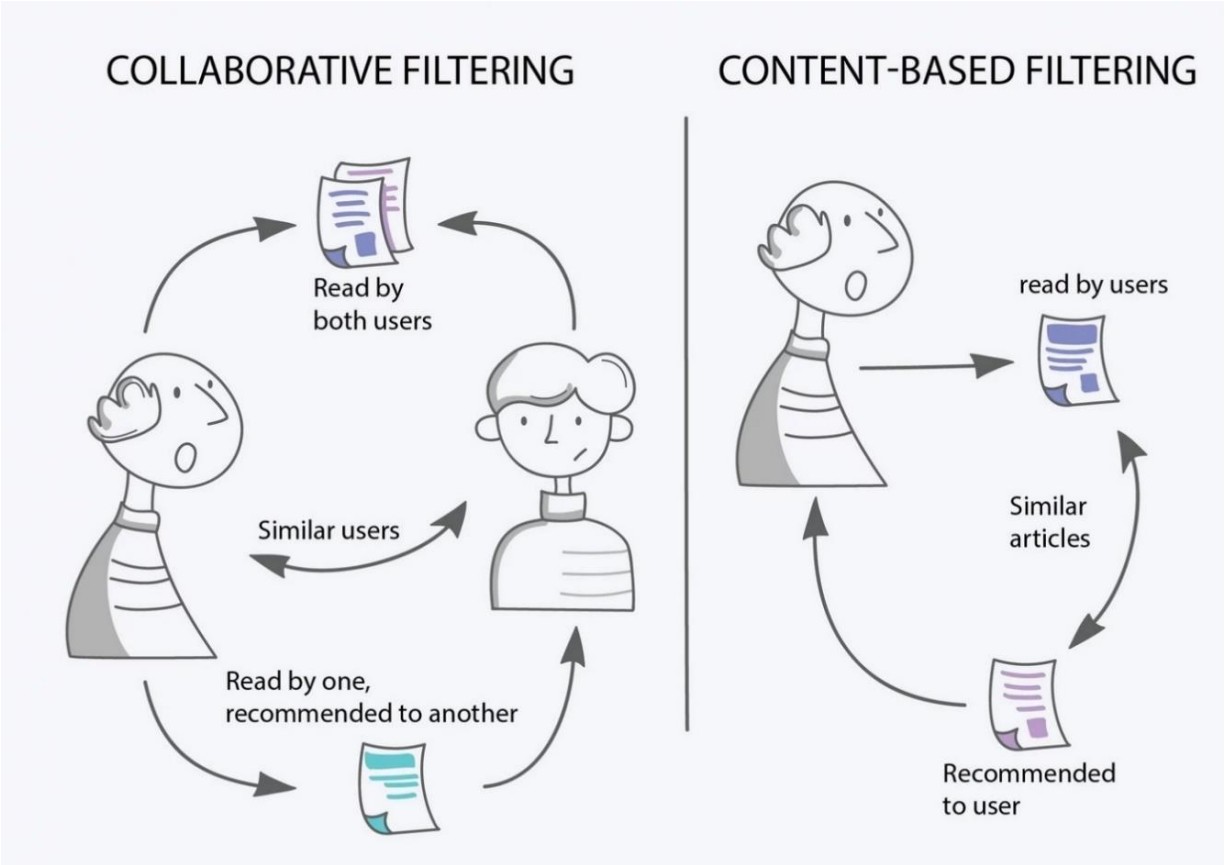

Рисунок 1.2 – Різниця між колаборативною фільтрацією та контентною фільтрацією

Існує два підходи колаборативної фільтрації:

а) методи, засновані на пам’яті (memory-based methods), які також називають алгоритмами колаборативної фільтрації на основі сусідства (neighborhood-based collaborative filtering), в яких рейтинги комбінацій елементів користувача прогнозуються на основі їх «сусідства». Ці «сусідства» можна визначити одним із двох способів:

1) колаборативна фільтрація на основі користувачів (User-based): спосіб полягає у тому, щоб знайти інших, схожих людей, і рекомендувати предмети, які їм сподобались;

2) колаборативна фільтрація на основі об’єктів уподобання (Item-based): рекомендувати товари, що купили люди, які також купували речі, які сподобалися цільовому користувачу;

б) методи, засновані на моделях (model-based methods): використовують методи машинного навчання для отримання прогнозів для рейтингових даних, розглядають задачу як звичайну задачу машинного навчання.

1.4 Гібридні підходи

Також існує гібридний підхід, що поєднує дві рекомендаційні методики (контентну фільтрацію та колаборативну фільтрацію) для того, щоб отримати кращий результат та зменшити недоліки та похибки цих методів [7].

Гібридні підходи діляться на декілька видів [8]:

- зважений метод (weighted): чисельна комбінація кожного рекомендованого компоненту, в ході отримується інша оцінка системою;
- переключення (switching): система має кілька варіантів вибору різних елементів рекомендацій для користувача і вибирає бажаний відповідно до уподобань користувача;
- змішаний метод (mixed): система рекомендує користувачеві кілька різних елементів одночасно;
- комбінація ознак (feature combination): кілька джерел знань об’єднуються для створення особливостей системи рекомендацій (recommendation system features);
- аугментація ознак (feature augmentation): однією з важливих частин техніки є аугментація об’єктів, яке використовуються для обчислення набору особливостей рекомендаційних систем;
- каскадний метод (cascade): у рекомендаційному списку знаходиться зважений пріоритет, об'єкт інтересу, що має вищий пріоритет, з'являється першим, далі з'являються об'єкти з пріоритетами нижче;

– метарівень (meta-level): це один із методів введення інформації, який використовується для генерації та створення якоїсь моделі для наступного кроку алгоритму системи рекомендацій.

Об'єднання цих методів дозволяє досягти високої ефективності та полегшити проблеми та труднощі, що виникають при використанні лише контентної фільтрації або коллаборативної фільтрації.

1.5 Аналіз систем послідовних рекомендацій

Системи послідовних рекомендацій (Sequential recommender systems, SRSs), також відомі як sequence-aware recommendations, пропонують користувачам об’єкти (товари), які їх можуть зацікавити, в основному моделюючи послідовні залежності з взаємодій між об’єктами інтересу користувача у послідовності (наприклад, в ситуаціях коли користувач переглядає або купує товари в Інтернеті). Традиційні системи рекомендацій, включаючи контентну та колаборативну фільтрації, моделюють взаємодію між елементами користувача в статичному режимі й можуть враховувати лише загальні уподобання користувачів. На відміну від цього, послідовні рекомендації трактують взаємодію між елементами користувача як динамічну послідовність та беруть до уваги послідовні залежності, щоб врахувати поточні та останні уподобання користувача для отримання більш точних рекомендацій [9]. На рисунку 1.3 відображено схему роботи послідовних рекомендацій.

Рисунок 1.3 – Приклад роботи системи послідовних рекомендацій

Згідно з [1] система послідовних рекомендацій приймає за вхідні дані о послідовності взаємодій між об’єктами й користувачем та намагається передбачити подальші взаємодії між ними, які можуть відбутися найближчим часом, шляхом моделювання складних послідовних залежностей. Інакше кажучи, враховуючи послідовність взаємодій між об’єктом та користувачем, список рекомендацій, що складається з найвищих елементів у рейтинговому списку кандидатів (об’єктів), генерується шляхом максимізації значення функції корисності (наприклад, ймовірності):

$$R = \arg\max f(s), \tag{1.1}$$

де f – це функція корисності що виводить рейтингові оцінки для кандидатів;

S = $\{i_1, i_2, ..., i_{|S|}\}$ – це послідовність взаємодій між користувачем та об’єктами, де кожна взаємодія $i_j = <u, a, v>$ є потрійною, що складається з користувача u, дії користувача a, та відповідний об’єкт v. Дії можуть бути різного типу (наприклад, клік, додавання до кошика, купівля) і відбуватися в різних контекстах (час, місце, погода);

R – це список елементів, упорядкованих за рейтинговою оцінкою.

Важливою метою послідовних рекомендацій є підвищення якості та ефективності рекомендацій. Хоча було розроблено багато методів та алгоритмів, послідовні рекомендаційні системи все ще перебувають на початковій стадії вивчення.

1.5.1 Традиційні методи побудови послідовних рекомендацій

Традиційні популярні методи до побудови систем послідовних рекомендацій включають пошук шаблонів що зустрічаються найчастіше (frequent pattern mining), метод k -найближчих сусідів, ланцюги Маркова, факторизація матриць та навчання з підкріпленням [10]. Вони, як

правило, застосовують факторизацію матриць для розгляду довгострокових уподобань користувачів у різних послідовностях. Також використовуються ланцюги Маркова першого порядку для фіксації короткотермінового інтересу користувачів у послідовності [11]. Розглянемо більш детально дані методи.

Задача пошуку шаблонів що зустрічаються найчастіше (frequent pattern mining) полягає у пошуку зв'язків між елементами бази даних. Автори [12] визначають цю задачу наступним чином: враховуючи базу даних D з транзакціями T_1 ... T_N, визначити всі шаблони P, які є принаймні в частці транзакцій s. Спочатку задача була запропонована в контексті даних ринкових кошиків, щоб знайти групи предметів, які купуються разом найчастіше.

У задачі послідовних рекомендацій шаблони – це набори елементів, які часто зустрічаються в межах послідовності, а потім використовуються для надання рекомендацій. Не дивлячись на те, що ці підходи є простими у впровадженні та ясні для користувачів, вони зазвичай страждають від обмеженої проблеми масштабованості. Це зв’язано з тим, що узгодження шаблонів для рекомендацій дуже трудомістке. Іншим недоліком є складність визначення відповідних порогових значень показників підтримки та впевненості. Низький показник підтримки або впевненості призведе до занадто великої кількості виявлених закономірностей, тоді як при великому показнику алгоритм буде просто визначати товари, що зустрічаються разом, з дуже високою частотою, в результаті чого лише декілька елементів будуть рекомендовані, або небагато користувачів можуть отримати ефективні рекомендації [13].

У роботі [14] було створено персоналізовану структуру рекомендацій на основі аналізу послідовних зразків. Автори використовують новий показник оцінки компетентності, тому запропонована структура ефективно вивчає специфічні для користувача знання у послідовності та використовує ці додаткові знання для персональних рекомендацій.

Другий метод, метод k-найближчих сусідів (KNN) – це простий та найбільш популярний алгоритм машинного навчання з вчителем, який використовується для вирішення задач класифікації та регресії. Цей метод легко інтегрувати, але він має недолік – він починає значно сповільнюватись коли обсяг цих даних, що використовуються, зростає. KNN знаходить відстань між запитом та всіма прикладами в даних, вибирає вказану кількість прикладів (K), найближчу до запиту, потім голосує за найпоширенішу мітку (у разі класифікації) або усереднює мітки (у випадок регресії). На рисунку 1.4 представлено графічне зображення методу k-найближчих сусідів.

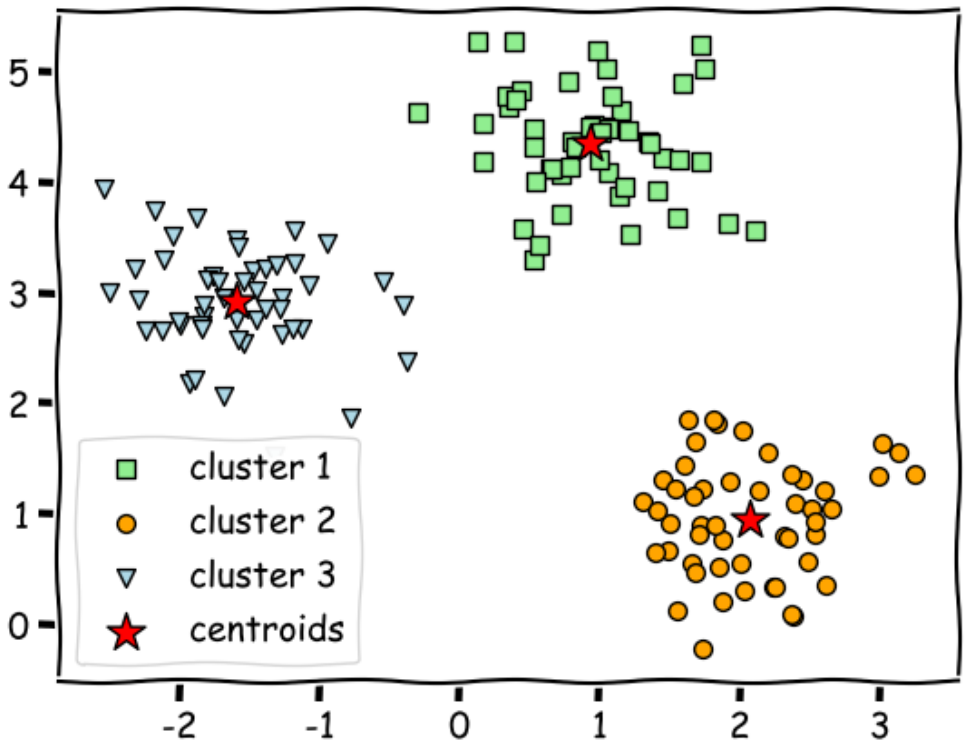

Рисунок 1.4 – Графічне зображення методу k-найближчих сусідів

KNN на основі об’єктів (item-based KNN) [15] враховує лише останню поведінку в даному сеансі та рекомендує елементи, найбільш схожі на

об’єкт поведінки (item), де подібність обчислюють за допомогою подібності косинусів або за допомогою інших вимірювань.

Однак, KNN на основі сеансу (session-based KNN) [16] працюють інакше:вони порівнюють весь сеанс з усіма минулими сеансами та рекомендують елементи обчислюючи подібності за допомогою індексу Жаккарда або подібності косинусів на бінарних векторах в просторі елементів. Методи KNN можуть генерувати дуже пояснювані рекомендації. Крім того, оскільки подібності можна попередньо розрахувати, системи рекомендацій на базі KNN можуть швидко формувати рекомендації. Однак цей тип алгоритмів, як правило, не враховує послідовну залежність між елементами.

Інший метод, Ланцюг Маркова є стохастичною моделлю, яка описує послідовність можливих подій, в яких ймовірність кожної події залежить лише від стану, отриманому у попередній події. Ланцюги Маркова є досить поширеним і відносно простим способом статистичного моделювання випадкових процесів. Вони використовувались у багатьох різних сферах – від генерації тексту до фінансового моделювання. Загалом, ланцюги Маркова концептуально досить інтуїтивні і їх можна реалізувати без використання будь-яких передових статистичних або математичних концепцій. На рисунку 1.5 зображено ланцюг Маркова.

У послідовній рекомендації моделі Маркова припускають, що майбутня поведінка користувачів залежить лише від останніх чи останніх кількох видів поведінки. Послідовні рекомендації на основі ланцюгів Маркова поділяються на: базові методи, що базуються на ланцюгах Маркова та підходи на основі латентного ембедінгу Маркова (latent Markov embedding-based method). Перша група методів обчислює ймовірність переходу на основі явних спостережень [17], тоді як друга спочатку вбудовує ланцюги Маркова в евклідовий простір, а потім обчислює ймовірності переходів між взаємодіями на основі їх евклідової відстані.

