

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук,
проф.

_____ Ю. П. Кондратенко
« ____ » _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ
ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ
МЕРЕЖ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.21810217

Виконала студентка 4-го курсу, групи 402

_____ *В.В. Луцишина*

« » червня 2022 р.

Керівник: ст. викладач каф. ІІЗ

_____ *С.В. Дворецька*

« » червня 2022 р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти бакалавр
Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)
Галузь знань 12 «Інформаційні технології»
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук,
проф.

_____ Ю. П. Кондратенко
«___» _____ 2022р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Видано студентці групи 402 факультету комп'ютерних наук Луцишиній
Валерії Володимирівні.

1. Тема кваліфікаційної роботи «Інформаційний вебзастосунок підбору туристичного туру за індивідуальними вподобаннями на основі нейронних мереж».

Керівник роботи Дворецька Світлана Володимирівна, ст. викладач каф. ІІЗ.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» _____ 20__ р. №

2. Строк представлення кваліфікаційної роботи студентом «___» _____ 20__ р.

3. Вхідні (початкові) дані до роботи: результати тестування деякої групи людей.

Очікуваний результат: інформаційний веб-застосунок підбору туристичного туру за індивідуальними вподобаннями на основі нейронних мереж з використанням заданих результатів тестування.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- що таке веб-сайт та його місце у повсякденному житті;
- система обміну даними між вузлами мережі інтернет;
- аналіз інструментів розробки сучасних веб-сайтів;
- огляд нейронних мереж та сфери їх застосування;
- структура та проектування власного веб-сайту;
- огляд реалізації проектування.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Правила охорони праці при роботі з комп'ютером».

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А.О к.т.н., доцент	

Керівник роботи ст. викладач каф. ПІЗ Дворецька С.В.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Луцишина В.В.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 22 » _____ листопада _____ 2021 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Інформаційний вебзастосунок підбору туристичного туру за індивідуальними вподобаннями на основі нейронних мереж

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	30.10.2021	30.10.2021	Виконано
2	Отримання завдання на виконання БКР	22.11.2021	22.11.2021	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	07.12.2021	09.12.2021	Виконано
4	Отримання завдання на переддипломну практику	22.05.2022	22.05.2022	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	23.05.2022	04.06.2022	Виконано
6	Розробка звіту з переддипломної практики	04.06.2022	06.06.2022	Виконано
7	Виконання БКР: аналіз сучасного стану задачі відслідковування очей, огляд існуючих технологій, розробка ПЗ	28.02.2022	27.03.2022	Виконано
8	Попередній захист БКР на засіданні комісії кафедри	30.05.2022	30.05.2022	Виконано
9	Доробка та остаточне оформлення БКР	02.06.2022	17.06.2022	Виконано
10	Подання БКР рецензенту	20.06.2022	20.06.2022	Виконано
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	21.06.2022	21.06.2022	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	27.06.2022	27.06.2022	Виконано

Розробила студентка Луцишина В.В.
(прізвище, ім'я, по батькові студента) _____ (підпис)

Керівник роботи _____ ст.викладач каф. ПЗ Дворецька С. В.
_____ (посада, прізвище, ім'я, по батькові) _____ (підпис)

« 9 » _____ 12 _____ 2021 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студентки групи 402 ЧНУ ім.
Петра Могили

Луцишиної Валерії Володимирівни

Тема: «Інформаційний вебзастосунок підбору туристичного туру за
індивідуальними вподобаннями на основі нейронних мереж»

Об'єкт роботи – процес підбору туристичного туру за індивідуальними
вподобаннями.

Предмет роботи – засоби та технології розробки вебзастосунків для
підбору елементів із множини за параметрами на основі нейронних мереж.

Мета роботи – підвищити рівень зацікавленості клієнтів та зручності
процесу отримання інформації щодо наявних туристичних турів шляхом
розробки вебзастосунку для їх підбору із врахуванням індивідуальних
вподобань на основі нейронних мереж.

Для досягнення мети необхідно вирішити наступні задачі:

- розкриття актуальності обраної теми у сучасному світі;
- розкриття поняття «вебсайт»;
- з'ясування принципів роботи нейронних мереж;
- огляд наявних інструментів для створення вебзастосунків;
- проектування майбутньої програми;
- тестування готового вебзастосунку.

Розроблений інформаційний веб-застосунок у складі бази даних, Web-
інтерфейсу та програмного коду згідно отриманого завдання.

Для створення серверної частини обробки даних були застосовані такі
технології, як Node.js та MongoDB. Клієнтський інтерфейс було створено за
допомогою HTML, CSS, JS, а також бібліотеки React.

Кількість сторінок -86, таблиць -0, рисунків -45, додатків -0 та 28 джерел.

Ключові слова: Web-інтерфейс, сервер, клієнт, HTML, CSS, JS, React,
Node.js.

ABSTRACT

of bachelor's qualification work

student of group 402 of ChNU named after Petra Mogili

Lutsyshyna Valeria Volodymyrivna

Topic: «Information web application of tour selection according to individual preferences based on neural networks»

The object of work is the process of selecting a tourist tour according to individual preferences.

Subject of work - tools and technologies for developing web applications for the selection of elements from the set of parameters based on neural networks.

The purpose of the work is to increase the level of interest of customers and the convenience of the process of obtaining information about existing tourist tours by developing a web application for their selection based on individual preferences based on neural networks.

To achieve this goal it is necessary to solve the following tasks:

- revealing the relevance of the chosen topic in the modern world;
- disclosure of the concept of "website";
- clarification of the principles of neural networks;
- review of available tools for creating web applications;
- designing a future program;
- testing of the finished web application.

Developed informational web application consisting of database, Web-interface and program code according to the received task.

Technologies such as Node.js and MongoDB were used to create the server part of data processing. The client interface was created using HTML, CSS, JS, and the React library.

Number of pages - 86, tables - 0, figures - 45, appendices - 0 and 28 sources.

Keywords: Web-interface, server, client, HTML, CSS, JS, React, Node.js.

ЗМІСТ

АНОТАЦІЯ.....	5
ВСТУП.....	4
1 АНАЛІЗ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ВЕБЗАСТОСУНКАХ	6
1.1 Актуальність обраної теми.....	6
1.2 Розкриття поняття «вебсайт».....	10
1.3 Огляд принципів роботи штучних нейронних мереж.....	14
1.4 Постановка задачі	23
Висновки до розділу 1.....	24
2 ІНСТРУМЕНТИ, МЕТОДИ ТА СЕРЕДОВИЩА ДЛЯ СТВОРЕННЯ ВЕБСАЙТІВ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ	27
2.1 Функціональна мова програмування JS.....	27
2.2 HTML – основа вебсторінок.....	30
2.3 Таблиця стилів CSS	32
2.4 Node.js.....	39
2.5 MongoDB.....	42
2.6 Бібліотека React.....	46
Висновки до розділу 2.....	48
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ З НЕЙРОННИМИ МЕРЕЖАМИ.....	50
3.1 Структура системи	50
3.2 Проектування туристичного сайту	51

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

3.3 Алгоритм зворотного поширення помилки.....	54
Висновки до 3 розділу.....	59
4 ОГЛЯД ГОТОВОГО ПРОГРАМНОГО ЗАСТОСУНКУ	60
4.1 Інтерфейс програми.....	60
4.2 Серверна частина	63
Висновки до 4 розділу.....	65
5 ОХОРОНА ПРАЦІ	68
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79

ВСТУП

Сайт – одна або декілька зв'язаних за змістом веб-сторінок з певною тематикою контенту. Це один з найсучасніших засобів для реклами, яким уже користуються мільйони людей по всьому світу. Веб-сайт повинен мати гарне оформлення, просту та інтуїтивно зрозумілу логіку, а також певну інформацію про заданий об'єкт. Перші веб-сторінки мали дуже простий дизайн. Проте з розвитком браузерів та інструментів для створення сайтів ситуація змінилася. Сьогодні можна побачити багато цікавих сайтів зі складною анімацією та більшою взаємодією з користувачем.

Сьогодні стає все більш популярним використання нейронних мереж у програмах. Вони можуть розпізнавати звуки, зображення, класифікувати дані, передбачати наступний крок тощо. Штучні нейронні мережі – це математична програмна модель, побудована за принципом функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Це є спробою відтворити роботу реального мозку людини.

Наше мислення – результат роботи мережі нейронів в мозку. Нейрони передають одне одному хімічні сигнали за допомогою електричних імпульсів. Внаслідок такої активності з'являються наші відчуття та мислення. Штучний інтелект і когнітивне моделювання намагаються імітувати деякі властивості біологічних нейронних мереж. Штучні нейронні мережі є системою з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай досить прості (особливо в порівнянні з процесорами, використовуваними в персональних комп'ютерах). Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він періодично посилає іншим процесорам. І, тим не менше, будучи з'єднаними в досить велику мережу з керованим взаємодією, такі окремо прості процесори разом здатні виконувати досить складні завдання.

Пояснювальна записка

до кваліфікаційної роботи

на тему:

ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.21810217

Виконала студентка 4-го курсу, групи 402

_____ *В.В. Луцишина*

(підпис, ініціали та прізвище)

« » _____ червня _____ 2022р.

Керівник: ст. викладач каф. ПІЗ.

_____ *С.В. Дворецька*

(підпис, ініціали та прізвище)

« » _____ червня _____ 2022р.

Миколаїв – 2022

1 АНАЛІЗ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ВЕБЗАСТОСУНКАХ

1.1 Актуальність обраної теми

В інтернеті можна знайти багато сайтів, створені на різноманітні теми: інтернет-магазин, спортивні секції, послуги різних майстрів модної індустрії тощо. Найчастіше такі сайти містять інформацію про контакти: номери телефонів, місце знаходження, сторінки соціальних мереж. Вони необхідні для того, аби клієнти могли поставити запитання, вирішити проблеми, зробити замовлення чи просто проконсультуватися. Уся ця робота вимагає часу та працівників, які повинні отримувати платню.

Нейронні мережі [1] здатні економити час та бюджет фірми. З їх допомогою можна створювати безліч сайтів, які допоможуть відвідувачам підібрати найбільш підходящий товар або послугу. Наприклад, один з найвідоміших онлайн-кінотеатрів у світі – Netflix – підійшов до питання рекомендації фільмів дуже наполегливо та відповідально. Компанія запустила проект системи рекомендацій на основі концепції під назвою «глибоке навчання». Це цілий набір методів, досліджень та практик в галузі штучного інтелекту, головною задачею якого є поки що нереалізоване бажання людей змусити комп'ютер мислити, як людський мозок. У цьому сенсі Netflix пішов за веб-гігантами, такими як Google та Facebook. Вищезазначені компанії вже активно використовують глибоке навчання для найрізноманітніших цілей: від сортування зображень до розпізнавання голосу.

Однак підхід Netflix принципово відрізняється від інших. Річ в тім, що, на відміну від Google та Facebook, які проводять експерименти зі штучним інтелектом на власних потужностях, Netflix планує запускати свої алгоритми на серверах Amazon. І це вкотре показує, що хмарні технології дозволяє середньому бізнесу конкурувати з корпораціями на рівних.

Штучний інтелект – потенційний технологічний хіт нового десятиліття: по прогнозам, до 2030-ого року сім з десяти компаній будуть застосовувати його в онлайн-бізнесі. Штучний інтелект стане справжнім асистентом клієнта: допоможе з пошуком потрібних товарів, дасть рекомендації, збереже та нагадає пізніше про товари, які сподобалися клієнту.

Ці технології працюють вже сьогодні. Завдяки штучному інтелекту маркетплейс Overstock підвищив дохід майже на чверть (23%). Все тому, що штучний інтелект фіксував клієнтів, які серйозно налаштовані на покупку, і робив їм персональні пропозиції. Інструментарій штучного інтелекту (відслідковування клієнтських вподобань, чат-боти) активно використовує кожен другий ефективний маркетолог (49% маркетологів – дані з Salesforce). Все це дає деяким маркетплейсам до 40% продажів через онлайн.

Штучний інтелект допомагає утримувати користувачів на сайтах. Це важливо, адже третина покидає вебсайти через нецікавий, непривабливий опис продукту (дані Retail Dive). Саме штучний інтелект формує контент (опис товарів) на eBay на основі ретельного аналізу 130-ти мільйонів зображень в 16-ти тисячах товарних категоріях.

Однак, і це ще неповний перелік можливостей штучного інтелекту. Сервіс Blueshift виробляє персоналізовану маркетингову стратегію, враховуючи дії користувача, Market Brew вивчає як проходить оновлення пошукових ресурсів і допомагає коригувати SEO-поширення. Сервіс Zoomph займається аналізом поведінки аудиторії в соціальних мережах.

Штучні нейронні мережі [2] стають все більш популярними у повсякденному житті людини. Дедалі частіше їх можна зустріти у таких сферах життя: медицина, економіка, охорона і безпека життя, будівництво та проектування тощо.

Глибокі нейронні мережі можуть допомогти в інтерпретації медичних сканів патологій, електрокардіограм, ендоскопії. Особлива увага приділяється

радіології – використанню нейромереж для аналізу рентгенівських знімків. Google використовували алгоритми для інтерпретації знімків грудної клітки, щоб поставити 14 різних діагнозів, від пневмонії до гіпертрофії серця та колапсу легені. Нейронні мережі також здатні діагностувати окремі види раку, переломи, крововиливи, ураження шкіри та безліч інших захворювань. Алгоритми можуть покращити роботу дерматологів, кардіологів, офтальмологів та навіть психотерапевтів, дозволяючи відстежувати розвиток депресії.

Нейронні мережі можуть бути корисними в медицині на таких рівнях:

- допомога лікарям швидко та точно інтерпретувати зображення;
- зменшують кількість лікарських помилок;
- допомагають пацієнтам самим слідкувати за станом свого здоров'я за допомогою даних з датчиків.

У сфері бізнесу нейронні мережі допомагають банкам видавати кредити, оцінюючи при цьому кредитну історію клієнта, динаміку розвитку компанії, основні її фінансові показники і багато іншого. Такі системи вже успішно застосовуються у деяких банках США. Вони допомагають виявляти потенційних неплатників. Іншими важливими сферами є автоматичний рейтинг облігацій, аналіз ринку, консультаційні системи щодо біржової торгівлі, автоматичне зчитування чеків тощо.

Нейронні мережі також допомагають у проектуванні нових телекомунікацій. З їх допомогою знаходять оптимальний шлях трафіку між вузлами. Перетворювати фотографії на картини та автоматично редагувати їх допомагають такі сервіси, як Prisma та MSQRD.

Розпізнавати мову по губах краще за людину. Це вдалося зробити розробникам Оксфордського університету. Як вихідний матеріал автори роботи взяли базу даних Grid, в якій було зібрано понад 32 тисячі відеозаписів і, слідуючи алгоритму, навчали нейромережу. Результати тестування

показали, що LipNet може правильно розпізнавати мову по губах у 93,4 відсотках випадків. Це має серйозно допомогти людям із порушеннями слуху.

У сфері захисту та оборони нейронні мережі допомагають у боротьбі з тероризмом, розпізнаванні облич, придушенні шуму, ідентифікації людини, розпізнаванні голосу та людей в натовпі, розпізнаванні автомобільних номерів, аналізі аерокосмічних знімків, виявленні підробок, відстеженні цілі та керуванні зброєю.

НМ також знайшли місце й у освітній сфері. За їх допомогою забезпечується адаптивне програмне забезпечення для навчання, динамічне прогнозування, аналіз та прогнозування системи освіти, моделювання успішності студентів та профілювання особистості.

Якість та кількість врожаю залежить від багатьох факторів. Проаналізувати їх все і прийняти правильне рішення не в змозі жоден досвідчений фахівець, тому в цьому напрямку застосування сучасних технологій просто неминуче.

Нейромережу [3] можна навчити відрізняти корисні культури від бур'янів. «Вивчивши» кілька мільйонів фотографій зразків необхідної культури, система за допомогою відеокамери може за кілька мілісекунд визначити, чи перед нею корисна рослина, чи бур'ян. Після низки етапів застосування гербіциду та подальшого аналізу ефективності, нейромережа навчиться розуміти, у яких випадках та яка кількість речовини дійсно необхідна, і сама зможе приймати рішення. Це здається фантастикою, але саме така нейромережа керує роботом, розробленим у США.

Без особливих складнощів можна навчити мережу відрізняти не лише корисну культуру від бур'янів, а й здорову рослину від хворої. Вона працюватиме за тим самим принципом. Вивчивши напочатку базу зображень здорових та хворих рослин на різних стадіях росту, система зможе оцінити рівень загрози для врожаю та запропонувати способи вирішення проблеми.