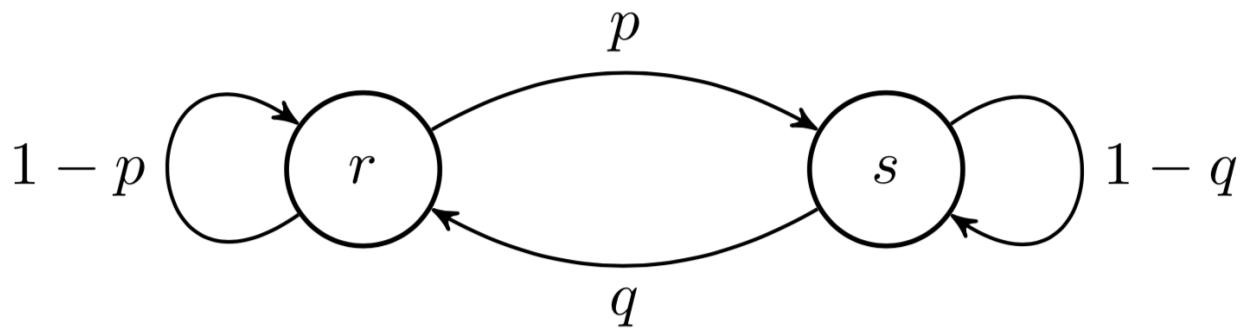

Рисунок 1.5 – Ланцюг Маркова

Рекомендаційні системи на основі ланцюгів Маркова можуть захоплювати лише короткочасні залежності, ігноруючи довгострокові, завдяки властивості Маркова, яка передбачає, що поточна взаємодія залежить тільки від одного або декількох останніх взаємодій. З іншого боку, вони можуть фіксувати лише точкові залежності, ігноруючи колективні залежності від взаємодії між елементами користувача. Врахування лише останній вид поведінки або декількох видів поведінки робить моделі, що базуються на ланцюгах Маркова, не в змозі використовувати залежності між поведінкою у відносно довгій послідовності, і, отже, їм не вдається вловити складну динаміку більш складних сценаріїв. Крім того, вони також можуть страждати від проблем з розрідженістю даних. Тому вони дедалі менше використовуються у системах послідовних рекомендацій.

Ще один традиційний підхід до розв'язання задачі побудови системи послідовних рекомендацій – метод факторизації матриць (matrix factorization). Даний метод намагається розкласти матрицю взаємодії об’єкт-користувач на дві матриці низького рангу. Методи, що використовують факторизацію, для систем послідовних рекомендацій зазвичай використовують факторизацію матриці або тензорну факторизацію, щоб факторизувати спостережувані взаємодії між об’єктами інтересу та користувачами у латентні (приховані) фактори користувачів та об’єктів для

рекомендацій [18]. На рисунку 1.6 зображено схему роботи методу факторизації матриць.

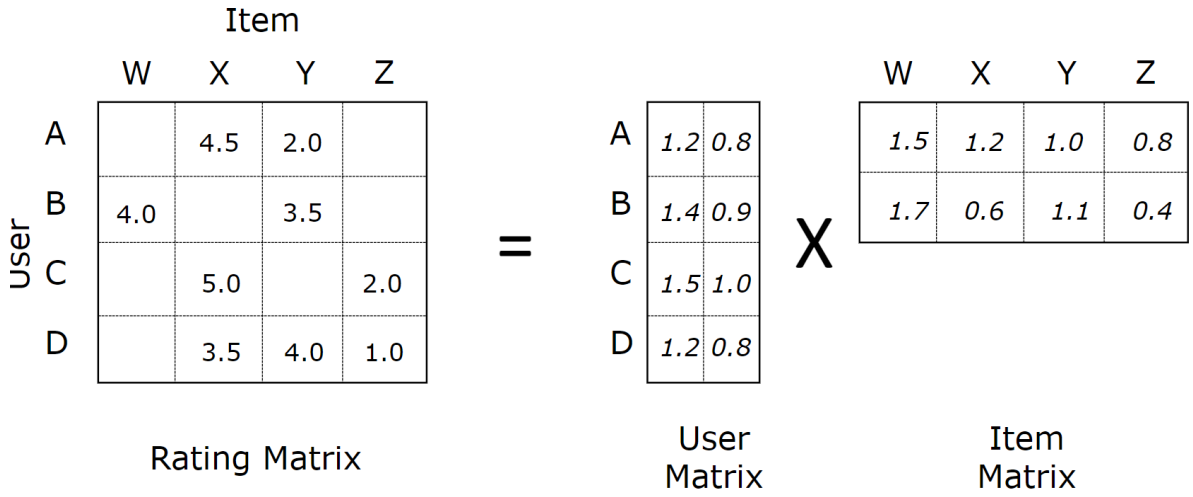

Рисунок 1.6 – Схема роботи методу факторизації матриць

На відміну від колаборативної фільтрації, матриця або тензор, що підлягає факторизації, складається з взаємодій, а не з оцінок. На таку модель впливає розрідженість спостережуваних даних і, отже, вона не може досягти ідеальних рекомендацій.

Іншими недоліками методів, що використовують факторизацію є той факт, що більшість з них враховує лише взаємодії низького порядку (першого та другого) серед прихованих факторів, але вони ігнорують можливі взаємодії високого порядку. Також ці методи (за винятком деяких, що враховують тимчасову інформацію), зазвичай, ігнорують залежність від часу у поведінках як у межах сеансу, так і в різних сеансах.

Наступний підхід до до розв'язання задачі побудови системи послідовних рекомендацій – методи з використанням ембедінгів. Ембедінг (embedding) – це відносно низьковимірний простір, в який можна переміщувати вектори високих розмірів. Ембедінги полегшують роботу машинного навчання у випадках коли на вхід подається велика кількість

даних (наприклад вектори, що представляють слова). Ембедінги можна вивчити та повторно використовувати в різних моделях.

Методи послідовних рекомендацій, що використовують ембедінги, вивчають приховані представлення для кожного користувача та об’єкта для подальших рекомендацій шляхом кодування всіх послідовних взаємодій між об’єктами інтересу та користувачем у прихований простір. Деякі методи [19] беруть вивчені приховані представлення як вхідні дані мережі для подальшого обчислення оцінки взаємодії між користувачами та об’єктами або послідовних дій користувачів. Ці методи демонструє виняткові результати завдяки своїй простоті та ефективності.

Навчання з підкріпленням (reinforcement learning) – це традиційний метод послідовних рекомендацій. Він є методом навчання моделей машинного навчання для прийняття послідовності рішень. Агент вчиться досягати мети в невизначеному, потенційно складному середовищі. У цьому виді навчання штучний інтелект знаходиться в ігровій ситуації. Комп’ютер використовує метод спроб і помилок, щоб знайти розв'язання задачі. Щоб примусити машину виконувати завдання, штучний інтелект отримує винагороду, або покарання за вчинені ними дії. Його мета – максимізувати загальну винагороду. На рисунку 1.7 зображено схематичний приклад роботи агента для методів навчання з підкріпленням.

Суть методів на основі навчання з підкріпленням полягає в оновленні рекомендацій відповідно до взаємодії користувачів та систем, що рекомендують. Коли система рекомендує користувачеві предмет, позитивна винагорода призначається, якщо користувач виявляє свою зацікавленість (перегляд або купівля предмету). За допомогою такого виду навчання системи послідовних рекомендацій можуть динамічно пристосовуватися до налаштувань користувачів. Однак цей вид методів також має недолік у вигляді відсутності інтерпретації. Крім того, що більш важливо, у наукових колах є небагато відповідних платформ або ресурсів

для розробки та тестування методів, заснованих на навчанні з підкріпленням.

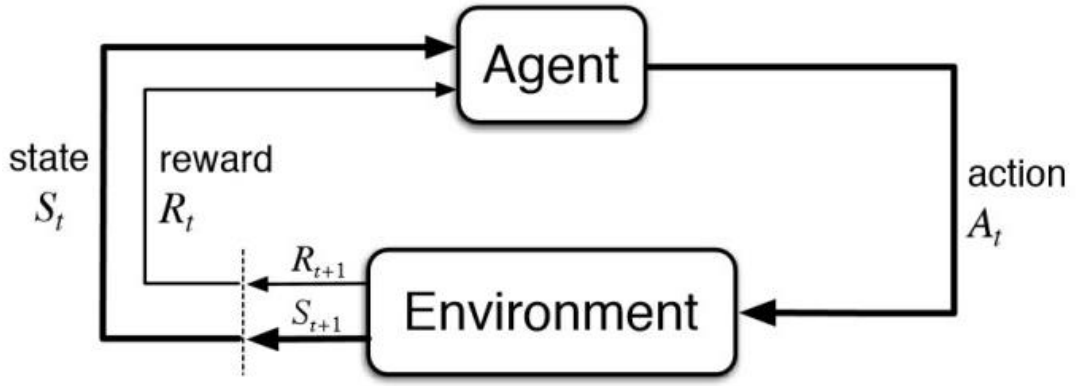

Рисунок 1.7 – Схема роботи агента у методах навчання з підкріпленням

1.5.2 Методи глибинного навчання

Традиційні підходи завжди використовують, як правило, лише обмежену інформацію про об’єкти та моделюють лише односторонні переходи між сусідніми елементами, що ускладнює моделювання детально визначених уподобань користувачей. Вони також ігнорують складні переходи між віддаленими елементами. Тому традиційні методи не в змозі розв'язувати усі задачі послідовних рекомендацій. На зміну їм почали застосовуватись методи, що використовують глибокі нейронні мережі. Глибокі нейронні мережі активно застосовуються в галузі послідовних рекомендацій протягом останніх кількох років. Значущі досягнення були здобуті завдяки цим методам.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) найчастіше використовують для систем послідовних рекомендацій, тому що вони мають природну силу у моделюванні послідовностей, але вони також мають дефекти. Нещодавно згорткові нейронні мережі (Convolutional Neural

Networks, CNN) та графічні нейронні мережі (Graph Neural Networks, GNN) теж почали застосовуватися у послідовних рекомендаціях, задля усунення дефектів RNN. Розглянемо більш детально кожен з цих методів.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) – це тип штучних нейронних мереж, які використовують послідовні дані або дані часових рядів. Ці види мереж, як правило, використовуються для часових або порядкових проблем, таких як переклад текстів, обробка природної мови (Natural Language Processing, NLP), розпізнавання мови; вони включені в такі популярні програми, як Siri, голосовий пошук та Google Translate. Рекурентні нейронні мережі (як і прямі та згорткові нейронні мережі) використовують тренувальні дані для навчання. Вони відомі своєю «пам’яттю», тобто вони беруть інформацію з декількох попередніх входів та впливають на поточний вхід і вихід. У той час як традиційні глибинні нейронні мережі припускають, що входи та виходи не залежать один від одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності, тому ці мережі й стали такі популярні у послідовних рекомендаціях. Хоча майбутні події також могли б допомогти у визначенні результату послідовності, однонапрямні рекурентні нейронні мережі не можуть враховувати ці події у своїх прогнозах. На рисунку 1.8 зображено різницю між рекурентною нейронною мережею та нейронною мережею прямого поширення.

Іншою характеристикою рекурентних нейронних мереж є те, що вони мають одні й ті ж самі параметрами у кожному рівні мережі. Хоча мережі прямого поширення мають різну вагу на кожному вузлі, рекурентні нейронні мережі мають однаковий вагомий параметр у кожному рівні мережі. Проте, ці ваги все ще коригуються в процесі зворотного розповсюдження та градієнтного спуску для полегшення навчання з підкріпленням.

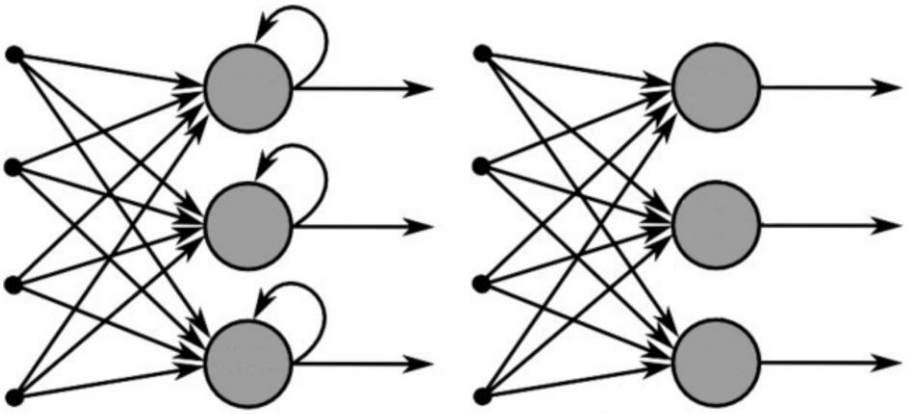

Рисунок 1.8 – Різниця архітектур рекурентної нейронної мережі та нейронної мережі прямого поширення

Рекурентні нейронні мережі використовують алгоритм зворотного розповсюдження в часі (Backpropagation through time, BPTT) для визначення градієнтів. Цей алгоритм дещо відрізняється від традиційного зворотного поширення, оскільки є специфічним для даних послідовностей. Принципи BPTT такий же, як традиційне зворотне розповсюдження, коли модель тренується сама, обчислюючи помилки від вихідного рівня до вихідного рівня. Ці розрахунки дозволяють правильно відрегулювати та підігнати параметри моделі. BPTT відрізняється від традиційного підходу тим, що BPTT підсумовує помилки на кожному часовому кроці, тоді як мережі прямого пересилання не потребують підсумовування помилок, оскільки вони не мають ті ж самі параметри на кожному рівні. Через цей процес RNN стикаються з двома проблемами – вибухові та зникаючі градієнти. Коли градієнт занадто малий, він продовжує зменшуватися, оновлюючи вагові параметри до тих пір, доки вони не стануть незначними – тобто 0. Коли це відбувається, алгоритм більше не навчається. Вибухові градієнти виникають у ситуаціях коли градієнт занадто великий, створюючи нестабільну модель. У цьому випадку ваги моделі зростуть занадто великими, і вони врешті-решт будуть представлені як NaN. Для вирішення

цього питання, як правило, зменшують кількість прихованих шарів у рекурентній нейронній мережі, тим самим усуваючи деякі складності в моделі RNN.

Моделі на основі RNN застосовуються найчастіше серед усіх моделей на основі глибокого навчання. У порівнянні з традиційними моделями, моделі послідовних рекомендацій на основі рекурентних мереж можуть добре фіксувати залежності між елементами в рамках сеансу або між різними сеансами. Враховуючи послідовність історичних взаємодій між об’єктами інтересу та користувачами, системи послідовних рекомендацій на основі RNN намагається передбачити наступну можливу взаємодію шляхом моделювання послідовних залежностей враховуючи дані взаємодії. RNN також були розроблені для охоплення довгострокових залежностей у послідовності. В останні роки системи послідовних рекомендацій на основі рекурентних нейронних мереж домінують у дослідженнях систем послідовних рекомендацій на основі глибокого навчання або навіть у галузі послідовних рекомендацій. Окрім базової структури RNN, існують також варіанти для захоплення більш складних залежностей у послідовності, наприклад ієрархічна RNN [20].

Основним обмеженням RNN для послідовних рекомендацій є те, що моделювати залежності в довгій послідовності відносно важко, а навчання є «дорогим», особливо зі збільшенням довжини послідовності. Також до недоліків можна віднести те, що легко створюються фальшиві залежності через надмірно строге припущення, що будь-які сусідні взаємодії в послідовності повинні бути залежними, що не завжди так у реальному світі, оскільки всередині послідовності зазвичай є нерелевантні або «шумні» взаємодії. До того ж RNN схильні до охоплення точкових залежностей при ігноруванні колективних залежностей (кілька взаємодій спільно впливають на наступну).

Графові нейронні мережі (Graph Neural Networks, GNN) – це клас методів глибокого навчання, призначений для виведення даних, описаних

графами. В інформатиці граф – це структура даних, що складається з двох компонентів: вузлів (вершин) і ребер.

Графові нейронні мережі – це нейронні мережі, які можна безпосередньо застосувати до графів. Вони забезпечують простий спосіб виконання завдань прогнозування на рівні вузла, ребра та на рівні графа. На рисунку 1.9 наведено графову нейронну мережу.

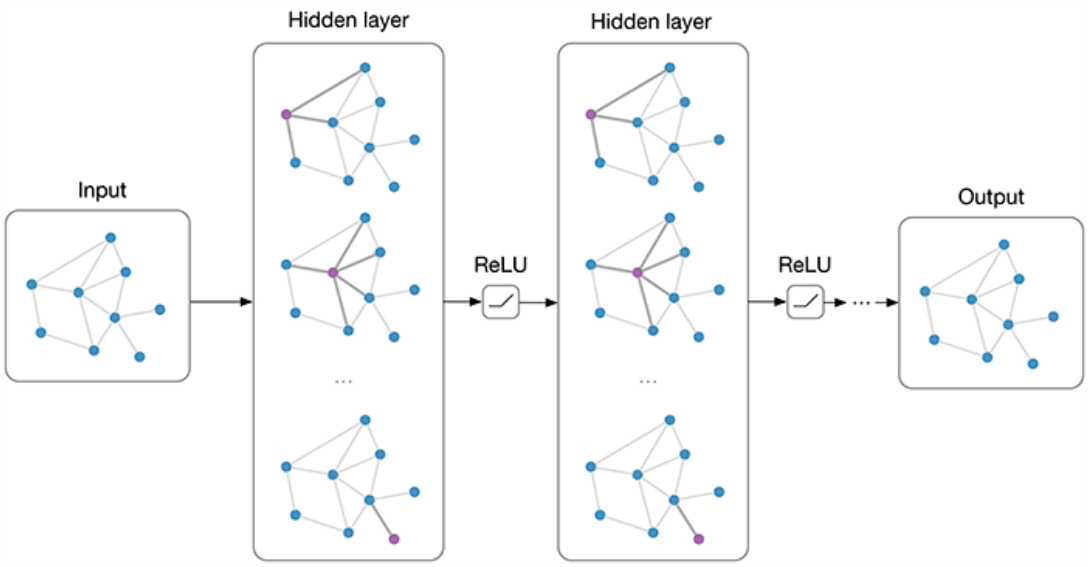

Рисунок 1.9 – Приклад графової нейронної мережі

Графи мають великий потенціал, тому привертають багато уваги у галузі машинного навчання. Кожен вузол має ембедінг, пов’язаний з ним, який визначає вузол у просторі даних. Графові нейронні мережі відносяться до архітектур нейронних мереж, які працюють на графі. Мета GNN полягає в тому, щоб кожен вузол у графі вивчав ембедінг, що містить інформацію про його сусідство (вузли, безпосередньо з'єднані з цільовим вузлом через ребра). Потім цей ембедінг можна використовувати для різних задач. Таким чином, після ембедінгу, пов’язаного з кожним вузлом, можна модифікувати ребра, додаючи шари нейронної мережі прямого поширювання та комбінуючи графи й нейронні мережі.

Останнім часом, графові нейронні мережі дуже швидко розвиваються, тому у системах послідовних рекомендацій GNN використовуються для моделювання та фіксації складних переходів під час взаємодії користувачів та об’єктами інтересу в послідовності. Зазвичай спочатку будується спрямований граф на даних послідовності, приймаючи кожну взаємодію як вузол у графу. Потім ембедінг користувачів або об’єктів вивчаються на графіку, щоб інтегрувати більш складні відносини в цілому графіку [21]. Хоча графи складно аналізувати, послідовні рекомендації на основі GNN мають потенціал для надання пояснюваних рекомендацій, виявляючи складні взаємозв'язки між рекомендованими елементами та відповідним послідовним контекстом. Ці послідовні системи все ще перебувають на початковій стадії і до кінця не вивчені дослідниками.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) зазвичай застосовується для обробки даних часових рядів та зображень. Типова структура CNN складається із шарів згортки, шарів пулінгу (pooling layers) та повно з'єднаних шарів прямого поширення. Дана структура підходить для фіксації залежних зв’язків в локальній інформації (наприклад, співвідношення між пікселями в певній частині зображення або залежності між кількома сусідніми словами в реченні).

На відміну від RNN, з огляду на послідовність взаємодій між користувачем та об’єктами інтересу, CNN спочатку розташовують всі ембедінги цих взаємодій у матрицю, а потім розглядають цю матрицю як «зображення» у часовому та латентному просторах. Потім ця мережа вивчає послідовні зразки як локальні особливості зображення, використовуючи згорткові фільтри для подальших рекомендацій. Оскільки CNN вивчають закономірності між областями на «зображенні», а не над взаємодіями, то можна сказати, що послідовні рекомендації на основі CNN можуть певною мірою усунути недоліки таких систем на основі RNN [22].

1.6 Постановка задач дослідження

Глибинне навчання є актуальною технікою для вирішення різноманітних задач як у сфері обробки даних, так і у сфері машинного навчання. Одним із найпопулярніших методів глибокого навчання є згорткові нейронні мережі. Вони знайшли своє застосування у різних галузях діяльності від комп'ютерного зору до обробки аудіо, відео та природної мови. Ці методи на практиці довели свою ефективність. З розвитком рекомендаційних систем, згорткові мережі також почали застосовуватись у цих задачах та приносити свої плоди.

Об’єктом дослідження даної магістерської роботи є система рекомендаційного типу з використанням згорткової нейронної мережі. Предметом дослідження є методи побудови систем послідовних рекомендацій.

Мета даної роботи – обґрунтування, вибір та використання методу побудови системи послідовних рекомендацій в заданій предметній області.

Відповідно до поставленої мети необхідно вирішити наступні задачі:

- провести аналіз науково-дослідних публікацій у галузі послідовних рекомендацій;
- проаналізувати існуючи підходи до побудови систем послідовних рекомендацій;
- на основі попереднього аналізу вибрати метод для вирішення задачі побудови системи послідовних рекомендацій з використанням згорткових нейронних мереж;
- вибрати архітектуру згорткової нейронної мережі;
- сформувати тренувальну вибірку для навчання нейронної мережі;
- перевірити працездатність побудованої системи послідовних рекомендацій;
- проаналізувати отримані результати.

2 НАДАННЯ ПОСЛІДОВНИХ РЕКОМЕНДАЦІЙ З ВИКОРИСТАННЯМ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

2.1 Поняття згорткової нейронної мережі

Згорткові нейронні мережі (Convolutional Neural Network, CNN) є класом глибоких штучних нейронних мереж, що найчастіше застосовуються в задачах аналізу візуальних зображень. Даний вид мереж може приймати вхідне зображення, призначати важливість (завдяки зважуваним вагам та зміщенню) різним аспектам/об’єктам на зображенні та мати можливість відрізняти одне зображення від іншого. Попередня обробка (pre-processing), необхідна CNN набагато менше ніж іншим алгоритмами класифікації.

Архітектура згорткових нейронних мереж натхненна організацією зорової кори. Окремі нейрони реагують на подразники в обмеженій площині зорового поля, відомому як рецептивне поле. Колекція таких полів перекривається, щоб охопити всю зорову зону.

Вперше згорткові нейронні мережі були представлені в 1980-х роках Янном ЛеКуном, докторським дослідником інформатики. Рання версія CNN, що називається LeNet, могла розпізнавати рукописні цифри [23]. CNN знайшли свою нішу на ринку банківських та поштових послуг та банківської справи, де вони читали поштові індекси на конвертах та цифри на чеках. Архітектура цих мереж була натхненна біологічними нейронами, які взаємодіють між собою та генерують результати, що залежать від вхідних даних. Хоча робота над цими мережами розпочалася на у 80-х роках минулого сторіччя, вони стали популярними лише завдяки останнім технологічним вдосконаленням та обчислювальним можливостям, які дозволяють обробляти великі обсяги даних та навчати складним алгоритмам за досить невеликий проміжок часу.

У 2012 році була представлена згорткова мережа AlexNet [24], що змінило уявлення про згорткові нейронні мережі. Наявність великих наборів даних, а саме датасету даних ImageNet, що містить мільйон маркірованих зображень та потребує величезних обчислювальних ресурсів, дозволило дослідникам створювати складні мережі CNN, які могли виконувати завдання комп'ютерного зору, які раніше були неможливі.

Сьогодні CNN застосовується у програмах з віртуальним асистентом на основі штучного інтелекту, у програмах автоматичного тегування фотографій, у задачах розпізнаванні образів, маркування відео тощо. CNN також застосовуються у програмах для самокерованих автомобілів, робототехніки, безпілотників, безпеки, медичних діагнозів та лікування людей з вадами зору. Згорткові нейронні мережі можуть також виконувати більш банальні (і більш вигідні) бізнес-орієнтовані задачі, такі як оптичне розпізнавання символів, щоб оцифрувати текст і зробити можливим обробку природних мов в аналогових та рукописних документах, де зображення є символами, які слід розшифрувати.

Однак CNN не обмежуються розпізнаванням зображень. Вони також активно застосовуються у задачах аналізу тексту. І вони можуть застосовуватися до звуку, коли він представлений візуально у вигляді спектрограми, і графічних даних із графічними згортковими мережами.

2.2 Архітектура згорткової нейронної мережі

Згорткові нейронні мережі складаються з безлічі шарів штучних нейронів. Штучні нейрони, груба імітація їх біологічних аналогів, є математичними функціями, які обчислюють зважену суму кількох входів і виводять активаційне значення. На рисунку 2.1 зображено архітектуру штучного нейрона.

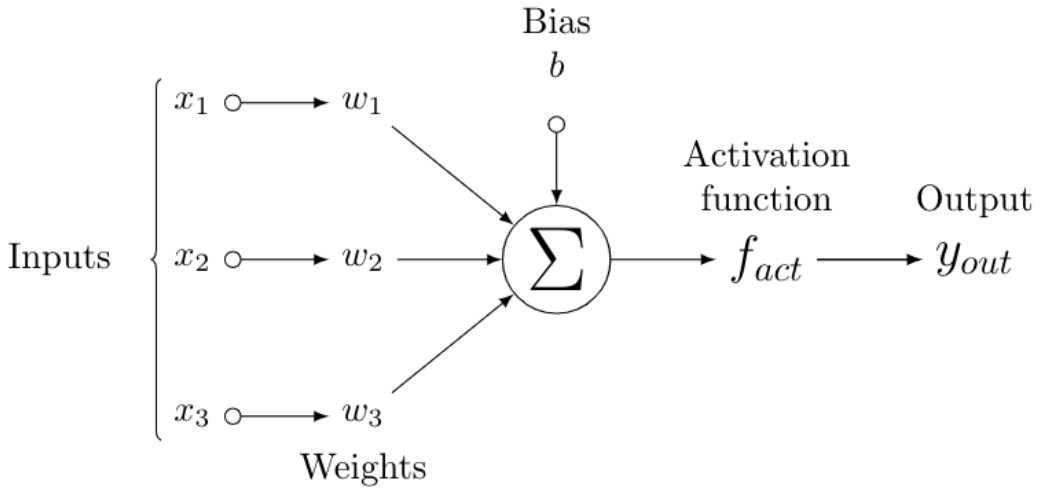

Рисунок 2.1 – Архітектура штучного нейрона

Поведінка кожного нейрона визначається його вагою. При подачі на вхід зображення на вхід у CNN, кожен з його шарів генерує кілька мап активації. Мапи активації знаходять відповідні особливості зображення. Кожен з нейронів бере в якості вхідного сигналу ділянку пікселів, множить їх значення кольору на його вагу, підсумовує їх і пропускає через функцію активації. Операція множення значень пікселів на ваги та їх підсумовування називається «згорткою» (convolution). CNN зазвичай складається з декількох згорткових шарів, але вона також може містити інші компоненти.

Шари згортки складаються з набору фільтрів, подібних до двовимірної матриці чисел. Фільтр «згортається» із вхідним зображенням для отримання вихідних даних. У кожному з шарів згортки фільтр проводиться («ковзає») по зображенню, щоб виконати операцію згортки. Основним порядком роботи операції згортки є матричне множення значень фільтра та пікселів зображення, а отримані значення підсумовуються На виході отримується «мапа ознак» («feature map»). На рисунку 2.2 зображено операцію згортки.