Впровадити такий «розумний фільтр» можна для роботи в полях та сортування вже зібраного врожаю.

Ще одним прикладом застосування нейронних мереж у сільському господарстві є впровадження машин, які можуть самостійно збирати врожай. Машина під управлінням нейронної мережі зможе сама оцінити форму, вагу та колір ягід, збираючи лише стиглі, придатні для продажу одиниці.

Нейронні мережі можуть допомагати вам об'їжджати затори вже зараз, а в майбутньому працювати помічниками автопілотів автомобілів, постійно аналізуючи отриману інформацію.

Корпорація Google представила НМ [4] MetNet, яка може прогнозувати погоду на вісім годин уперед. У компанії стверджують, що алгоритм може передбачити опади з точністю до одного кілометра на місцевості та двох хвилин за часом. Вхідні дані до мережі надходять автоматично від радіолокаційних станцій та супутникових мереж без необхідності перевірки людиною. Система отримує дані про наявність та склад хмар над поверхнею Землі, а також про швидкість їхнього руху. На виході MetNet показує розподіл ймовірностей, який використовується для визначення найімовірніших рівнів опадів у різних регіонах США. Модель розраховує прогноз погоди для квадратів 64 на 64 кілометри з роздільною здатністю в один кілометр. При цьому для кожної ділянки нейромережі потрібні дані із сусідніх квадратів.

Також одним з найпопулярніших напрямків розвитку НМ є Artificial Intelligence Art або просто AI-Art, коли на основі аналізу великого масиву даних машина створює нові пісні популярних виконавців.

1.2 Розкриття поняття «вебсайт»

У широкому розумінні програма – набір команд, необхідних для виконання завдання. Комп'ютерна програма складається з команд, які комп'ютер розуміє та здатен виконувати.

Архітектура програмного забезпечення – спосіб створення такої програми. Найпростішим та найпоширенішим прикладом є клієнт-серверна архітектура. [5] Вона передбачає набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них, тобто виконують запит на сервер. А також набір клієнтів, які отримують відповіді на свої запити для подальшої обробки та виведення на екран користувача. Також існує архітектура peer-to-peer, в основі якої лежить мережа рівноправних вузлів. Тобто, немає такого явного розділення функцій, як у попередньому прикладі.

Веб-застосунок – програма, в якій клієнт виступає браузером, а сервером – веб-сервер. Це є прикладом класичної клієнт-серверної архітектури. Істотною його перевагою є те, що він виконується в браузері і не залежить від операційної системи користувача. Тобто, не потрібно писати декілька програм для операційної системи Windows, Linux, Mac OS. Достатньо лише один раз написати таку програму, і вона буде працювати на будь-якій операційній системі користувача. Це значно прискорює створення таких програм та полегшує їх використання.

Веб-сайт складається з веб-сторінок, які згруповані разом і пов'язані між собою. Веб-сторінка - документ, який можна відобразити у веб-браузері, такому як Firefox, Google Chrome, Opera, Microsoft Internet Explorer тощо. Їх також часто називають просто «сторінками». Такі документи написані мовою HTML. Веб-сторінка може містити різноманітні ресурси, наприклад стилі (CSS), скрипти (додаються для інтерактивності), медіа (зображення, звуки, відео).

Багато новачків в Інтернеті плутають пошукові системи та браузери. Браузер — це частина програмного забезпечення, яка відправляє запити на сервер, отримує відповідь у вигляді веб-сторінки та відображає її. Пошукова система – це веб-сайт, який допомагає людям знаходити веб-сторінки з інших веб-сайтів. Плутанина виникає через те, що під час першого запуску браузера, він відображає домашню сторінку пошукової системи.

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Інтернет – велика мережа зв’язаних між собою вузлів. Вузли можуть бути персональним комп’ютером, телефоном, планшетом, ноутбуком, роутер тощо. Зв’язок між ними може бути провідним(оптичне волокно, коаксіальний кабель, вита пара) та безпроводним(Wi-Fi).

Для того, аби усі вузли могли обмінюватися інформацією через відкритий зв’язок, повинен бути набір правил, який забезпечує доставку саме до відповідного вузла даних, захищеність інформації від сторонніх очей, надійність та швидкість. Цей набір називається протоколом. Існує безліч протоколів інтернету. Кожен з них виконує певну функцію. Наприклад, протокол TCP – один з основних протоколів передачі даних інтернету. Набір даних, який передається цим протоколом, називається сегментом. TCP забезпечує попереднє встановлення зв’язку, а у разі втрати даних виконує повторний запит даних, а також перешкоджає дублюванню у разі отримання двох копій одного сегмента. [6]

Цей протокол є протоколом транспортного рівня моделі TCP/IP, яка описує процес передачі даних. Ця модель має таку структуру:

- прикладний рівень,
- транспортний рівень,
- мережевий рівень,
- канальний рівень.

Кожен з рівнів виконує свої функції та має свій набір протоколів. Канальний рівень описує кодування даних для передачі за допомогою нулів та одиниць. Мережевий рівень відповідає за маршрутизацію та доставку інформації до кінцевого вузла. Транспортний протокол забезпечує передачу даних конкретній програмі, перевіряє їх цілісність тощо. На прикладному рівні є протоколи, які безпосередньо працюють з прикладними програмами. Протокол HTTP є прикладом цього рівня, який описує роботу такої програми, як браузер. Цей протокол спочатку був призначений саме для отримання даних у форматі HTML. Зараз використовується для передачі будь-яких даних.

Спочатку браузер повинен сформувати HTTP-запит, який складається з заголовку, який описує службову інформацію та тіла з даними. Далі цей запит відправляється на сервер. Браузер отримує HTTP-відповідь, яка також складається з заголовків та тіла, в якому розміщується HTML-сторінка. Він аналізує та відображає користувачеві цю сторінку. HTTP протокол має декілька методів запиту, які відображають мету цього запиту:

- GET – метод для запиту вказаного ресурсу,
- POST – призначений для відправки даних на сервер,
- DELETE – видаляє вказаний ресурс,
- PATCH – частково змінює ресурс,
- PUT – створює новий ресурс або замінює даними, які представлені у тілі запиту.

Це лише основні методи. HTTP-запит може бути різної структури, в залежності від методу. Наприклад, запит з методом GET не має тіла. Таким чином браузер здійснює запити та відображає користувачу ту чи іншу сторінку сайту.

Кожен вузол у мережі інтернет має IP-адресу, яка однозначно ідентифікує пристрій для передачі йому даних. Але ці адреси дуже довгі і складні для запам'ятовування. З цією проблемою добре справляється система DNS. Служба DNS – це глобальний розподільний сервіс, який перетворює домени в IP-адреси. Тобто, кожній IP-адресі ставиться у відповідність доменне ім'я, яке легше запам'ятовується та сприймається людиною.

Служба DNS має структуру, яка складається з авторитативних та рекурсивних DNS-сервісів. Клієнти (браузери) зазвичай не відправляють запити напряму до авторитативних сервісів, тому рекурсивні сервіси відіграють роль проміжної ланки, яка містить тимчасову пам'ять – кеш. У ньому зберігаються відповіді на раніше зроблені запити.

Веб-сторінки сайту розміщуються на сервері. Користувач повинен ввести URL-адресу сторінки в браузері, аби сервер міг зрозуміти, яка саме веб-

сторінка потрібна користувачу, а потім відправити її браузеру для відображення. URL складається з протоколу, домену та шляху і відіграє роль ідентифікатора ресурсу.

1.3 Огляд принципів роботи штучних нейронних мереж

Менше ніж через десять років після того, як зламав нацистську шифруючу машину Enigma і допоміг об'єднаним силам перемогти у Другій світовій війні, математик Алан Тьюрінг змінив історію вдруге, поставивши просте запитання: «Чи можуть машини мислити?» [7]

За своєю суттю штучний інтелект є галуззю інформатики, яка прагне дати ствердну відповідь на запитання Тьюрінга. Це спроба відтворити або імітувати людський інтелект у машинах. Нейронні мережі – один із способів реалізації штучного інтелекту. Взагалі у штучний інтелект виокремлюють велику область – машинне навчання. Вона вивчає методи побудови алгоритмів, здатних самостійно вчитися. Це необхідно, якщо не існує чіткого вирішення будь-якої задачі. У цьому випадку простіше не шукати правильного рішення, а створити механізм, який сам вигидає метод для його пошуку. У багатьох статтях можна зустріти термін «глибоке» або «глибинне» навчання. Під ним розуміють алгоритми машинного навчання, що використовують багато обчислювальних ресурсів. Найчастіше під ним розуміють просто «нейронні мережі». [8]

Нейронна мережа моделює роботу людської нервової системи, особливістю якої є здатність до самонавчання з урахуванням попереднього досвіду. Так, всі вони, так само, як і мозок людини, складаються з великої кількості пов'язаних між собою однотипних елементів - нейронів, які імітують нейрони головного мозку.

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

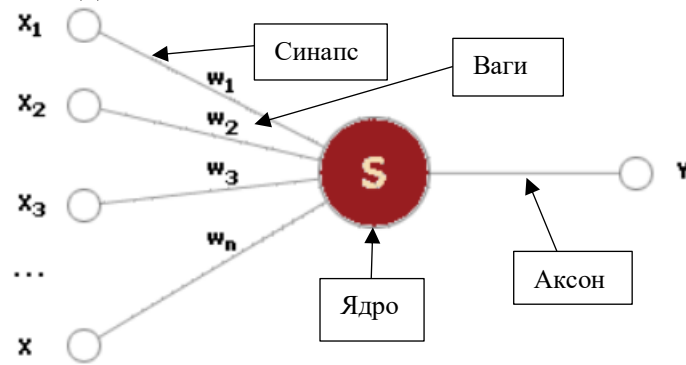


Рисунок 1.1 – Схема штучного нейрона

З малюнка видно, що штучний нейрон, як і живий, складається з синапсів, що зв'язують входи нейрона з ядром; ядра нейрона, що здійснює обробку вхідних сигналів та аксона, який пов'язує нейрон з нейронами наступного шару. Кожен синапс має вагу, який визначає, наскільки відповідний вхід нейрону впливає на його стан. Стан нейрона визначається за формулою:

(1.1)

де S – стан нейрона;
 n – число входів нейрона;
 x_i – значення i -го входу нейрона;
 w_i – вага i -го синапсу.

Потім визначається значення аксона нейрона за формулою:

(1.2)

де Y – значення нейрона;
 f – деяка функція, яка називається функцією активації;
 S – стан нейрона.

Найчастіше в якості функції активації використовується так звана сигмоїда, яка має наступний вигляд:

(1.3)

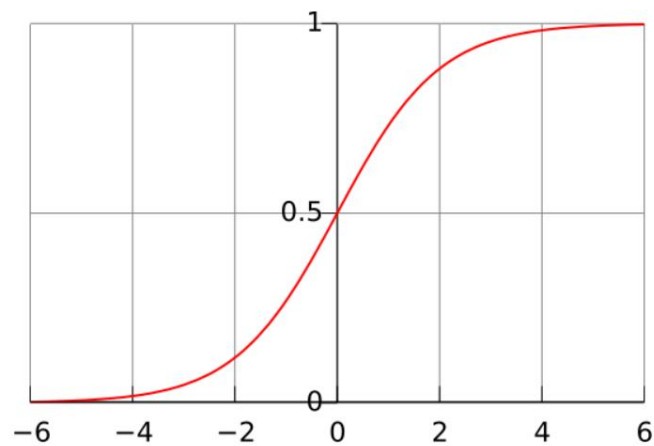


Рисунок 1.2 – Графік сигмоїдної функції активації

Також функція активації може бути задана такими функціями:

– порогова функція;

(1.4)

де a – пороговий параметр.

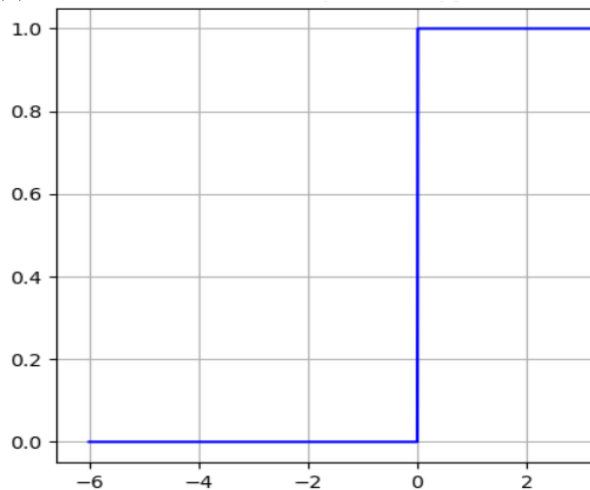


Рисунок 1.3 – Графік порогової функції, при $a = 0$

– лінійна функція;

(1.5)

де c – коефіцієнт пропорційності.

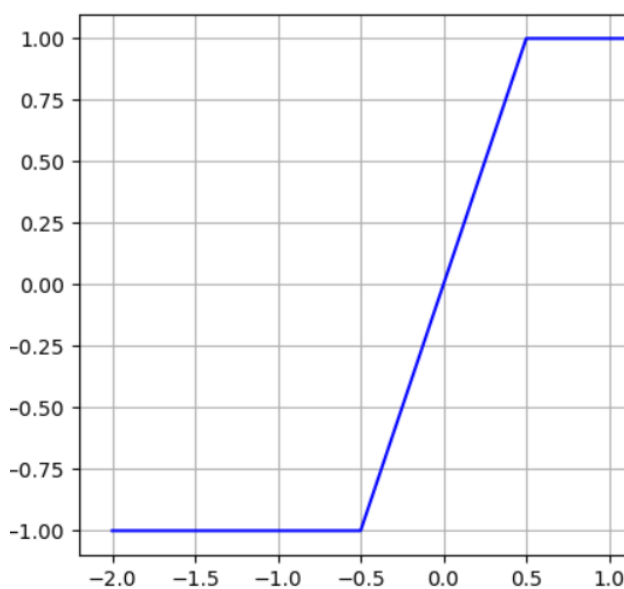


Рисунок 1.4 – Графік лінійної функції, при $c = 2$

– логістична функція.

(1.6)

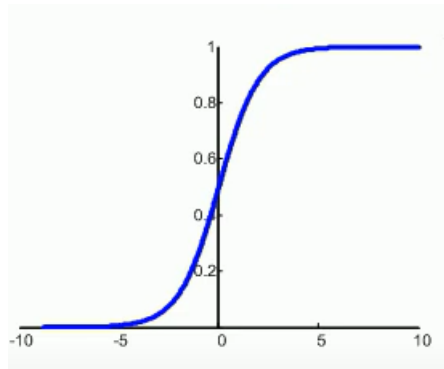


Рисунок 1.5 – Графік функції ReLU

Крім формул, представлених вище, існують безліч інших функцій активації. Дослідники в галузі штучного інтелекту постійно шукають нові способи покращити нейронні мережі, у тому числі модифікуючи вже існуючі способи та прийоми обробки вхідних сигналів. Вибір активаційної функції одне з першочергових завдань для побудови правильної нейронної мережі, однак, не існує єдиного правила, у якому випадку функцію застосовувати. Є лише практика, яка показує, що сигмоїда частіше використовується у завданнях класифікації.

При вирішенні складних завдань з використанням багатошарових мереж можливо комбінувати функції на різних шарах, щоб досягти найвищої точності та швидкості навчання.

Існує безліч правил для навчання нейронної мережі. Усі вони є похідними від чотирьох основних:

- правило корекції за помилкою;
- правило навчання Больцмана;
- дельта правило;
- навчання методом змагання.

Правило корекції за помилкою передбачає для кожного вхідного вектору бажаний вихід d . Якщо вихід відрізняється від бажаного, вага змінюється на одиницю, а знак визначається протилежним від знаку помилки. У випадку правильних вихідних даних, навчання не відбувається, ваги не змінюються.

Правило навчання Больцмана є стохастичним правилом навчання, яке виходить з інформаційних теоретичних і термодинамічних принципів. Метою навчання Больцмана є таке налаштування вагових коефіцієнтів, при якому стани видимих нейронів задовольняють бажаний розподіл вірогідності.

Дельта правило полягає у навчанні за допомогою вектора помилок:

$$E = D - Y, \quad (1.7)$$

де E – вектор помилки;

D – очікуваний вектор;

Y – вихідний вектор.