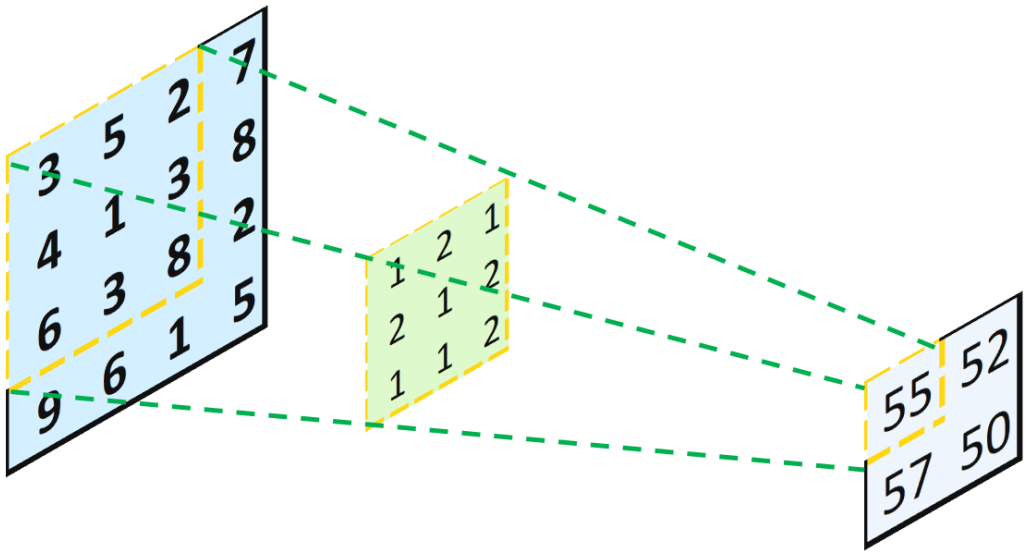

Рисунок 2.2 – Схематичне зображення операції згортки

Отже, згорткові мережі здійснюють своєрідний пошук. Фільтри ковзають зліва направо по великому зображенню, і починають все спочатку коли вони досягають кінця у рамках одного проходу (як це роблять друкарські машини). Це рухоме вікно здатне розпізнавати лише один об’єкт. Кожного разу, коли фільтри знаходять збіг, він відображається у просторі об’єктів, характерному для цього візуального елемента. У цьому просторі фіксується місце розташування кожного збігу для об’єкта, що розшукується. Згорткова нейронна мережа проводить багато пошуків за одним зображенням – горизонтальними лініями, діагональними лініями, стільки, скільки є візуальних елементів, які потрібно шукати.

Ключовою особливістю операції згортки є розподіл ваги: ядра спільно використовуються для всіх позицій зображення. Процес навчання моделі CNN щодо рівня згортки полягає у визначенні ядер, які найкраще працюють для даного завдання на основі заданого набору даних для навчання. Ядра (фільтри) – це єдині параметри, які автоматично вивчаються під час тренувального процесу на рівні згортки. З іншого боку, розмір фільтрів, кількість фільтрів, пединг та крок – це гіперпараметри, які потрібно встановити перед початком навчального процесу.

Як вже зазначено вище, в операції згортки бере участь фільтр. В свою чергу філтр має дві особливості: крок(stride) та педінг (padding). Крок – це кількість зсувів пікселів по вхідній матриці. Коли крок дорівнює 1, фільтр переміщується через 1 піксель за раз. Коли крок дорівнює 2, фільтр переміщується на 2 пікселі одночасно і т.д. Але іноді фільтр не підходить до вхідного зображення повністю. В такому випадку можна заповнити пусті пікселі зображення нулями (zero-padding), щоб він помістився. Або можна пропустити ту частину зображення, де фільтр не «дістає» (зазвичай крайні рядки зображення). Це називається дійсним заповненням (valid padding), яке зберігає лише дійсну частину зображення.

Перший шар згорткової нейронної мережі зазвичай виявляє основні ознаки, такі як горизонтальні, вертикальні та діагональні ребра. Вихід першого шару подається на вхід наступного шару, який виявляє більш складні ознаки, такі як кути та комбінації ребер. По мірі поглиблення в згорткову нейронну мережу шари починають виявляти ознаки вищого рівня, такі як об’єкти, обличчя тощо. Останній рівень згорткової нейронної мережі – це рівень класифікації, який приймає вихідні дані останнього згорткового шару як вхідні. На основі мапи активації останнього рівня згортки класифікаційний рівень видає набір оцінок вірогідності (значення від 0 до 1), які вказують, наскільки ймовірно, що зображення належить до того чи іншого «класу».

Згорткова нейронна мережа містить:

- згорткові шари;
- шари пулінгу;
- повно зв'язані шари.

Шар згортки є будівельним елементом згорткової нейронної мережі. Він відповідає за виконання основної частини обчислювального навантаження мережі. Шар згортки відіграє ключову роль у CNN. Він складається з набору математичних операцій, таких як згортка, яка є спеціалізованим типом лінійних операцій.

Шар пулінгу (pooling layer) допомагає зменшити просторовий розмір представлення, що зменшує необхідну кількість обчислень і ваг. Пулінг забезпечує певну незмінність переміщення, що означає, що об'єкт можна буде впізнати незалежно від того, де він відображається на кадрі. На рисунку 2.3 зображено згортковий шар та шар пулінгу.

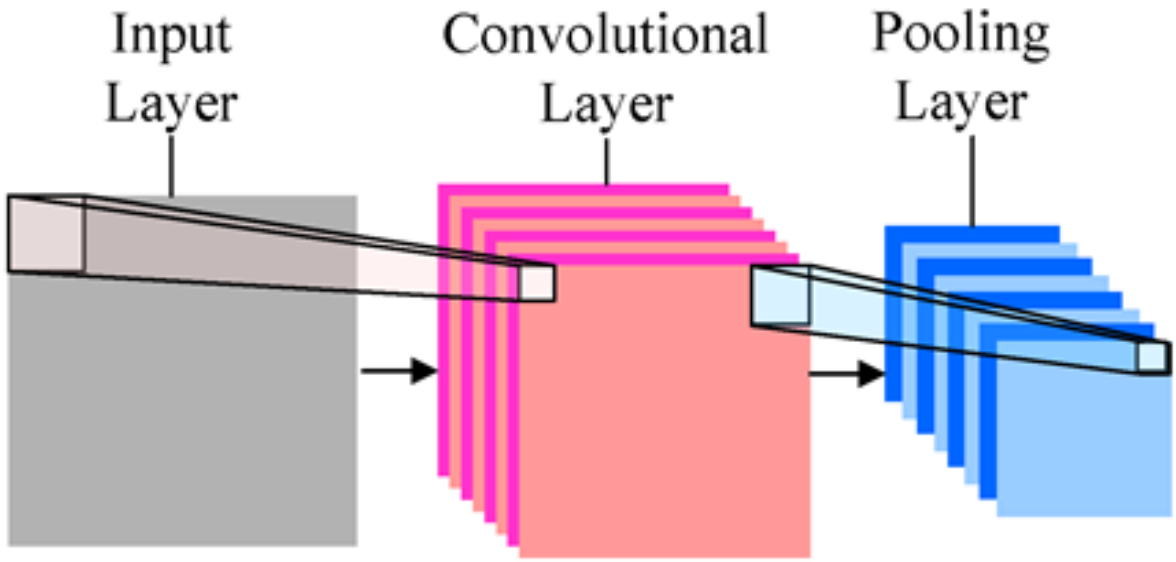

Рисунок 2.3 – Схематичне зображення згорткового шару та шару пулінгу

Нейрони повно зв'язаного шару (fully connected layer): мають повний зв’язок з усіма нейронами попереднього та наступного шару, точно так же як і у звичайних нейронних мережах прямого поширення. Його можна обчислити, зазвичай, шляхом множення матриці з подальшим урахуванням зміщення. Повно зв'язанні шари допомагають відобразити представлення між входом і виходом. На рисунку 2.4 зображено повно зв'язаний шар.

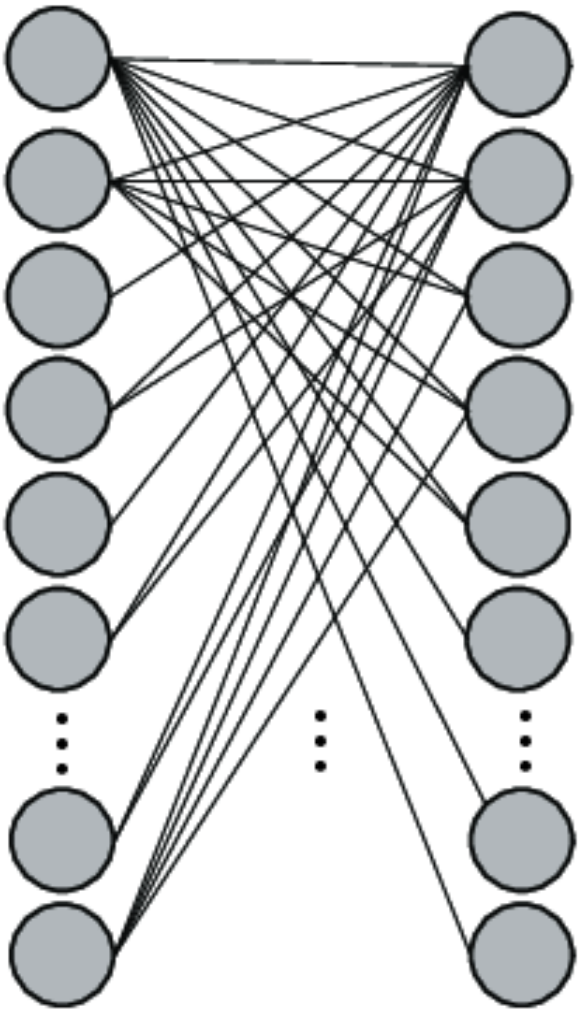

Рисунок 2.4 – Схематичне зображення повно зв'язаного шару

Оскільки згортка є лінійною операцією, а зображення не є лінійними, шари нелінійності часто розміщуються безпосередньо після шару згортки, щоб ввести нелінійність на мапу активації. Існує кілька видів нелінійних операцій, найпопулярніші з них – сигмоїда, гіперболічний тангенс, ReLU (The Rectified Linear Unit).

Сигмоїдна нелінійність має математичний вигляд:

$$f(x) = \frac{1}{1+e^{(-x)}}. \quad (2.1)$$

Ця функція приймає дійсне число і стискає його в діапазон від 0 до 1 [25]. Сигмоїда страждає від зникаючої проблеми зникаючого градієнта, тобто коли локальний градієнт стає дуже малим, а зворотне поширення призводить до загибелі градієнта.

На рисунку 2.6 зображено графік сигмоїдної функції.

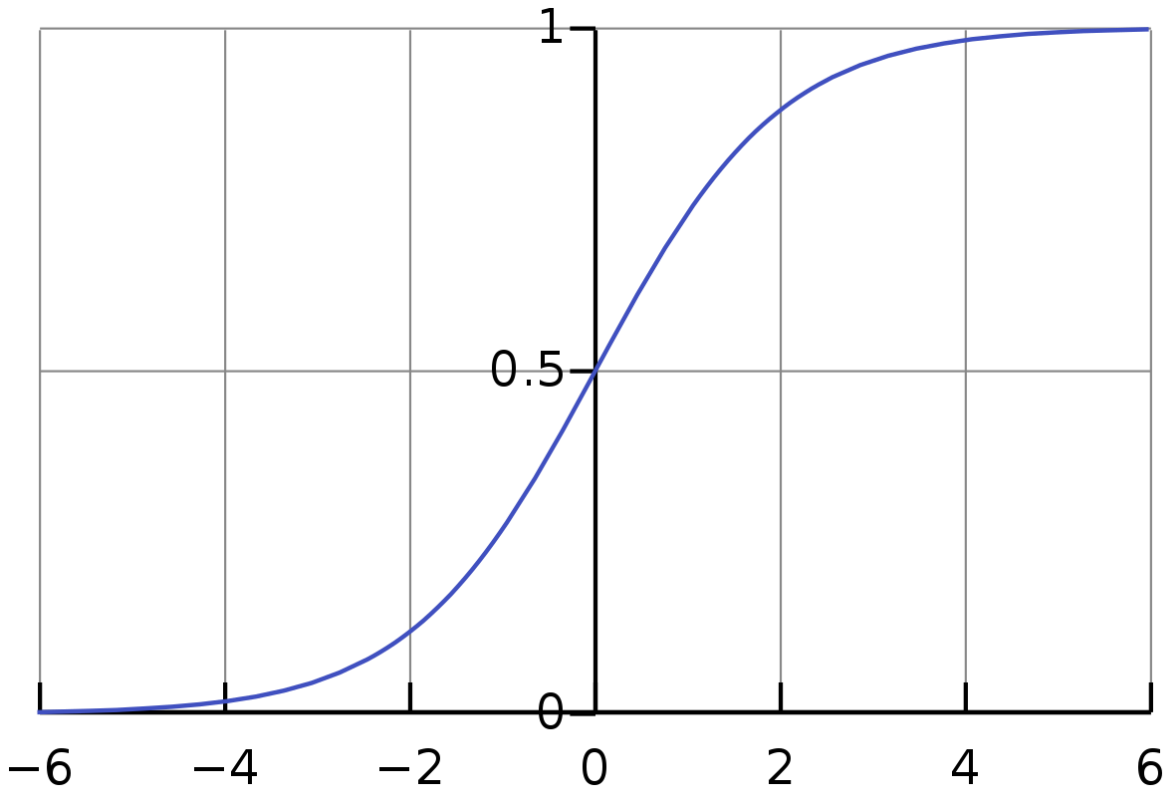

Рисунок 2.6 – Графік сигмоїдної функції

Гіперболічний тангенс стискає дійсне число до діапазону від -1 до 1. Історично, ця функція стала використовуватись частіше аніж сигмоїдна функція, оскільки гіперболічний тангенс давав кращі характеристики для багатошарових нейронних мереж ніж сигмоїдна функція. Але функція гіперболічний тангенс не вирішила проблему зникаючого градієнта, від якої страждає сигмоїдна функція [26].

На рисунку 2.7 зображено графік гіперболічного тангенса.

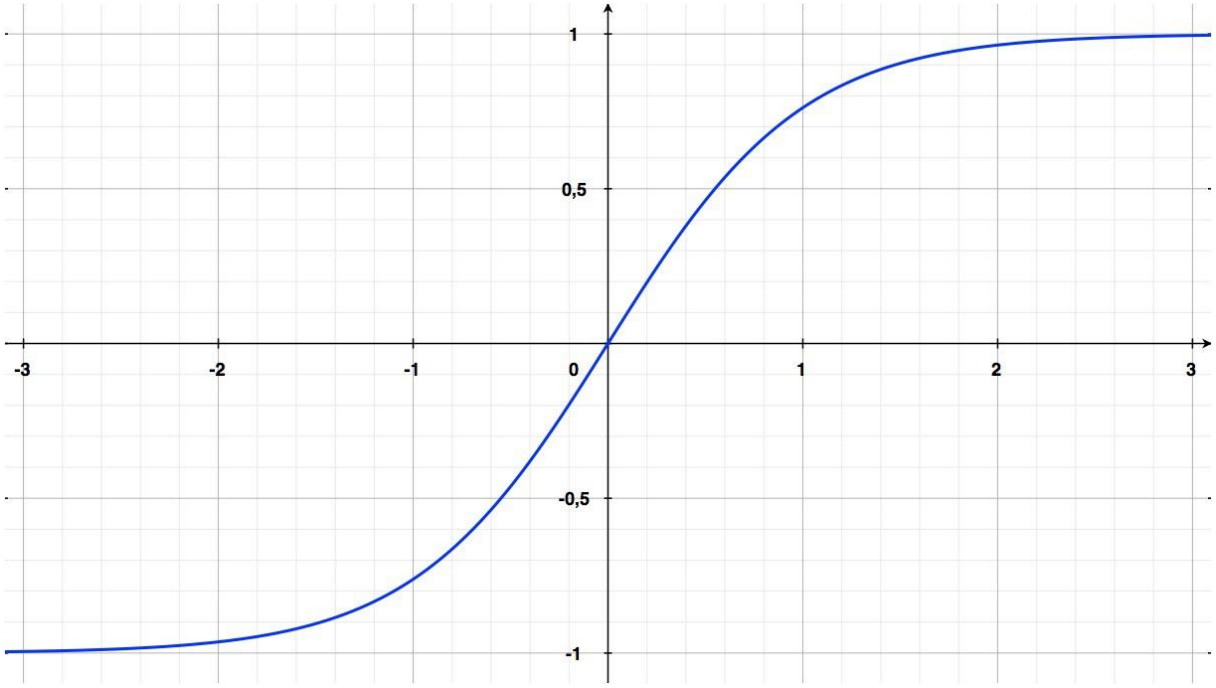

Рисунок 2.7 – Графік функції гіперболічний тангенс

ReLU обчислює наступну функцію:

$$f(x) = \max(0, x). \tag{2.2}$$

ReLU – це тип функції активації, яка є лінійною у позитивному вимірі, але нульовою у негативному вимірі. У порівнянні з сигмоїдою та гіперболічним тангенсом ReLU є більш надійною функцією [27]. Вона також прискорює конвергенцію в шість разів.