Тоді формулу для коригування ваг можна записати наступним чином:

$$w_j(t+1) = w_j(t) + e_i x_j, \quad (1.8)$$

де w – вага;

j – номер сигналу;

t – номер поточної ітерації;

i – номер нейрону. [9]

Таким чином, вага вхідного сигналу нейрона змінюється у бік зменшення помилки пропорційно величині сумарної помилки нейрона. Часто вводять коефіцієнт пропорційності, на який множиться величина помилки. Цей коефіцієнт називається швидкістю навчання. Отже, кінцева формула для коригування ваг має такий вигляд:

$$w_j(t+1) = w_j(t) + e_i \eta x_j, \quad (1.9)$$

де w – вага;

j – номер сигналу;

t – номер поточної ітерації;

i – номер нейрону;

η – швидкість навчання.

При навчанні змагання вихідні нейрони змагаються між собою за активізацію. Це явище відоме як правило «переможець бере все».

Існує багато типів нейронних мереж. На рис. 1.1 зображено модель персептрона – одношарова нейронна мережа. У цьому типі нейронної мережі немає прихованих шарів. Він приймає вхідні дані та обчислює зважені вхідні дані для кожного входу. Після цього він використовує функцію активації.

Наступним типом нейронної мережі є нейронна мережа з прямим зв'язком, яка зображена на рис. 1.6.

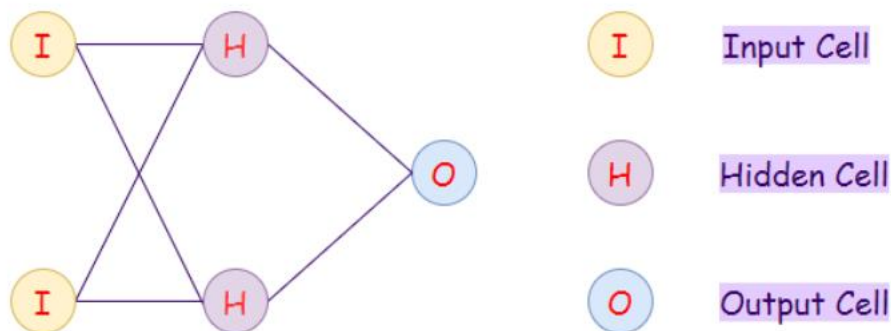


Рисунок 1.6 – Нейронна мережа з прямим зв'язком

Нейронна мережа з прямим зв'язком – це штучна нейронна мережа, в якій вузли ніколи не утворюють цикл. Приховані шари не мають зв'язку з зовнішнім світом, тому їх називають прихованими шарами. У нейронній мережі з прямим зв'язком кожен персептрон в одному шарі пов'язаний з

кожним вузлом наступного шару. Тому всі вузли повністю з'єднані. Ще варто зауважити, що між вузлами в одному шарі немає видимого чи невидимого зв'язку.

Ще одним прикладом нейронної мережі є глибока мережа прямої передачі.

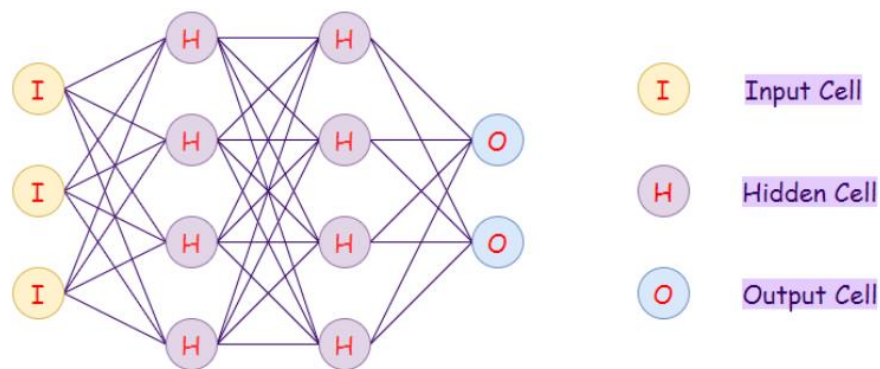


Рисунок 1.7 – Глибока мережа прямої передачі

Глибока мережа прямої подачі – це мережа з прямим зв'язком, яка використовує більше ніж один прихований шар.

Рекурентні нейронні мережі є різновидом мереж прямої подачі. Першою мережею такого типу була так звана мережа Йордана, коли кожна з прихованих клітин отримувала свій вихід з фіксованою затримкою — одну або кілька ітерацій. Ми використовуємо цей тип нейронної мережі, де нам потрібно отримати доступ до попередньої інформації в поточних ітераціях. Наприклад, коли ми намагаємося передбачити наступне слово в реченні, нам спочатку потрібно знати раніше вживані слова. Розмір моделі не збільшується з розміром вхідних даних. Однак проблема цієї нейронної мережі полягає в повільній швидкості обчислень. Більше того, він не може враховувати будь-які майбутні внески для поточного стану. Він не може пам'ятати відомостей з давнього часу.

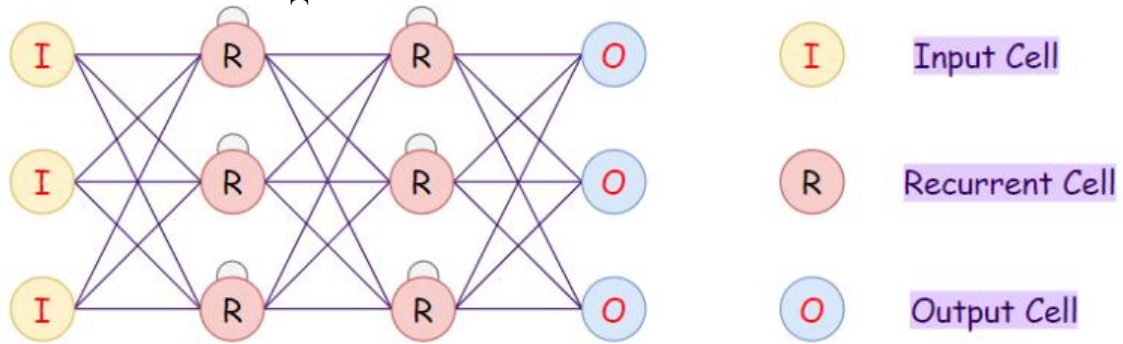


Рисунок 1.8 – Рекурентна нейронна мережа

Радіальна базова мережа – це фактично нейронна мережа з прямим зв'язком, які використовують радіальну базисну функцію як функцію активації замість логістичної функції. Яка різниця? Логістична функція відображає деяке довільне значення в діапазоні $0 \dots 1$, відповідаючи на запитання так чи ні. Навпаки, радіальні базисні функції відповідають на питання «Як далеко ми від цілі?»

Ланцюг Маркова — це математична система, яка переживає перехід з одного стану в інший на основі деяких імовірнісних правил. Імовірність переходу в будь-який конкретний стан залежить виключно від поточного стану та часу, що минув. На рис. 1.9 зображено таку мережу.

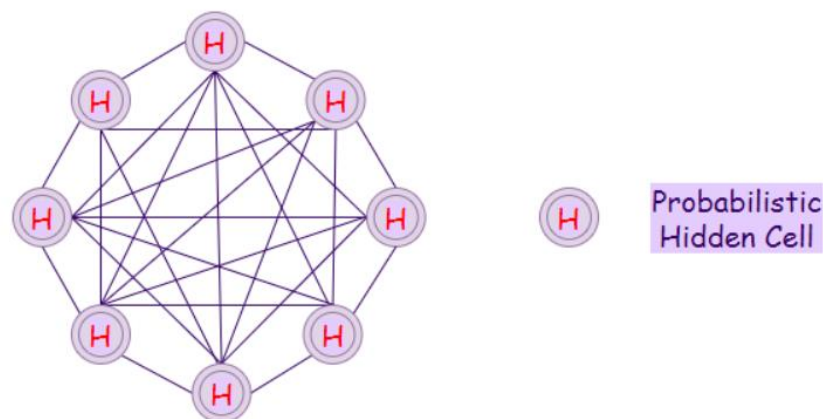


Рисунок 1.9 – Мережа ланцюга Маркова

У нейронній мережі Хопфілда кожен нейрон безпосередньо пов'язаний з іншими нейронами. У цій мережі нейрон увімкнено або вимкнено. Стан нейронів може змінюватися, отримуючи вхідні дані від інших нейронів. Зазвичай ми використовуємо мережі Хопфілда для зберігання шаблонів і пам'яті. Коли ми навчаємо нейронну мережу на наборі шаблонів, вона може розпізнати шаблон навіть, якщо він дещо спотворений або неповний. [10]

Навчання нейромережі можна поділити на навчання з вчителем та без вчителя. Навчання з вчителем полягає у такому: подання вибірки вхідних сигналів нейромережі, отримання вихідних сигналів та порівняння з готовим рішенням. Вчителем необов'язково є людина. Мережа потребує тренувань сотнями та тисячами годин, тому у 99% випадків тренуванням займається комп'ютерна програма. Навчання без вчителя полягає у створенні вибірки вхідних сигналів, але правильних відповідей на виході немає.

Нейронні мережі використовуються для вирішення складних завдань, які вимагають аналітичних обчислень подібних до тих, що робить людський мозок. Найпоширенішими застосуваннями нейронних мереж є класифікація, передбачення та розпізнавання. [11]

1.4 Постановка задачі

Отже, розглянувши особливості функціонування вебзастосунків, необхідно створити свій вебсайт. Він повинен мати клієнт-серверну архітектуру, базу даних та взаємодіяти з нейронною мережею. Інтерфейс програми повинен бути зручним, інтуїтивно зрозумілим, містити цікаві анімації для більшої привабливості користувачів. Кожна сторінка сайту повинна містити меню для навігації між сторінками. Інтерфейс не повинен бути занадто яскравим, містити забагато елементів, що може не дуже сподобатися користувачу.

Програма повинна бути поділена на дві частини: клієнтську та серверну. Перша частина повинна містити інтерфейс програми: картинки, аудіо, відео, стилі, скрипти та HTML. За допомогою CSS можна додати стилі для анімації та красивого відображення сторінки. Бібліотека React допоможе створити гнучкий інтерфейс та зменшить витрати часу на розробку сайту з його можливостями.

Серверна частина буде створена на програмній платформі Node.js. Вона повинна взаємодіяти з базою даних, відправляючи запити та отримуючи відповіді для подальшої обробки. Можливості функціональної мови програмування JS у середовищі Node.js допоможуть створювати запити та відправляти їх до бази даних, приймати відповіді, обробляти запити з клієнтської частини програми, встановлювати з'єднання з базою даних тощо.

У якості системи керування базою даних було обрано MongoDB, яка набула популярності останнім часом, завдяки простоті.

І звичайно, необхідно реалізувати нейронну мережу, яка допоможе користувачу обрати для себе підходящу країну для відпочинку. Нейронна мережа повинна проаналізувати масив результатів тестування та навчитися прогнозувати вподобання користувача на основі його відповідей на питання тесту.

Висновки до розділу 1

У цьому розділі було проведено аналізування актуальності застосування штучного інтелекту. Зважаючи на наведені факти, можна зробити висновок про ефективність використання нейронних мереж сучасними вебсайтами. Нейромережі допомагають суттєво покращити якість обслуговування: аналізувати поведінку відвідувачів, формувати список найбільш підходящих товарів, створювати опис товарів на основі проаналізованих зображень, консультувати, сортувати наявні товари. Усе це допомагає зменшити

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
навантаження на працівників, підвищити об'єм продажів та зекономити
бюджет компаній.

Було розглянуто основні поняття та принципи роботи веб-застосунків. З'ясовано, що веб-сайти – програми, які побудовані за клієнт-серверною архітектурою, яка поділяє програму на дві частини: клієнтську та серверну. Клієнтська частина відповідає за інтерфейс – картинки, текст, відео, аудіо, анімацію тощо. Серверна ж частина повинна забезпечити обробку запитів та зберігання даних у базі даних.

Клієнт-серверна архітектура – архітектура, де клієнт (браузер) – посилає запити на сервер для відображення якогось інтернет-ресурсу. Веб-сервер повинен забезпечувати обробку таких запитів за допомогою бази даних та відправляти відповіді. Браузер аналізує цю відповідь та відображає дані відповіді.

IP-адреса – адреса вузла в мережі інтернет. Завдяки їй, можна однозначно ідентифікувати пристрій в мережі для передачі даних. Однак, ці адреси є не дуже комфортними у використанні. Тому, існує спеціальна служба DNS, яка ставить кожній IP-адресі у відповідність доменні ім'я, яке є більш зручним для запам'ятовування.

Нейронні мережі – розділ штучного інтелекту, який займається «навчанням» машин роботі, яку робить людський мозок. Одиниці нейронної мережі є прототипом людської нервової клітини нейрона. Вони складаються з входів, виходів, ядра для обчислень та застосовують різні функції активації для обчислення результату. Штучні нейрони нейронної мережі розташовані шарами. Найпростішим типом є одношаровий персептрон, який складається лише з одного штучного нейрона. Існують також інші типи нейронних мереж: з прямим зв'язком, прямої передачі, рекурентні, радіальні тощо. Вони можуть відрізнитися структурою, призначенням, функцією активації.

Нейронні мережі використовуються для вирішення складних задач, які може вирішувати мозок людини. Вони застосовуються для розпізнавання,

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
класифікації, передбачення наступного кроку. Розрізняють навчання з вчителем та без вчителя. Вони відрізняються наявністю готових результатів, з якими і порівнюються результати, отримані для певної вибірки вхідних даних. Навчання найчастіше проводить комп'ютерна програма, так як необхідно обробити велику кількість даних.

Існує безліч правил, за якими можна навчати нейронні мережі. Основними є чотири правила: правило корекції за помилками, правило навчання Больцмана, дельта правило і навчання методом змагань. Усі інші правила є модифікованими версіями цих основних правил.

У житті людини нейронні мережі допомагають в таких сферах: економіка, медицина, бізнес, охорона, сільське господарство, мистецтво, прогноз погоди тощо. Усі вони застосовують різні види нейронних мереж, які можуть відрізнитися структурою, функцією активації, способом навчання тощо.

2 ІНСТРУМЕНТИ, МЕТОДИ ТА СЕРЕДОВИЩА ДЛЯ СТВОРЕННЯ ВЕБСАЙТІВ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Функціональна мова програмування JS

Неможливо уявити сучасну веб-сторінку без застосування такої мови програмування, як JavaScript. Перші веб-сторінки були дуже простими. Вони містили лише текст та деякі стилі. З часом вони ставали все більш складними та вимагали більшої взаємодії з користувачем. Так з'явилася функціональна мова програмування JavaScript, яка забезпечує цю інтерактивність.

JS існує з середини 90-х років. Перша версія цієї важливої мови була створена лише за 10 днів Бренданом Айхом. Всі сучасні веб-браузери можуть виконувати JS, додаючи складність та логіку програмування до веб-дизайну, що було неможливим з простим HTML. [12]

Мова виконується завдяки спеціальній програмі, яка може аналізувати та виконувати код JS – інтерпретатор JS. У всі сучасні браузери вже вбудовані такі інтерпретатори, тому вони мають змогу виконувати JS-код. Різні веб-браузери мають різні інтерпретатори: V8 для Chrome, Opera та Edge, SpiderMonkey для FireFox тощо. Існують й інші менш відомі інтерпретатори: Chakra, JavaScriptCore, Nitro тощо. [13]

Інтерпретатори працюють складно, але основи їх функціонування можна зрозуміти:

- інтерпретатор зчитує текст скрипта;
- потім він перетворює скрипт в машинний код;
- після цього машинний код запускається та працює досить швидко.

Інтерпретатор JS застосовує різні методи оптимізації на кожному з етапів. Він навіть може продивитися скомпільований код під час його роботи, аналізуючи дані, які проходять через нього. На основі цього аналізу, він може застосувати оптимізацію до машинного коду. В такому разі, скрипти працюють досить швидко.

JS – мова, яка не має доступу до пам'яті чи процесору, бо спочатку вона створювалася для браузерів, які не потребують цього. Саме тому її називають «безпечною» мовою програмування.

Існує безліч оточень, в яких може виконуватись JS: браузер, Node.js тощо. Функції та можливості цієї мови залежать від того, в якому з оточень вона працює. Наприклад, можливості JS в браузері:

- додавати новий HTML-код на веб-сторінку, змінювати наявний контент, модифікувати стилі;
- реагувати на дії користувача: клацання миші, натискання клавіш, зміна вказівника тощо;
- робити запити на віддалений сервер, завантажувати файли такими технологіями, як AJAX та COMET;
- отримувати та встановлювати куки, ставити питання користувачу, відображати повідомлення;
- зберігати дані на стороні клієнта.

Проте, веб-браузер має деякі обмеження для виконання коду JS:

- так як JS не має доступу до системних функцій операційної системи, то вона не може читати/записувати файли на жорсткий диск, копіювати їх чи запускати програми;
- різні вікна, вкладки не знають один про одного. Іноді одне вікно, використовуючи JavaScript, відкриває інше вікно. Але навіть у цьому випадку JavaScript з однієї сторінки не має доступу до іншої, якщо вони прийшли з різних сайтів (з іншого домену, протоколу чи порту). Це називається Same Origin Policy. Щоб обійти це обмеження, обидві сторінки повинні погодитися з цим і містити JavaScript-код, який спеціально обмінюється даними;
- ще одним обмеженням безпеки є нездатність мови JS отримувати дані з інших сайтів/доменів. І хоча це можливо, для цього потрібна явна згода з віддаленою стороною.

Існує, як мінімум, три сильні сторони мови JS:

- повністю взаємодіє з HTML/CSS;
- простота у використанні;
- підтримка всіма основними браузерами, на яких працює за замовчуванням.

Але можливості мови JS підходять не для всіх. Різні проекти потребують різних методів, інструментів та способів розробки програмного продукту. Саме тому останнім часом з'явилося багато нових мов програмування зі своїми особливостями для полегшення створення програм. Такі мови здатні транспілюватися (конвертуватися) в JS, перед тим як виконуватися в браузері. Наприклад, такі:

- CoffeeScript має більш короткий синтаксис, який дозволяє писати чистий та лаконічний код.
- TypeScript додає типізацію для розробки більш складних проектів.
- Flow також додає типізацію, але інакше.
- Dart – особливий випадок. Він має свій інтерпретатор, який працює поза браузером. Спочатку був запропонований Google для заміни JS, але на даний момент він все ще потребує транспіляція, як і іншим мовам.
- Brython транспілює Python в JS, що дозволяє писати застосунки на чистому Python.

Існують й інші мови, які транспілюються в JS. В будь-якому випадку, для того, щоб використовувати хоча б одну з цих мов програмування, потрібно розуміти JS. Сучасні інструменти роблять процес транспіляції дуже швидким, дозволяючи розробнику писати на іншій мові програмування не задумуючись над подальшими перетвореннями цього коду перед виконанням.

2.2 HTML – основа вебсторінок

HTML – мова, якою написано більшість веб-сайтів. Вона використовується для створення сторінок та надання їм функціональності.

HTML вперше створили Тім Бернерс-Лі, Роберт Кайо та інші, починаючи з 1989 року. Це означає Hyper Text Markup Language. Гіпертекст означає, що документ містить посилання, які дозволяють читачеві перейти до інших місць документа або взагалі до іншого документа. Остання версія відома як HTML5. Мова розмітки — це спосіб, за допомогою якого комп'ютери спілкуються один з одним, щоб контролювати, як текст обробляється та представляється. Для цього HTML використовує дві речі: теги та атрибути. Вони є основою HTML. Вони працюють разом, але виконують різні функції. [14]

Теги – основні одиниці, з яких складається HTML-документ. Вони вкладаються в кутові дужки, наприклад «<p>». Теги бувають ті, які треба закривати (парні), і ті, які не потрібно закривати (непарні). Парний тег має вигляд «<p></p>». Між відкриваючим і закриваючим тегом може бути текст. Непарний тег має вигляд «».

Також теги поділяють на блокові та рядкові. Усю інформацію про елементи та їх характеристику можна знайти у офіційній документації. Тут наведено лише декілька прикладів блокових та рядкових тегів.

Блокові елементи: <html>, <div>, <body>, <p>, <h1>-<h6>, , .

Рядкові елементи: <a>, ,
, <code>, <input>.

Кожна веб-сторінка повинна мати обов'язкову структуру, яка зображена на рис. 2.1.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Рисунок 2.1 – Базова структура HTML

Кожна сторінка повинна містити таку базову структуру, яка зображена на рис. 2.1. Елемент «<!DOCTYPE html>» визначає тип документа. Тег «<html>» є кореневим, так як містить в собі усі інші елементи. «<head>» визначає конфігурацію документа: заголовок, кодування, масштабування тощо. Елемент «<body>» містить у собі структуру документа: тексти, картинки, відео, аудіо тощо.

Існує безліч редакторів коду, які представляють свої інструменти та методи для створення веб-сторінок: WebStorm, Visual Studio Code, Sublime Text тощо.

Блокові формують видимий блок на сторінці. Вони будуть на новому рядку після будь-якого контенту, і будь-який контент після них теж буде на новому рядку. Блокові елементи не можуть бути вкладені в рядкові елементи, але іноді можуть вкладатися в інші елементи блокового рівня. Вони займають весь доступний простір по горизонталі.

Рядкові елементи – це ті, які знаходяться в блокових елементах та оточують лише малі частини документа. Вони не призводять до появи нового рядка у документі. Блокові елементи мають таку структуру, яка зображена на рис. 2.2.

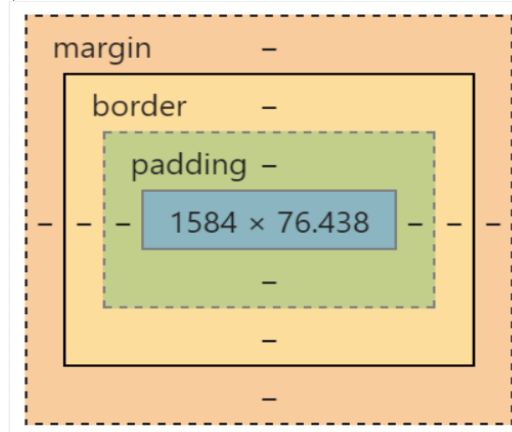


Рисунок 2.2 – Структура блокового елемента HTML

На рис. 2.2 зображено такі частини блокового елемента:

- margin – зовнішній відступ;
- border – рамка елемента;
- padding – внутрішній відступ; [15]

Синім позначено контент елемента. Кожна частина налаштовується окремо. Margin, border, padding мають верхню, нижню, праву та ліву границі.

Атрибути містяться всередині тегу, та визначають додаткову інформацію про тег.

2.3 Таблиця стилів CSS

Cascading Style Sheets (CSS) – каскадна таблиця стилів, яка використовується для представлення зовнішнього вигляду документа, написаного на HTML або XML. Якщо HTML призначався для опису вмісту веб-сторінки, то CSS – для стильового форматування цієї сторінки. Якби HTML-документ містив будову веб-сторінки та ще й колір, розмір шрифтів, відступів тощо, то це був би величезний документ, який дуже важко читати та підтримувати. Тому Консорціум World Wide Web Consortium (W3C) створив CSS. Стили зазвичай зберігають у зовнішніх файлах з розширенням «.css».

Завдяки такому зовнішньому файлу можна легко змінювати форматування сайту, змінивши лише один або декілька файлів. Це значно спрощує та прискорює створення веб-сайтів. [16]

CSS складається з правил. Правило складається з селектора та блоку оголошення. Синтаксис наведено на рис. 2.3.

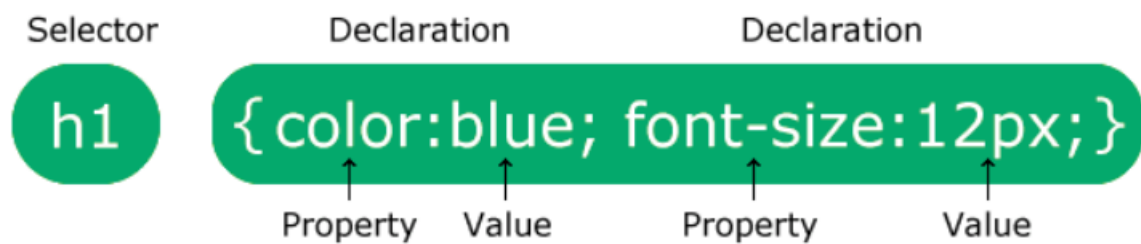


Рисунок 2.3 – Синтаксис CSS

Селектор вказує на елемент HTML, який потрібно створити. Блок декларації містить одну або кілька декларацій, розділених крапкою з комою. Кожна декларація містить назву властивості CSS і значення, розділені двокрапкою. Кілька декларацій CSS розділяються крапкою з комою, а блоки декларацій оточені фігурними дужками. [17]

На рис. 2.4 наведено приклад використання CSS-правила.

```
p {
  color: red;
  text-align: center;
}
```

Рисунок 2.4 – CSS-правило

- «p» - це селектор, який вказує на елемент, який ми хочемо стилізувати;
- «color» - властивість, а «red» - її значення;
- «text-align» - властивість, а «center» - її значення.

Селектори бувають декількох типів:

- прості селектори;
- селектори комбінаторів;
- селектори псевдокласів;
- селектори псевдоелементів;
- селектори атрибутів.

Прості селектори обирають елементи на основі назви, ідентифікатора або класу. На рис. 2.5 – 2.7 наведено приклади таких селекторів.

```
p {  
  text-align: center;  
  color: red;  
}
```

Рисунок 2.5 – Селектор, який обирає елемент на основі назви

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

Рисунок 2.6 – Селектор, який обирає елемент зі значенням id para1

```
.center {  
  text-align: center;  
  color: red;  
}
```

Рисунок 2.7 – Селектор класу

Комбінатор - це те, що пояснює зв'язок між селекторами. Селектор CSS може містити більше одного простого селектора. Між простими селекторами

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
ми можемо включити комбінатор. Розрізняють такі комбінатори: нащадків (пробіл), дочірній («>»), сусідній («+»), братів і сестер («~»).

Комбінатор нащадків відповідає всім елементам, які є нащадками зазначеного елемента. Дочірній комбінатор обирає усі елементи, які є дочірніми для зазначеного елемента. Сусідній комбінатор використовується для вибору елемента, який знаходиться безпосередньо після іншого певного елемента. Ці два елемента повинні знаходитися всередині одного батьківського елемента. Загальний комбінатор братів і сестер обирає усі елементи, які є наступними братами і сестрами зазначеного елемента. На рис. 2.8 – 2.11 наведено приклади використання комбінаторів.

```
div p {
  background-color: yellow;
}
```

Рисунок 2.8 – Комбінатор нащадків

```
div > p {
  background-color: yellow;
}
```

Рисунок 2.9 – Комбінатор дітей

```
div + p {
  background-color: yellow;
}
```

Рисунок 2.10 – Комбінатор сусідніх братів і сестер

```
div ~ p {
  background-color: yellow;
}
```

Рисунок 2.11 – Загальний комбінатор братів та сестер

Псевдоклас – це ключове слово, яке додається до селектора та встановлює його особливий стан. Наприклад, «:hover» може бути використан для зміни кольору кнопки при наведенні вказівника на неї. На рис. 2.12 наведено приклад такого псевдокласу.

```
div:hover {  
    background-color: #F89B4D;  
}
```

Рисунок 2.12 – Псевдоклас «:hover»

Псевдоелемент CSS використовується для стилізації певних частин елемента. З їх допомогою можна стилізувати першу літеру або рядок елемента, вставити вміст до або після вмісту елемента тощо. На рис. 2.13 наведено приклад псевдоелементу.

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

Рисунок 2.13 – Псевдоелемент «::first-line»

На рис. 2.13 зображено псевдоелемент, який використовується для додавання спеціального стилю до першого рядка тексту. Зверніть увагу на позначення подвійної двокрапки. Подвійна двокрапка замінила позначення з однією двокрапкою для псевдоелементів у CSS3. Це була спроба W3C розрізнити псевдокласи та псевдоелементи.

Селектор атрибуту використовується для виділення елементів із заданим атрибутом. На наступному рис. 2.14 наведено приклад такого селектора.

```
a[target] {  
    background-color: yellow;  
}
```

Рисунок 2.14 – Селектор, який обирає всі «<a>» з атрибутом target

Також існує універсальний селектор CSS, який впливає на всі елементи документу без винятку. Він показаний на рис. 2.15.

```
* {  
    text-align: center;  
    color: blue;  
}
```

Рисунок 2.15 – Універсальний селектор

Іноколи буває так, що необхідно застосувати одні і ті ж стилі для декількох елементів. Для уникнення дублювання коду використовується групування селекторів. На рис. 2.16 наведено приклад такого групування.

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

Рисунок 2.16 – Групування селекторів

Коли браузер прочитає таблицю стилів, він відформатує HTML-документ відповідно до інформації в таблиці стилів. Існує три способи вставки таблиці стилів: зовнішній, внутрішній, вбудований.

Зовнішній CSS зберігається в окремому файлі, внутрішній – в тегу «<style>», який знаходиться всередині «<head>». Приклад внутрішнього стилю зображено на рис. 2.17.

```
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
```

Рисунок 2.17 – Приклад застосування внутрішнього стилю

Вбудований стиль може використовуватися для застосування унікального стилю для окремого елемента. Щоб його застосувати, необхідно додати атрибут `style` до відповідного елемента. Приклад застосування вбудованого стилю зображено на рис. 2.18.

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Рисунок 2.18 – Приклад застосування вбудованого стилю

Вбудований стиль необхідно застосовувати обережно, адже він руйнує ідею розділення на стилі та вміст сторінки, змішуючи їх в одному файлі. У випадку, якщо є декілька варіантів стилів, CSS-стилі мають таку ієрархію пріоритетів:

- 1) Вбудований стиль.
- 2) Зовнішні та внутрішні таблиці стилів.
- 3) Стилi браузера за замовчуванням.

Тобто, стилі з найбільшим пріоритетом будуть застосовані до визначеного елемента. Таким чином, вбудований стиль має найвищий

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
пріоритет і замінить зовнішні та внутрішні стилі та параметри браузера за
замовчуванням.

Специфіка – ранг/оцінка, що визначає, яке оголошення стилю в кінцевому підсумку буде застосовано до елемента. Якщо є два або більше правил CSS, які вказують на один і той самий елемент, селектор з найвищим значенням специфічності буде мати пріоритет, а його стилі будуть застосовані до цього елемента HTML. Кожен селектор CSS займає своє місце в ієрархії специфіки. Існує чотири категорії, які визначають рівень специфічності селектора:

- вбудовані стилі (значення специфічності – 1 000);
- ідентифікатори (значення специфічності - 100);
- класи, псевдокласи, селектори атрибутів (значення специфічності - 10);
- елементи та псевдоелементи (значення специфічності - 1).

Для розрахунку специфічності конкретного правила, потрібно знайти суму специфічності усіх селекторів цього правила.

2.4 Node.js

Node.js – безкоштовне серверне середовище з відкритим вихідним кодом. Воно може працювати на багатьох платформах, таких як Windows, Linux, Unix, Mac OS тощо.

Одним з найпоширеніших завдань сервера може бути відкриття файлу та повернення вмісту клієнту. Node.js використовує асинхронне програмування. Ось як PHP або ASP обробляє запит файлу:

- 1) Відправляє завдання до файлової системи комп'ютера.
- 2) Чекає, поки файлова система відкриється, і прочитає файл.
- 3) Повертає вміст клієнту.
- 4) Готовий обробити наступний запит.

І як Node.js обробляє запит на файл:

- 1) Відправляє завдання до файлової системи комп'ютера.
- 2) Готовий обробити наступний запит.
- 3) Коли файлова система відкриє і прочитає файл, сервер повертає вміст клієнту. [18]

Node.js усуває очікування і просто продовжує виконувати наступний запит. Воно запускає однопотокове, неблокуюче, асинхронне програмування, що дуже економить пам'ять.

Можливості Node.js:

- може генерувати динамічний вміст веб-сторінок;
- може створювати, відкривати, читати, записувати, видаляти та закривати файли на сервері;
- може збирати дані форми;
- може додавати, видаляти, змінювати дані у вашій базі даних. [19]

Файли Node.js містять завдання, які будуть виконуватися під час певних подій. Типова подія — це спроба отримати доступ до порту на сервері. Файли Node.js мають розширення «.js».

Node.js має модулі. Один модуль – це один файл JS. Модулі можна розглядати як бібліотеки, які можна підключати та додавати функціонал. Node.js має набір вбудованих модулів, які можна використовувати без попередньої інсталяції. Щоб включити модуль, використовується функція «require()» з назвою модуля. На рис. 2.19 наведено приклад підключення модуля Http, який є основою для обробки мережових запитів.

```
var http = require('http');
```

Рисунок 2.19 – Приклад підключення модуля

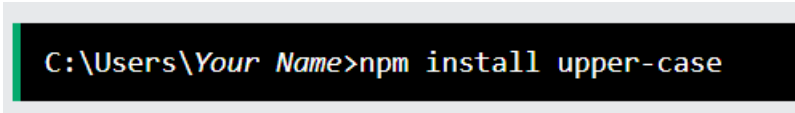
Також є можливість створювати власні модулі та легко підключати їх до своїх програм. Використовуйте ключове слово «exports», аби зробити властивості та методи доступними за межами файлу модуля. На рис. 2.20 показано приклад створення такого модуля, який може бути без проблем підключений до застосунка.

```
exports.myDateTime = function () {  
    return Date();  
};
```

Рисунок 2.20 – Приклад створення модуля

Після створення модуля, його можна підключити до програми за допомогою функції «require()», яка повертає посилання на об'єкт, який містить усі експортовані функції та змінні.