ReLU є лінійною для значень, більших за нуль, тобто вона має масу бажаних властивостей лінійної функції активації при навчанні нейронної мережі з використанням зворотного розповсюдження. Однак це нелінійна функція, оскільки негативні значення завжди виводяться як нуль. Функція ReLU найчастіше всього використовується в архітектурах згорткових нейронних мереж.

На рисунку 2.8 зображено графік ReLU.

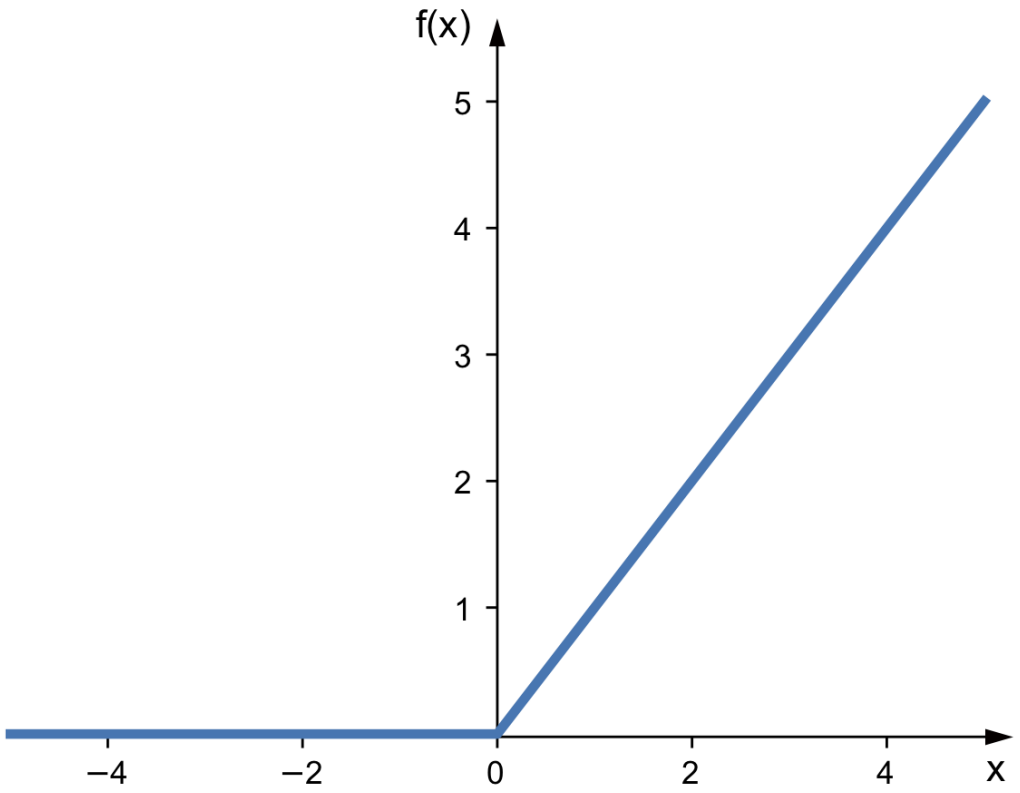

Рисунок 2.8 – Графік функції ReLU

2.3 Навчання згорткової нейронної мережі

Однією з найбільших труднощів у створенні згорткової нейронної мережі є регулювання ваг окремих нейронів для вилучення правильних рис із зображень. Процес регулювання цих ваг називається «тренуванням» нейронної мережі.

На початку згорткова нейронна мережа ініціалізується випадковими вагами. Під час навчання мережі надається великий набір зображень (датасет), з відповідними класами. CNN обробляє кожне зображення з її випадковими значеннями ваг, а потім порівнює вихід мережі з правильною міткою зображення. Якщо вихід мережі не відповідає мітці вона робить невелику корекцію ваг своїх нейронів, щоб наступного разу, коли вона побачить одне і те ж зображення, її результат буде трохи ближче до правильної відповіді. Виправлення здійснюються за допомогою методу

зворотного поширення помилки (backpropagation). По суті, зворотне поширення помилки оптимізує процес налаштування та полегшує мережі вибір рішення, які блоки налаштувати, замість випадкових виправлень [28]. Кожен цикл (прогін) усього навчального датасету називається «епохою». Під час тренувань згорткова нейронна мережа проходить кілька епох, регулюючи свої ваги. Після кожної епохи мережа стає дещо кращою при класифікації навчальних зображень. У процесі вдосконалення мережі коригування ваг стає все меншим і меншим. У якийсь момент мережа «сходиться», що означає, що вона по суті стає якомога кращою.

Після навчання CNN, для перевірки її точності, використовується тестовий набір даних. Тестовий набір даних – це набір маркованих зображень, які не були частиною навчального датасету. Кожне зображення проходить через мережу, і вихідні результати порівнюються з фактичною міткою зображення. По суті, тестовий набір даних оцінює, наскільки згорткова нейронна мережа добре класифікує зображення, яких раніше не бачила (тобто які не входили до тренувального датасету).

Якщо згорткова нейронна мережа отримує добрі результати при обробці навчальних даних, але погані при обробці тестових даних, це значить що мережа «перенавчилась» (overfitting). Зазвичай це трапляється, коли в навчальних даних недостатньо різноманітності або коли CNN навчається занадто довго (проходить через занадто багато епох) з тренувальним датасетом. Успіх згорткових нейронних мереж значною мірою зумовлений наявністю величезних наборів даних зображень, розроблених за останнє десятиліття. Також, у багатьох випадках можна використовувати попередньо навчену модель і доналаштувати (finetune) її для іншої більш спеціалізованої задачі.

2.4 Convolutional Sequence Embedding Recommendation Model

ConvolutionAl Sequence Embedding Recommendation Model (Caser) – це модель послідовних рекомендацій, що використовує згорткову нейронну мережу[29]. Ця модель, створена Джіаксі Тангом та Ке Вангом, призначена для рекомендацій декількох об’єктів, що з найбільшою ймовірністю сподобаються користувачеві. Рекомендації засновані на попередніх взаємодіях користувача з об’єктами.

Ідея полягає в тому, щоб «вбудувати» послідовність останніх елементів (елементів, з якими взаємодіяв користувач) у «зображення» у часі та латентних просторах та шукати послідовні шаблони як локальні особливості цього «зображення» за допомогою конволюційних фільтрів. Цей підхід забезпечує уніфіковану та гнучку мережеву структуру для захоплення як загальних уподобань, так і послідовних моделей. Однак, на відміну від розпізнавання зображень, це «зображення» не надається одразу на вході, його треба вивчити одночасно з усіма фільтрами.

Caser складається з трьох компонентів: компонент пошуку ембедінгів (Embedding Look-up), згорткові шари та повно зв'язані шари. На рисунку 2.9 зображено архітектуру мережі Caser.

Для тренування CNN, для кожного користувача u, витягується по L послідовних елементів (об’єктів), у якості вхідних даних, та їхні наступні T елементів, у якості цільових об’єктів, з послідовності користувача S_u. Це робиться шляхом ковзання вікна розміром L+T над послідовністю користувача, і кожне вікно генерує тренувальний екземпляр для u, позначеного триплетом (u, попередні L елементів, наступні T елементів).

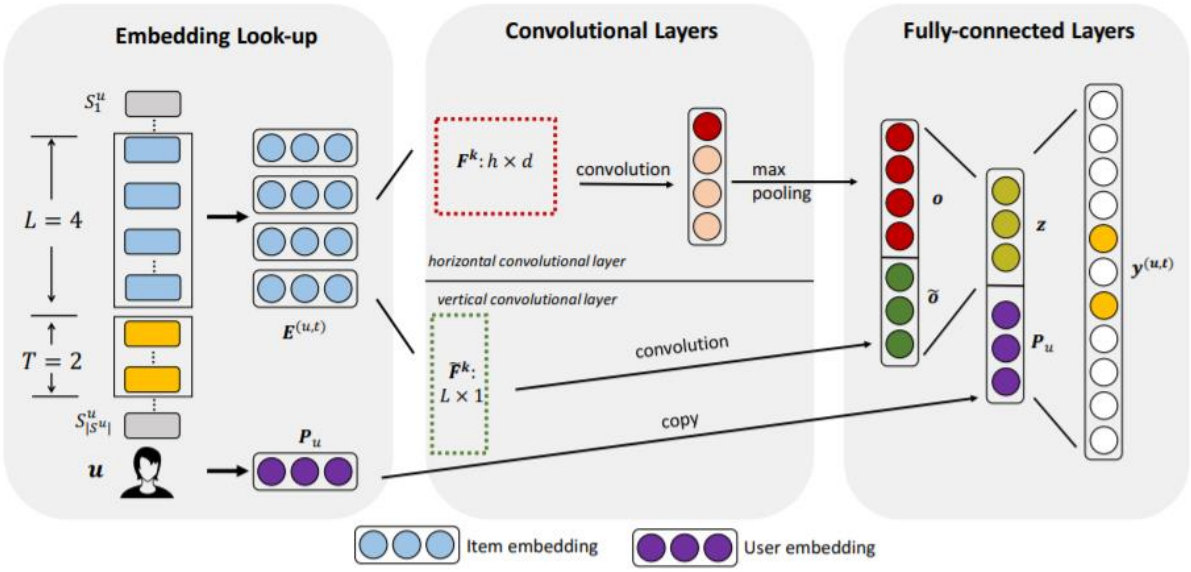

Рисунок 2.9 – Архітектура мережі Caser

Caser фіксує особливості послідовності у латентному просторі, подаючи ембедінги попередніх об’єктів L у нейронну мережу. Операція пошуку ембедінгу (embedding look-up operation) отримує ембедінги попередніх елементів L та складає їх разом, що призводить до отримання матриці $E^{(u, t)}$ для користувача u на кроці часу t. Поряд з ембедінгами об’єктів також є ембедінги P_u для користувача u, що представляє особливості користувача у латентному просторі.

Позичаючи ідею використання CNN у класифікації тексту [30], автори розглядають матрицю E (розміром L×d) як «зображення» попередніх L об’єктів у латентному просторі та розглядає послідовні патерни як локальні особливості цього «зображення». Цей підхід дає можливість використовувати згорткові фільтри для пошуку послідовних шаблонів. Два «горизонтальні фільтри» фіксують два послідовні шаблони на рівні об'єднання. Ці фільтри, представлені у вигляді матриць (розміром h×d), мають висоту h = 2 і повну ширину, рівну d. Вони підбирають сигнали для послідовних шаблонів, ковзаючи по рядках E. Аналогічним чином, «вертикальний фільтр» є матрицею L×1 і ковзає по стовпцях E. На відміну від процесу розпізнавання зображення, «зображення» E не дається одразу,

бо ембедінги для всіх об’єктів потрібно вивчати одночасно з усіма фільтрами.

Виходи горизонтальних фільтрів подаються на вхід до шарів пулінгу (max pooling), щоб витягти максимальне значення з усіх значень, створених цим конкретним фільтром. Горизонтальні фільтри взаємодіють з кожним послідовним елементом h через їх ембедінг E. Як ембедінг, так і фільтри вивчаються для мінімізації цільової функції, яка кодує помилку прогнозу цільових елементів. Отже, горизонтальні фільтри фіксують шаблони на рівні об'єднання з різними розмірами об'єднань. Водночас вертикальні фільтри фіксують точкові послідовні шаблони.

Далі, виходи двох згорткових шарів об'єднуються і подаються у повністю пов'язаний шар нейронної мережі, щоб отримати більш високі та абстрактні ознаки. Щоб зафіксувати загальні вподобання користувача, генерується ембедінг користувача P_u і об’єднуються два двовимірні вектори z та P_u разом. Потім ці вектори проєктуються на вихідний шар та отримується значення $y^{(u, t)}$. Це значення у вихідному рівні асоціюється з ймовірністю того, наскільки ймовірно, що користувач u буде взаємодіяти з елементом і на кроці часу t. Ембедінг z намагається зафіксувати короткострокові послідовні шаблони, тоді як ембедінг користувача P_u враховує довгострокові загальні уподобання користувача.

Для тренування мережі було використано варіант стохастичного градієнтного спуску (Stochastic Gradient Descent, SGD), який називається адаптивна оцінка моменту (Adaptive Moment Estimation, Adam) для швидшої конвергенції. Для контролю складності моделі та уникнення перенавчання були використані два типи методів регуляризації: L2 норма застосовувалась для всіх параметрів моделі, а на повністю з'єднаних шарах застосовувалась методика dropout із коефіцієнтом викидання 50%.

Для тестування моделі було проведено експеримент на чотирьох датасетів: MovieLens, Gowalla, Foursquare, Tmall. Перші 70% даних (дій у послідовності кожного користувача) було використано як навчальний набір,

наступні 10% даних було використано для формування валідаційного набору, щоб знайти оптимальні налаштування гіперпараметрів для всіх моделей. Решта 20% даних усіх датасетів були використані у якості тестового набору для оцінки ефективності моделі.

Експерименти на вищезгаданих датасетах показали, що модель Caser перевищив показники традиційних (ланцюги Маркова, факторизація матриць) та декількох сучасних моделей (Fossil, GRU4Rec) для послідовних рекомендацій. Результати експериментів зображені на рисунку 2.10.

Dataset	Metric	POP	BPR	FMC	FPMC	Fossil	GRU4Rec	Caser	Improv.
MovieLens	Prec@1	0.1280	0.1478	0.1748	0.2022	0.2306	**0.2515**	0.2502	-0.5%
	Prec@5	0.1113	0.1288	0.1505	0.1659	0.2000	0.2146	**0.2175**	1.4%
	Prec@10	0.1011	0.1193	0.1317	0.1460	0.1806	0.1916	**0.1991**	4.0%
	Recall@1	0.0050	0.0070	0.0104	0.0118	0.0144	**0.0153**	0.0148	-3.3%
	Recall@5	0.0213	0.0312	0.0432	0.0468	0.0602	0.0629	**0.0632**	0.5%
	Recall@10	0.0375	0.0560	0.0722	0.0777	0.1061	0.1093	**0.1121**	2.6%
	MAP	0.0687	0.0913	0.0949	0.1053	0.1354	0.1440	**0.1507**	4.7%
Gowalla	Prec@1	0.0517	0.1640	0.1532	0.1555	0.1736	0.1050	**0.1961**	13.0%
	Prec@5	0.0362	0.0983	0.0876	0.0936	0.1045	0.0721	**0.1129**	8.0%
	Prec@10	0.0281	0.0726	0.0657	0.0698	0.0782	0.0571	**0.0833**	6.5%
	Recall@1	0.0064	0.0250	0.0234	0.0256	0.0277	0.0155	**0.0310**	11.9%
	Recall@5	0.0257	0.0743	0.0648	0.0722	0.0793	0.0529	**0.0845**	6.6%
	Recall@10	0.0402	0.1077	0.0950	0.1059	0.1166	0.0826	**0.1223**	4.9%
	MAP	0.0229	0.0767	0.0711	0.0764	0.0848	0.0580	**0.0928**	9.4%
Foursquare	Prec@1	0.1090	0.1233	0.0875	0.1081	0.1191	0.1018	**0.1351**	13.4%
	Prec@5	0.0477	0.0543	0.0445	0.0555	0.0580	0.0475	**0.0619**	6.7%
	Prec@10	0.0304	0.0348	0.0309	0.0385	0.0399	0.0331	**0.0425**	6.5%
	Recall@1	0.0376	0.0445	0.0305	0.0440	0.0497	0.0369	**0.0565**	13.7%
	Recall@5	0.0800	0.0888	0.0689	0.0959	0.0948	0.0770	**0.1035**	7.9%
	Recall@10	0.0954	0.1061	0.0911	0.1200	0.1187	0.1011	**0.1291**	7.6%
	MAP	0.0636	0.0719	0.0571	0.0782	0.0823	0.0643	**0.0909**	10.4%
Tmall	Prec@1	0.0010	0.0111	0.0197	0.0210	0.0280	0.0139	**0.0312**	11.4%
	Prec@5	0.0009	0.0081	0.0114	0.0120	0.0149	0.0090	**0.0179**	20.1%
	Prec@10	0.0007	0.0063	0.0084	0.0090	0.0104	0.0070	**0.0132**	26.9%
	Recall@1	0.0004	0.0046	0.0079	0.0082	0.0117	0.0056	**0.0130**	11.1%
	Recall@5	0.0019	0.0169	0.0226	0.0245	0.0306	0.0180	**0.0366**	19.6%
	Recall@10	0.0026	0.0260	0.0333	0.0364	0.0425	0.0278	**0.0534**	25.6%
	MAP	0.0030	0.0145	0.0197	0.0212	0.0256	0.0164	**0.0310**	21.1%

Рисунок 2.10 – Результати експерементів для моделі Caser

3 МОДЕЛЬ 2D CONVOLUTIONAL NEURAL NETWORKS FOR SEQUENTIAL RECOMMENDATION

3.1 Загальна характеристика моделі 2D Convolutional Neural Networks for Sequential Recommendation

2D Convolutional Neural Networks for Sequential Recommendation (CosRec) – це модель на основі 2D згорткової нейронної мережі, розроблена А. Яном, В-Ч. Кангом, М. Ваном, Дж. МакОулем [31]. Ця модель кодує послідовність елементів у тристоронній тензор; вивчає локальні особливості за допомогою двовимірних згорткових фільтрів; і агрегує взаємодії високого порядку у зворотному напрямку. CosRec є першим підходом на базі двовимірної згорткової мережі, що застосовується для задач рекомендації наступного товару (next item recommendation).

Більшість існуючих моделей для задач послідовних рекомендацій працюють безпосередньо з упорядкованим рядом зображення об’єктів (item representation), і тому обмежуються односпрямованою ланцюговою структурою послідовностей дій. Це дає перевагу в тому, що ці алгоритми здатні зберігати локально зосереджену динаміку. У ситуаціях коли у послідовності об’єктів потрапляють «шумні» об’єкти (також відомі як «skip» behavior), локальність ланцюгової структури може бути порушена. Наприклад у моделі Caser є недолік, який полягає у тому що він може погано працювати при наявності шумних або неактуальних взаємодій. Такі випадки можуть бути подолані за допомогою попарного кодування (pairwise encoding), що реалізовано у підході CosRec.

Модель CosRec забезпечує взаємодію між несуміжними елементами, завдяки простому, але ефективному модулю парного кодування. Завдяки цьому модулю автори показали, що стандартні двовимірні згорткові ядра можуть застосовуватися для вирішення задач послідовних рекомендацій. Ця

модель також дозволяє розширювати каркас двовимірної згорткової нейронної мережі для її адаптації до різних завдань.

Ще одним недоліком моделі Caser є використання вертикальних фільтрів. Цей спосіб націлений на отримання зваженої суми всіх попередніх елементів, тоді як воно виконує лише підсумовування по кожному виміру, і воно немає взаємодій по каналах. Ця зважена сума також призводить до неглибокої мережевої структури, яка підходить лише для одного шару, що призводить до проблем при моделюванні довгих залежностей або великомасштабних потоків даних, де потрібна більш глибока архітектура. Метод CosRec із двовимірними ядрами забезпечує взаємодію каналів між векторами, а також забезпечує гнучкість адаптації мережі до маленьких або глибоких структур для різних завдань, шляхом застосовування операції педінгу або шляхом змінення розмір ядра.

CosRec складається з трьох модулів: шару пошуку ембедінгів (the embedding look-up layer), модуля парного кодування (the pairwise encoding module) та двовимірного модуля згортки (2D convolution module). Архітектура моделі CosRec представлена на рисунку 3.1.

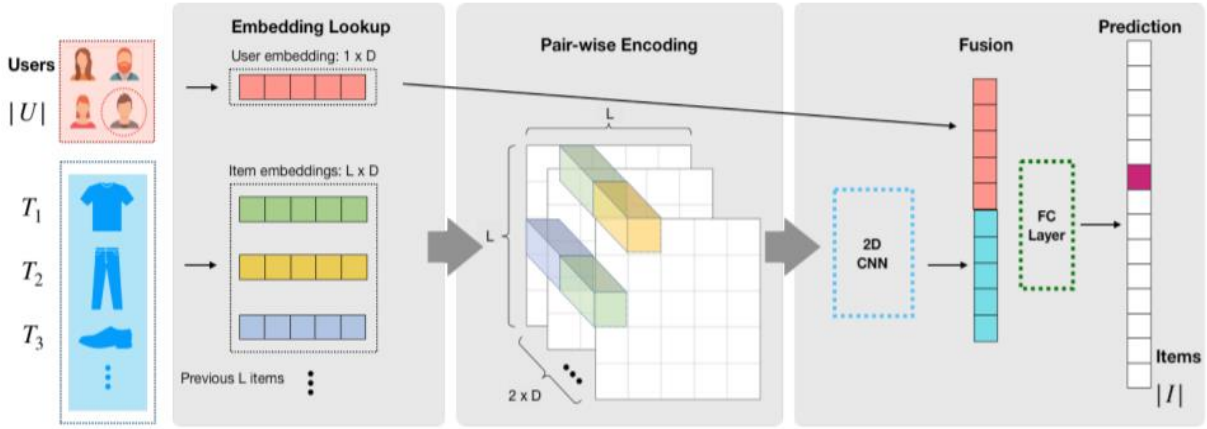

Рисунок 3.1 – Архітектура моделі CosRec

Автори моделі формулюємо задачу послідовних рекомендацій наступним чином. Припустимо, є набір користувачів U і набір

об’єктів (товарів) I. Для кожного користувача u∈U, враховуючи послідовність об’єктів, з якими була взаємодія раніше $S^u=(S^u_1, ..., S^u_{|Su|})$, $S^u \in$ I, ми прагнемо передбачити наступний вибір (об’єкт) відповідно до уподобань користувача.

У шарі пошуку ембедінгів об’єкти та користувачі «вбудовуються» у дві матриці $E_I \in R^{|I| \times d}$ та $E_U \in R^{|U| \times d}$, де d –латентна розмірність, e_i та e_u позначають i-й та u-й рядки в матрицях E_I, E_U відповідно. Потім для користувача u на кроці часу t отримується вхідна матриця ебедінгів $E^L_{(u, t)} \in R^{L \times d}$, шляхом пошуку попередніх L елементів $(S^u_{t-L}, ..., S^u_{t-1})$ у матриці ембедінгів E_I.

Після шару пошуку ембедінгів автори методу пропонують застосовувати кодування (encoding), щоб забезпечити гнучкі попарні взаємодії між елементами. Безпосередньо у шарі парного кодування (Pairwise Encoding) створюється тривимірний тензор $T^L_{(u, t)} \in R^{L \times L \times 2d}$ поверх вхідних ембедінгів $E^L_{(u, t)}$, де (i, j)-й вектор є конкатенованим ембедінгом пари об’єктів (i, j): $[e_i; e_j]$, i, j ∈ $(S^u_{t-L}, ..., S^u_{t-1})$. На відміну від попередніх підходів [29, 32], коли згорткові фільтри застосовуються безпосередньо на вхідній матриці $E^L_{(u, t)}$, автори моделі застосовують згорткові шари на цьому результуючому тензорі $T^L_{(u, t)}$. Це робиться для того, щоб можна було захопити складні патерни. Цікаво, що закодований тензор має таку ж саму форму як і «карта особливостей зображення» у стандартних моделях згорткових нейронних мережах для задач комп’ютерного зору.

Для того, щоб захопити високорівневі (головні) послідовні патерни, вищезазначена «карта особливостей зображення», що представлена тензором $T^L_{(u, t)}$, подається на вхід до двовимірної згорткової нейронної мережі. Архітектура данної мережі представлена у таблиці 3.1. D – розмірність вхідних ембедінгів об’єктів. D_1, D_2, D_3 – приховані розмірності кожного шару.

Таблиця 3.1 – Архітектура згоркової мережі

Шар	Розмір вихідних даних	Розмір ядра
вхідний	D × 5 × 5	-
згортка1_1	D_1 × 5 × 5	1 × 1
згортка1_2	D_1 × 3 × 3	3 × 3
згортка2_1	D_2 × 3 × 3	1 × 1
згортка2_2	D_2 × 1 × 1	3 × 3
повно зв’язний1	D_3 × 1 × 1	dropout
повно зв’язний2	|I| × 1 × 1	sigmoid

У цьому модулі кожен блок згортки складається з двох згорткових шарів: перший шар використовує ядра 1 × 1 для поліпшення якості зображення ознак об’єктів. Другий шар із розміром ядра 3 × 3 об'єднує послідовні ознаки та витягує більш складні відносини по мірі того, як мережа стає глибшою.

Кожен згортковий шар супроводжується батч нормалізацією та активацією ReLU. Після цих двох згорткових блоків застосовується повністю зв’язаний шар із дропаутом (викиданням). Таким чином отримується кінцевий послідовний вектор ознак $v_{(u, t)} \in R^d$.

Для того, щоб захопити глобальні уподобання користувачів, послідовний вектор $v_{(u, t)}$ об'єднується з ембедінгом користувача e_u. Далі все це проектується на вихідний шар з |I| вузлів, і, у кінці, застосовується сигмоїдна функцію для отримання кінцевих оцінок ймовірності $\sigma (y^{(u, t)}) \in R^{|I|}$.

Для тренування моделі використовується функція втрати бінарної крос ентропії (binary cross-entropy loss) у якості цільової функції. Мережа оптимізується за допомогою оптимізатора Adam, що є варіантом стохастичного градієнтного спуску (SGD) з адаптивною оцінкою моменту. У кожній ітерації випадково відбирається N негативних прикладів (j) для кожного цільового об’єкта S^u_t.

Модель CosRec була випробувана на датасетах MovieLens (ML-1M) та Gowall. Ці набори данних містять 900 тисяч та 500 тисяч дій (покупка товару, перегляд товару і т.д.) відповідно. Перші 80% дій у послідовності кожного користувача були застосовані для навчання та перевірки моделі, а решта 20% дій була застосована для тестування та оцінки ефективності моделі.

Експерименти на вищезазначених датасетах показали, що модель CosRec перевищила показники традиційних моделей, а також показники моделі Caser. Результати експериментів зображені на рисунку 3.2.

Dataset	Metric	PopRec	BPR	FMC	FPMC	GRU4Rec	Caser	CosRec-base	CosRec	Improvement
ML − 1M	MAP	0.0687	0.0913	0.0949	0.1053	0.1440	0.1507	0.1743	**0.1883**	+25.0%
	Prec@1	0.1280	0.1478	0.1748	0.2022	0.2515	0.2502	0.2892	**0.3308**	+31.5%
	Prec@5	0.1113	0.1288	0.1505	0.1659	0.2146	0.2175	0.2521	**0.2831**	+30.2%
	Prec@10	0.1011	0.1193	0.1317	0.1460	0.1916	0.1991	0.2256	**0.2493**	+25.2%
	Recall@1	0.0050	0.0070	0.0104	0.0118	0.0153	0.0148	0.0186	**0.0202**	+32.0%
	Recall@5	0.0213	0.0312	0.0432	0.0468	0.0629	0.0632	0.0771	**0.0843**	+33.4%
	Recall@10	0.0375	0.0560	0.0722	0.0777	0.1093	0.1121	0.1331	**0.1438**	+28.3%
Gowalla	MAP	0.0229	0.0767	0.0711	0.0764	0.0580	0.0928	0.0821	**0.0980**	+05.6%
	Prec@1	0.0517	0.1640	0.1532	0.1555	0.1050	0.1961	0.1712	**0.2135**	+08.9%
	Prec@5	0.0362	0.0983	0.0876	0.0936	0.0721	0.1129	0.1012	**0.1190**	+05.4%
	Prec@10	0.0281	0.0726	0.0657	0.0698	0.0782	0.0571	0.0762	**0.0884**	+13.0%
	Recall@1	0.0064	0.0250	0.0234	0.0256	0.0155	0.0310	0.0265	**0.0337**	+08.7%
	Recall@5	0.0257	0.0743	0.0648	0.0722	0.0529	0.0845	0.0752	**0.0890**	+05.3%
	Recall@10	0.0402	0.1077	0.0950	0.1059	0.0826	0.1223	0.1107	**0.1305**	+06.7%

Рисунок 3.2 – Результати експерементів для моделі CosRec

Як видно з рисунку 3.2 показники ефективності на датасеті ML-1M є особливо значним (покращення на 25% враховуючи метрику MAP), імовірно, завдяки тому, що ML-1M є відносно щільним набором даних із багатими послідовними сигналами.