Для керування своїми модулями Node.js використовує менеджер модулів NPM. Ця програма автоматично встановлюється на вашому комп'ютері, коли ви встановлюєте Node.js. Перед тим як почати використовувати невбудовані модулі, їх потрібно встановити. Це і робить NPM, який може встановлювати, видаляти модулі. Робиться це за допомогою командного рядка та інструкції «npm install». На рис. 2.21 наведено приклад завантаження модуля «upper-case».



```
C:\Users\Your Name>npm install upper-case
```

Рисунок 2.21 – Встановлення модуля «upper-case»

Після того, як було завантажено необхідний модуль, його можна підключити до програми за допомогою функції «require()», як звичайний модуль.

Існує дуже багато модулів, які виконують різні функції. В офіційній документації є повна інформація про кожного з них. Тут буде наведено лише деякі з них:

- «http» дозволяє передавати дані через протокол передачі гіпертексту – НТТР.
- «url» розбиває веб-адресу на частини, які можна читати.
- «fs» дозволяє користуватись файловою системою комп'ютера (читання/створення/оновлення/видалення/перейменування файлів).
- «formidable» дозволяє користувачеві завантажувати файли на комп'ютер.
- «events» призначений для створення, прослуховування та запуску власних подій.
- «nodemailer» дозволяє легко надсилати електронні листи з вашого комп'ютера.

2.5 MongoDB

MongoDB реалізує новий підхід для створення баз даних, де немає таблиць, на відміну від класичних реляційних баз даних. Раніше, до появи інших видів баз даних, усі дані зберігалися в реляційних базах даних. Але з появою нового покоління баз даних, розробники можуть обирати, яка система керування базами даних більше підходить для зберігання даних. Звичайно, адже задачі різні, і підходи та інструменти можуть відрізнитися для вирішення того чи іншого завдання. [20]

Уся система MongoDB може представляти не тільки одну базу даних, яка знаходиться на одному фізичному сервері. Вона дозволяє зберігати декілька різних баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними та зберігати цілісність.

JSON є одним з найпопулярніших стандартів обміну та зберігання даних. Він добре підходить для опису складних за структурою даних. MongoDB використовує схожий спосіб зберігання даних, хоча формально JSON не використовується. MongoDB має спеціальний формат для опису даних, який називається BSON. З таким форматом швидше виконується пошук та обробка даних. Якщо порівнювати BSON та JSON, незважаючи на те, що дані в форматі BSON займають більше місця, ніж в форматі JSON, цей недолік повністю перекривається швидкістю.

MongoDB написана на мові програмування C++, тому може легко бути встановлена на різні платформи: Windows, Unix, Mac OS. Також можна завантажити вихідний код та самому скомпілювати, але рекомендується користуватися програмами з офіційного сайту.

Порівнюючи MongoDB та реляційні бази даних, можна провести аналогію: замість рядків, MongoDB використовує документи, а замість таблиць – колекції. Документи схожі на звичайні JS-об'єкти. Вони можуть зберігати більш складні дані, та представляють собою набір ключів та їх значень.

Ключ представляє собою назву деякої частини інформації.

Однак, існує деяка схожість між MongoDB та класичними реляційними базами даних. Ця схожість полягає у наявності первинного ключа, який представлений деяким стовбцем у реляційних базах даних, який має тільки унікальні значення. А MongoDB має для кожного документа унікальний ідентифікатор, який називається `_id`. Якщо не вказати його явним способом, система автоматично згенерує для нього значення. Це буде спеціальне бінарне значення розміром 12 байт. Це значення складається з декількох частин: значення типу Timestamp розміром 4 байти, випадкове число з 5-ти байт і рахівник з 3 байт, який ініціалізується випадковим числом. Поле `_id` повинно мати унікальне значення в межах колекції. Якщо ми спробуємо додати в

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

колекцію два документи з однаковим ідентифікатором, то додається тільки один з них, а при додаванні іншого ми отримуємо помилку.

Ще одна різниця між MongoDB та реляційними базами даних полягає у тому, що, якщо в реляційній базі даних немає значення для якогось з полів, то система може призначити NULL. В MongoDB все інакше. Якщо якийсь ключ не має значення, він просто опускається та не зберігається в документі.

Якщо в традиційних реляційних SQL є таблиці, то у світі MongoDB є колекції. Такі таблиці можуть зберігати лише однотипні, жорстко структуровані дані, тоді як колекції – більш складні структури даних, які можуть мати різний набір властивостей.

MongoDB має унікальну систему зберігання даних, яка представляє собою набір реплік. Цей набір має основний вузол, а також може бути набір вторинних вузлів. Останні зберігають цілісність та автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол чомусь виходить з ладу, то один з вторинних вузлів стає головним.

Завдяки тому, що MongoDB не має жорсткої структури даних, нові дані можна легко створювати та зберігати з даними іншої будови, без необхідності перероблювати усю наявну базу даних. Це значно скорочує витрати часу на створення та підтримку баз даних.

Одна з головних проблем зберігання даних – зберігання великих об'ємів даних. Такі дані можна розділити та зберігати у різних файлах. Не зважаючи на те, що MongoDB може зберігати різні за структурою дані, розмір документа обмежується 16-ю мб. Але MongoDB має спеціальну технологію GridFS, яка дозволяє зберігати дані, які більше, ніж 16 мб. Ця система складається з двох колекцій. В першій, яка називається files, зберігаються імена файлів, а також їх метадані, наприклад, розмір. А у другій колекції, яка називається chunks, у вигляді невеликих сегментів зберігаються дані файлів, зазвичай сегментами по 256 кб. Для тестування GridFS можна використати спеціальну утиліту mongofiles, яка йде в пакеті mongodb.

Кожна колекція має своє унікальне ім'я – ідентифікатор, який складається з максимум 128-ми різних алфавітно-цифрових символів та знаку нижнього підкреслення. Такі колекції, на відміну від таблиць реляційних баз даних, не мають заданої кількості стовпців. Центральним поняттям бази даних MongoDB є документ. Його можна уявити як об'єкт, який зберігає деяку інформацію у вигляді набору пар ключ-значення. На рис. 2.22 зображено приклад такого документа.

```
{  
  "name": "Tom",  
  "surname": "Smith",  
  "age": "37",  
  "company": {  
    "name" : "Microsoft",  
    "salary" : "100"  
  }  
}
```

Рисунок 2.22 – Представлення документа

Ключі являють собою рядки. Значення ж можуть бути різних типів. В даному випадку, у нас майже всі значення представляють собою рядки, і лише один ключ (company) посилається на окремий об'єкт. Всього є наступні типи значень:

- String – строковий тип даних, який використовує кодування UTF-8;
- Array – тип даних для зберігання масивів елементів;
- Binary data – тип для зберігання даних в бінарному форматі;
- Boolean – тип даних, який зберігає логічне true чи false;
- Date зберігає дату в форматі часу Unix;
- Double – числовий тип даних для зберігання чисел з плаваючою точкою;

- Integer використовується для зберігання цілісних значень розміром 32 біта;
- Long використовується для зберігання цілісних значень розміром 64 біта;
- JavaScript – тип даних для зберігання коду JS;
- Min key/Max key використовується для порівняння значень з найменшим/найбільшим елементом BSON;
- Null – тип даних для зберігання значення null;
- Object використовується для вбудованих документів;
- ObjectId – тип даних для зберігання id документа;
- Regular expression використовується для зберігання регулярних виразів;

Запити MongoDB мають реєстрозалежність та сувору типізацію.

Існує багато функцій для виконання запитів, основні з них:

- «use » встановлює дану базу даних в якості поточної;
- «db» виводить поточну базу даних;
- «insert()» додає документи;
- «find()» - вибірка документів.

2.6 Бібліотека React

React – це бібліотека JS, яка використовується для створення користувацького інтерфейсу. Перший реліз її відбувся в березні 2013 року. Початкове завдання цієї бібліотеки було створення сайтів, веб-застосунків. Пізніше з'явилася платформа React Native, яка була випущена для створення застосунків на мобільних пристроях. React позиціонувалася як ідеальний інструмент для створення SPA – односторінкових сайтів (Single Page Application). [21]

Вебсторінка для браузера являє собою DOM (Document Object Model), яка являє собою структуру елементів HTML, якими можна маніпулювати: змінювати, видаляти або додавати нові. Аби виконувати такі маніпуляції, ми застосовуємо мову JS. Однак, коли ми намагаємося змінити DOM-елементи, ми можемо зіштовхнутися з такою проблемою, як зниження рівня швидкодії. Адже прямі маніпуляції з DOM-елементами вимагають багато часу. Для вирішення такої проблеми якраз і було створено концепцію віртуального DOM. Цей DOM являє собою полегшену версію звичайного DOM. React працює саме з віртуальним DOM.

При необхідності зміни HTML-елементів вебсторінки, вони вносяться спочатку у віртуальний DOM. Потім новий стан віртуального DOM порівнюється з поточним станом. У разі виникнення розбіжностей, React виконує мінімальну кількість маніпуляцій, які необхідні для оновлення звичайного DOM. Тобто, можна сказати, що React дуже спрощує процес оновлення звичайного DOM. Вона мінімізує кількість операцій, які необхідні для приведення до нового стану звичайний DOM, що значно пришвидшує роботу та полегшує навантаження на комп'ютер.

Основними елементами React-застосунку є елементи. На рис. 2.23 зображено приклад такого елемента.

```
const element = <h1>Hello, world</h1>;
```

Рисунок 2.23 – Приклад React-елемента

Цей на перший вигляд дивний синтаксис має назву JSX – розширення JS. Такий синтаксис дозволяє використовувати HTML-елементи у JS. Він має свої особливості, однією з яких є використання JS-виразів у фігурних дужках. Якщо потрібен рядковий літерал, необхідно застосовувати лапки. Але

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
використання і лапок, і дужок заборонено. Такий інструмент як Babel
компілює JSX у звичайний JS.

На відміну від DOM-елементів, елементи React – це звичайні об’єкти, які не займають багато ресурсів. Для рендерингу React-елементу викликається функція «render()», яка приймає такі параметри: React-елемент та DOM-елемент, у який необхідно записати React-елемент.

Компоненти дозволяють розбити код на частини, які можна групувати та повторно використовувати. Вони бувають двох видів: функціональні та класові. Перші являють собою функцію, а інші – класи. На початку існування React функції були більш простим способом організувати код, так як вони не мали стану. Класові компоненти використовувалися для більш складного коду, який передбачав використання стану компоненти. Наразі React пропонує нововведення, яке називається хуки. Вони дозволяють імітувати у функціональних компонентах стан. Тож, різниця між класовими та функціональними компонентами зникає.

І функціональні, і класові компоненти приймають пропси – дані, які передаються в компоненту. Завдяки ним, можна використовувати одну і ту ж компоненту декілька разів з різними пропсами, що дозволяє значно прискорити процес створення застосунку та легко підтримувати його у подальшому.

Висновки до розділу 2

У даному розділі було розглянуто інструменти, необхідні для створення майбутньої програми: мову програмування, систему керування базами даних, бібліотеку тощо. Можна зробити висновок, що основними інструментами для побудови клієнтської частини (інтерфейсу) сайту є HTML, CSS, JavaScript. Для серверної частини необхідно програмне середовище розробки, мова

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
програмування, а також система керування базами даних для обробки запитів до баз даних.

Для розробки майбутнього вебсайту було обрано таке програмне середовище, як Node.js. Воно може працювати на багатьох операційних системах. У порівнянні з іншими програмними середовищами, Node.js використовує однопотокowe, неблокуюче, асинхронне програмування, що значно пришвидшує роботу сервера.

У якості мови програмування було обрано JavaScript. Це функціональна мова програмування. Її універсальність полягає у використанні і при створенні інтерфейсу, і при розробці сервера. Що полегшує написання коду і виключає необхідність додатково опановувати ще одну мову програмування.

MongoDB – сучасна система керування базами даних. Вона принципово відрізняється від звичайних реляційних систем керування структурою даних та простішим синтаксисом. Саме тому вона була обрана у якості системи керування базою даних для майбутнього застосунку.

На сьогоднішній день відомо дуже багато бібліотек JavaScript. Це вже готовий код, який можна імпортувати та використовувати без необхідності писати самостійно. Це значно полегшує та пришвидшує розробку програм. Кожна з бібліотек написана для вирішення конкретних задач. Прикладом однієї з таких бібліотек є React. Вона використовується для створення односторінкових вебсайтів. Особливістю її є наявність спеціального синтаксису JSX, який допомагає створювати елементи – складові частини React-програми. За рахунок розподілу на такі частини, розробники зможуть швидше писати код, використовуючи одну частину декілька разів. Також буде простіше підтримувати такий код.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ З НЕЙРОННИМИ МЕРЕЖАМИ

3.1 Структура системи

Вебзастосунок повинен бути створений на задану тему. В даному випадку- туристичний сайт. Тобто, у ньому повинна бути інформація про тури, готелі, контакти фірми, відгуки та інша додаткова інформація.

Так як, вебсайти використовують клієнт-серверну архітектуру для створення застосунку, програма буде поділена на дві частини: клієнтську та серверну. Перша частина повинна містити зручний інтерфейс для користувача, а друга – обробляти запити, які надходять з першої частини. Обидві частини будуть використовувати таку мову програмування, як JS.

Вебсайт буде умовно поділятися на декілька сторінок: «Головна», «Каталог», «Контакти», «Відгуки», «Де відпочити?». Кожна з них повинна відповідати назві. Сторінка «Каталог» повинна містити опис наявних та доступних турів з короткою характеристикою. Сторінка «Контакти» повинна мати інформацію про фірму, яка надає туристичні послуги: телефони, адресу та посилання на їх сторінки у соціальних мережах. Головна сторінка повинна містити узагальнюючу інформацію або просто якийсь слоган. Сторінка «Відгуки» має містити інформацію про туристичний тур очима людей, які вже спробували його. Кожна умовна сторінка повинна мати панель навігації між сторінками.

Особливою є сторінка «Де відпочити?», яка містить тест, який побудований за допомогою використання нейронних мереж. Цей тест має допомогти користувачеві з вибором найбільш підходящого місця для відпочинку.

3.2 Проектування туристичного сайту

Майбутній веб-сайт буде зразком SPA. SPA – односторінкові веб-застосунки, які мають лише одну HTML-сторінку, яка завантажується при першому запиті та містить увесь необхідний код: HTML, CSS, JS. Оновлення сторінки ніколи не відбувається. При подальших запитах, браузер отримує дані, які відображає завдяки вже завантаженому коду, прибираючи старий контент єдиної вебсторінки. Саме тому поділ програми на сторінки є умовним. [22] На рис. 3.1 зображено приклад взаємодії клієнта (ноутбука) та сервера.

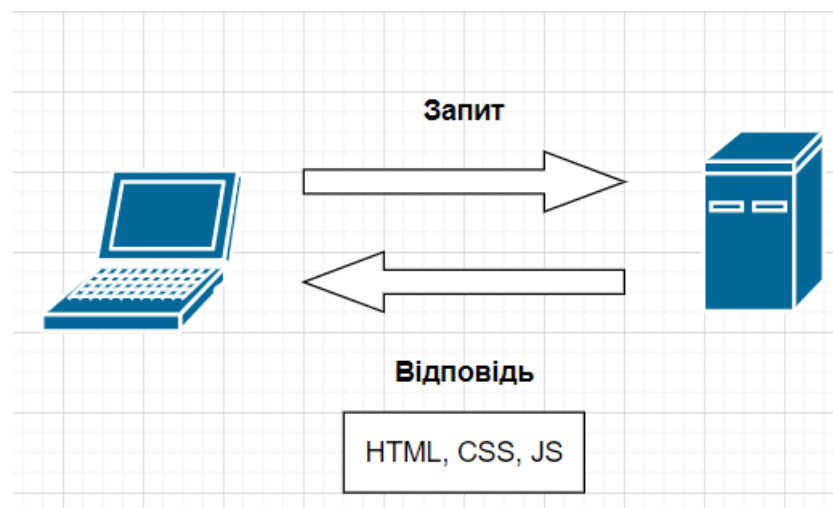


Рисунок 3.1 – Приклад взаємодії двох частин програми

Згідно з протоколом HTTP, клієнт (браузер) створює та відправляє запити на сервер, а сервер обробляє цей запит та посилає відповідь. При першому запиті на сервер, браузер використовує GET-метод, який немає тіла. У відповідь сервер відправляє відповідь, яка має тіло, всередині якого необхідний код. При подальших запитах, сервер відправляє лише дані, а браузер вже містить інформацію для їх правильного відображення. Така технологія має назву Ajax. Вона передбачає оновлення даних на сторінці без перезавантаження її. Це досягається шляхом фонового обміну даними між

браузером та сервером. У результаті вебзастосунки стають швидшими та зручнішими.