Для того, щоб оцінити ефективність модуля 2D CNN, автори моделі CosRec створили базову версію CosRec (CosRec-base), яка використовує багатошаровий персептрон замість 2D згортки у модулі парного кодування.

Треба зазначити, що на наборі даних ML-1M, навіть базова модель CosRec (CosRec-base) перевершує показники існуючих найсучасніших

методів, що підтверджує ефективність парного кодування для отримання більш складних шаблонів.

3.2 Удосконалення архітектури згорткової мережі для моделі 2D Convolutional Neural Networks for Sequential Recommendation

Для того, щоб якісно навчити згорткову нейронну мережу для певної задачі, потрібно багато різноманітних даних. Для їх обробки, зазвичай, потрібно більше згорткових шарів. Також відомо, що для того, щоб захопити більш маленькі (складні) деталі вхідних даних, до архітектури згоркових нейронних мереж теж додають більше шарів згортки (наскільки більше, залежить від складності вхідних даних). Але за це доводиться «платити» обчислювальною потужністю.

Можна припустити, що якщо до архітектури згорткової нейронної мережі у моделі CosRec додати нові згорткові шари, то на виході можна отримати точніші рекомендації, за умов достатньо великого числа вхідних даних. Тобто, архітектура згорткової нейронної мережі буде трохи глибшою. Як відомо, більш глибокі мережі охоплюють природну «ієрархію», яка присутня всюди у реальному світі. . Таким чином, перевага багатошарових мереж полягає в тому, що вони можуть вивчати особливості на різних рівнях абстракції.

Також, автори архітектури CosRec запевняють [31], що пропонована ними модель є гнучкою та може бути змінена під певну предметну область або для навчання зі специфічними даними.

Задля підвищення ефективності рекомендацій, для адаптації моделі CosRec для галузі електронної комерції, а також для перевірки гнучкості моделі, архітектуру згорткової нейронної мережі було змінено. Удосконалена архітектура представлена у таблиці 3.2.

Таблиця 3.2 – Удосконалена архітектура згорткової нейронної мережі

Шар	Розмір вихідних даних	Розмір ядра
вхідний	$D \times 5 \times 5$	-
згортка1_1	$D_1 \times 5 \times 5$	1×1
згортка1_2	$D_1 \times 3 \times 3$	3×3
згортка2_1	$D_2 \times 3 \times 3$	1×1
згортка2_2	$D_2 \times 3 \times 3$	3×3
згортка3_1	$D_3 \times 3 \times 3$	1×1
згортка3_2	$D_3 \times 1 \times 1$	3×3
повно зв’язний1	$D_4 \times 1 \times 1$	dropout
повно зв’язний2	$\|I\| \times 1 \times 1$	sigmoid

Як видно з таблиці, до мережі було додано один додатковий згортковий блок. Після кожного блоку згортки додано шари пулінгу (average pooling), що враховує усі значення у матриці. Шари згортки ініціалізовані за допомогою ініціалізатора Glorot (популярна схема ініцілізації). Як і в оригінальній моделі, після кожного шару згортки йде батч нормалізація та функція активації ReLU. Функція втрати також не змінилась – це бінарна крос ентропія. У якості оптимізатора було використано оптимізатор Adam, який, доречі, було зроблено спеціально для тренування глибоких нейронних мереж.

В іншому, схема роботи моделі CosRec залишилась незмінною.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Постановка практичної експериментальної задачі

В даній кваліфікаційній роботі ставиться наступна практична задача: модифікувати модель CosRec для перевірки її гнучкості та можливості налаштування під специфічну задачу або предметну область. Предметною областю для поставленої задачі є електронна комерція, тому що ця сфера є однією з провідних сфер, де застосовуються системи рекомендацій.

Вибір датасетів є одним з головних моментів у побудові систем послідовних рекомендацій, адже взаємодія між об’єктами та користувачами може мати різні характеристики. Ці характеристики обумовлені неоднозначністю та складністю поведінки користувачів, характеристиками різних товарів та контекстом магазину чи системи В залежності від цих характеристик при вирішенні задачі надання послідовних рекомендацій можуть виникати певні види труднощів. Наприклад, коли послідовність взаємодії користувача з об'єктами досить довга, то виникає необхідність вивчати послідовні залежності вищого порядку або довготривалі послідовні залежності. З першою проблемою справляються ланцюги Маркова, а з другою – LSTM системи. У випадках послідовностей з гнучким порядком виникає потрібність вивчення колективних послідовних залежностей за умови гнучкого порядку. З цим завданням справляються CNN. А коли у послідовність взаємодій користувача з об’єктом потрапляють нерелевантні або «шумні» дані, то треба вивчати послідовні залежності уважно та розбірливо. Для цього можна використовувати attention models або мережі з пам'яттю (memory networks). Ще можуть виникати послідовності взаємодій з ієрархічними структурами. Інакше кажучи, демографічні показники користувачів можуть певною мірою визначати уподобання цих користувачів і надалі впливати на їхні взаємодії з об’єктами. У таких

випадках треба вивчати ієрархічні залежності. З такою задачею впораються ієрархічні RNN або RNN, збагачені функціями (Feature-enriched RNN).

Для проведення експерименту було обрано дві вибірки даних: «User Behavior Data from Taobao for Recommendation» [33] та «Amazon (Electronics)» [34]. Дані вибірки було взято по причині того що вони повною мірою відображають сферу електронної комерції, бо Taobao та Amazon є одними з найбільших ретейлерів для китайського та американського ринку відповідно. До того ж вони мають велику базу клієнтів та широкий вибір непродовольчих товарів з найрізноманітніших категорій. Також ці датасети містять усю необхідну інформацію для експерименту (включаючи різні види взаємодій користувачів з товарами та час кожної взаємодії для встановлення послідовностей).

Обидві вибірки подаються на вхід до оригінальної моделі CosRec та до її модифікації. Отримані на виході рекомендації оцінюються за допомогою метрики MAP (Mean Average Precision).

4.2 Обгрунтування та вибір програмно-інструментальних засобів для експериментальних досліджень

Для реалізації експерименту була застосована мова програмування Python і його бібліотека PyTorch, яка є однією з найпопулярніших фреймворків, що застосовуються у моделях глибинного навчання [35].

Python – це мова програмування високого рівня загального призначення, яка широко використовується в науці про дані та для створення алгоритмів глибокого навчання. Він простий для розуміння і його можна швидко впровадити. Python стає найпопулярнішою мовою програмування у світі. Завдяки простоті, універсальності та простоті обслуговування Python є мовою програмування для багатьох відомих брендів, серед яких Facebook, Google, Quora, Amazon, Netflix. Він

використовується в найбільш захоплюючих та інноваційних технологіях, включаючи машинне навчання, штучний інтелект та робототехніку.

Слід почати з того, що Python подібний до англійської мови, якою люди користуються у повсякденному житті. Простота синтаксису дозволяє мати справу зі складними системами та гарантувати, що всі елементи мають чіткий взаємозв'язок між собою. Завдяки цьому більшість програмістів-початківців можуть вивчати мову і швидше приєднуватися до спільноти програмістів. Ця мова програмування дозволяє представляти дані у форматі, який можна легко зрозуміти, за допомогою різних графіків та діаграм. Вони є ефективним способом візуального представлення та розуміння даних. Компанії веб-розробки використовують бібліотеки Python, які дозволяють візуалізувати дані та створювати чіткі та зрозумілі звіти.

Однією з основних причин, чому Python є найкращою мовою для глибинного навчання – це доступ до багатьох бібліотек. Наявність доступу до різних бібліотек дозволяє розробникам виконувати складні завдання без необхідності переписувати багато рядків коду. Оскільки машинне навчання значною мірою залежить від математичної оптимізації, ймовірності та статистики, бібліотеки Python допомагають науковцям легко виконувати різні дослідження. Мова Python дуже гнучка. Розробники можуть використовувати цю мову разом з іншою мовою програмування для досягнення своїх цілей. Їм не потрібно перекомпілювати код. Будь-які зміни можна вносити миттєво. Завдяки гнучкості Python, ймовірність виникнення помилок мінімальна. Код, що написано на мові Python легко читати, тому будь-який розробник Python може легко його реалізувати, скопіювати або надати спільний доступ, коли потрібна зміна коду. У порівнянні з іншими мовами програмування, наприклад такими як Java, Python має менш обмежений підхід до програмування. Він має декілька парадигм і може підтримувати безліч стилів програмування, включаючи процедурний, об'єктно-орієнтований та функціональний. Це робить Python чудовою

мовою для стартапів, оскільки іноді програмістам може знадобитися змінити підхід у будь-який момент.

Python не залежить від платформи. Незалежність від платформи відноситься до мови програмування або фреймворку, що дозволяє розробникам реалізовувати речі на одній машині та використовувати їх на іншій машині без будь-яких (або лише з мінімальними) змін. Одним із ключових факторів популярності Python є те, що це незалежна від платформи мова. Python підтримується багатьма платформами, включаючи Linux, Windows та macOS. Код Python може бути використаний для створення самостійних програм для більшості поширених операційних систем, а це означає, що програмне забезпечення Python можна легко розповсюджувати та використовувати в цих операційних системах без інтерпретатора Python.

Завдяки своїй потужній інтеграції з такими мовами програмування як C, C++ та Java, Python може стати в нагоді для скриптових програм. Розроблений для вбудовування з самого початку, він може бути дуже корисним для налаштування великих програм та створення розширень для них. Python також використовується для автоматизації тестів. Багато фахівців з автоматизації якості (QA automation specialists) вибирають Python за його простоту у навчанні. Він також чудовий для тих, хто має більш обмежений технічний досвід. Він має сильну спільноту, чіткий синтаксис та читабельність. Python навіть має простий у використанні фреймворк модульного тестування (unit-testing framework).

До речі, Python має групу прихильників, і варто зазначити, що якщо виникають проблеми, є хтось, хто зможе запропонувати руку допомоги. Python – це мова з відкритим кодом, це вказує на те, що існує широке коло ресурсів, якими можуть користуватися як початківці, так і професіонали. Документація на Python доступна в Інтернеті, і багато питань, що часто трапляються, обговорюються на форумах та спільнотах.

PyTorch – це оптимізована тензорна відкрита бібліотека для глибокого навчання. Ця бібліотека популярна завдяки легко зрозумілому API та імперативному підходу. PyTorch може бути використаний для цілого ряду завдань, але він «спеціалізується» на задачах глибинного навчання, таких як комп'ютерний зір та обробка природної мови.

PyTorch містить пакет функцій високого рівня, таких як обчислення на багатовимірних масивах даних, також відомих як тензори. Ця можливість схожа на NumPy, але PyTorch забезпечує потужну підтримку графічного процесора, на відміну від NumPy, який працює з центральними процесорами. Ще PyTorch забезпечує підтримку глибоких нейронних мереж та глибокого навчання. PyTorch таклж добре підтримується основними хмарними платформами, що забезпечує легке масштабування.

З останнім релізом PyTorch став доступним сервіс мобільного деплою та сервіс TorchServe, що полегшує переміщення моделей PyTorch у продакшн. TorchServe підтримує деплой навчених моделей без необхідності писати власний код, включаючи кінцеві точки RESTful для інтеграції додатків. TorchServe підтримує будь-яке середовище машинного навчання, включаючи Amazon SageMaker, Kubernetes та Amazon EKS.

4.3 Реалізація експеременту

Для навчання згорткової нейронної мережі для моделі послідовних рекомендацій CosRec було використано набір даних «User Behavior Data from Taobao for Recommendation». Даний датасет є набором поведінки користувачів китайського інтернет-магазину Taobao для задач надання рекомендацій із неявним зворотним фідбеком (відгуком).

Для цього набору даних випадковим чином було відібрано близько одного мільйона користувачів, які мають таку «поведінку», як клацання (click) або ж перегляд сторінки товару, купівля, додавання товару до кошика та додавання товарів у список фаворитів у період з 25 листопада

по 3 грудня 2017 року. Кожен рядок датасету представляє конкретну взаємодію користувача з товаром, яка складається з ідентифікатора користувача, ідентифікатора елемента, ідентифікатора категорії товару, типу поведінки та позначки часу, все це розділено комами.

Кількість даних датасету «User Behavior Data from Taobao for Recommendation» відображено у таблиці 4.1.

Таблиця 4.1 – Статистика даних датасету «User Behavior Data from Taobao for Recommendation»

Розмірності	Кількість
Користувачи	987,994
Товари	4,162,024
Категорії	9,439
Взаємодії	100,150,807

Приклад даних з датасету «User Behavior Data from Taobao for Recommendation» зображено на рисунку 4.1.

Інша вибірка даних для навчання та тестування обох моделей є «Amazon». Дана вибірка є дуже об’ємною, тому вона розділена на декілька більш менших: «Electronics», «CDs-Vinyl», «Clothing-Shoes», «Digital-Music», «Home-Kitchen», «Beauty», «Movies-TV», «Toys-Games» та інші. Для даного експерименту із зазначеного датасету було вибрано категорію «Electronics», яка є меншою за обсягом ніж вибірка «User Behavior Data from Taobao for Recommendation». Вибір однієї категорії замість декількох одазу було зроблено для того, щоб проаналізувати як працюють обидві моделі на різних за масштабами вибірках даних.

	user_id	item_id	category_id	behavior	timestamp
0	1	2268318	2520377	pv	1511544070
1	1	2333346	2520771	pv	1511561733
2	1	2576651	149192	pv	1511572885
3	1	3830808	4181361	pv	1511593493
4	1	4365585	2520377	pv	1511596146
5	1	4606018	2735466	pv	1511616481
6	1	230380	411153	pv	1511644942
7	1	3827899	2920476	pv	1511713473
8	1	3745169	2891509	pv	1511725471
9	1	1531036	2920476	pv	1511733732

Рисунок 4.1 – Фрагмент даних з датасету «User Behavior Data from Taobao for Recommendation»

Amazon – це найбільша у світі платформа електронної комерції, яка має найбільший і найширший обсяг даних про товари, користувачей та їх поведінку. Продукція Amazon охоплює більшість сфер сучасного життя і має гарне різноманіття. Оригінальні датасети містять перегляди товарів та метадані від Amazon, зокрема датасет включає 142,8 мільйона переглядів у період з травня 1996 року по липень 2014 року вибраних випадковим чином. Цей набір даних включає відгуки (рейтинги, текст відгуку, оцінку корисності), метадані товару (опис, інформацію про категорії, ціну, торгову марку та особливості зображення) та посилання на товари. У даному експерименті використовується лише інформація про користувачів, товари та їх взаємодії. Amazon у своїх наборах даних заздалегідь відсіяли користувачів та товари з менш ніж 5 відгуками та видалили велику кількість нерелевантних даних.

Кількість даних датасету «Amazon (Electronics)» відображено у таблиці 4.2.

Таблиця 4.2 – Статистика даних датасету «Amazon (Electronics)»

Розмірності	Кількість
Користувачи	42,991
Товари	25,048
Категорії	773
Взаємодії	821,100

Приклад даних з датасету «Amazon (Electronics)» зображено на рисунку 4.2.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
 Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Рисунок 4.2 – Фрагмент даних з датасету «Amazon (Electronics)»

Як і в оригінальному методі, вибірки було поділено на тренувальну та тестову. 80% було відведено на тренувальні та верифікаційні дані, інші 20% було відведено для тесту. Ідентифікатори часу (timestamps) використовувались для визначення порядку послідовності дій користувачів в Інтернет-магазині. Наявність перегляду або оцінки для товару

розцінювалось як неявний зворотний зв'язок (тобто користувач взаємодіяв з елементом). Крім того, із вибірки «User Behavior Data from Taobao for Recommendation» було видалено користувачів та товари з холодним стартом (cold-start users and items). Тобто викидались товари та користувачі, які мали меньше ніж 5 дій (взаємодій), оскільки задача надання рекомендацій за умов холодного старту зазвичай розглядається як окреме питання [36].

Оскільки метод прогнозує топ 3 рекомендованих об’єктів (товарів), для оцінки якості прогнозування була використана метрика MAP (Mean Average Precision) [37].

MAP – це дуже популярна метрика оцінки алгоритмів, які здійснюють пошук інформації. Ця метрика також корисна для систем рекомендацій, наприклад, коли система показує короткий список продуктів, які, по її міркам, користувач також можете придбати після того, як він додав щось у кошик. Використання MAP для оцінки рекомендованого алгоритму означає, що задача рекомендації вирішується у контексті завдання ранжування. Для такого роду завдань потрібна метрика, яка винагороджує за отримання безлічі «правильних» або релевантних рекомендацій, і винагороджує за те, що вони знаходяться у списку раніше (мають вищий рейтинг).

Для розрахунку MAP потрібно спочатку розрахувати значення Precision для кожного об'єкту, а також значення Average Precision. Метрика Precision – це базова метрика якості ранжирування для одного об'єкта. З її допомогою можна порахувати частку релевантних елементів. Значення Precision можна знайти наступним чином:

$$P = \frac{TP}{TP+FP}, \tag{4.1}$$

де TP (True Positive) – це об’єкти, що були прогнозовані як позитивні приклади й дійсно виявилися позитивними (тобто, об’єкти, що були вірно спрогнозовані);

FP (False Positive) – об’єкти що були спрогнозовані як позитивні приклади, але насправді вони такими не являються (тобто, об’єкти, що були спрогнозовані невірно).

Метрика Precision не враховує порядок елементів в «топі» (напочатку списка). Цей недолік можна виправити порахувавши метрику Average Precision at N. Для задачі рекомендації N елементів, при кількості релевантних елементів у повному просторі елементів m, значення Average Precision для N елементів дорівнює:

$$AP@N = \frac{1}{m}\sum_{k=1}^{N}(P(k) \cdot rel(k)) , \quad (4.2)$$

де P(k) – це значення Precision для елементу k, rel(k) – це показник, який говорить про те, чи був цей k-й елемент релевантним чи ні. Якшо елемент k був релевантним, тоді rel(k) = 1, якщо елемент не був релевантним – тоді rel(k) = 0.

Насамкінець розраховуємо метрику Mean Average Precision (MAP). Ідея цієї метрики полягає в тому, щоб порахувати значення Average Precision at N (AP@N) для кожного об'єкта вибірки й усереднити. Для того, щоб розрахувати значення Mean Average Precision треба скористатися наступною формулою:

$$MAP@N = \frac{1}{|U|}\sum_{u=1}^{|U|}(AP@N)_u , \quad (4.3)$$

де $(AP@N)_u$ – значення Average Precision для користувача u;

|U| – загальна кількість користувачів.

Як вже було зазначено вище, модель рекомендує наступні 3 товара, які можуть зацікавити користувачів. Для навчання моделі значення параметру learning rate було встановлено як 0,001, обсяг партії (батчу)

становив 512, параметр negative sampling rate дорівнював 3 і розмір дропауту становив 0,5.

Результати експеременту для моделі CosRec, а також для модифікованого методу представлені у таблицях 4.3 та 4.4.

Таблиця 4.3 – Результати експерименту з моделлю CosRec та з її модифікацією на датасеті «User Behavior Data from Taobao for Recommendation»

Модель / Метрика	CosRec	CosRec модифікований
MAP	0.1871	0.1912

Таблиця 4.4 – Результати експерименту з моделлю CosRec та з її модифікацією на датасеті «Amazon (Electronics)»

Модель / Метрика	CosRec	CosRec модифікований
MAP	0.0389	0.0396

Як видно з таблиць 4.3 та 4.4 у проведеному експерименті показники модифікованого методу є кращими ніж показники звичайної моделі CosRec. У випадку з датасетом «User Behavior Data from Taobao for Recommendation» різниця у показника становить 2.1%, а у випадку з набором даних «Amazon (Electronics)» різниця між показниками – 1.8%. Можна сказати, що модифікований метод працює краще ніж оригінальний метод для вищезазначених вибірок. Підвищення показників ефективності вдалося досягти завдяки невеликої зміни в архітектурі згорткової нейронної мережі, що використовувалась у модулі попарного кодування. Тобто архітектура для модифікованої моделі CosRec була сгонфігурована вірно для обраних датасетів.

Також видно, що для вибірки «User Behavior Data from Taobao for Recommendation» модифікований метод дає більш значні результати ніж з датасетом «Amazon (Electronics)». Це може бути пов’язано з тим, що перша вибірка містить значно більше даних ніж друга. Можна зробити висновок що модифікація дає кращі результати, при роботі з великими об’ємами даних. З маленькими об’ємами даних виграш в ефективності не такий значний.

Експеримент показав, що модель CosRec може бути ефективно впроваджена та використана у веб-системах, що застосовуються у галузі електронної комерції, через те що дана модель може надавати релевантні рекомендації користувачам, а також вона легко налаштовується під конкретну задачу.

До того ж слід зазначити, що дана модель є простою й гнучкою у застосуванні. Це дає можливість легко впроваджувати модель CosRec у різних галузях або масштабувати її під різноманітні дані (для випадків, коли доступна історя взаємодій користувачів з об’єктами інтересу) й при цьому не вносити істотні зміни у структуру самої моделі, а лише трохи змінювати або налаштовувати її параметри.

Враховуючи вищезазначене можна зробити висновок, що модель CosRec, як система послідовних рекомендацій, може бути ефективно застосована у багатьох різних веб-системах надання рекомендацій зі сфер, де є історія взаємодії користувачів з об’єктами.

ВИСНОВКИ

Дана магістерська кваліфікаційна робота присвячена дослідженню систем послідовних рекомендацій, їх особливостям та основних підходів до побудови цих систем. У ході виконання даної кваліфікаційної роботи була проаналізована предметна галузь дослідження методів побудови систем послідовних рекомендацій. Було проаналізовано науково-дослідні публікації у галузі послідовних рекомендацій. На основі розглянутого матеріалу був проаналізован сучасний стан проблеми предметної області, основні поняття і існуючи рішення задачі побудови системи послідовних рекомендацій.

В ході дослідження було виявлено, що моделі з використанням згорткових нейронних мереж можуть бути впроваджені в системи послідовних рекомендацій, а також ефективно працювати, надаючи релевантні рекомендації, на базі попередніх взаємодій користувачів та об’єктів інтересу. Було розглянуто дві моделі послідовних рекомендацій, що використовують CNN – Caser та CosRec. У результаті аналізу було виявлено, що друга модель вирішує задачу надання рекомендацій набагато ефективніше. У процесі виконання роботи модель CosRec була модифікована. Проведені експерименти на датасетах «User Behavior Data from Taobao for Recommendation» та «Amazon (Electronics)» довели, що модифікована модель працює краще для заданих датасетів. На основі проведених експериментів можна зробити висновок, що модель CosRe є гнучкою у застосуванні. Це дає можливість легко її впроваджувати у різних галузях або масштабувати її під різноманітні дані й при цьому не вносити значні зміни у структуру самої моделі, а лише трохи змінювати її параметри.

Результати роботи відображені у тезах доповіді у Матеріалах 25-го Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті [22].

ПЕРЕЛІК ВИКОРИСТАННИХ ДЖЕРЕЛ

1. Lakshmi S., Dr. Lakshmi T.Adi. Recommendation Systems: Issues and challenges. *International Journal of Computer Science and Information Technologies*. 2014. Vol. 5(4). P.5771–5772.

2. Mohammed F., Manjaiah D.H. A survey on recommendation systems for social media using big data analytics. *International Journal of Latest Trends in Engineering and Technology*. 2017. Special Issue. P.48–58.

3. Aggarwal Charu C. Recommender Systems: The Textbook. Springer, 2016. 498p.

4. Бодянский Е.В., Рябова Н.В., Золотухин О.В. Многослойная адаптивная нечеткая вероятностная нейронная сеть в задачах классификации текстових документов. *Радиоэлектроника, информатика, управление.* 2015. №1(32). С. 39–45.

5. Debashis D., Laxman S., Sujoy D. A Survey on Recommendation System. *International Journal of Computer Applications*. 2017. Vol. 160(7). P.6-10.

6. Mladenic, D. Text-learning and Related Intelligent Agents: A Survey. *IEEE IntelligentSystems*. 1999. Vol.14(4). P. 44–54.

7. Shah K., Zafar A., Irfan U. Recommender Systems: Issues, Challenges, and Research Opportunities. *Information Science and Applications*. 2016. Vol. 1. P.1179–1189.

8. Yagnesh G. P., Vishal P.P. A Survey on Various Techniques of Recommendation System in Web Mining. *International Journal of Engineering Development and Research*. 2015. Vol. 3(4). P.696–700.

9. Xu C., Hongteng X., Yongfeng Zh. Sequential recommendation with user memory networks. *WSDM'18*: In Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Marina Del Rey, California, USA, 5-9 February, 2018. P.108–116.

10. Massimo Q., Paolo C., and Dietmar J. Sequence-aware recommender systems. *ACM Computing Surveys*. 2018. Vol. 51. P. 1–36.

11. He R., McAuley J. Fusing similarity models with markov chains for sparse sequential recommendation. *ICDMW:* Proceedings of the IEEE International Conference on Data Mining series, Barcelona, Spain, 12-15 December, 2016. P. 191–200.

12. Aggarwal Charu C. An Introduction to Frequent Pattern Mining. Springer, 2014. 471p.

13. Fang H., Danning Zh., Yiheng Sh., Guibing G. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Transactions on Information Systems*. 2020. Vol. 1. P. 1–41.

14. Ghim-Eng Y., Xiao-Li L., and Philip Y. Effective next-items recommendation via personalized sequential pattern mining. *In Database Systems for Advanced Applications*. 2012. Vol. 5. P. 48–64.

15. Davidson J., Liebald B., Liu J., Nandy P., Van Vleet T, Gargi U., Gupta S., He Y., Lambert M., Livingston B. The YouTube video recommendation system. *RecSys*. 2010. Vol. 1. P. 293–296.

16. Lerche L., Jannach D., Ludewig M. On the value of reminders within e-commerce recommendations. *UMAP*. 2016. Vol. 17. P.27–35.

17. Garcin F., Dimitrakakis C., Faltings B. Personalized news recommendation with context trees. *ACM*: Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 4 March, 2013. P. 105–112.

18. Hidasi B., Tikk D. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*. 2016. Vol. 30(2). P. 342–371.

19. Wang P., Guo J., Lan Y., Xu J., Wan S., Cheng X. Learning hierarchical representation model for next basket recommendation. SIGIR '15: Proceedings of the 38th ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile 15 August, 2015. P. 403–412.

20. Quadrana M., Karatzoglou A. Personalizing session-based recommendations with hierarchical recurrent neural networks. *RecSys '17*: Proceedings of the 11th ACM Conference on Recommender Systems, Como, Italy, 12 August, 2017. P. 130–137.

21. Wu Sh., Tang Y. Session-based recommendation with graph neural networks. *AAAI'16:* Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA, 27 January – 1 February, 2019. P. 1-9.

22. Сухомлінова Ю. І. Дослідження методів побудови систем послідовних рекомендацій. *Радіоелектроніка і молодь у XXI столітті*: тез. докл. 25-го Міжнародного молодіжного форуму. Харків: ХНУРЕ, 2021. С. 13–14.

23. LeCun, Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*. 1989. Vol. 1(4). P. 541–551

24. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Vol. 60(6). P. 84–90.

25. Zhang Y.N., Qu L., Chen J.W., Liu J.R., Guo D.S. Weights and structure determination method of multiple-input Sigmoid activation function neural network. *Appl. Res. Comput*. 2012. Vol. 29. P. 4113–4116.

26. Luo P., Li H.F. Research on Quantum Neural Network and its Applications Based on Tanh Activation Function. *Comput. Digit. Eng*. 2012. Vol. 16. P. 33–39.

27. Tang Z., Luo L., Peng H., Li S. A joint residual network with paired ReLUs activation for image super-resolution. *Neurocomputing*. 2018. Vol. 273. P. 37–46.

28. Buscema M. Back propagation neural networks. *Substance Use & Misuse*. 1998. Vol. 33. P. 233–270.

29. Tang J., Wang K. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. *WSDM 2018*: In Proceedings of the ACM International Conference on Web Search and Data Mining, Los Angeles, California, USA, 5-9 February, 2018. P. 565–573.

30. Kim Y. Convolutional Neural Networks for Sentence Classification. *EMNLP 2016*: In Proceedings of the Conference on Empirical Methods on Natural Language Processing, Austin, Texas, USA, 1 November, 2016. P. 1756-1751

31. Yan A., Cheng Sh., Kang W-C., Wan M., McAuley J. CosRec: 2D Convolutional Neural Networks for Sequential Recommendation. *CIKM '19:* Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 6 November, 2019. P. 2173–2176.

32. Yuan F., Karatzoglou A., Arapakis I., Jose J. M., He X. A Simple Convolutional Generative Network for Next Item Recommendation. *WSDM 2019*: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 1 February, 2019. P. 51–55.

33. User Behavior Data from Taobao for Recommendation. *TC Lab*. URL: https://tianchi.aliyun.com/dataset/dataDetail?dataId=649 (дата звернення 01.04.2021)

34. Amazon product data. *UCSD*. URL: http://jmcauley.ucsd.edu/data/amazon/ (дата звернення 01.04.2021)

35. Pytorch [Basics] – Intro to CNN. *Towards data science*. URL: https://towardsdatascience.com/pytorch-basics-how-to-train-your-neural-net-intro-to-cnn-26a14c2ea29/ (дата звернення 07.04.2021)

36. He R., McAuley J. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. ICDMW: Proceedings of the IEEE International Conference on Data Mining series, Barcelona, Spain, 12-15 December, 2016. P. 191–200.

37. Mean Average Precision (MAP) For Recommender Systems. *SonyaSawtelle*. URL: https://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html#Further-Reading-on-MAP/ (дата звернення 07.04.2021)