Існує багато інструментів, які допомагають створювати SPA. Одним з них є бібліотека React. Саме її я буду використовувати для створення клієнтської частини вебзастосунку. Вона має деякі особливості, які допомагають створювати та підтримувати односторінкові застосунки, роблячи код більш організованим та чистим.

Для розробки сервера буде застосована програмна платформа Node.js. За допомогою вбудованих функцій, можна обробляти запити, працювати з файловою системою тощо. Ще однією з переваг є використання JS, адже це є обов'язковою мовою для написання інтерфейсів. Тобто, не потрібно вчити різні мови. Достатньо знати лише JS, і можна писати як зовнішній вигляд сторінок, так і серверну частину у Node.js.

Під час написання серверу можуть знадобитися додаткові функції, змінні. Для цього підключаються модулі. Найбільш популярним модулем для написання серверної частини є Express. Він дозволяє створювати функції для обробки запитів значно простіше. Також цей модуль дозволяє керувати політикою CORS. Це технологія сучасних браузерів, яка дозволяє надати веб-сторінкам доступ до ресурсів іншого домена.

У якості бази даних було обрано MongoDB. Її було обрано насамперед через простий синтаксис, який подібний до JS. Ще однією перевагою для даного вебзастосунку є формат зберігання файлів, який подібний до звичайних об'єктів JS, які можуть зберігати дані більш складної структури. У колекціях можуть зберігатися документи різної структури, що не може бути в таблицях звичайних реляційних баз даних. За допомогою MongoDB буде створена база даних, яка складається з двох колекцій: у першій будуть зберігатися питання до тесту, а в другій - відгуки.

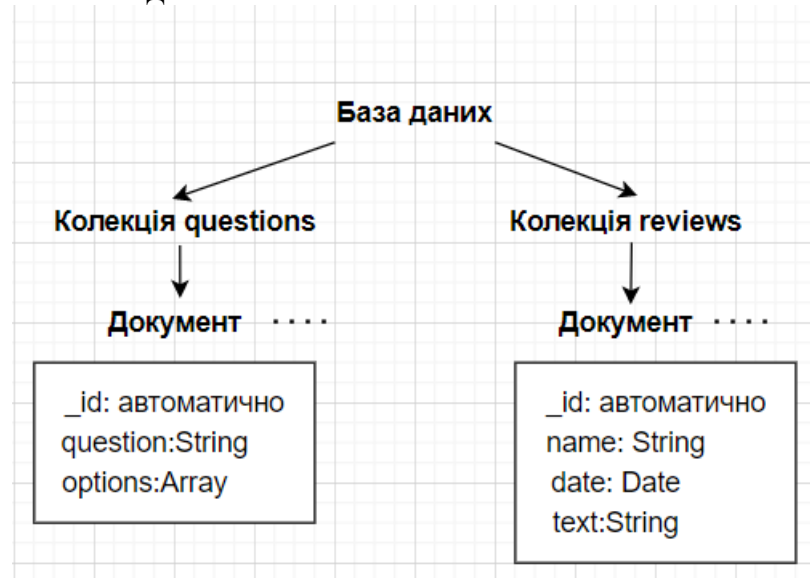


Рисунок 3.2 – Схема бази даних для веб-сайту

У якості редактора коду було обрано Web Storm. Це розумний редактор, який значно спрощує розробку вебзастосунків. Він має вбудовану підтримку різноманітних технологій, в тому числі усіх тих, які необхідні мені для створення вебсайту: JS, CSS, HTML, React, Node.js. Це означає, що не потрібно витрачати час на налаштування, завантаження плагінів, оскільки все вже встановлено, налаштовано та готово для роботи. Цей редактор також має такі інструменти, як автодоповнення коду, аналіз якості коду, рефакторинг, швидкий перегляд документації, попередній перегляд HTML-файлів тощо.

Тестова частина буде базуватися на нейронній мережі. Кількість нейронів вхідного шару дорівнює кількості питань тесту – 12-ти. Кількість вихідних нейронів – 1. Не існує єдиної формули для визначення точної кількості нейронів у прихованому шарі. Різні спеціалісти в області штучного інтелекту пропонують різні методи для визначення їх числа. В даному випадку, буде використано досвід доктора Джеффа Хітона: число прихованих нейронів повинно становити $2/3$ від кількості нейронів вхідного шару, плюс розмір вихідного шару. [23] Таким чином, маємо кількість нейронів прихованого шару – 9.

3.3 Алгоритм зворотного поширення помилки

Навчання нейронних мереж полягає у такому корегуванні ваг зв'язків, після якого можна отримати бажаний результат при заданих вхідних даних. Існує багато різних алгоритмів навчання. У даному застосунку для навчання нейронної мережі буде використано алгоритм зворотного поширення помилки:

1. Визначення результату нейронної мережі у при заданому вектору вхідних значень за допомогою обраної функції активації. Вона повинна бути диференційованою. В даному випадку обираємо логістичну функцію.

2. Визначення локального градієнта для вихідного шару.

3. Корегування ваг.

4. Крок 2-3 повторюються для кожного нейрона прихованого шару.

5. Кінець однієї ітерації.

Кількість ітерацій визначається самостійно. Чим більшою вона буде, тим більш навченою буде нейронна мережа, тобто більш правильні будуть результати. На кожній ітерації будуть різні значення вхідних даних x . Початкові значення ваг обираються випадково. На рис. 3.3 зображено нейронну мережу для першого кроку алгоритму. [24]

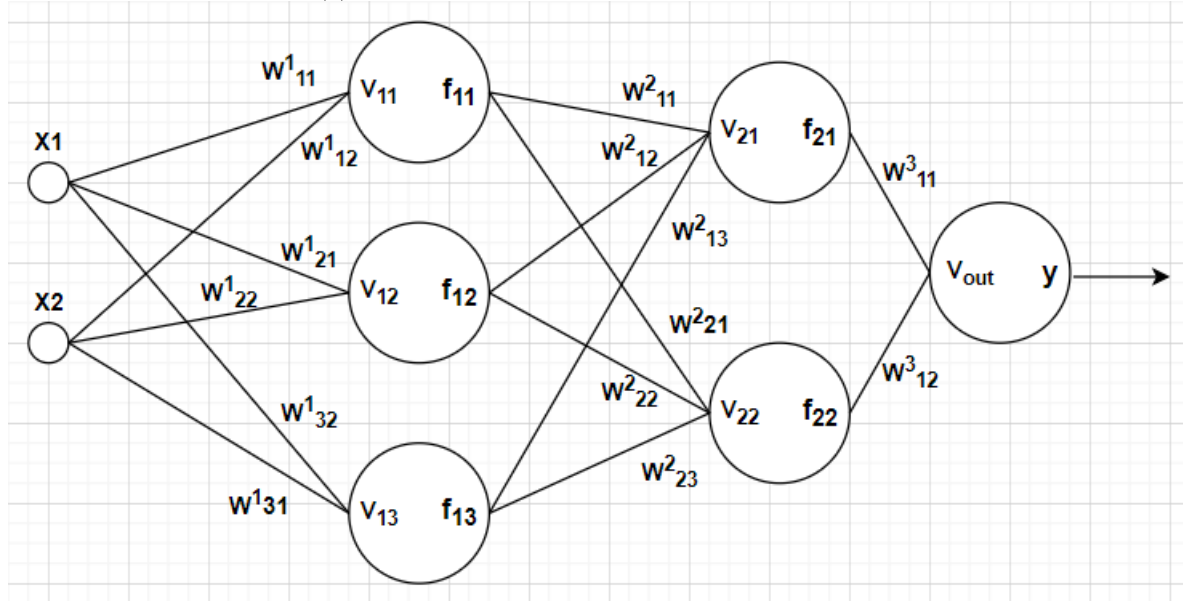


Рисунок 3.3 – Нейронна мережа першого кроку алгоритма зворотнього поширення помилки

(3.1)

де v – стан нейрона;

w – ваги;

x – вхідні дані;

f - функція активації;

y – результат роботи нейронної системи.

Отже, на першому етапі алгоритму було отримано кінцеве значення роботи нейронної мережі y . Наступним кроком є визначення локального градієнта на вихідному шарі. Для цього необхідно спочатку обчислити похибку – різницю між отриманим значенням y та бажаним d :

(3.2)

де e – похибка;

y – отримане значення;

d – очікуване значення.

Після цього можна знайти локальний градієнт:

де δ – локальний градієнт;

e – похибка;

$f(v_{out})$ – функція активації;

v_{out} – кінцевий стан нейронної мережі.

Так як, в якості функції активації було обрано логістичну функцію, то:

де f - логістичної функції;

δ – локальний градієнт;

e – похибка;

v_{out} – кінцевий стан нейронної мережі;

y – отриманий результат роботи нейронної мережі.

Отже, ми отримали локальний градієнт для вихідного шару. Тепер необхідно змінити ваги.

(3.3)

де $w(\text{new})$ – нове значення ваг;
 $w(\text{old})$ – поточне значення ваг;
 λ – крок навчання, обирається самостійно;
 δ – локальний градієнт;
 f – результат функції активації.

Таким чином було скореговано ваги вихідного шару. Далі необхідно визначити локальний градієнт для прихованого шару нейронів:

(3.4)

де δ – локальний градієнт;
 w – ваги;
 f – функція активації.

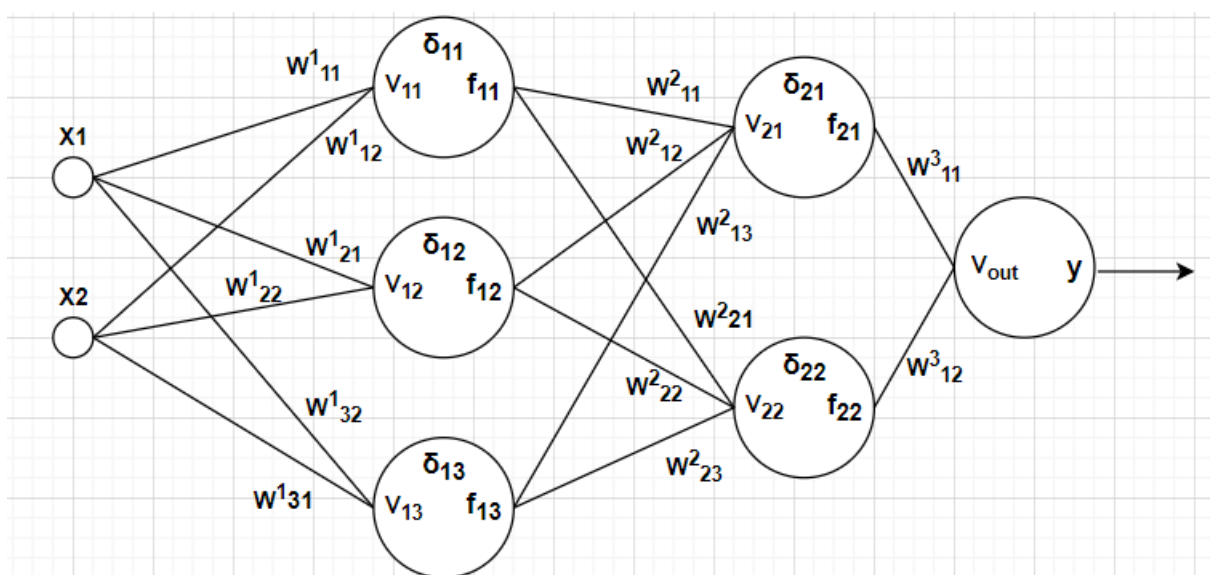


Рисунок 3.4 – Нейронна мережа з локальними градієнтами

Отже, було визначено локальні градієнти для останнього стовпця прихованого шару. Далі змінюємо ваги:

(3.5)

де $w(\text{new})$ – нове значення ваг;

$w(\text{old})$ – поточне значення ваг;

λ – крок навчання;

δ – локальний градієнт;

f – результат функції активації.

Таким чином, вдалося скорегувати ваги прихованого шару. Таким же чином знаходимо локальні градієнти наступних нейронів у зворотному напрямку і коригуємо їх ваги. На даному етапі завершено одну ітерацію.

Функція активації в алгоритмі зворотного розповсюдження помилки повинна володіти декількома важливими характеристиками: безперервністю, диференційністю і бути монотонно незменшною. Більше того, заради ефективності обчислень бажано, щоб її похідна легко знаходилася. Часто активаційна функція також є функцією з насиченням.

Навчання мережі методом зворотного поширення помилки включає три етапи: подачу на вхід даних, з подальшим поширенням даних у напрямку виходів, обчислення і зворотне поширення відповідної помилки і коригування ваг. Після навчання передбачається лише подача на вхід мережі даних та розповсюдження їх у напрямі виходів.

Висновки до 3 розділу

На основі вищенаведених фактів, можна зробити висновок про доцільність використання алгоритму зворотнього поширення помилки для розв'язання поставленої задачі. Він є відносно простим та поширеним.

Майбутній сайт буде прикладом SPA – односторінкового вебзастосунку. За структурою він буде складатися з таких розділів: «Головна», «Каталог», «Контакти», «Відгуки», «Де відпочити?». Кожен з них містить інформацію, яка відповідає заголовкам. Розділ «Де відпочити» містить тест, який допоможе користувачу обрати найбільш підходящу частину світу для подорожі. Він заснований на навченій нейронній мережі.

У якості бази даних було обрано MongoDB, яка вирізняється новим та простим синтаксисом, універсальною структурою для зберігання даних тощо. Майбутня база даних буде складатися з двох колекцій: питання для тесту та відгуки.

Серверна частина буде використовувати програмне середовище Node.js, яке має своє оточення з вбудованими функціями та змінними. Для додаткової зручності можна використати модуль Express, який допомагає з налаштуванням сервера та захистом сервера, наприклад політикою CORS. Така політика дозволяє здійснювати запити на інші домени.

4 ОГЛЯД ГОТОВОГО ПРОГРАМНОГО ЗАСТОСУНКУ

4.1 Інтерфейс програми

Вебсайт має п'ять частин: «Головна», «Відгуки», «Каталог», «Контакти», «Де відпочити?». Кожна з них складається з меню, логотипу фірми та контенту, який відрізняється в залежності від частини. На рисунках 4.1 - 4.5 зображено ці частини.



Рис. 4.1 – Головна сторінка сайту

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ



Рисунок 4.2 – Сторінка «Каталог»

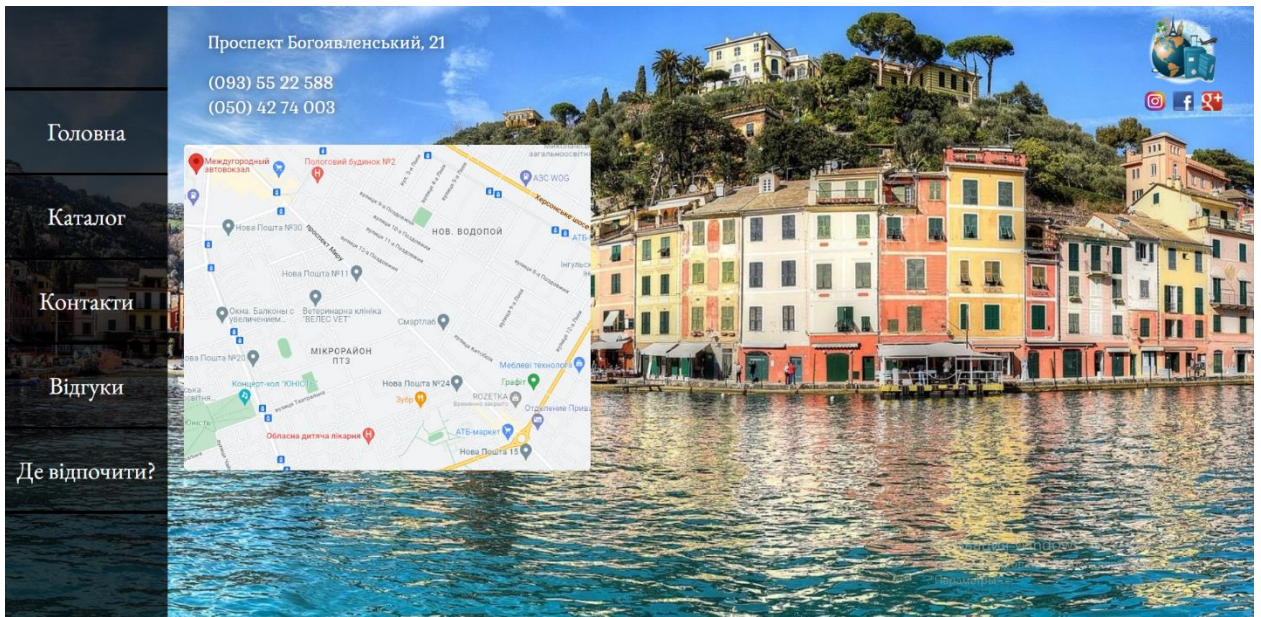


Рисунок 4.3 – Сторінка «Контакти»

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

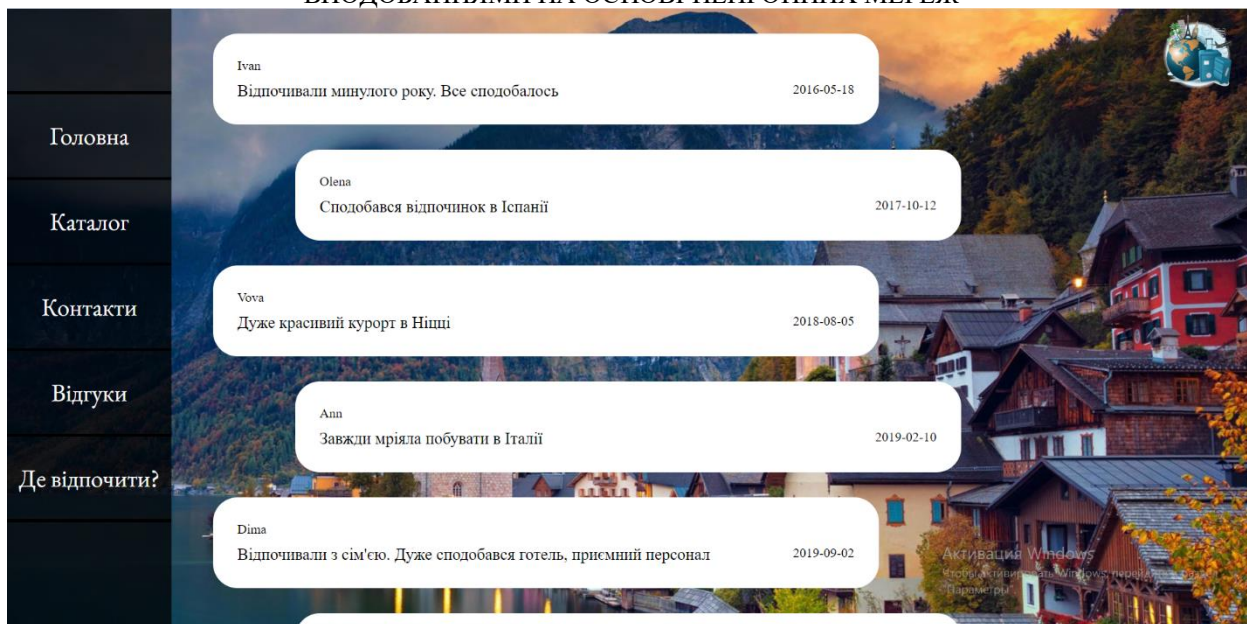


Рисунок 4.4 – Сторінка «Відгуки»

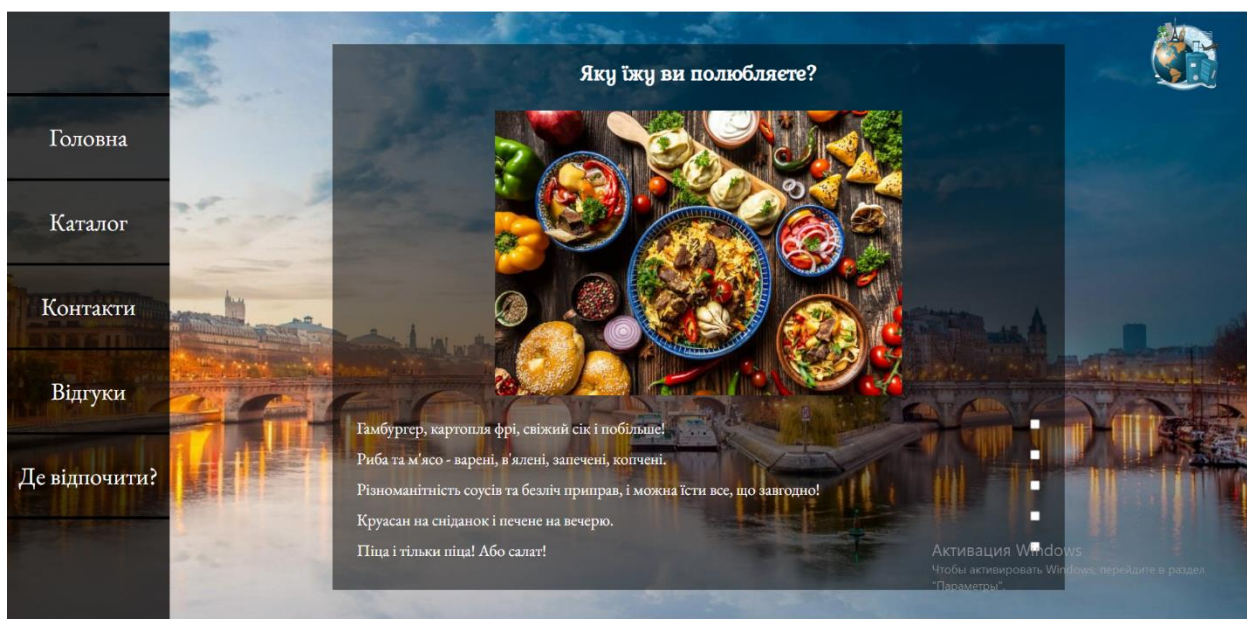


Рисунок 4.5 – Сторінка «Де відпочити?»

На рис. 4.5 зображено перше питання з дванадцяти тесту, який міститься у частині «Де відпочити?». Він базується на використанні нейронних мереж для виявлення найбільш підходящого туристичного туру для конкретного відвідувача.

Кафедра Інтелектуальних інформаційних систем
ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК ПІДБОРУ ТУРИСТИЧНОГО ТУРУ ЗА ІНДИВІДУАЛЬНИМИ
ВПОДОБАННЯМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

На рис. 4.6 зображено результат роботи програми з описом найбільш привабливих для відвідувача країн.

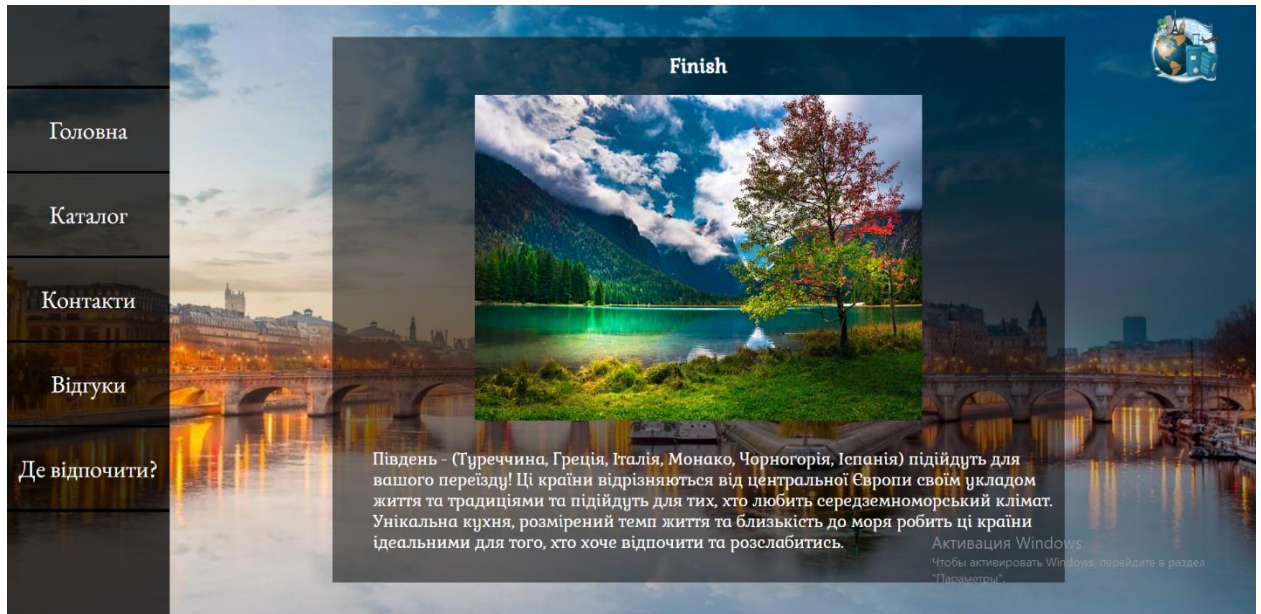


Рисунок 4.6 – Результат виконання тесту

4.2 Серверна частина

Модулі серверної частини показані на рис. 4.7.

```
const express = require('express');
const cors = require('cors');
const { MongoClient } = require("mongodb");
```

Рисунок 4.7 – Модулі серверної частини

Кожен з модулів підключає додатковий функціонал для написання коду. Наприклад, модуль Express відповідає за створення вебсервера, прийняття, обробку запитів та формування відповідей для клієнта (браузера). Модуль Cors відповідає за надання можливості браузеру виконувати запити на інші домени. MongoDB модуль відповідає за підключення до бази даних та виконання запитів до неї.

Для роботи з нейронною мережею є модуль Brain. Також необхідно позначити країни відповідними числами для навчання мережі та отриманні відповіді. На рис. 4.8 зображено відповідний код.

```
var brain = require('brain.js');
var net = new brain.NeuralNetwork();

let country={
  '0': 'Європа ',
  '1': 'Америка',
  '2': 'Північ',
  '3': 'Південь',
  '4': 'Азія',
  '5': 'Азія'
};

net.train([ {input: [5,2,5,4,2,4,1,2,5,5,1,5], output: [0.0]},
```

Рисунок 4.8 – Частина коду сервера

Функція train відповідає за тренування нейронної мережі даними.

На рис. 4.9 зображено фрагмент коду сервера, відповідального за обробку запиту на результат тестування.

```
app.get('/result', (req, res) => {
  var output = net.run(req.query.answers);
  console.log(country[Math.round(output*10)] );
  res.send(country[Math.round(net.run(req.query.answers)*10)]);
});
```

Рисунок 4.9 – Код обробки запиту результату тесту

Функція run модуля Brain відповідає за запуск навченої нейронної мережі для отримання результату тестування.

Висновки до 4 розділу

На основі усіх наведених у попередніх розділах міркувань, спостережень та висновків, було створено вебзастосунок туристичного туру, який містить в собі тест для автоматичного підбору туристичного туру на основі навченої нейронної мережі за індивідуальними вподобаннями відвідувача. Даний тест складається з дванадцяти питань. Як і будь-який інший сайт, цей складається з інтерфейсної та серверної частини. Створений вебзастосунок відповідає усім вимогам, які були поставлені у ході проектування системи.

Інтерфейс готового вебсайту має п'ять розділів: «Головна», «Каталог», «Контакти», «Відгуки», «Де відпочити?». Кожен розділ містить інформацію, що відповідає його назві. Наприклад, розділ «Контакти» включає в себе карту з місцем розміщення офісу, телефони та посилання на сторінки у соціальних мережах.

Серверна частина відповідає за обробку запитів, виконання обчислень та формування відповідей клієнту (браузеру). У цьому допомагають модулі. Модуль Express відповідає за створення сервера, обробку запитів. Brain допомагає створити нейронну мережу та навчити її. Модуль MongoDB забезпечує взаємодію з сервером бази даних та виконання запитів до нього.

Використана бібліотека React забезпечує зручний та інтуїтивно зрозумілий інтерфейс. За допомогою цього інструмента, код програми має структуру, яка дозволяє легко підтримувати його.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Спеціальний розділ

ОХОРОНА ПРАЦІ

до кваліфікаційної роботи

на тему:

**«Інформаційний веб-застосунок підбору туристичного
туру за індивідуальними вподобаннями на основі
нейронних мереж»**

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.21810217

Виконала студентка 4-го курсу, групи 402

_____ *Луцишина В.В.*

(підпис, ініціали та прізвище)

«__» _____ 2022 р.

Консультант _____ *к.т.н., доцент*

(наук. ступінь, вчене звання)

_____ *А. О. Алексеева*

(підпис, ініціали та прізвище)

«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ВСТУП.....	68
5.1 Нормативно-правові акти експлуатації комп'ютера	69
5.2 Загальні правила безпечної роботи за комп'ютером	72
5.3 Наслідки неправильної роботи за комп'ютером.....	74
5.4 Комплекс вправ для профілактики захворювань опорно-рухового апарату та очей	75
Висновки до розділу 5.....	76

ВСТУП

Сьогодні важко собі уявити будь-який державний чи підприємницький заклад, який не містить комп'ютера чи ноутбука. У закладах освіти повинне бути обладнання для директора, секретаря, вчителя, а також комп'ютерний клас для дітей. У судових установах зазвичай працюють секретарі, які повинні документувати усі свідчення та матеріали справ. Заклади медицини не зможуть надавати допомогу пацієнтам, адже без комп'ютера неможливо наразі виписати рецепт, призначити прийом до лікаря тощо.

Будь-яка компанія, фірма, підприємство повинно мати хоча б одного бухгалтера, який повинен вести справи. В електронному вигляді це робити швидше та простіше. Такі програмні засоби як Word, Excel та інші допомагають багатьом працівникам.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Ця система створена для того, аби забезпечити гідні та безпечні умови праці для кожної людини. Вона передбачає ряд нормативно-правових документів, які визначають правила поведінки з технікою, користування виробничим приміщенням, поведінки під час надзвичайних ситуацій, а також систему профілактично-лікувальних заходів для попередження негативного впливу на здоров'я людини. [25]

Особливим розділом охорони праці є охорона праці за комп'ютером. Він визначає правила безпечного поведінки з електронно обчислювальною технікою, а також методи уникнення негативного впливу іонізуючого впливу на здоров'я людини. Усі ці заходи допомагають забезпечити комфортну та ефективну роботу працівників.

5.1 Нормативно-правові акти експлуатації комп'ютера

З метою створення належних, безпечних і здорових умов праці при роботі з екранними пристроями роботодавець має керуватися наступними нормативно-правовими актами:

– наказом Мінсоцполітики «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» від 14.02.2018 р. № 207, який поширюється на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми і видів діяльності та встановлює мінімальні вимоги безпеки та захисту здоров'я під час здійснення роботи, пов'язаної з використанням екранних пристроїв незалежно від їхнього типу та моделі.

– Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затвердженими постановою Головного державного санітарного лікаря України 10.12.1998 р. №7, що застосовуються на підприємствах, організаціях, установах незалежно від форми власності та поширюються на умови й організацію праці при роботі з візуальними дисплейними терміналами усіх типів вітчизняного та закордонного виробництва на основі електронно-променевих трубок, якими укомплектовані електронно-обчислювальні машини колективного використання та персональні ЕОМ.

Згідно з п. 5 наказу Мінсоцполітики від 14.02.2018 р. № 207, для досягнення зазначеної мети роботодавець повинен за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку працівників.

Перерви встановлюються відповідно до Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98, затверджених постановою

Головного державного санітарного лікаря України від 10 грудня 1998 року №7.

Наприклад, для розробників програм має бути встановлено перерву для відпочинку:

- тривалістю 15 хвилин через кожну годину роботи за комп'ютером;
- для операторів ЕОМ – 15 хвилин через кожні 2 години;
- для операторів комп'ютерного набору – 10 хвилин після кожної години роботи за комп'ютером.

Якщо виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи за комп'ютером не повинна перевищувати 4 годин.

При 12-годинній робочій зміні регламентовані перерви мають установлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4 годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин. [26]

Вимоги безпеки до робочих місць працівників з екранними пристроями:

- робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.

– для забезпечення безпеки та захисту здоров'я працівників усе випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня (вплив на людину факторів довкілля - шуму, вібрації, забруднювачів, температури тощо, який не спричиняє соматичних або психічних розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій) з погляду безпеки та охорони здоров'я працівників.

- організація робочого місця працівника з екранними пристроями має забезпечувати відповідність усіх елементів робочого місця та їх розташування

ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.

– освітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном і навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.

– мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями має підтримуватись на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 42 (далі - ДСН 3.3.6.042-99).

– робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.

– робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення. Сидіння має регулюватися по висоті, спинка сидіння - як по висоті, так і по нахилу. Слід передбачати підніжку для тих, кому це необхідно для зручності.

Ці вимоги не поширюються на:

– робочі місця здобувачів освіти у комп'ютерних класах (кабінетах, аудиторіях тощо) закладів освіти;

– робочі місця пілотів, водіїв або операторів транспортних засобів, обладнані екранними пристроями у системах оброблення даних на борту засобів сполучення, екранні пристрої у складі машин і обладнання, що переміщуються в процесі роботи. [27]

5.2 Загальні правила безпечної роботи за комп'ютером

Деякі принципи правильної роботи за комп'ютером:

- у робочому приміщенні (кімнаті), де встановлені комп'ютери, щодня потрібно виконувати вологе прибирання;
- приміщення, у якому знаходяться комп'ютери, потрібно провітрювати щогодини;
- після кожної години роботи рекомендується робити десяти хвилинну перерву, яку зручно суміщати з провітрюванням. За будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин. Під час перерви не варто читати або дивитися телевізор. Перерва, яку ви проводите за комп'ютером (наприклад, граючись або шукаючи матеріали в Інтернеті), просто не має сенсу;
- необхідно постійно слідкувати за станом екрану монітора: він має бути чистим, без плям та пилу. Крім того, обов'язково слідкуйте за чистотою окулярів – комп'ютерних чи звичайних;
- слідкуйте за поставою: ноги твердо стоять на підлозі чи на спеціальній підставці; стегна розташовані під прямим кутом до тулуба, а гомілки – під прямим кутом до стегон; сидіти потрібно прямо або злегка нахилившись вперед; пальці рук знаходяться на рівні зап'ястків або трохи нижче – у такому положенні вони найбільш рухливі; плечі мають бути розслаблені та вільно опущені, що сприяє розслабленню рук; відстань від очей до екрану монітора – не менше 55-60 см; центр екрану має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;
- щоб попередити „синдром сухого ока”, кліпайте очима кожні 3-5 секунд;
- у процесі роботи за комп'ютером обов'язково звертайте увагу на дихання: воно має бути рівномірним, без затримок;

- при роботі з текстом рекомендується, щоб колір шрифту був темним, а колір фону – світлим (ідеальний варіант – чорний шрифт на білому фоні);
- якщо шрифт занадто малий, то потрібно збільшити масштаб документу (наприклад, до 150% чи більше);
- при наборі текстів з паперів чи книг рекомендується помістити джерело якомога ближче до монітору. Це дозволить уникнути частих рухів головою та очима;
- якщо є можливість, змінійте вид діяльності, якою займаєтеся протягом дня;
- у процесі роботи рекомендується періодично (приблизно раз на 20-30 хвилин) переводити погляд з екрану на найбільш віддалений предмет у кімнаті, а ще краще – на віддалений об'єкт за вікном;
- якщо з'явилося відчуття втоми, напруження, сонливості, тяжкості в очах, потрібно припинити роботу та хоча б трохи відпочити;
- сидіть глибоко на твердому стільці з високою спинкою, що має вигин для попереку, - це вирівняє спину і дасть підтримку шії. Край стільця не повинен тиснути на судини під колінами;
- правильно організуйте освітлення робочого місця. При слабкому світлі очі напружуються і болять. Зменште яскравість екрану. Літери і цифри на екрані це маленькі світлові промені, які йдуть прямо в очі. Потрібно відрегулювати їх контрастність, щоб світло не було дуже яскравим;
- переміщуйте погляд по всій площі екрану, намагайтеся не дивитися напружено в одну точку. Нехай по черзі працюють всі м'язи очей, а не окремі групи, на які в цьому випадку буде падати максимальне навантаження. [28]

5.3 Наслідки неправильної роботи за комп'ютером

Незважаючи на видиму безпеку та розвитку сучасних технологій, при роботі за комп'ютером є ряд чинників, які можуть вплинути на здоров'я людини. Усі негативні чинники під час роботи за комп'ютером можна поділити на такі основні групи:

- чинники, що впливають на опорно-руховий апарат.
- чинники, що впливають на сенсорні органи.
- чинники, що впливають на психологічний стан.

Всі ці чинники викликані двома основними причинами: неправильною роботою за обладнанням і неправильним вибором обладнання.

Хвороби, що можуть виникнути у разі недотримання правил роботи за комп'ютером:

– неправильна постава - це призводить до подальшого розвитку викривлення хребта сколіозу, лордозу, кіфозу, а як результат головні болі, болі в області шиї і всього хребта, болі в області таза. Неправильне положення ніг може призвести до артриту (запалення суглобів), артрозу (деформації).

– тунельний синдром - найвідоміше захворювання людей, які працюють за комп'ютером. Воно ж - синдром зап'ястного каналу. Тунельний синдром проявляється після кількох годин напруженої роботи за комп'ютером. Синдроми - почуття «бігають мурашки» по кисті, біль пронизує кисть, оніміння кисті, важкість у руці, знесилення кисті. Тунельний синдром створює травму зап'ястя. Через зап'ястний канал між кістками (тунель) проходять серединний нерв і 9 сухожиль м'язів кисті. Серединний нерв забезпечує чутливість пальців, а також управляє м'язами, що забезпечують рух великого, вказівного і середнього пальців. Сам тунельний канал дуже вузький. Він стискає, защемляє серединний нерв. Під час частих, повторюваних рухів кистей рук в незручному положенні сухожилля труться об кістки зап'ястя і

зв'язки. Постійно повторювані малі рухи пальцями призводять до внутрішніх мікротравм. Накопичуючись, вони і дають про себе знати в початковій стадії хвороби тремтінням, сверблячкою, набряком і поколювання в пальцях.

– геморої - хвороба аналогічна варикозному розширенню вен, тільки в більш інтимному місці. Причиною хвороби може бути застій крові у венах.

– розвиток короткозорості - через те, що екран монітора за контрастом вище, ніж навколишні об'єкти розвивається короткозорість. Це захворювання органів зору, яке характеризується нечітким сприйняттям предметів на далекій відстані. При цьому предмети поблизу розрізняються добре і чітко. Короткозорість трапляється тому, що картинка формується перед сітківкою ока, в той час, коли вона повинна формуватися на самій сітківці.

– порушення фокусування - наслідком напруженої роботи за монітором є порушення фокусування, яке може бути викликано перенапруженням очних м'язів.

– сухість очей - через рефлекс, заснований на тому, що при погляді на джерело світла очі починають менше кліпати, і виникає «обсушування» рогівки ока, яке призводить до болі в очах.

5.4 Комплекс вправ для профілактики захворювань опорно-рухового апарату та очей

Аби запобігти вищенаведеним хворобам є кілька вправ:

– нахиліть голову, дістаючи підборіддям груди. Нахиливши голову вперед, повільно повертайте її вправо і вліво кілька разів.

– переміщайте голову назад, притискаючи до грудей підборіддя.

– тримаючи руки на стегнах, зводите і розводите лопатки.

– покладіть ногу на ногу й ліктем протилежної руки зіпріться на зовнішню поверхню стегна. Ведіть плече вперед, до упору скручуючи хребет.

Помінявши ногу, виконуйте той же рух в інший бік.

– рух очей по вертикалі вгору – вниз.

– рух очей по горизонталі вліво - вправо.

– рух очей по діагоналі лівий нижній кут - правий верхній кут.

– рух очей по діагоналі лівий верхній кут - правий нижній кут.

– обертання очей за годинниковою стрілкою (уявіть перед собою великі годинники і переміщайтеся очима у слід за стрілкою).

– обертання очей проти годинникової стрілки (уявіть перед собою великі годинники і переміщайтеся очима у слід за стрілкою).

– піднесіть палець до носа і подивіться на нього протягом декількох секунд.

– швидко кліпайте очима протягом декількох (3-5) секунд. Як варіант можна виконувати цю вправу по разі між попередніми.

– стисніть очі з усіх сил. Розслабте. Допомагає при сухості в очах.

– прикладіть долоні до очей так, щоб пальці рук перетиналися на лобі.

При цьому перевірте, щоб не було видно світла.

Висновки до розділу 5

Охорона праці є важливим компонентом безпечної праці. Це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Вона визначає нормативно-правові акти експлуатації приладів, виробничих приміщень, безпеки під час аварій або інших надзвичайних

ситуацій. Також вона передбачає заходи, які спрямовані на попередження захворювань, травм тощо.

Роботодавці повинні користуватися нормативно-правовими актами охорони здоров'я з метою забезпечення гідних умов праці для кожного працівника фірми, компанії, підприємства, а саме: наказом Мінсоцполітики «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» від 14.02.2018 р. № 207 та Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затвердженими постановою Головного державного санітарного лікаря України 10.12.1998 р. №7. Ці документи визначають правила безпечного користування електронними приладами, мінімальні правила безпеки, а також розпорядок приміщення, який повинен забезпечити роботодавець.

У разі недотримання правил поведінки з електронною технікою або неправильно обраним приладом можуть виникнути хвороби опорно-рухового апарату та очей: сухість очей, викривлення хребта, тунельний синдром, розвиток короткозорості тощо. Для попередження таких захворювань існують спеціальні вправи для м'язів опорно-рухового апарату та очей.

ВИСНОВКИ

Отже, у ході виконання бакалаврської дипломної роботи було досліджено роботу нейронних мереж, їх сфери застосування, види та дано оцінку їх актуальності. Дано визначення поняттям «нейронні мережі», «вебсайт». Було проведено огляд роботи мережі інтернет, основних протоколів передачі даних.

Майбутня програма була спочатку спроектована, а потім реалізована. У ході проектування вдалося дослідити наявні інструменти для створення вебзастосунків, виявити їх переваги та недоліки. У якості СКБД було обрано MondoDB, яка вирізняється простим синтаксисом та гнучкою структурою для зберігання даних. JavaScript – мова, яка була обрана через її можливість працювати як на серверній частині, так і створювати інтерфейс програми, що значно скорочує час та спрощує код вебзастосунку. Програмне середовище Node.js забезпечує виконання коду JavaScript на стороні серверу. HTML та CSS є базовими інструментами для розробки сторінок вебсайту.

Нейронні мережі є математичною моделлю реальної системи нервових клітин людини. Основною функцією мозку людини є здатність до навчання, розпізнавання. Штучні нейронні мережі були розроблені саме для імітації цих процесів. Навчаючи нейронні мережі, науковці отримують систему, здатну класифікувати, розпізнавати, передбачати майбутній крок.

У ході реалізації вебсайту було проведено пошукову роботу, досліджено різні джерела інформації. Було покращено навички програмування, повторено та закріплено вже пройдений матеріал на базі чотирьох років навчання. У результаті виконання роботи вдалося реалізувати інформаційний вебзастосунок підбору туристичного туру за індивідуальними вподобаннями на основі нейронних мереж.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Літаючи в хмарах: як NETFLIX запусив штучний інтелект на AWS. *КОМПЬЮТЕРРА*: вебсайт. URL: <https://www.computerra.ru/182669/vitaya-v-oblakah-kak-netflix-zapustil-iskusstvennyiy-intellekt-na-aws/> (дата звернення: 29.05.2022).
2. Сфери застосування штучного інтелекту. *AICONFERENCE*: вебсайт. URL: <https://aiconference.com.ua/uk/news/oblasti-primeneniya-iskusstvennogo-intellekta-92253> (дата звернення: 29.05.2022).
3. Як штучний інтелект та роботи підвищують врожайність полів та ефективність сільського господарства. *Аграрії разом*: вебсайт. URL: <https://agrarii-razom.com.ua/article/roboti-i-polunichka-yak-ai-pidvishue-vrojajnist-poliv> (дата звернення: 29.05.2022).
4. Google створила нейромережу для визначення прогнозу погоди на 8 годин вперед. *Голос Карпат*: вебсайт. URL: <https://goloskarpat.info/rus/tech/5e7dd0656829b/> (дата звернення: 29.05.2022).
5. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення: монографія. Харків: Фабула, 2019. 416 с.
6. Комп'ютерні мережі. Книга 1 / Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Львів: Магнолія 2006, 2013. 256 с.
7. Чи може машина мислити? *Хабр*: вебсайт. URL: <https://habr.com/ru/post/51521/> (дата звернення: 29.05.2022).
8. Руденко О.Г., Бодянський Є.В. Штучні нейронні мережі: навч. посіб. Київ: Компанія СМІТ, 2006. 404 с.
9. Хайкін С. Нейронні мережі: монограма. Діалектика, 2019. 1104 с.
10. Основні типи нейронних мереж та їх застосування — Навчальний посібник. *Towards AI*: вебсайт. URL: <https://towardsai.net/p/machine->

[learning/main-types-of-neural-networks-and-its-applications-tutorial-](#)

[734480d7ec8e](#) (дата звернення: 30.05.2022).

11. Що таке нейромережі (НМ) та як вони працюють. *Реобзор*: вебсайт.
URL: <https://reobzor.ru/reviews/2020/12/что-такое-нейросети-и-как-они-работают> (дата звернення: 30.05.2022).

12. Що таке JavaScript і чому Gmail його блокує? *Gadgetshelp.com*: вебсайт. URL: <https://gadgetshelp.com/windows/что-такое-javascript-i-pochemu-gmail-ego-blokiрует/> (дата звернення: 30.05.2022).

13. Введення в JavaScript. *Сучасний підручник JavaScript*: вебсайт. URL: <https://learn.javascript.ru/intro> (дата звернення: 30.05.2022).

14. Дакетт Д. HTML і CSS. Розробка і дизайн вебсайтів: монограма. Ексмо, 2022. 480 с.

15. Блокова модель. *Metanit*: вебсайт. URL: <https://metanit.com/web/html5/6.9.php> (дата звернення: 30.05.2022).

16. Мейер Е. А., Уейл Е. CSS повний довідник: довідник. Діалектика, 2019. 1088 с.

17. CSS синтаксис та селектори. *Html5css*: вебсайт. URL: https://html5css.ru/css/css_syntax.php (дата звернення: 01.06.2022).

18. Node.js Вступ. *W3schools*: вебсайт. URL: https://www.w3schools.com/nodejs/nodejs_intro.asp (дата звернення: 01.06.2022).

19. Пауерс Ш. Вивчаємо Node.js : монограма. Пітер, 2014. 400 с.

20. Що таке MongoDB. *Metanit*: вебсайт. URL: <https://metanit.com/nosql/mongodb/1.1.php> (дата звернення: 01.06.2022).

21. React – JavaScript бібліотека для створення інтерфейсів користувача. *React*: вебсайт. URL: <https://ru.reactjs.org/> (дата звернення: 01.06.2022).

22. Що таке SPA. *Netpeak Journal*: вебсайт. URL: <https://netpeak.net/ru/blog/kak-optimizirovat-spa-sayty/> (дата звернення: 01.06.2022).

23. Heaton J. Introduction to Neural Networks with Java: монограма. Heaton research inc, 2008. 380 с.

24. Нейронна мережа зворотного поширення: що таке алгоритм зворотного поширення в машинному навчанні? *Guru99*: вебсайт. URL: <https://www.guru99.com/backpropagation-neural-network.html> (дата звернення: 01.06.2022).

25. Навчання з питань охорони праці на підприємствах міста Запоріжжя. *Zp.gov*: вебсайт. URL: https://zp.gov.ua/upload/editor/navchannya_z_pitan_ohoroni_praci.pdf (дата звернення: 02.06. 2022).

26. Ви маєте створити безпечні умови праці за комп'ютером для своїх працівників – ПОН. *MCFR*: вебсайт. URL: <https://www.kadrovik01.com.ua/news/7219-vi-mate-stvoriti-bezpechn-umovi-prats-za-kompyuterom-dlya-svoh-pratsvnykv-pon> (дата звернення: 02.06.2022).

27. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями: наказ М-ва соціальної політики України від 14.02.2018 р. № 207. Загальні положення.

28. Правила безпечної роботи на комп'ютері. *Білоцерківський гуманітарно-педагогічний фаховий коледж*: вебсайт. URL: <https://pedcollege.kiev.ua/index.php/77-roboty-koledzhu/okhorona-pratsi/589-pravyla-bezpechnoi-roboty-na-kompiuteri> (дата звернення: 02.06.2022).